CSE 515 MULTIMEDIA AND WEB DATABASES
PROJECT PHASE 2 REPORT

Kovidnath Reddy, Pyla
kpyla@asu.edu

Tharun Kumar Viralam Sreedhar
tviralam@asu.edu

Vishal Tyagi
vtyagi7@asu.edu

Anh The Nguyen
atnguy37@asu.edu

Dhriti Anil Shah
dshah25@asu.edu

Rinku Nemade
rnemade@asu.edu

ABSTRACT

Multimedia database has a huge amount of information which becomes difficult to compute as the size of database increase. As the dimensionality of databases increases the volume of the data increases so fast that the data becomes sparse. But sparsity is an issue if our application needs features which are dependent on statistical metrics. Therefore, it is necessary to handle such huge amount of data efficiently. In Phase 2 of the project main aim is to handle the data using dimensionality reduction. Four types of dimensionality reduction methods are considered. Latent Dirichlet Allocation(LDA) ,Principal component analysis(PCA), Singular value decomposition(SVD), Non-negative matrix factorization(NMF) methods are used in this phase.

***K*eywords** Multimedia Web Databases · HOG (Histogram of Oriented Gradients) · SIFT (Scale-invariant feature transform) · Euclidean Distance · Lowe's ratio test · Image Vector Models · Image Database Search · Image similarity · LDA (Latent Dirichlet Allocation) · PCA (principal component analysis) · SVD (singular value decomposition) · NMF (non-negative matrix factorization)

1 Introduction

In phase 1, we performed similar image search using image features. This search is very expensive as the dimensions of the features are very huge. The cost of these calculations can be reduced by using dimentionality reduction algorithms which minimize the loss. In this phase, we have implemented dimensionality reduction techniques - PCA, LDA, NMF, SVD and selected top latent semantics for given dataset. In this phase, we have also used labelled dataset from tasks 3 to 5. The features used are Color moments, HOG, LBP, SIFT. Each of these dimensionality reduction techniques preserve some property of data. NMF preserves the positivity of the original data. PCA and LDA preserves the variance in the original data. SVD preserves the data itself. Today's world produces data at a very rapid rate, especially multimedia data due to all the social media. Due to this there is a huge need for efficient way to index these images and find efficient methods to store them and find the relevance with other images. In this phase of the project we will be implementing Histogram of Oriented gradients (HOG) and Scale-invariant feature transform (SIFT) to retrieve the image feature-set. We also use PosrgreSQL to save these image features for quick and efficient data retrieval. In the end we implement L2 distance or otherwise known as Euclidean distance to find the distance measure and we use Lowe's ratio test to select features between images to find the optimal key points to use when finding similarities in SIFT.

1.1 Terminology

1. Feature descriptor - in terms of computer vision feature descriptors are basic image elements such as shape, color, texture or motion in images and videos.
2. HOG - Histogram of Oriented gradients, is an image feature extraction model which is mainly used for object detection.
3. SIFT - Scale Invariant Feature Transform [1], is an image feature extraction model which is used to describe the local features.

4. Euclidean Distance - Is a similarity measure which measures the distance between two vectors. It is also referrer to as the L2 distance measure and is defined as the sum of the squares of differences between corresponding features of two vectors.
5. LDA: LDA is a generative probabilistic model for collections of discrete data.[1] LDA is most commonly used to categorize text data. LDA is a bag-of-words approach - the order of words (features here) is neglected.
6. PCA: Principal Component Analysis is a statistical method which is well known for dimensionality reduction. It uses orthogonal transformation to convert the image dataset into a set of linearly uncorrelated variables called principal components.
7. NMF: Non Negative Matrix Factorization is a multivariate analysis algorithm where a non negative matrix is decomposed into two matrices which approximately result in the original matrix on multiplication.
8. SVD: In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix. It is the generalization of the eigendecomposition of a positive semidefinite normal matrix (for example, a symmetric matrix with non-negative eigenvalues) to any $m \times nm \times nmatrixviaanextensionofthepolardecomposition$[9].

1.2 Problem Specification

In the second phase of the project we will use the feature descriptors calculated in Phase 1 and reduce to dimensionality of the data to avoid curse of dimensionality problem. PCA ,SVD ,NMF ,LDA dimensionality methods are used in Phase 2. Since each of these models result in a very large feature space, there is a requirement to reduce the dimensionality so as to not be affected by the dimensionality curse. Each of the techniques decompose the original image data matrix into a pair of data latent semantics and feature latent semantic matrices. The data latent semantics can be reordered by the weights in the basis vectors to determine the contribution done by each image to the latent semantics. Similarly the feature latent semantics can be multiplied with each image to determine the placeholder image which contributes the most in terms of features to the latent semantics.

The purpose of task 1 is to implement a program which reports the top k latent semantics of the images data by using dimensionality reduction methods. Task 2 will use this k latent semantics of the images data and find m most related images to given image id by calculating matching scores. The purpose of Task 3 is to identify the k latent semantics of the images with chosen metadata as input by the user. In Task 4 we will calculate matching scores and find m most related images to given image id using metadata selected and k latent semantics calculated.

1.3 Assumptions

1. The dataset that is being used in out project is from 11k hands dataset. Being said so, we can assume that the resolution of all the images are same in order to perform the similarity calculations. In this dataset there are 11,076 images and each of the size (1600x1200) pixels, and the images of the hands are from both male and female test subjects of the age ranging from 18 years to 75 years.
2. Loss of information is possible when there is feature extraction, dimensionality reduction and hence we will be neglecting the inaccuracies caused due to this.
3. Manual verification of similarities are performed as to find if the similar images are truly similar.

2 Implementation

2.1 Task 1

The purpose of this task is to implement a program which reports the top k latent semantics of the images data. This task takes input from the previous Phase of the project, which takes the pixel data of the images and converts them into features using one of the following feature model conversions:

- Image Moments
- LBP
- SIFT
- HOG

Since each of these models result in a very large feature space, there is a requirement to reduce the dimensional- ity so as to not be affected by the dimensionality curse. Each of the techniques decompose the original image data matrix into a pair of data latent semantics and feature latent semantic matrices. The data latent semantics can be re-ordered by the weights in the basis vectors to determine the contribution done by each image to the latent semantics. Similarly, the feature latent semantics can be multiplied with each image to determine the placeholder image which contributes the most in terms of features to the latent semantics.

PCA: Principal Component Analysis is a statistical method which is well known for dimensionality reduction. It uses orthogonal transformation to convert the image dataset into a set of linearly uncorrelated variables called principal components. The principal components are determined by estimating the direction of most variance. PCA decomposes the covariance matrix of the image dataset into eigen vectors and eigen values. The vectors corresponding to the k biggest eigen values are used as the principal components. After calculating the principal components the images are projected onto the new reduced space. The image visualizer is used to display a ranked list of the images based on the weights in the eigen vectors for both the image space and the feature space.

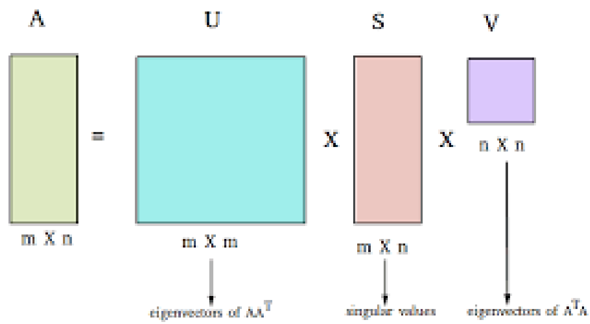


Figure 1: Principal Component Analysis

NMF: Non Negative Matrix Factorization is a multivariate analysis algorithm where a non negative matrix is decomposed into two matrices which approximately result in the original matrix on multiplication. The matrices are approximated iteratively by using an optimizer function to update the values of the matrices. The initial martices are estimated by randomizing the values. The loss function used for updating the values is the frobenius function generally considered as the Euclidean distance. The optimization function used here is the beta divergence function. The two matrices that are decomposed from the original matrix can be used as the basis vectors for the data. The image visualizer is used to display a ranked list of the images based on the weights in the eigen vectors for both the image space and the feature space.

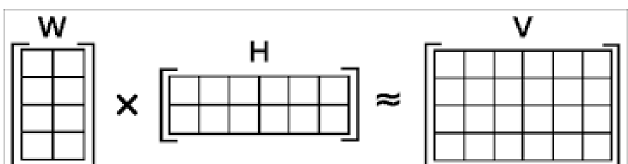


Illustration of approximate non-negative matrix factorization: the matrix *V* is represented by the two smaller matrices *W* and *H*, which, when multiplied, approximately reconstruct *V*.

Figure 2: Non-Negative Matrix Factorization

SVD: In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix. It is the generalization of the eigen decomposition of a positive semi definite normal matrix (for example, a symmetric matrix with non-negative eigenvalues) to any m x n matrix via an extension of the polar decomposition. It has many useful applications in signal processing and statistics[9].

LDA: Latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word’s presence is attributable to one of the document’s topics.[8] LDA was originally developed for text with an assumption that the ordering of words in the text can be neglected. But LDA model is modular and not necessarily

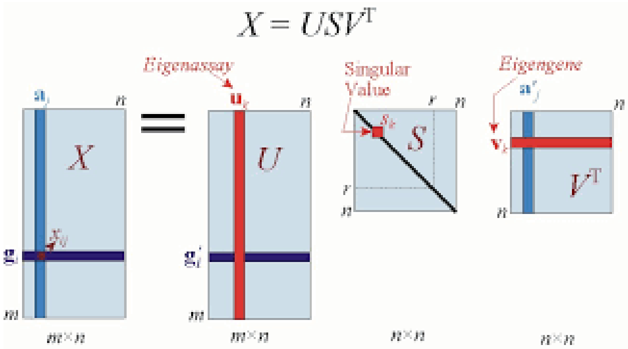


Figure 3: Difference between Singular Value Decomposition and Principal Component Analysis

tied to text, and has applications to other problems involving collections of data, including data from domains such as collaborative filtering, content-based image retrieval.[6] We only observe the feature values within the images and the other structures are hidden variables[7]. In figure 4, the model of LDA is explained using plate notation as it is a probabilistic graphical models, the dependencies among the many variables can be captured concisely. The boxes are "plates" representing replicates, which are repeated entities. The outer plate represents documents, while the inner plate represents the repeated word positions in a given document, each of which position is associated with a choice of topic and word. M denotes the number of documents, N the number of words in a document.W is grey which specifies that W is the only observed variables and others are hidden variables. The variable names are defined as follows[8]:

- α is the parameter of the Dirichlet prior on the per-document topic distributions,
- β is the parameter of the Dirichlet prior on the per-topic word distribution,
- Theta is the topic distribution for document i,
- z is is the topic for the j-th word in document i,
- w is the specific word

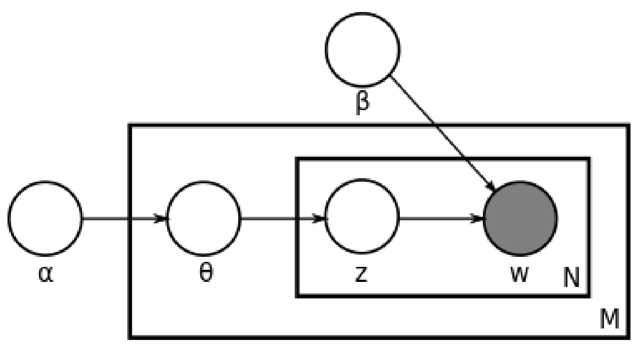


Figure 4: Latent Dirichlet Allocation Model

class sklearn.decomposition. LatentDirichletAllocation(n_components=k, learning_method='online', ran- dom_state=None,doc_topic_prior=None, topic_word_prior=None) No of components i.e. No of latent se- mantics are taken as input from the user. We have a large data set and hence it is preferred to use online learn- ing method='online' is used if the data is large else learning method='batch' is used.random_state=None sig- nifies that the random number generator is the RandomState instance used by np.random. Alpha and Beta i.e. doc_topic_prior=None, topic_word_prior=None respectively have default values 1/no_of_components.We also tried Alpha=0.5 and Beta=0.01 but the default value selection gave better results.

