CBIR using Image Ranking and BOW

Ismail El-Sayad[#1], Samih Abdul-Nabi[#2], Hussien Kassem[#3,] Maher Santina[#4], Jamal Mansour [#5]

[#] Lebanese International University, Beirut, Lebanon

ABSTRACT

We enhance CBIR system using user Feedback and BOW representation. This enhancement included the ability to save a new ranking chosen by a number of users for the top relevant images. This was done by changing the representations of images using the user ranking. The enhancement can be updated every time users choose to, which, to a certain degree, increases further the precision of the image ranking.

Keywords: **BOW, CBIR, SURF, Image Retrieval, and Relevance Feedback.**

I. INTRODUCTION

Content-Based Image Retrieval [1,2] is the ability to retrieve an image based on its content (Ex: colors, features, shapes etc...). Image retrieval problems are arising throughout the years due to the growing number of images in databases and the need to retrieve an image based on its content. This problem cannot be solved by Concept-Based Image Retrieval that relies on meta-data to identify a picture (Ex: Title, Indexing, Name etc…).

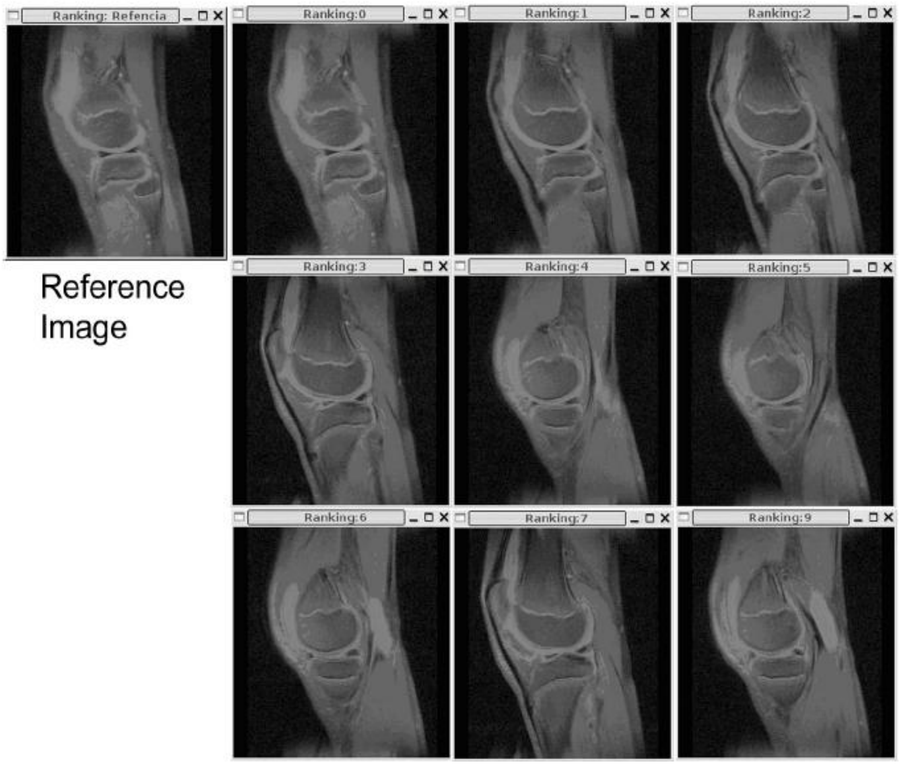


Figure 1: CBIR in Medical Lab

Google, for example, has a mechanism to get images similar to the image supplied by the user. This solution solves many limitations found in Concept Based Retrieval, but it also opens the door for more challenges. This technique is commonly needed in different fields nowadays like in Medical Applications, weather forecasting and others.

