CSE 515: MULTIMEDIA AND WEB DATABASES

Fall 2019 Phase 2

Group Members

Anjali, Athul Pramod, Manoj Tiwaskar, Md Shadab, Prashant Singh Chauhan, Tyler Giles

October 20, 2019

Abstract

The project aims at developing a strong foundation in handling and processing high dimensional data and experiment with various dimensionality reduction techniques like PCA, SVD, NMF and LDA and similarity and distance measures like euclidean distance and cosine similarity. The dataset used in this project is associated with the publication "*Mahmoud Afifi. "11K Hands: Gender recognition and biometric identification using a large dataset of hand images." M. Multimed Tools Appl (2019) 78: 20835."* The dataset has been downloaded from the following website. https://sites.google.com/view/11khands

Keywords

Dimensionality Reduction, Latent Semantic Analysis, Principal Component Analysis, Singular Value Decomposition, Nonnegative Matrix Factorization, Latent Dirichlet Allocation, Eigen Vectors, Basis Vectors, EigenValues, Matrix Factorization, Color Model, Color Moments, Histogram of Oriented Gradients, Scale Invariant Feature Transform, Local Binary Patterns, KL Divergence, Euclidean Distance, Non-negative Vector Similarity Coefficient based Distance

1 Introduction

With the magnification of multimedia data like images, videos and audios, Image retrieval has become an important area of research to retrieve relevant information. The voluminous amount of multimedia data available in different domains has made storage, management and retrieval of data a matter of utmost importance. Making sense of such a huge and complex data is necessary to identify important and previously unknown patterns and trends.

High dimensional data can be represented in multiple ways like vector spaces, graphs, trees, strings and sequences. We will be experimenting with vector spaces in this phase of the project. High dimensional data also suffers from dimensionality curse. Having a lot of variables in the dataset gives rise to various problems. To analyze the data, it is important to understand the relationships between variables and there is also a danger of overfitting your model with data. To handle such problems, it is necessary to reduce the dimension of the feature space. Two things can be done in this situation. One, feature elimination, which involves keeping eliminating the redundant and unnecessary features from the data. Identifying the important features is a challenging task. Two, feature extraction, which involves extracting and combining the variables into features. The new reduced set of features should be a close representation of the original dataset.

Performing dimensionality reduction helps in reducing the complexity of the data and removes redundancy in the features thus increasing discrimination power. It helps in understanding and realizing the relationship between different variables and how they depend on each other. There are multiple techniques that can be used in dimensionality reduction and in this project, we will be using Principal Component Analysis, Singular Value Decomposition, Non-Negative Matrix Factorization and Latent Dirichlet Allocation. Popular similarity and distance measures like cosine similarity, Euclidean distance and KL divergence has been used to perform similarity analysis among different images.

1.1 Terminology

- *Dimensionality Curse*: It refers to a phenomenon that arises in analysis data in high dimensional space. It means that as the number of dimensions grows, the amount of data needed to generalize accurately grows exponentially.
- *Dimensionality Reduction*: It is the process of reducing the number of variables under consideration by using a set of principal variables. Feature selection and feature extraction are the two approaches used for dimensionality reduction.
- *Feature Selection*: It is the process of selecting a subset of relevant, important features for constructing models. The redundant and irrelevant features are removed while constructing the models.
- *Feature Extraction*: It is a process of dimensionality reduction which aims at reducing the features by creating new ones from existing ones and discarding the original ones.
- *Matrix Factorization*: It is a way of reducing a matrix into constituent parts that makes it easier to calculate complex mathematical problems.
- *Latent Semantic Analysis*: It is a vector space model used for indexing and retrieval. It is used to select important latent semantic features of an image. The reduced dimensions can be used to represent important information of the semantic space.
- *Bag of Words*: It is a way of extracting features from text or images to make models which are used in machine learning algorithms. These features consists of keypoints and descriptors. Keypoints

are the points in an image that are scale and rotation invariant and descriptor is the description of keypoint.

- *Principal Component Analysis*: It is a feature extraction technique to find a new set of dimensions such as these dimensions are orthogonal and linearly independent. In this transformation technique, the primary principal components are the ones with high variance.
- *Singular Value Decomposition*: It is a matrix factorization technique used to decompose a complex matrix into constituent parts. As compared to eigen decomposition, SVD works on non-square matrices too.
- *Non-Negative Matrix Factorization*: Nonnegative matrix factorization (NMF) has become a widely used tool for the analysis of high dimensional data as it automatically extracts sparse and meaningful features from a set of nonnegative data vectors.
- *Latent Dirichlet Allocation*: It represents an unsupervised machine learning technique used to explain a set of observations by unobserved groups.
- *Color Moments*: A method to represent the first three center moments of color distribution in an image: mean, standard deviation and skewness.
- *Histogram of Oriented Gradients*: A technique that represents the feature descriptor used for object detection. Every pixel in the image return a vector of size <1*9>. Due to normalization, at a time 4 pixels are taken together to form a window of 16*16 size. Each window returns a vector of size <1*36>.
- *Scale-Invariant Feature Transform*: A technique that extracts the key points of an image and its corresponding feature descriptors in a vector. The feature descriptor of SIFT is not affected by the orientation of the image or the scale of the image. In this project, we are limiting the number of key points obtained to 70. This is done so as to maintain consistency with the number of clusters obtained after doing K-Means clustering for each image.
- *Local Binary Pattern*: It is a visual descriptor used for classification which thresholds the neighboring pixels based on the value of the current pixel. It can efficiently capture the local spatial patterns.
- *Feature Descriptor*: A representation of an image that retrieves useful information to represent image features.
- *Euclidean distance*: The euclidean distance is the shortest distance between two vectors, often used as a distance measure. The euclidean distance is classified under P-2 Norm distance. The smaller the Euclidean distance between two images, the more similar the two images are.

$$\text{Euclidean Distance } \boldsymbol{(x,y)}=\sqrt{(x1-y1)2+(x2-y2)2+\ldots.+(xn-yn)2}$$

- *Cosine Similarity*: A similarity measure used to compare two non-zero vectors that measures the cosine of the angle between them. The values of the cosine similarity range from -1 to +1, where +1 means the images are exactly similar and -1 means that the images are opposite to each other. A cosine similarity of 0 means that the vectors are orthogonal or perpendicular to each other. The below formula is used for computing the cosine similarity.

$$\cos\theta=\frac{a.b}{|a||b|}$$

- *Manhattan Distance*: A p-1 norm distance measure used to compute the similarity between two vectors. The below formula computes manhattan distance between two vectors.

$$\text{Manhattan distance } (n, m) = |\mathbf{x}_n - \mathbf{x}_m| + |\mathbf{y}_n - \mathbf{y}_m|$$

- *KL Divergence:* It is a measure of how one probability distribution is different from another probability distribution.

$$D_{KL}(p,q) = \Sigma P(x) log(\frac{p(x)}{q(x)})$$

- *Non-negative vector similarity coefficient-based (NVSC) distance*: The effectiveness of the NVSC measure stems from the fact that it is specifically designed for non-negative vectors, so it is the most appropriate for NMF-based face recognition.. NVSC distance in conjunction with NMF algorithm always achieves the best result and better than any distance measures combined with Eigenface method [1]. It's formula is represented below

$$d(X,Y) = 1 - \gamma^2(X,Y), \quad \gamma(X,Y) = \frac{\sum_{i=1}^{n} \min(x_i, y_i)}{\sum_{i=1}^{n} \max(x_i, y_i)}$$

1.2 Goal description

1. Task 1 To implement a program which when given a feature model (CM, HOG, SIFT and LBP), a positive integer value k and one of dimensionality reduction techniques (PCA, SVD, NMF and LDA), prints the top k-latent semantics in the corresponding vector space. Each latent semantic is presented as a term-weight pair and is ordered in decreasing order of weights. For task 1, a data latent semantic visualizer is implemented which shows a ranked list of image thumbnails along with their scores for each latent semantics and a feature latent semantic visualizer is presented which shows the value of the highest dot product of a given latent semantic with a set latent semantics of other images.

2. Task 2 To implement a program which when given a feature model and k latent semantic (calculated from task 1) for that feature model, displays the top m similar images. Here the feature model, value of k and m are taken as an input from the user.

3. Task 3 To implement a program which when given a feature model and one of the labels (left, right, dorsal, plamar, with accessories, without accessories, male and female), extracts the k latent semantics. Here, the feature model, label name and the value of k is given by the user.

4. Task 4 To implement a program which when given a feature model, one of dimensionality reduction techniques and k latent semantic associated with one of the labels, gives the most similar m images. Here, the feature model, dimensionality reduction technique, the value of k and m are given by the user.

5. Task 5 To implement a program which when given a feature model, one of dimensionality reduction techniques, k latent semantics associated with one of the labels and an unlabeled imageID, labels the image as the following:

a. Dorsal vs Palmar
b. Left vs Right
c. With Accessories or Without Accessories
d. Male vs Female

6. Task 6 To implement a program which when given an subjectID, visualizes the most similar 3 subjects. We have used LDA and all the four feature models to test the results.

7. Task 7 To implement a program which when given a value of k, creates a subject-subject similarity matrix, performs NMF on the matrix and returns the top k latent semantics.

8. Task 8 To implement a program which when given a value of k, creates a binary image-metadata matrix. NMF is performed on this matrix and the top k latent semantics in the image space and metadata space are returned.

1.3 Assumptions

- It has been assumed that all the images including the query image and the images stored in the database are of the same size (1600x1200).
- Images that are retrieved as similar to the query image can be alike in different features depending on the feature descriptor model used.
- For tasks related to color moments, more weightage has been given to the Y component of YUV than U and V. This assumption is made while calculating the weighted distance matrix.
- All the images in the dataset are in jpg format.
- For Task 2 and Task 4, the query image is assumed to be present in the dataset provided.
- For Task 6, we have assumed that the subject is present in the test dataset.

2 Proposed Solution and Implementation

2.1 Task 0

We have used MongoDB, K Means and Bag of Words to implement the task 0 in phase 2. The details of these concepts are given below.

MongoDB

The Database used for the project is the MongoDB database. The database stores the collection. The basic or the fundamental entry in the database being a ‘document’, which is the ‘row’ equivalent of RDBMS. All the documents are grouped into a collection. The RDBMS analogue of collection is Table. The main reason for using MongoDB in the project is its property of being schema less and also due to its efficiency in handling large datasets. Each document in the collection has a primary key called “_id”, which is similar to the key_value in RDBMS.

K Means

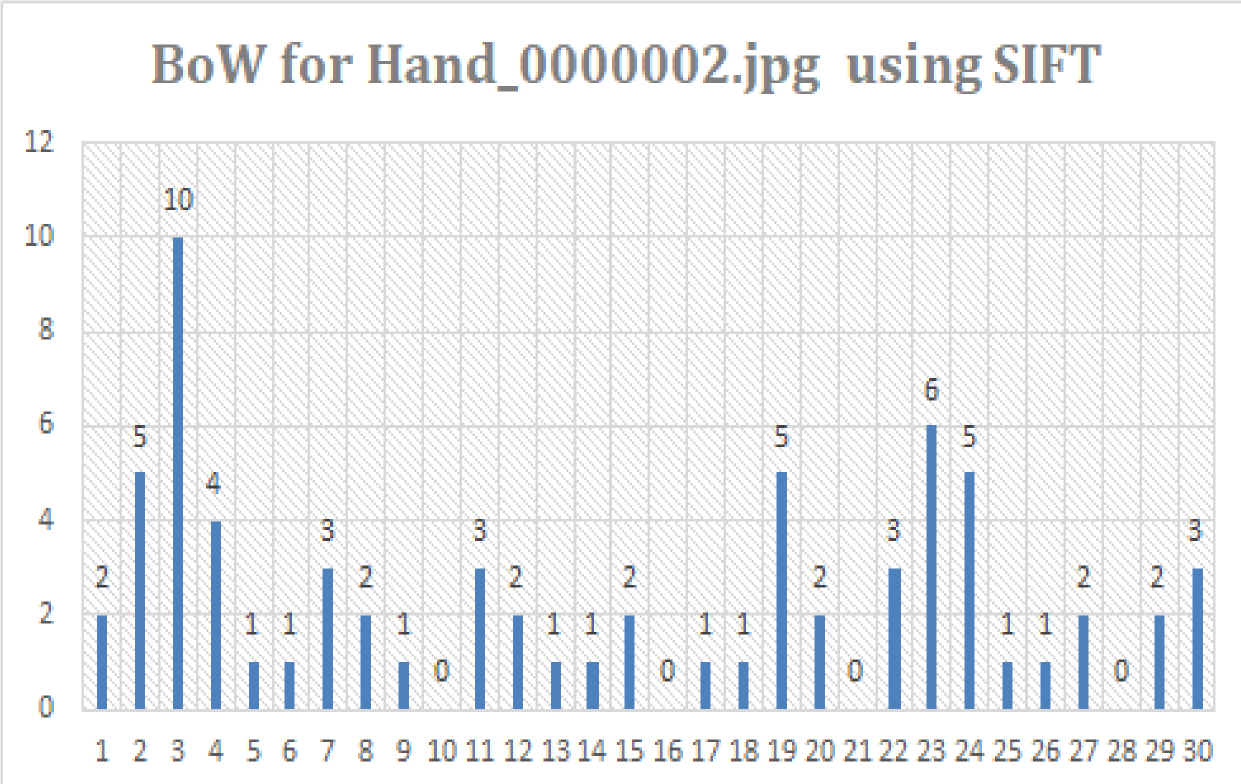
K Means is a popular clustering algorithm. The objective of K Means is to group similar data points together. K Means algorithm looks for a fixed number (k) of clusters in the dataset. The k value decides the number of centroids that we need in our dataset. A centroid is a point representing the center of the cluster. Every observation is then allocated to the nearest cluster. Initially, random centroids are selected and the observations are allocated to the clusters. Again, the centroids are calculated for each cluster and observations are allocated to the clusters. This is an iterative process and it is repeated till the cluster centroids converge and the defined number of iterations (in our case 1000) have been achieved. After testing the results out for various values of k, we found k = 40 to be the optimum value. After testing the results for various values of K, we found better results using K = 40. The following command and parameters have been used to calculate K Means.