Handling negatives values for NMF and LDA: Being probabilistic models, NMF and LDA do not work for object- feature matrix with negative values. Out of all the features we have considered, color moments can contain negative values for 3rd color moment (skewness). To handle this we have tried two techniques - normalization and counting frequencies. For normalization, we have used following functions:

- Exponential: $\frac{1}{e^x}$

(where x is value in the color moment feature matrix)

- Normalization to range (0,1):

$$\frac{x - min}{max - min}$$

(where min and max are the minimum and maximum values in the color moments matrix, x is value in the color moment feature matrix)

This function preserves the data ordering giving all the positive values. Hence, is a good choice for pre-processing data for NMF and LDA. The alternative method to handle negative values is to cluster the input feature matrix using K-means and calculate histograms.The third alternative does not give accurate results intuitively compared to previ- ous methods applied. For counting frequencies, we have created 100 bins for each of the 1st, 2nd and 3rd color mo- ment. These bins are bounded by minimum and maximum values for each of these color moments. This technique reduces the dimensions of the color moments matrix to 3 rows. This method is not as effective as normalization and causes data loss.

2.2 Task 2

The purpose of this task to find m related images from k latent semantics of the images data. First step is to find feature descriptor using one of the following feature model conversions:

- Image Moments
- LBP
- SIFT
- HOG

Next step is to reduce the data dimensionality into k latent semantics using one of the following dimensionality reduction methods:

- SVD
- PCA
- NMF
- LDA

Third step is to find Matching scores of the given image id as input to other images in the database and find m similar images.Different similarity metrics can be used to calculate distance metrics.L2 norm (Eucledian Distance) distance metric is used to find the similarity.

2.3 Task 3

In addition to task 1, this task takes label and dimensionality reduction technique as input. Then performs the dimen- sionality reduction on the images from labelled dataset. The images are labelled as:

- left-hand
- right-hand
- dorsal
- palmar
- with accessories
- without accessories
- male

- female These labels for images are defined in HandInfo.csv.

This task outputs k latent semantics for images with the corresponding metadata using, one of the following techniques:

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

2.4 Task 4

In addition to Task 3, Task 4 calculates Matching scores of the given image id as input to other images in the database and find m similar images. L2 norm distance metric is used to find the similarity.

2.5 Task 5

The purpose of this task is to implement a program which takes in an unlabeled image and labels it as one of the following:

- Left hand vs Right hand
- Dorsal vs Palmar
- With vs Without accessories
- Male vs Female

Also, for this task only one of the labels will be provided to perform a classification. These labels could be one the following:

- left-hand
- right-hand
- dorsal
- palmar
- with accessories
- without accessories
- male
- female

As only one of the label will be provided as input to the program, for example left hand, this task can be identified as a unary classification task. That is we classify whether the image is a left hand or not. Since the image data we are provided with is an output from task 3, meaning that the provided dataset has to been projected onto the latent semantic space that is calculated by performing one of the dimensionality reductions methods(like PCA, NMF, LDA or SVD). Therefore the image provided is also projected onto this space so that the comparison done between the given image and the hands images is more accurate in the same space(reduced dimensionality).

In order to achieve this task we tried out multiple methods, such as:

- Centroid approach
- Local Outlier Factor (LOF)
- One Class SVM

2.5.1 Centroid approach

In this approach the centroid of the image data for the label is calculated and the distance from the furthest point is taken as the diameter to construct a circular boundary. This boundary is used to estimate if the given image can be considered as belonging to the label or not. Since the data even after dimensionality reduction can be considered as high dimensional data, using such a simplistic approach to classify the given image was

providing low accuracy
results. This could also be because constructing a circular boundary using the furthest point is not a very strict bound- ary thus providing inaccuracies.

2.5.2 Local Outlier Factor

This is traditionally an anomaly detection algorithm that was proposed by Markus M. Breuig etal, which is employed to find anomalies in data by measuring local deviation of a data point given its neighbors. The concept of reachabil- ity distance is used here, which can be defined as the max of the L2 distance between two points and the k distance of the point being considered. Local densities are calculated for the image using this approach and it is estimated if the point is indeed an anomaly or not, which in our case determines if the image belongs to the label or not. Consider k-distance(A) be the distance of an object A to k-th nearest neighbour. $N_k(A)$ is the set of k nearest neighbours. Here the reachability distance between points A and B is calculated as:

$$\text{reachability-distance}_k(A, B) = \max\{\text{k-distance}(B), d(A, B)\} \tag{1}$$

The local reachability density is calculated as follows:

$$\text{LOF}_k(A) := \frac{\sum_{B\in N_k(A)} \frac{\text{lrd}(B)}{\text{lrd}(A)}}{|N_k(A)|} = \frac{\sum_{B\in N_k(A)} \text{lrd}(B)}{|N_k(A)|}/\text{lrd}(A) \tag{2}$$

If $\text{LOF}_K(A)$ is around 1 means it has similar density as its neighbours. If its less than 1 its inliers and if its greater than one then its outliers.

2.5.3 One Class SVM

One class SVM uses the classic Support vector concept to create a strict decision boundary around the imagedata. It relies on creating a a small hyper-sphere consisting of all data points. RBF kernel is used to map the image data into a space where this decision boundary can be constructed. Once the boundary is constructed around the image data it becomes easy to estimate if the given image belongs to the label or not.So we assume the images as a set of i.i.d samples. Now we take in a threshold of δ, so as to fine tune the model. The formulae for this one class SVM is as follows:

$$\min_{r,c,\zeta} r^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\zeta_i \tag{3}$$

subject to, $||\Phi(x_i) - c||^2 \leq r^2 + \zeta_i \;\; \forall i = 1, 2, ..., n$ where R is the radius of the sphere, and ν in (0,1), and c is the center of the sphere.

2.6 Task 6

The purpose of this task is to identify the three most similar Subject IDs given a certain Subject ID by the user. The subject ID is a group in which the images belong as provided in the meta data file. The meta data is stored in the database as a table of the name img_meta, which contains the image mapping to the subject ID. Using this meta data information we can understand what are the images that belong to each of the subject IDs. Once this is ascertained we are given a choice of aggregating the images so as to come up with the flagship image that represents the subject ID in total. Since each of the images are represented by a vector of a particular feature model and dimensionality reduction has been done on the resulting feature space, we have chosen to perform a simple centroid calculation amongst all the images present in the particular Subject ID to represent the flagship image of the subject ID. Once the centroids have been calculated for each of the subject ids, the distance between all the centroids of each of the subject IDs are calculated. Finally, the similarity is calculated by inverting the distance metric and adding one to avoid cases of division by zero.

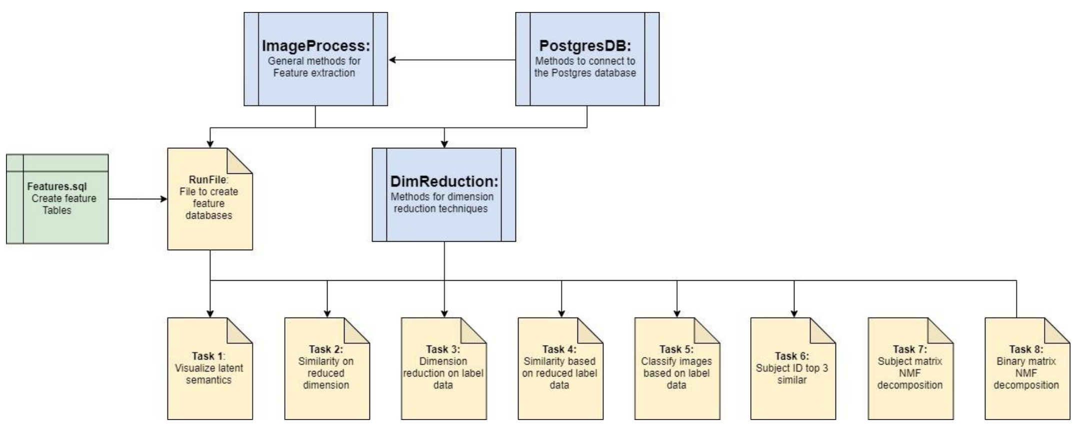
2.7 Task 7

The purpose of this task is to create the subject-subject similarity matrix and perform Non Negative Matrix factor- ization and report the latent semantics gleaned from it. The subject-subject similarity matrix is calculated using the same approach as taken for task 6, where first the distance between two subject ids is calculated by first estimating the centroids of each subject id and then calculating the distance between each centroid pair and inverting the dis- tance to result in the similarity measure. Once the Similarity matrix is constructed it is fed as input into the NMF function so as to decompose the subject-subject similarity matrix into it’s corresponding basis vector matrices.

Here we use NMF - Non-Negative matrix factorization a multivariate analysis algorithm, where we take a non neg- ative matrix and decompose it into two matrices. These decomposed matrix is approximately equivalent to the orig- inal matrix when multiplied together. Here the initial matrices are estimated by using randomized values. We also use frobenius fucntion as as the loss function to update these initial values. We use the beta divergence function as the optimization function. After several iterations we find the appropriate values.We use the two matrices as the ba- sis vectors for the data, then sorted according to their corresponding weights, to represent the contribution done by each subject per latent semantic. By calculating this we can understand the relationship between each of the subject ID(Each individual who volunteered for the hands dataset) and the basis vectors, hence gaining a better understand- ing of the composition of the image dataset.Task 8

The purpose of this task is to create a binary image meta data matrix and perform Non Negative Matrix Factorization to report the image and metadata latent semantics from the resulting matrices. The binary image meta data matrix is constructed such that each row represents a particular image and each column is a binary value representing if
it belongs to a certain label, the columns are constructed in the order Left, Right, Dorsal, Palmar, Acessories, No Accessories, Male and Female. The meta data columns here act as the new features for each of the images. The new image data matrix that has been constructed is then decomposed using the Non Matrix Factorization method as has been done in the previous task.

The image visualizer is used to display a ranked list of the images based on the weights in the eigen vectors for both the image space and the feature space.



3 Interface Specification

The interface we will be using for our tool is the python console. In this console we will take the input for each of the task accordingly. Here the data is loaded and displayed in the console. The images are displayed using mathpoltlib library functions. We use various functions such as gridspec and pyplots.to create a grid of images. The data can be further checked or verified using the pdAdmin tool that uses a web interface to view and edit data.

4 Code Implementation and Structure

In this phase to implement the given tasks, we have implemented a class based structure as follows. We have a Post- gresDB.py where we create the connection object, which is inherited by both imageProcess.py and DimReductoin.py. Also the imageProcess.py is inherited by DimReduction.py. As the names suggest, imageProcess.py contains all the methods to calculate the image features, such as color moments, SIFT, HOG or LBP. It also contains various other methods such as similarity calculations, distance calculations, database fetch, database save, image plotting and other helper functions. In DimReduction, we have all the methods to reduce the number of dimensions of the features cal- culated in imageProcess.py. It has various methods such as SVD, PCA, NMF or LDA. It has other methods needed to perform other tasks from phase 2.

