CSE 515: MULTIMEDIA AND WEB DATABASES

Fall 2019 Phase 3

Group 12

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

Abstract

The third phase of the project focuses on interpreting and analyzing data that can support efficient indexing, clustering, classification and relevance feedback system of the data items. In this phase of the project, we are using Support Vector Machine (SVM), K Means Clustering, Decision Tree, Personalized Page Rank (PPR), Relevance Feedback System and Locality Sensitive Hashing. The various algorithms used are evaluated and visualized for each task. 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."*

Keywords

K-Means Clustering, Locality Sensitive Hashing (LSH), Personalized Page Ranking (PPR), Support Vector Machine (SVM), Decision Tree, Gini Index, Relevance Feedback System, Latent Semantics, Euclidean Distance, Dimensionality Reduction, Color Moments, Histogram of Oriented Gradients, Scale Invariant Feature Transform, Local Binary Patterns, Non-negative Vector Similarity Coefficient based Distance

1 Introduction

Multimedia and web databases deal with the storage of multimedia data points. Different features of the multimedia data points are extracted using various visual models like Color Moments (CM), Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT) and Local Binary Patterns (LBP).
To handle large amounts of data, efficient algorithms for clustering the data points, indexing the data and retrieving it for the end-users is a must. For quick and efficient retrieval as well as for the purpose of clustering the dataset, the dimensionality of the data points present in the database is reduced using dimension reduction techniques like SVD, PCA, NMF, and LDA.

In this phase, we implement various classifiers like SVM classifier, decision tree classifier, and PPR based classifier. We experimented with personalized page ranking for visualizing the k most dominant images. Also in this phase, we experimented with Locality Sensitive Hashing (LSH) tool for creating an in-memory index structure containing the given set of vectors.

This task of representing this heterogeneous data allows us to query and retrieve multimedia data. For this various distance and similarity measures are used such as euclidean and Non-negative vector similarity coefficient-based distance (NVSC).

1.1 Terminology

- *PPR*: Personalized Page Ranking (PPR)
- *SVM*: Support Vector Machine, a supervised classifier for classification.
- *LSH*: Locality Sensitive Hashing, Duplicate detection technique used for nearest neighbor search and data clustering.
- *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 measure 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. 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.

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

- *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 local spatial patterns.

- *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.
- *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.
- *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.
- *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 local spatial patterns.
- *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 returns 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.

1.2 Task Description

1. **Task 1**: Implement a program which, given a folder with dorsal/palmar labeled images, computes k latent semantics (in the feature space) associated with dorsal-hand images, computes k latent semantics (in the feature space) associated with palmar-hand images, and, given a folder with unlabeled images, the system labels them as dorsal-hand vs palmar-hand using only these latent semantics.

2. **Task 2**: Implement a program which, given a folder with dorsal/palmar labeled images and for a user-supplied c,
    1. computes c clusters associated with dorsal-hand images (visualize the clusters)
    2. computes c clusters associated with palmar-hand images (visualize the resulting image clusters).
    3. given a folder with unlabeled images, the system labels them as – dorsal-hand vs palmar-hand using only descriptors of these clusters.

3. **Task 3**: Implement a program which, given a value k, creates an image-image similarity graph, such that from each image, there are k outgoing edges to k most similar/related images to it. Given 3 user-specified image ids on the graph, the program identifies and visualizes ‘K’ most dominant images using Personalized Page Rank (PPR) for a user-supplied K.

4. **Task 4**: Implement a program which, given a folder with dorsal/palmar labeled images,

   1. Creates an SVM classifier
   2. Creates a decision-tree classifier
   3. Creates a PPR based classifier

   and, given a folder with unlabeled images, the system labels them as
   – dorsal-hand vs palmar-hand using the classifier selected by the user

5. **Task 5**:
   A: Implement a Locality Sensitive Hashing (LSH) tool (for Euclidean distance) which takes as input (a) the number of layers, L, (b) the number of hashes per layer, k, and (c) a set of vectors as input and creates an in-memory index structure containing the given set of vectors.
   B: Implement a similar image search algorithm using this index structure and a visual model function of your choice (the combined visual model must have at least 256 dimensions): for a given query image and integer t, visualizes the t most similar images (also outputs the numbers of unique and overall number of images considered).

6. **Task 6**: Let us consider the label set "Relevant (R)" and "Irrelevant (I)". Implement
   1. An SVM based relevance feedback system
   2. A decision-tree based relevance feedback system
   3. A PPR-based relevance feedback system
   4. A probabilistic relevance feedback system

   which enable the user to label some of the results returned by 5b as relevant or irrelevant and then return a new set of ranked results, relying on the feedback system selected by the user, either by revising the query or by re-ordering the existing results.

1.3 Assumptions

- Images retrieved as similar images can be similar in terms of features.
- It is assumed that the query images are of a similar kind as the training images.
- For tasks related to color moments, more weight has been given to the Y component of YUV than U and V. This assumption is made while calculating the weighted distance matrix.
- 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).
- All the images in the dataset are in jpg format.
- For task 6, we are expecting to have at least one irrelevant and relevant feedback as the classification is binary.

2 Proposed Solutions

The following algorithms have been used to implement various tasks in the final phase of the project.

2.1 Personalized Page Rank

Page rank is an algorithm to rank the pages as per the priority. In page ranking, the images are saved as a node. Each image has a score given to it. PPR is a variation of page rank. Instead of a uniform distribution of page score, it will be biased towards seed pages. The page rank score calculated in one iteration will be used to calculate the scores of the next iterations. In search engines, the page rank assumes the fact that the more the number of links a page has from other pages, the more important the page is [1].

$$P(j) = \frac{1-\alpha}{|seed\ nodes|} + \alpha\Sigma \frac{P'(i)}{|out(j)|}$$

For non-seed nodes, the page rank is calculated as,

$$P(J) = \alpha \sum \frac{P'}{|out(J)|}$$

P(J) - The page rank score assigned to a given node, J.
P'(i) - The page score assigned to node j.
α- The damping factor.

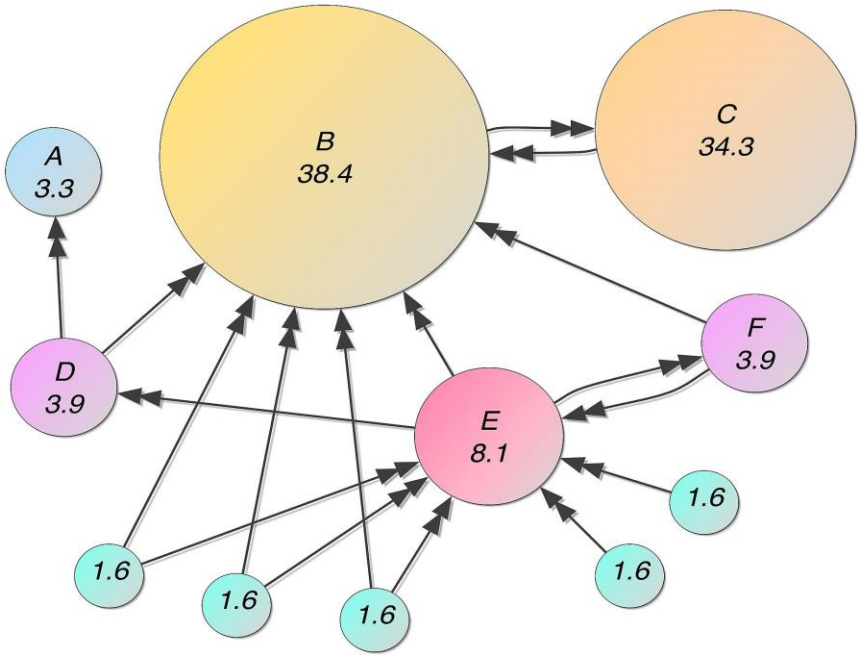


Fig: Schematic Diagram for Page Rank
Source: *https://en.wikipedia.org/wiki/PageRank*

$$P(J) = \alpha \sum \quad |P(i) - P(i)'|$$

If there are n seed pages then the value $|seed\ values|$ are n. The seed values are started with a value of 1/n. The seed page has more importance than the other node, the initial page rank vector has all values initialized to 0.
Page rank scores are calculated iteratively until the stopping condition is reached.

2.2 K-Means Clustering

K-Means clustering is a clustering method that is used for cluster analysis in the field of data mining. K-means clustering partitions the given data sets into various clusters and each dataset belongs to a cluster. The algorithm picks k random nodes from the given data set as centroids for the k clusters. The data sets are reassigned to the closest cluster mean, and the clusters are recalculated.

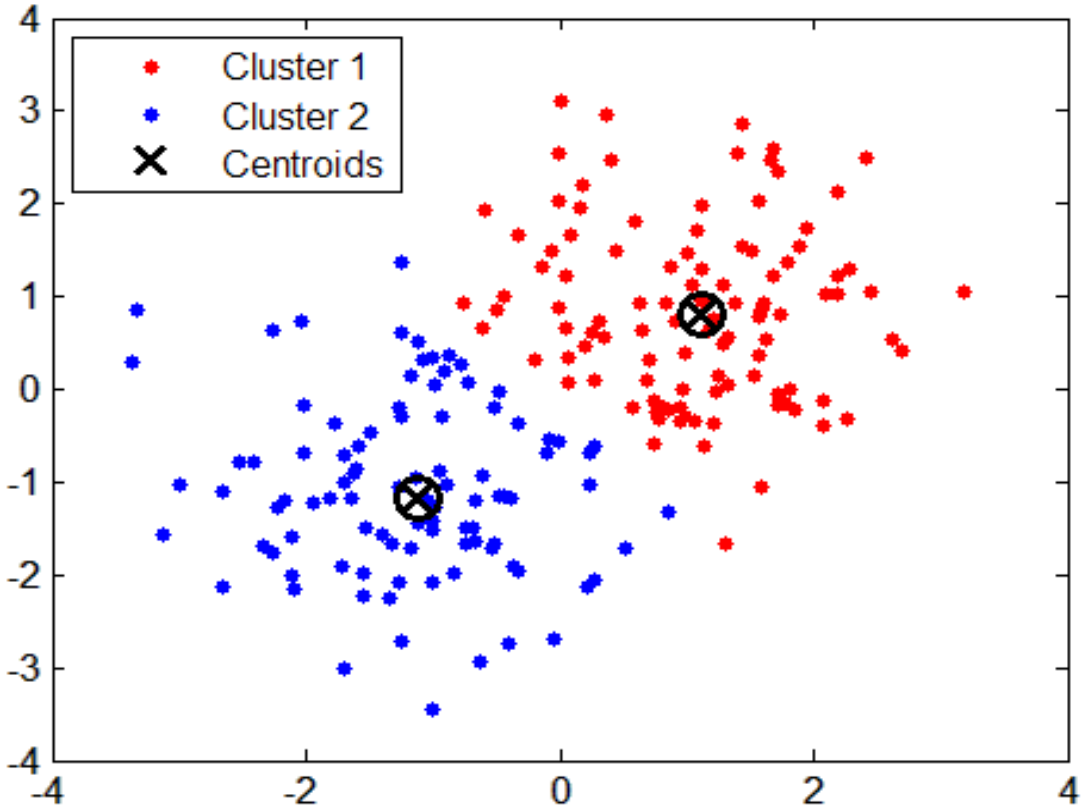


Fig: K-Means clustering, cluster formation for the given data sets.

Source: *https://www.google.com/search?q=K-means+clustering&sxsrf=ACYBGNQUoVFkt_rLGuM5k-7vmo98rfcgDw:1575333525163&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjM463ZnpjmAhUyJzQIHR7HC-AQ_AUoAnoECBAQBA&biw=1440&bih=789#imgrc=gDMmT9s4gun-rM:*

The entire process is repeated until it is converged. Convergence happens in two ways. Either when the centroids do not change between iterations or the number of iterations has been fixed beforehand.

2.3 Locality Sensitive Hashing

The Locality Sensitive Hashing is an algorithmic technique that hashes similar input items into the same buckets, based on if the two data points have the same fingerprints. Every data point in the database will be assigned a hash code/ fingerprint, based on the hash function used. The data objects with the same fingerprint or hash code are mapped onto the same hash bucket. Since similar objects come in the same bucket, the technique is usually used for data clustering and nearest neighbor search [2]. In our project, the hash function we are using is:

$$\text{Hash function} = \frac{a.v + b}{w}$$

The problem with LSH is that it can lead to false positives and misses. Two different data points that are not near to each other can be mapped into the same hash bucket even if their hash values are different. For that, one way is to improve that hash function.

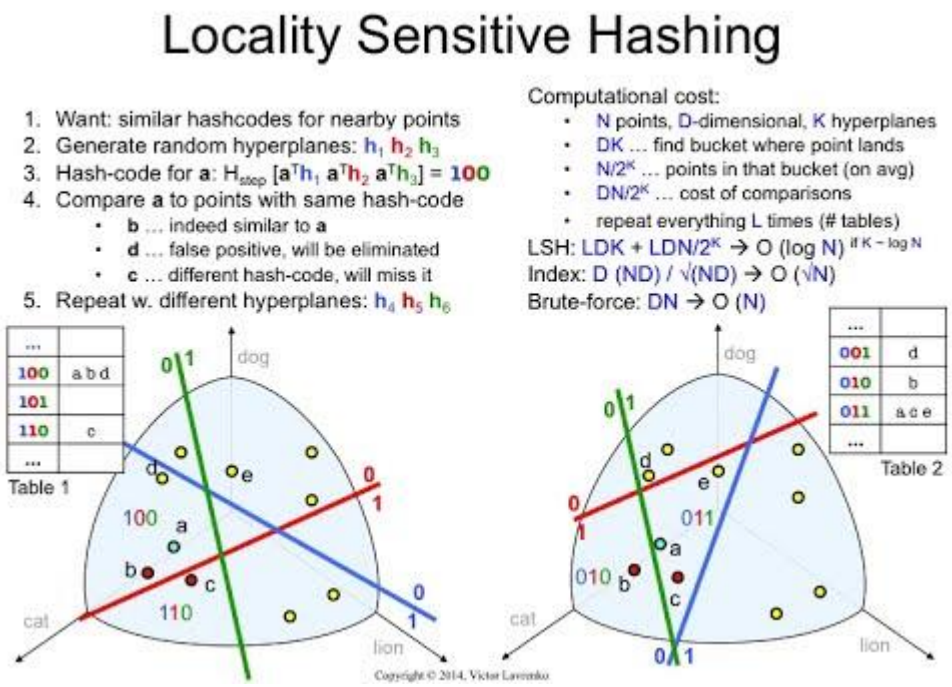


Fig: LSH Performed on 3-D space.

Source: *https://www.google.com/search?q=locality+sensitive+hashing+algorithm&sxsrf=ACYBGNQbN_2IQ-466I27TZMO8x1HnGg9Uw:1575332007212&source=lnms&tbm=isch&sa=X&ved=2ahUKEwimucWFmZjmAhWMtZ4KHcBcAGEQ_AUoAXoECA8QAw&biw=1440&bih=740&dpr=2#imgrc=_h0rDjOY12Mw6M*

But improving the hash function is always a difficult task. So the approach we are taking in this project is to take the conjunction of multiple hashes. So we get only the same points falling in the same buckets. The rest of the points are safely discarded. Multiple layers of hashes are made to and the disjunction is performed to avoid the "misses".

2.4 Classification

Classification is the process of labeling an unlabeled set of data, given the program is trained previously with a set of labeled data. The process comes under supervised learning.

SVM Classifier

The support vector machine is a supervised learning model that analyzes data used for classification and regression analysis. More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Sometimes, it will be difficult to get a classifier in the two-dimensional space. So the process of kernelization is carried out where the dimensionality of the data points is increased to higher dimensions till a classifier is obtained.