KMeans(n_clusters=k, max_iter=1000).fit(feature_desc)

Bag of Words

Bag of words is a simplified representation of the feature descriptors of the images in the dataset. Each slot in the bag represents the count of the number of similar feature descriptors present in the entire set of images. In other words, one slot in the bag represents the total count of feature descriptors which are similar. The bag of words in our project is implemented using K-Means clustering. All the images in the dataset is taken at a time and their corresponding feature descriptors for all the Feature-Models is calculated. Then all the feature descriptors of a feature model is taken and k-means clustering is performed. In the project, as the HOG model returns a feature descriptor of size 9576, with just a single row, if we do K-means clustering on that, it will result in giving just one cluster. So in order to overcome this issue, the HOG feature descriptor is resized into <266*36> matrix. The resultant feature descriptor obtained is used for K-Means clustering. The max_iteration for K-Means clustering in our project is taken as 1000. The resulting

“Number of clusters” obtained is taken as the size of the bag. Each slot in the bag holds the value which is equal to the count of feature descriptors that fit the corresponding cluster in the K-Means.



To implement the tasks of phase 2, we need to follow certain steps.

1. For a given folder containing images, we need to calculate the feature descriptor of each image using all the four models (CM, LBP, HOG and SIFT). CM, LBP, HOG and SIFT function from phase 1 has been used to compute the results of the feature descriptors in this phase. CM, LBP, HOG and SIFT modules are imported in this phase. The results are then stored in the ‘image_models’ collection of the imagedb. The results are stored as key, value pair in the database where model name is the key name and the corresponding feature descriptor is the value.

2. A test dataset containing 200 images have been used in the project. We would apply K Means algorithm (Value of K has been assumed as 40) to the feature descriptors obtained in step 1 for each model. That means, K Means is applied to all the feature descriptors obtained using CM, HOG, LBP and HOG. The results of K Means will give us 40 cluster centers which are stored as a list. K Means has been calculated using the K Means function in python. K Means is imported from the sklearn.cluster package. It is a clustering algorithm which partitions n observations into K clusters.

The results obtained in the above step are stored in the 'centroids' collections of the imagedb. These results can be used again during task 5 and task 6. A new image which does not exist in the database can be given at runtime and this result can be used to compute the bag of words for the new image. The K value is regarded as the bag size. In this phase we will be using the concept of Bag of Words to represent the features of the image.

3. After calculating 40 Cluster Centers, for each image we will identify the features that belong to these 40 bags. The results are then stored as a key, value pair in the image_model collection of the imagedb. Here the key represents the bag of each model and the value presents the bag of words. They can be referenced using the imageID.

4. Download the csv file containing the metadata of the original 11k Hands dataset. We will run a python script over the test dataset images and create a new csv file which contains the metadata of the images contained in the test dataset. This csv file is then transferred to the database and named as ImageMetadata collection. The following command has been used to implement this step.

   mongoimport --db imagedb --type csv --file E:\data\db\dump\ImageMetadata.csv --headerline

5. We would then be creating a 'subjects' collection in the imagedb which contains the aspect of hand and orientation information for each subject. We have created four records such that all the dorsal left, dorsal right, palmar left and palmar right images are stored separately for each subject.

2.2 Task 1

We have used the following dimensionality reduction techniques in our project.

Principal Component Analysis

The principal component analysis is a mathematical technique to reduce the dimensionality of the data. It works on the principle of factoring matrices to extract the patter of the linear system. The underlying principle in PCA is that it first finds the maximum variations in the original dataset. The data is then projected on to new axis called the U V axis. The direction of U and V axis is known as principal components.The first few principal components are rich in information whereas the last features are the least significant aong the others.

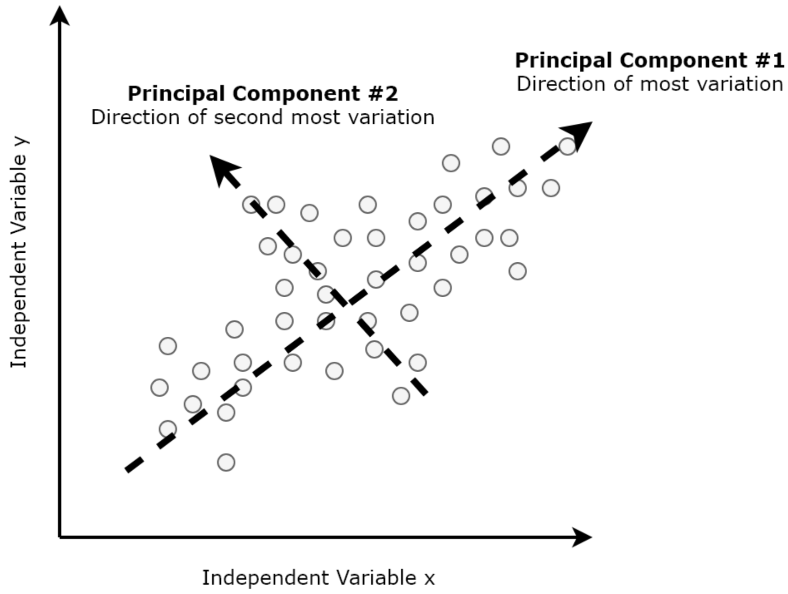


Fig: *A graphical representation of Principal Component Analysis*

Singular Value Decomposition (SVD)

The singular value decomposition matrix performs the factorization of a matrix P into the product of 3 matrices.

$$\boldsymbol{X = U\,S\,V^T}$$

The columns of U and V are orthonormal and the matrix S is diagonal with positive entries.

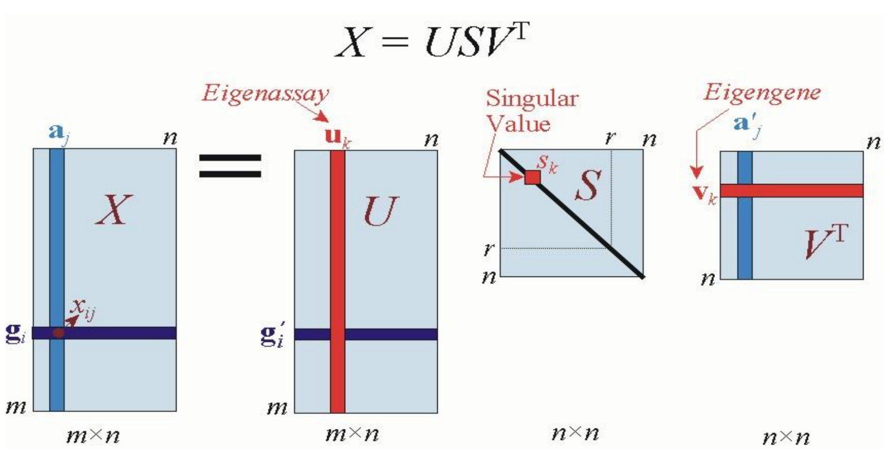


Fig: *Singular Value Decomposition representation*

Fig source: https://public.lanl.gov/mewall/kluwer2002.html [7]

Latent Dirichlet Analysis (LDA)

LDA as a technique is used in document processing. It is typically used to detect underlying topics in text document. According to LDA, every document contains a number of topics and each topic contains a distribution of words associated with it.

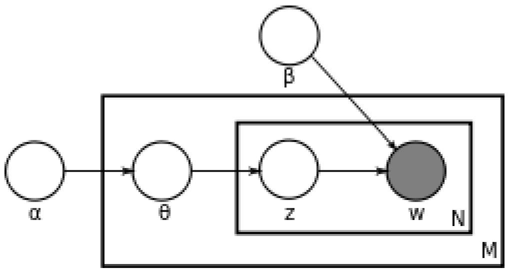


Fig source:
https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation#/media/File:Latent_Dirichlet_allocation.svg

α= The parameter of the Drichlet prior on the per-document topic distribution.

β= The parameter of the Drichlet prior on the per topic word distribution.

θ_m= The topic distribution for the topic m

z = the topic for the nth word in the document m

w = The specific word under consideration

In the usual implementation of LDA, several preprocessing steps are performed which involves data cleaning, tokenizing and stemming. In the project, however, we will not be following this since we are dealing with the bag of words which we have already got by performing the K-Means clustering on the feature descriptors for all images. In the project, we have used *LatentDirichletAllocation(k, max_iter)* function which is imported from sklearn.decomposition library. The max_iteration is kept as 25.

Non-Negative Matrix Factorization (NMF)

Nonnegative Matrix Factorization (NMF) is a matrix factorization method where we constrain the matrices to be nonnegative. In order to understand NMF, we should clarify the underlying intuition between matrix factorization. Suppose we factorize a matrix X into two matrices W and H so that $X \approx WH$. There is no guarantee that we can recover the original matrix, so we will approximate it as best as we can. The rank r of the factorization is generally chosen so that (n+m)r<nm, and the product WH can be regarded as a compressed form of the data in X [2]. Task 2 and Taks 4 for NMF uses NVSC distance similarity measure.

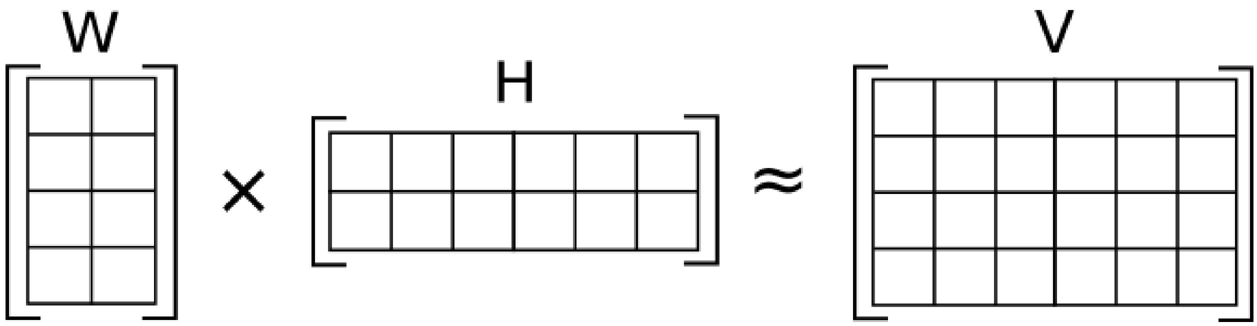


Fig source: *https://www.wikiwand.com/en/Non-negative_matrix_factorization [10]*

For task 1, the following inputs are taken from the user.

1. One of the four feature models (CM, HOG, LBP, SIFT)
2. A positive value ‘k’
3. One of the four dimensionality reduction techniques (PCA, SVD, LDA, NMF)

We need to find the top k-latent semantics for a given feature model using one of dimensionality reduction techniques. To implement task 1, we will be constructing a data matrix using the 200 images in our test dataset and the 40 bags that we created in task 0 will be used as the variables of the original data matrix. Therefore, the data matrix consists of 200 observations and 40 variables.

For SVD, the data matrix decomposes into a core matrix and two factor matrices (data-latent semantic matrix and feature latent semantic matrix). To find the decomposition matrices, the svd function from scipy.linalg package in python is used.

svd(feature_desc, full_matrices=False)

The data-latent semantic matrix U has 200 x k dimensions, the core matrix has k x k dimension and the feature-latent semantic matrix has k x 40 dimensions. The diagonal elements of the core matrix with dimensions k x k represent eigenvalues. The diagonal elements (i,i) of the core matrix correspond to the ith column of the object-latent semantic matrix. These values are the eigenvectors. The eigenvalues are arranged in nondecreasing order and the eigenvectors corresponding to these eigenvalues are also sorted. The imageIDs that are associated with these eigenvectors are also arranged accordingly. For each latent semantic, the imageID is determined as the term and the corresponding value in the eigenvector is displayed as weight. The term weight for each latent semantic is then displayed in the visualizer.

For PCA, covariance matrix C is calculated using the cov function of the numpy package in python. The eigenvalues and the corresponding eigenvectors are calculated for the created covariance matrix. The eigenvectors for each latent semantic is then sorted in nondecreasing order and the corresponding imageIDs are arranged as well. This represents the term weight pair of each latent semantic, where the term denotes the imageID and the weight represents the corresponding eigenvector value. We are performing the matrix decomposition in PCA without using any existing libraries.

For LDA, the object feature matrix is decomposed into object-latent semantic matrix and feature latent semantic matrix. The matrices can be found by using the LatentDirichletAllocation function sklearn.decomposition package in Python.

LatentDirichletAllocation(k, max_iter=25)

Each latent semantic (ith column of the object latent semantic matrix) is sorted in nondecreasing order and the imageIDs are arranged accordingly. The term-weight pair of each latent semantic is displayed, where the term represents the imageID and weight denotes the corresponding value in the latent semantic.

For NMF, a similar process is followed. Here the object feature matrix consists of all non-negative values. The data matrix is decomposed into data-latent semantic matrix (W) and feature-latent semantic matrix (H). For finding the constituent matrices in NMF, the NMF function from sklearn.decomposition package in Python is used.

NMF(n_components=k, init='random', random_state=0)

For each latent semantic in the data-latent semantic matrix, the values are arranged in nondecreasing order and the corresponding imageIDs are sorted accordingly. The term weight pairs are then displayed on the console for each latent semantic.

