University of Science and Technology of Hanoi

Bachelor Thesis

Image Indexing Methods

Authors:
Nguyen Quoc Thong
Information Communication Technology
BI9-214

Supervisor:
Dr. TRAN Giang Son, USTH ICT Lab

Requirement for obtaining the degree of Information and communications technology

Agreed to submit for defense

Trần Giang Sơn

Hanoi - July, 2021

Contents

Abstract

The 21st century came up with a rising in technology, amount of data over the world increases at a surprising speed, including image data. Digital images have become more popular, and the number of images taken had been rising over and over for the last decade. The medical industry reckons images are one of the most important for learning and study for better insight. To be able to categorize images with the help of technology is nothing we can not pass by. Data over the world have a large portion that comes from the image. The biggest problem with digital image data comes from how to categorized different images into the same section in data. With the same image stored in indexed storage, it is easier to access the image needed. The goal of the image indexing system is to find similar images and store them together to be easier to access. Furthermore, the rising of deep learning also brings new ways to make the indexing system more precise. For this report, we will cover the basic principles of the Image indexing methods and the way to implement a Content-based image retrieval system for the dataset of the U-lake system.

Index-term: U-lake system, Content-based image retrieval, deep learning, Image indexing methods.

Acknowledgements

After complete this dissertation, first of all, I would like to express my appreciation to Dr. Tran Giang Son, my supervisor for his constant guidance throughout the internship. Furthermore, I want to thank the team (Hieu, Thiet, Thao, Phong, Hoang) and Mr. Hiep Le Nhu Chu for helping me to complete my project. Undoubtedly without their help, I can not complete my project during the internship.

Without the knowledge I have been learned at the USTH, I can not have enough ability to complete this project, I want to show my most appreciation to my teachers at USTH, especially from the ICT lab.

My special thank to Vu Dinh Anh of the ICT departments for helping me with my stuck of the project, who not hesitantly support me with many problems I came up with during the internship.

Last but not least, I would like to give my special appreciation to my family. Without their help, I can not finish three years of education at USTH and complete my dissertation.

List of Figures

List of Abbreviations

USTH	University of Science and Technology of Hanoi
ICT	Information Communication Technology
CBIR	Content-based image Retrieval
CSV	Comma-separated values
RGB	Red, Green, Blue
HSV	Hue, Saturation, Value
U-Lake	USTH datalake
TBIR	Text-based image retrieval
PWT	Pyramid wavelet transform
TWT	Tree wavelet transform
DFS	Distributed File System

1 Introduction

1.1 Context and Motivation

Data lake is the collection of single-purpose data mart build using big data technology, a designed data warehouse with low technology and high scalability with self-support service for users able to find and use data set to use without the help of deep knowledge of Information technology. Data lake also supports data even without require a project to gain information.[1]

U-lake is a data lake system create to support teaches and students at USTH with the ability to store and acquire information for learning and teaching.

The U-lake system consists of 5 main components:

- U-lake Core (Database, Metadata enricher, File system, HTTP interface)
- U-lake Crawler.
- U-lake Web dashboard and API.
- U-lake Query.
- U-lake Fuse.

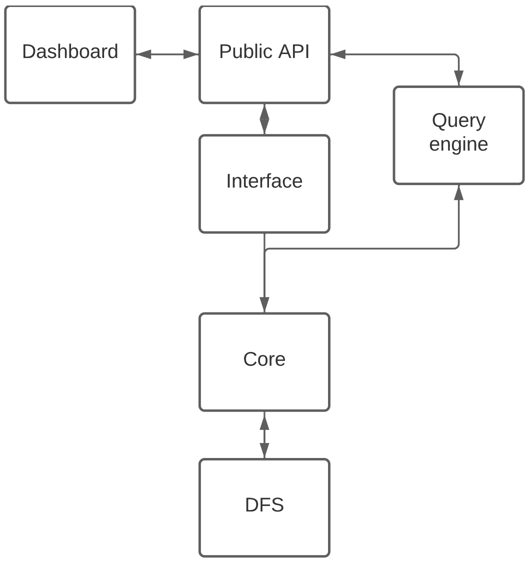


Figure 1: U-lake Architecture

The image indexing system takes part in U-lake Crawler and U-lake Query. The image crawled from the Crawler after being stored in the database indexed and stored as labeled images. When the query system proceeds, the image queried will be analyzed and stored with similar images stored in the database.