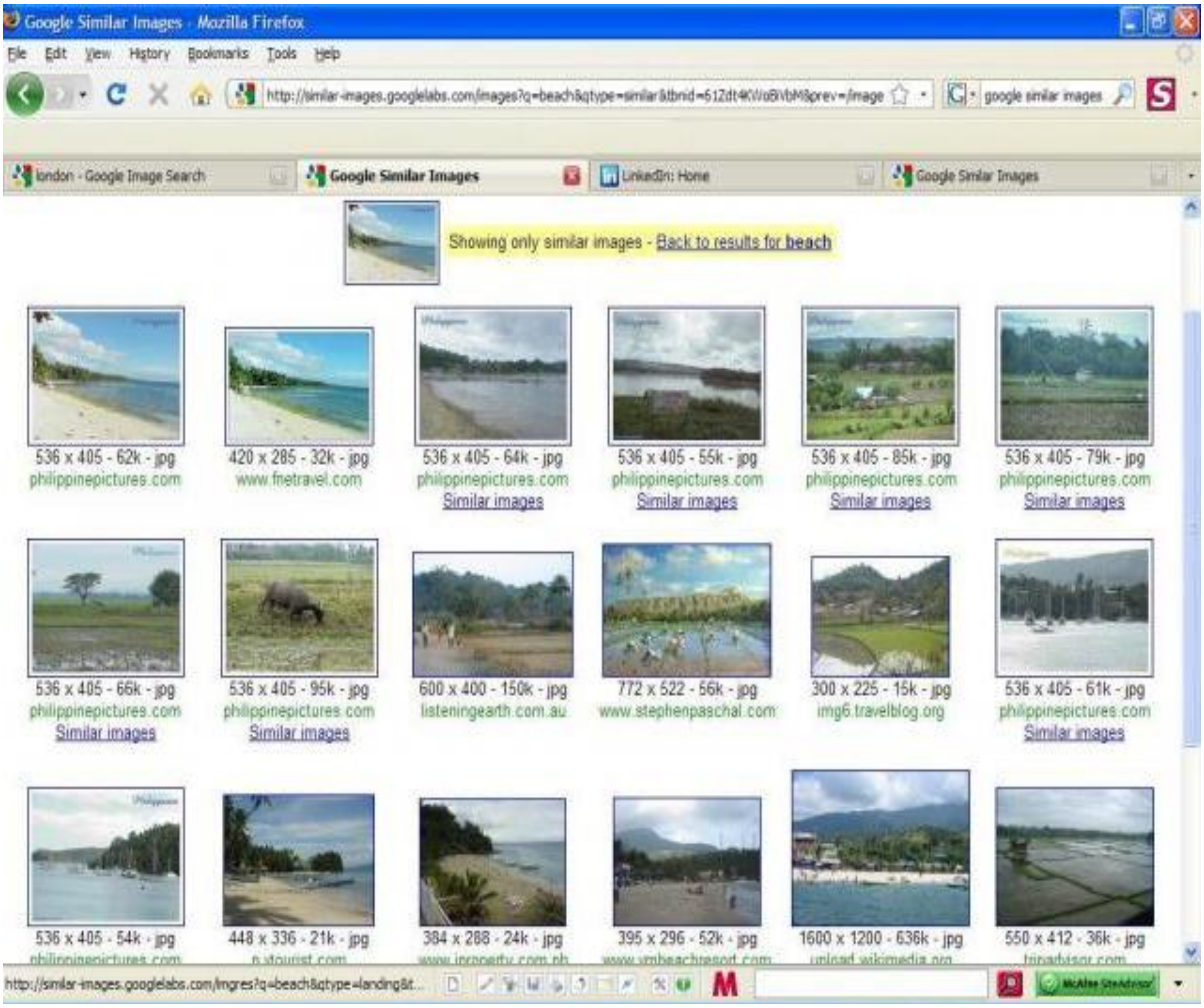


Figure 2: Google CBIR

The challenge to image indexing is studied in the context of image databases. As mentioned before image indexing has become one of the promising and important research areas for researchers from a wide range of application disciplines. These disciplines include art galleries, museums, archaeology, architecture design, geographic information systems, weather forecast, medical imaging trademark databases etc…

CBIR includes methods in Digital Signal Processing, Data-Mining, Mathematics and Programming. Hence, Mathematicians, Scientists and Engineers are working together to improve this domain. Therefore, a designer can combine a lot of techniques in several domains to create a CBIR system that matches his/her needs.

Previously, systems used low level image retrieval techniques such as texture or color histograms. But it was noticed that these techniques are insufficient and impractical for large databases and more precise image retrieval.

The goal of this project is to create a CBIR system based on Relevance feedback. This system will adapt an algorithm that uses low-level features (discussed below) such as “Surf” [3,4,5] and models such as “Bag of Words” in addition to **feedback** [5,6] from the user to improve the quality of the retrieval.

However, pure calculations of component weights don’t give satisfactory results since the interpretation of the image differs from one user to another. The meaning of the image depends on the personality of each user, plus the likes and dislikes etc...Therefore, the unwanted dimensions are determined by the user feedback. Receiving the ranking from several users would aid in searching for the dimensions that are least important.