Other than these class files, we have task files ranging from task 1 to task 8, which is the console file which imple- ments each of the task.

Also we have the prerequisites of feature data database and tables creation from phase 1. To implement this,we have Features.sql and runfile.py to create all the prerequisites.

Figure 5: Project Structure Flow-Chart

5 System Requirements and Installations

These are the steps needed to install and prepare the environment as to run the project with the explained interface.

5.1 System Specifications

1. Python Version: 3.6.5
2. Postgres DB Version: PostgreSQL 11.5
3. Our testing environment: 64-bit Operating System Windows 10.
4. For viewing data in the database, we can either use psql console or PgAdmin web interface.

5.2 Python Packages needed for running the scripts

1. Pillow v6.1.0
2. configparser v3.8.1
3. cycler v0.10.0
4. glob2 v0.7
5. kiwisolver v1.1.0
6. matplotlib v3.1.1
7. numpy v1.17.2
8. opencv-contrib-python v3.4.2.16
9. opencv-python

v3.4.2.16 10. pip v19.2.3

11. progressbar v2.5
12. psycopg2 v2.8.3
13. pyparsing v2.4.2
14. python-dateutil v2.8.0
15. scipy v1.3.1
16. setuptools

v39.1.0 17. six

v1.12.0

18. tqdm v4.35.0

5.3 Steps to Implement

The steps involved in running the project are as follows:

1. After installing the necessary modules for Python and setting up the environment for Postgres the necessary initial tables must be created. Run the feature_table.sql file to create the initial tables in Postgres.
2. To populate each of the initial feature tables, a script must be run, for this purpose the runFile.py must be run with the appropriate values for each of the features.
3. After the initial feature tables are created, create a 'Models' and 'Output' folder in the same directory as python files. These folders will contain the outputs from each of the tasks, the model folder will contain the K-means and dimensionality reduction models.
4. Each of the tasks are implemented in separate python files which can be run independent of each other, the python files take the inputs as provided in the specifications and provide results either in the form of

images in the output folder or in the console.

5. Functions from the class dimReduction can be called to delete multiple tables at the same time, and separate script files task1Auto.py and task3Auto.py have been written to perform task1 and task3 for all the Features and Dimensionality reduction techniques to assist with further tasks.
6. Make sure to add the image folder in the same path as the Python files and add the meta data csv file along with the said images so that meta data information can be loaded into the database as img_meta table.

6 Results

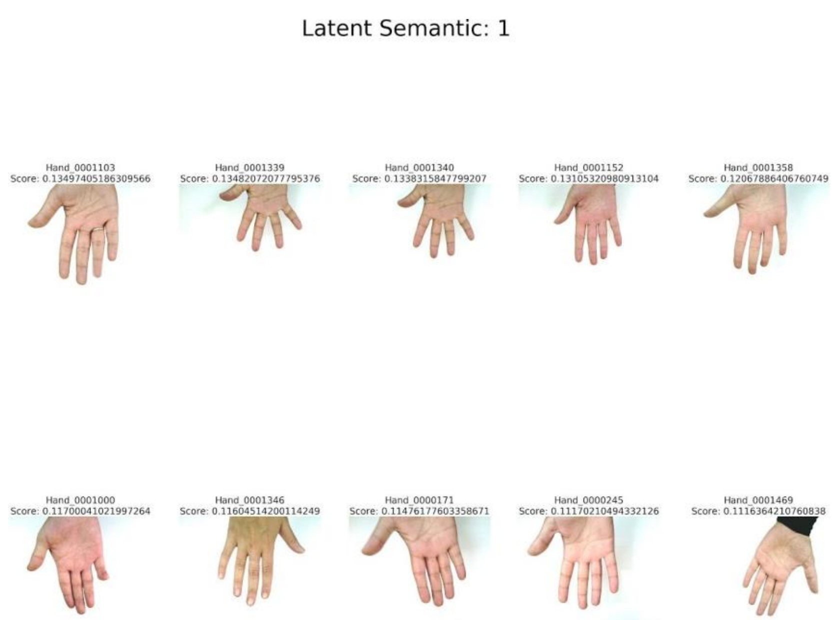
Task1

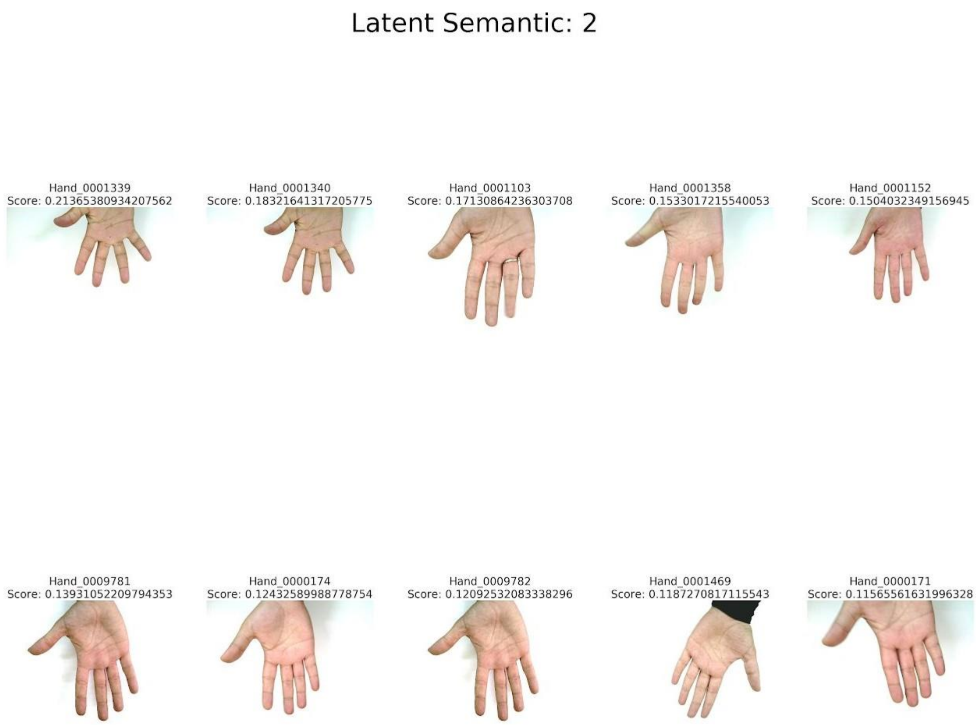
The requirements for task 1 are as follows:

	Model	k	Technique
Query 1	SIFT	20	Singular Value Decomposition(SVD)
Query 2	SIFT	40	Singular Value Decomposition(SVD)
Query 3	Color Moments	20	Latent Dirichlet Analysis (LDA)
Query 4	Color Moments	40	Latent Dirichlet Analysis (LDA)

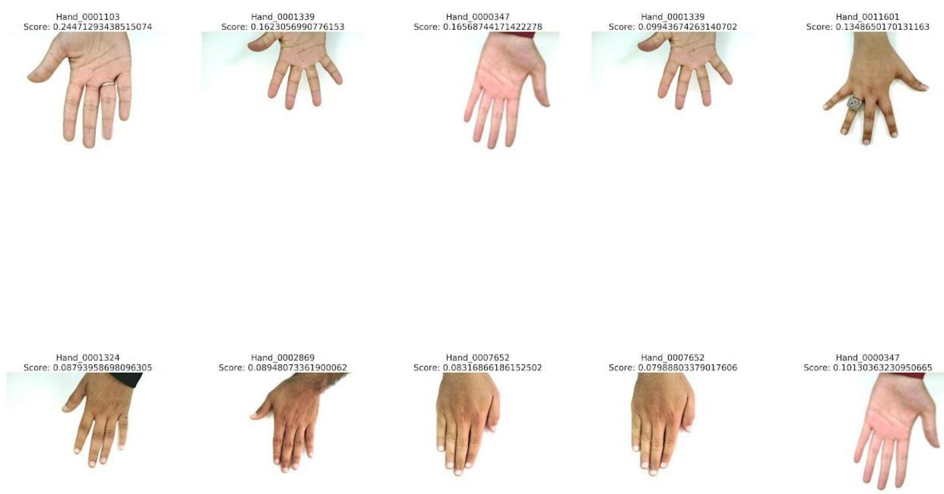
The data latent semantics and the feature latent semantics were calculated for each of the dimensionality reduction techniques. A visualizer was created to help understand the images that contributed the most to each of the latent semantics, similarly a place holder image was generated for each of the features to visually represent what characteristic of the image most contributed to each latent semantic.

Query 1: Model: Sift, K: 20, Technique: Singular value decomposition (SVD)

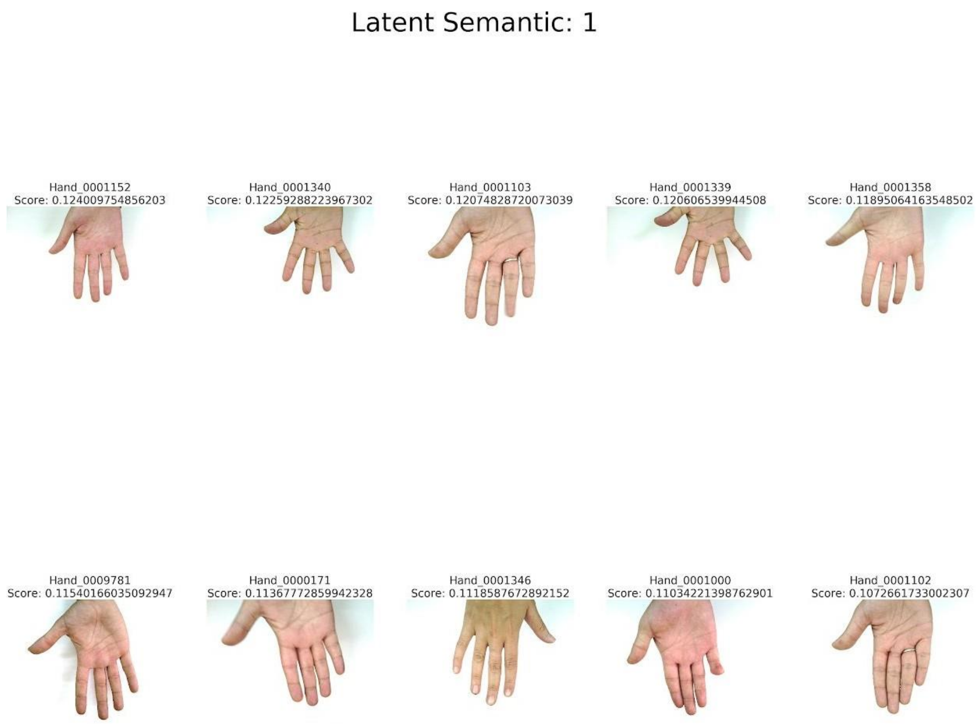


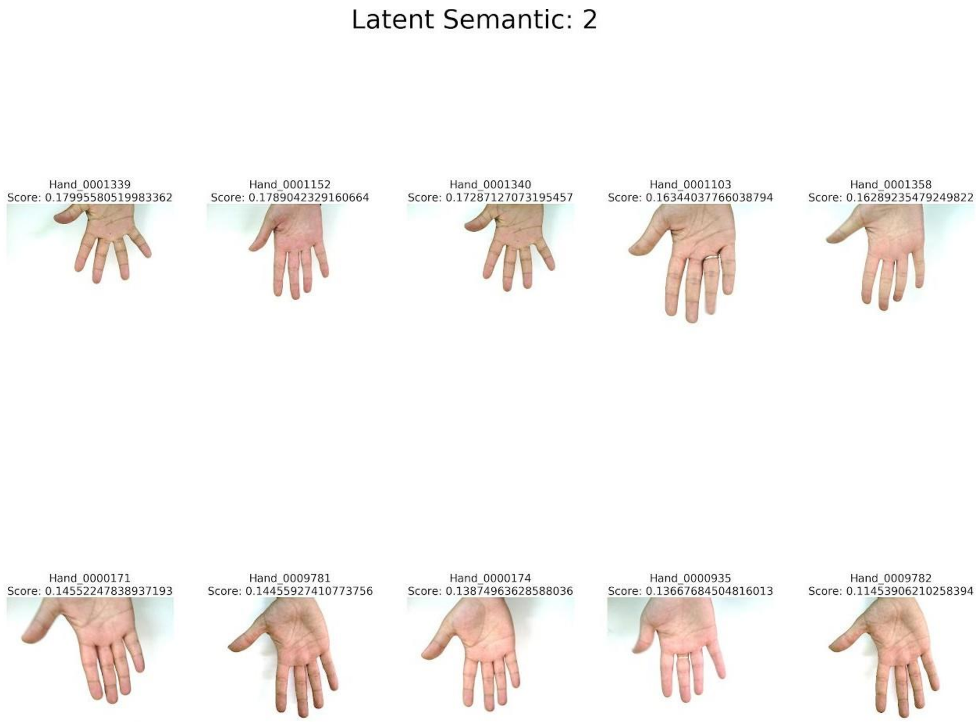


Feature Latent Semantic:



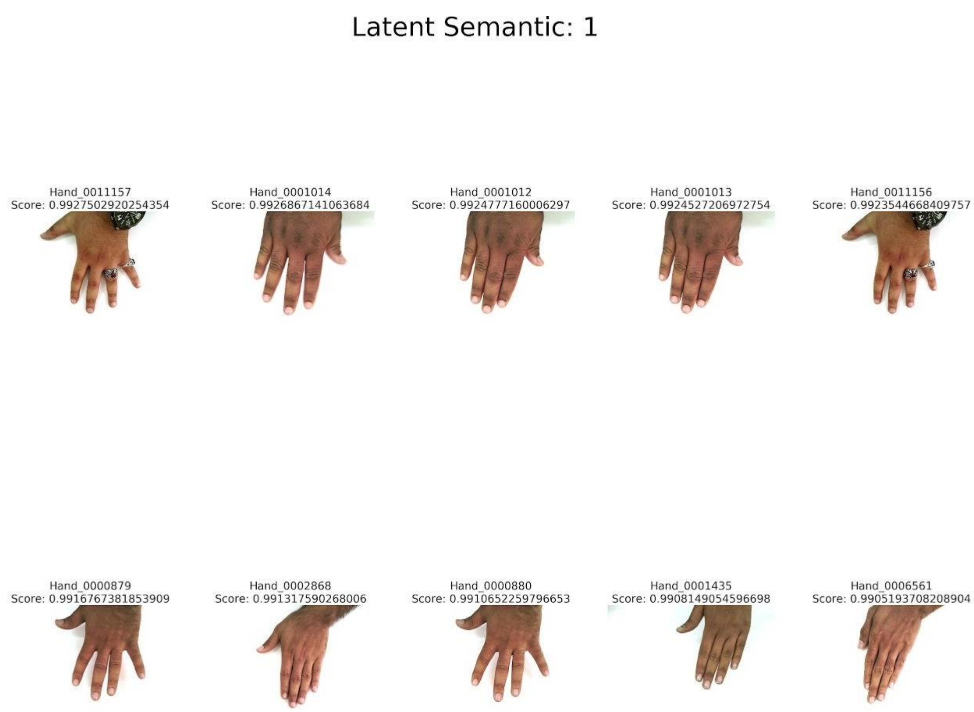
Query 2: Model : Sift,K : 40,Technique: Singular value decomposition (SVD)



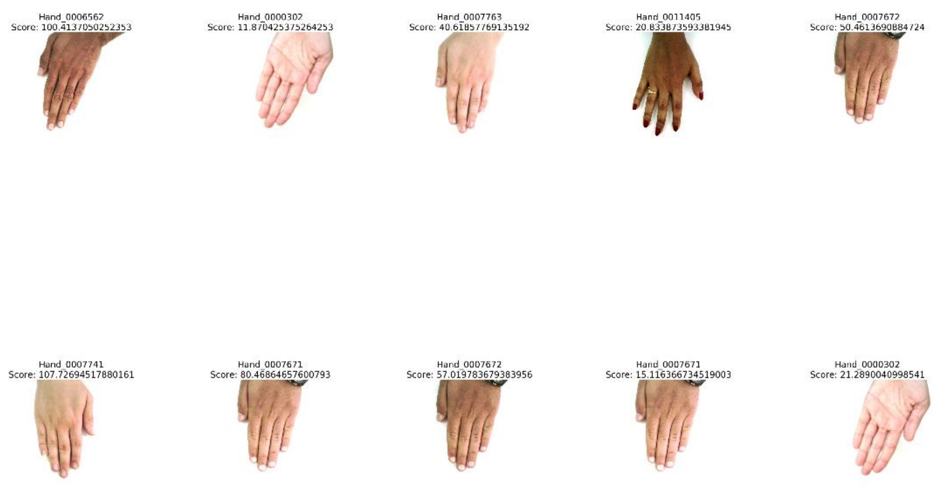


Feature Latent Semantic:

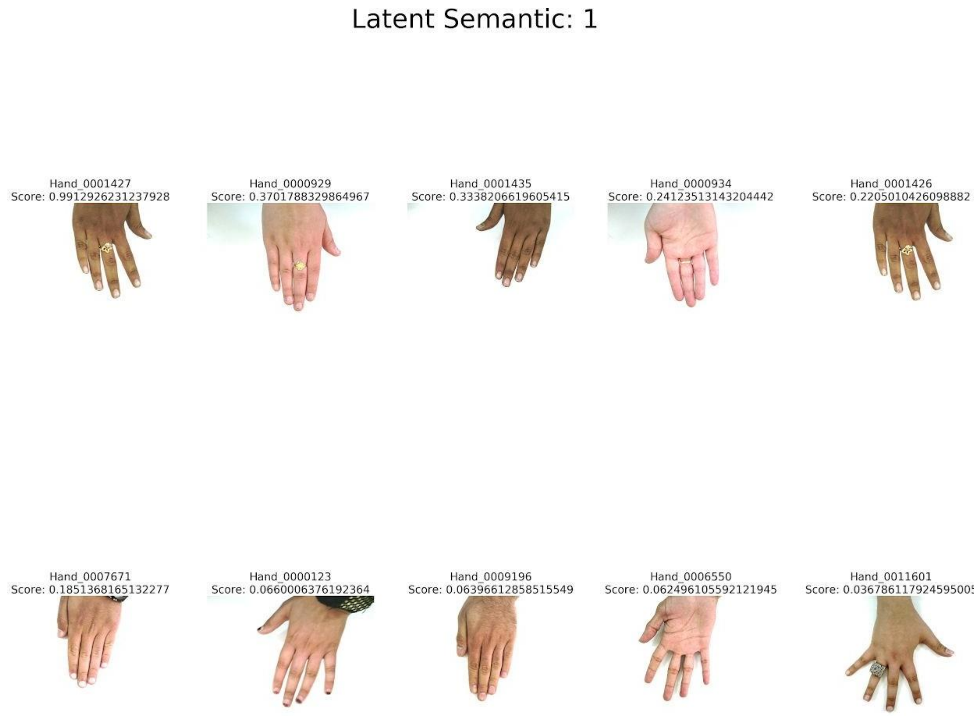
Query 3: Model : Color Moments,K : 20, Technique: Latent dirichlet analysis (LDA).

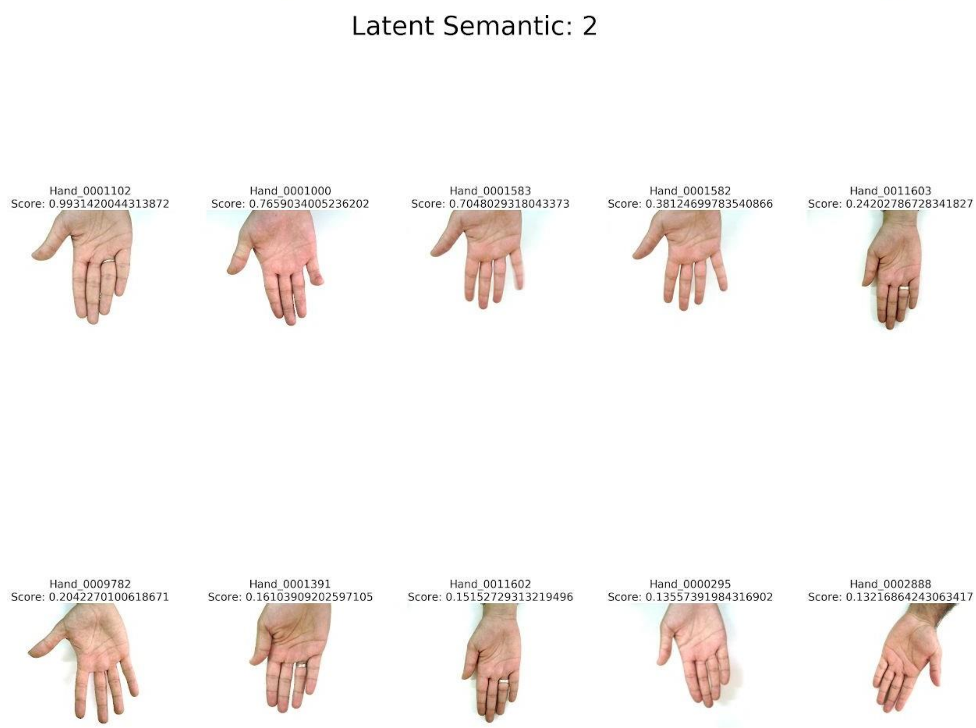


Feature Latent Semantic:

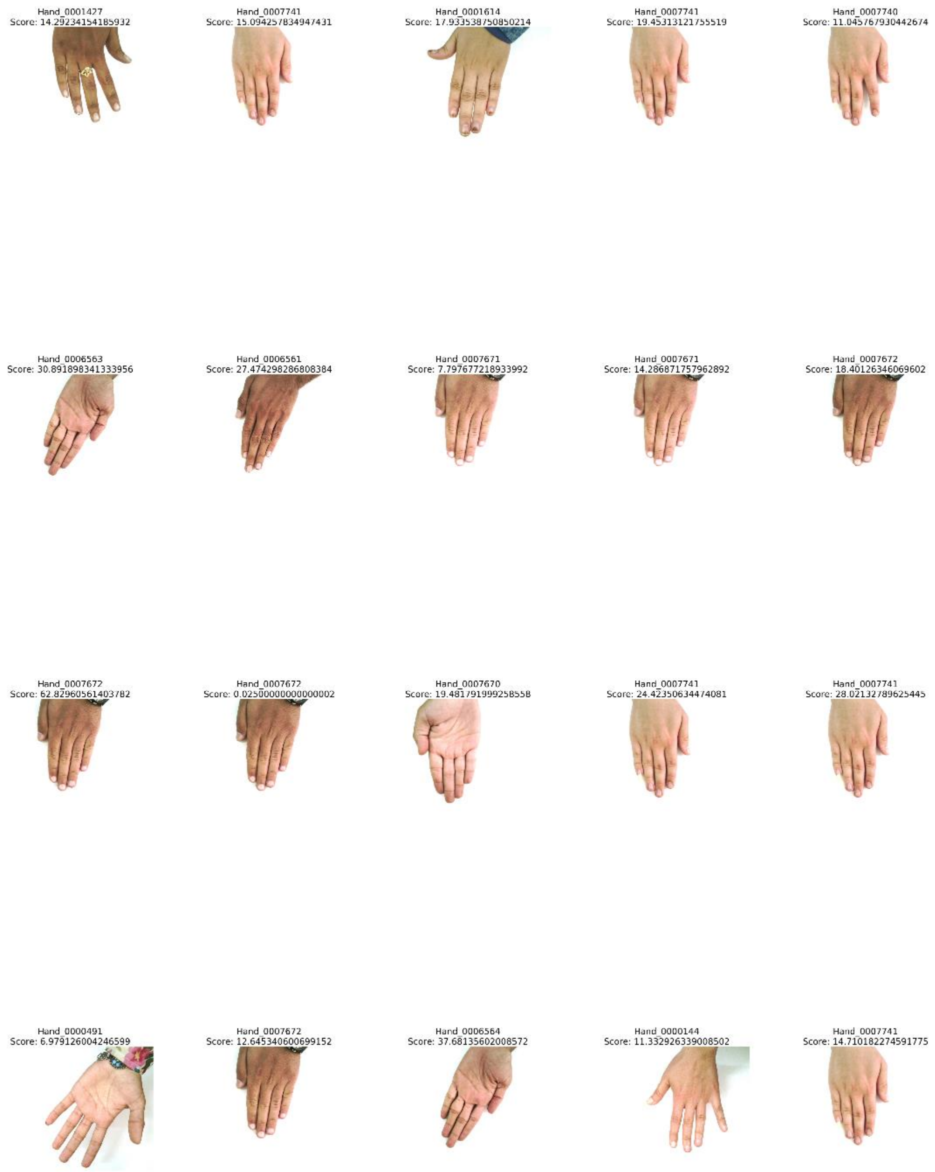


Query 4: Model: Color Moments, K: 20, Technique: Latent dirichlet analysis (LDA).





Feature Latent Semantic:



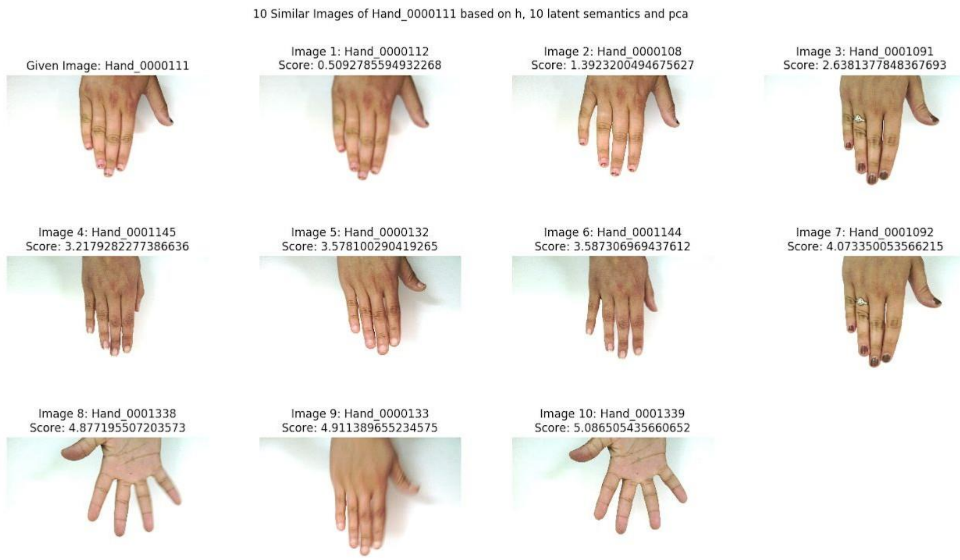
Task 2

The most similar images for a given image id were displayed in this task. The results that were gathered were very good since the normal L2 distance metric was providing good results for images that were reduced by k that is greater than 10.

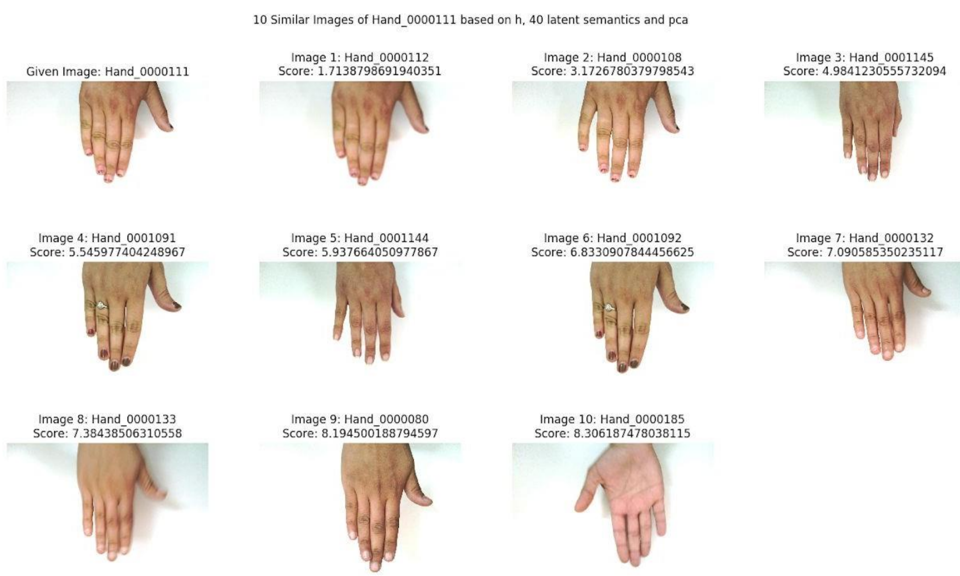
The requirements of task 2 are as follows:

	Model	K	M	Image ID	Technique
Query 1	HOG	10	10	Hand_0000111.jpg	Principal Component Analysis (PCA)
Query 2	HOG	40	10	Hand_0000111.jpg	Principal Component Analysis (PCA)
Query 3	LBP	10	10	Hand_0000200.jpg	Non-negative matrix factorization (NMF)
Query 4	LBP	40	10	Hand_0000200.jpg	Non-negative matrix factorization (NMF)

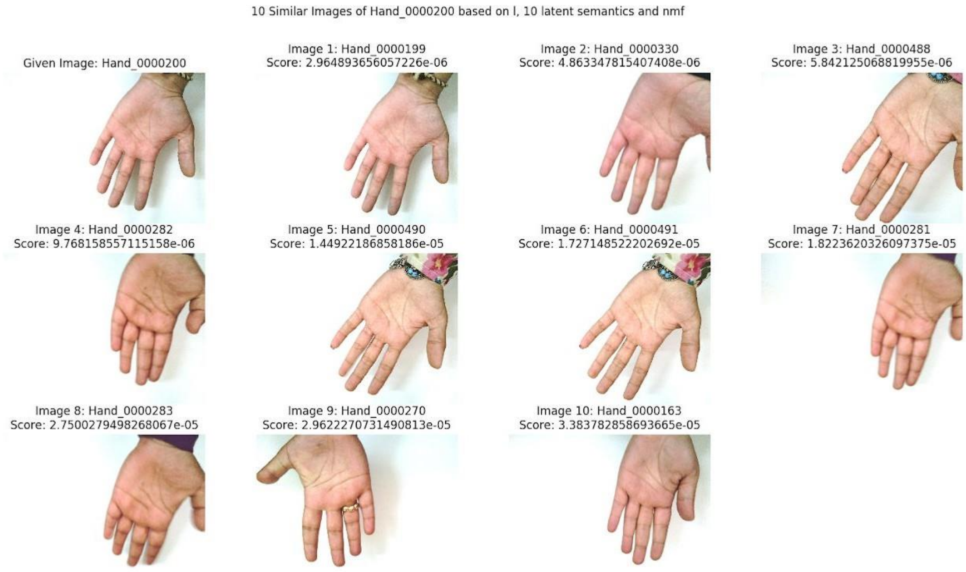
Query 1: 10 similar images to Hand_0000111.jpg using Hog and PCA(K=10)



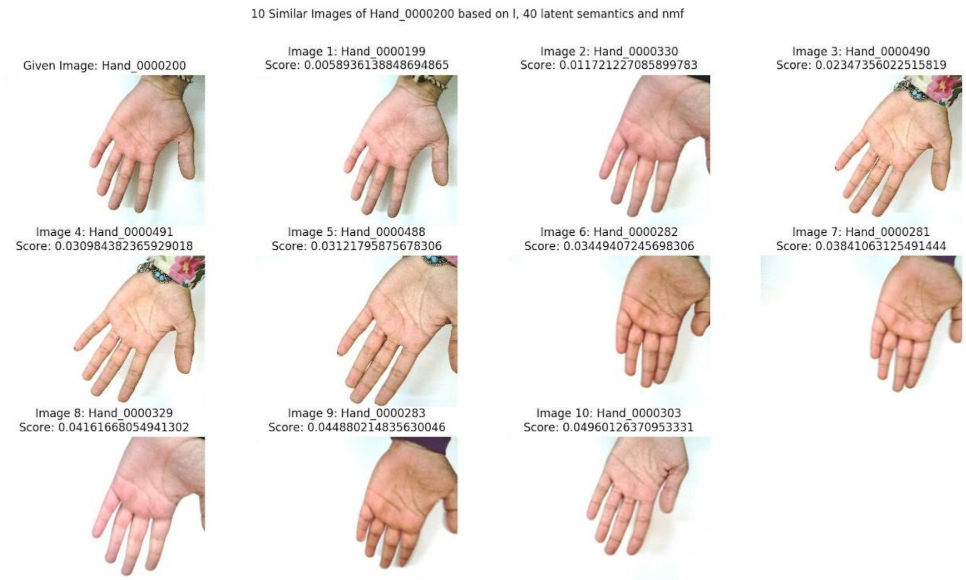
Query 2: 10 similar images to Hand_0000111.jpg using Hog and PCA(K=40)



Query 3: 10 similar images to Hand_0000200.jpg using Lbp and NMF(K=10)



Query 4: 10 similar images to Hand_0000200.jpg using Lbp and NMF(K=40)