For all the four dimensionality reduction techniques, the top k latent semantics are displayed in the console. Since each latent semantics is a ranked list of images, we have created a data latent semantic visualizer that shows a ranked list of image thumbnails along with their scores, for each latent semantic.We have also created a feature-latent semantic visualizer which, for each latent semantic, selects the image with the highest dot product to that latent semantics and visualizes that image as the visual-placeholder for that latent semantic.

2.3 Task 2

For task 2, the following inputs are taken from the user.

1. One of the four feature models (CM, HOG, LBP, SIFT)
2. One of the four dimensionality reduction techniques (PCA, SVD, LDA, NMF)
3. A positive k value
4. A value m for finding m similar images
5. imageID

We have to visualize and display the top m similar images. The value of m and k are given by the user. We are assuming that the feature descriptor of imageID in task 2 is already present in the database. The original data matrix has 200 x 40 dimensions as there are 200 images in our test dataset and we have taken 40 as bag size. This data matrix is then project to k-dimensional latent space. The k-latent semantic of the given imageID is taken from the transformed data matrix and and it is compared with the latent semantics of other images. The following similarity and distance measures are used for calculating the similarity score of a given image with a set of images.

1. **Euclidean Distance**
   The Euclidean distance is the straight line distance between the two given vectors. In other words, it is the shortest distance between any two vectors in the Euclidean space. It is a P-2 Norm distance metric. Mathematically, the euclidean distance can be formulated as follows:

   $$\textit{\textbf{Euclidean distance (x,y)}}=\sqrt{(x1-y1)2+(x2-y2)2+\ldots.+(xn-yn)2}$$

   We have used the Euclidean distance metric in both the PCA as well as SVD, because, as we are handling the features for each object in the form of "Bag of Words", So we are trying to find the difference in the frequency of each word in the bag, thus making the Euclidean distance the most convenient distance metric to be used for the aforementioned techniques.

2. **KL Divergence**
   KL divergence is a measure of how one probability distribution is different from a second reference probability distribution. So, it is used to measure the distance between two distributions.

   $$D_{KL}(p,q) = \Sigma P(x) log(\frac{p(x)}{q(x)})$$

   The reason why we are using KL Divergence is because LDA deals with probability distribution and for probability distribution, KL Divergence is the distance measure that we have used in this project.

3. **Non-negative vector similarity coefficient-based distance**
   This distance metric calculates the maximum and minimum distance between the vectors. The reason why we are using Non-negative vector similarity coefficient-based distance is because the paper mentioned in the reference [2], the author has tried it with 19 different similar metrics out of which the non-negative vector similarity coefficient-based distance gives the best result.

Note: In case of NMF, the rank r of the factorization is generally chosen so that (n+m)r<nm, and the product WH can be regarded as a compressed form of the data in X. But in task 2 query 4 we have been given K=40 and we have taken bag size of 40. Which implies that the constraint is not being satisfied **(please relate to results for task 2, query 4).**

2.4 Task 3

For task 3 implementation, the following inputs are taken from the user.

1. One of the four feature models (CM, HOG, SIFT, LBP)
2. One of the following labels.
    a. Dorsal
    b. Palmar
    c. Left Hand
    d. Right Hand
    e. With Accessories
    f. Without Accessories
    g. Male
    h. Female
3. A positive value ‘k’

To implement task 3, select all the images corresponding to the input ‘label’ from the ImageMetadata collection created in imagedb. We create a data matrix using these selected images of dimension m x 40, where m is the number of images corresponding to the given label and 40 is the bag size. This data matrix is then projected to the k-dimensional latent semantic space. The transformed data matrix can then be used to find the top k-latent semantics. This data matrix is then decomposed using one of the dimensionality reduction techniques discussed in task 1 and the data-latent semantic matrix is used to display the k latent semantics.

2.5 Task 4

For task 4 implementation, the following inputs are taken from the user.

1. One of the feature models (CM, HOG, LBP, SIFT)
2. One of dimensionality reduction techniques (PCA, SVD, NMF, LDA)
3. A positive value ‘k’
4. A label
5. A value m for finding m similar images
6. An imageID

In task 4, we have assumed that the query image will be given from the test dataset. As per the input label given by the user, we need to find all the images corresponding to that label. These images are lifted from the database and a m x 40 dimensional data matrix is created. Here, m represents the number of images corresponding to a given label and 40 is the bag size. The data matrix is then projected to k-dimensional latent space. The latent semantic corresponding to the input imageID is then compared with the latent semantic of other images using distance and similarity measures. For PCA and SVD, we are using Euclidean Distance, LDA uses KL-Divergence and NMF uses Nonnegative Vector Similarity Coefficient based Distance. The details of these distance and similarity measures are given in task 2.

2.6 Task 5

For task 5, we will be taking the following inputs from the user.

1. One of the four feature models (CM, HOG, SIFT, LBP)
2. One of the four dimensionality reduction techniques (PCA, SVD, LDA, NMF)
3. A positive value ‘k’
4. An unlabeled image ID

In task 5, we need to classify the unlabeled imageID as

1. Dorsal or Palmar
2. Left or Right
3. Male or Female
4. With or Without Accessories

The unlabeled image can be from the test dataset or the original 11k dataset. Feature descriptor for the unlabeled image is calculated and the bag of words for the unlabeled image is computed. We would then create four data matrices for all dorsal left, dorsal right, palmar left and palmar right images. The unlabeled image’s feature descriptor is then projected to these data matrices. This result is then compared to all other observations in the four data matrices and similarity score is computed for each combination. For PCA and SVD, we will be using Euclidean distance, KL Divergence for LDA and NVSC for NMF. We are following the same similarity measures that we used in the previous tasks. The label which has the lowest similarity score can be selected. The same process is followed for classifying the image as with or without accessories and for identifying the gender. ‘With Accessories’, ‘Without Accessories’, ‘Male’ and ‘Female’ data matrices are created which contains all the images that belong to a specific label. The data matrices are then transformed to k-dimensional latent space. The latent semantic of the query image is then compared to the latent semantics of all other images. The least minimum similarity score between the results of ‘With Accessories’ and ‘Without Accessories’ is selected and labeled accordingly. The same is followed with identifying the gender label.

2.7 Task 6

For task 6, the following input is taken from the user.

1. subjectID

In this task, we have to find the most similar 3 subjects to a given subject (subjectID taken as input from the user). To accomplish this task, we will be using the following algorithm.

1. Create a subjects collection in the imagedb which stores the dorsal left, dorsal right, palmar left and palmar right images for specific subjects. The documents can be referenced using the subjectID.
2. Create 4 data matrices where each matrix corresponds to dorsal left, dorsal right, palmar left and palmar right images. For instance, we will create a data matrix of all dorsal left images of a subjectID and append the dorsal left feature vector (dimension 1 x 40) of the query image to the

data matrix created before. The resultant data matrix is then projected to k-latent semantic space using LDA.

3. Repeat step 2 for dorsal right, palmar left and palmar right.
4. After completing the transformation, the latent semantic for the dorsal left image of the query is compared with all other dorsal left images of a particular subject. This comparison is done using KL Divergence method as we have to compare two probability distributions. The least minimum similarity score between the query image and the given set of images is selected. This step is repeated for all the dorsal left images of the query image and the mean of all similarity scores obtained during this step is calculated.
5. Repeat step 4 for dorsal right, palmar left and palmar right images.
6. If dorsal left, dorsal right, palmar left or palmar right image property is not present in either query image or the subject in consideration, that property is not selected for computing the final similarity score.
7. The results from task 5 are summed to compute the final similarity score between two subjects.
8. The similarity scores of all the subjects with the given query subject is calculated and sorted. The top three results are displayed.

Following is the mathematical equation to compute subject similarity score.

$$Subject\ Similarity\ Score = \sum_{L} \frac{\sum_{i=0}^{n} min\,(\,KL(L_i\,, L_j\ ))}{n} \quad \text{where j} \in (1,2,....m \text{ images}).$$

Where L is the labels such that L∈ {palmar left, palmar right, dorsal left, dorsal right}, $L_i$is the projected BoW of image for the label in the given query subject , n is the number of images for label and $L_j$is the projected BoW of image for the label and j is the number of images for each subject in the database . KL refers to KL divergence.

We have taken 20 as the value of k and used SIFT as our feature model in this task. To display similarity results, we are giving more attention to orientation than color features. We used LDA because after testing various combinations of different dimensionality reduction techniques and feature models, we found that LDA with SIFT gives better results.

2.8 Task 7

For task 7, the following input is taken by the user.

1. A positive value 'k' to show the top k latent semantics

Using the results from task 6, we will be constructing a subject-subject similarity matrix. Here the diagonal elements will be zero as when a subject is compared to itself, the resultant value should be zero and (i,j)th element in the matrix will represent the similarity score between the ith subject and the jth subject. We will then perform NMF on the subject-subject similarity matrix and display the term weight pair. We will get reduced dimensions on performing NMF and the data-latent semantic matrix can be used to display the term weight pairs.

2.9 Task 8

For task 8, the following inputs are taken by the user.

1. A positive value ‘k’ for displaying the top k latent semantics in the image space and in the metadata space.

We have used 'imageName', 'aspectOfHand', 'Orientation', 'gender' and 'accessories' metadata from the image metadata.csv file to make a binary image metadata matrix. We have chosen only those metadata features which have only two values i.e which can be given as binary numbers 0 or 1. For example, in the case of gender, male is assigned a label “0” and female is assigned a label “1”.

NMF is then performed on binary image metadata matrix which decomposes the original matrix into image-space and metadata space. Each latent semantic is presented in decreasing order of weights for image space and metadata space. The image space is rescaled to basis matrix. The data latent semantic visualizer is implemented which shows a ranked list of image thumbnails along with their scores for each latent semantics in image space and a feature latent semantic visualizer is presented in a way which shows a ranked list of metadata along with their scores for each latent semantics in metadata space.

3 Interface Specifications

3.1 System and Data

This project has been implemented on Windows and Mac OS and uses Python 3 as the programming language with code written and commented. Code has been tested on machine with 8GB and 16GB RAM.

3.2 Query Specification

Inputs are taken using the command line. All tasks are implemented separately. When the program starts, user is prompted to give an input for task number , followed by the inputs required by each task so that the program can be run successfully. Please refer the following image for passing command line arguments.

```
#Parsing the command line arguments
parser = argparse.ArgumentParser()
parser.add_argument('-M', '--model',action="store", dest="model",help="Provide any of these model: LBP, SIFT, CM, HOG", default="SIFT")
parser.add_argument('-T', '--technique',action="store", dest="technique",help="Provide any of these technique: PCA, LDA, NMF, SVD", default="SVD")
parser.add_argument('-d', '--dir',action="store", dest="dir",help="Provide directory name", default="None")
parser.add_argument('-l', '--label',action="store", dest="label",help="Provide label of image: left, access, dorsal etc", default="None")
parser.add_argument('-i', '--imageid',action="store", dest="imageid",help="Provide image name", default="None")
parser.add_argument('-k', '--klatent',type=int, dest="klatent",help="Provide k value to get k latent symantics", default=20)
parser.add_argument('-m', '--mimage',type=int, dest="mimage",help="Provide m value to get m similar images", default=10)
parser.add_argument('-s', '--subject',type=int, dest="subject",help="Provide subject Id", default=-1)
parser.add_argument('-t', '--taskid',type=int, dest="taskid", help="Provide the task number", default=-1)
```

To run the tasks, use the following command.

python phase2_main_script.py

3.2.1 Task 0

To setup the imagedb with collections for subjects, subject metadata, image models and bag of words, the following queries are executed.

python phase2_main_script.py -d ../Dataset2 -t 0

mongoimport --port 27018 --db imagedb --type csv --file Desktop/Study/MWDB/ProjectTest/csv/ImageMetadata.csv --headerline

python phase2_main_script.py -t 9

3.2.2 Task 1

For a given feature model, k value and dimensionality reduction technique, execute the following query.

python phase2_main_script.py -M CM -k 20 -T LDA -t 1

3.2.3 Task 2

For a given feature model, k value, m value, imageID, dimensionality reduction technique, execute the following query to run task 2.

python phase2_main_script.py -k 10 -m 10 -i Hand_0000111.jpg -T PCA -M HOG -t 2

3.2.4 Task 3

For a given feature model, k value, label and dimensionality reduction technique, execute the following query to run task 3.

python phase2_main_script.py -k 20 -l left -T LDA -M LBP -t 3

3.2.5 Task 4

For a given feature model, dimensionality reduction technique, label, imageID, k value and m value, execute the following query to run task 4.

python phase2_main_script.py -k 10 -m 10 -i Hand_0000200.jpg -l palmar -T NMF -M LBP -t 4

3.2.6 Task 5

For a given k value, image, label dimensionality reduction technique and feature model, execute the following query to run task 5.

python phase2_main_script.py -k 10 -d ../phase2 -i Hand_0000896.jpg -l right -T SVD -M SIFT -t 5

3.2.7 Task 6

For a given subjectID, run the following query to execute task 6.

python phase2_main_script.py -s 27 -t 6

3.2.8 Task 7

For a given k value, run the following query to run task 7.

python phase2_main_script.py -k 10 -t 7

3.2.9 Task 8

For a given k value run the following query to run task 8.

python phase2_main_script.py -k 4 -t 8

4 System Requirements, Installation and Execution

The system requires Python 3.7.x to be installed in the Anaconda environment. The set of packages that needs to be installed in the system are:

- opencv-contrib-python
- pandas 0.25.1
- Scikit-image-0.15.0
- numpy 1.16.4
- scipy 1.3.1
- scikit-learn 0.21.1
- pillow
- pymongo

The packages described above can be installed using pip installer for python. However, as we are using the Anaconda environment, it is recommended to use the conda installer instead of pip.

In addition, as we are using MongoDB database for the project, the system should be installed with MongoDB 4.x. The windows version of the MongoDB can be installed from the MongoDB community download center. For system running on MacOS, it is advised to install MongoDB using the HomeBrew package installer for Mac. Also, a file db should be created in the data folder which in turn needs to be created in the bin folder of MongoDB. It is advised to install MongoDB Compass along with MongoDB for the viewing of database to the user, although not necessary.