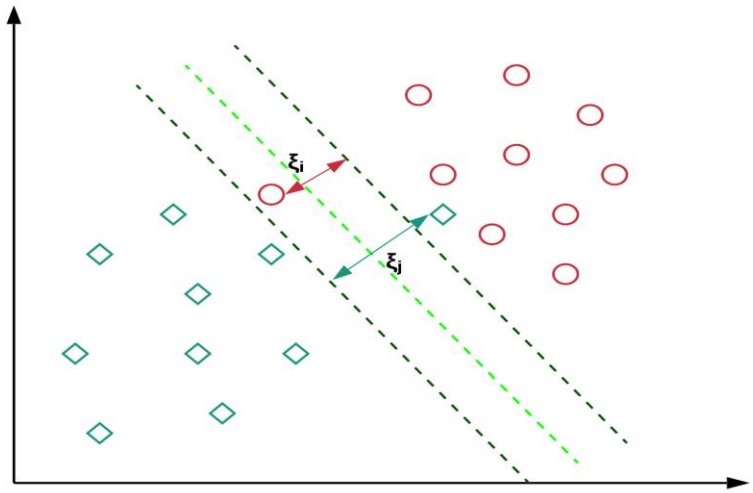


Fig: Support Vector Machine (SVM)

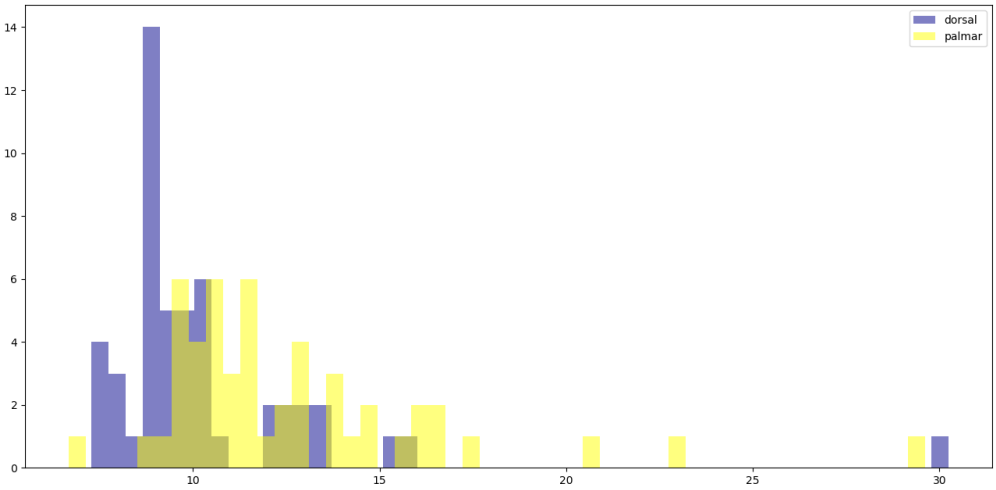
Source:
https://www.google.com/search?q=support+vector+machine&sxsrf=ACYBGNRJ0k3IX_K8wzvmNKFp4cd2XScbIA:1575330953178&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjtqfiOlZjmAhWXq54KHRo3DcwQ_AUoBHoECBIQBg&biw=1440&bih=789#imgrc=uclWseVFEzgYBM

SVMs can perform both linear classifications as well as non-linear classification. For performing the non-linear classification, the kernelization trick is performed, where the inputs are mapped to high-dimensional feature space.

3 Implementation

Task 1:

The objective of the task1 is to compute k latent semantics in the feature space for both the dorsal images and palmar images. Given a folder with unlabeled images, the system has to label it as palmar or dorsal using the latent semantics.
Here we tried two approaches. At first, we have used HOG as the feature descriptor in this task. The dimensionality reduction technique used here is PCA. The distance measure that we have used here is euclidean distance. The basis vectors were projected on to the latent semantic for both dorsal and palmar and we get the values of the projection. The value is sorted out and the maximum value among both is taken to classify the image as either dorsal or palmar. However, this approach failed as it was giving inaccurate results.
So we took another approach. In this new approach, we will find the latent semantic for both dorsal as well as palmar images. We used the PCA technique for dimensionality reduction and SIFT bag of words for feature extraction. Then we find the centroid of the latent semantic for both the cases. Then we store the distance of each image latent semantic to the centroid in a list. The process is done for both palmar and dorsal. Then we are plotting a histogram where the x-axis reflects the distance of the image from the centroid. We found that, in the resultant histogram, there is an overlap between the dorsal and palmar distribution. We mark a threshold in this overlap. So we classify those images that comes after the threshold as palmar and those that come before the threshold as dorsal.



Task 2:

In this task, the folder of dorsal/palmar labeled images is given and for a user-supplied c, we have to compute c clusters associated with dorsal images and palmar images. Given an unlabeled set of images, the system has to label it as dorsal or palmar. Here, for clustering, we have implemented the k-means algorithm for clustering. we are using the euclidean distance to find the distance between the given query image and

the centroid of the clusters. The training data set images SIFT descriptor is computed and the corresponding bag of words is obtained for bag size 70. This bag of words is used for clustering the data into clusters. We are storing the centroid of each cluster in the database. The query images bag of words is also obtained. The distance of the query images bag of words is measured from each of the centroids of the cluster for respective labels and the minimum distance from both palmar and dorsal images clusters is taken for classifying the image to either dorsal or palmar.

Task 3:

In this task, we have to Implement a program which, given a value k, creates an image-image similarity graph, such that from each image, there are k outgoing edges to k most similar/related images to it. Given 3 user-specified image ids on the graph, the program identifies and visualizes K's most dominant images using Personalized Pagerank (PPR). Here we are using weighted convergence. The page rank scores calculated in an iteration is used for the next iteration as well. Here the seed value is 3 since we are taking 3 seed nodes. Here we assume that the seed node has more importance than the other nodes. Once the stopping condition is reached, the iteration is stopped and the user can give an input k to visualize the K most dominant images.
Here we are using HOG as the feature extraction technique and on top of that, we are performing PCA dimensionality reduction technique. The number of latent semantic features that we are considering here is 55, as we tried with the other numbers but 55 dimensions were yielding the accurate results.

Task 4:

SVM classifier
The support vector machine is a supervised learning model that analyzes data used for classification and regression analysis. More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Sometimes, it will be difficult to get a classifier in the two-dimensional space. So the process of kernelization is carried out where the dimensionality of the data points is increased to higher dimensions till a classifier is obtained. In our implementation, we are using the polynomial kernel and we have implemented a soft margin SVM classifier as we had some corner case data that lies in the part of the classifier which is not meant to be present in. So in order to deal with such corner cases, we are using soft margin implementation.
Here, our objective is to maximize width W.

```
W = ((X+ - X-).w)/|w|
```

The aim is to maximize the margin (distance from the hyperplane to the closest instance) 2/||w||. Here we have used LaGrange multipliers. The Lagrange function used is:

$$L(x, \lambda, \alpha) = f(x) + \sum_{\blacksquare} \blacksquare \lambda_{\blacksquare_i} h_{\blacksquare_i} + \sum_{\blacksquare} \blacksquare \alpha_{\blacksquare_i} g_{\blacksquare_i}$$

The idea for the usage of kernels is that in case if we are not able to find the classifier in lower-dimensional space, then the dataset is mapped to higher-dimensional space [6]. We have implemented the fit function and predict function for the SVM classifier.

Decision Tree Classifier

Here we have implemented mainly three functions: fit, predict and Gini impurity. Gini impurity is defined as $1 - \sum p2$ over all classes with p as the frequency of a class with a node.

We calculate the Gini impurities of each split. Gini impurity for each split is the weighted Gini impurity of the children. We are building the decision tree by recursively finding the best split, splitting recursively until the maximum depth is reached. We find the best split by computing the information gain.

*Information gain = entropy(parent) - [weights average] * entropy(children)*

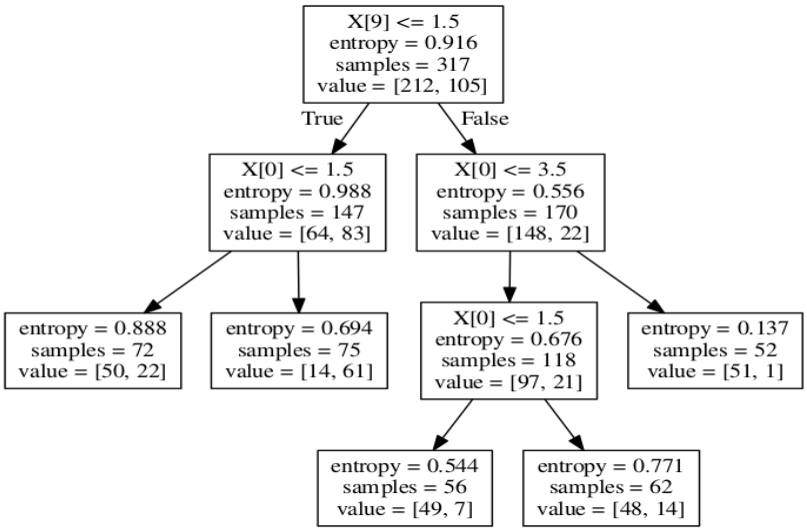


Fig: Decision Tree schematic diagram

Source: https://stackoverflow.com/questions/47873366/can-sklearn-decisiontreeclassifier-truly-work-with-categorical-data

PPR based classification

The Personalized Page Rank algorithm assign page rank scores to nodes with respect to the seed nodes. Nodes that are related to the seed node are ranked higher than other nodes. Here we use the labeled values corresponding to each label as seed values and run the PPR algorithm and get the score of the page for all the data sets for classification. The label for an image is calculated by the formula:

$$label_i = label|P_{ilabel} = max(P_{ilabel1,}P_{ilabe2,}\ldots\ldots\ldots\ldots P_{ilabeln})$$

The image at the index is given the label with respect to the page rank vector with the max page score value. There can also be a chance that even after the stopping condition has reached, there can be some nodes that are not assigned any page rank. So some of the nodes will be assigned a score of 0 for some or for all the vectors.

Task 5:

In task 5a, the locality-sensitive hashing (LSH) tool is implemented. We create an LSH based index for a given input dataset, the number of layers taken here is L and the hashes per layer ask. We are using list of dictionaries.

For task 5b, we are using the visual model HOG and we are obtaining the HOG descriptors. Then we perform dimensionality reduction using PCA. The reason why we are performing dimensionality reduction is to eliminate the chances of the system encountering the memory error.

Here we construct the l-number hash table, each having m dictionaries. Each data point that we have is iterated over and each object is pushed into a bucket. During the iteration, there is a high chance for multiple images to have the same hash value. Thus they fall into the same bucket, ie, we get a collision. Since these are set of image id, we perform a set intersection between all the values in the hash table. The set intersection performed here corresponds to the logical conjunction. Then we perform the set union between all the results that we got after performing conjunction.

Task 6:

In the task 6, we are implementing the Relevance Feedback System on the results of task 5b. For the given t-similar images returned by the LSH query results, the user will select few images as Relevant and Irrelevant images and few images will remain unlabelled. The information of labels given to each image will be sent to feedback system. The following are the details of each feedback algorithm that process the t-similar images based on the labels assigned:

1. SVM based relevance feedback system- Using the label given to the images such as some images are marked as relevant and some are marked as irrelevant and the rest are not marked at all. The relevant and irrelevant mark is considered as training labels in the SVM classifier and the images which were not marked are treated as test images. The SVM classifier trains the training labels and classifies the not marked images as either relevant or irrelevant. It considers the distance from classification boundary as the ranking in classification. For example, an image A is at a distance of x form the boundary and and image B is at a distance of x+1 from the boundary, image B has better rank and appears over top of image A in the reordered list.
2. Decision Tree based relevance feedback system- Using the label given to the images such as some images are marked as relevant and some are marked as irrelevant and the rest are not marked at all. The relevant and irrelevant mark is considered as training labels in the Decision tree classifier and the images which were not marked are treated as test images. The Decision Tree classifier trains the training labels and classifies the not marked images as either relevant or irrelevant.
3. The Personalized Page Rank algorithm assign page rank scores to nodes with respect to the seed nodes. Nodes that are related to the seed node are ranked higher than other nodes. Here we use the labeled values corresponding to each label(relevant, irrelevant) as seed values and run the PPR algorithm and get the score of the page for all the data sets for classification. The image at the index is given the label with respect to the page rank vector with the max page score value. There can also be a chance that even after the stopping condition has reached, there can be some nodes that are not assigned any page rank. After we get page rank score for the above two labels, the difference of the score is used to provide new ranking.
4. Probabilistic Relevance Feedback system: We have implemented Naive bayes classifier for probabilistic relevance feedback system. Using the label given to the images such as some images

are marked as relevant and some are marked as irrelevant and the rest are not marked at all. The relevant and irrelevant mark is considered as training labels in the Naive bayes classifier and the images which were not marked are treated as test images. The Naive bayes classifier trains the training labels and classifies the not marked images as either relevant or irrelevant.

4 Interface Specification

4.1 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. The code has been tested on a machine with 8GB and 16GB RAM.

4.2 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.

```
parser = argparse.ArgumentParser()
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 labelled csv name", default="None")
parser.add_argument('-u', '--unlabel',action="store", dest="unlabel",help="Provide unlabelled csv name", 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('-c', '--centers',type=int, dest="centers",help="Provide centers count", default=10)
parser.add_argument('-L', '--layers',type=int, dest="layers",help="Provide layers count", default=10)
parser.add_argument('-m', '--mimage',type=int, dest="mimage",help="Provide m value to get m similar images", default=10)
parser.add_argument('-t', '--taskid',type=int, dest="taskid", help="Provide the task number", default=-1)
parser.add_argument('-I', '--list',type=list, action='store',dest='list',help='Pass the image list',required=True)
parser.add_argument('-T', '--type',action="store", dest="type",help="Provide type of classifier", default="None")
```

4.2.1 Task 0

To setup the imagedb with the collections for all the labelled and unlabelled data samples metadata and feature extraction for required extraction technique used in all the task.

python phase3_main_script.py -d ../Dataset -t 0

4.2.2 Task 1

For given k latent semantics, labelled image set and unlabelled mage set, execute the following query.

python phase3_main_script.py -t 1 -k =30 -l labelled_set1 -u unlabelled_set2

4.2.3 Task 2

For given C Cluster, labelled image set and unlabelled mage set, execute the following query.

python phase3_main_script.py -t 2 -c 5 -l labelled_set2 -u unlabelled_set1

4.2.4 Task 3

For given k outgoing edges in image-image similarity, K most dominant images, labelled set folder and the list of user specified images on the graph, execute the following query

python phase3_main_script.py -t 3 -c 5 -m 10 -I "Hand_0008333.jpg Hand_0006183.jpg Hand_0000074" -l labelled_set2

4.2.5 Task 4

For given C similar images, classification algorithm, labelled image set and unlabelled mage set, execute the following query.

python phase3_main_script.py -t 4 -c 5 -T PPR -l labelled_set2 -u unlabelled_set2

4.2.6 Task 5

For given L Layers, k number of hash per layers, Query image name and m similar image, execute the following query.

python phase3_main_script.py -t 5 -i Hand_0000674.jpg -m 20 -L 10 -k 10

4.2.7 Task 6

For task 6 we don't have to run the query. The results from task 5 will pop the visualizer with the m similar images, user will have to select the Relevant/Irrelevant/None radio button given beside each image.

5 System Requirements, Installations, and Executions

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

- OpenCV-control-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 a 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 the database to the user, although not necessary.