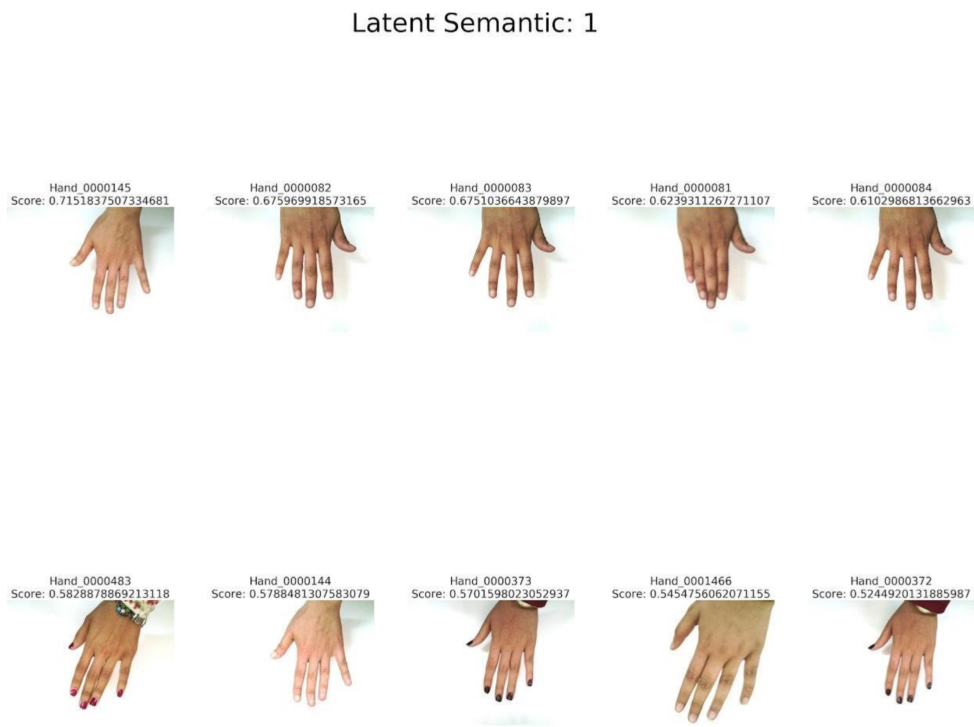
Task 3

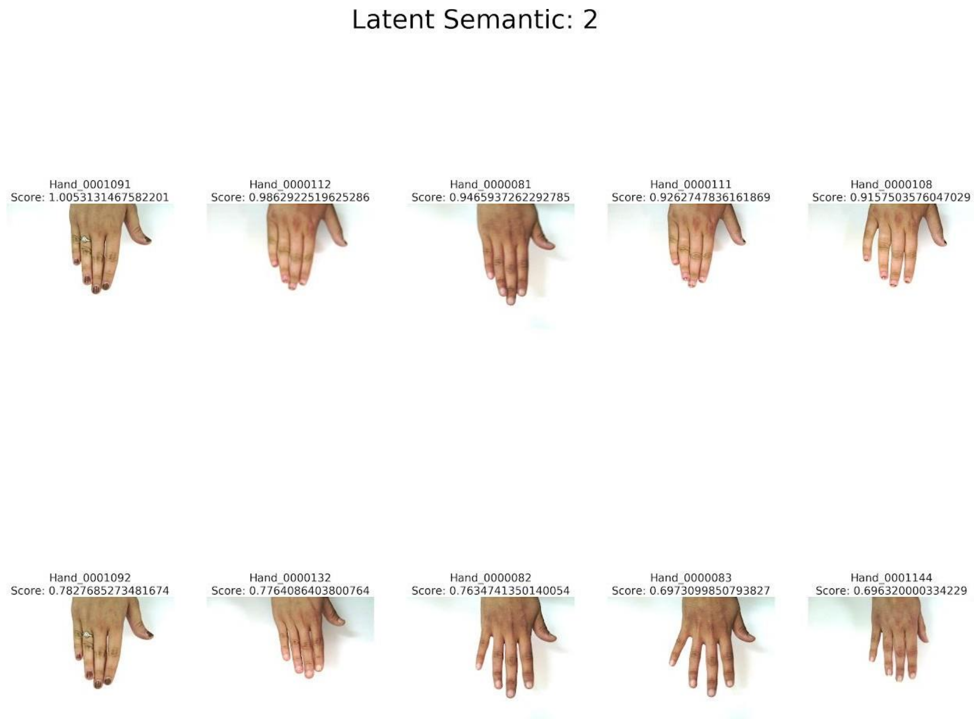
We implemented task 3 to identifies (and lists) k latent semantics for images with the corresponding label from metadata. We save the data latent semantics to database. Visualization of data latent semantics and feature latent semantices are demonstrated in order to evaluate the contribution of each image or each feature to every latent semantics

The requirements for task 3 are as follows:

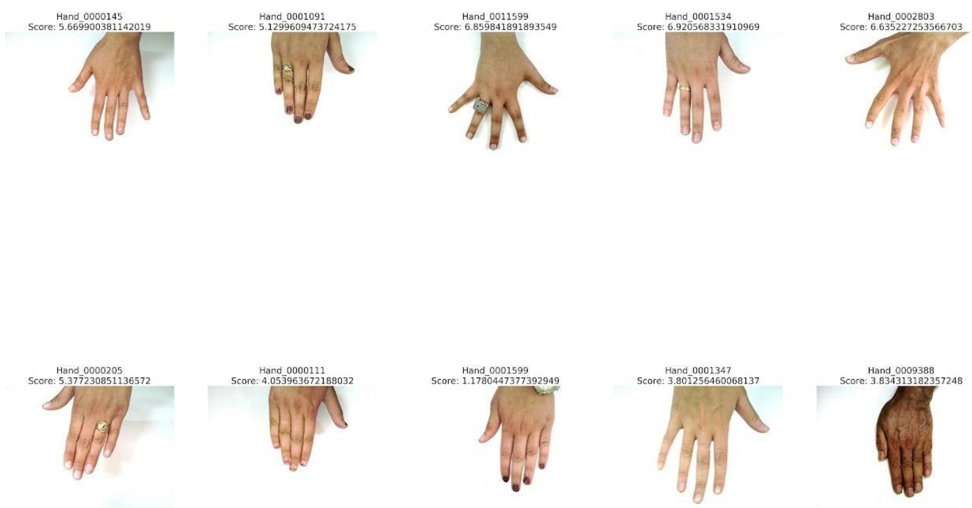
	Model	K	Label	Technique
Query 1	HOG	20	Dorsal	Principal Component Analysis(PCA)
Query 2	HOG	30	Dorsal	Principal Component Analysis(PCA)
Query 3	Color Moments	20	Left	Latent dirichlet analysis (LDA)
Query 4	Color Moments	30	Left	Latent dirichlet analysis (LDA)

Query 1: Model: HOG, K: 20, Label: Dorsal, Technique: Principal component analysis (PCA)
Data Latent feature:

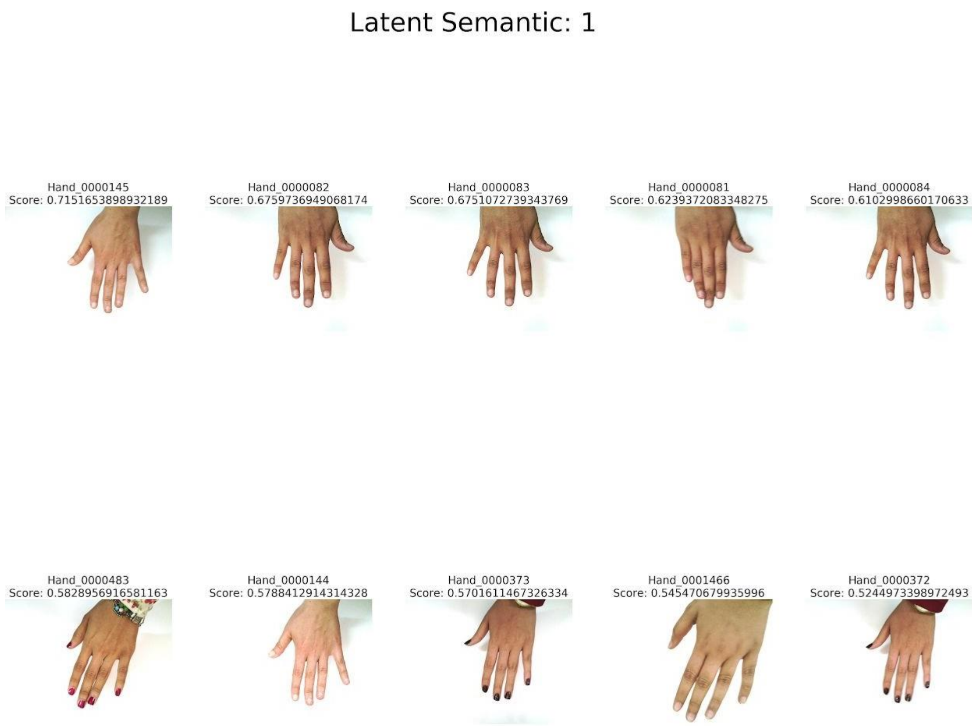


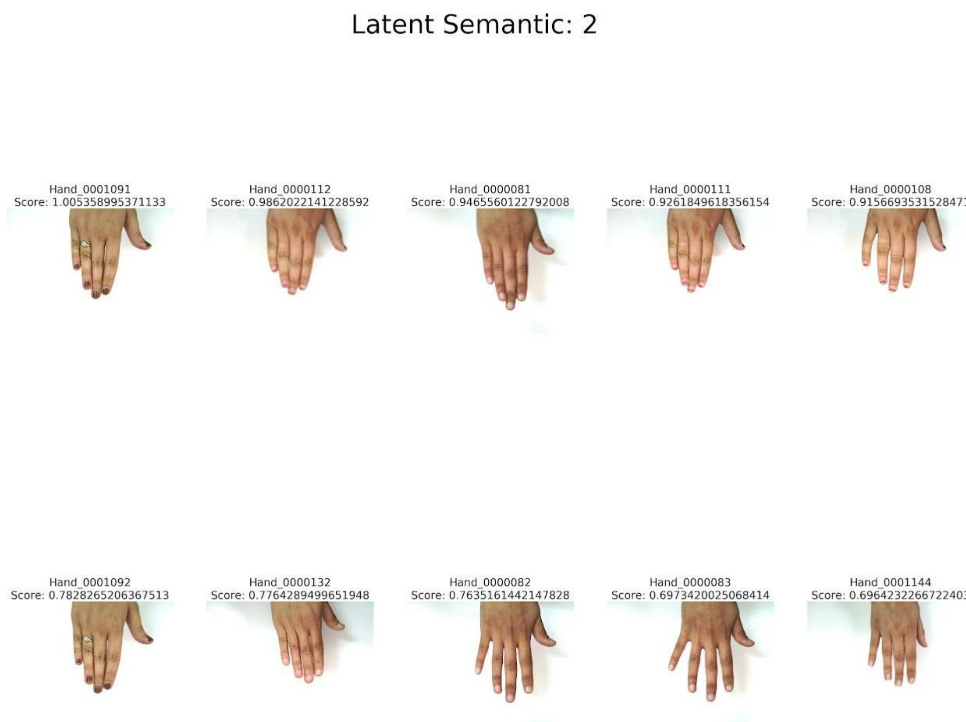


Feature Latent Semantics:

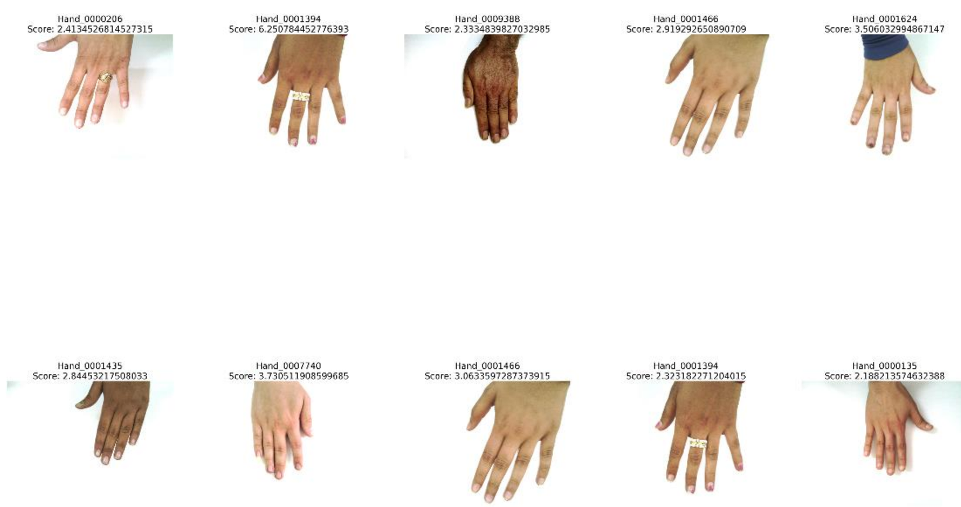


Query 2: Model: HOG, K: 30, Label: Dorsal, Technique: Principal component analysis (PCA)