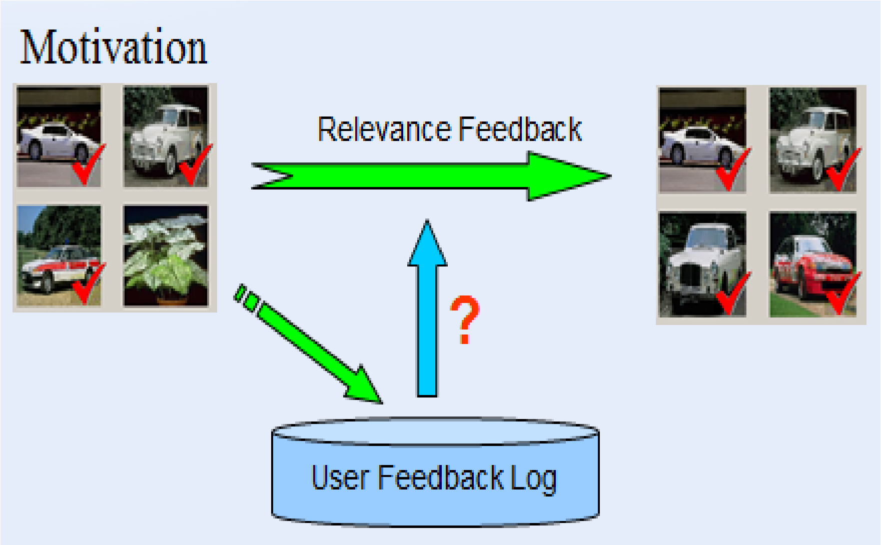


Figure 3: Enhancing image retrieval with User Feedback

This system will use mid-level algorithms such as “Surf” and models such as “Bag of Words”. In addition, we will use feedback from the user to improve the quality of the retrieval. The feedback from the user will point out which dimensions are least important.

II. PROPOSED METHODOLOGY

Figure 4 shows an overview of the system we designed. It’s clear that the system needs two phases, the Training Phase and the Testing Phase. Each Phase will be discussed in this section.

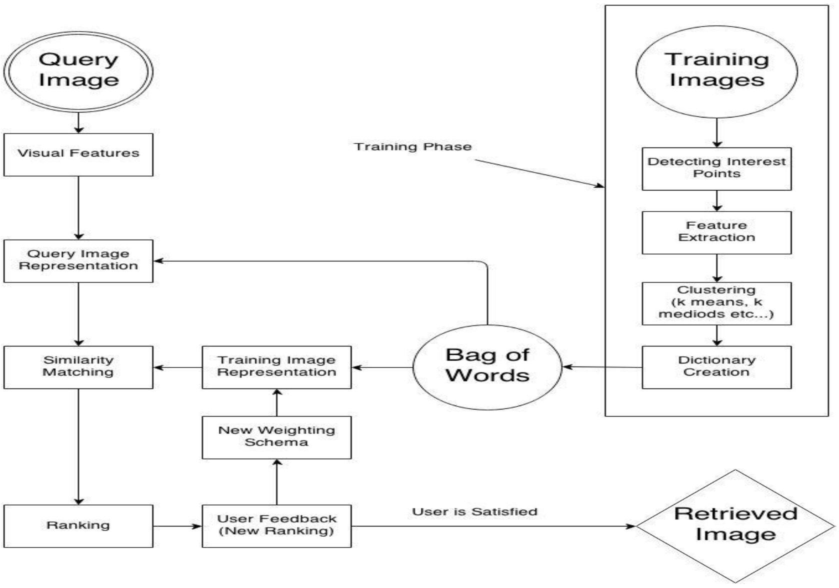


Figure 4: System Design

Training Phase

Any image retrieval system requires a training phase which needs a predefined library of images. The set of images will be split into two categories, training images and query images. Only training images will be used in this phase.

First of all, the system will extract interest points from the training images which are likely candidates to have useful information. Then, the information at each interest point will be reduced using a dimensionality reduction technique and only important features are extracted. Later on, the features that are closely related to each other will be clustered into groups. Therefore, each cluster will represent a group of features that are highly related to each other. Finally, a dictionary of visual words is created such as the index of each word represents the index of each cluster. After this step, a Bag of Words will be available that contains all sets of features that will be used to represent all images.

Testing Phase

At this point, a Dictionary of words is previously created. Each new query image given will undergo visual feature extraction. This helps in representing this image using the words available.

Now, both training images and the query image are represented using the created dictionary, so similarity matching can be feasible. The query image will be compared to all images in the collection and the distance between the query image and each training image is calculated. The system will output only the top ten images which have the least distances with the query image. The outputted images will be sorted in increasing distance which will be the system ranking of this query. If the user is not satisfied with this ranking he/she can give a feedback with a new ranking. We have created a new weighting algorithm which involves an equation that gives different priorities to words based on the ranking. The system will modify the representation and save it if the user desires to do a new query based on this new representation.

We have also implemented the ability for several users to input their feedback to the system, and it will calculate the total voting of the users and use it to enhance the representation.

