Journal Homepage: - www.journalijar.com
INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)
ISSN NO. 2320-5407
Article DOI: 10.21474/IJAR01/7342
DOI URL: http://dx.doi.org/10.21474/IJAR01/7342

RESEARCH ARTICLE

A NEW WEIGHTING SCHEME FOR CONTENT-BASED IMAGE RETRIEVAL IN THE MULTIMODAL INFORMATION SPACES.

Ismail El Sayad, Samih Abdul-Nabi, Hussien Kassem, Georges Moubarak and Ahmad Saleh.
Lebanese International University, Beirut, Lebanon.

Manuscript Info

Manuscript History

Received: 03 May 2018
Final Accepted: 05 June 2018
Published: July 2018

Keywords:-
Image Retrieval, Image Processing, Bag of visual words, Virus, Text annotation, cross modality, semantic analysis.

Abstract

In content based image retrieval, many researchers have worked to improve image retrieval results. Many papers where written to address that problem in many ways. Most papers have worked with how to represent the image in order to find a match to it. Some worked in representing the image using Scale-invariant feature transform (SIFT) descriptors other using Speeded up Robust Features (SURF) or representing it as a set of visual words or many other representations. But all these methods relied purely on representing the image visually without any reference to the semantic of the image. Our object was to introduce the image semantics to the retrieval by combining textual annotation with the visual representation to give a refined representation of the image that will improve retrieval results. This report is divided mainly into two parts. In the first part we introduce a literature review about content based image retrieval and annotation based image retrieval. In the second part, we introduced our own methodology and backed it up with tests and results. Our approach is evaluated over 14 categories with each containing example image(s) and annotation statements.

Introduction:-

Annotation based image retrieval (ABIR) is when queries are texts and targets are images. Normally, images are provided with few words as annotation that are not sufficient for retrieval because, for example, one can refer to the image of a house as “home” and another can refer to it as “house”, these two words are not similar, hence if a user searches for images of a “house” he wouldn’t get any image of “home”. Thus a thesaurus is needed to expand the terms and perform the retrieval. A thesaurus is a reference that lists words and groups them according to their similarity in meaning. Images are represented by either global or local features. Global features are capable of generalizing an entire image with a single vector, describing colour, texture, or shape. Local features are computed at multiple points of interest on an image and are capable of recognizing objects. Zhou et al. [1] suggested that Content Based Image Retrieval (CBIR) is limited because it relies solely on low-level visual features.

They proposed the use of textual information within the CBIR framework. They also mentioned the problem of word sparseness. They used relevance-feedback (RF) for estimating word associations in annotated images. [2]Masashi Inoue and Naonori Ueda [3] used, query expansion. In QE the words that are related to the original query words are added to the query.

Corresponding Author:- Ismail El Sayad.
Address:- Lebanese International University, Beirut, Lebanon.

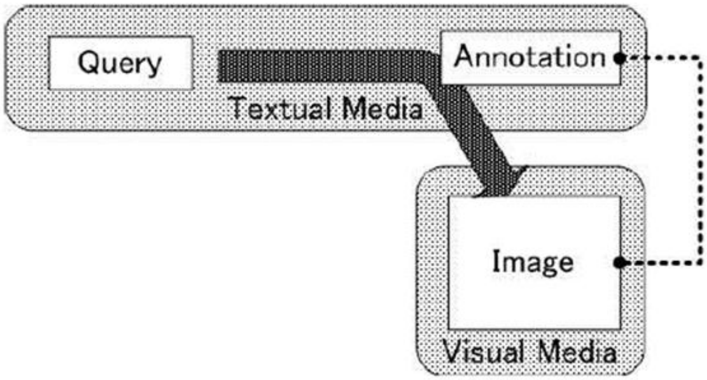


Figure 1:- Annotation-based query-by-text image retrieval. [3]

To apply QE they needed a thesaurus, in particular they used WordNet to create associations between words. As seen in Figure 1, the query provided is a text as the annotation of the image. For image retrieval, when a user provides a query to the system, it ranks all documents according to their relevance to that query. Such a model is called a language model (LM), and it is frequently used for the textual information retrieval. The problem of LM-based information retrieval models is that the likelihood of a document will be zero unless the document contains the query words. That is, LM-based IR is only capable of term-by- term matching.

However, users often want documents to be retrieved that contain semantically similar words that are not the query words themselves. Thus, the system should be capable of associating the query words to the other words. For this purpose, they adopted one successful modification of the LM based IR model called the statistical translation model (STM). The STM incorporates the knowledge of word associations and associates the query word with the document word. Similarly to the standard LM-based IR, the likelihood of a document generating is assumed to be its relevance to the query. There experiments showed that using STM is better than using LM and when we increase the number of the words in the annotation, the retrieval is more efficient.

The bag-of-visual-words approach, models an image as a bag of visual words, which is formed by a vector quantization of these local patches descriptors. The bag of visual words model describes images as sets of elementary local features called visual words. The whole visual word set is called the visual vocabulary. The description of an image database using bags of visual words relies on two steps:

1. Construction of a visual vocabulary.
2. Description of images using this vocabulary.

The vocabulary is built as follows:

1. Detection of interest regions on a set of images (detectors: Harris-affine, Hessian-affine, MSER...): some regions with geometric particularities (presence of corners, homogeneity...) are automatically extracted from the image.
2. Description of each interest region as a local descriptor (descriptors: SIFT, SURF. . .): the interest regions are described as multidimensional numerical vectors, according to their content.