5 Related Work

A similar work done using LDA is in web spam filtering by István Bíró, Jácint Szabó and András A. Benczúr. The work focuses on applying modification of LDA for web spam classification. Similar to our project, the work implements the bag of words document for every website and run LDA on both the corpus of sites labeled as spam and non spam [6].

Another work done in the field of PCA was by Kwang In Kim, Keechul Jung, and Hang Joon Kim in their paper "Face Recognition Using Kernel Principal Component Analysis". In their work, rather than using the generic PCA, an extension of PCA called kernel principal component analysis was used, which is a non linear extension of PCA. The paper uses kernel PCA as the mechanism to extract facial features. '*By using the polynomial kernel,the principal components is computed within the space spanned by high order correlation of input pixels making up a facial image, thereby producing good performance*' *[Face Recognition Using Kernel Principal Component Analysis, Kwang In Kim, Keechul Jung, and Hang Joon Kim] [4].*

6 Results

Note: Given the size of the results returned some of the attached images are only able to show a portion of the data returned. The visualizers that have been created, however are able to scroll vertically or horizontally to display more of the data when it is too much to fit in one window.

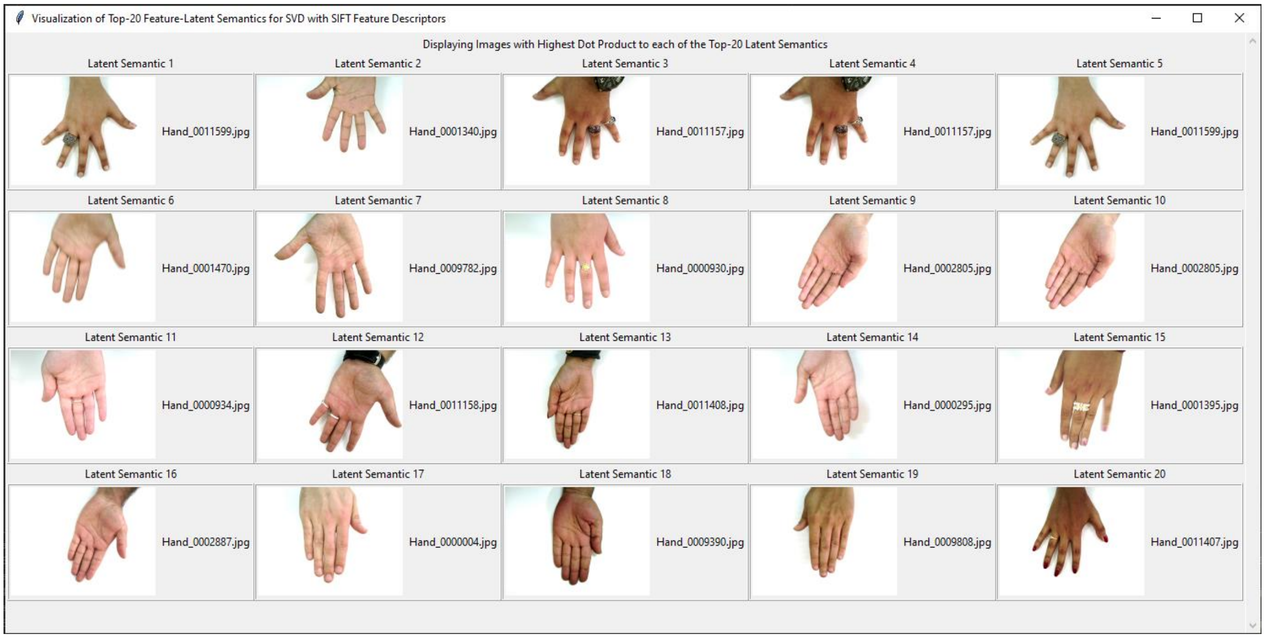
6.1 Task 1

In testing Task 1 the following 4 queries were performed.