Implementation

Figure 5: TOP-SURF

To implement our project we used the open source library TOP-SURF [7, 8] that handles low level image processing, feature extraction and comparison. Our contribution is that we added a user feedback system. This system aids in understanding which features are important to the user and helps create a modified representation. This modification is saved in a text file if the user wants to do a new query with this representation.

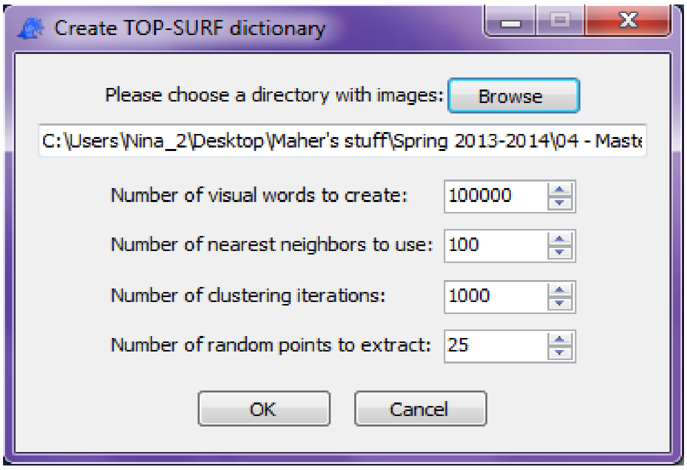
The collections of images are taken from Caltech101 that has a wide variety of sets of images majorly used in computational vision. We have tested the system on several numbers of images and several different sets. The results will be demonstrated in the next chapter.

Create Dictionary, Save Dictionary, Load Dictionary

The first step in the procedure is to initialize a dictionary that will contain all the visual words. TOP-SURF gives us the ability to create a dictionary from a directory that contains all the training images. This dictionary will be saved in memory throughout the retrieval process and will be deleted after program termination.

Therefore, there are two options that allow us to save and load a dictionary for later use.
As shown in Figure 6, Create a Dictionary option requires few parameters such as:

- Number of visual words to create: The number of words that will be saved to the dictionary.
- Number of nearest neighbors to use: The number of inspected neighbors in each cluster
- Number of clustering iterations: Each clustering algorithm becomes more precise after each clustering iteration.
- Number of random points to extract: number of interest points to extract for clustering



Create TOP-SURF dictionary

Please choose a directory with images: Browse

C:\Users\Nina_2\Desktop\Maher's stuff\Spring 2013-2014\04 - Maste

Number of visual words to create: 100000

Number of nearest neighbors to use: 100

Number of clustering iterations: 1000

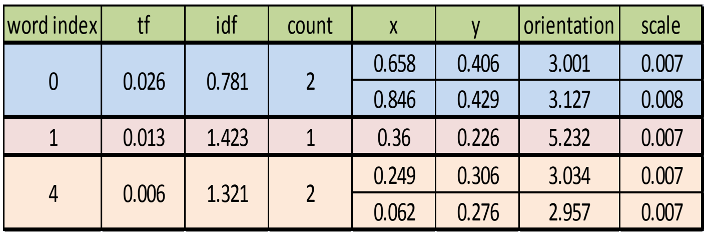
Number of random points to extract: 25

Figure 6: Create the Dictionary

TOP-SURF will automatically run through all images in the directory, extract visual words and save them to the dictionary.

Extract Descriptors, Save Descriptors, Release Descriptors

TOP-SURF already has a mechanism to extract descriptors from images. The output of this phase will be a list of visual words (contained in the image) and all the details related to them (location, index, tf, idf etc...). The succeeding table is an example about this information:



word index	tf	idf	count	x	y	orientation	scale
0	0.026	0.781	2	0.658	0.406	3.001	0.007
				0.846	0.429	3.127	0.008
1	0.013	1.423	1	0.36	0.226	5.232	0.007
4	0.006	1.321	2	0.249	0.306	3.034	0.007
				0.062	0.276	2.957	0.007

Figure 7: Extracted Descriptors

- Word index: the index that is equal to the index of the cluster of the corresponding feature.
- TF: Term Frequency, which is the number of times the word exists with respect to the whole words in the image.
- IDF: Inverse Document Frequency, which is a measure of how much information the word provides, obtained by dividing the total number of images over the images that contain this word.
- Count: how much the word is repeated in the image.
- X, Y, Orientation, Scale: the x-y coordinates of each word, its size and direction.