With the large amount of data store in U-lake, it is impossible to find an image with a similar object by manual searching. The Image indexing system's purpose is to help people to categorize images. And from that helping people easier to find similar images of the query image. The most common and efficient method used for image indexing is Content-Based Image Indexing which categorized images from features acquire from images.

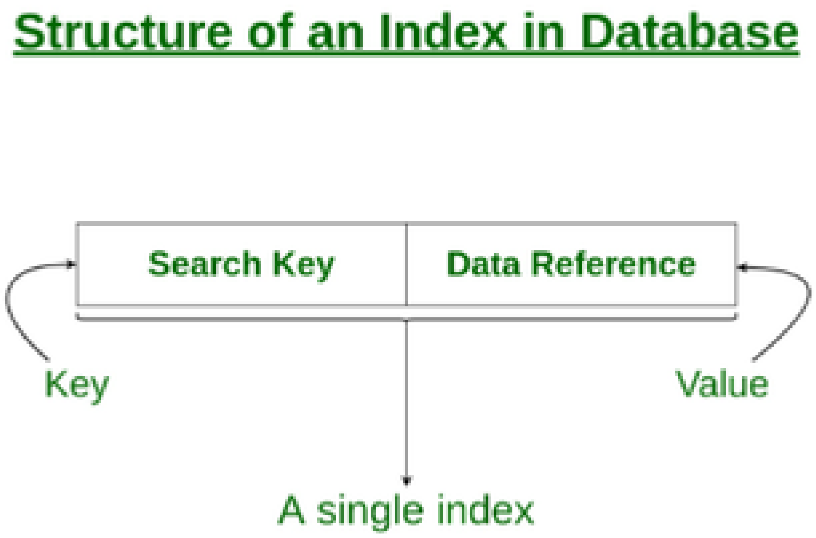


Figure 2: Principals of indexing[2]

1.2 Objective

The objective of the dissertation is to create an image indexing for the Data lake which support image which categorized when downloaded and found with key-labeled after the images queried.

In this Thesis, we also brought out the basic principles of the image indexing system. Moreover are several methods and algorithms are mainly used to create an image indexing system.

1.3 Structure of Thesis

With the first part of the Thesis are used to introduce the Thesis, the rest of the thesis can be organized as below:

- Part II introduction about the image indexing methods, which mainly talk about Content-based image indexing (CBIR). The first part is to gather features of all images in the data set, in this matter, the data is crawled and stored in U-lake. The second part of the CBIR is using data mining to find similar images in the data set using K-mean clustering methods.
- Part III will provide the proposed structure to build an Image indexing system using Content-based image Retrieval.
- Part IV gives the information about how the system had been implemented.
- Part V and VI will sum up what I have achieved with the project during my internship. Furthermore are what can be done in the future to increase the performance of the system.

2 Background

2.1 Image Indexing Definition

Image Indexing or Image Retrieval is a computer system searching and finding similar images from a database consist of digital images.[3]

The two most known image indexings are Text-based image retrieval and content-based image retrieval. However, the TBIR requires the database to be annotated, describe manually making the TBIR system unideal for the project.[4]

- Part I : Feature Extraction. Using different methods to extract different features from images such as colors, shapes, textures, edges.
- Part II : Index. Using machine learning methods to categorized images from features extracted.

2.2 Content-Based Image Retrieval(CBIR)

Text-based image indexing systems mostly consist of manually labeled images which consume a lot of time and hard work. A content-based image system that is harder to classify the images but does not need manual data preprocessing used for this project.[4]

CBIR is known for a different approach to the project with have multiple ways to extract the feature of the image with color, texture, shape, edge, layout. The system also has many ways to index the information extracted from the image. Two of them can be named as the K-mean clustering algorithm and mean shift algorithm.[5]

Input: The input of a typical Content-based image indexing system is images, for upload images into the dataset to extracted features and indexing and for query images.

