# Content Based Image Retrieval

Nikhil Yadav, Anushree Das

April 2021

**Abstract**

We present a study of content based image retrieval system or an image based search engine which looks for similar images with respect to the query image and returns them. We compare different feature extractors resulting from different approaches like color based histogram[4][2], texture from gabor filter[1], history of oriented gradients(HOG)[3], neural network like Resnet50, and their corresponding performances.

## 1 Introduction

We are trying different approaches on Content Based Image Retrieval and comparing their performance. Currently, we have experimented with Gabor filter, Color histogram, History of Oriented Gradients(HOF), and Resnet50 NN. Theoretically, a NN should outperform all the other algorithms as it is a better feature extractor than the other approaches. The Gabor filter tries to capture the texture features in the dataset, the color histogram captures the color features, the HOG captures the orientation of edges and the NN however, tries to capture everything it can inorder to best represent the image from lines, edges, color, textures, shapes, and the other minute details with each layer have their own role in capturing something from the image that helps determine it. We try to create an embedding or feature space of all the images that are present in the dataset, and in that vector space, we look for similarity between the query image and the closest images to it using the euclidean distance.

## 2 Background

CBIR has been attempted by several groups using different features. The system aims to query by example and retrieve similar images to the query image. Let's discuss the several attempts at creation of this system.

### 2.1 Content-Based Image Retrieval using Gabor Filtering

They propose a Gabor filter based image extraction. Thus, 3D image feature vectors using even-symmetric 2D Gabor filters are computed for the images of

a large collection and for the input image. At each step an input image is selected, from the output set obtained in the previous step, and the most similar images from the collection are retrieved. The bidimensional Gabor filtering is a widely used tool in shape and texture analysis and also other image processing domains such as: image smoothing, edge detection, fingerprint recognition, iris recognition or image coding was essential . The Gabor filter, or Gabor wavelet, represents a bandpass linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function. The two-dimensional Gabor filter can be viewed as a sinusoidal plane of particular frequency and orientation, modulated by a Gaussian envelope. Basically, Gabor filters are a group of wavelets, with each wavelet capturing energy at a specific frequency and a specific direction. It captures both local orientation and frequency information from a digital image. The proposed system has to perform the following retrieving task: finding the desired image, or images, from a large collection (database) using an initial example image. Options for providing query images to the system include: 1) A preexisting image may be supplied by the user or chosen from a random set.2) The user draws a rough approximation of the image he is looking for. The content-based image retrieval technique operates in a number of steps, working as follows. In the first step, the initial image is compared with the ones from the collection. The most relevant images, representing those having a similar visual content with the example image, are extracted and displayed. If the desired image(s) can be found in this retrieved set, then the searching process ends. Otherwise, the feedback mechanism is used. The image that is closest to the desired one, in terms of content similarity, is interactively selected from the displayed set and becomes the new input. The retrieval process performs these operations at each step, until the user makes a final decision. He could reach his goal and find the desired image or decide that the respective image cannot be found in that database, and he should abandon this search. An optimal moment to stop this relevance feedback based retrieval procedure is when the current query image becomes more similar to the desired image(s) than all images retrieved in that step. Then, the input image can be considered the desired one by the user, or no image from the collection is acceptable to him. At the heart of the system is an image retrieval device, which takes input image, sends it to feature extractors, receives back the extracted features, then looks for the K similar images in the image database, receives those images and outputs them, and receives the feedback, and repeats until stopping condition is met, i.e all the images are relevant.

## 2.2 Improved HOG

This paper introduces a content retrieval algorithm based on improved HOG. The method has two steps which are adjusting the HOG structure by scanning the image with a sliding HOG window and reducing feature dimension by principle component analysis (PCA)technique. The experimental results show that: the precision rate of this method has improved significantly compared with the method of transforming the size of images to calculate HOG feature

and the method of extracting color feature. The experimental results show that: the precision rate of this method has improved significantly compared with the method of transforming the size of images to calculate HOG feature and the method of extracting color feature. The first method is transforming the size of images. HOG has no scale invariance, so this way will damage the original gradient information. The second method is improved HOG. This method combines sliding window mechanism with the feature of HOG. The method can extract the feature of different sizes of images, and avoid the error produced by transforming the size of image. Improved HOG feature will increase the feature dimensions, then increase amount of calculation and retrieval time greatly. In order to keep feature vector dimensions consistent and reduce the amount of calculation, the proposed algorithm reduces feature dimensions by PCA technique. HOG descriptor depicts the target information using gradient feature and edge feature. The HOG detector window is tiled with a grid of overlapping blocks which is divided into smaller square unit (cell). In practice the histogram of each cell is achieved by accumulating a local 1-D histogram of gradient directions or edge orientations over every pixel within the cell. The combination of these histograms represents the HOG descriptor. To construct the HOG descriptor, pedestrian images are used to detect humans, by training 64*128 pedestrians' images, using 64*128 detector window to detect human from complex environment. Image retrieval needs the feature information of the whole image. The 64*128 detector window cannot meet the request of image retrieval system including images of various sizes. Key steps followed are : 1) Transformation of image size 2) apply improved HOG.