Visualize Descriptors

TOP-SURF has an option that can show us the location, orientation and scale of words on the image itself. Figure 8 shows an example of that:

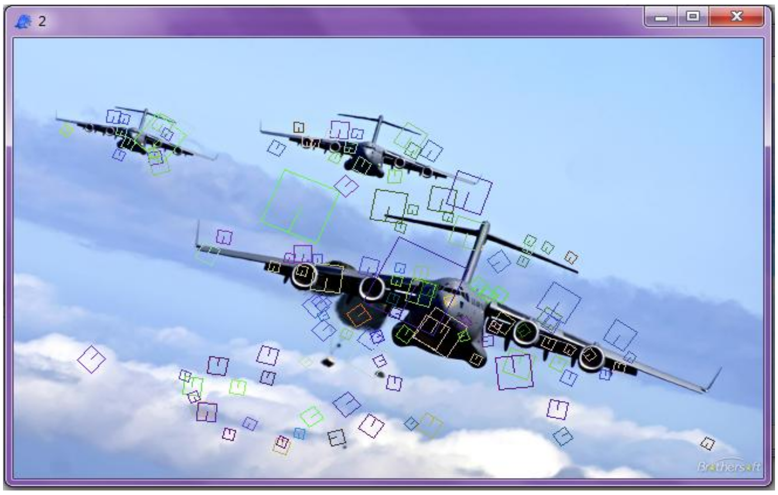


Figure 8: Visualize Descriptors

Compare Descriptors

The most important step in the image retrieval system is comparing the representations of the query image with all the training images and output the top ten images that have the least distance with the query image. Figure 9 will give an example about the comparison.

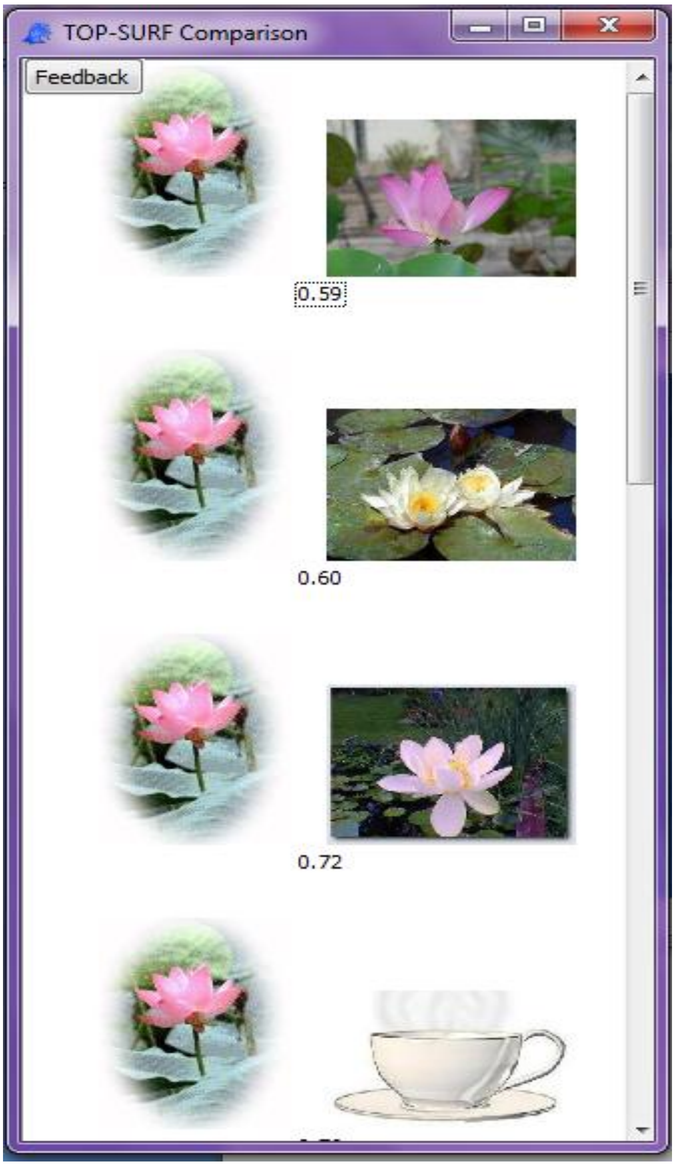


Figure 9: Compare Descriptors

There are two modes of comparison:

- Cosine Normalized Difference
- Absolute Difference

These algorithms mainly check similar word identifiers (indices) and multiply their tfs with their idfs and add the values to a distance variable.

Therefore, the distances shown in figure 9 resemble how much the words in the query image and training image are closely related to each other.

User Feedback

To this point, all methods discussed were implemented by TOP-SURF developers. However, we added and edited several methods in the program to include user feedbacks that modify image representations as shown in figure 10.

