CSE 515: Multimedia and Web Databases
Phase 2 - Group 10

“Dimensionality Reduction Of Features”

[1]Mansi Panpaliya(1217179197),[2]Rekha Yale Gnanaprakash(1217158670),[3]Harika Kondru(1217032934), [4]Asmi Pattnaik(1217240531),[5]Anantha Krishnan Kumar(1217112208),[6]Arijit Panda(1217155056)

Abstract: Retrieval of relevant data is one of the most primary concerns in the field of multimedia. A multimedia object contains many different features associated with it. Of these features, few are salient, meaning they are enough to identify the object containing them. Reducing the dimensions of the features of the multimedia objects improves the processing involved for various operations. In this phase of the project, dimensionality reduction techniques such as PCA, SVD, LDA, and NMF are applied on an image dataset stored in the filesystem and then operations involving similarity and relevance are performed.
Keywords: PCA, SVD, LDA, NMF, data, Euclidean, Similarity, Cosine, Latent Semantics, LBP, HOG, Color Moments, SIFT, Dimensionality Reduction.

1. Introduction

This is the continuation of phase 1 where we experimented with image datasets and found similar images based on distance/similarity measure using different feature descriptor models (LBP, HOG, SIFT, Color Moment).
The main objective of this phase of the project is to reduce the dimensions of the features of the images in the given dataset and improve the processing for operations like similarity calculation and labeling, using the top latent semantics. The similarity is based on the similarity or distance measures between the feature descriptors extracted from the images. The dataset used in this project is provided by ‘Mahmoud Afifi, "11K Hands: Gender recognition and biometric identification using a large dataset of hand images."
Multimedia Tools and Applications, 2019.’, which contains 11,076 hand images (1600 x 1200 pixels) of 190 subjects, of varying ages between 18 - 75 years old. The techniques, PCA, SVD, LDA and NMF are applied to the image dataset to reduce the dimensions of the features and measures like Euclidean distance and Cosine similarity are used to find out and retrieve images that are similar to a given image.

1.1 Terminology

1.1.1 Local Binary Patterns (LBP)

“Local Binary Patterns (LBP) is a non-parametric descriptor whose aim is to efficiently summarize the local structures of images(Huang, Shan et al 2017:1, https://hal.archives-ouvertes.fr/hal-01354386/document)[1].”

1.1.2 HOG

Histogram of oriented gradients (HOG) is a feature descriptor used to detect objects in computer vision and image processing. The HOG descriptor technique counts the occurrences of gradient orientation in localized portions of an image - detection window, or region of interest (ROI), "Developer reference for Intel Integrated Performance Primitives 2018:1[2]"

1.1.3 Euclidean Distance

The Euclidean distance between two points a and b is the length of the line segment connecting them, "*Euclidean distance",* Wikipedia, https://en.wikipedia.org/wiki/Euclidean_distance [3].

1.1.4 Cosine Similarity

The cosine similarity between two vectors is the measure of the cosine of the angle between them. It is a measure of orientation and not magnitude.

1.1.5 SIFT

The scale-invariant feature transform (SIFT) is a feature detection algorithm in computer vision to detect and describe local features in images. *"Scale-invariant feature transform",* Wikipedia, https://en.wikipedia.org/wiki/Scale-invariant_feature_transform [4]

1.1.6 Color Moment

Color moments are the evaluation of the color distribution in an image e.g mean, variance, skew, etc. It works well under different lighting conditions as it encodes shape and color. https://en.wikipedia.org/wiki/Color_model [5]

1.1.7 PCA

For a given dataset, PCA uses the covariance matrix and identifies alternative bases along which spread is maximum. It is based on Eigen Decomposition.

1.1.8 SVD

SVD is directly applied to the *Object-Feature* matrix itself. $S = PCP^{-1}$, *where S* is an original object-feature matrix, *C* is the diagonal matrix of eigenvalues and *P* is an orthonormal matrix consisting of the eigenvectors of *S*.

1.1.9 LDA

LDA or Latent Dirichlet Allocation is a generative probabilistic model for collections of discrete data. It is a three-level Bayesian model. LDA is a three-level hierarchical Bayesian model, in which each item of this collection is modeled as a finite mixture over an underlying set of topic probabilities[6].

1.1.10 NMF

Non-negative Matrix Factorization is a technique for finding the parts-based, linear representations of non-negative data[7]. Geometrically, it is like a cone which contains a cloud of data points and which is contained in the positive orthant.[8]

1.2 Goal Description

This phase of the project has 8 tasks that deal with Image Features, Vector Models and Dimensionality Reduction.
The detailed specification of every task is described below:

Task 1

Implement a program which (a) lets the user choose among one of the four feature models from Phase 1 and (b) given positive integer value, k identifies and reports the top-k latent semantics in the corresponding vector space using (c) one of the following techniques chosen by the user:

– principal component analysis (PCA),
– singular value decomposition (SVD),
– non-negative matrix factorization (NMF)
– latent Dirichlet analysis (LDA).