Clustering of the local descriptors: the descriptors are grouped using a clustering algorithm. Each resulting cluster (or group) corresponds to a visual word. We can then describe any image as a vector of visual words occurrences, as follows:

1. Detection and description of interest regions in the image.
2. Assignation of each local feature to its nearest visual word in the vocabulary.
3. Description of the image as a vector of visual word frequencies (i.e. number of occurrences).Images can then be matched by computing a distance between the vectors describing them. [4]

Ismail Elsayad, et al. [5] used BOW representation instead of low level features and proceeded like this: Fast-Hessian detector is used to extract interest points. In addition, the canny edge detector is used to detect edge points. From both sets of interest and edge points, they used a clustering algorithm to group these points into different clusters in the 5 dimensional color-spatial features. The clustering result is needed to extract the edge context and to estimate the spatial weighting scheme for the visual words.

Anna Bosch, et al. [6] used these steps to extract BOW features:

1. Automatically detect regions/points of interest.
2. Compute local descriptors over those regions/points.
3. Quantize the descriptors into words to form the visual vocabulary.
4. Find the occurrences in the image of each specific word in the vocabulary for constructing the BOW feature (or a histogram of word frequencies).
5. Show that a better classification is achieved when a semantic representation is used in order to deal with the gap between low and high level.

It is clear that using both **visual content** and **textual content** to represent an image and retrieving similar images is much more efficient and effective than the retrieval using only visual content or textual content. Using only one modal to represent an image does not always express all the content of that image. Textual annotations are often subjective and do not express all aspects of the image. Content based image retrieval does not retrieve images based on concepts but on visual features so it might retrieve images of different concepts

Proposed Methodology:-

The problem that we are trying to solve in this project is that machine don't understand the semantics behind images [7, 8]. It only interprets the image as a set of color, shapes, etc. and then transforms them into visual words. We have proposed a new method that combines textual modality i.e. annotations with visual modality i.e. visual words to better represent the images and extracting only what is **significant** for understanding the semantics of the image. An image could have some visual words that are not significant to understand an image [9, 10]. So we have created a mechanism **to filter the insignificant** visual words based on the textual annotation. Figure 2 shows a flow chart of our proposed methodology.

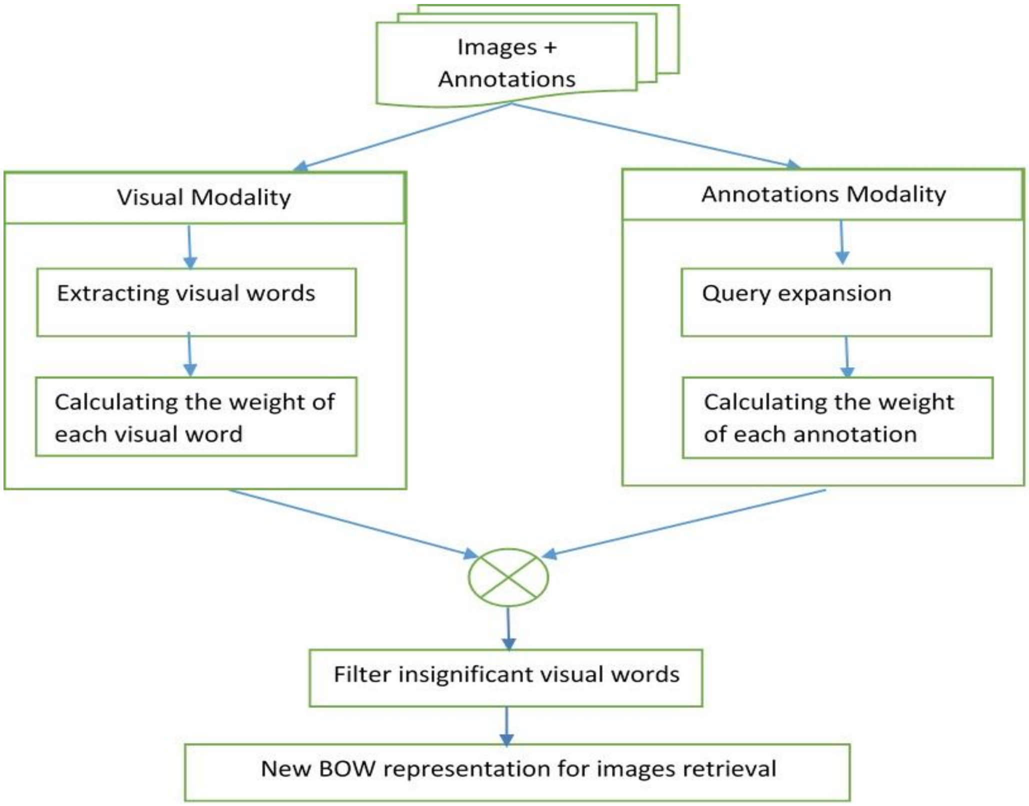
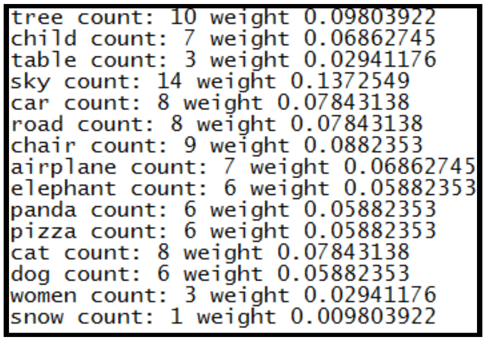

Images +
Annotations
Visual Modality
Extracting visual words
Calculating the weight of each visual word
Annotations Modality
Query expansion
Calculating the weight of each annotation
Filter insignificant visual words
New BOW representation for images retrieval

Figure 2:- Flow chart of our proposed methodology.