Feature Latent Semantics:



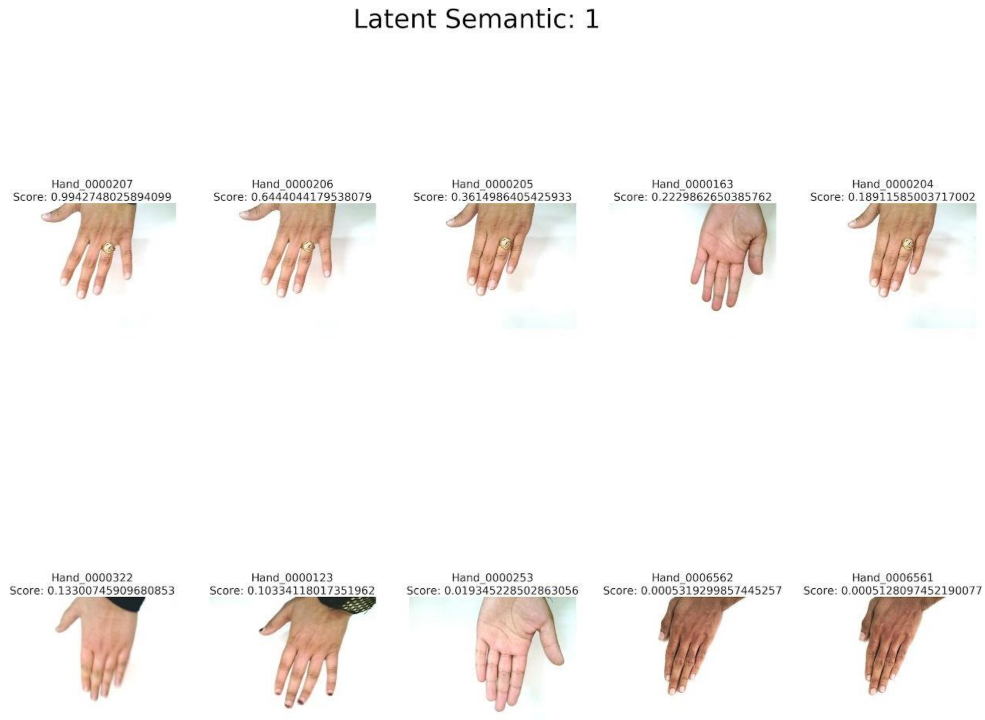

Hand_0000145
Score: 5.669799084244896
Hand_0001091
Score: 5.129522192078981
Hand_0011599
Score: 6.859634721321475
Hand_0001534
Score: 6.920552224070315
Hand_0002803
Score: 6.636219106377435

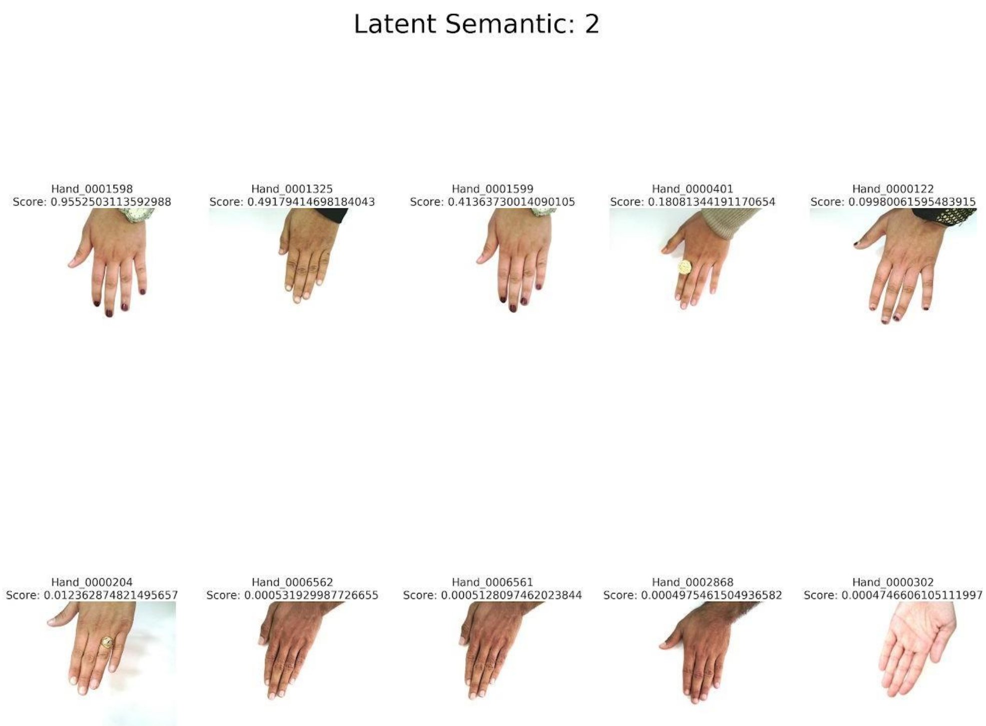

Hand_0000205
Score: 5.375771188042857
Hand_0000111
Score: 4.0531332211655675
Hand_0001599
Score: 1.1822649585151113
Hand_0001347
Score: 3.8134371603777817
Hand_0009388
Score: 3.8334700375680826


Hand_0011104
Score: 2.987086021104071
Hand_0009389
Score: 3.90708993369575
Hand_0000145
Score: 3.596649320912568
Hand_0001534
Score: 2.551651370961216
Hand_0000880
Score: 4.443573688375878


Hand_0011103
Score: 3.7598760507934665
Hand_0000931
Score: 3.647106817846048
Hand_0000483
Score: 2.002339213114695
Hand_0000373
Score: 4.275500159468105
Hand_0000145
Score: 3.3465646090095493

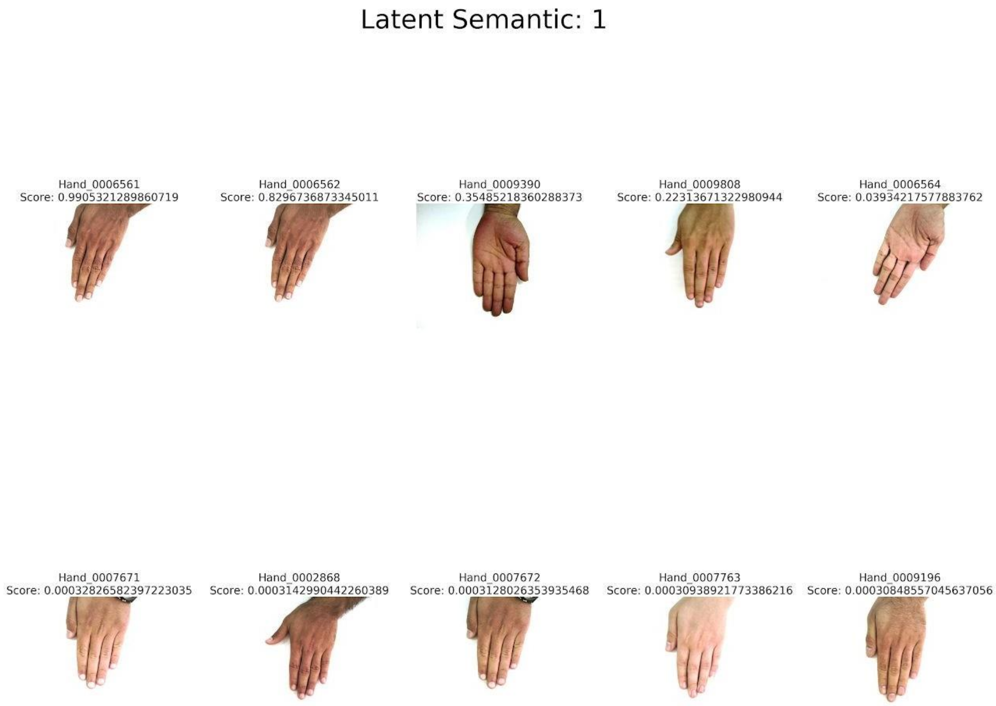
Query 3: Model: Color Moments, K: 20,Label : Left, Technique: Latent dirichlet analysis (LDA).

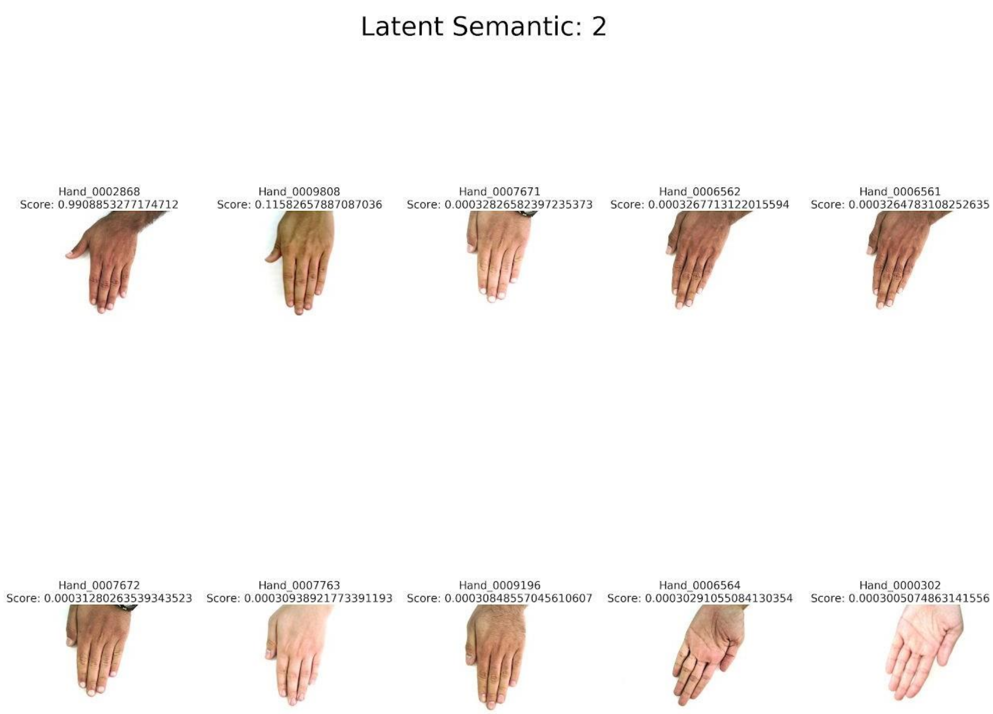




Feature Latent Semantic:

Query 4: Model: Color Moments, K: 30, Label: Left, Technique: Latent dirichlet analysis (LDA).