Task 2

Implement a program which (a) lets the user to choose among one of the four feature models and (b) given the top-k latent semantics for that feature model, created using (c) a dimensionality reduction technique chosen by the user, and given (d) an image ID, the system identifies the most related m images in the k-dimensional latent space (list also the matching scores).

Task 3

Implement a program which (a) lets the user choose among one of the four feature models and (b) given one of the labels:

– left-hand,
– right-hand,
– dorsal,
– palmar,
– with accessories,
– without accessories,
– male, or
– female

identifies (and lists) k latent semantics for images with the corresponding metadata using (c) one of the following techniques chosen by the user:

– principal component analysis (PCA),
– singular value decomposition (SVD),
– non-negative matrix factorization (NMF), or – latent Dirichlet analysis (LDA).

Task 4

Implement a program which (a) lets the user choose among one of the four feature models and (b) one of the four techniques (PCA, SVD, NMF, or LDA) and (c) given the k latent semantics associated with one of the labels:

– left-hand,
– right-hand,
– dorsal,
– palmar,
– with accessories,
– without accessories,
– male, or
– female

and (d) given an image ID, identifies the most related m images using these k latent semantics (the list also the matching scores).

Task 5

Implement a program which (a) lets the user choose among one of the four feature models and (b) one of the four techniques (PCA, SVD, NMF, or LDA) and (c) given the k latent semantics associated with one of the labels:

– left-hand,
– right-hand,
– dorsal,
– palmar,

– with accessories,
– without accessories,
– male, and
– female and

(d) an unlabeled image ID, the system labels it as

– left-hand vs right-hand,
– dorsal vs palmar
– with accessories vs. without accessories
– male vs. female

Task 6

Implement a program which given a subject ID, identifies and visualizes the most related 3 subjects.

Task 7

Implement a program which, given a value k:

– creates a subject-subject similarity matrix, – performs NMF on this subject-subject similarity matrix, and
– reports the top-k latent semantics.

Task 8

Implement a program which, given a value k:

– creates a binary image-metadata matrix,
– performs NMF on this image-metadata matrix, and
– reports:
top-k latent semantics in the image-space.
top-k latent semantics in the metadata-space.

1.3 Assumption

- Prerequisite knowledge of Python programming language.
- The dataset would not change or update once the feature descriptors are computed for the images.
- The total number of similar images to be displayed should be less than the number of images in the dataset.
- We are assuming that the tasks will be run in the numeric order as the input for a task is the output of the previous task.
- The dataset on which this is computed should come with its own metadata i.e., Each image in the dataset should be labeled (Male/Female, Left Hand/Right Hand, etc)
- The test image given during the evaluation of any task should be 1. Either from the given test data 2. Or should be given before running task1.

2. Problem Solution

Task 1

SIFT

In sift, we get a 2d matrix for each image. We had to pass in a 2d matrix, which contains the feature descriptor of all the images. So we used an approach to achieve this. First, we computed the number of descriptors for all the images in the folder. Then we selected the descriptor that consisted of the least amount of key points. We decided to reduce all the descriptors to that size of minimum keypoints. The reason being that we get descriptors of similar size instead of it being a ragged one. Because the matrix/feature descriptor that we are going to pass needs to be of equal size to the technique, we reduce the size of descriptors by taking each row in the minimum sized descriptor and computing the distance to each row of all the other image descriptors. By doing so, we calculate the descriptor that has the minimum distance to each row in the descriptor of the image that has the minimum key points. That way, we select 1 row in the descriptor for each row in the descriptor with minimum key points. So we eliminate the other rows that have non-similar and irrelevant data like background screen. So this way get n row descriptors for all the images. After this, we combine n rows and convert it to a single row and then pass that data to any technique in phase1.

As a part of this task, we implemented a program that uses the feature descriptors of one of the feature models from phase 1 and outputs the k-latent semantics using one of the following techniques: SVD, PCA, NMF, and LDA.

For SVD, initially, we used the Truncated-SVD but the library generates an error when k is equal to n(total number of features) which is why we went ahead with linalg() which has no tight bound that k should be less than n.

For PCA, we use Scikit-learn which calculates the covariance matrix and then processes the eigendecomposition. StandardScaler() standardizes the features by removing the mean and scaling to unit variance.

For LDA, we use sklearn's LatentDirichletAllocation which is similar to SVD or PCA but this uses a probabilistic model for calculating features.

For NMF, we use sklearn's NMF(). We calculate features by finding the parts-based, linear representations of non-negative data. As NMF works only on non-negative data, we shift the values in the matrix such that the smallest number is shifted to '0' if and only if there are negative numbers in the matrix.

In this task, we are reading the feature descriptors, the result of phase 1, from a CSV file and applying a dimensionality reduction technique and sorting them in decreasing order of term-weight pairs and storing the resulting matrix in a CSV file.

When we decompose a feature descriptor matrix we obtain a diagonal "core" matrix indicating of the latent semantics of the data and two "factor" matrices (data-latent semantics, U, and feature-latent semantics, V) that describe the discovered latent semantics in terms of the data (for U) and features (for V).

We have implemented a latent semantic visualizer where:

- for data latent semantic, a ranked list of image thumbnails along with their scores for each latent semantics is created.
- for feature latent semantic, a ranked list of images with the highest dot product with the latent semantics is created.