Query 1: Feature Model: SIFT, K: 20, Technique: SVD

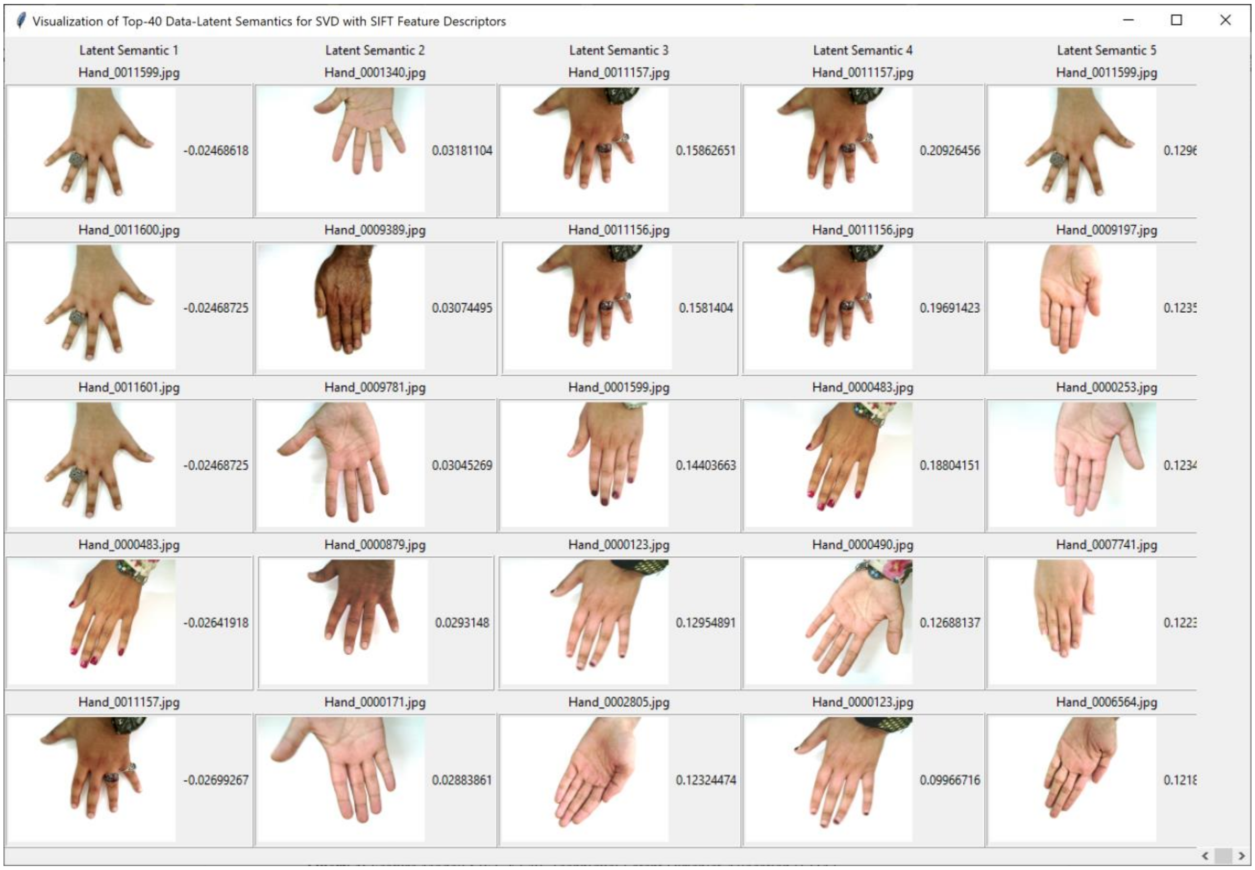
Data Latent Semantic Visualizer

Feature Latent Semantic Visualizer

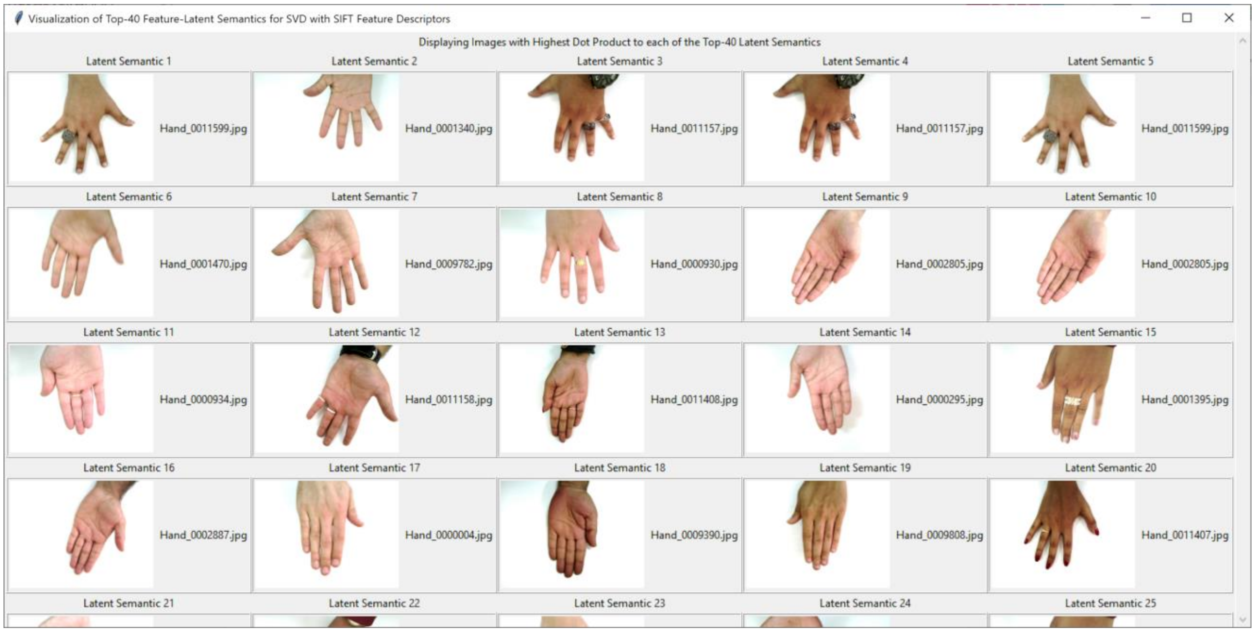


Query 2: Feature Model: SIFT, K: 40, Technique: SVD

Data Latent Semantic Visualizer

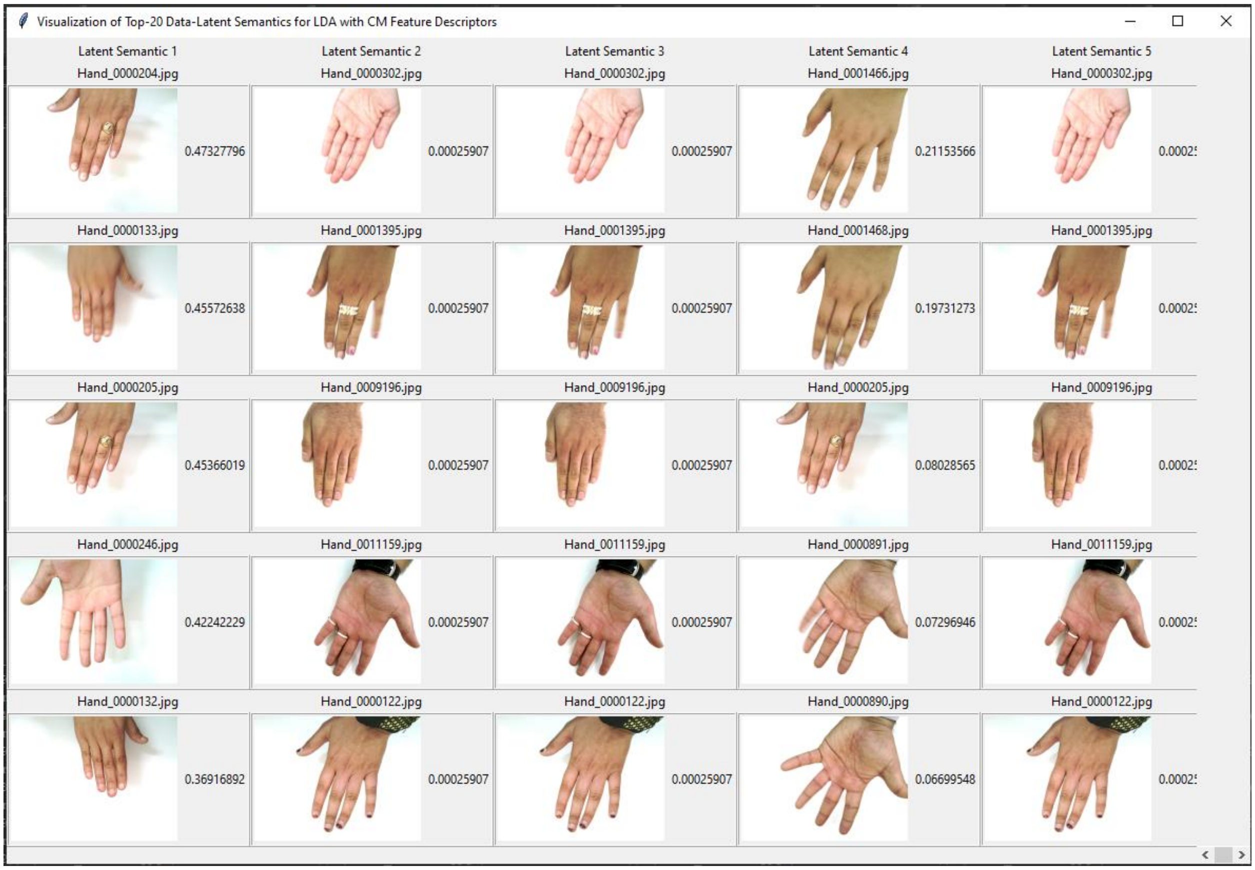


Feature Latent Semantic Visualizer

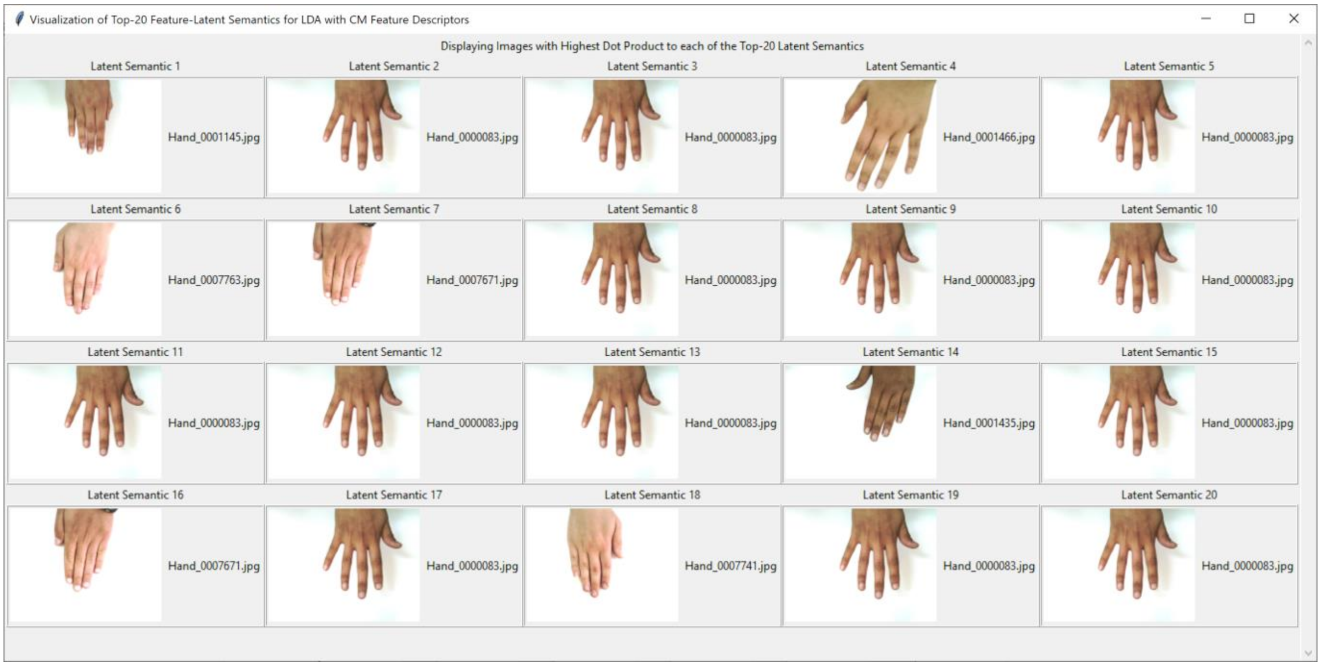


Query 3: Feature Model: Color Moments, K: 20, Technique: LDA

Data Latent Semantic Visualizer

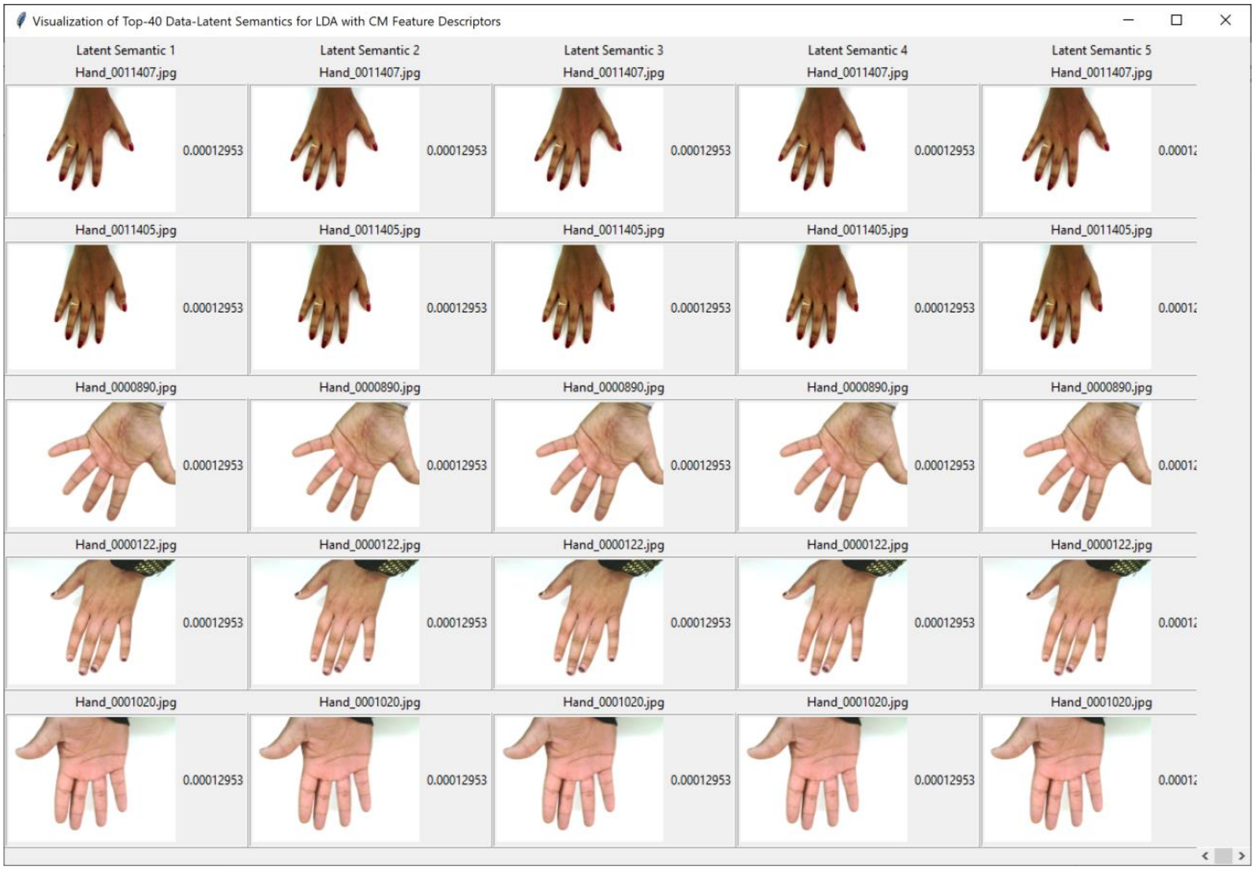


Feature Latent Semantic Visualizer

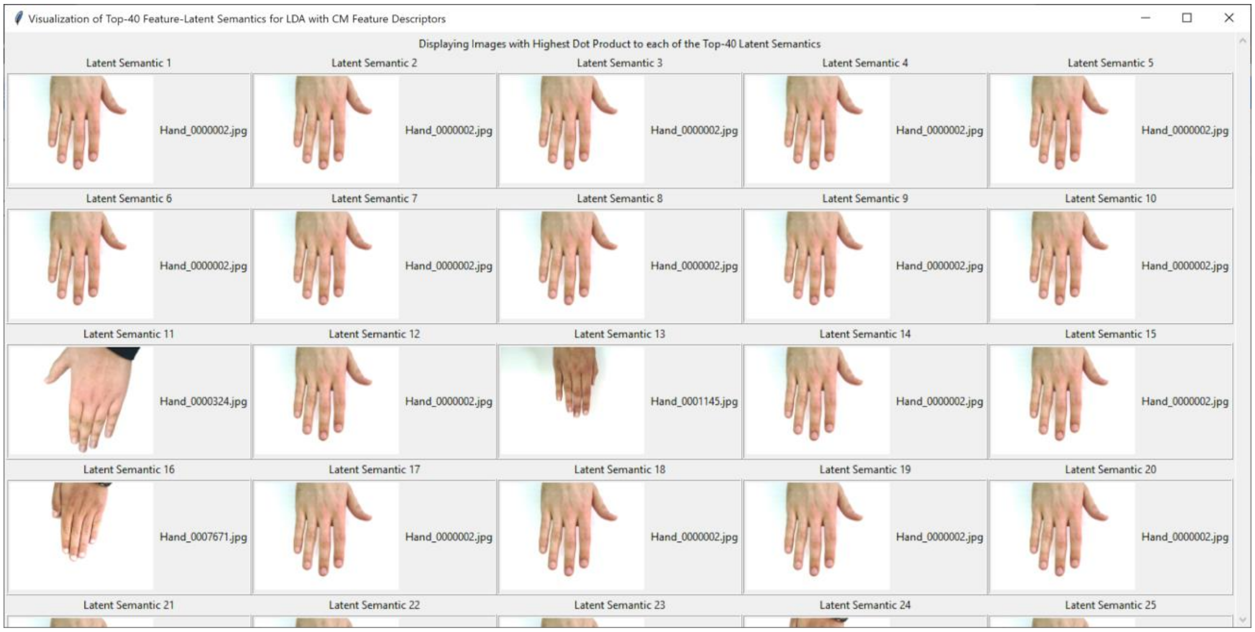


Query 4: Feature Model: SIFT, K: 40, Technique: LDA

Data Latent Semantic Visualizer



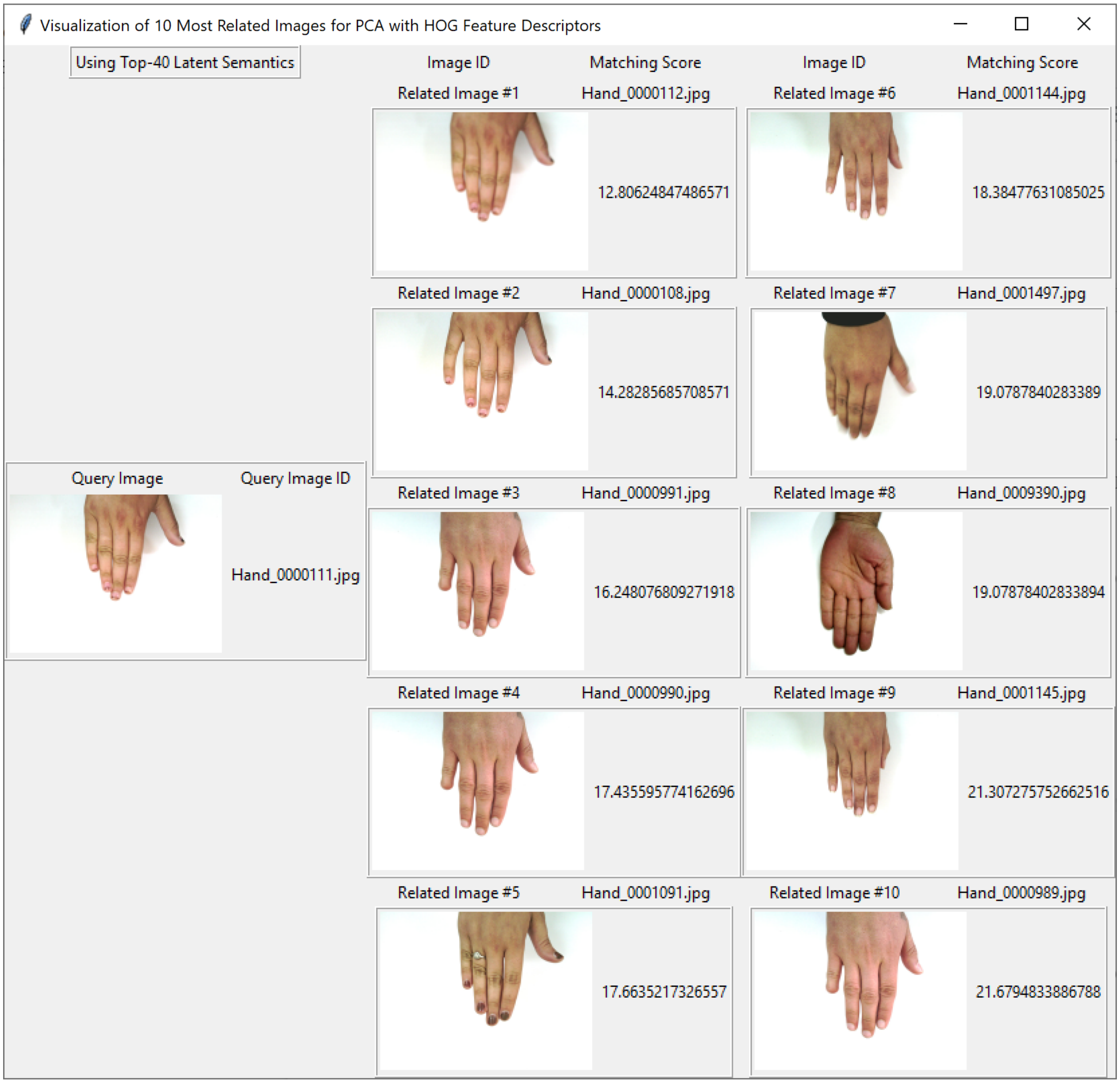
Feature Latent Semantic Visualizer



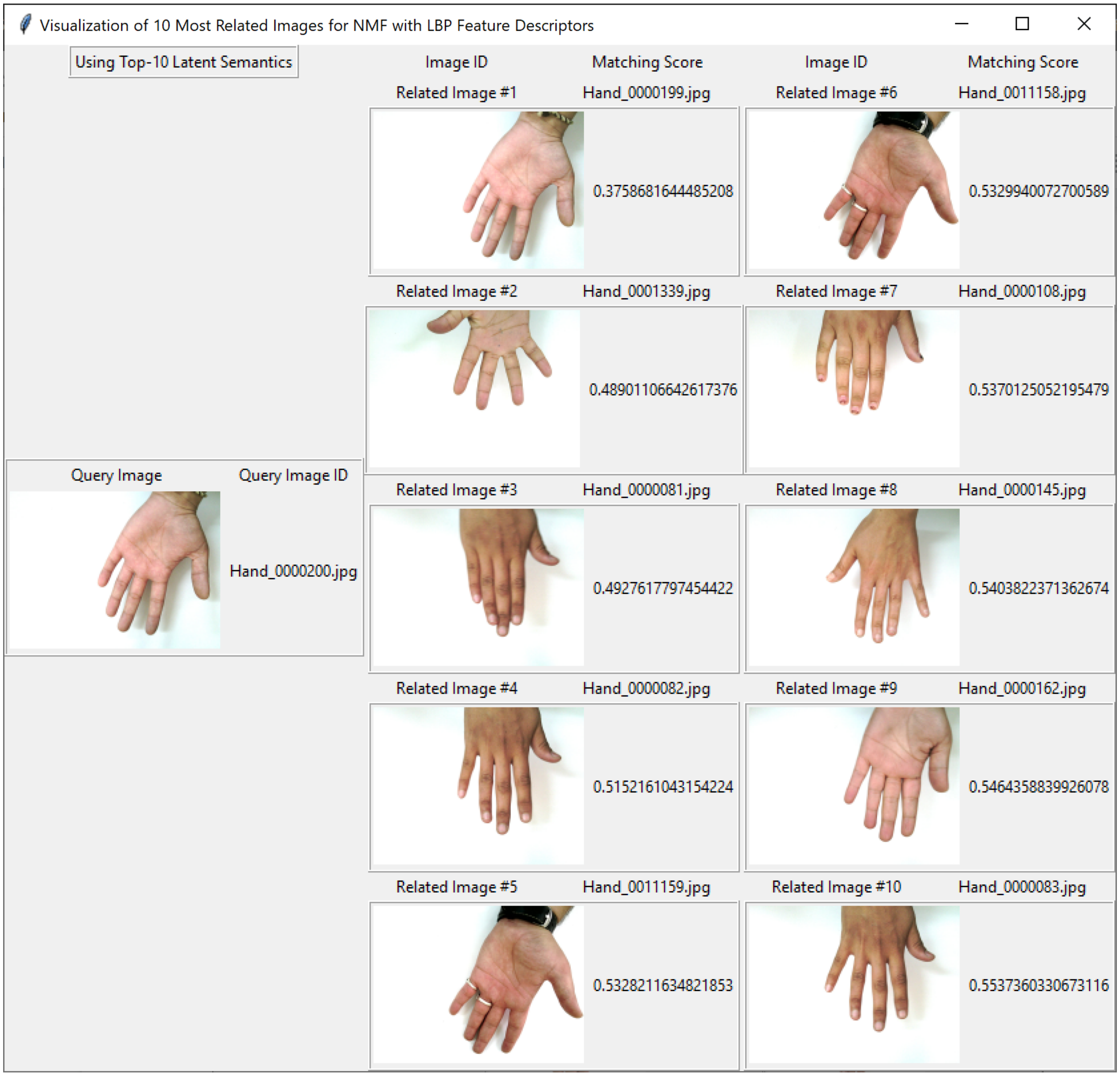
6.2 Task 2

Query 1: Feature Model: HOG, K: 10, M: 10, Technique: PCA, Image ID: Hand_0000111.jpg

Query 2: Feature Model: HOG, K: 40, M: 10, Technique: PCA, Image ID: Hand_0000111.jpg



Query 3: Feature Model: LBP, K: 10, M: 10, Technique: NMF, Image ID: Hand_0000200.jpg



Query 4: Feature Model: LBP, K: 40, M: 10, Technique: NMF, Image ID: Hand_0000200.jpg

6.3 Task 3

Query 1: Feature Model: HOG, K: 20, Technique: PCA, Label: Dorsal

Visualization of Top-20 Feature-Latent Semantics for PCA with HOG Feature Descriptors and label: dorsal

Latent Semantic 1		Latent Semantic 2		Latent Semantic 3		Latent Semantic 4		Latent Semantic 5	
