IMAGE SEARCH ENGINE USING COLOR HISTOGRAM

Yogeetha Amrutheshwar Kamath[1], Pallavi Shetty[2], Balachandra Rao[3]
[1]Student, Department of MCA, NMAMIT, Nitte.
[2]Associate Professor, Department of MCA, NMAMIT, Nitte.
[3]Assistant Professor, Department of MCA, NMAMIT, Nitte.

ABSTRACT

In this paper we tend to advise a model for a pursuit engine where an image is uploaded from the native database of the user to retrieve similar images as query image. This can be like the standard keyword search utilized by most of the search engines with the soul distinction that is picture itself is uploaded as a query instead of textual keywords. The fact that the image being employed as query makes the search even more difficult because the content of the image has to be analyzed and matched to find the data corresponds to the uploaded image. This is most apt for searching information regarding images of historical monuments, places or any specific place or factor that's recognizable.

Keywords: Features, indexing dataset, image descriptor.

1. INTRODUCTION

A **search engine** is an information retrieval system designed to help find information stored on a computer system. Example search engines are Google, Bing. This usually takes textual data as query and produces the related information by searching in its webservers. Whereas for image search engine instead of textual data we produce image itself as query to get similar or related information or images.

Searching an image can be done in many ways some are traditional searching technique that is manually comparing images and finding similar images but manual searching of an image is a tedious job when there are many images to be searched. Scalability is a major issue. It is very time consuming to search for a required image. It makes things very difficult when we want to search an image based on a particular location because we need to search all the images when there are more than 200 images.

Other techniques are *search by metadata*, *search by example*, *hybrid search*. Search by metadata which is similar to traditional keyword based searching; In this system content of the image is examined occasionally, rather it depends on textual hints such as manual annotations and tagging performed by humans along with auto generated contextual clues, such as the text that appears near to the image on webpage. When user uses search by meta-data system for searching purpose user provides a query in textual format just like a traditional search engine which produces resultant images that have similar tags or annotations.

In search by example system rely solely on the contents of the image no keywords are assumed to be provided. The image is analyzed, quantified, and stored so that similar images are returned by the system during a search. Image search engines that quantify the contents of an image are called **Content - Based Image Retrieval** (CBIR) systems. It is relying *strictly* on the contents of the image and not any textual annotations associated with the image.

By utilizing some sort of algorithm to extract “features” (i.e., a list of numbers to quantify and abstractly represent the image) from the image itself. Then, once a user submits a query image, you extract features from the query image and compare them to your database of features and take a look at to search out similar images.

Again, it’s important to reinforce the point that Search by Example systems rely strictly on the contents of the image. These types of systems

tend to be extremely hard to build and scale, but allow for a fully automated algorithm to govern the search no human intervention is required.

And in hybrid search which combines the both techniques mentioned above for this kind of search we can take example of twitter in which by using hash-tags and images searched results will be produced .

In this paper we are concentrating on the CBIR technique that is search by example using mage features for searching similar images as a result.

2. LITERATURE SURVEY

2.1 Existing System: Existing system refers to the system that is being followed till now. Presently all the image searching is done manually which is very hectic.

Limitations of Existing System:

- Time Consuming.
- Searching for a particular image makes very difficult.
- Manual searching makes the job of the User tedious.
- Scalability is a major issue.

2.2 Proposed System: The main objectives of the systems are to provide easy access to search the image in advanced and ordered fashion. It is easy to search and retrieve the similar matched images with respect to the query image. It helps to save the image that is returned for future use. We can find images all without labeling or tagging a single image.

Advantages of proposed system:

- Removes the need of manual search.
- User friendly and Interactive. Security of data.
- Saving the images for further use.
- Scalable enough to add any number of images.

3. ABOUT FEATURE EXTRACTION AND IMAGE DESCRIPTOR:

When building an image search engine we will first have to index our dataset. Indexing a dataset is the process of quantifying our dataset by utilizing an image descriptor to extract features from every image.

An image descriptor defines the algorithm that we are utilizing to describe our image. For example The mean, and standard deviation of each Red, Green, and Blue channel, respectively, The statistical moments of the image to characterize shape. The gradient magnitude and orientation to explain each form and texture. The important takeaway here is the image descriptor governs how the image is quantified.

Features, on the other hand, are the output of an image descriptor. When you put an image into an image descriptor, you will get features out the other end. In the most basic terms, features (or feature vectors) are just a list of numbers used to abstractly represent and quantify images.

Feature vectors can then be compared for similarity by using a ***distance metric*** or ***similarity function***. Distance metrics and similarity functions take two feature vectors as inputs and then output a number that represents how “similar” the two feature vectors are.

4. IMPLEMENTATION

At first defining image descriptor is required for that instead of using typical color histogram we have applied some tricks to make our process little robust and powerful. Our image descriptor will be a 3D color histogram in the HSV color space (Hue, Saturation, Value). Typically, images are represented as a 3-tuple of Red, Green, and Blue (RGB). We often think of the RGB color space as “cube”. However, while RGB values are simple to understand, the RGB color space fails to mimic how humans perceive color. Instead, we are going to use the HSV color space which maps pixel intensities into a cylinder(figure 1).