Output: Image Indexing/Retrieval system output are Images similar to the image queried and have the images indexed in the database.

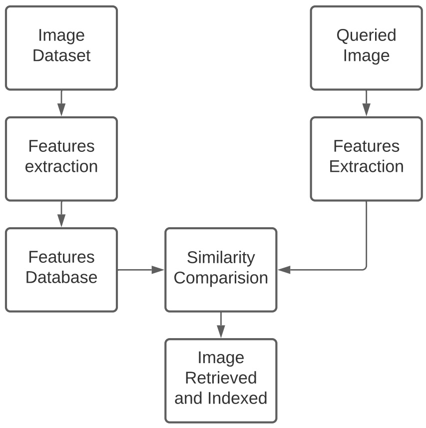


Figure 3: principals of content-based image indexing

2.2.1 Feature Extraction

For choosing which Feature will be extracted from images. For example, the most popular features used in a CBIR system are the color, texture, shape of the image.[5]

Edge feature extraction - Sobel algorithm: Edge feature extraction focus on finding regions that have a change in intensity in an image. The big color change can be shown as steep change and shallow change.
The image will represent as a m x n matrix and project through a filter called Sobel mask, a 3x3 matrix used to find change in the x-axis and y-axis on the image matrix are the edge of the image.[6]

-1	-2	-1
0	0	0
1	2	1

(a)

-1	0	1
-2	0	2
-1	0	1

(b)

Figure 4: Sobel mask[7]

Color feature extraction - Image color histogram : Image color histograms are the presentation of the distribution of colors in an image.

As image's color histogram represent in a bar graph which shows a number of pixels throughout an area.[5]

Image color histogram can be acquired from various types of color space like RGB, HSV, and HSL. A common method to build color histogram data from an image is to divide the normal image into small portions which are bins, then we counting the numbers of pixels contained in each bin and created a histogram of the color span over an area, which in this case was bins.[5]

		red			
		0-63	64-127	128-191	192-255
blue	0-63	43	78	18	0
	64-127	45	67	33	2
	128-191	127	58	25	8
	192-255	140	47	47	13

Figure 5: 2-dimensions histogram of images

2.2.2 Indexing Methods

As image indexing system always contain two parts. The first part is Feature extraction has been explained above, the other part is Image indexing which uses to categorized images in the data with the features extracted from the image.

The most used indexing methods are usually the Mean-shift algorithm and K-mean clustering algorithm.

Mean-shift algorithm Mean shift is a non-parametric feature-space analysis technique for locating the maxima of a density function, a so-called mode-seeking algorithm.[8]

Mean shift algorithm can be easily understandable with the following step:

- Step 1: Randomly assign the region of interest with all the observation inside it belong to the cluster.
- Step 2: Calculate the the mean of the region of interest with all the observation between.
- Step 3: Move the centroid of the region to the mean calculated in step 2 and repeat step 2-3 until the centroid no longer move.

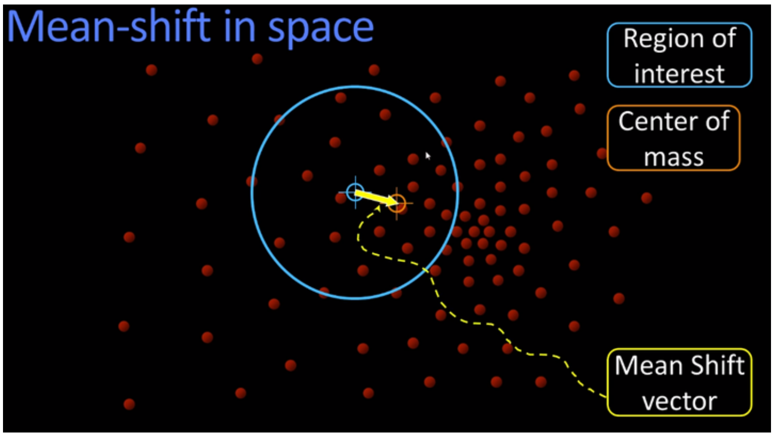


Figure 6: Mean shift algorithm principle [9]