Annotation modality:-

To start our work, we first estimate the weight of each annotation word as shown in Figure 1. These weights represents how frequent is each annotation and how important it is in the dataset. This weight will later reflect on the significance of the visual words. In figure 3 we see the weights of the different annotations.

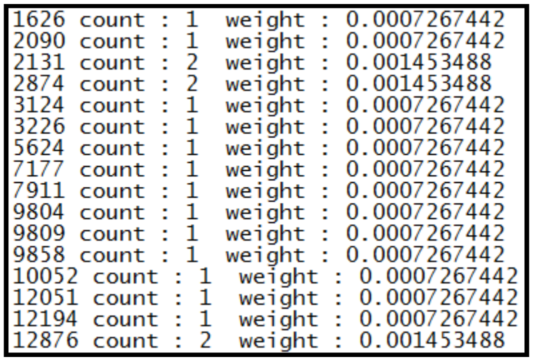


```
tree count: 10 weight 0.09803922
child count: 7 weight 0.06862745
table count: 3 weight 0.02941176
sky count: 14 weight 0.1372549
car count: 8 weight 0.07843138
road count: 8 weight 0.07843138
chair count: 9 weight 0.0882353
airplane count: 7 weight 0.06862745
elephant count: 6 weight 0.05882353
panda count: 6 weight 0.05882353
pizza count: 6 weight 0.05882353
cat count: 8 weight 0.07843138
dog count: 6 weight 0.05882353
women count: 3 weight 0.02941176
snow count: 1 weight 0.009803922
```

Figure 3:-Weights of the different annotations.

Visual Modality:-

We cluster the images of the training set into clusters based on their annotation e.g. for the "Child" annotation, we group all the images that contain the annotation "Child". After the images are clustered, the visual words are extracted from the images from each cluster and then their weights are being estimated. Figure 4 shows an example of the extracted visual words of the "cat" cluster where we can see the id of the visual words, their count and the relevant weight.



```
1626 count : 1  weight : 0.0007267442
2090 count : 1  weight : 0.0007267442
2131 count : 2  weight : 0.001453488
2874 count : 2  weight : 0.001453488
3124 count : 1  weight : 0.0007267442
3226 count : 1  weight : 0.0007267442
5624 count : 1  weight : 0.0007267442
7177 count : 1  weight : 0.0007267442
7911 count : 1  weight : 0.0007267442
9804 count : 1  weight : 0.0007267442
9809 count : 1  weight : 0.0007267442
9858 count : 1  weight : 0.0007267442
10052 count : 1  weight : 0.0007267442
12051 count : 1  weight : 0.0007267442
12194 count : 1  weight : 0.0007267442
12876 count : 2  weight : 0.001453488
```

Figure 4:-Some of the visual words extracted from the "cat" cluster along with their count and weight.

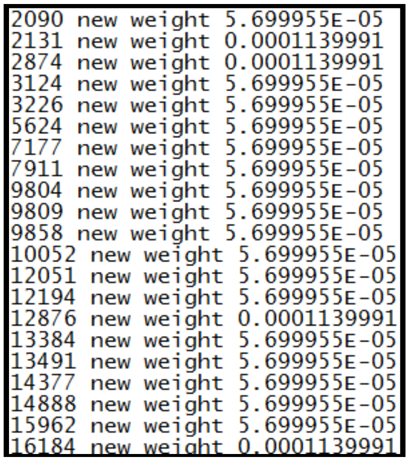
Combination of Both Modalities:-

Figure 2 shows the detailed flow chart of the combined modalities. After all these visual words are extracted and weighted, each weight is then multiplied by the weight of the relevant annotation. The weights are calculated using equation 4.1.

$$Final\ weight\ of\ VWi = \sum_{i} Weight\ of\ VWi\ \times Weight\ of\ annotation\ j \qquad (4.1)$$

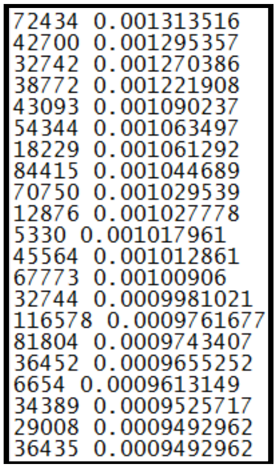
The weights of the visual words extracted inside each cluster are called local weights and then after being multiplied by the weight of the annotation, they are called global weights. In our approach we took advantage of both local and global weights to get an accurate idea about the most important visual words.

