# Content-Based Image Retrieval: A Comprehensive Study

Iman AlSamman
Software Engineer Dep.
Damascus University
Damascus, Syria
iman.alsamman@gmail.com

Mohamed Khair Dimashky
Software Engineer Dep.
Damascus University
Damascus, Syria
mohamed.khair.dimashky@gmail.com

Albatool AlHorany
Software Engineer Dep.
Damascus University
Damascus, Syria
batool.horany@gmail.com

***Abstract*—in this paper, we attempt to explore different algorithms and methods to retrieve images based on its content. Three different methods introduce, the first one is color based and the rest are deep content using neural network. The proposed CBIR system is evaluated by querying different images and the efficiency of the proposed system is evaluated by means of the precision-recall value of the retrieved results.**

***Keywords—CBIR, color histogram, neural network, object detection.***

## I. INTRODUCTION

Content Based Image Retrieval (CBIR) becomes one of the most important search field in information retrieval. In general, retrieving images can have done based on its labels/tags or its visual content. Visual contents can represented through vector of features that model the image from different aspects, such as color/texture/shape. As development of the neural networks and machine learning in the last decade, deep learning improve retrieving images by giving semantic meaning to them, filling the "semantic gap" between low level representation of the images and human needs. The rest of this paper is organized as follows, section 2 presents previous work. Material and methods are presented in section 3. Proposed system is evaluated with Precision/Recall and finally conclusion is given in section 5.

## II. RELATED WORK

Simple search engine was introduced, focused only on news that represented as text. Vector model was used to model documents and cosine similarity for ranking. Evaluated by The New York Time news benchmark.

## III. MATERIAL AND METHODS

Building any CBIR system can be boiled down into 4 distinct steps:

- Defining image descriptor.
- Indexing the dataset.
- Defining similarity metric.
- Searching: using query by example or query by text.

### A. *Feature Extraction Phase*

Features are the output of an image descriptor. In the most basic terms, features are just a list of numbers used to abstractly represent and quantify images.

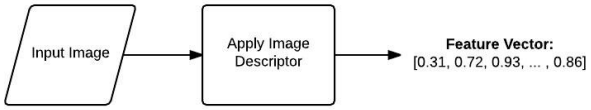



Figure1, producing a feature vector.

#### 1) *Color Histogram*

The first and the most basic extracted feature is color. The most basic technique for the color feature is the color histogram feature. The histogram is the graphical representation of the data. This color histogram represents the information of about how much quantity of the same color is present in the image. Here quantity simply means how many pixels are of the same color. This can be computed really very fast and this feature of the query image is stored in an array. At the search time, the user specifies the proportion of the color in the image and the image matching that proportion is returned.

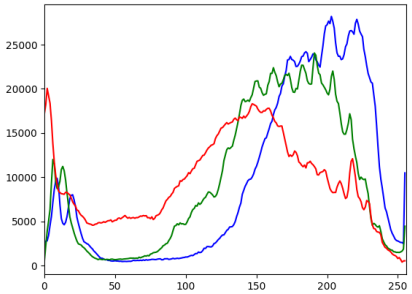



Figure2, example of RGB histogram.

Using a global histogram, we would be unable to determine where in the image the some color occurs. Instead, we would just know that there exists some percentage of each color.


Figure 3, Example of dividing our image into 5 different segments.

### 2) *Conventional Neural Network*

Deep Learning has the ability to automatically extract **meaningful** representations when trained on a large enough dataset, so since 2012, Deep Learning has slowly started overtaking classical methods such as Histograms of Oriented Gradients (HOG) in perception tasks like image classification or object detection. A Convolutional Neural Network (CNN) is a Deep Learning algorithm, which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNN have the ability to learn these filters/characteristics.[5]

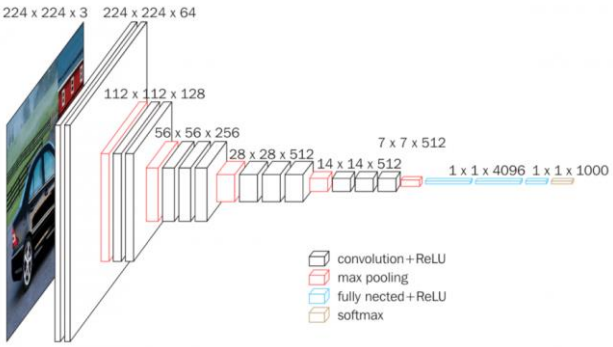



Figure 4, VGG16 network, example of CNN.