K-mean clustering algorithm K-mean clustering method is a method of clustering, which first assigns a number of observations as the centroid of clusters. All other observations will compare distance with each centroid, and belong to the cluster with the nearest mean. After calculated the centroid of the clusters, observations will be partition into the cluster with the nearest mean again and repeat until no observations change after an iteration.[10]

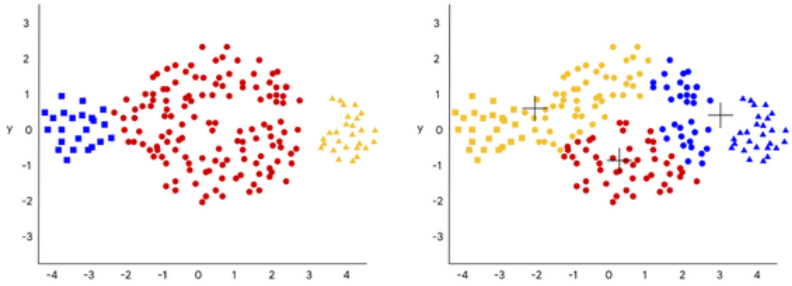


Figure 7: Principles of k-mean clustering algorithm

2.3 Image indexing system overview

Over the last ten years, many image indexing systems had been introduced and used widely throughout the internet. The most usually used system known is Tineyes image search engine and Google reverse image retrieval system.

Tineyes image indexing system: Reverse image retrieval engine find image from image's fingerprint which is images feature matrix represent in form of 64-bit number fingerprint and compare with other images fingerprint to find similar images. From the fingerprint, even rotation or resize version of the images can be found.[11]

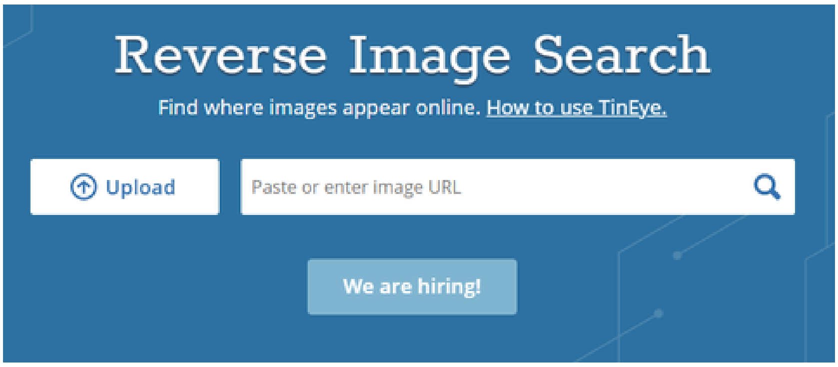


Figure 8: Tineye Search engine [11]

Google image indexing system: Reverse image retrieval system allows users able to submit an image or and Url contain an image and using its algorithm to analyze the mathematical information of images(digital form of the features extracted) to bring out similar images.[12]