Figure 5 shows the new weights of the visual words of the cluster "cat" after being multiplied by the weight of the "cat" annotation. After these calculations we add the weights of the same visual words presented in each cluster to get a final version of the weights of the different visual words. These final weights gives an idea of the most frequent visual words i.e. the visual words that are the most suitable to represent images. Figure 6 shows the weights of the visual words ordered from the highest weight to the lowest. Since our approach was to better represent the images using the most important visual words, we filter the visual words based on a threshold. We first took the threshold as being the average weight of all the visual words. The average weight calculated was 0, 0002406286 and then we only retained the visual words with a weight higher than the average.



```
2090 new weight 5.699955E-05
2131 new weight 0.0001139991
2874 new weight 0.0001139991
3124 new weight 5.699955E-05
3226 new weight 5.699955E-05
5624 new weight 5.699955E-05
7177 new weight 5.699955E-05
7911 new weight 5.699955E-05
9804 new weight 5.699955E-05
9809 new weight 5.699955E-05
9858 new weight 5.699955E-05
10052 new weight 5.699955E-05
12051 new weight 5.699955E-05
12194 new weight 5.699955E-05
12876 new weight 0.0001139991
13384 new weight 5.699955E-05
13491 new weight 5.699955E-05
14377 new weight 5.699955E-05
14888 new weight 5.699955E-05
15962 new weight 5.699955E-05
16184 new weight 0.0001139991
```

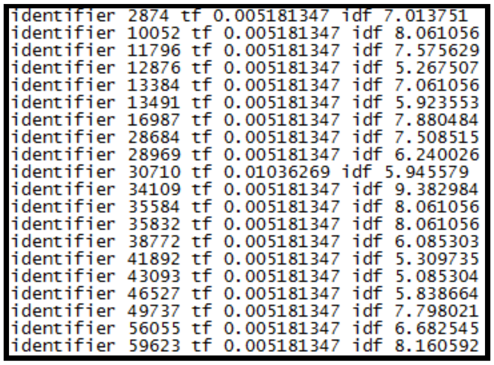
Figure 5:-The new weights of the visual words after being multiplied by the weight of the annotation



```
72434 0.001313516
42700 0.001295357
32742 0.001270386
38772 0.001221908
43093 0.001090237
54344 0.001063497
18229 0.001061292
84415 0.001044689
70750 0.001029539
12876 0.001027778
5330 0.001017961
45564 0.001012861
67773 0.00100906
32744 0.0009981021
116578 0.0009761677
81804 0.0009743407
36452 0.0009655252
6654 0.0009613149
34389 0.0009525717
29008 0.0009492962
36435 0.0009492962
```

Figure 6:-The total weights of the visual words after being added and ordered.

After this filtering, we represent the images using the tf –idf of filtered visual words shown in Figure 7. After extracting the visual words for each image, we recalculate the distance between a query image and a set of images.



```
identifier 2874 tf 0.005181347 idf 7.013751
identifier 10052 tf 0.005181347 idf 8.061056
identifier 11796 tf 0.005181347 idf 7.575629
identifier 12876 tf 0.005181347 idf 5.267507
identifier 13384 tf 0.005181347 idf 7.061056
identifier 13491 tf 0.005181347 idf 5.923553
identifier 16987 tf 0.005181347 idf 7.880484
identifier 28684 tf 0.005181347 idf 7.508515
identifier 28969 tf 0.005181347 idf 6.240026
identifier 30710 tf 0.01036269 idf 5.945579
identifier 34109 tf 0.005181347 idf 9.382984
identifier 35584 tf 0.005181347 idf 8.061056
identifier 35832 tf 0.005181347 idf 8.061056
identifier 38772 tf 0.005181347 idf 6.085303
identifier 41892 tf 0.005181347 idf 5.309735
identifier 43093 tf 0.005181347 idf 5.085304
identifier 46527 tf 0.005181347 idf 5.838664
identifier 49737 tf 0.005181347 idf 7.798021
identifier 56055 tf 0.005181347 idf 6.682545
identifier 59623 tf 0.005181347 idf 8.160592
```

Figure 7:-Representation of the image after filtering

Results And Discussion:-

To begin with our project, we first started with a tool called TopSurf [11] that helped us with our project and saved us a lot of time between getting a huge data set, extracting its descriptors and creating a dictionary of hundreds of thousands of visual words which were necessary to complete our project. TopSurf is an open source tool that extracts SURF descriptors from images and provides tools for calculating the distance between two images.

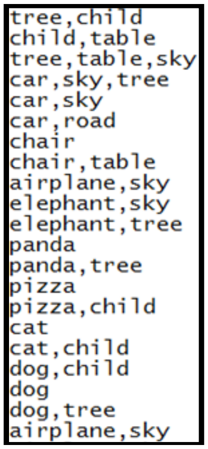
Another tool that we used in our project is WordNet [12]. WordNet is a large lexical database of English words. Words are grouped into sets of synonyms. WordNet also provides other functionalities, but we only used the mentioned earlier. Other than the downloaded TopSurf application, we added a few new functions to the existing application. Also we have created an application using Visual C# under Visual Studio 2010. Our project could be divided mainly into two parts, the visual part and the textual part. At a first instance, we worked with these two modalities separately, than we created a method to combine these two modalities in order to create a better retrieval system. The methodology that we adopted was first to work with the annotations by replacing every word by a reference word using WordNet, then calculating the weight of each annotation. The second part of the project was to cluster the images based on their annotations, extract the descriptors from each cluster, and then calculate the weight of each visual word and multiply the weight of the visual word by the weight of the corresponding annotation. Then for each visual word add its weight from each cluster. Then filter the visual words based on their weights using a threshold to retain the most important visual words. Finally re-represent the images based on the filtered visual words. The training data set was based on 1000 images from Caltech 101 [13] which is a subset of the data set used to create the dictionary in TopSurf.