### 2.3 Dataset

We have used a dataset of dog breeds taken from udacity, where the original purpose was to build an image classifier of dogs. It has 133 labels or different dog breeds, with larger data spread into different folders to training, validation and testing. The training set has 6681 images of different dog breeds, 835 images for validation, and 836 test images.

## 3 Process

### 3.1 Data Collection

The data was collected and stored in the format of train, test and valid. It is a standard dataset. We tried our hands on different datasets, initially had a look at CIFAR10 dataset, however the size of the images in it are to small for human viewing, even though it is a huge database with a lot of images and only 10 classes. There was no direct database available for CBIR, hence we used a standard dataset usually preferred for image classification tasks to build this CBIR system.

Figure 1: Gabor Kernel used for this project

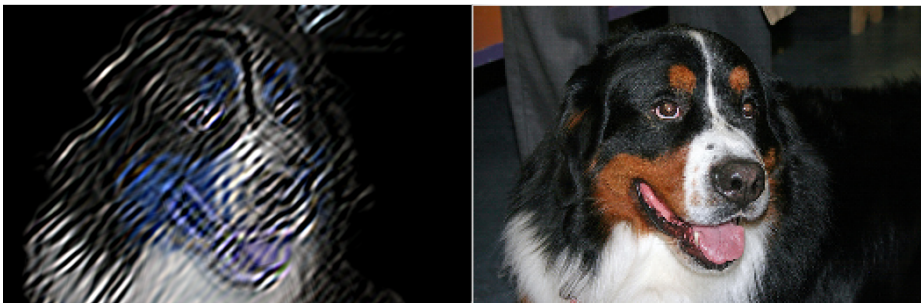

Figure 2: Texture obtained after applying Gabor Filter

### 3.2 Gabor Filter

We use a pre-trained model for segmenting the dog from the images. This was done to retrieve images based on only the features of dog in it, while ignoring the background. We pass each of the images in the training set, which is segmented out and its resized to 128 by 128 image and its mask returned. We apply a gabor filter kernel(See figure 1) of size 21 by 21 onto it, a standard deviation of 8.0 of the gaussian envelope, a phase offset of 0, an orientation of normal to parallel stripes of gabor function by 45 degrees, a wavelength of sine function is set to 10. The resulting texture after applying the mentioned filter is shown in Figure 2. All the features are stored in a dictionary for each images, as a result a feature dictionary is built and stored using python's pickle library. For a query, we take as input a query image, segment the image and calculate its gabor filter features. We then calculate the euclidean distance between the input query image and all the images in the dataset using the gabor features. We then sort

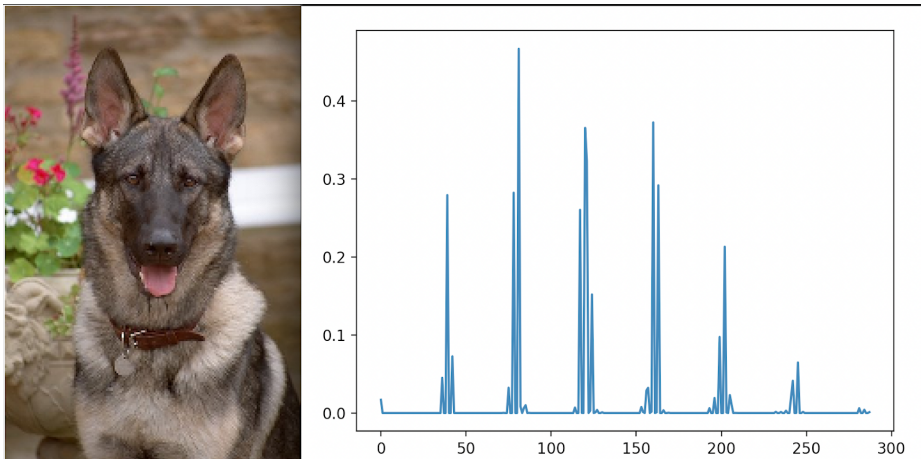


Figure 3: Color Histogram for an image of dog

it, and return the image ids with the nearest distances to the query image. As a result, we get the images with the nearest gabor features to the query image.