VGG16 is a convolutional neural network model. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. VGG16 was trained for weeks and was using NVIDIA Titan Black GPU's [1]. With pre-trained models, we can generate embedding (which is the penultimate layer of the network) for our images. Embedding can be stored as a vector of number (that represent as we say the final layer before the classification layer).

| | 4085 | 4086 | 4087 | 4088 | 4089 | 4090 | 4091 | 4092 | 4093 | 4094 | 4095 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.4546 | 0 | 6.1516 | 0 | 1.3763 | 0 | 0 | 1.1756 | 0.4681 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1.8355 | 0 | 0 | 0 | 0 | 0.6923 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0.4420 | 5.2478 | 0 | 0 | 1.6778 | 0 | 0 | 0 |
| 4 | 0.1924 | 0 | 0 | 0 | 0 | 0 | 0 | 0.8986 | 0 | 0 | 0 |
| 5 | 0 | 0 | 2.0729 | 0 | 0 | 0 | 0 | 0.0996 | 0.2377 | 0 | 0 |
| 6 | 0 | 0 | 6.8556 | 0 | 1.3900 | 0 | 0 | 0.6859 | 1.1272 | 0 | 0 |
| 7 | 0 | 0 | 2.2498 | 0 | 0 | 0 | 1.8188 | 0.1840 | 0 | 0 | 0 |
| 8 | 0 | 0 | 6.0193 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0.9562 | 0 | 0.3197 | 0 | 1.8738 | 3.5308 | 0 | 1.3911 |
| 10 | 0 | 0 | 0.2099 | 0 | 0 | 0 | 1.4480 | 1.3150 | 1.1056 | 2.1684 | 0 |

Figure 5, Image Embedding's.

Through image embedding's, comparing similarity between images transformed into calculating distance between its embedding through a suitable distance metric and return closed images first.

Another use of Neural Network is classification and object prediction, which can turn an image to a set of objects (terms) with different accuracy (here we have taken the classification layer). After we extract all objects from all images, we build an inverse index with object name as a key and images with its accuracy as value, and apply a matching function to find nearest images to the query image, so we have transformed the image content into set of objects.

ResNet 50, which is a short name for Residual Network. In residual learning that can lead to create residual network. Residual can be simply understood as subtraction of feature learned from input of that layer. The "50" refers to the number of layers it has.

As we see in Figure 6, ResNet50 become more accurate than VGG16, because is much deeper (50 layers) than it (16 layers).

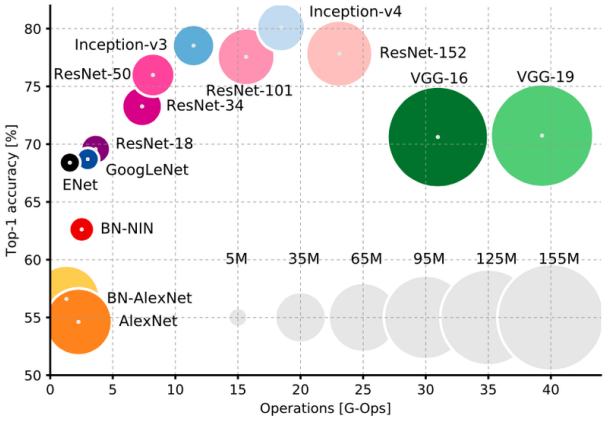



Figure 6, CNN models comparison

## B. *Distance Metrics*

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.