Task 2

As a part of this task, given an image ID, we are supposed to find the "m" number of similar images.

To find similar images, we calculated the Euclidean Distance between the decomposed data latent semantic(that we have found in task 1) of the given image and that of other images in the dataset. To find the top "m" similar images, images corresponding to "m" least distances are taken.

We tested both Euclidean distance and Cosine Similarity to find similar images and they gave very similar results but cosine similarity took significantly more time when we computed the task for a large dataset which is why we went ahead with Euclidean Distance to measure distance as a measure to find similar images.

Task 3

As part of this task, we implemented a program that uses the feature descriptors of one of the feature models from phase 1 and outputs the k-latent semantics of the “labeled data” using one of the following techniques: SVD, PCA, NMF, and LDA. Here, labeled data refers to data that belongs to one following label: “Left hand”, “Right hand”, “Dorsal, Palmar”, “ With Accessories”, “Without Accessories”, “ Male” and “Female”. The label is passed as an argument. Similar to task 1, here, when we decompose a feature descriptor matrix we obtain a diagonal "core" matrix indicating of the latent semantics of the data and two "factor" matrices (data-latent semantics, U, and feature-latent semantics, V) that describe the discovered latent semantics in terms of the data (for U) and features (for V).

We implemented a latent semantic visualizer where:

- for data latent semantic, a ranked list of image thumbnails along with their scores for each latent semantics is created.
- for feature latent semantic, a ranked list of images with the highest dot product with the latent semantics is created.

Task 4

As a part of this task, given an image ID, we are supposed to find “m” number of similar images using k-latent semantics associated with one of the labels (left hand, right hand, dorsal, palmar, with accessories, without accessories, male, female). To find similar images, we calculated the Euclidean Distance between the decomposed data latent semantic(that we have found in task 3) of the given image and that of other images in the dataset. To find the top “m” similar images, images corresponding to “m” least distances are taken. We tested both Euclidean distance and Cosine Similarity to find similar images and they gave very similar results but cosine similarity took significantly more time when we computed the task for a large dataset which is why we went ahead with Euclidean Distance to measure distance as a measure to find similar images.

Task 5

To successfully complete this task, we implemented a program that uses feature descriptors of labeled data. Using a dimensionality reduction technique we fit a model on the training data(labeled data). Assuming that the test image is in the complete dataset, we perform dimensionality reduction on the whole data set. The Euclidean distance between the data semantic between the test image and all other images in the dataset is calculated. These distances are sorted in increasing order. Then the labels of the top 7 images are analyzed and based on the label with the highest count, the image is identified whether it can be labeled or not. With the trial and error method, we identified the value for choosing the top images for comparison. Value 7 gave us good results and hence this was finalized. We also analyzed a couple of other methods. One of which computed the centroid of the complimentary labels(eg left and right hand), computed the centroid of each label. Then we calculated the distance of the test image with these centroids and whichever distance was low, assigned it the corresponding label. Also, we analyzed the method which calculated the 2 farthest points. Then the distance between the test image and the farthest points is calculated. If these distances are less than the distance between the farthest point then the image is labeled. But these techniques failed when the data spread is uneven along the axis. Also, as centroid and the farthest points represent the data very sparsely, these approaches may not be a good choice. We found some exceptions to our proposed approach. There is a misclassification of images in some cases. For example, in dorsal and palmar data, if the hand is fair in color the image is not labeled appropriately.

Task 6

As a part of this task, we implemented a program that given a subject ID return the most similar subjects in the given dataset. First, we take the latent semantics (‘u’) which resulted from task1 using one of the techniques: SVD, PCA, NMF, and LDA. We calculate the subject-subject similarity matrix. We find all the images for a particular subject. Then we take the column-wise mean for all the images and a single row for a subject is created from factor matrix ‘U’. The distance between each and every subject is calculated and accordingly, the subject-subject similarity matrix is created. Here we compute mean as it represents complete data for a particular subject. Once the new latent semantics for all the subject ID is formed we compute the Cosine Similarity between all the subject IDs which results in the subject-subject similarity matrix. Now we select the given subject ID and compute the top 3 similar subjects after sorting the data.

Task 7

As a part of this task, we select the subject-subject similarity matrix computed in task 6. We pass this subject-subject similarity matrix to NMF. We obtain a diagonal "core" matrix indicating of the latent semantics of the data and two "factor" matrices (data-latent semantics, U, and feature-latent semantics, V) that describe the discovered latent semantics in terms of the data (for U) and features (for V).

We implemented a latent semantic visualizer where:

- For the subject latent semantic, a ranked list of subjects along with their weights for each latent semantics is created.
- For latent semantic subjects, a ranked list of subjects with the weights for each latent semantics is created.