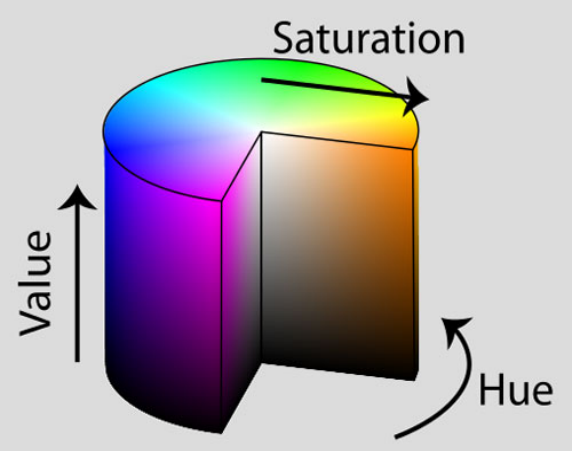

Figure 1: Example of RGB Cylinder

So we have selected our color space, we need to define number of bins for histogram. Histograms are used to give a rough sense of the density of pixel intensities in an image. Essentially, our histogram will estimate the probability density of the underlying function, or in this case, the probability *P* of a pixel color *C* occurring in our image.

It's vital to notice that there's a trade-off with the amount of bins you choose for your histogram. If you choose too few bins, then your histogram will have less parts and unable to disambiguate between images with well totally different color distributions. Likewise, if you utilize too several bins your histogram can have several parts and pictures with terribly similar contents could also be regarded and "not similar" once actually they are (figure2&3).

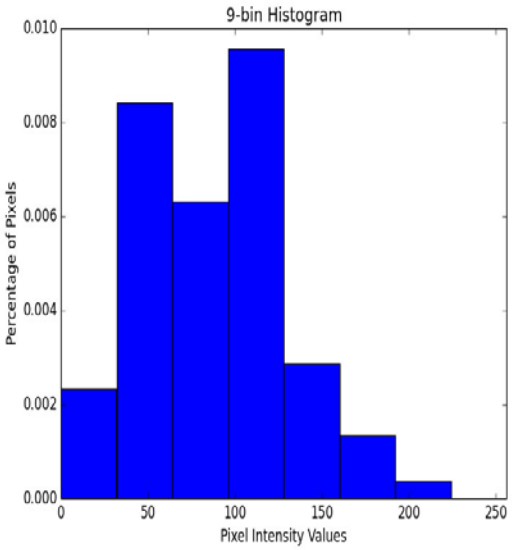

Figure 2: 9-bin histogram.

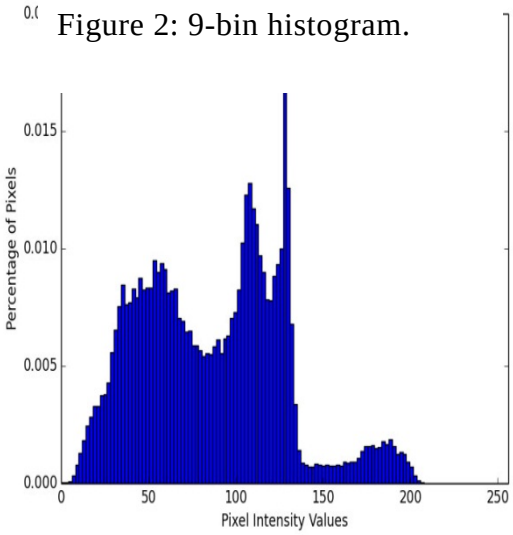

Figure 3: 128-bin histogram.

Instead of computing 3D HSV for entire image we compute it for different regions in an image. So that it allows us to simulate locality in color distribution. Once the descriptor is defined indexing the data set is done by extracting the features from each image and storing it in database for future purpose of searching. After that by defining similarity matric searching can be done to retrieve similar images as query image.

Conclusion:

This Engine makes searching of the mages much easier than expected. It allows the user to organize the images into collections and access them. It is very useful while there are many images so that the user does not check each image to find the required image.

Reference:

[1] https://www.pyimagesearch.com/

[2] https://www.techopedia.com/definition/12708/search-engine-world-wide-web

[3] Rekhil M Kumar, Sreekumar K, "A Survey on Image Feature Descriptors", International Journal of Computer Science and Information Technologies, Vol. 5 (6), 2014, 7668-7673

[4] http://www.nlpr.ia.ac.cn/fanbin/slides/CVPR17 Tutorial on Local Feature-PartI.pdf

[5] Abdel-Hakim, A.E., Farag, A.A.: CSIFT: A SIFT descriptor with color invariant characteristics. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2.

[6] Brown, M., Hua, G., Winder, S.: Discriminative learning of local image descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **33**(1), 43–57.