**CSE 515: Multimedia and Web Databases**

**Phase 3**

**Group 17**Aaryan Gupta  
Fenil Madlani  
Krisha Gala  
Pranav Katariya  
Radhika Ganapathy  
Shivam Malviya

**Abstract**

In the final phase of this project, we perform image indexing using various clustering and classification algorithms such as Support Vector Machine, Personalized Page Rank and Decision Tree algorithms. Index structures were built using algorithms like Locality-Sensitive Hashing and VA-Files to carry out similar image search. In addition to this, we worked on relevance feedback systems to improve the nearest neighbour matches which helps account for the relevant and irrelevant results produced. Lastly, a query interface was built for the user to smoothly enter the query, retrieve results and give feedback which then produced revised results.

***Keywords—*** image features, image retrieval, indexing, vector models, classification, SVM, Decision Tree, Personalized Page Rank, clustering, LSH, VA Files, relevance feedback.

**1 Introduction**

* 1. **Terminology**
     1. **SVM**
     2. **DT**
     3. **PPR**
     4. **LSH**
     5. **VA Files**
     6. **Relevance Feedback**
  2. **Goal Description**

**Task 1:**

In this task, we have to implement a program in which we are given a folder of images, one of the three feature models, and a user specified value of k. We have to compute k latent semantics (if not already computed and stored), and given a second folder of images, it should associate **X** labels to the images in the second folder using the classifier selected by the user. The classifiers to be implemented are - SVM classifier, Decision-Tree classifier, Personalized Page Rank. We also have to compute and print false positive and miss rates.

**Task 2:**

In this task, we have to implement a program in which we are given a folder of images, one of the three feature models, and a user specified value of k. We have to compute k latent semantics (if not already computed and stored), and given a second folder of images, it should associate **Y** labels to each image in the second folder using the classifier selected by the user. The classifiers to be implemented are - SVM classifier, Decision-Tree classifier, Personalized Page Rank. We also have to compute and print false positive and miss rates.

**Task 3:**

In this task, we have to implement a program in which we are given a folder of images, one of the three feature models, and a user specified value of k. We have to compute k latent semantics (if not already computed and stored), and given a second folder of images, it should associate **Z** labels to each image in the second folder using the classifier selected by the user. The classifiers to be implemented are - SVM classifier, Decision-Tree classifier, Personalized Page Rank. We also have to compute and print false positive and miss rates.

**Task 4:**

In this task, we have to implement a Locality Sensitive Hashing (LSH) tool, which takes as input the number of layers (L), the number of hashes per layer (κ) and a set of vectors (generated by other tasks) and creates an in-memory index structure containing the given set of vectors. In addition to this, we have to implement similar image search using this index structure. Given a folder of images and one of the three feature models, the images are stored in an LSH data structure (the program also outputs the size of the index structure in bytes). For any given image and “t”, the index tool returns the “t” most similar images. The program should also return the number of buckets searched, the unique and overall number of images considered, false positive and miss rates.

**Task 5:**

In this task we have to implement a VA-file index tool and conduct nearest neighbour search operations. We will be given a parameter “b” denoting the number of bits per dimensions used for compressing the vector data and a set of vectors (generated by other tasks) as input. The program creates an in-memory index structure containing the indexes of the given set of vectors. The program also returns the size of the index structure in bytes. We then have to implement similar image search using this index structure. Given a folder of images and one of the three feature models, the images will be stored in a VA-file data structure (the program also outputs the size of the index structure in bytes). For any given image and “t”, the tool returns the “t” most similar images. The program should also return the number of buckets searched, the unique and overall number of images considered, false positive and miss rates.

**Task 6:**

In this task we have to implement a Decision-Tree based relevance feedback system to improve nearest neighbour matches. This enables the user to label some of the results returned by the search task as relevant or irrelevant and then returns a new set of ranked results, either by revising the query or by re-ordering the existing results.

**Task 7:**

In this task we have to implement a SVM based relevance feedback system to improve nearest neighbour matches. This enables the user to label some of the results returned by the search task as relevant or irrelevant and then returns a new set of ranked results, either by revising the query or by re-ordering the existing results.

**Task 8:**

In this task we have implement a query interface. This allows the user to provide a query and relevant query parameters (including how many results to be returned). Query results are presented to the user in decreasing order of matching. The result interface should also allow the user to provide positive and/or negative feedback for the ranked results returned by the system. User feedback is then taken into account and a new set of ranked results are returned.

* 1. **Assumptions**

All latent semantic files generated in the previous phase are rigtht. SVD, LDA, Kmeans, PCA implementations are producing the right results.

Any other assumptions made???

1. **Description of Proposed Solution and Implementation**
2. **Interface Specification**
3. **System Requirements and Execution Instructions**
   1. **Environment Set up**
   2. **Execution**
4. **Related Work**
5. **Conclusions**
6. **Bibliography**
7. **Appendix**