As mentioned before, the training set is a collection of 1000 images from Caltech 101 dataset which is a subset of the dataset used in TopSurf. **Table 1** shows the number of images that we have chosen for testing and training.

Table 1:-Number of images used for training and testing and the number of categories used.

Number of images for training	**Number of images for testing**	**Number of categories**
1000	500	14

We chose the Caltech 101 dataset for two reasons, the first is that this dataset provides annotations for the images, where an example is shown in Figure 8, and the other reason was to get consistent results since it is a subset of the dataset used to build the dictionary. The only reason that we didn’t built our own dictionary was the lack of time, since extracting visual words for millions of images would take up to several weeks or even months using super computers. As mentioned earlier, TopSurf is an open source application. So we used its libraries and methods to edit some functions or customizing it to fit our project. The “Extract Descriptors” method provided by TopSurf, extracts the visual words of the selected images and these visual words were taken from the dictionary and then the images were represented with these visual words. In Figure 9, we can see an example of how the images are represented by their visual words. When humans see images, they interpret it as objects and semantics, but to a computer, the images are seen as a set of visual words. That’s why it is important how to represent an image and our work is focused around that point.



```
tree,child
child,table
tree,table,sky
car,sky,tree
car,sky
car,road
chair
chair,table
airplane,sky
elephant,sky
elephant,tree
panda
panda,tree
pizza
pizza,child
cat
cat,child
dog,child
dog
dog,tree
airplane,sky
```

Figure 8:-Some of the images' annotation

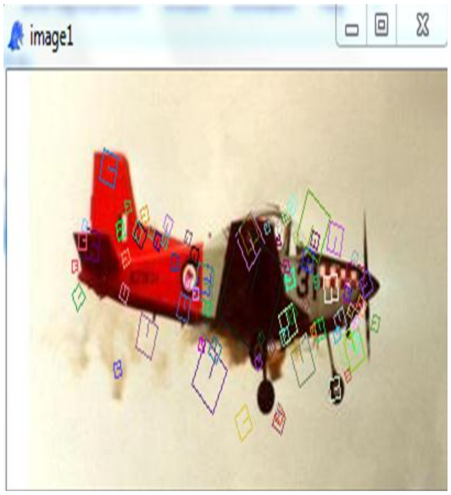


Figure 9:-Visual words extracted from an image using TopSurf

Another function provided by TopSurf is “Compare Descriptors”. “Compare Descriptors” enables us to calculate the distance between a query image and other images that we already have extracted their descriptors or visual words. In Figure 10 we can see an example the distances created by “Compare Descriptors”. This distance is a measure of how close the images are to each other in terms of their extracted visual words. A distance of 1 means that the two images are completely different and 0 means that that the two images are the same. We also customized this function to use it later in validating our work.

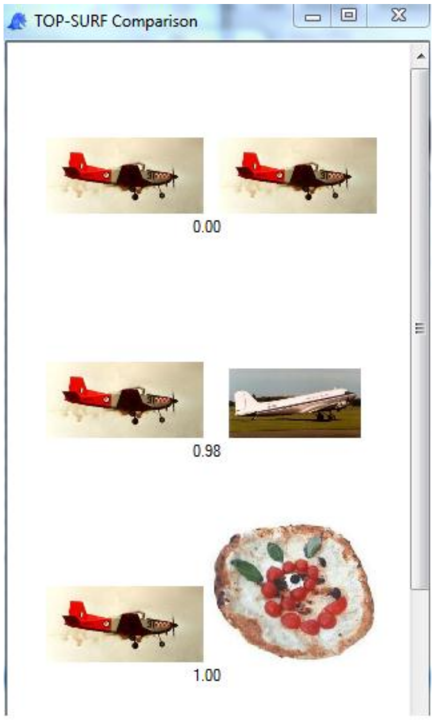
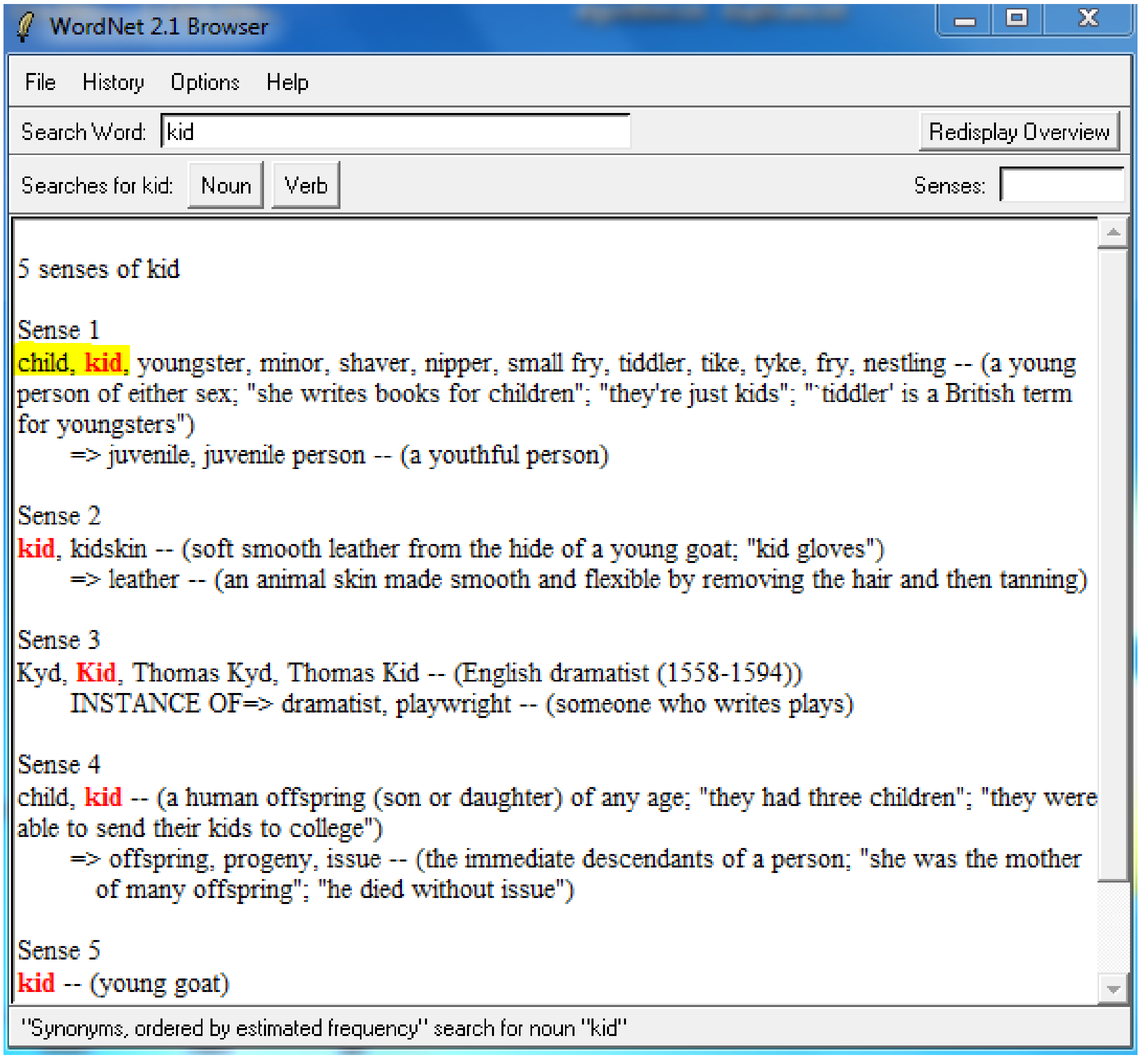