Task 8

As a part of this task, we implemented a program to create a binary image-metadata matrix by using the metadata of the images. For every label “Left hand, Right hand, Dorsal, Palmar, With Accessories, Without Accessories, Male and Female” a series array of binary values of each label is calculated for each image. A data-frame to represent these binary values of every label for each image is created. As NMF function is created in task-1, this data-frame matrix of binary image-metadata matrix and the value of k-latent semantics is passed as an argument to the function. Here, when we decompose a matrix we obtain a diagonal "core" matrix indicating of the latent semantics of the data and two "factor" matrices (data-latent semantics, U, and feature-latent semantics, V) that describe the discovered latent semantics in terms of the data (for U) and features (for V). As the output, we create an HTML for the top k latent semantics in image-space and metadata-space in decreasing order of weights. For image space, the factor matrix ‘U’ is sorted in decreasing order of weights along the column and for metadata-space, the factor matrix ‘Vt’ is sorted in decreasing order of weights along the rows and the result is displayed accordingly.

2.1 Sample and Query Output

Task 1

Query 1 :

Model : SIFT k: 20 Technique: Singular value decomposition (SVD)

Refer to the file datasemantic_task1_query1.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task1_query1.html in the folder Output/HTML Files.

Query 2 :

Model : SIFT k: 40 Technique : Singular value decomposition (SVD)

Refer to the file datasemantic_task1_query2.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task1_query2.html in the folder Output/HTML Files.

Query 3 :

Model: Color Moments k: 20 Technique: LDA

Refer to the file datasemantic_task1_query3.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task1_query3.html in the folder Output/HTML Files.

Query 4 :

Model: Color Moments k: 40 Technique: LDA

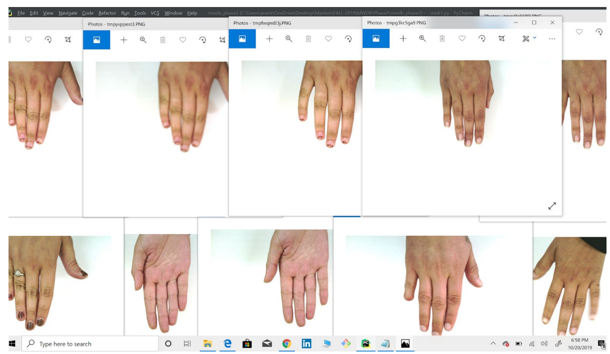
Refer to the file datasemantic_task1_query4.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task1_query4.html in the folder Output/HTML Files.

Query 1 :

K : 10 M : 10 Image ID : Hand_0000111.jpg
Technique : Principal component analysis (PCA)

ImageId:Hand_0000111.jpg Score:0.000000
ImageId:Hand_0000112.jpg Score:5.925817
ImageId:Hand_0000108.jpg Score:6.846490
ImageId:Hand_0001145.jpg Score:17.547246
ImageId:Hand_0001144.jpg Score:19.311032
ImageId:Hand_0001091.jpg Score:32.391096
ImageId:Hand_0001154.jpg Score:35.304660
ImageId:Hand_0001153.jpg Score:36.558831
ImageId:Hand_0000990.jpg Score:39.951792
ImageId:Hand_0001324.jpg Score:40.923982



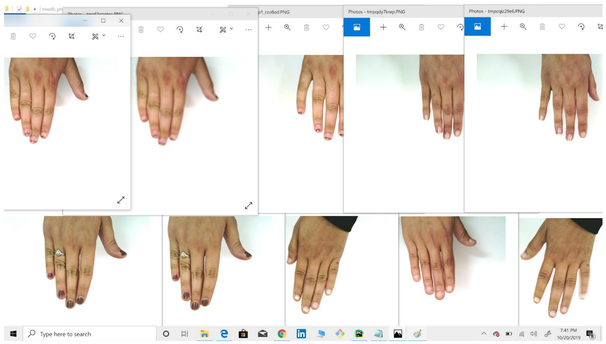
Query 2:

Model : HOG K : 40 M : 10 Image ID : Hand_0000111.jpg

Technique : Principal component analysis (PCA)

ImageId:Hand_0000111.jpg Score:0.000000
ImageId:Hand_0000112.jpg Score:16.188631
ImageId:Hand_0000108.jpg Score:27.056813
ImageId:Hand_0001145.jpg Score:37.373170
ImageId:Hand_0001144.jpg Score:45.054504
ImageId:Hand_0001091.jpg Score:49.636455
ImageId:Hand_0001092.jpg Score:60.161197
ImageId:Hand_0001325.jpg Score:66.725578
ImageId:Hand_0000132.jpg Score:68.750755
ImageId:Hand_0001324.jpg Score:70.791631

Task 2:

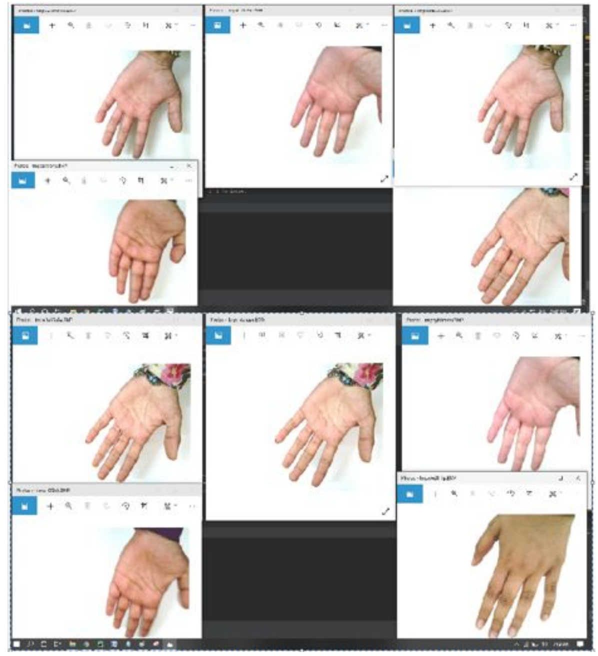


Query 3:

Model : LBP   K : 10   M : 10   Image ID : Hand_0000200.jpg

Technique: Non-negative matrix factorization (NMF)

ImageId:Hand_0000200.jpg Score:0.000000
ImageId:Hand_0000330.jpg Score:0.328226
ImageId:Hand_0000199.jpg Score:0.457021
ImageId:Hand_0000488.jpg Score:0.478226
ImageId:Hand_0000282.jpg Score:0.701752
ImageId:Hand_0000490.jpg Score:1.300131
ImageId:Hand_0000491.jpg Score:1.500885
ImageId:Hand_0000329.jpg Score:1.606854
ImageId:Hand_0000283.jpg Score:1.629197
ImageId:Hand_0001466.jpg Score:1.648980

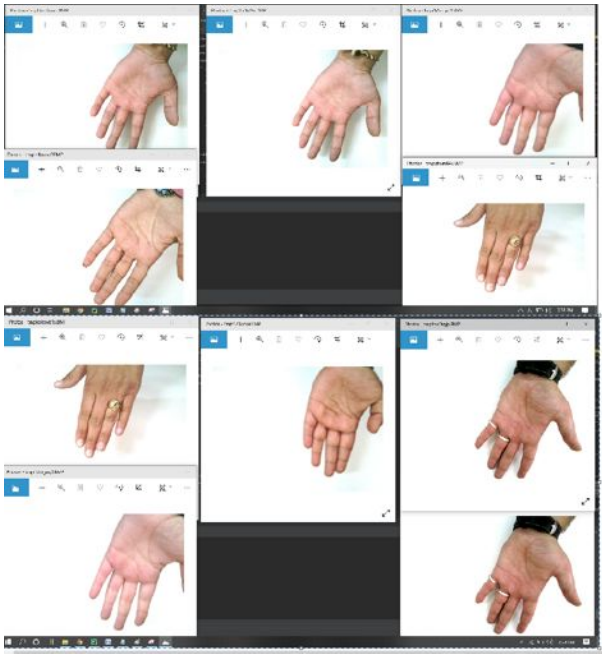


Query 4:

Model : LBP   K : 40   M : 10   Image ID : Hand_0000200.jpg

Technique : Non-negative matrix factorization (NMF)

ImageId:Hand_0000200.jpg Score:0.000000
ImageId:Hand_0000199.jpg Score:1.786988
ImageId:Hand_0000330.jpg Score:2.064967
ImageId:Hand_0000488.jpg Score:2.555205
ImageId:Hand_0000204.jpg Score:3.405612
ImageId:Hand_0000205.jpg Score:3.495886
ImageId:Hand_0000282.jpg Score:3.747003
ImageId:Hand_0011158.jpg Score:3.927982
ImageId:Hand_0000329.jpg Score:3.966061
ImageId:Hand_0011159.jpg Score:4.014334



Task 3

Query 1:

Model: HOG K: 20 Label: Dorsal

Technique: Principal component analysis (PCA)

Refer to the file datasemantic_task3_query1.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task3_query1.html in the folder Output/HTML Files.

Query 2:

Model: HOG K: 30 Label: Dorsal   Technique: Principal component analysis (PCA)

Refer to the file datasemantic_task3_query2.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task3_query2.html in the folder Output/HTML Files

Query 3:

Model: Color Moments K: 20 Label: Left Technique: Latent Dirichlet Analysis (LDA)

Refer to the file datasemantic_task3_query3.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task3_query3.html in the folder Output/HTML Files.

Query 4:

Model: Color Moments K: 30 Label: Left Technique: Latent Dirichlet Analysis (LDA)

Refer to the file datasemantic_task3_query4.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task3_query4.html in the folder Output/HTML Files.