Feature Identifier	Feature Score	Feature Identifier	Feature Score	Feature Identifier	Feature Score	Feature Identifier	Feature Score	Feature Identifier	Feature Score
12	0.21268679	11	0.45185096	4	0.7477736	21	0.38399942	12	0.61655127
17	0.10843012	21	0.34934758	11	0.16434175	4	0.28662848	17	0.23896453
11	0.09076973	25	0.29897154	25	0.11099669	2	0.26554449	35	0.20909285
35	0.07868542	12	0.11359952	17	0.10996126	39	0.23217956	0	0.1595769
25	0.06501561	32	0.11033244	3	0.08565604	13	0.14989916	21	0.12514597

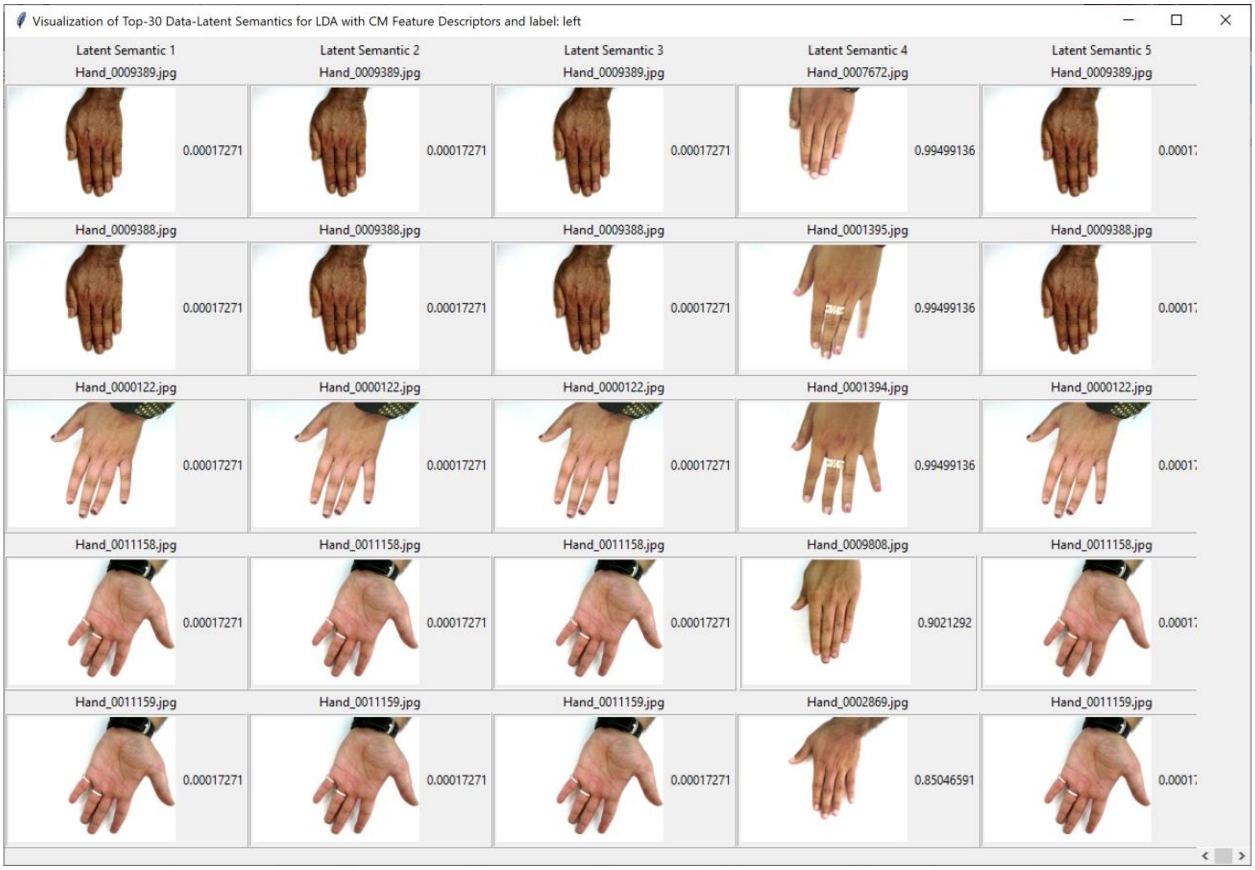
Query 2: Feature Model: HOG, K: 30, Technique: PCA, Label: Dorsal

Visualization of Top-30 Feature-Latent Semantics for PCA with HOG Feature Descriptors and label: dorsal

Latent Semantic 1		Latent Semantic 2		Latent Semantic 3		Latent Semantic 4		Latent Semantic 5	
Feature Identifier	Feature Score	Feature Identifier	Feature Score	Feature Identifier	Feature Score	Feature Identifier	Feature Score	Feature Identifier	Feature Score
12	0.21268679	11	0.45185096	4	0.7477736	21	0.38399942	12	0.61655127
17	0.10843012	21	0.34934758	11	0.16434175	4	0.28662848	17	0.23896453
11	0.09076973	25	0.29897154	25	0.11099669	2	0.26554449	35	0.20909285
35	0.07868542	12	0.11359952	17	0.10996126	39	0.23217956	0	0.1595769
25	0.06501561	32	0.11033244	3	0.08565604	13	0.14989916	21	0.12514597

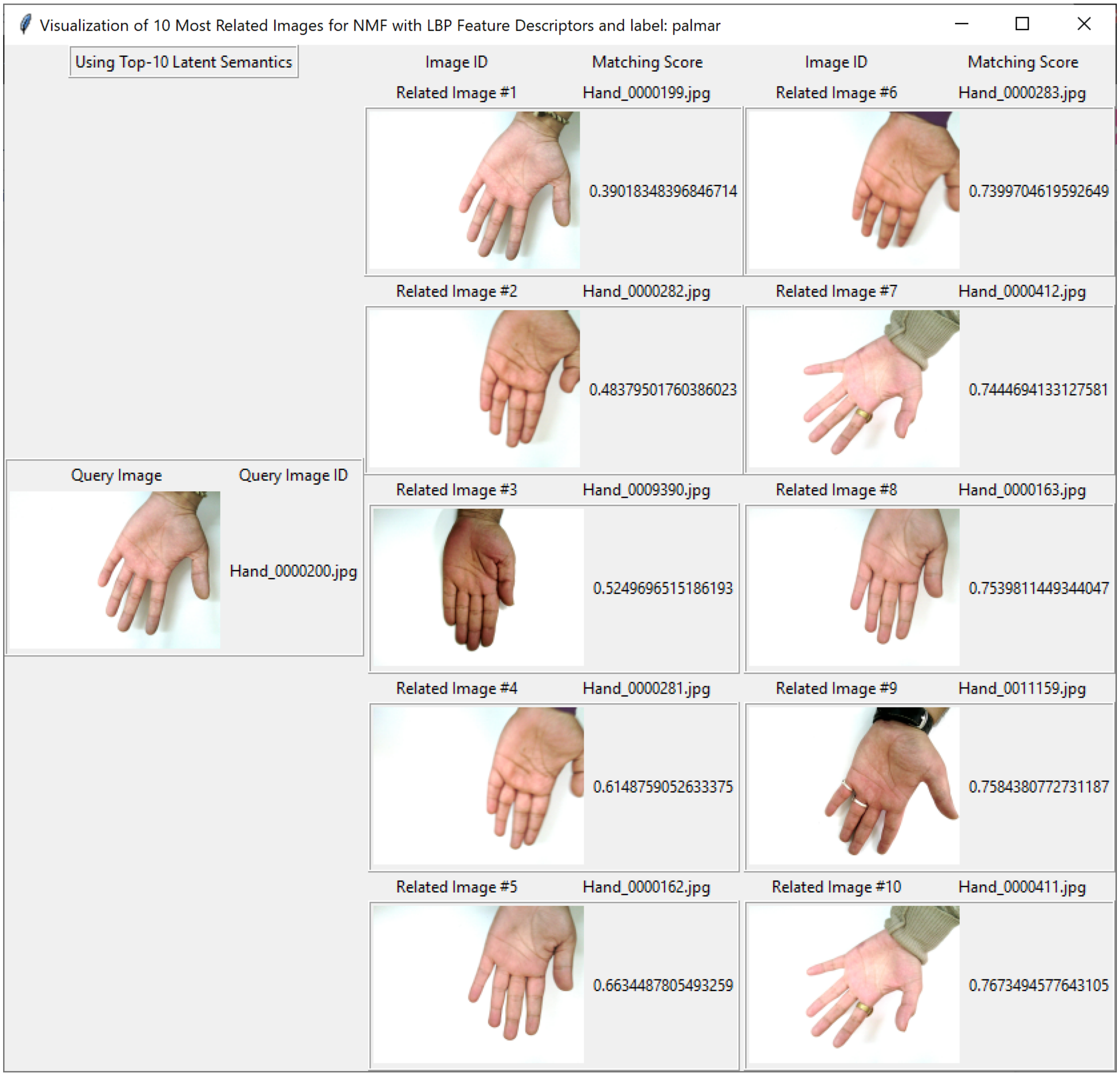
Query 3: Feature Model: Color Moments, K: 20, Technique: LDA, Label: Left

Query 4: Feature Model: Color Moments, K: 30, Technique: LDA, Label: Left



6.4 Task 4

Query 1: Feature Model: LBP, K: 10, M:10, Technique: NMF, Image ID: Hand_0000200.jpg, , Label: Palmar