6 Related Works

The work Duplicate Detection for Identifying Social Spam in Microblogs concentrates on identifying social spam in microblogs using the concept of duplicate detection. The paper focuses on identifying the potential spammers who copy a piece of information from others. The paper also discusses identifying various tweets that are actually copied. In this paper, however, they have used LSH with filtering rather than going with the conventional LSH approach. The MapReduce implementation of the algorithm is also presented in the paper. The duplicate detection is an area where extensive research is ongoing [5].

The paper Efficient near-duplicate Detection and Sub-image Retrieval introduces a system for near-duplicate detection and sub-image retrieval, which specifically focuses on finding copyright violations and detecting forged images. Again, the paper deploys locality-sensitive hashing to index the local descriptors. However, there were some limitations to the proposed model, which includes matching similar images of the same scene, even if they are not near duplicates. [4]

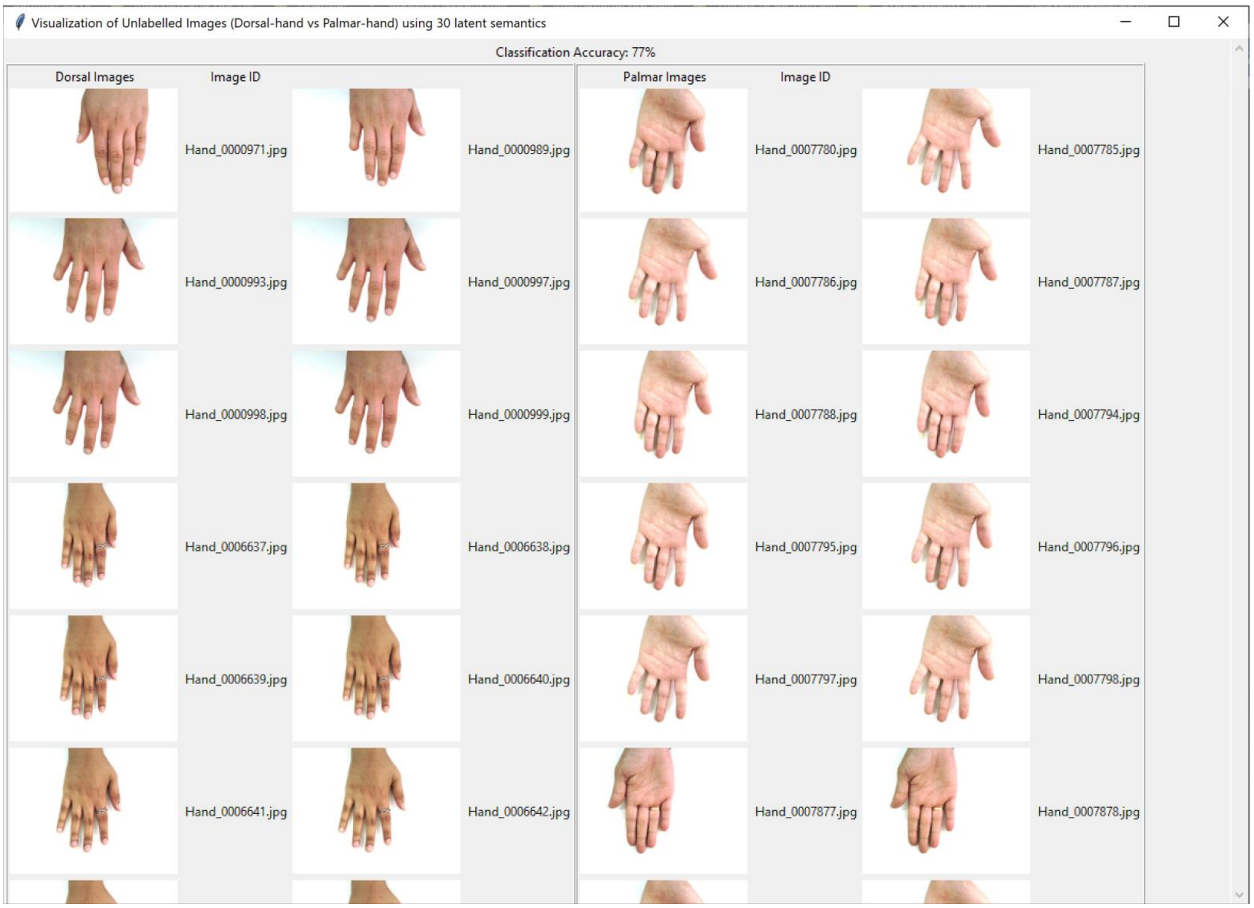
7 Results

7.1 Task 1

In testing Task 1 the following 4 queries were performed.

Query 1: k: 30, Folder: Labelled/Set1, Classify: Unlabelled/Set1

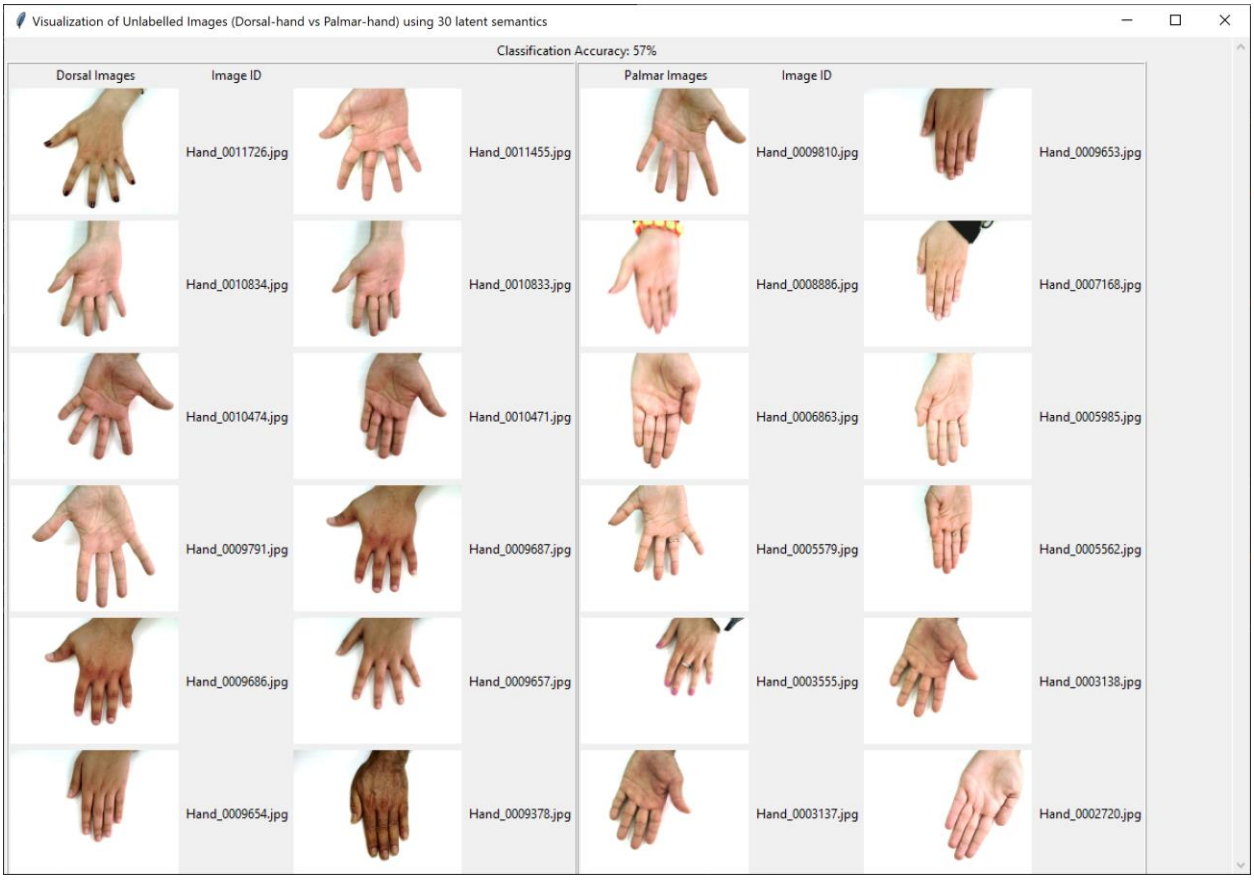
Visualizer for Unlabelled Images using 30 Latent Semantics



Interpretation: We are getting the above results which seems to have significant improvement from phase 2 task5. The accuracy we are getting for this query is 77%. We have done Normal distribution of distance of transformed feature space for “dorsal” and “palmar” from its centroid respectively. As we are using the Set 1(which has more similar images) for labelled as well as unlabelled images, the results we are getting are better as compared to when used labelled and unlabelled images from different sets. We have used SIFT bag of words as feature descriptors in this Task.

Query 2: k: 30, Folder: Labelled/Set1, Classify: Unlabelled/Set2

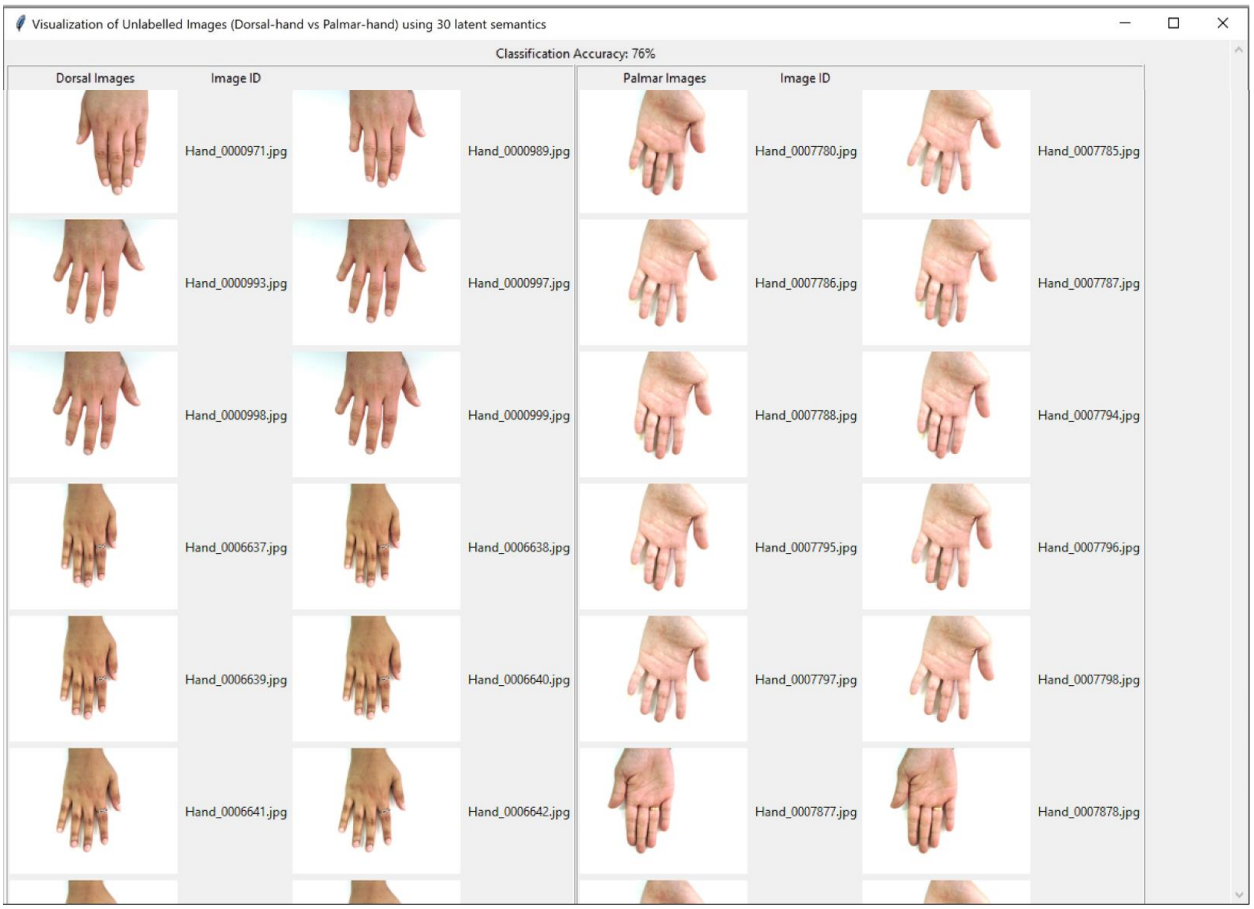
Visualizer for Unlabelled Images using 30 Latent Semantics



Interpretation: Now in this query we have training set from labelled set1 and test set unlabelled set2, the accuracy from has come down to 57% . As the training and testing sets are significantly different from each other, their transformed feature space is also different. As a result we can see some misclassification for dorsal as well as palmar images.

Query 3: k: 30, Folder: Labelled/Set2, Classify: Unlabelled/Set1

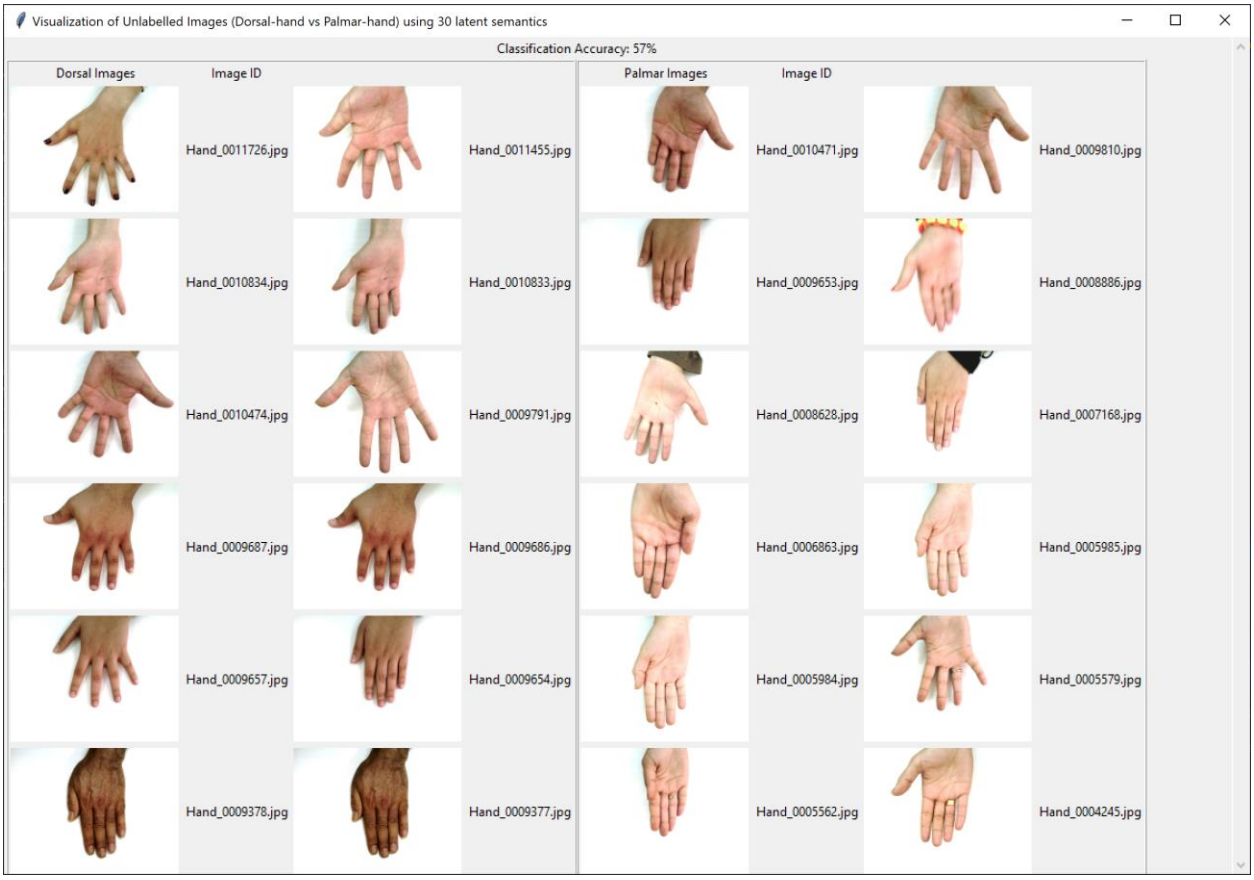
Visualizer for Unlabelled Images using 30 Latent Semantics



Interpretation: Now in this query we have training set from labelled set2 and test set unlabelled set1, the accuracy in this query is 76%. As the training and testing sets are significantly different from each other, their transformed feature space is also different. As a result we can see some misclassification for dorsal as well as palmar images. We have done Normal distribution of distance of transformed feature space for "dorsal" and "palmar" from its centroid respectively. In this query, the distance of more number of dorsal images are less than threshold value.