Figure 10:-An example of distance between three images.

We can see in the first case that the distance is 0 because it is the same image, in the second image the distance is 0.98 and in the third case the distance is 1 indicating that the images are completely different.

To a person, the words “house” and “home” means the same object, also the terms “kid” and “child” refer to the same thing. To a computer, every term means a different thing and a standalone computer could never link between these synonyms. To solve this problem, we decided to take a reference word for each set of synonyms to later replace a word in the annotation by the reference word which is also its synonym. For example, if an image is annotated with “kid”, the system should replace it by the word “child” as it is the reference word. To do all that we used WordNet 2.1 to give us the reference word for each word in the annotation as shown in Figure 11.



5 senses of kid

Sense 1
child, kid, youngster, minor, shaver, nipper, small fry, tiddler, tike, tyke, fry, nestling -- (a young person of either sex; "she writes books for children"; "they're just kids"; "`tiddler' is a British term for youngsters")
=> juvenile, juvenile person -- (a youthful person)

Sense 2
kid, kidskin -- (soft smooth leather from the hide of a young goat; "kid gloves")
=> leather -- (an animal skin made smooth and flexible by removing the hair and then tanning)

Sense 3
Kyd, Kid, Thomas Kyd, Thomas Kid -- (English dramatist (1558-1594))
INSTANCE OF=> dramatist, playwright -- (someone who writes plays)

Sense 4
child, kid -- (a human offspring (son or daughter) of any age; "they had three children"; "they were able to send their kids to college")
=> offspring, progeny, issue -- (the immediate descendants of a person; "she was the mother of many offspring"; "he died without issue")

Sense 5
kid -- (young goat)

"Synonyms, ordered by estimated frequency" search for noun "kid"

Figure11:- Example of word net annotation

In this image we can see that if we gave the application the word kid, it returns a list of all its synonyms where the reference word child is the first word.

Also if we give this application any of “child’s” synonyms, it will return the word “child” as the reference word. To replace all the annotations, we have created a C# application that will talk about later. Along the existing tools, we have created our own C# application to help us create the project. Figure 12 shows the GUI of the application.