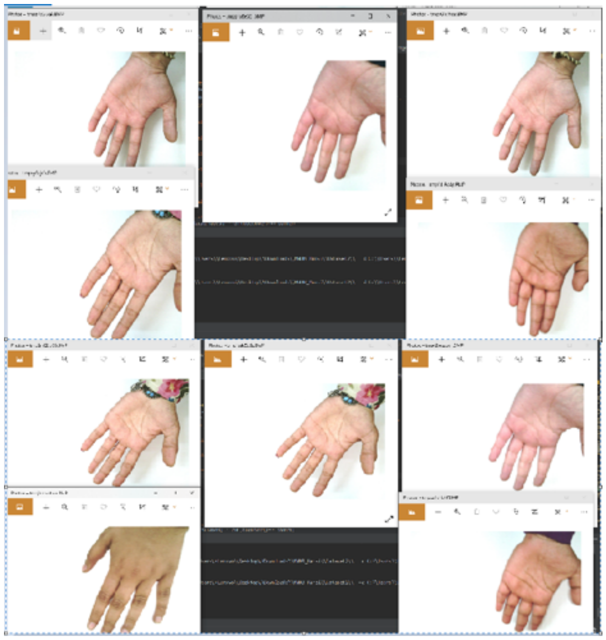
Task 4

Query 1:

Model : LBP   K : 10   M : 10   Label : Palmer
Technique : Non-negative matrix factorization (NMF)

Image ID : Hand_0000200.jpg

ImageId:Hand_0000200.jpg Score:0.000000
ImageId:Hand_0000330.jpg Score:0.328226
ImageId:Hand_0000199.jpg Score:0.457021
ImageId:Hand_0000488.jpg Score:0.478226
ImageId:Hand_0000282.jpg Score:0.701752
ImageId:Hand_0000490.jpg Score:1.300131
ImageId:Hand_0000491.jpg Score:1.500885
ImageId:Hand_0000329.jpg Score:1.606854
ImageId:Hand_0000283.jpg Score:1.629197
ImageId:Hand_0001466.jpg Score:1.648980

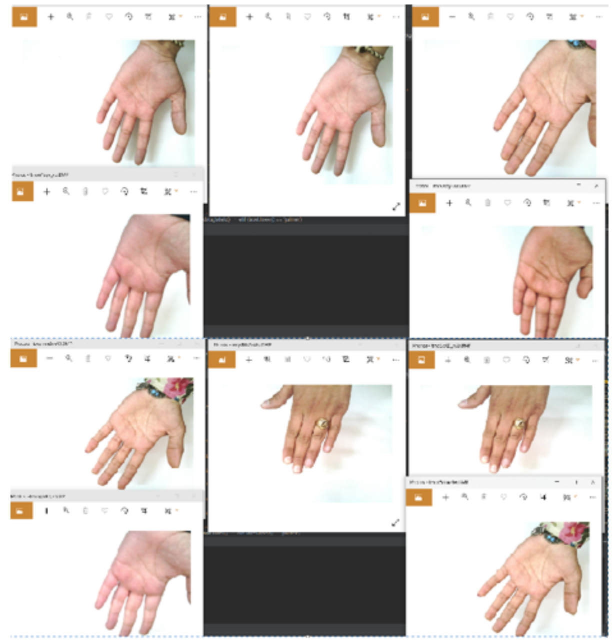


Query 2:

Model : LBP K : 30 M : 10 Label :Palmer Technique: Non-negative matrix factorization (NMF)

Image ID : Hand_0000200.jpg

ImageId:Hand_0000200.jpg Score:0.000000
ImageId:Hand_0000199.jpg Score:1.020905
ImageId:Hand_0000488.jpg Score:1.827584
ImageId:Hand_0000330.jpg Score:1.937412
ImageId:Hand_0000282.jpg Score:3.034903
ImageId:Hand_0000490.jpg Score:3.041869
ImageId:Hand_0000204.jpg Score:3.521146
ImageId:Hand_0000205.jpg Score:3.728970
ImageId:Hand_0000329.jpg Score:3.747792
ImageId:Hand_0000491.jpg Score:3.757151

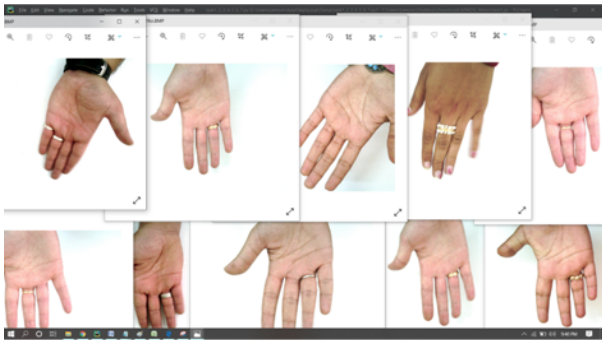


***Query 3*:**

Model : Sift K : 10 M : 10 Label : With Accessories Technique: Singular value decomposition (SVD)

Image ID : Hand_0011160.jpg

ImageId:Hand_0011160.jpg Score:0.000000
ImageId:Hand_0001538.jpg Score:0.335686
ImageId:Hand_0000488.jpg Score:0.344334
ImageId:Hand_0001395.jpg Score:0.376794
ImageId:Hand_0001103.jpg Score:0.379015
ImageId:Hand_0000934.jpg Score:0.383742
ImageId:Hand_0000935.jpg Score:0.387124
ImageId:Hand_0011106.jpg Score:0.394424
ImageId:Hand_0001539.jpg Score:0.409299
ImageId:Hand_0000270.jpg Score:0.416621



Query 4 :

Model : Sift   K : 30   M : 10   Label : With Accessories   Technique: Singular value decomposition (SVD)