Query 4: k: 30, Folder: Labelled/Set2, Classify: UnlabelledSet2

Visualizer for Unlabelled Images using 30 Latent Semantics

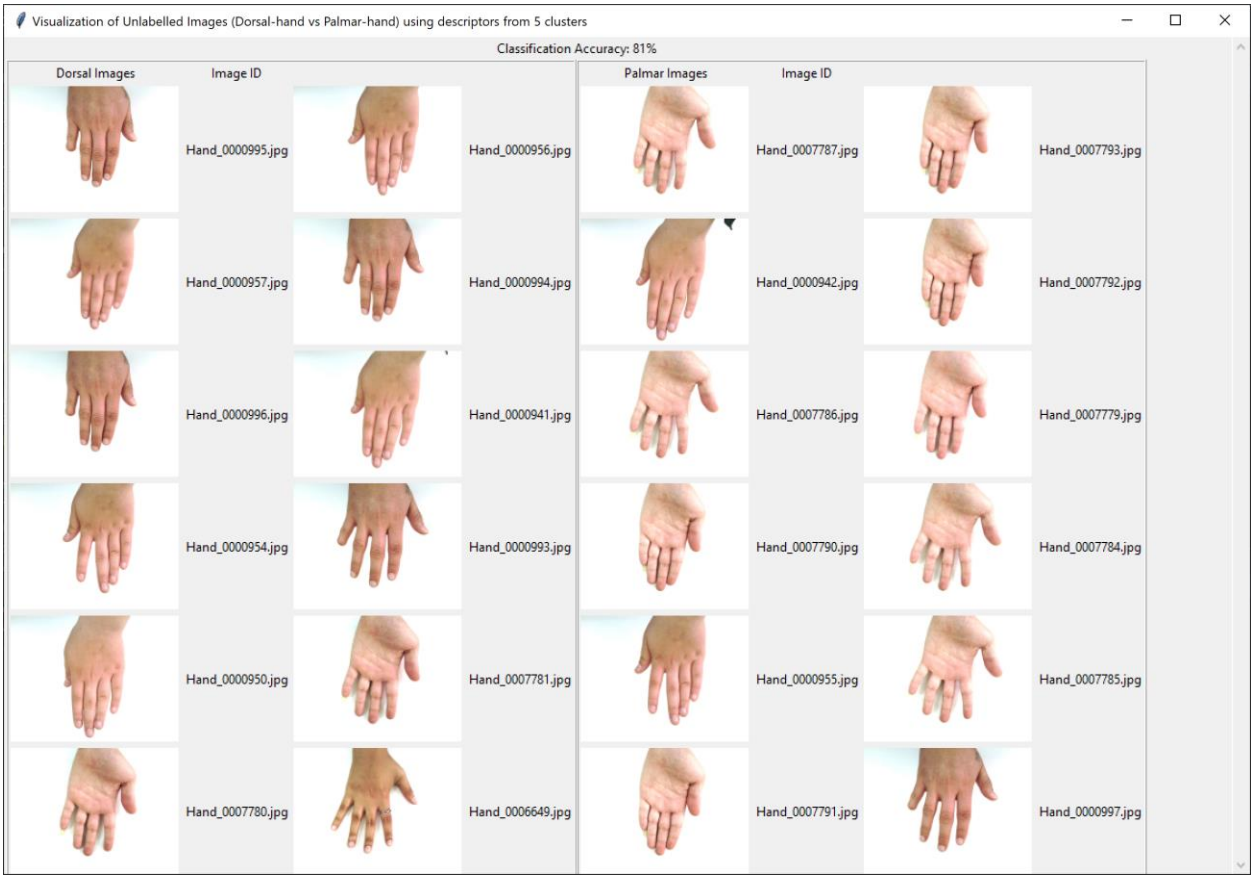


Interpretation: Now in this query we have training set from labelled set2 and test set unlabelled set1, the accuracy has come down to 57% . As the training and testing sets are significantly different from each other, their transformed feature space is also different. As a result we can see some misclassification for dorsal as well as palmar images. We have used SIFT bag of words as feature descriptors in this Task.

7.2 Task 2

Query 1: c: 5, Folder: Labelled/Set2, Classify: Unlabelled/Set1

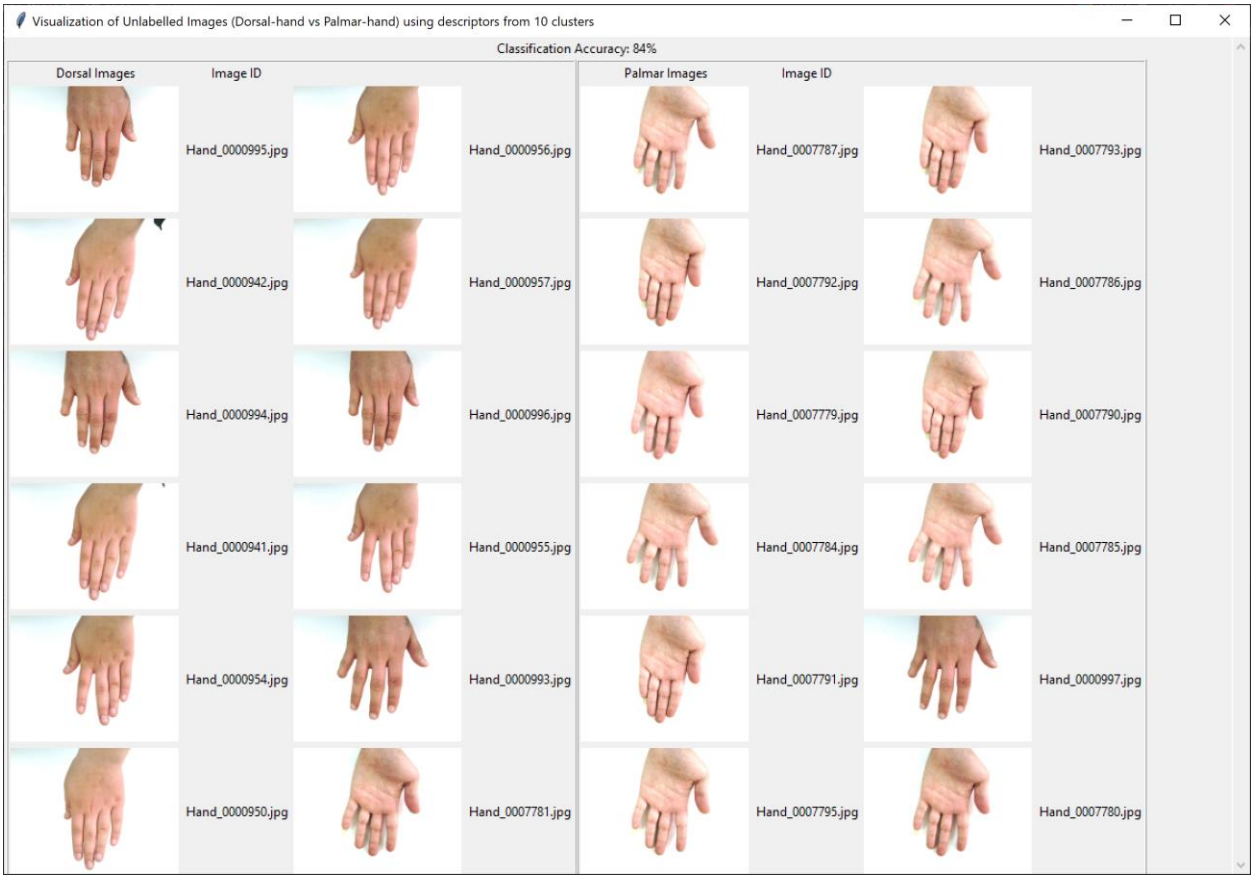
Visualizer for Unlabelled Images using descriptors from 5 clusters



Interpretation: The Accuracy of Unlabelled/Set1 folder is found to be 81% which is pretty good when 5 clusters are formed on training images for dorsal and palmar respectively, which has some noise of accessories or cloth the person has worn. This is because SIFT has identified the key points for dorsal hands and palmar sides of hand correctly apart from noise and the features descriptors of these identified keypoint are able to discriminate. So, when we try to compare the query image bag and with the cluster representative's bag of words it able to find the perfect match for most of the unlabelled images.

Query 2: c: 10, Folder: Labelled/Set2, Classify: Unlabelled/Set1

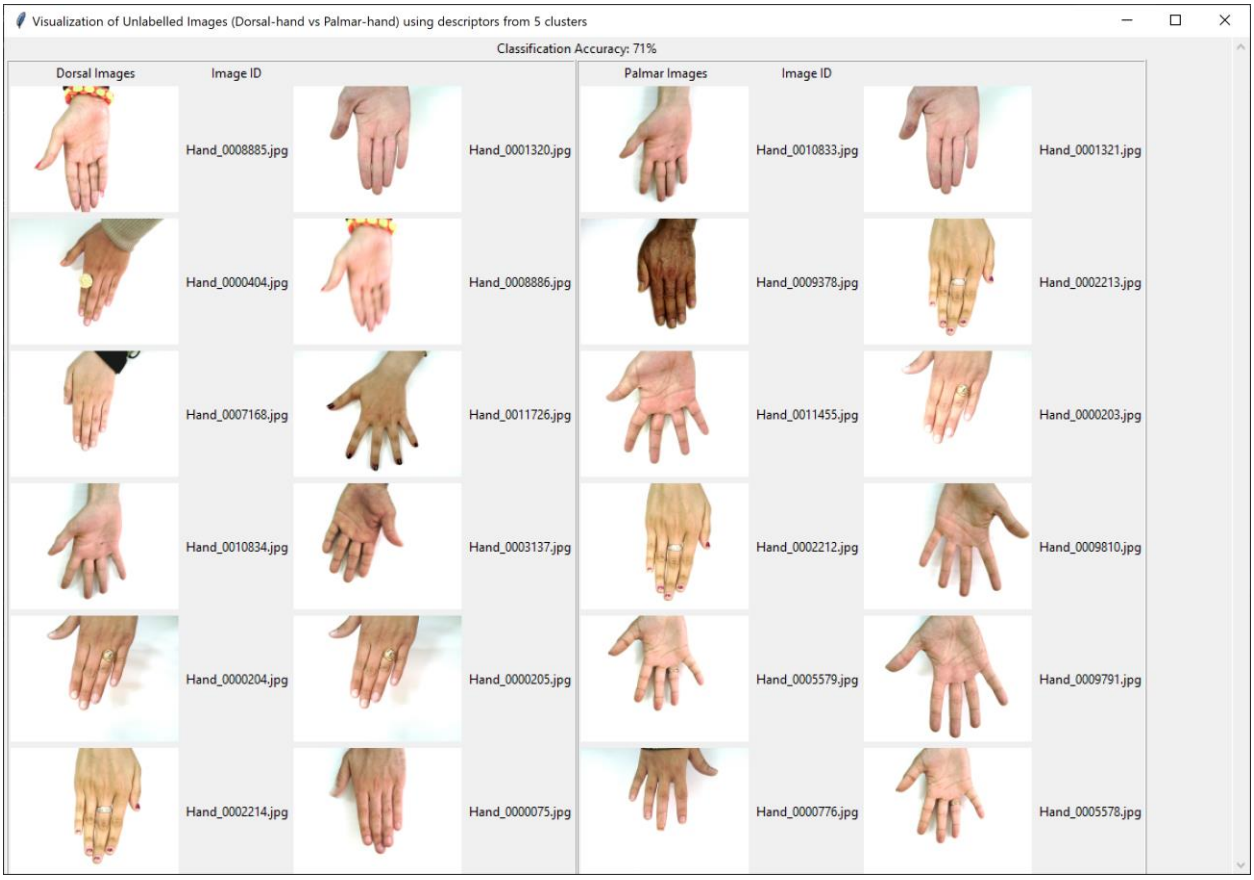
Visualizer for Unlabelled Images using descriptors from 10 clusters



Interpretation: The Accuracy of Unlabelled/Set1 folder is found to be 84% which is really good when 10 clusters are formed on training images for dorsal and palmar respectively, which has some noise of accessories or cloth the person has worn. This is because SIFT has identified the key points for dorsal hands and palmar sides of hand correctly apart from the noise and the features descriptors of these identified keypoint are able to discriminate. So, when we try to compare the query image bag and with the cluster representative's bag of words it able to find the perfect match for most of the unlabelled images. Thus, with the Increase in the number of cluster gives good results when unlabelled data has less noise in it.

Query 3: c: 5, Folder: Labelled/Set2, Classify: Unlabelled/Set2

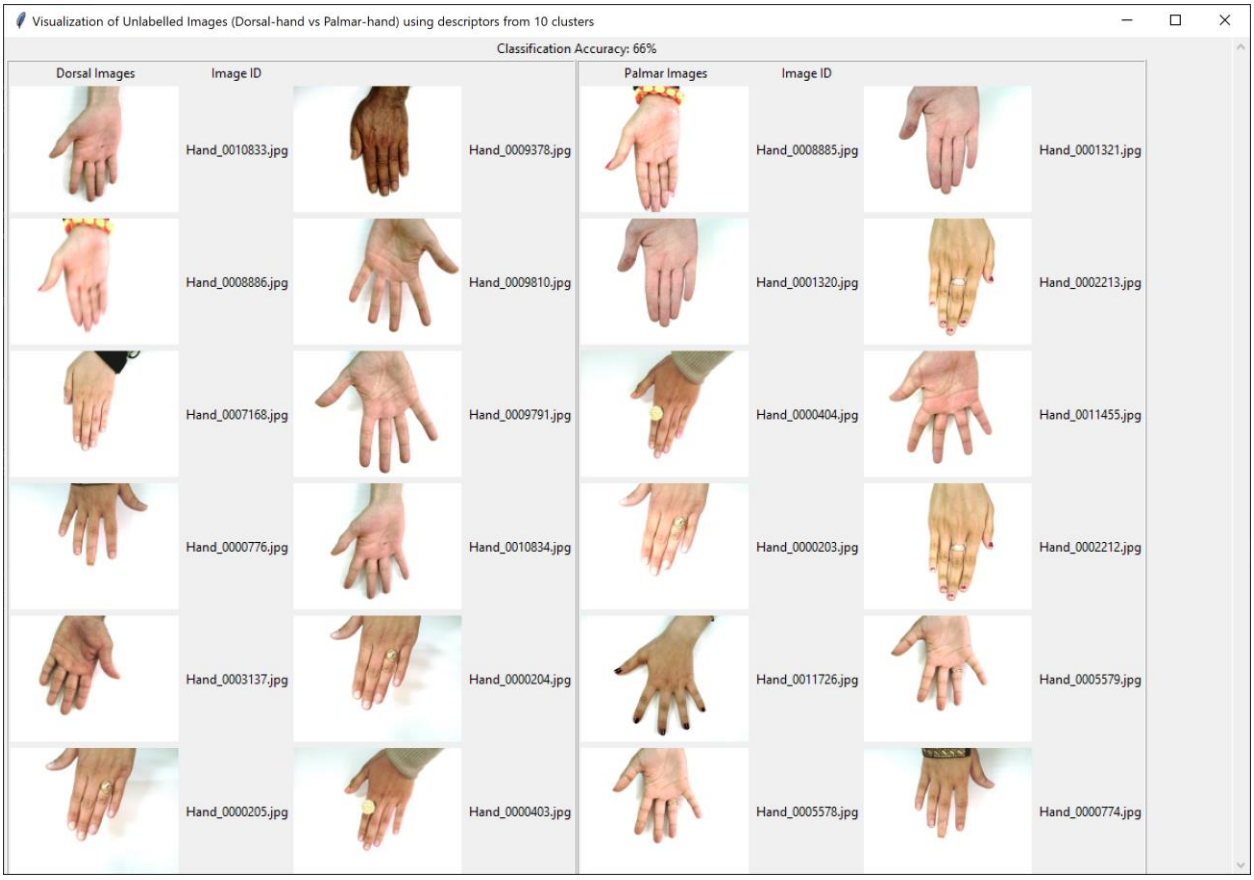
Visualizer for Unlabelled Images using descriptors from 5 clusters



Interpretation: The Accuracy of Unlabelled/Set2 folder is found to be 71% which is not good in comparison with the previous results, when 5 clusters are formed on training images for dorsal and palmar respectively. It was found that both the labelled/Set 2 and Unlabelled/Set2 has images with noise in it. This leads to extraction of noise features in SIFT and it decreases the discriminative power of SIFT features. So, when we try to compare the query image bag and with the cluster representative's bag of words it is able to find the perfect match for 71% of the unlabelled images which has less noise in it. Thus, noise leads to a decrease in predictive power of SIFT features.