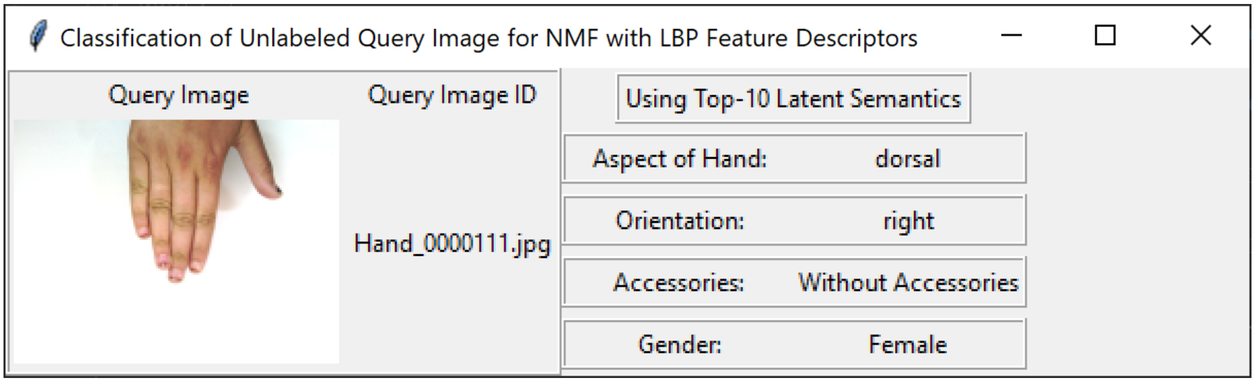
Query 2: Feature Model: LBP, K: 30, M:10, Technique: NMF, Image ID: Hand_0000200.jpg, Label: Palmar

Query 3: Feature Model: SIFT, K: 10, M:10, Technique: SVD, Image ID: Hand_0011160.jpg, Label: With Accessories

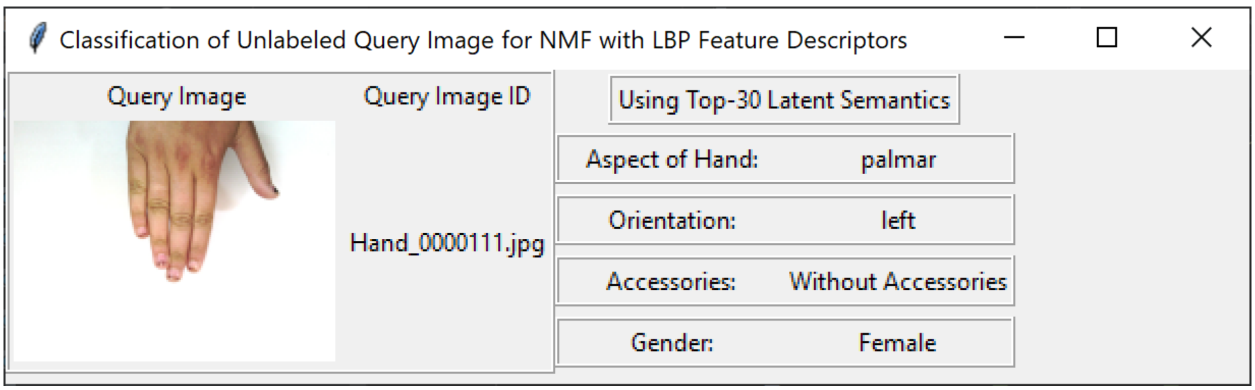
Query 4: Feature Model: SIFT, K: 30, M:10, Technique: SVD, Image ID: Hand_0011160.jpg, Label: With Accessories

6.5 Task 5

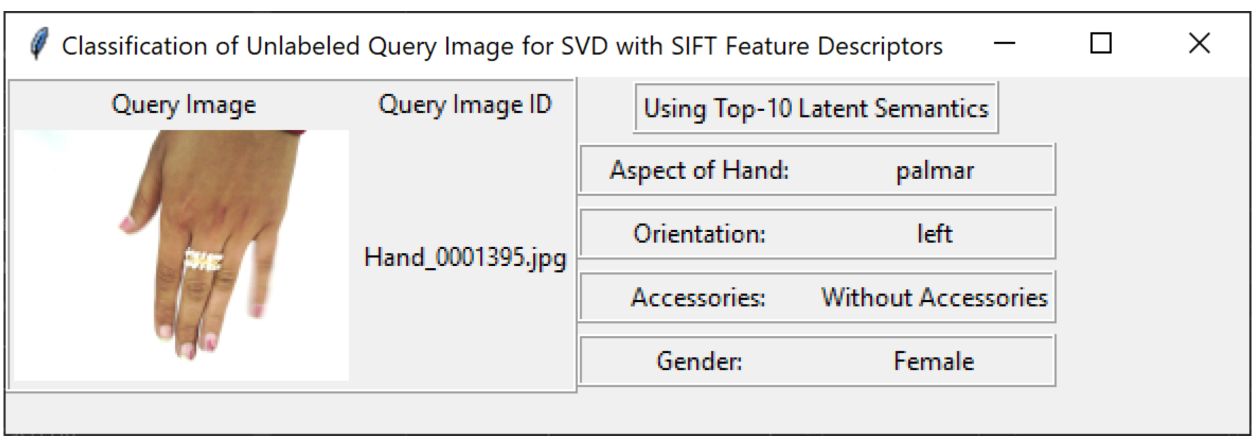
Query 1: Feature Model: LBP, K: 10, Technique: NMF, Image ID: Hand_0000111.jpg



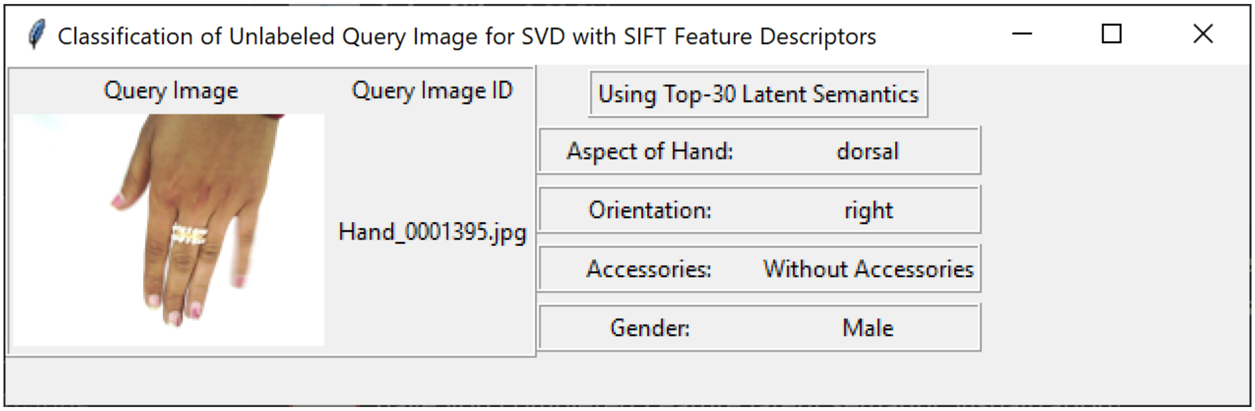
Query 2: Feature Model: LBP, K: 30, Technique: NMF, Image ID: Hand_0000111.jpg