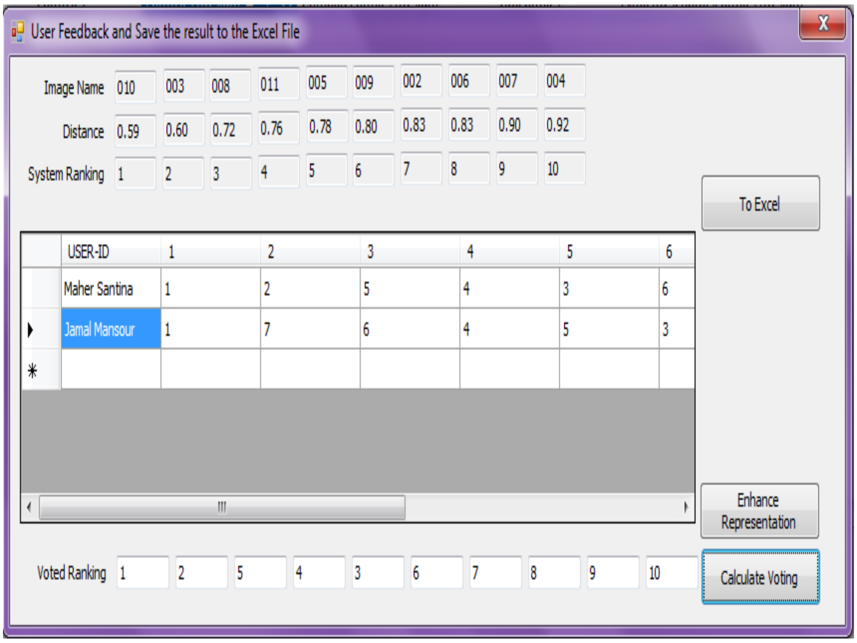


Image Name	010	003	008	011	005	009	002	006	007	004
Distance	0.59	0.60	0.72	0.76	0.78	0.80	0.83	0.83	0.90	0.92
System Ranking	1	2	3	4	5	6	7	8	9	10

	USER-ID	1	2	3	4	5	6
	Maher Santina	1	2	5	4	3	6
▶	Jamal Mansour	1	7	6	4	5	3
*							

Voted Ranking	1	2	5	4	3	6	7	8	9	10

Figure 10: User Feedback

The image Name, Distance and System Ranking are read only number since they hold information about the top ten retrieved images and will be used to organize the voting procedure.

We added a part of an excel sheet in this dialog to hold the feedbacks of each user.

As shown in the figure each row holds the user's ID and the voted ranking of each image. The "To Excel" button saves all user feedbacks in an excel file of our choice. We can add later on the ability to import feedbacks from an existing file. "Calculate Voting" and "Enhance Representation" are discussed in details below.

Calculate Voting

As soon as the user feedbacks are inputted in the excel sheet, "Calculate Voting" can be pressed to calculate the overall voting of all users.

The voting procedure is calculated as follows:

3 Each column of user voting is taken and processed alone
For example, the first array of voting corresponds to the ranks of image 1 chosen by each user.
4 The count of each rank is stored in the array
For example, in figure 27, the first array will have 2 as the count for number "1" and all other counts will be zeros.
5 In each array, the number with the highest count is chosen as the new ranking
6 If there are two equal counts always pick the smaller rank
7 No overall rank can be used twice for two different images

Enhance Representation

This step is the core of our contribution. However, enhancing the representation only works if the overall voting is previously calculated. It works as follows:

Calculate a weight for each word existing in the query image according to this equation:

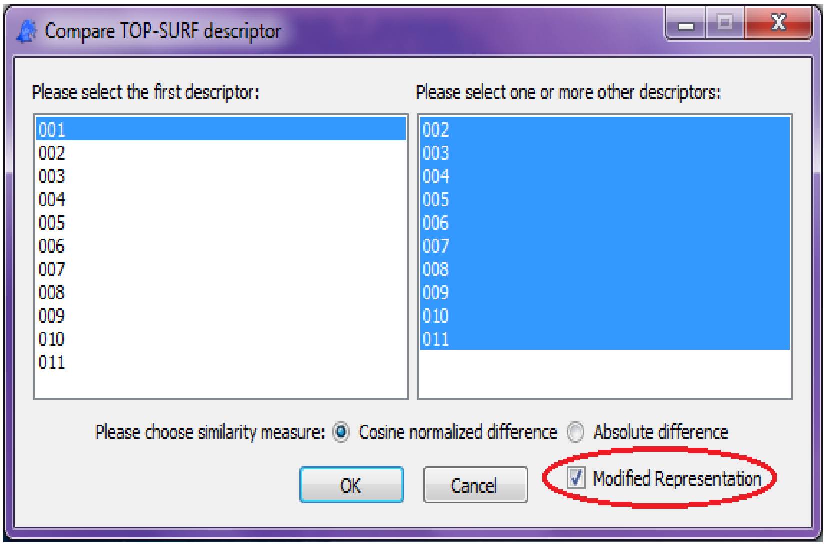
$$weight(word\ x) = \sum_{r=1}^{n}(n+1-r) * count(word\ x, pic\ r) \quad (1)$$

Where:

r is the voted rank of each picture
n is the total number of images

Get rid of all the words that have 0 weight
Remove 5 words that have the least weights
Save remaining words into a text file for later use

After enhancing the representation, comparing descriptors can be done again but with the modified representation by checking the “Modified Representation” checkbox as shown in figure 28.