Query 4: c: 10, Folder: Labelled/Set2, Classify: Unlabelled/Set2

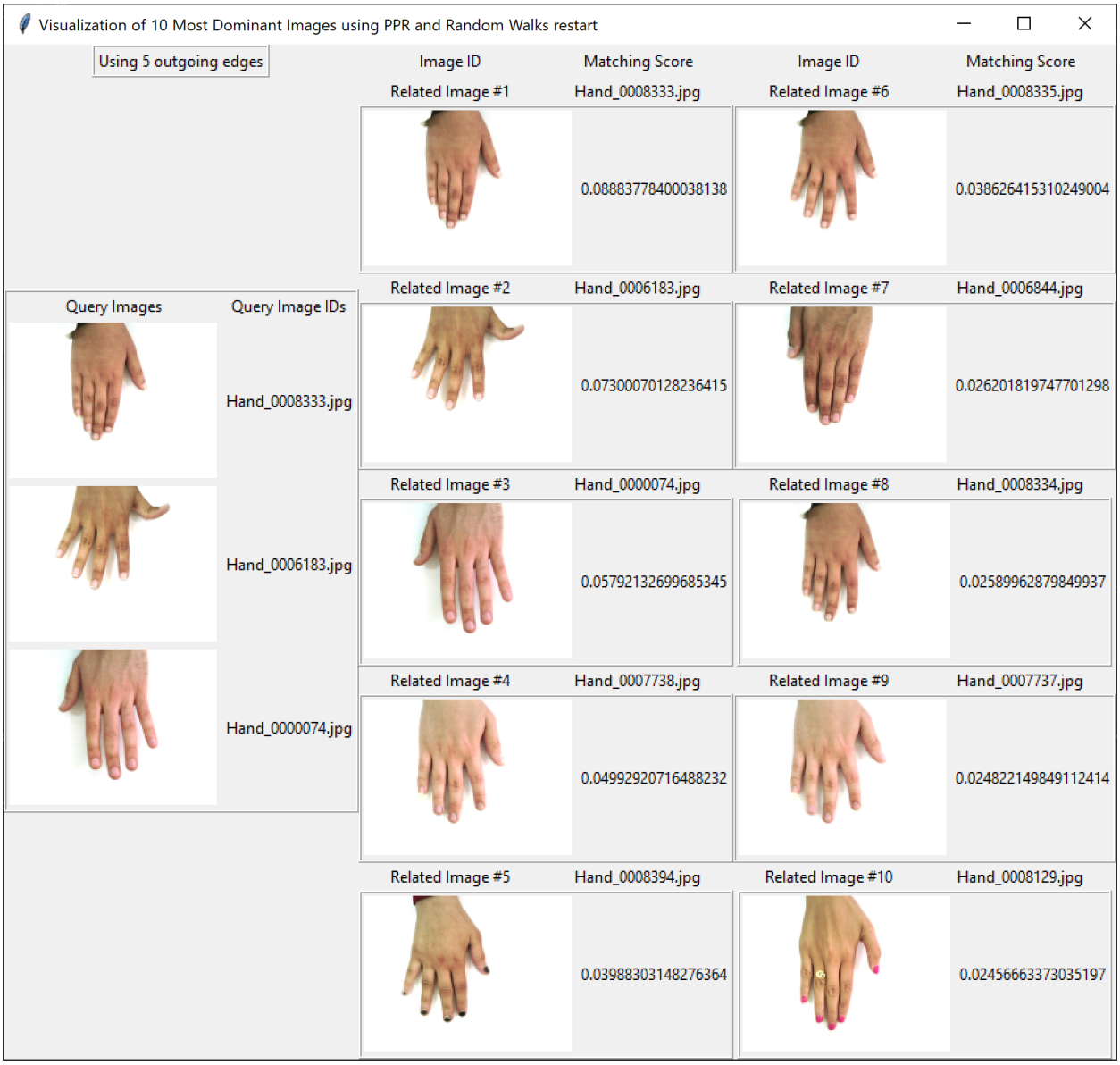
Visualizer for Unlabelled Images using descriptors from 10 clusters



Interpretation: The Accuracy of Unlabelled/Set2 folder is found to be 66% which is not good in comparison with the previous results, when 10 clusters are formed on training images for dorsal and palmar respectively. It was found that both the labelled/Set 2 and Unlabelled/Set2 has images with noise in it. This leads to extraction of noise features in SIFT and it decreases the discriminative power of SFIT features. So, when we try to compare the query image bag and with the cluster representative's bag of words it is able to find the perfect match for 66% of the unlabelled images which has less noise in it. Thus, an increase in the number of clusters decreases the predictive power of SIFT features when we have more noisy data.

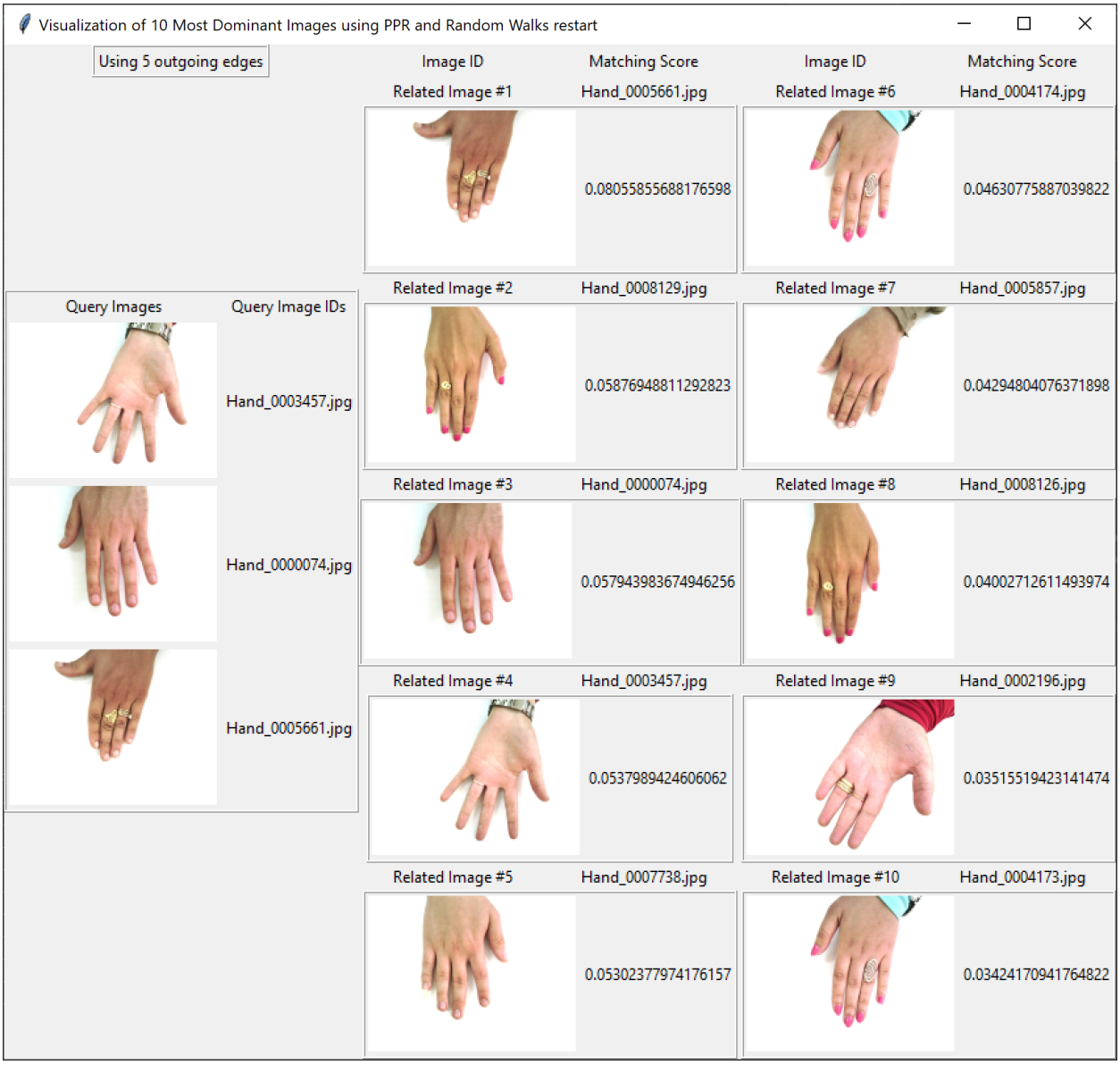
7.3 Task 3

Query 1: k: 5, K: 10, Folder: Labelled/Set2, Image IDs: [Hand_008333.jpg, Hand_0006813.jpg, Hand_0000074.jpg]



Interpretation: The above images show the top 10 images with maximum score after PPR has converged . Here, a graph of 5 outgoing edges was created first and then random walk from the given images was done. We are getting all dorsal images with same orientation because it the input seed images were dorsal in nature

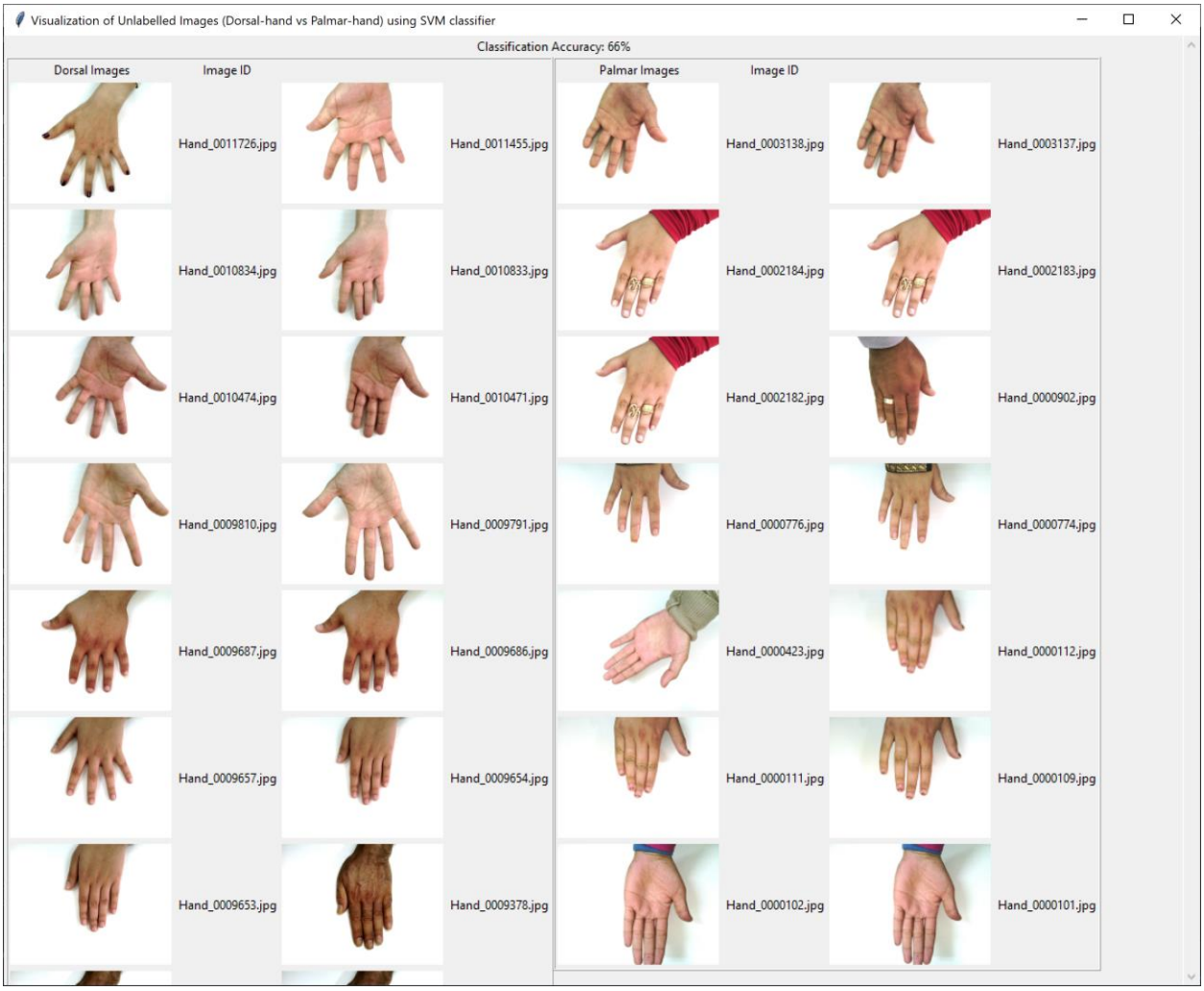
Query 2: k: 5, K: 10, Folder: Labelled/Set2, Image IDs: [Hand_003457.jpg, Hand_0000074.jpg, Hand_0005661.jpg,]



Interpretation: The above images show the top 10 images with maximum score after PPR has converged . Here, a graph of 5 outgoing edges was created first and then random walk from the seed images was done. We are getting mostly dorsal images with same orientation and few palmer because the input seed images were having 2 dorsal and only one palmer image