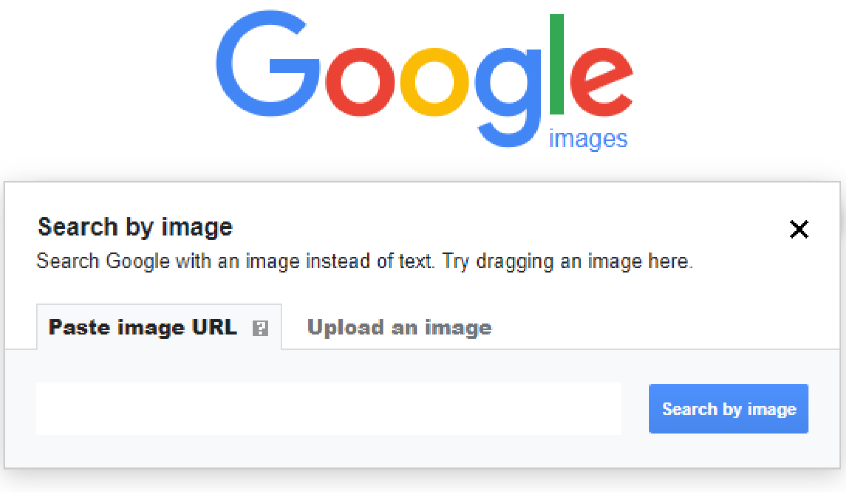


Figure 9: Google Search engine

Image Retrieval System of Naveena, A.K and N.K. Narayanan: The system builds using features extracted from the color, texture, and shape of the image.

- Color: extracted by counting the moments of color in L*a*b space.
- Texture using wavelet, decompose the image into four sub-band frequencies and obtain two main wavelets to analyze: pyramid structure wavelet transform (PWT) and tree-structured wavelet transform (TWT)
- Shape: using a Canny edge detector, applies a high and low threshold to the gradient. The edge extracted will act as a shape histogram in this system.
- Indexing method: SVM classifier will separate the data or support vectors. Using kernel tricks to choosing those hyperplanes with maximum margin.[13]

3 Methodology

3.1 Use-case Model

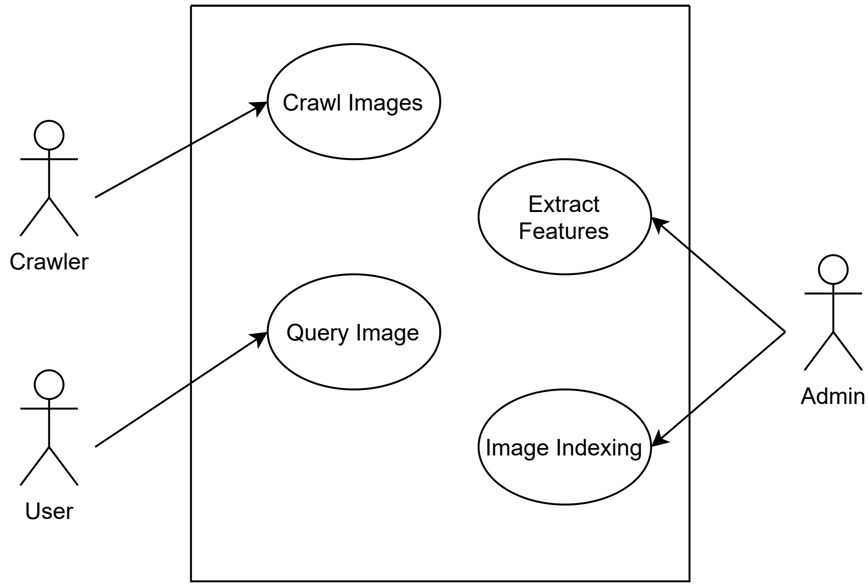


Figure 10: Use case diagram

Upload Image Dataset This use case allows the admin of the database to upload images and have their features extracted and stored in the database.

1. The admin upload the images to the database.
2. Images features being extracted.
3. Images's features stored in the database

Query Image This use case allows people access to the system can query an images and have it's features extracted then compare to the features data stored to index the image.

1. The user query an Image.
2. Image features being extracted.
3. Index the image to the most similar images set.

3.2 Sequence Diagram

3.2.1 Indexing image

The use-case is when the admin proceeds to extract features and indexing images stored in the database. First, the image indexing will Ul receive the command then the control object will receive image information from the database and proceed to extract feature. When the admin starts indexing the image, the UI will send a message to the object system and take the feature data from the database to the indexing image. After finishing indexing, the database will return a notification to the admin.