Compare TOP-SURF descriptor

Please select the first descriptor:

001
002
003
004
005
006
007
008
009
010
011

Please select one or more other descriptors:

002
003
004
005
006
007
008
009
010
011

Please choose similarity measure: ● Cosine normalized difference ○ Absolute difference

☑ Modified Representation

Figure 11: Comparing Descriptors with Modified Representation

After comparing descriptors again, the system ranking of the images will change. Some images might disappear from the top ten, and new images might appear.

The results in the next chapter will show the degree of enhancement of image retrieval after user feedback.

When the user checks “Modified Representation” checkbox, the built-in method that extracts descriptors will be disabled and our version of the method will take all details of the remaining visual words from the text file. Those words will be compared to the words in the training images and the output will be displayed.

III. RESULTS AND DISCUSSION

As we discuss previously, our new technique to enhance query image representation to retrieve more relevant images. We use two approached: Bag of visual words (BOVW) for image retrieval, and the second approach (Our New Technique) is called Enhanced Bag of visual words Using User Feedback (EBOVWUUF) for image retrieval. We apply EBOVWUUF for two rounds of feedback: EBOVWUUF Technique 1 (feedback round 1) and EBOVWUUF Technique 2 (feedback round 2). We apply these techniques by using three different databases: Animals Database, Vehicles Database, and Mix Database (Animals and Vehicles) [9]. Also, it is shown how our new technique (EBOVWUUF) contributes in enhancing query image representation by retrieving more relevant images.

Figure 12 shows chart which represents Mean Average Precision by using BOVW and EBOVWUUF Techniques for animal Database.

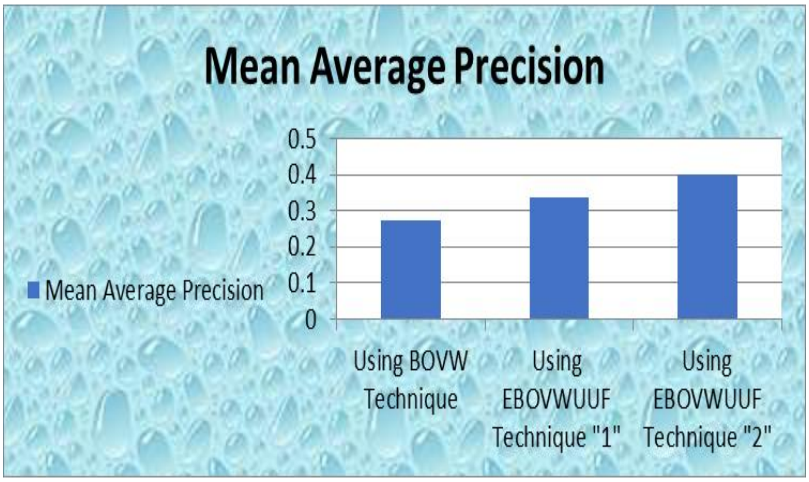


Figure 12: Chart of Mean Average Precision for animal database.

Figures 13 Chart which represents Mean Average Precision by using BOVW and EBOVWUUF Techniques for vehicles Databases.