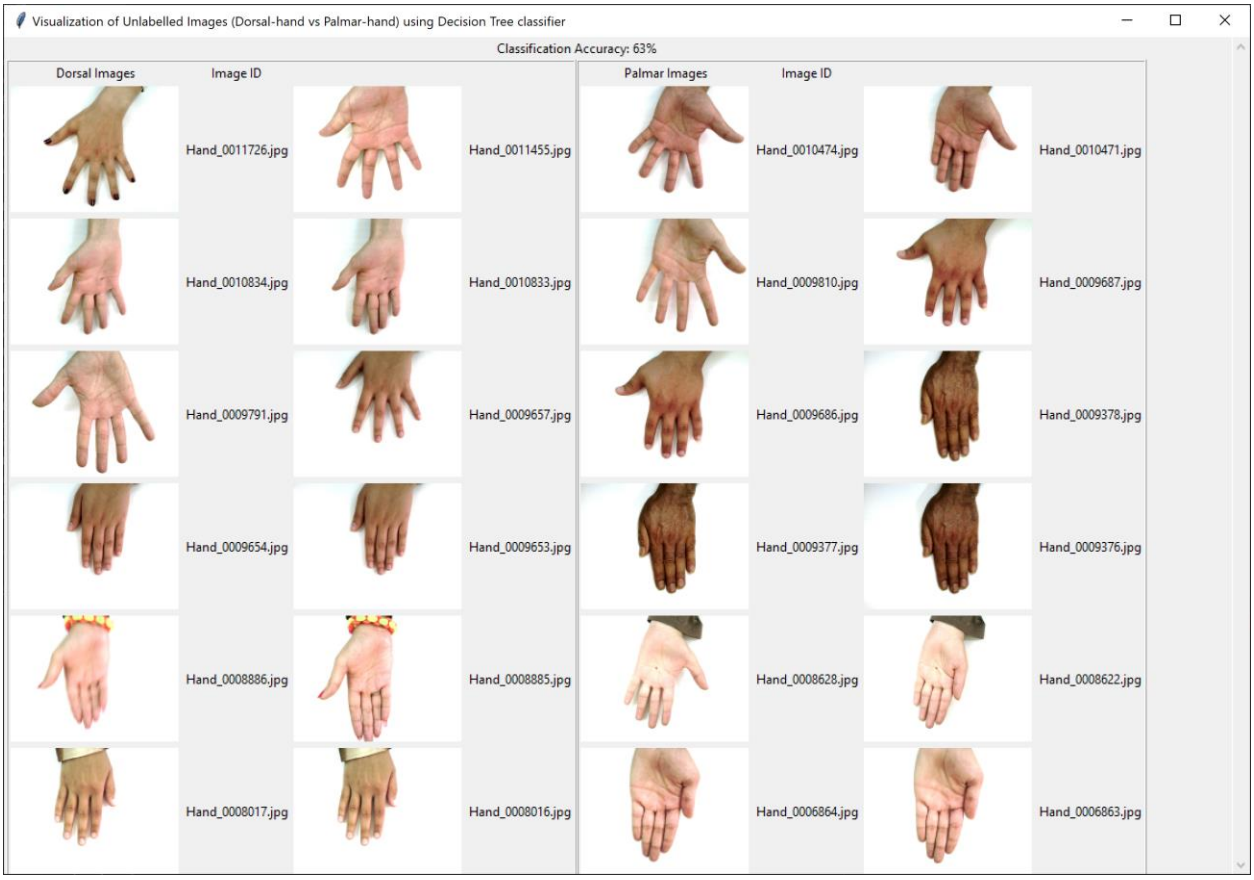
7.4 Task 4

Query 1: classifier: SVM, Labeled Image Folder: Labelled/Set2, Classify: Unlabelled/Set2



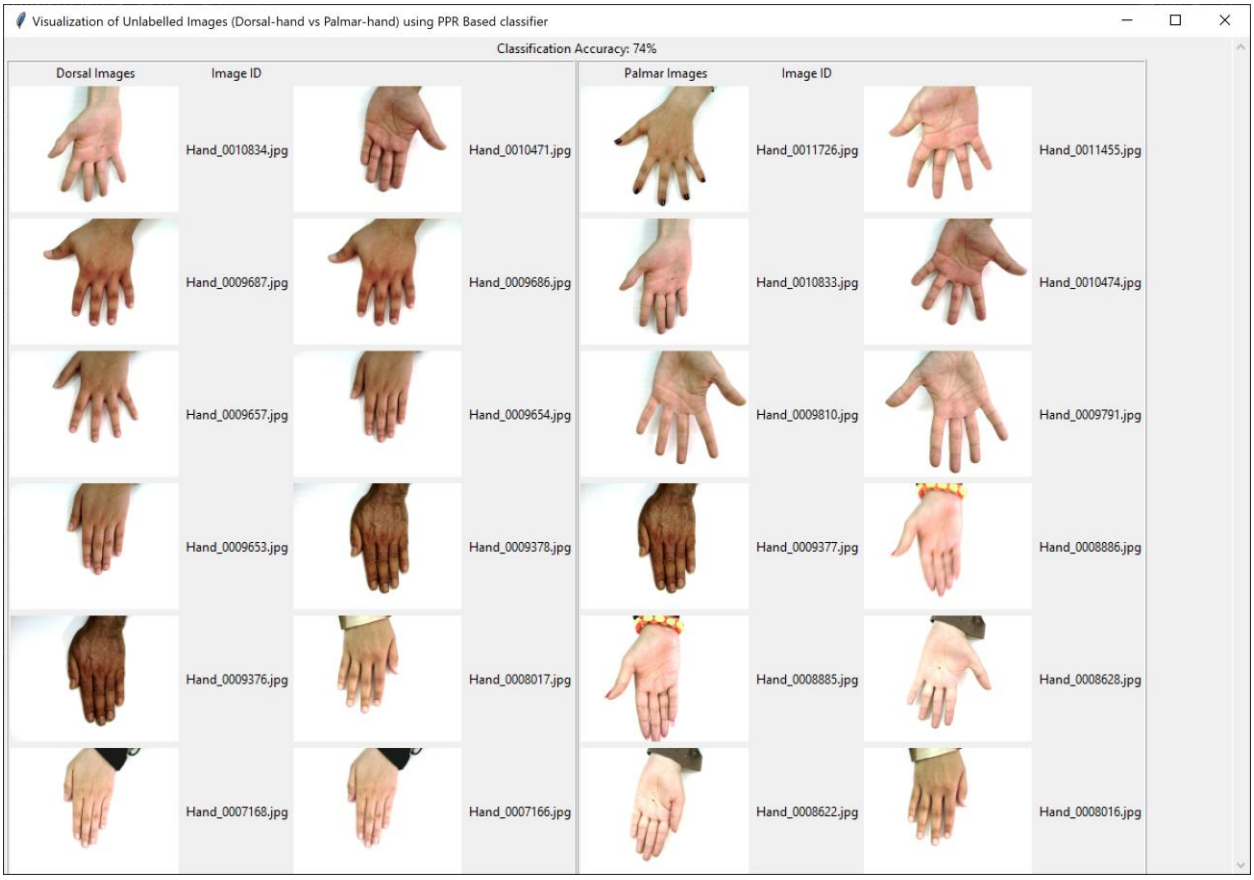
Interpretation: We are getting the above results with the accuracy of 66%. SVM finds only features useful for separating the data. So, looking at the score in the terminal we could say that the absolute size of the coefficient relative to the other ones gives an indication of how important the feature was for the separation. As the data of transformed features of "dorsal" and "palmar" images don't are not linearly separable in the transformed feature space, using the polynomial kernel with soft margin error gives the above results which has the best accuracy in multiple runs which we tried so far.

Query 2: classifier: Decision Tree, Labeled Image Folder: Labelled/Set2, Classify: Unlabelled/Set2



Interpretation: We are getting the above results with the accuracy of 63% with depth of tree as 4. Though we can see that misclassification is relatively high for Decision tree than SVM. We split the transformed features into two or more homogeneous sets based on most significant splitter / differentiator. As we can see from the above results the splits are not significantly differentiating between the transformed features of "dorsal" images and transformed features of "palmar" images.

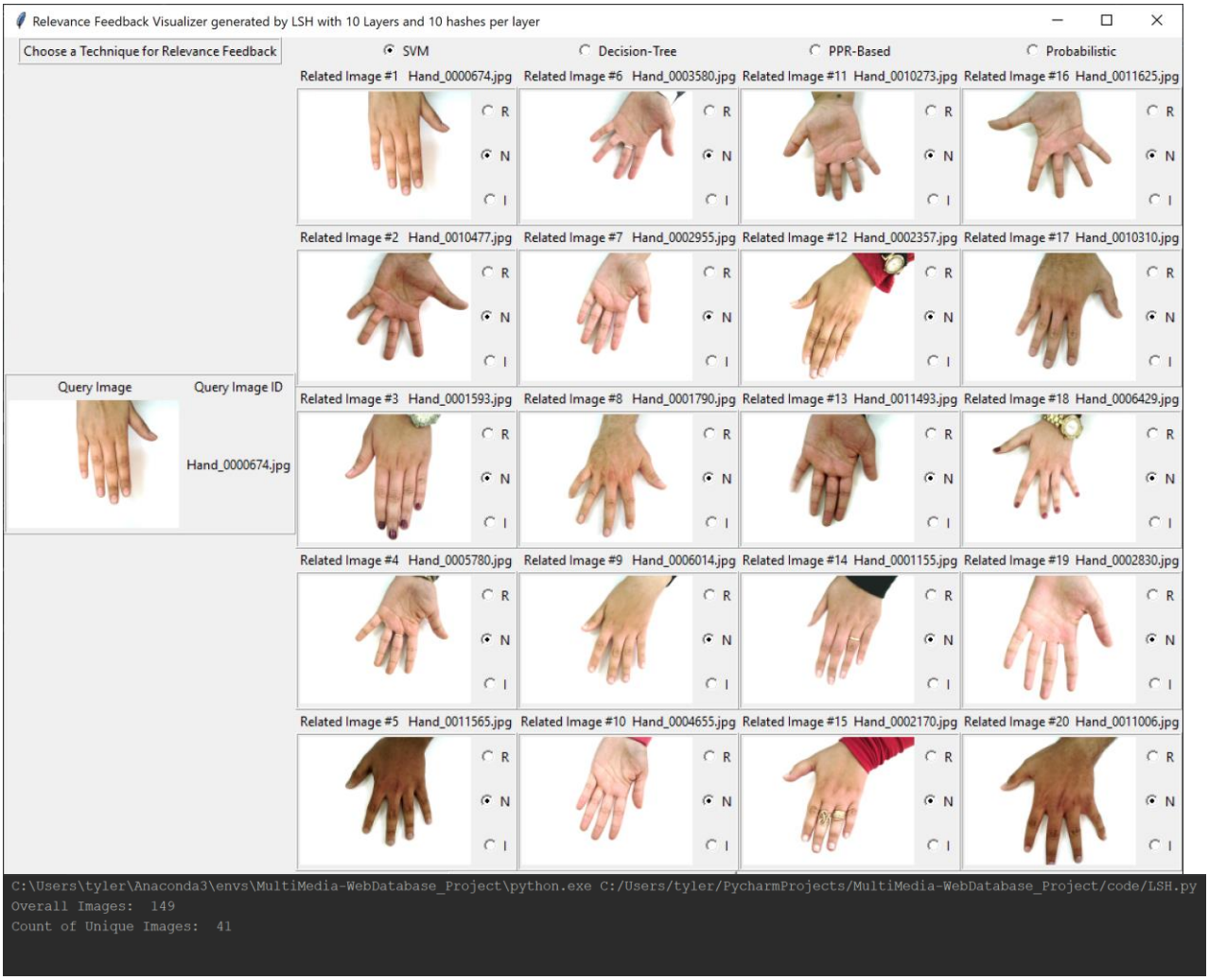
Query 3: classifier: PPR, Labeled Image Folder: Labelled/Set2, Classify: Unlabelled/Set2



Interpretation: We are getting the above results with the accuracy of 74% where we have started PPR with labelled images as 2 classes Though we can see that misclassification is relatively less for PPR compared to above 2 classifiers.