### 3.3 Color Histogram

In a similar fashion as mentioned in the above process, a segmentation module is used to get the mask for segmenting the dogs from the images and resized. This helps us to retrieve images based on the color histogram of only the dog while ignoring the background. The segmented dog is then passed to a Feature descriptor which captures the color histogram of the masked region of the image and that calculated histogram is then normalized to have scale invariance. The resulting histogram is shown in Figure3 We pass the training images, segment the dogs, extract their color features and store it in a dictionary, which is then pickled to have a color feature space of all the images in the dataset.

When it comes to querying, we pass an input image, segment the dog in the image, and calculate its histogram and normalize it. We then pass load the pickled model of features, iterate through the entire feature set in the dataset, and calculate the euclidean distance of the color features of the input image with each image in the dataset. We then sort the distances, and return the nearest images to the features of the input image. As a result, we retrieve the images closest to the input image in terms of the color histogram features.

### 3.4 History of Oriented Gradients

We have implemented a simpler version of the HOG feature extraction. Similar to the previous two approaches, the image is segmented to get remove backgound from the image and the features a re extracted only for the dog present in the image. The features are extracted by using the HOGDescriptor class provided

by the OpenCV library. The features are extracted for all the images in the training dataset and stored in a pickle file. The HOG features are calculated for the given query image and compared with the features stored in the pickle file. The images with the least Euclidean distance between their features and the query's features are displayed in the result.

### 3.5 ResNet50

We take a different approach in case of the neural network, pytorch library is being used to fetch a trained ResNet50 image classifier trained on the Imagenet dataset. Before passing our data for training, each image goes through a set of random transformations, in other words we augment our dataset by random rotations, horizontal flips, cropping, resizing. We then retrain the fully connected layers of the model, on our dog dataset, for 4 epochs with Adam optimizer and cross entropy loss. During the training process, the training and validation loss is noted, we achieve a loss of 0.503 on the validation set after the final epoch. The model when tested on our test set, achieves an accuracy of 84 percent. We can then, pass our query image, which would then classify the image into one of the 133 classes, and return sample images from the dataset belonging to that class.

## 4 Learnings

This project helped us understand some of the feature extraction techniques such as features based on color histogram, texture based features, history of oriented gradients. Learned to use Google Colab to train neural network models, and use a pre-trained tensorflow model to segment dog from the image. Learned to use pytorch to train the neural network and apply transfer learning by training the fully connected layers of the model.

## 5 Results

### 5.1 Color Histogram

As it is shown in Figure 4, we can get images of dogs with similar colors using the histogram of colors.

### 5.2 Gabor Filter

Features based on texture of the image seems to be similar to features based on edges. As you can see in the result obtained for images retrieved based on texture generated using Gabor filter in Figure 5, the positions and postures of the dogs are similar.

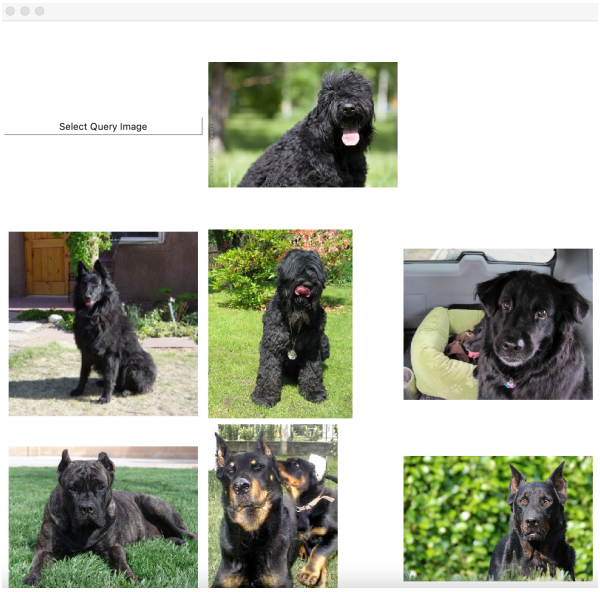


Figure 4: Result for image retrieval based on Color Histogram

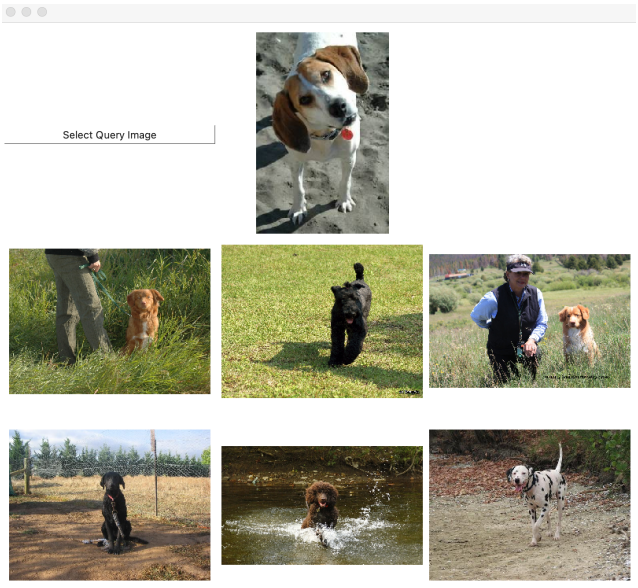


Figure 5: Result for image retrieval based on texture obtained after applying Gabor Filter