Image ID : Hand_0011160.jpg

ImageId:Hand_0011160.jpg Score:0.000000
ImageId:Hand_0000483.jpg Score:0.832808
ImageId:Hand_0011158.jpg Score:0.848672
ImageId:Hand_0000929.jpg Score:0.863594
ImageId:Hand_0001428.jpg Score:0.906966
ImageId:Hand_0000934.jpg Score:0.923949
ImageId:Hand_0000488.jpg Score:0.943051
ImageId:Hand_0000410.jpg Score:0.946401
ImageId:Hand_0001538.jpg Score:0.969503
ImageId:Hand_0001533.jpg Score:0.971742

Task 5

Query 1:

Model: LBP K: 10 Label: Right   Technique: Non-negative matrix factorization (NMF)
Image ID: Hand_0000111.jpg

System labels image Hand_0000111.jpg as right

Query 2:

Model: LBP K: 30 Label: Right   Technique: Non-negative matrix factorization (NMF)
Image ID: Hand_0000111.jpg

System labels image Hand_0000111.jpg as right

Query 3:

Model: Sift K: 10 Label: With No Accessories Technique: Singular value decomposition (SVD)
Image ID: Hand_0001395.jpg

The system does not label image Hand_0001395.jpg as without accessories

Query 4:

Model: Sift K: 30 Label: With No Accessories Technique: Singular value decomposition (SVD)
Image ID: Hand_0001395.jpg

The system does not label image Hand_0001395.jpg as without accessories

Task 6

Query 1:

Subject ID : 27

Subject_id:41 Score:0.994472
Subject_id:1506 Score:0.977645
Subject_id:28 Score:0.974391

Query 2:

Subject ID : 55

Subject_id:54 Score:0.998718
Subject_id:1073 Score:0.994667
Subject_id:505 Score:0.989496

Task 7

Query 1:

K: 10

Refer to the file datasemantic_task7_query1.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task7_query1.html in the folder Output/HTML Files.

Query 2:

K: 20

Refer to the file datasemantic_task7_query2.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task7_query2.html in the folder Output/HTML Files.

Task 8

Query 1:

K: 4

Refer to the file datasemantic_task8_query1.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task8_query1.html in the folder Output/HTML Files.

Query 2:

K: 6

Refer to the file datasemantic_task8_query2.html in the folder Output/HTML Files.

Refer to the file featuresemantic_task8_query2.html in the folder Output/HTML Files.

3. Interface Specification

Command Line Interface is used to run all the python files in this project.

The syntax is as follows:
python filename.py [-arg1] [-arg2] … [-arg n]

Refer the Readme file more information

4. System Requirements and Installations

The software and tools mentioned below were used throughout this phase of the project. The readme file contains instructions about the installation and execution of the program.

4.1 Operating System:

Windows 10, Mac OS and Ubuntu

4.2 Python:

- Download Link: https://www.python.org/downloads/windows/
- For installing Python 3.7.0, instructions are provided at https://www.ics.uci.edu/~pattis/common/handouts/pythoneclipsejava/python.html

Python Libraries:

S.No	Library	Version
1	pillow	6.2.0

2	numpy	1.17.2
3	opencv-python	3.4.2.16
4	opencv-contrib-python	3.4.2.16
5	scipy	1.3.1
6	scikit-learn	0.21.3
7	pandas	0.25.1
8	scikit-image	0.15.0

4.3 Execution Steps:

- Create a Python virtual environment
  - Install virtualenv
    $ sudo apt-get install python3-venv
  - Go to the desired directory and create a virtual environment
    $python3 -m venv venv
  - Activate virtual environment
    $ source venv/bin/activate

- Install Python Libraries
  Note: All commands mentioned below are being run in a python virtual environment
  - Install the necessary libraries:
    $ pip3 install -r <path_to_source>/requirements.txt

Note: It is assumed that you are in the python virtual environment

Below are the steps to run the project for various tasks:

Task 1:

Command

>> python3 task1.py -f <dataset path> -m <feature model> -r <dimensionality reduction technique> -k <latent semantic> -t 1
<dataset path>: the path of dataset folder
<feature model>: acceptable values sift/hog/cm/LBP
<dimensionality reduction technique>: PCA/svd/LDA/nmf
<latent semantic>: number of latent semantic(k)

Example:
>>python3 task1.py -f '/home/mansi/Documents/sem1/MWDB/phase/phase2/Dataset2/' -m hog -r pca -k 2 -t 1

Task 2:

Command
>> python3 task1.py -f <dataset path> -m <feature model> -r <dimensionality reduction technique> -k <latent semantic> -t 3 -l <label>
<label>: acceptable values right/left/dorsal/palmar/male/female/withaccessories/withoutaccessories

Example:
python3 task1.py -f '/home/mansi/Documents/sem1/MWDB/phase/phase2/Dataset2/' -m hog -r pca -t 2 -i Hand_0000185.jpg -s 3

Task 3:

Command
>>python3 task1.py -f <dataset path> -m <feature model> -r <dimensionality reduction technique> -k <latent semantic> -t 3 -l <label>