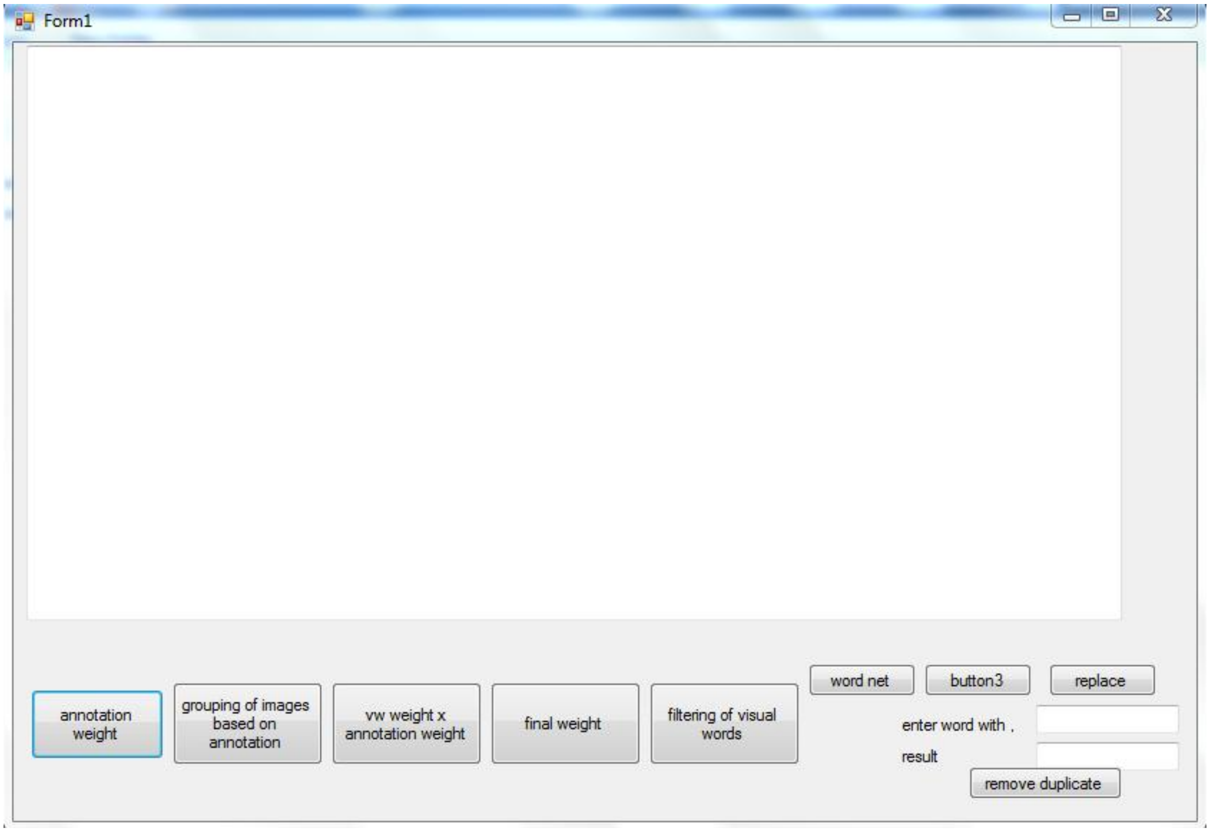


Figure 12:-The GUI of our created application.

This application provided us with tools to replace the annotations with their references, calculating the annotation weight, clustering the images, calculating the final weights of each visual word and finally filtering the visual words based on several thresholds. All these functions will be explained in details later on.

At this point, we have the TopSurf tool ready with all the modifications we added to it, the training dataset and its annotations. Now we are ready to proceed to the next phase of our project.

After completing all the steps mentioned in the proposed methodology, we have tested our system by calculating the precision of the retrieved images using a query image from each category. The calculated average precision as is shown in Figure 13.

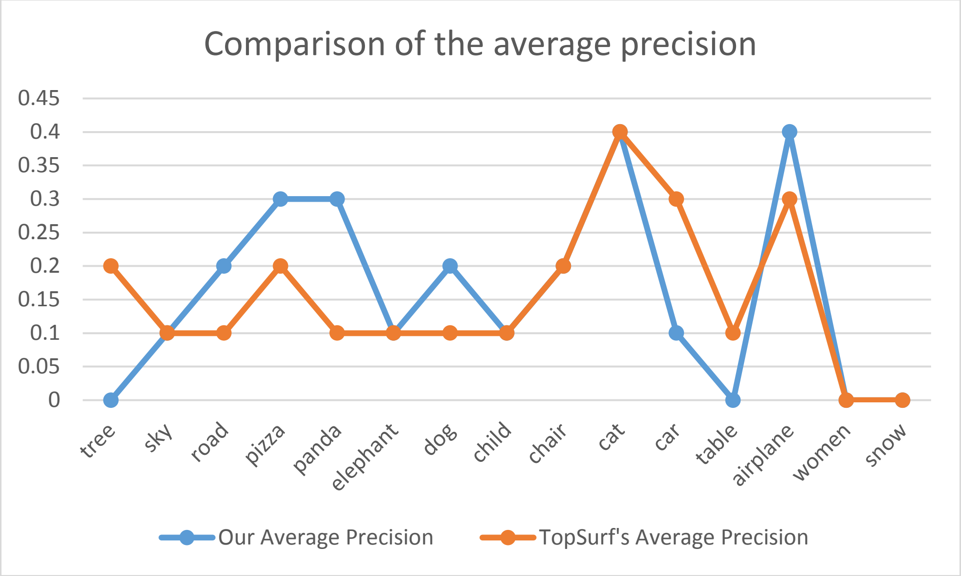


Figure 13:-Comparison of the average precision using our methodology and BOW representation.

After calculating the precision we have calculated the mean average precision of our system and for BOW. Our mean average precision was 0.17 and Bow's was 0.13 thus showing an improvement of 26.3% over Bow's method.

