CONTENT BASED IMAGE RETRIEVAL

By, Akhiyar Ahmed

Suhaib Abdulghani

Supervisor: Dr. Imran Ahmad

**What is CBIR?**

Content Based Image Retrieval (CBIR) is the application of computer vision techniques. Initially, image meta search was used by annotating an image with metadata and inputting keyword descriptions. Due to the importance for visual content, such as images, there has been a significant development in improving this method because many images online are not annotated. Thus began the introduction of CBIR.

The purpose of CBIR is to make the process of image retrieval faster by using techniques like query or content comparison using image distance. CBIR eliminates the process of manually annotating images which is time consuming and might not be a 100% accurate.

**CBIR techniques**

There have been many development in CBIR systems but the problem still remains in retrieving images on the basis of their pixel content (Lew, Sebe, Djeraba, & Jain, 2006). There are two techniques used in CBIR, the first is query technique, which includes: semantic retrieval, relevance feedback, and Iterative/Machine Learning. The second technique being content comparison using image distance; this includes colour, texture shape and spatial location (Roebuck, 2012). This report will primarily focus on content comparison using image distance measures.

Image distance measures is used to compare the similarity of two individual images using different approaches (colour, texture, shape, and others). To get the most accurate match, the distance measure must be 0; the closer to 0 the more accurate the result of the image. Using distance measure search results can be sorted based on their distance to the queried image (Stockman & Shapiro, 2001).

Using the colour approach, to compute the image distance measure, works by computing a [colour histogram](http://en.wikipedia.org/wiki/Color_histogram) for each image. That being said, "There are so many methods used to retrieve the colour feature. They include colour histogram, autocorrelogram, colour moments" (V. & Kumar, 2014).

With texture, distance measure is done by finding some visual patterns and the way they are being defined. "Textures are represented by pixels which are then placed into a number of sets, depending on how many textures are detected in the image. These sets not only define the texture, but also where in the image the texture is located" (Stockman & Shapiro, 2001).

As for shape, the idea is to look for a defined shape in a specific area of an image. "Shapes will often be determined [by] first applying segmentation or edge detection to an image. Other methods use shape filters to identify given shapes of an image" (Tushabe & Wlkinson, 2008)

**Advantages and Disadvantages of Colour**

As mentioned in the article, "Examining images based on the colours they contain is one of the most widely used techniques because it can be completed without regard to image size or orientation" (Rui, Huang, & Chang, 1999). Hence the technique used in this project was to measure image distance using the colour approach. The advantages of using colour histograms are the increase in speed and low memory space (V. & Kumar, 2014).

On the other hand, Colour Histogram also have disadvantages, as stated in this article, “a) CH is sensitive to noisy interferences such as illumination changes and quantization errors; b) large dimension of CH involves large computation on indexing, c)It does not take into consideration colour similarity across different bins, d)It cannot handle rotation and translation. It means that information about object location, shape, and texture is discarded. e) Two perceptually very different images with similar colour distribution will be deemed similar by a colour histogram based retrieval system “ (Sharma, Rawat, & Singh, 2011).

**Process Implementation**

When the program is executed it gives the user three buttons: first button is choose file, second is to upload, third is to find similar images, the program also provides the user with a drop box where the user can drop an image instead of clicking on choose file and browsing. Lastly, the program provides the user with a slider of 0-100, the slider indicates the accuracy of the returning results.

When an image is chosen or dropped into the box, the user can either upload the image or search the database for relevant images. The user can drop or chose multiple images, but will only be able to upload them, if user tries to search using multiple images an error would occur. First, when “upload” is clicked the program will run a function called “index\_picture.” This function as indicated will index the chosen image.

At first the index\_picture function was done by breaking the image down to a 3bit image, the six primary and secondary colours including black and white. The program was able to pull results with the accuracy margin being very high, more than 3% of the library almost always returned with a 100% result. To read an image as a 3bit image, every pixel was first read as a RGB (red, green, blue); each R, G, and B was divided by 128 which would return either a 0 or 1 value representing each colour. After which, depending on what the RGB value equaled, an array of 8 received a +1 for that colour. This process broke the image down to a 3bit image of 8 colours. For example, if the RGB value was [220, 140, 8], respectively, by diving each value by 128 the new RGB value would be [1,1,0], which in two bit is six. This new value would now provide a +1 to the array[6]. Consequently, this would read all the pixels in an image and find how many times each colour appears in the image. The results for the process of 3bit image was not very accurate, hence Mr. Ahmad suggested to try either 4bit or 6bit as it would have a better accuracy return.

The code for indexing an image was then re-done to be read as a 6bit image, this process involved dividing the RGB value by 85 and this would give a range of 0-3 for each RGB representing a total of 64 colours. The adding process for counting the pixels of a 3bit image was replicated for a 6bit image. After this process is completed, each array value is divided by the total number of pixels to determine the percentage of how often a colour appears in the image. For example if array [17] = 450 and the image size was 4096 pixels, then the percentage value would be 10.98. When the program indexed images using 6bit, it resulted in a more relevant images than with the 3bit indexing.