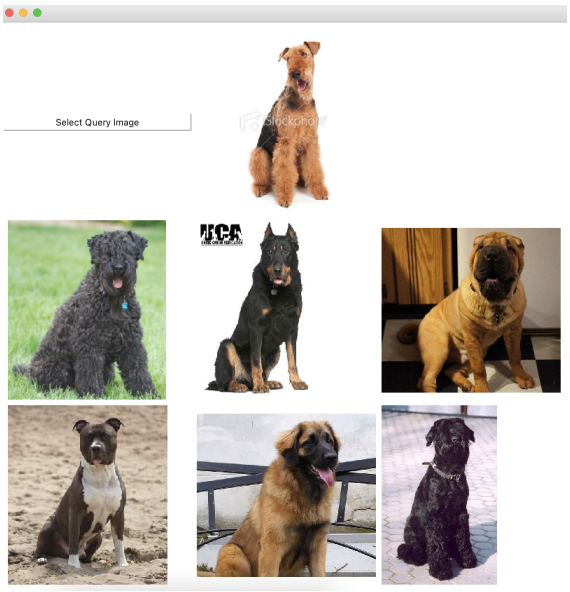


Figure 6: Result for image retrieval based on HOG

### 5.3 History of Oriented Gradients

This image retrieval system also retrieves images based on the positions and postures of the dogs, similar to the texture based image retrieval system mentioned above, since it takes into account the orientations of edges. But, as you can intuit from the Figure 6, the performance of HOG based features seems to look a little better when compared with the features obtained from Gabor filter.

### 5.4 Resnet50 NN

The image retrieval system using a Resnet50 performs better than the other image retrieval system with previously mentioned feature extraction techniques. The result for this system is shown in the Figure 7. The images of dogs retrieved are very close to the image of dog in given query.

## 6 Conclusion

It is possible to build a content based image retrieval system(CBIR) based on different types of features. For retrieving images based on their color component, histogram of colors can be used. For retrieving images based on edges, Gabor filter or HOG can be used to extract features and compared using Euclidean distance to find similar images. To obtain more precision in similarity, one can use a Neural Network, like Resnet50, to extract features and classify the images.

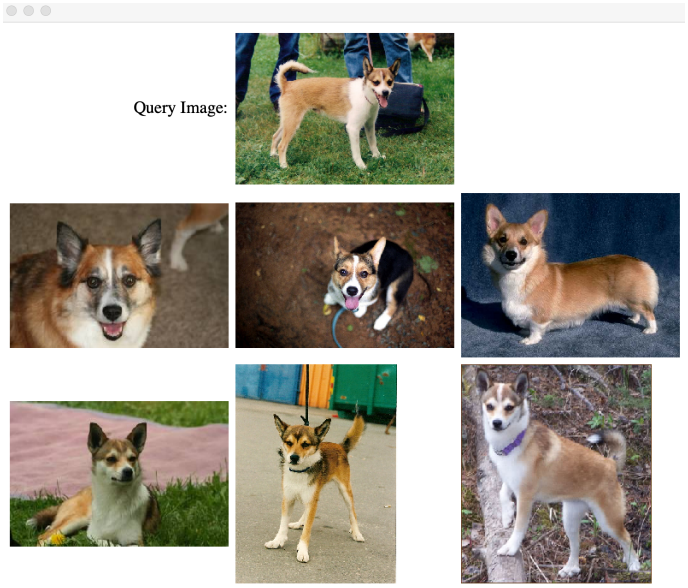


Figure 7: Result for image retrieval using Resnet50 NN

# References

[1] T. Barbu. Content-based image retrieval using gabor filtering. In *2009 20th International Workshop on Database and Expert Systems Application*, pages 236–240, 2009.

[2] A. N. Fierro-Radilla, M. Nakano-Miyatake, H. Pérez-Meana, M. Cedillo-Hernández, and F. Garcia-Ugalde. An efficient color descriptor based on global and local color features for image retrieval. In *2013 10th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pages 233–238, 2013.

[3] S. Pan, S. Sun, L. Yang, F. Duan, and A. Guan. Content retrieval algorithm based on improved hog. In *2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence*, pages 438–441, 2015.

[4] C. Su, H. Chiu, and T. Hsieh. An efficient image retrieval based on hsv color space. In *2011 International Conference on Electrical and Control Engineering*, pages 5746–5749, 2011.