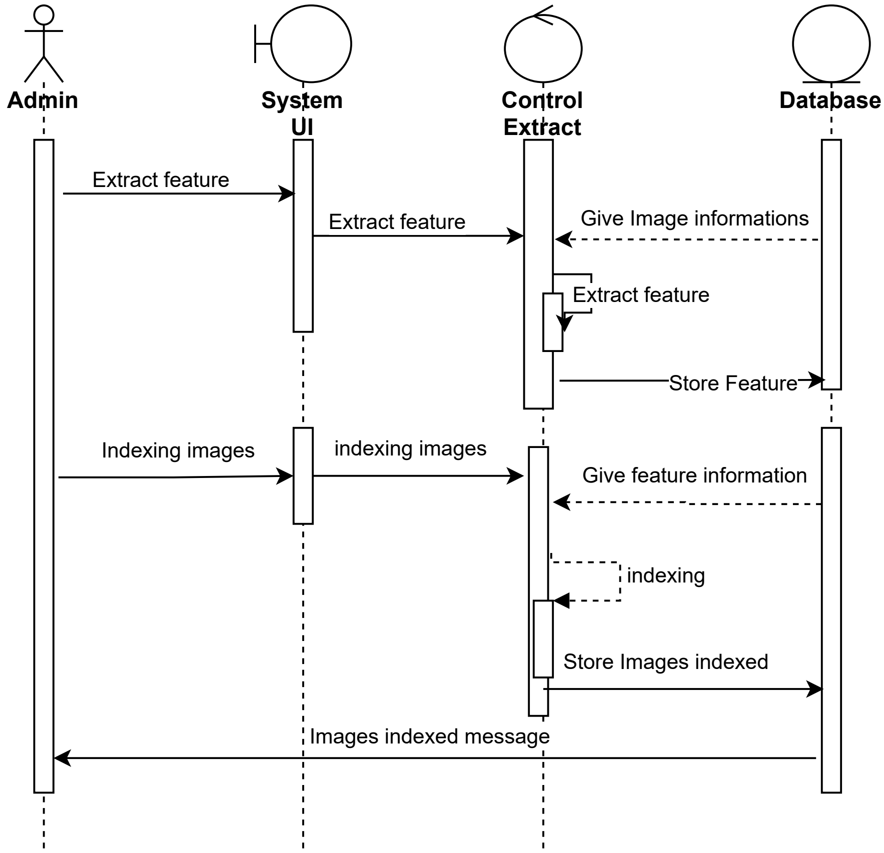


Figure 11: Use case diagram

3.3 Dataset preprocessing

For color feature extraction, the size of the image is important, Since the major idea of the color histogram was to count the pixels in each bin of different images, and for a different size, the images come with a different number of pixels with exact but have smaller size image, therefore the bins of a larger image might contain more pixels than the bins of the smaller image.

The images pixels count from different channels, with the numbers of bins are 4. each bin has intensities from 0-63, 64-127, 128-191, and 192-255. Each channel is a combination of RGB spaces with different bins. For example, the First channel, counting all the pixels of color which have intensities of all three

26515	15	0
7475	12773	0
0	0	0
850	477	0
0	17436	0

Figure 12: Example image color histogram (256x256)

16710	8	0
4607	7616	0
0	0	0
518	266	0
0	10275	0

Figure 13: Example image color histogram (200x200)

R, G, B from 0-63. The following channel is where R, G still have intensities from 0-63 but B intensities are 64-127 and keep go on.

The figures above showed the number of pixels counted from the first 15 channels of 64 channels of the images.

With the large number of images stored in the data with different sizes, we need to resize all the images into the same size.

Since the dataset contains different images with different sizes, in order to have the same size for all the images in data, instead of re-scale for each image, we set a standard size for all the images.

3.3.1 Color Histogram

Histogram is a type of representation data in shape of a bar-graph as x-axis represent to the area and the y-axis represent to the data used to count.