When indexing is complete the function returns the array back to the caller of the function, if the button for upload was clicked it would take the indexed image and call the function “upload\_pic.” The function takes the array and path of the indexed images to be uploaded as a parameter. In this upload function, it writes the indexed data of the image into a text file ending the line with the next numeric value of the image. This image is then copied over into the database directory and renaming the image to the numeric value, as mentioned previously. For example if the database has a 100 images, the next uploaded image would be renamed to 101.jpg.

If the button “find similar images” was clicked it would first index the image and then run the function “find\_similar.” This function then would take the array of the indexed image as a parameter alongside the accuracy chosen by the slider. It would then compare the indexed image to all the images in the database, while returning the most similar as a result. For example, index\_image = [10,20,0,30,0,10,30,0] and the image in database its comparing to is [0,30,20,0,10,20,20,0]. It would start by comparing the first number from each array. If any number equates to 0 then the result for that colour is also 0, or else the average of the two given numbers are taken. Hence when 10 is compared to 0, it equates to 0, and when 20 compared to 30, it equates to 25. The final result of comparison would then be [0,25,0,0,0,15,25,0]. This accuracy would add up to 65%.

If the accuracy of the image is higher than of the slider chosen by the user, the image number is saved into an array called results. The function continuously repeats this comparison for all the images in the databases. Once completed it will rearrange the results from highest to lowest accuracy, returning the top 250 images.

It was noted that an accuracy of a 100% was not being returned for the exact same image. The problem was being caused by the index image function. The function was storing a value to only 2 decimal places, but when the accuracy was increased to 3 decimal places it fixed the issue and provided more accurate results.

If more time was provided the program would have been implemented to index an 8bit image representing 256 colours. A better way to implement this would be to read the image as HSV (hue, saturation, and value/brightness) instead of RGB. HSV is better way of reading an image because the hue value does not need to change to pick a darker or lighter version of the same colour, whereas compared to RGB one might need to change a blue value to pick a lighter version of red instead of being able to only modify the red value. Using HSV will also help as it would change the complexity of the program. Each colour (hue) can be represented in its own text file, this process would eliminate the need to search the entire database when the primary colours are known.

Also to increase accuracy of results more techniques could be implemented, such as shape and texture. If shape was implemented alongside colour, it would narrow the results. For example, if a red phone booth was being searched for in a database, it would eliminate the result of a red chair, which would be a result when using just colour.

**Conclusion**

Content Based Image Retrieval helped with overcoming the limitations image meta search by taking in consideration the features of a specific image. Currently, CBIR is being used in many applications such as architectural and engineering designs, geographical information and remote sensing systems, medical diagnosis, historical research, and many other fields.

Based on the choices of distance measure techniques, colour was chosen because of the time constraint and in contrast to shape, colour would show accurate visual results. The results computed by the program can be greatly improved if more time was provided.

In conclusion, CBIR enabled an in-depth understanding with regards to the importance, the limitations, the techniques and the process; the process being how CBIR is used to extract the features of an image and then calculate the colour histogram which is then used to find similar images in a large database. This project also introduced the concept of the various fields and applications of CBIR.

**References**

Lew, M., Sebe, N., Djeraba, C., & Jain, R. (2006, February). *Content-based Multimedia Information Retrieval: State of the Art and Challenges* . Retrieved from http://www.liacs.nl/~mlew/mir.survey16b.pdf

Roebuck, K. (2012). *Gesture Recognition: High-impact Emerging Technology - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendor.* Emereo Publishing.

Rui, Y., Huang, T., & Chang, S.-F. (1999). *Image Retrieval: Current Techniques, Promising Directions, and Open Issues.* Academic Press.

Sharma, N., Rawat, P., & Singh, J. (2011). Efficient CBIR Using Colour Histogram Processing. *Signal & Image Processing : An International Journal*.

Stockman, G., & Shapiro, L. (2001). *Computer Vision.* Prentice Hall .

Tushabe, F., & Wlkinson, M. (2008). *Content-Based Image Retrieval Using Combined 2D Attribute Pattern Spectra.* Springer Berlin Heidelberg.

V., A., & Kumar, S. (2014, May). *SURVEY ON CONTENT BASED IMAGE RETRIEVAL TECHNIQUES.* Retrieved from Esat Journals: http://esatjournals.org/Volumes/IJRET/2014V03/I19/IJRET20140319135.pdf

Shapiro, Linda G. (2002). *Consistent Line Clusters for Building Recognition in CBIR*. Proceedings of the International Conference on Pattern Recognition, http://imagedatabase.cs.washington.edu/

Li, Jia, & Wang, James. Z. (2003). *Automatic linguistic indexing of pictures by a statistical modeling approach*, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* http://wang.ist.psu.edu/docs/related/