Example:
python3 task1.py -f '/home/mansi/Documents/sem1/MWDB/phase/phase2/Dataset2/' -d /home/mansi/Documents/sem1/MWDB/phase/pha

se2/newHandInfo.csv -m hog -r pca -k 2 -t 3 -l female

Task 4:

Command
>> python3 task1.py -f <dataset path> -m <feature model> -r <dimensionality reduction technique> -t 4 -l <label> -i <test image> -s <similar image count>

Example:
python3 task1.py -f '/home/mansi/Documents/sem1/MWDB/phase/phase2/Dataset2/' -d /home/mansi/Documents/sem1/MWDB/phase/phase2/newHandInfo.csv -m hog -r pca -t 4 -l female -i Hand_0011599.jpg -s 3

Task 5:

Command
>> python3 task1.py -f <dataset path> -m <feature model> -r <dimensionality reduction technique> -k <latent semantic> -t 5 -l <label> -i <test image>

Example:
python3 task1.py -f '/home/mansi/Documents/sem1/MWDB/phase/phase2/Dataset2/' -d /home/mansi/Documents/sem1/MWDB/phase/phase2/newHandInfo.csv -m hog -r pca -k 2 -t 5 -l left-hand -i Hand_0011599.jpg

Task 6:

Command
>> python3 task1.py -f <dataset path> -m <feature model> -r <dimensionality reduction technique> -k <latent semantic> -t 6 -u <subject id>

Example:
python3 task1.py -f '/home/mansi/Documents/sem1/MWDB/phase/phase2/Dataset2/' -d /home/mansi/Documents/sem1/MWDB/phase/phase2/newHandInfo.csv -m hog -r nmf -k 2 -t 7

Task 7:

Command
> python3 task1.py -f <dataset path> -m <feature model> -r <dimensionality reduction technique> -k <latent semantic> -t 7

Example:
python3 task1.py -f '/home/mansi/Documents/sem1/MWDB/phase/phase2/Dataset2/' -d /home/mansi/Documents/sem1/MWDB/phase/phase2/newHandInfo.csv -m hog -r nmf -k 2 -t 7

Task 8:

Command
> python3 task1.py -f <dataset path> -r <dimensionality reduction technique> -k <latent semantic> -t 8

Example:
python3 task1.py -f '/home/mansi/Documents/sem1/MWDB/phase/phase2/Dataset2/' -d /home/mansi/Documents/sem1/MWDB/phase/phase2/newHandInfo.csv -r nmf -k 2 -t 8

5. Conclusion

We have successfully implemented all the 8 tasks as per the problem specification of this phase with a good understanding of dimensionality reduction and distance/similarity concepts. We have implemented various dimensionality reduction techniques such as SVD, LDA, PCA, and NMF along with calculating similarity and distance measures like Euclidean distance and Cosine similarity. After implementing dimensionality reduction on the feature descriptors we found that computation was faster while giving very similar results. We learned how the feature matrix is formed for images using different feature models and the way to give them as input to feature reduction algorithms such as LDA, PCA, SVD, and NMF. We learned how the transformation happens using packages and how these packages work.

Bibliography:

[1] Di Huang, Caifeng Shan, Mohsen Ardabilian, Yunhong Wang, Liming Chen. Local Binary Patterns and Its Application to Facial Image Analysis: A Survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Institute of Electrical and Electronics Engineers, 2011, 4, 41, pp.1-17. ff10.1109/TSMCC.2011.2118750ff. Ffhal-01354386f

[2]https://software.intel.com/en-us/ipp-dev-reference-histogram-of-oriented-gradients-hog-descriptor

[3]https://en.wikipedia.org/wiki/Euclidean_distance

[4]https://en.wikipedia.org/wiki/Scale-invariant_feature_transform

[5]https://en.wikipedia.org/wiki/Color_model

[6]Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet allocation." *Journal of Machine Learning research* 3.Jan (2003): 993-1022.

[7]http://www.jmlr.org/papers/v5/hoyer04a.html

[8]Donoho, David, and Victoria Stodden. "When does non-negative matrix factorization give a correct decomposition into parts?." *Advances in neural information processing systems*. 2004.

[9]https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60

Appendix:

For the successful completion of this phase of the project, we worked as a team to implement all the given tasks, everyone sharing their conceptual and implementation ideas and helping each other understand every task. On a broad basis, the table below demonstrates how we have shared this phase of the project between ourselves :

S.no	Task	Team Members
1.	Creating/extracting metadata for data	Harika, Rekha
2.	Research on libraries to be used and it's output	All
3.	SIFT	Rekha, Ananth
4.	Task1 PCA Task1 SVD Task1 LDA Task1 NMF	Mansi, Asmi Ananth, Arijit Asmi, Harika Arijit, Rekha
5.	Task 2 Similarity	Mansi, Arijit
6.	Task 3	Ananth, Rekha
7.	Task 4	Arijit, Asmi
8.	Task 5	Mansi, Asmi
9.	Task 6	Ananth, Asmi
10.	Task 7	Arijit, Harika
11.	Task 8	Harika, Mansi

12.	Query results	Harika, Arijit, Rekha, Ananth
13.	Project Report	All
14.	Readme	Mansi, Asmi