Feature Latent Semantic:


Hand_0007671
Score: 17.14660029240517
Hand_0007763
Score: 20.27644830053807
Hand_0006564
Score: 3.593803945459489
Hand_0007672
Score: 10.433136924006849
Hand_0001614
Score: 10.769052685483905
Hand_0000199
Score: 8.62517397779247
Hand_0000302
Score: 30.904192111870046
Hand_0009389
Score: 9.709614581377881
Hand_0006563
Score: 10.826952016360739
Hand_0000347
Score: 5.529695873989798

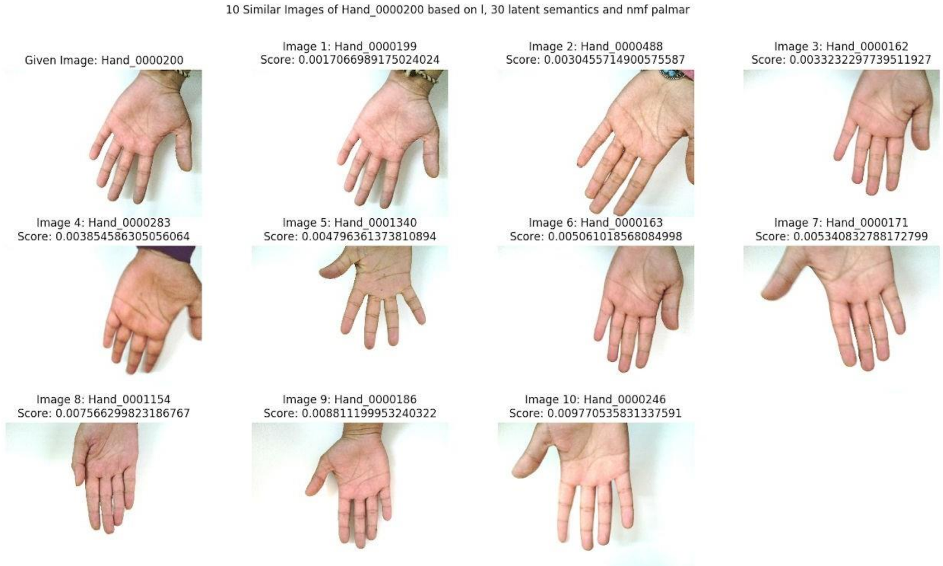
Task 4

We implemented task 3 to identifies (and lists) k latent semantics for images with the corresponding label from metadata. We save the data latent semantics to database. Visualization of data latent semantics and feature latent semantices are demonstrated in order to evaluate the contribution of each image or each feature to every latent semantics. The requirements of task 4 are as follows in the table:

	Model	K	M	Label	Technique	Image ID
Query 1	LBP	10	10	Palmer	Non-negative matrix factorization (NMF)	Hand_0000200.jpg
Query 2	LBP	30	10	Palmer	Non-negative matrix factorization (NMF)	Hand_0000200.jpg
Query 3	SIFT	10	10	With Accessories	Singular value decomposition (SVD)	Hand_0011160.jpg
Query 4	SIFT	30	10	With Accessories	Singular value decomposition (SVD)	Hand_0011160.jpg

Query 1: Model: LBP, K: 10, M: 10, Label: Palmer, Technique: Non-negative matrix factorization (NMF), Image ID: Hand_0000200.jpg

Model: LBP, K: 30, M: 10, Label: Palmer, Technique: Non-negative matrix factorization (NMF), Image ID: Hand_0000200.jpg


10 Similar Images of Hand_0000200 based on l, 30 latent semantics and nmf palmar
Given Image: Hand_0000200
Image 1: Hand_0000199
Score: 0.0017066989175024024
Image 2: Hand_0000488
Score: 0.0030455714900575587
Image 3: Hand_0000162
Score: 0.0033232297739511927
Image 4: Hand_0000283
Score: 0.0038545863050560064
Image 5: Hand_0001340
Score: 0.004796361373810894
Image 6: Hand_0000163
Score: 0.005061018568084998
Image 7: Hand_0000171
Score: 0.005340832788172799
Image 8: Hand_0001154
Score: 0.007566299823186767
Image 9: Hand_0000186
Score: 0.008811199953240322
Image 10: Hand_0000246
Score: 0.009770535831337591

Query 3: Model: Sift, K: 10, M: 10, Label: With Accessories, Technique: Singular value decomposition (SVD), Image ID: Hand_0011160.jpg

Query 4: Model: Sift, K: 30, M: 10, Label: With Accessories, Technique: Singular value decomposition (SVD), Image ID: Hand_0011160.jpg

Task 5

One class SVM was used to perform the classification between the images, this was taken because the overall accuracy percentages for most of the feature and dimensionality reduction techniques was providing good results over the centroid method. Since the one class SVM creates a decision boundary for one class classification based on euclidean distances, the results for SVM are relatively poor because the calculated distances are of a very low degree because of the pre-processing technique chosen for it. The K-means algorithm creates many centroids and then histogram is made based on descriptor vectors which will in turn result in very low values after dimensionality reduction by any technique. SIFT performs well for similarity calculation because the images are being compared with a small pool of images which vary greatly once the labels are changed. The value of K was increased iteratively for Task 5 to see if there would be any objective difference in the accuracies, but it was noticed that the accuracy remained either constant or deteriorated in the case of all the models. This can be especially noticed in Query 2 for task 5 where the result deteriorates as the value for k increases.

The requirements of task 5 are as follows in the table:

	Model	**K**	**Label**	**Technique**	**Image ID**
Query 1	LBP	10	Right	Non-negative matrix factorization (NMF)	Hand_0000111.jpg
Query 2	LBP	30	Right	Non-negative matrix factorization (NMF)	Hand_0000111.jpg
Query 3	SIFT	10	With Accessories	Singular value decomposition (SVD)	Hand_0001395.jpg
Query 4	SIFT	30	With Accessories	Singular value decomposition (SVD)	Hand_0001395.jpg

The results for task 5 are as follows:

Query 1	Success
Query 2	Failure
Query 3	Failure
Query 4	Failure

The accuracies for each individual task:

	PCA	**SVD**	**NMF**	**LDA**
HoG	83.33	76.67	70.0	73.33
SIFT	50.0	33.33	50.0	50.0
Color Moments	63.33	63.33	50.0	53.33
LBP	86.67	83.33	63.33	50.0

Task 6

The requirements of task 6 are as follows in the table:

	Subject ID
Query 1	27
Query 2	56

The subject's similarities are being calculated by measuring the euclidean distance and converting it into a similarity metric. Each subject id is represented by its centroid, for ease of visualization an image from each of the subject ids has e chosen as the place holder and the similarity measures have been reported as part of the console output. The top three subject Ids for input subject ID has been provided, each subject IDs place holder image has been given in the results.

Query 1:

Subject ID	Weight
0000529	0.3777619331243391
0001506	0.35652637204926735
0001074	0.34985169468419325

Query 2:

Subject ID	Weight
0000505	0.3000122525253304
0001509	0.28512258735951235
0000081	0.284157479610815

Task 7

The subject - subject similarity matrix has been calculated for a given k and the data latent semantics have been displayed for it, this can be represented in the form of the subjec Ids and their corresponding weights associated with it.

Please refer to the attached document for the results.

Task 8

The binary image meta data similarity matrix has been calculated for a given k and the data latent semantics and the feature latent semantics have been displayed for it, this can be represented in the form of the subjec Ids and their correspoding weights associated with it. The overall image latent semantics have been displayed as images ranked on their contribution or weight. The console output has been generated for the feature latent semantics. The requirements for task 8 are as follows:

	K
Query 1	4
Query 2	6

Query 1: K:4

Rank	Label	Latent Semantic 1 weight
1	right	1.5272610063956966
2	palmar	1.3035456349921835
3	female	0.9782274185930494
4	no-accessories	0.925268456930576
5	accessories	0.06952034946310524
6	male	0.0
7	dorsal	0.0
8	left	0.0

Rank	Label	Latent Semantic 2 weight
1	accessories	1.2701683197181484
2	dorsal	1.1213124611874485
3	female	1.0798693068539345
4	right	0.51078019741546
5	left	0.2012602955194338
6	male	0.0
7	no-accessories	0.0
8	palmar	0.0

Table 1: Task 8 Query 1 Top 2 Latent semantics for each label in metadata space

Query 2: K:6

Rank	Label	Latent Semantic 1 weight
1	female	2.406413702649867
2	no-accessories	2.4045179932313197
3	male	0.0
4	accessories	0.0
5	palmar	0.0
6	dorsal	0.0
7	right	0.0
8	left	0.0

Rank	Label	Latent Semantic 1 weight
1	female	1.6873200555707626
2	no-accessories	1.4473171320273865
3	male	0.06965822976986508
4	accessories	0.0
5	palmar	0.0
6	dorsal	0.0
7	right	0.0
8	left	0.0

Table 2: Task 8 Query 2Top 2 Latent semantics for each label in metadata space

7 Conclusion

A lot of experimentation was done in the case of each of the tasks, the main learning from this phase came from the kind of outputs that are gathered from each of the dimensionality reduction models, the way that each of them utilize their corresponding latent semantics and the way they were leveraged to perform dimensionality reduction. One of the major learnings that were gleaned from this was working with various kinds of feature models which had varied structures and needed to be preprocessed before they were ingested by the dimensionality reduction algorithms. K-means was employed in this scenario to estimate the histograms so that most of the processing could be done. Various kind of experimentations were done in the case of task 5 as well for the classification task. Since both label data was not provided a different methodology had to be devised for classifying the image. After experimentation with a naive centroid approach and a outlier detection algorithm, one class SVM was preferred because of the reported accuracies. It was learnt after calculating the accuracies over all the combinations that the accuracies varied greatly upon the kind of feature model and the dimensionality reduction algorithm used. This phase involved the whole group doing holistic analysis on the various dimensionality reduction methods taught in class.

8 References

[1] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints In *International Journal of Computer Vision, 2004*

[2] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." 2005.

[3] Giuseppe Amato, Fabrizio Falchi, Claudio Gennaro Geometric consistency checks for kNN based image classification relying on local features. In *Fourth International Conference on Similarity Search and Applications, SISAP 2011, Lipari Island, Italy,* June 30 - July 01, 2011

[4] Gang Qian, Shamik Sural, Yuelong Gu, S. Pramanik Similarity between Euclidean and cosine angle distance for nearest neighbor queries In *Conference: Proceedings of the 2004 ACM Symposium on Applied Computing (SAC), Nicosia, Cyprus,* March 14-17, 2004

[5] Amato, Giuseppe, Fabrizio Falchi, and Claudio Gennaro. "Geometric consistency checks for kNN based image classification relying on local features." Proceedings of the Fourth International Conference on SImilarity Search and APplications. ACM, 2011.

[6] Latent Dirichlet Allocation David M. Blei, Andrew Y. Ng, Michael I. Jordan Journal of Machine Learning Research 3 (2003) 993-1022

[7] https://user.eng.umd.edu/ smiran/LDA.pdf Latent Dirichlet Allocation (LDA) for Topic Modeling

[8] https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

[9] https://en.wikipedia.org/wiki/Singular_value_decomposition

9 Contributions

Member	**Task**
Anh The Nguyen	Task 1, Task 2, Task 3, Task 4
Drithi Shah	Task 1, Task 2, Task 3, Task 4 and Documentation
Rinku Nemade	Task 1, Task 2, Task 3, Task 4
Kovidnath Pyla Reddy	Task 5, Task 6, Task 7, Task 8
Vishal Tyagi	Task 5, Task 6, Task 7, Task 8
Viralam Sreedar Tharun Kumar	Task 5, Task 6, Task 7, Task 8 and Documentation