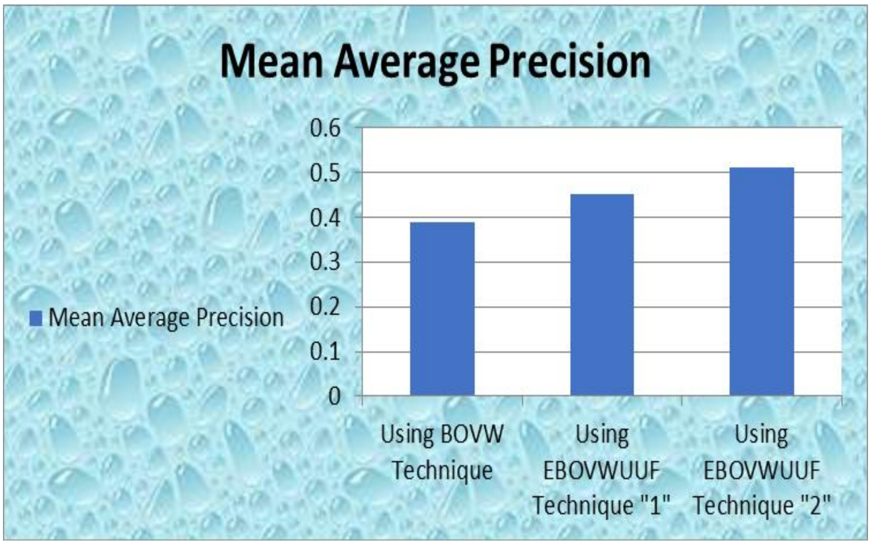


Figure 13: Chart of Mean Average precision for vehicle database

Figures 14 Chart which represents Mean Average Precision by using BOVW and EBOVWUUF Techniques for vehicle Databases.

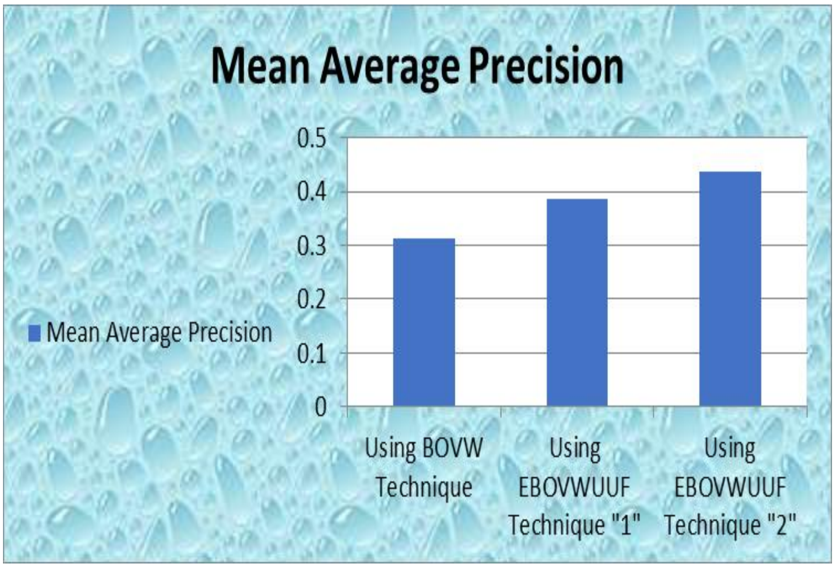


Figure 14: Mean average precision for the mixed database

CONCLUSION

After that, we give a detailed explanation for our system design and implementation. Our contribution is stated in details which involve user feedback and user ranking in order to enhance the BOW representation and to be more oriented to semantics. We test the system on different databases which hold different categories and we state the different results.

Due to the lack of scalability of enhancement techniques, our future work will include another enhancement technique that can be used with the already implemented one. Both techniques can be used progressively or together, and with trial and error, we can figure out the best technique combination for each image set.

Moreover, we can launch the system online so it would be an actual image retrieval software where users can upload images and retrieve the top relevant ones. Users can also provide their personal ranking and according to that the system updates the representation of images. Noisy data, huge amount of traffic and large number of images in database should be handled to maintain the functionality of the system.

REFERENCES

[1]. Liu, Y., Zhang, D., Lu, G., & Ma, W. Y. (2007). A survey of content-based image retrieval with high-level semantics. Pattern recognition, 40(1), 262-282.

[2]. Sykora, P., Kamencay, P., & Hudec, R. (2014). Comparison of SIFT and SURF methods for use on hand gesture recognition based on depth map. AASRI Procedia, 9, 19-24.

[3]. Su, J. H., Huang, W. J., Philip, S. Y., & Tseng, V. S. (2011). Efficient relevance feedback for content-based image retrieval by mining user navigation patterns. IEEE transactions on knowledge and data engineering, 23(3), 360-372.

[4]. Fodor, I. K. (2002). A survey of dimension reduction techniques. Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 9, 1-18.

[5]. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[6]. Hoi, S. C., Lyu, M. R., & Jin, R. (2005, April). Integrating user feedback log into relevance feedback by coupled svm for content-based image retrieval. In Data Engineering Workshops, 2005. 21st International Conference on (pp. 1177-1177). IEEE.

[7]. Thomee, B., Bakker, E. M., & Lew, M. S. (2010, October). TOP-SURF: a visual words toolkit. In Proceedings of the 18th ACM international conference on Multimedia (pp. 1473-1476). ACM.

[8]. Bradski, G., & Kaehler, A. (2000). OpenCV. Dr. Dobb's journal of software tools.

[9]. Griffin, G., Holub, A., & Perona, P. (2007). Caltech-256 object category dataset.