Query 3: Feature Model: SIFT, K: 10, Technique: SVD, Image ID: Hand_0001395.jpg

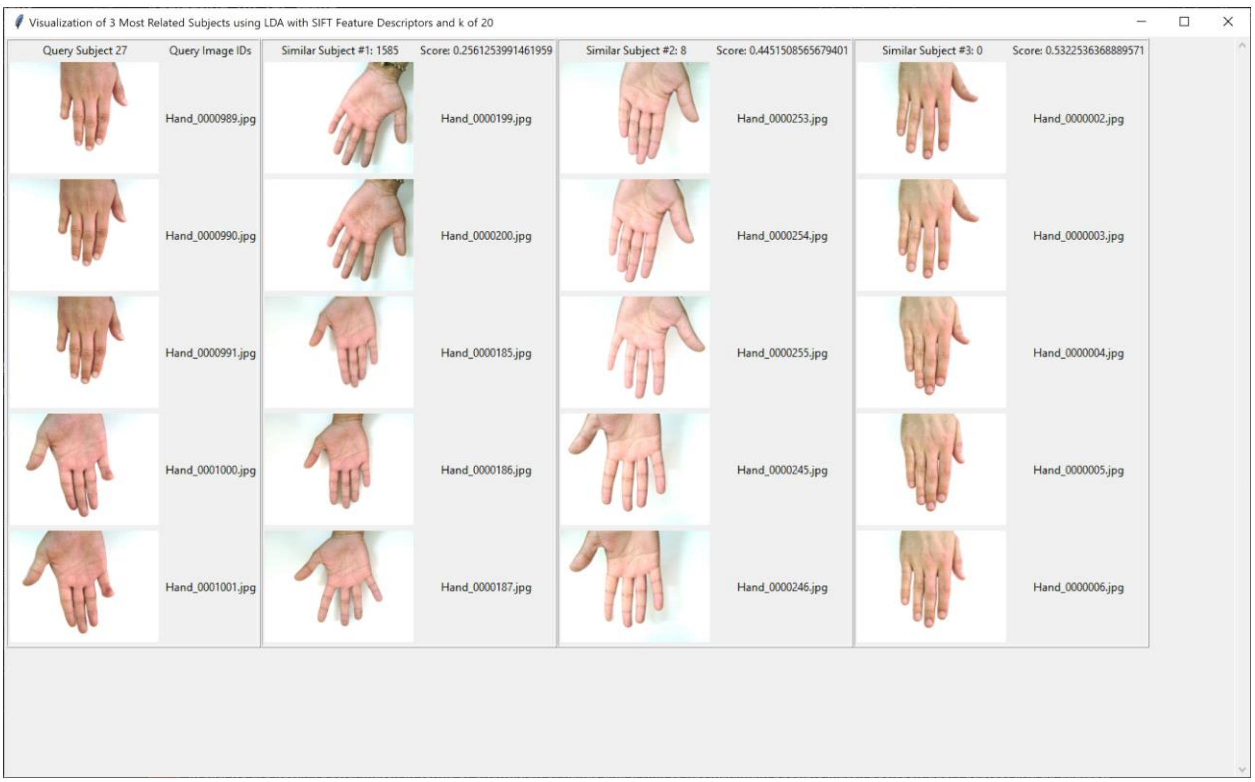


Query 4: Feature Model: SIFT, K: 30, Technique: SVD, Image ID: Hand_0001395.jpg

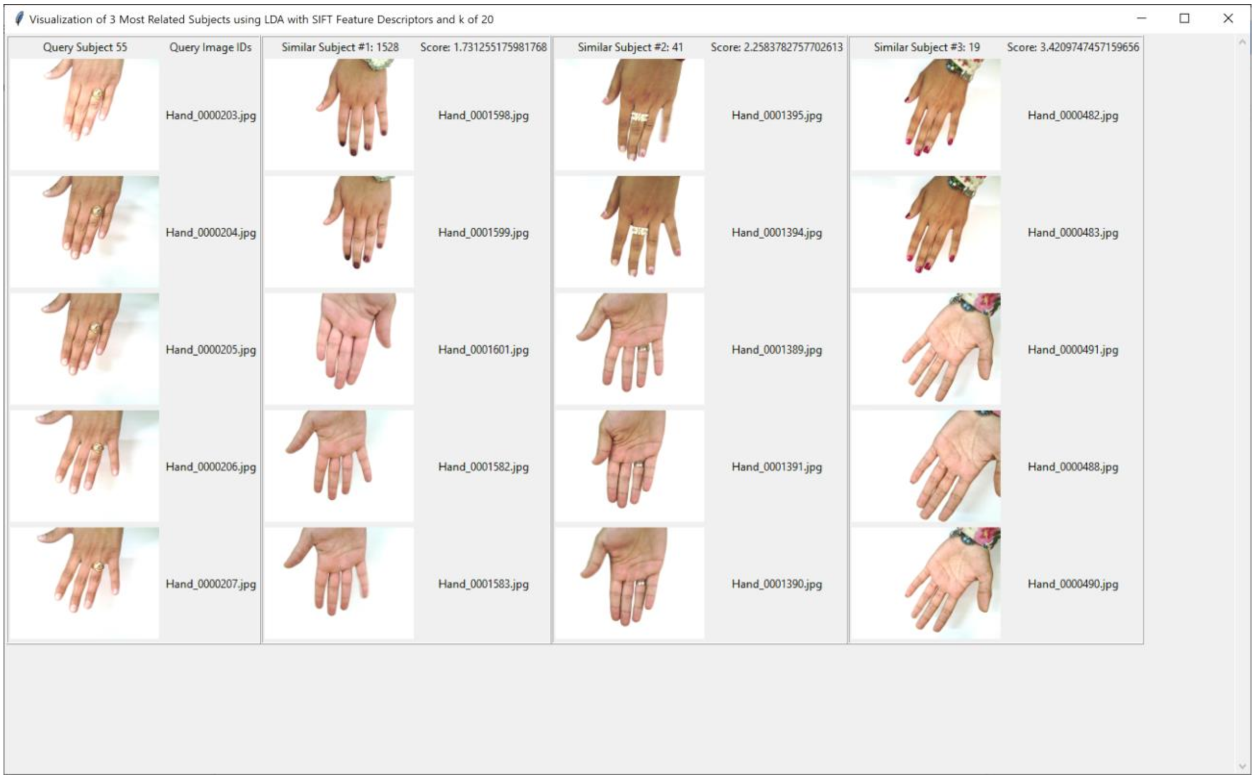


6.6 Task 6

Query 1: Subject ID: 27



Query 2: Subject ID: 55



6.7 Task 7

Query 1: K: 10

Visualization of Top-10 Latent Semantics as subject-weight pairs

Latent Semantic 1		Latent Semantic 2		Latent Semantic 3		Latent Semantic 4		Latent Semantic 5	
Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score
55	0.93303202	1535	0.90241194	65	0.84701687	1516	0.79856801	514	0.72861353
1073	0.69759334	529	0.73670333	0	0.82006907	1585	0.77533317	27	0.72733334
19	0.66264317	1506	0.69294374	504	0.78640574	8	0.54822093	594	0.68398789
1011	0.63695003	595	0.67825759	559	0.73152768	1000000	0.47476492	529	0.66501757
1528	0.62001986	92	0.6171878	1081	0.70635672	9000000	0.45296055	1509	0.63073262
65	0.49185856	1022	0.58572477	1009	0.64986379	1560	0.43724464	1528	0.56025256
9000000	0.41243731	0	0.56546052	505	0.63838646	1009	0.36616626	1506	0.55556636
1585	0.29869693	1009	0.36474277	9000000	0.53971237	594	0.35610612	1585	0.52523782
41	0.29241352	1516	0.34688926	1513	0.51817982	504	0.25903	8	0.46885625
81	0.2672737	1513	0.30536019	1011	0.4828681	1543	0.25816284	1011	0.46094239
1516	0.25910659	1509	0.26906371	1528	0.27591842	27	0.23007612	1022	0.41788126
594	0.19585418	1000000	0.26901048	1560	0.25289545	505	0.18874929	28	0.41570465
1515	0.18445378	1073	0.2338031	1000000	0.19259266	514	0.17957453	9000000	0.41277831
57	0.15407152	9000000	0.22556435	28	0.17031832	1073	0.15573501	1535	0.39313282
1513	0.08508377	1074	0.14622425	1516	0.15559256	1513	0.14955343	1074	0.34815449
1543	0.08405695	41	0.11575595	595	0.15474472	1081	0.12997255	1560	0.28638792
505	0.08009292	1011	0.09988537	25	0.14236912	1011	0.11579257	559	0.26080632
504	0.04618204	559	0.0607999	1074	0.13266876	1022	0.10079026	1024	0.19443698
1560	0.00139537	1081	0.05987736	41	0.11926047	54	0.09159362	1513	0.19310102
0	0.0	505	0.05869759	1506	0.10315769	25	0.08951051	41	0.1455587
559	0.0	1560	0.05563849	1022	0.10219677	1528	0.0295025	1081	0.12183581
8	0.0	28	0.04587046	1543	0.08699659	57	0.02419079	19	0.1163351
25	0.0	25	0.04379416	514	0.04414707	19	0.01643684	57	0.09778402
54	0.0	65	0.0	594	0.0	1515	0.0037071	505	0.09188548
529	0.0	594	0.0	1585	0.0	0	0.0	1543	0.08121729
27	0.0	1585	0.0	55	0.0	559	0.0	54	0.06934926
28	0.0	55	0.0	8	0.0	65	0.0	92	0.03143668
1535	0.0	8	0.0	54	0.0	55	0.0	504	0.02811922
1074	0.0	54	0.0	1515	0.0	81	0.0	0	0.02663267
514	0.0	1515	0.0	81	0.0	529	0.0	1073	0.02082203

Query 2: K: 20

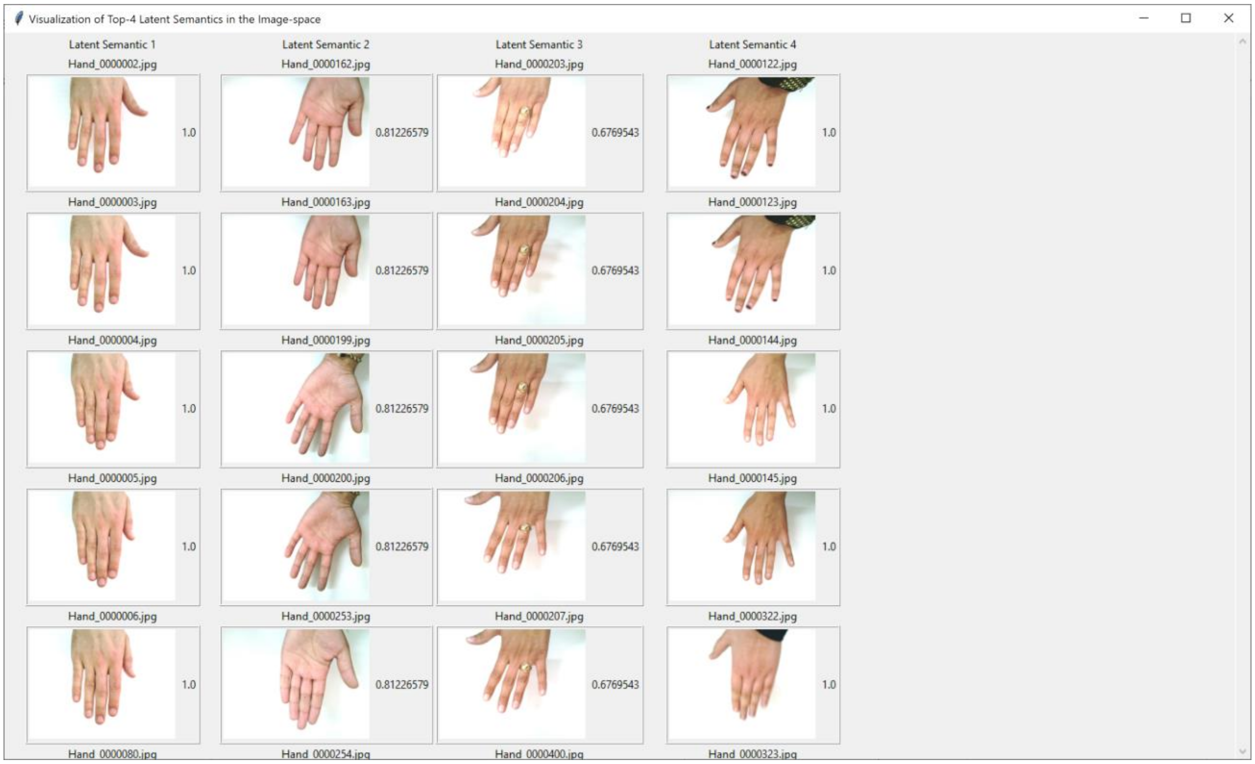
Visualization of Top-20 Latent Semantics as subject-weight pairs

Latent Semantic 1		Latent Semantic 2		Latent Semantic 3		Latent Semantic 4		Latent Semantic 5	
Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score
594	0.51619993	559	0.65845461	1000000	0.72952352	1515	0.90128947	1535	0.88003316
1073	0.49284935	1081	0.62057631	25	0.65329774	81	0.89826288	1506	0.84337602
57	0.40835444	504	0.60760628	1009	0.62024703	19	0.55035917	1022	0.65340623
1585	0.39614443	0	0.58264143	1516	0.52456858	1081	0.43848297	529	0.56840763
1528	0.39167089	595	0.50009714	1543	0.49062625	1073	0.3437028	1513	0.41232124
514	0.37782143	65	0.41534645	1513	0.38972846	595	0.29798338	28	0.40073332
8	0.36459809	1011	0.3939284	8	0.38542931	1011	0.29339051	1011	0.38542653
41	0.34657157	505	0.30379808	1011	0.2503999	28	0.28577693	27	0.1779478
9000000	0.34560027	8	0.28489037	54	0.11455548	57	0.16048295	595	0.15934622
1022	0.30138517	1513	0.21932234	1560	0.06521449	1509	0.1534555	1516	0.10622523
504	0.29994296	28	0.21348739	1535	0.0582273	1585	0.13346344	1009	0.09169927
1081	0.27768962	27	0.15141679	1585	0.0502858	1022	0.07370561	9000000	0.03181607
1000000	0.20802082	1560	0.14338581	92	0.04468402	594	0.07097244	1528	0.01858044
1024	0.16347742	9000000	0.1327141	0	0.02860163	1074	0.07083339	1000000	0.00760608
1011	0.15546237	1074	0.12495529	505	0.02535018	1024	0.05113636	0	0.0
65	0.15067763	1516	0.11823387	9000000	0.01535173	1506	0.04452278	559	0.0
505	0.14937042	25	0.10938819	559	0.0	25	0.04221625	65	0.0
54	0.14554698	1009	0.02143548	65	0.0	1009	0.01918765	505	0.0
55	0.12293114	1509	0.01060341	594	0.0	9000000	0.00481697	594	0.0
1516	0.0779634	514	0.00523335	55	0.0	92	0.00178241	1585	0.0
1543	0.074503	594	0.0	1515	0.0	0	0.0	55	0.0
1506	0.03197311	1585	0.0	81	0.0	559	0.0	8	0.0
28	0.02565389	55	0.0	1073	0.0	65	0.0	25	0.0
19	0.02443482	54	0.0	19	0.0	505	0.0	54	0.0
1074	0.02394911	1515	0.0	57	0.0	55	0.0	1515	0.0
1515	0.00387967	81	0.0	529	0.0	8	0.0	81	0.0
0	0.0	1073	0.0	27	0.0	54	0.0	1073	0.0
559	0.0	19	0.0	28	0.0	529	0.0	19	0.0
25	0.0	57	0.0	1074	0.0	27	0.0	57	0.0
81	0.0	529	0.0	514	0.0	1535	0.0	1074	0.0

6.8 Task 8

Query 1: K: 4

Top-k Latent Semantics in the image-space



Top-k Latent Semantics in the metadata-space

Visualization of Top-4 Latent Semantics in the Metadata-space

Latent Semantic 1		Latent Semantic 2		Latent Semantic 3		Latent Semantic 4	
Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score
Orientation	0.80788345	aspectOfHand	0.84113723	accessories	1.27515252	gender	1.17143303
aspectOfHand	0.0	accessories	7e-08	gender	0.00134045	Orientation	0.0
gender	0.0	Orientation	0.0	aspectOfHand	0.0	aspectOfHand	0.0
accessories	0.0	gender	0.0	Orientation	0.0	accessories	0.0

Query 2: K: 6

Top-k Latent Semantics in the image-space

Top-k Latent Semantics in the metadata-space

Visualization of Top-6 Latent Semantics in the Metadata-space

Latent Semantic 1		Latent Semantic 2		Latent Semantic 3		Latent Semantic 4		Latent Semantic 5		Latent Semantic 6	
Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score	Subject Identifier	Subject Score
gender	0.62952744	Orientation	0.53711081	accessories	1.00476367	gender	0.81497091	aspectOfHand	1.92869576	Orientation	1.61434073
Orientation	0.31255938	aspectOfHand	0.12334608	gender	2.34e-06	aspectOfHand	8.21e-06	accessories	0.0	accessories	0.0
aspectOfHand	0.0	accessories	0.0	aspectOfHand	0.0	Orientation	0.0	Orientation	0.0	aspectOfHand	0.0
accessories	0.0	gender	0.0	Orientation	0.0	accessories	0.0	gender	0.0	gender	0.0

7 Conclusion

In this phase, we experimented with vector models, image features and various dimensionality reduction techniques. Dimensionality reduction is usually performed in high dimensional datasets to find a lower dimensional representation of the dataset such that as much information as possible of the original dataset is preserved. Reducing the dimensionality of the dataset will result in reducing the computation complexity along with the inherent noise present in the dataset, thus enhancing the performance. In this project, we are using just the distance measures rather than any clustering method to classify the images we have. The use of such a simplistic distance/similarity measure proves to be inaccurate as the results that we are getting for classification seems to be inaccurate some of the images. So, we need better strategies or algorithms for better classification of the image. In this phase, we have learned how the dimensions of an image can be reduced without compromising the quality. Also, during this phase of the project, the major take away was learning how to use various techniques like PCA, SVD, LDA and NMF. In Addition, we switched from using the conventional csv file to using the MongoDB. This has significantly reduced the total computation time thus increasing the efficiency of the tasks execution.

8 Reference

[1] Yun Xuea , Chong Sze Tongb , Tiechen Li, “Evaluation of Distance Measures For NMF-Based Face Image Applications” JOURNAL OF COMPUTERS, VOL. 9, NO. 7, pp. 1704-11, JULY 2014

[2] Daniel D. Lee & H. Sebastian Seung, “Learning the parts of objects by non-negative matrix factorization” NATURE , VOL 401, pp 788-91, 21 OCTOBER 1999

[3] Principal component analysis to reduce dimension on digital image , School of Information Technology, SEGi University, PJU 5, Kota Damansara, Selangor 47810, Malaysia.

[4] Face Recognition Using Kernel Principal Component Analysis, Kwang In Kim, Keechul Jung, and Hang Joon Kim, IEEE SIGNAL PROCESSING LETTERS, VOL. 9, NO. 2, FEBRUARY 2002

[5] Unsupervised Classification of Hyperspectral Images using Latent Dirichlet Allocation by Thomas Boggs
http://blog.bogatron.net/blog/2014/07/16/unsupervised-hsi-classification-using-lda/

[6] Latent Dirichlet Allocation in Web Spam Filtering, István Bíró, Jácint Szabó, & András A. Benczúr, Computer and Automation Research Institute of the Hungarian Academy of Science

[7] Singular value decomposition and principal component analysis, Michael E. Wall, Andreas Rechtsteiner, Luis M. Rocha, Los Alamos, New Mexico 87545, USA.

[8] The Why and How of Non Negative Matrix Factorization by Adrian Coyler
https://blog.acolyer.org/2019/02/18/the-why-and-how-of-nonnegative-matrix-factorization/

[9] Making sense of the Kullback–Leibler (KL) Divergence by Marko Cotra,
https://medium.com/@cotra.marko/making-sense-of-the-kullback-leibler-kl-divergence-b0d57ee10e0a

[10] *https://www.wikiwand.com/en/Non-negative_matrix_factorizatio*

9 Appendix

Specific roles of team members.

1. Anjali: Database Implementation, Testing, LDA implementation, Documentation
2. Athul: Testing, NMF implementation, Documentation
3. Manoj: PCA implementation for all the tasks
4. Md Shadab: NMF implementation for all the tasks
5. Prashant: SVD implementation for all the tasks
6. Tyler: Visualization Creation, LDA implementation, Documentation