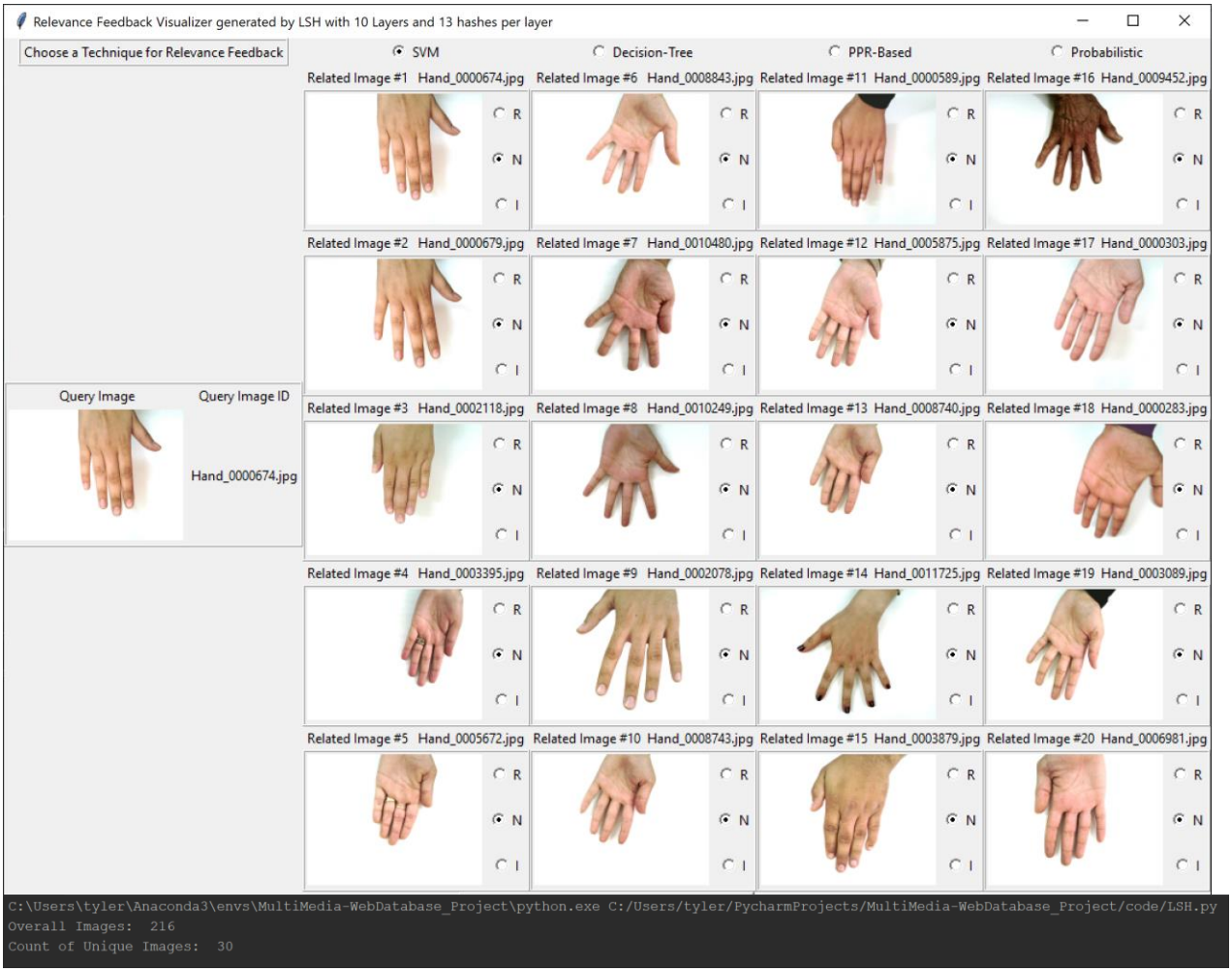
7.5 Task 5

Query 1: L: 10, k: 10, t: 20, Query Image: Hand_0000674.jpg



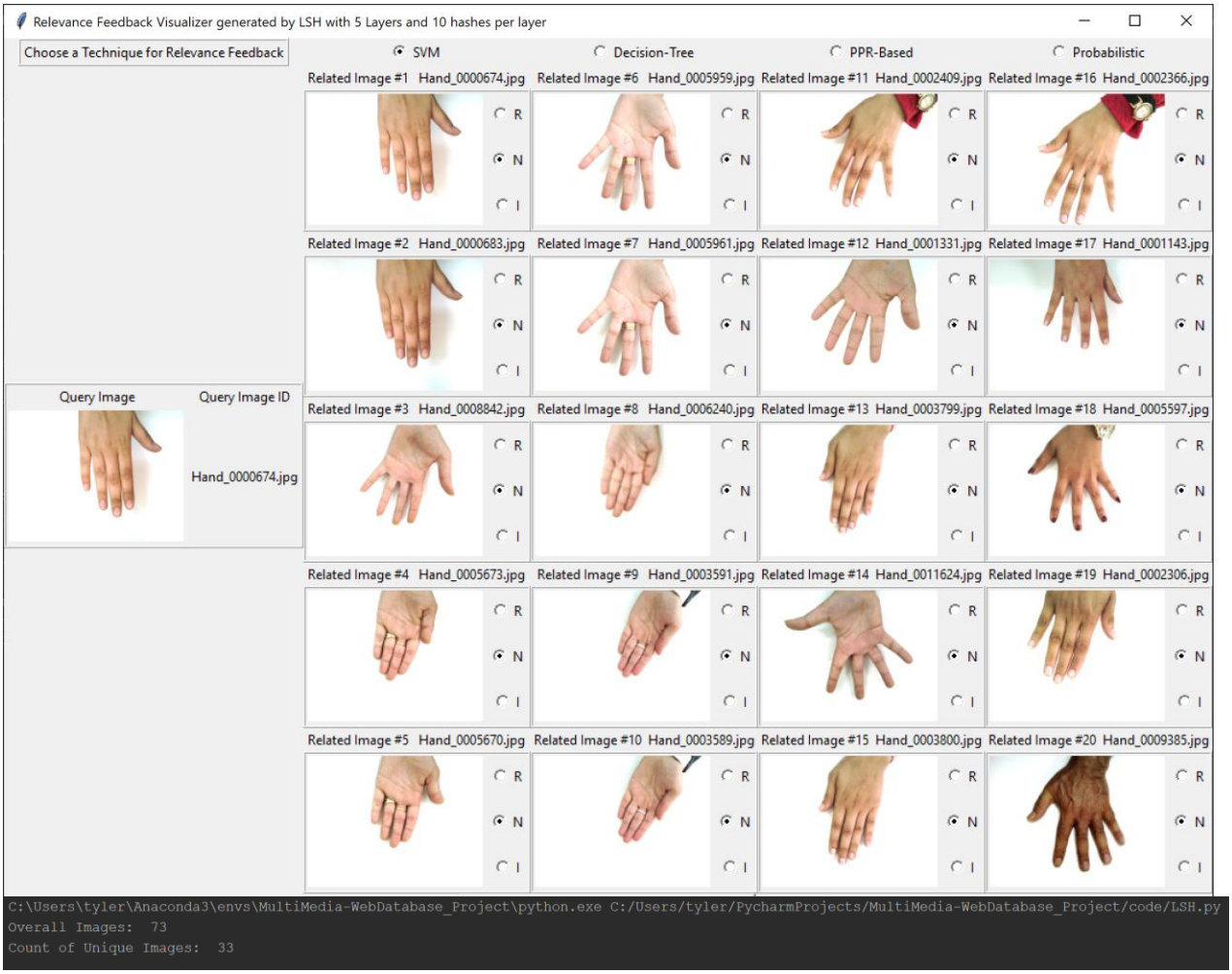
Interpretation: The above images show the top 20 (value of k) similar images to the given query image 'Hand_0000674.jpg'. An index structure using locality sensitive hashing was created and this index structure was then used to extract top 20 similar images. The number of layers have been taken as 10 and the number of hashes have been taken as 10. Total number of overall and unique images have been displayed to the console. The above output shows top 20 similar images from 41 unique images.

Query 2: L: 10, k: 13, t: 20, Query Image: Hand_0000674.jpg



Interpretation: The above images show the top 20 (value of k) similar images to the given query image 'Hand_0000674.jpg'. An index structure using locality sensitive hashing was created and this index structure was then used to extract top 20 similar images. The number of layers have been taken as 10 and the number of hashes have been taken as 13. Total number of overall and unique images have been displayed to the console. The above output shows top 20 similar images from 30 unique images.

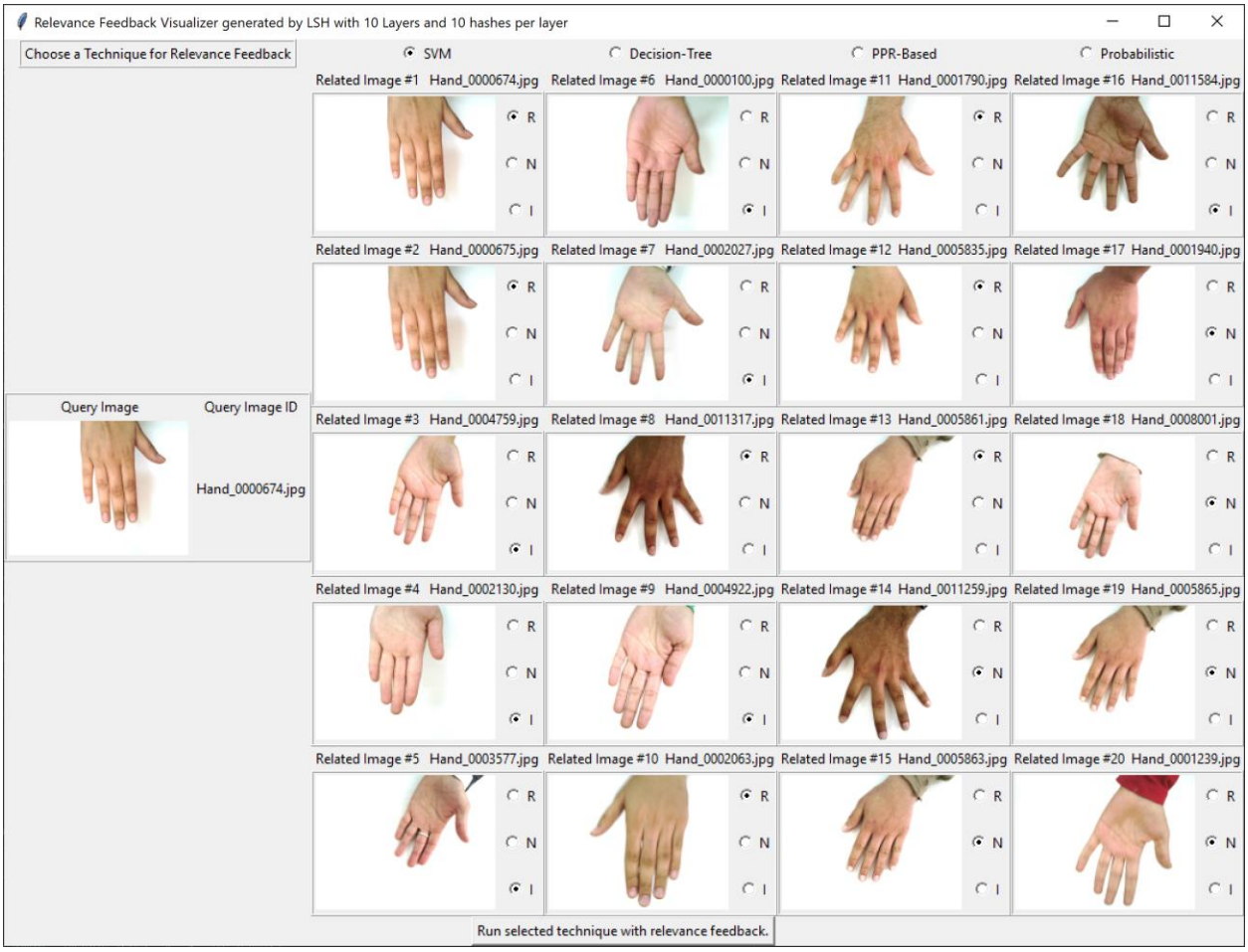
Query 3: L: 5, k: 10, t: 20, Query Image: Hand_0000674.jpg

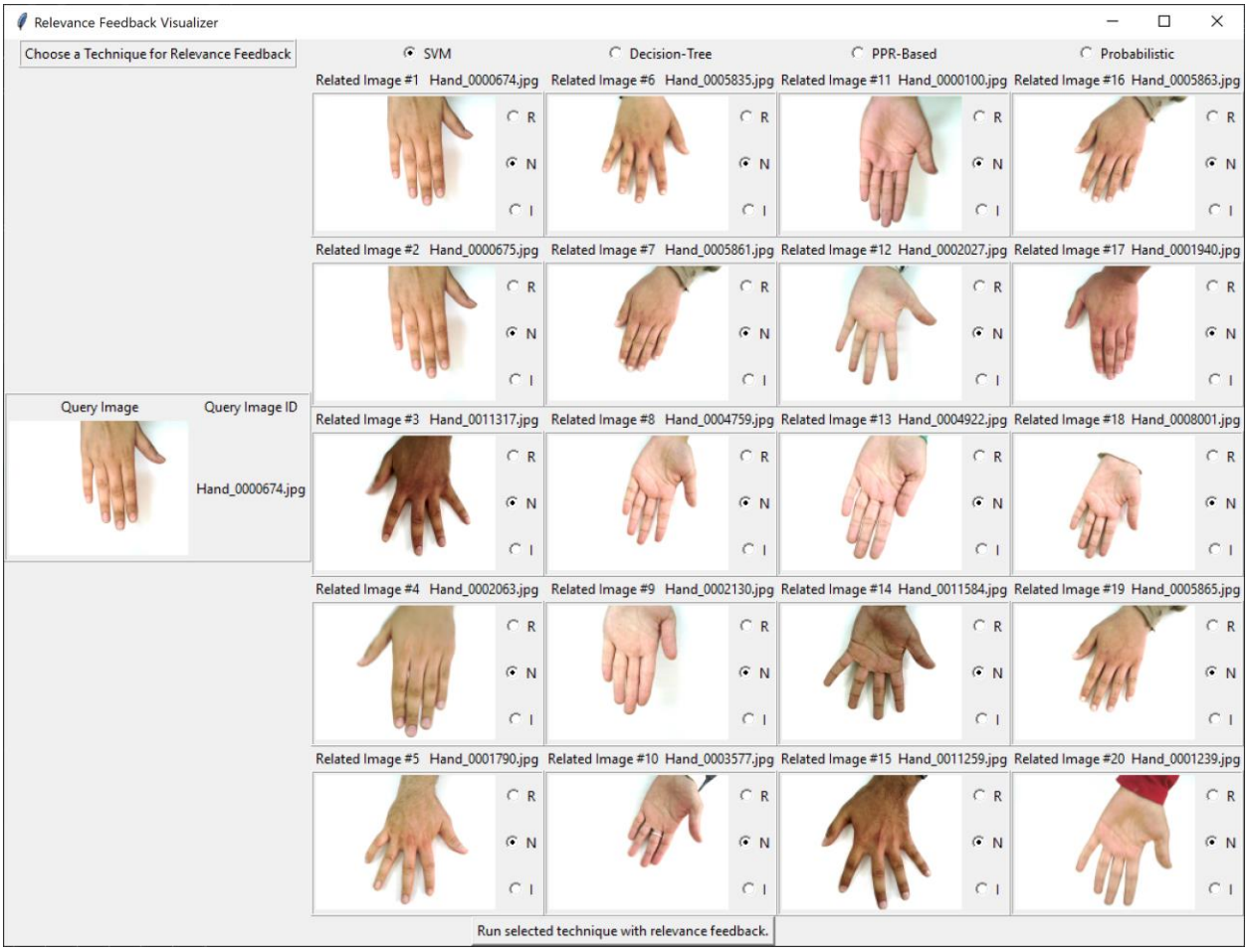


Interpretation: The above images show the top 20 (value of k) similar images to the given query image 'Hand_0000674.jpg'. An index structure using locality sensitive hashing was created and this index structure was then used to extract top 20 similar images. The number of layers have been taken as 5 and the number of hashes have been taken as 10. Total number of overall and unique images have been displayed to the console. The above output shows top 20 similar images from 33 unique images.

7.6 Task 6

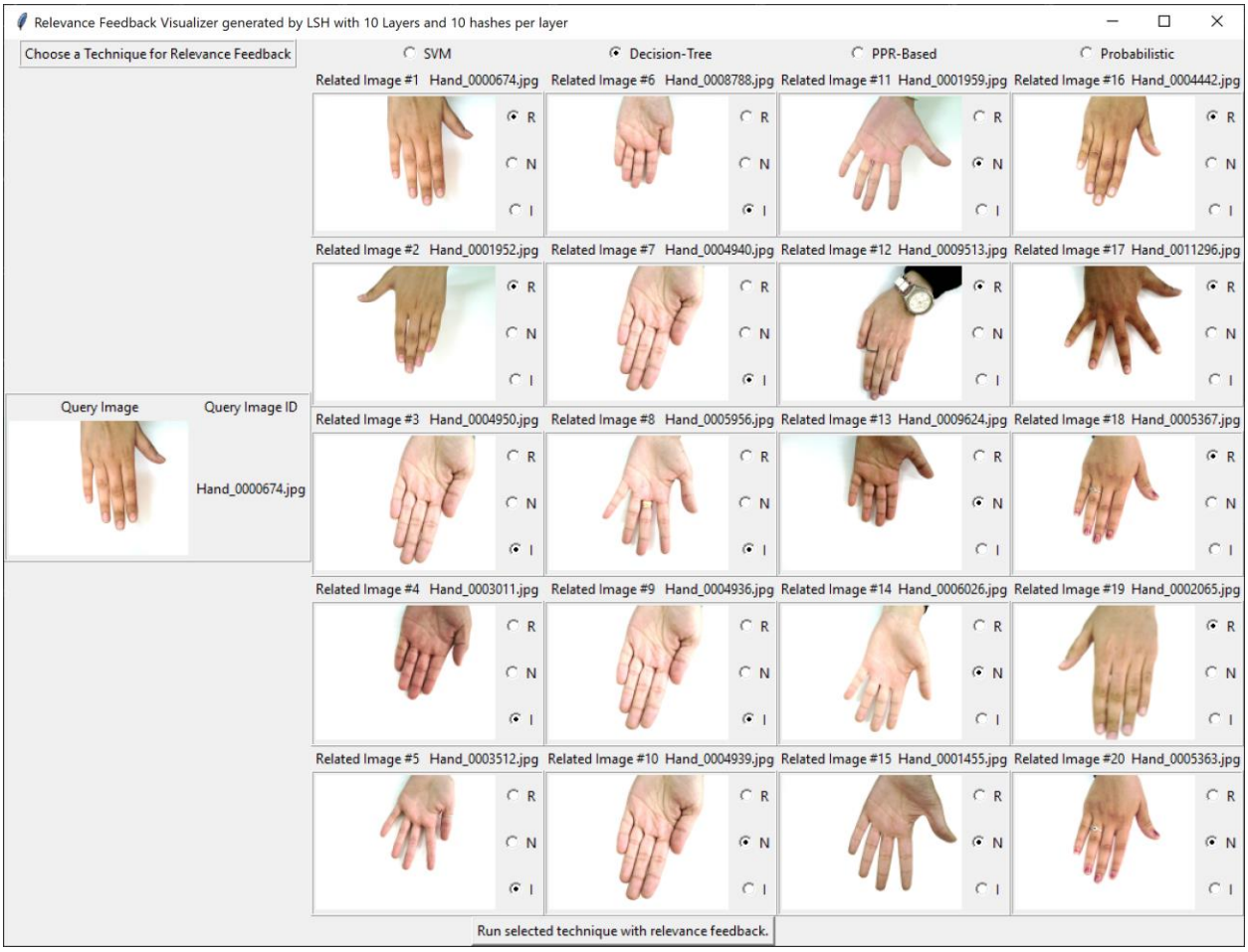
Query 1: Feedback System: SVM based *(Using results from Query 1 of Task 5)*


Relevance Feedback Visualizer generated by LSH with 10 Layers and 10 hashes per layer
Choose a Technique for Relevance Feedback
SVM
Decision-Tree
PPR-Based
Probabilistic
Query Image
Query Image ID
Hand_0000674.jpg
Related Image #1 Hand_0000674.jpg
Related Image #6 Hand_0000100.jpg
Related Image #11 Hand_0001790.jpg
Related Image #16 Hand_0011584.jpg
Related Image #2 Hand_0000675.jpg
Related Image #7 Hand_0002027.jpg
Related Image #12 Hand_0005835.jpg
Related Image #17 Hand_0001940.jpg
Related Image #3 Hand_0004759.jpg
Related Image #8 Hand_0011317.jpg
Related Image #13 Hand_0005861.jpg
Related Image #18 Hand_0008001.jpg
Related Image #4 Hand_0002130.jpg
Related Image #9 Hand_0004922.jpg
Related Image #14 Hand_0011259.jpg
Related Image #19 Hand_0005865.jpg
Related Image #5 Hand_0003577.jpg
Related Image #10 Hand_0002063.jpg
Related Image #15 Hand_0005863.jpg
Related Image #20 Hand_0001239.jpg
R
N
I
Run selected technique with relevance feedback.