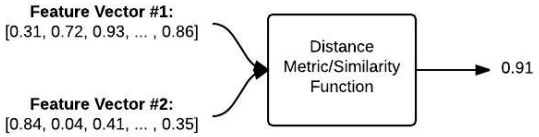



Figure7, similarity function.

### 1) *Chi Distance*

The chi-squared distance is a distance between two histograms having the same bins both. Moreover, both histograms are normalized, i.e. their entries sum up to one. The distance measure is usually defined as:

$$d(x,y) = sum(\frac{(xi - yi)^2}{xi + yi}) \ / 2$$

It is often used in computer vision to compute distances between some bag-of-visual-word representations of images.

### 2) *Euclidian Distance*

The Euclidean distance or Euclidean metric is the "ordinary" straight-line distance between two points in Euclidean space.

$$d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + ... + (q_n - p_n)^2}$$

## IV. IMPLEMENTATION AND TOOLS

In this section, we will expose tools that we used to achieve this system. The programming language that we selected is Python 3.

### *A. OpenCV*

We have used OpenCV library to find the histogram of the image and extract color features from it. OpenCV is a library of programming functions mainly aimed at real-time computer vision.

### *B. Keras*

Keras provides a set of state-of-the-art deep learning models along with pre-trained weights on ImageNet. These pre-trained models can be used for image classification, feature extraction, and transfer learning. We have used to extract embedding from images (VGG16), classify objects inside image with (ResNet50).

## V. EVALUATION

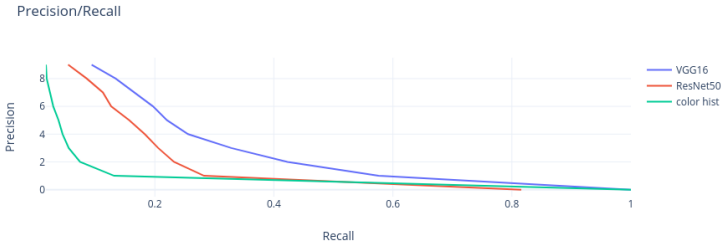
The dataset that we used to evaluate our system is SUN 09 dataset [4], which contains 12,000 annotated images. Relevance images were taken by their labels, for example, all images that start with "s_street_stree" are relevance to each other's. We also have ignored the ordering of the relevant images. Took 14 images as test queries that have different scenes and objects.

Mean Average Precision (MAP) has calculated for each method, as it looks in the table

*Table 1 MAP*

| | Color based | Embedding | Object det. |
|---|---|---|---|
| Precision | 0. 22695273 | **0. 33939855** | 0.3164314 |

Precision/Recall chart for introduced algorithms, it shows that after the recall reaches 0.5, the relevant images was almost retrieved in VGG16. The worst result was in color histogram where its curve indicates that retrieving relevant images was very slowly. ResNet50 gives a medium result between the other.




## CONCLUSION

In this paper we proposed an efficient CBIR system with three different methods, which retrieves the relevant images from a large database for a given query image. The first one extracts the color features from the query image as well as from the database images and calculates distance with chi metric. The second one extracts image embedding's as its features automatically using CNN. The last method, we used pre-trained model to predict objects from the images and index them in dictionary data structure based on inversed index concept and matching with Boolean model. Because using Boolean model in object detection, the results was not very good, we assume that if we use advance matching model will give better results.

## REFERENCES

[1] https://neurohive.io/en/popular-networks/vgg16/

[2] https://blog.insightdatascience.com/the-unreasonable-effectiveness-of-deep-learning-representations-4ce83fc663cf

[3] Stefan Ruger, "Multimedia Information Retrieval", Morgan & Claypool 2010

[4] https://groups.csail.mit.edu/vision/SUN/

[5] Ankita Doiphode and Sunil Yadav, "Image feature extraction System of CBIR using Neural Network", IOSR-JCE (India) 2017