Conclusion and future work:-

As shown in Figure 13, we have increased the precision in most categories. Because we have represented the images using only the most occurring visual words, we have lowered the distance between two images of the same category. This is the goal of our project, to enhance the retrieval of images. Retrieval systems that only rely on visual features to retrieve images has proven that they have limitations. We have tried to enhance this system by combining the annotation of the image with its visual features. The problem that we have solved in our project was that if there are two images with the same semantics but visually different, a retrieval system would consider them as different images. We have solved this problem by combining the textual modality with the visual modality to enhance a retrieval system. We have done this by finding the most frequent visual words present in the training set and representing the images using only these visual words. We have retrieved the most occurring visual words by first getting a local weight for each visual word and then getting a global weight by multiplying the weights of the visual words by the weight of the corresponding annotation. And then we have filtered the visual words based on a threshold that we chose to be the average weight. Now we can represent the image not only by using the visual modality but also by using the semantics of the image. After many results, we have proven that our method have shown an improved result over Bow's result.

We have started this project in order to enhance a retrieval system that is based on the bag of visual words representation. This retrieving system only relies on the visual aspect of the image which is the BoW representation. This method has its own drawbacks in respect to the semantic learning of the image, e.g. an image of a red ball is the same of an image of a tomato in the BoW's perspective. We have tried to enhance this method by inserting the concept of textual annotations to the retrieving system and image matching. These annotations were created by humans, so they add some sort of semantic to the image and thus an image of a red ball won't be the same of an image of a tomato in the new retrieval system.

To create our project, we have extracted VWs from a large training data set and then we have calculated the weights of the visual words and then multiplied the weight of each visual word by the weight of the corresponding annotation. After getting all these weights, we have filtered them based on the average weight and finally we re-represented the images using only the filtered visual words.

Finally we have tested our system using 500 images from the training set and retrieved images using our system and using the BoW's representation over 14 categories. We have then calculated the average precision for each category for both systems and then we have calculated the mean average precision (MAP). We can see that in most categories, we have surpassed the results of the Bow representation. Our MAP was 0.17 while BoW's MAP was 0.13. We have shown an improvement of 26.3% over BoW's results.

In this project, we have proven that using both modalities in image representation has shown a major improvement in image retrieval. This improvement is done by highlighting the semantics of the image. In our method, the image is no longer viewed as a set of VWs, but as a set of important VWs that represent the semantics in the images. We mean by important VWs, the most frequent VWs that are associated with a certain category. Our method have exploited the annotation to give a semantic to the image. This way, the retrieval system could differentiate between two images that are visually similar but belonging to different categories.

The main drawback in our project was mainly the long time required to perform the training. The limitations of this system are when we face two images that belong to the same category but are totally different. In this case we can't improve much the retrieving.

Finally we have created a better retrieval system that uses both textual and visual modalities. The combination of these two modalities have led to a better performing image retrieval system.

For future work on this project, we can include the use of the annotation in the retrieval system by not only giving an image as input, but also associate it with a textual word to narrow down the search results.

Another future work is to implement this system on an online system so users can benefit from it and also increase our data set by uploading images to the system and make it more effective.

Also this application could be used in many domains other than simple image matching. It can be costumed to serve the user's requirements like using it for medical purposes, then the training set would be a collection of medical images along with the diagnostic for each image, then by simply giving a new medical image to the system, it can give an estimated diagnostic based on its data set.

References:-

1. Zhou, Xiang Sean, and Thomas S. Huang. "Unifying keywords and visual contents in image retrieval." Multimedia, IEEE 9.2 (2002): 23-33.
2. Inoue, Masashi. "On the need for annotation-based image retrieval. "Proceedings of the Workshop on Information Retrieval in Context (IRiX), Sheffield, UK. 2004.
3. Inoue, Masashi, and Naonori Ueda. "Retrieving slightly annotated images."
4. Tirilly, Pierre, Vincent Claveau, and Patrick Gros. "Language modeling for bag-of-visual words image categorization." Proceedings of the 2008 international conference on Content-based image and video retrieval. ACM, 2008.
5. Elsayad, Ismail, et al. "A new spatial weighting scheme for bag-of-visual-words." Content-Based Multimedia Indexing (CBMI), 2010 International Workshop on. IEEE, 2010.
6. Bosch, Anna, Xavier Muñoz, and Robert Martí. "Which is the best way to organize/classify images by content?." Image and vision computing 25.6 (2007): 778-791.
7. Zhao, Rong, and William I. Grosky. "Narrowing the semantic gap-improved text-based web document retrieval using visual features." Multimedia, IEEE Transactions on 4.2 (2002): 189-200.
8. Sikka, Karan, et al. "Exploring bag of words architectures in the facial expression domain." European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2012.
9. Zheng, Liang, and Shengjin Wang. "Visual phraselet: Refining spatial constraints for large scale image search." IEEE Signal Processing Letters 20.4 (2013): 391-394.
10. Escalante, Hugo Jair, et al. "Evolving weighting schemes for the Bag of Visual Words." Neural Computing and Applications 28.5 (2017): 925-939.
11. B. Thomee, E.M. Bakker, and M.S. Lew, "TOP-SURF: a visual words toolkit", in Proceedings of the 18th ACM International Conference on Multimedia, pp. 1473-1476, Firenze, Italy, 2010.
12. George A. Miller, "WordNet: A Lexical Database for English"Communications of the ACM Vol. 38, No. 11: 39-41. 1995
13. Griffin, Gregory, Alex Holub, and Pietro Perona. "Caltech-256 object category dataset." (2007).