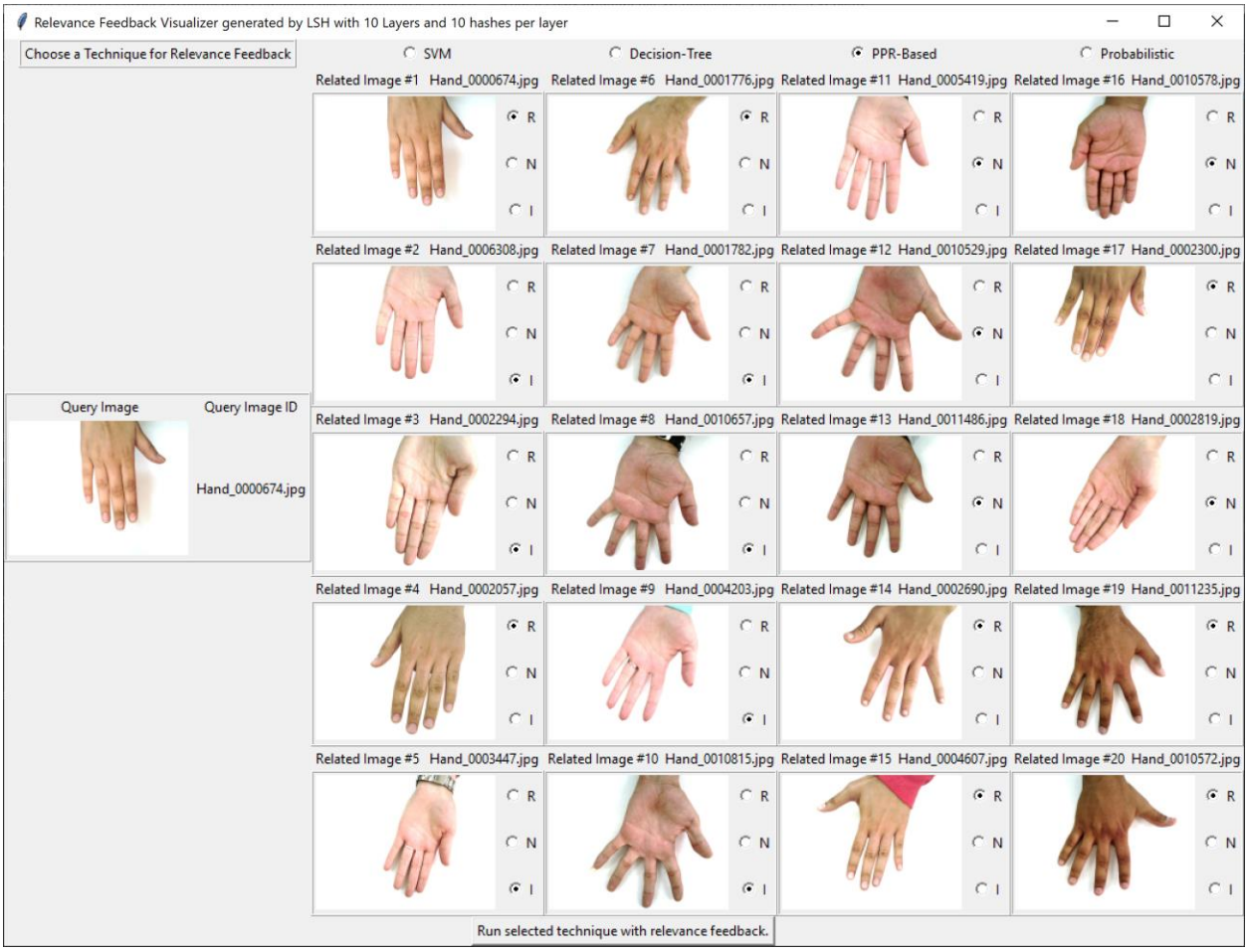
Interpretation: The above results show the relevant images and irrelevant images after the one relevance feedback cycle in the order with respect to the query image used in Task 5b. As we can see the SVM classifier is able to classify the images similar to it as more relevant with higher representation in the visualizer.

Query 2: Feedback System: Decision Tree based *(Using results from Query 1 of Task 5)*


Relevance Feedback Visualizer generated by LSH with 10 Layers and 10 hashes per layer
Choose a Technique for Relevance Feedback
SVM
Decision-Tree
PPR-Based
Probabilistic
Query Image
Query Image ID
Hand_0000674.jpg
Related Image #1 Hand_0000674.jpg
Related Image #6 Hand_0008788.jpg
Related Image #11 Hand_0001959.jpg
Related Image #16 Hand_0004442.jpg
Related Image #2 Hand_0001952.jpg
Related Image #7 Hand_0004940.jpg
Related Image #12 Hand_0009513.jpg
Related Image #17 Hand_0011296.jpg
Related Image #3 Hand_0004950.jpg
Related Image #8 Hand_0005956.jpg
Related Image #13 Hand_0009624.jpg
Related Image #18 Hand_0005367.jpg
Related Image #4 Hand_0003011.jpg
Related Image #9 Hand_0004936.jpg
Related Image #14 Hand_0006026.jpg
Related Image #19 Hand_0002065.jpg
Related Image #5 Hand_0003512.jpg
Related Image #10 Hand_0004939.jpg
Related Image #15 Hand_0001455.jpg
Related Image #20 Hand_0005363.jpg
R
N
I
Run selected technique with relevance feedback.

Interpretation: The above results show the relevant images and irrelevant images after the one relevance feedback cycle in the order with respect to the query image used in Task 5b. As we can see the Decision tree classifier is able to classify the images similar to it as more relevant with higher representation in the visualizer.

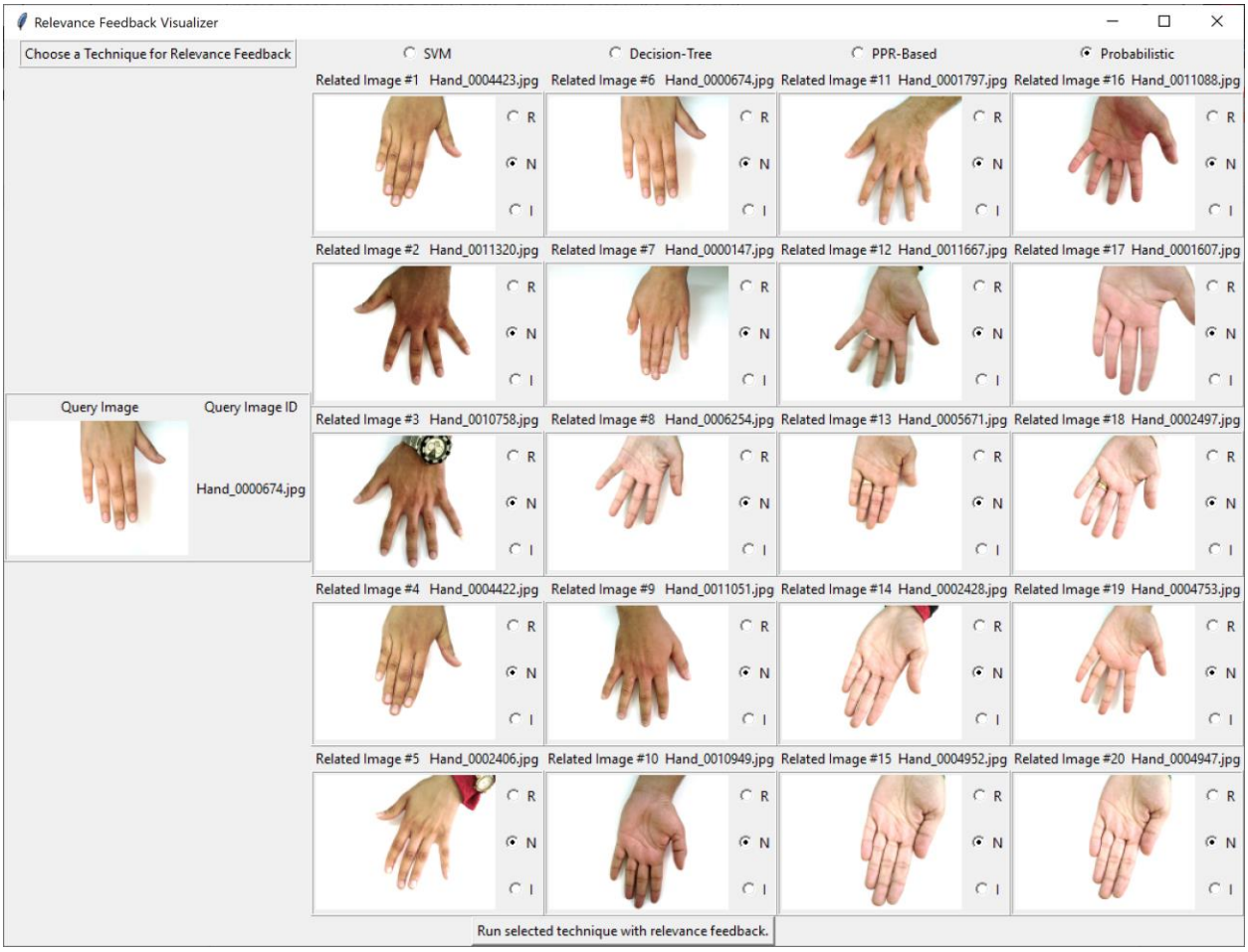
Query 3: Feedback System: PPR based *(Using results from Query 1 of Task 5)*


Relevance Feedback Visualizer generated by LSH with 10 Layers and 10 hashes per layer
Choose a Technique for Relevance Feedback
SVM
Decision-Tree
PPR-Based
Probabilistic
Query Image
Query Image ID
Hand_0000674.jpg
Related Image #1 Hand_0000674.jpg
Related Image #6 Hand_0001776.jpg
Related Image #11 Hand_0005419.jpg
Related Image #16 Hand_0010578.jpg
Related Image #2 Hand_0006308.jpg
Related Image #7 Hand_0001782.jpg
Related Image #12 Hand_0010529.jpg
Related Image #17 Hand_0002300.jpg
Related Image #3 Hand_0002294.jpg
Related Image #8 Hand_0010657.jpg
Related Image #13 Hand_0011486.jpg
Related Image #18 Hand_0002819.jpg
Related Image #4 Hand_0002057.jpg
Related Image #9 Hand_0004203.jpg
Related Image #14 Hand_0002690.jpg
Related Image #19 Hand_0011235.jpg
Related Image #5 Hand_0003447.jpg
Related Image #10 Hand_0010815.jpg
Related Image #15 Hand_0004607.jpg
Related Image #20 Hand_0010572.jpg
R
N
I
Run selected technique with relevance feedback.

Interpretation: The above results show the relevant images and irrelevant images after the one relevance feedback cycle in the order with respect to the query image used in Task 5b. As we can see the PPR based classifier is able to classify the images similar to it as more relevant with higher representation in the visualizer.

Query 4: Feedback System: Probabilistic Relevance based *(Using results from Query 1 of Task 5)*


Relevance Feedback Visualizer generated by LSH with 10 Layers and 10 hashes per layer
Choose a Technique for Relevance Feedback
SVM
Decision-Tree
PPR-Based
Probabilistic
Query Image
Query Image ID
Hand_0000674.jpg
Related Image #1 Hand_0000674.jpg
Related Image #6 Hand_0005671.jpg
Related Image #11 Hand_0000147.jpg
Related Image #16 Hand_0004423.jpg
Related Image #2 Hand_0001607.jpg
Related Image #7 Hand_0002497.jpg
Related Image #12 Hand_0010949.jpg
Related Image #17 Hand_0010758.jpg
Related Image #3 Hand_0004952.jpg
Related Image #8 Hand_0004753.jpg
Related Image #13 Hand_0002428.jpg
Related Image #18 Hand_0006254.jpg
Related Image #4 Hand_0004947.jpg
Related Image #9 Hand_0011667.jpg
Related Image #14 Hand_0002406.jpg
Related Image #19 Hand_0001797.jpg
Related Image #5 Hand_0011088.jpg
Related Image #10 Hand_0011320.jpg
Related Image #15 Hand_0004422.jpg
Related Image #20 Hand_0011051.jpg
R
N
I
Run selected technique with relevance feedback.



Interpretation: The above results show the relevant images and irrelevant images after the one relevance feedback cycle in the order with respect to the query image used in Task 5b. As we can see the Naive Bayes(Probabilistic) classifier is able to classify the images similar to it as more relevant with higher representation in the visualizer.

8 Conclusion

In this phase of the project, we had to work with PPR, LSH and also we had to implement the SVM classifier, PPR-based classifier, decision tree, and SVM. Some of the key takeaways from this phase of the project were that representing the multimedia objects in the form of graph structures really helps in the process of analysis. The analysis of large graphs using PPR helps in measuring the relative proximity of vectors in the graph. We also learnt about using LSH in making the fast memory index structure and using its application of duplicate detection to find the most similar images. In this phase, we were able to handle the complete 11k dataset successfully. After facing initial problems with handling such a huge dataset, we were able to come up with appropriate techniques for solving each task in the final phase. We got an opportunity to experiment with and implement various types of classification techniques. We analyzed the accuracy of each type of classifier and also brainstormed different ways on improving the performance of the classifier. Overall, this phase taught us about implementing indexing, clustering and classification techniques from scratch. We dived into the basic concepts and we were able to implement all the tasks in the final phase successfully.

9 References

[1] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In KDD, pages 653658, 2004

[2] Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions" (by Alexandr Andoni and Piotr Indyk). Communications of the ACM, vol. 51, no. 1, 2008, pp. 117-122.

[3] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science. 41, pp. 288-297, 1990.

[4] Efficient near-duplicate Detection and Sub-image Retrieval, Yan Ke, Rahul Sukthankar, Larry Huston.

[5] Duplicate Detection for Identifying Social Spam in Microblogs, Qunyan Zhang, Haixin Ma, Weining Qian, Aoying Zhou Center for Cloud Computing and Big Data, Software Engineering Institute, East China Normal University, Shanghai, China.

[6] SVM Implementation from Scratch using SMO Algorithm, Sai Srinadhu K, IIT Ropar.

10 Appendix

Specific roles of team members.

1.Anjali: LSH, Testing, Documentation

2.Athul: LSH, Testing, Documentation

3.Manoj: Clustering, Probabilistic classifier, Testing, Documentation

4.Md Shadab: SVM, Decision tree classifier, Task 1 classification, Testing, Documentation

5.Prashant: PPR classifier and relevance feedback, LSH, Testing, Documentation

6.Tyler: Visualization Creation, Probabilistic classifier, Testing, Documentation