Color histograms are the number of pixels counted in the image represented in bar-graph, the image was separated into a number of bins to represent the area used for easier to compare between images as the x-axis and for y-axis are the number of pixels counted.[5]

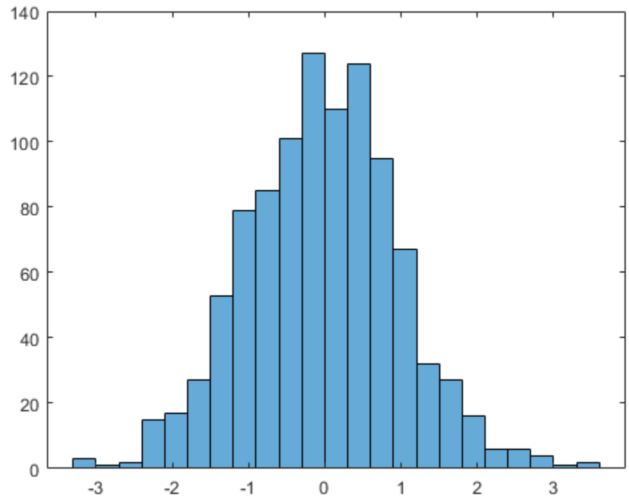


Figure 14: Histogram Example [14]


```
9819,381,0,0,210,53,0,0,0,0,0,0,0,0,0,0,892,236,0,0,10245,12930,309,0,808,1729,45,0,0,0,0,0,0,0,
12869,314,0,0,2183,1883,16,0,0,0,0,0,0,0,0,2,0,0,0,5491,14936,689,0,4,3149,5063,3,0,0,0,0,0,0,
8037,4,0,0,3456,82,0,0,0,0,0,0,0,0,0,0,645,3,0,0,4479,14436,13,0,58,2251,74,0,0,0,0,0,0,0,0,0,
5347,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,7482,2,0,0,6314,3557,1,0,0,0,0,0,0,0,0,5,0,0,0,3270,9098,0,
3120,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5924,0,0,0,1507,601,0,0,0,0,0,0,0,0,0,25,0,0,0,14354,3781,0
1462,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2271,3,0,0,2433,5338,0,0,0,0,0,0,0,0,0,0,0,0,0,53,7495,1,0,
7269,0,0,0,2,15,0,0,0,0,0,0,0,0,0,0,8750,155,0,0,10236,11561,264,0,0,26,142,0,0,0,0,0,6,252,0,0,
6313,123,0,0,4814,1957,640,0,1,0,9,0,0,0,0,0,3019,35,0,0,9997,12344,376,0,9,3754,2054,75,0,0,0,0
222,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,249,0,0,0,1344,62,0,0,0,0,0,0,0,0,0,0,0,0,6136,379,0,0,272
3758,1606,0,0,84,1730,0,0,0,0,0,0,0,0,0,0,200,232,0,0,371,22266,4578,0,0,2420,3337,0,0,0,0,0,0,1
10213,4192,0,0,5345,590,0,0,884,0,0,0,0,0,0,0,73,640,0,0,1195,4701,417,0,11846,2913,183,0,21,7,0
52957,50,0,0,2,27,0,0,0,0,0,0,0,0,0,0,185,11,0,0,386,2186,76,0,0,6,49,0,0,0,0,0,38,13,0,0,3,649,
1582,62,0,0,1,56,8140,0,0,0,0,0,0,0,0,0,2573,12,0,0,1650,1402,42179,15,0,10,329,53,0,0,0,0,5,0,0
1881,0,0,0,13,0,0,0,0,0,0,0,0,0,0,0,1280,0,0,0,13997,2067,0,0,3161,1445,0,0,0,0,0,0,8,0,0,0,128,
2149,76,0,0,702,457,0,0,0,0,0,0,0,0,0,0,3510,14,0,0,6058,8326,32,0,7,3076,27,0,0,0,0,0,126,30,0,
1926,14,0,0,9,23,3,0,0,0,0,0,0,0,0,0,261,9,0,0,687,300,17,0,0,1,40,30153,0,0,0,3,16,4,0,0,19,165
947,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,2943,7,0,0,1700,4994,92,0,0,169,564,0,0,0,0,0,687,4,0,0,2726,1
```


Figure 15: Image color histogram stored in CSV

The images will divide into four bins and with R, G, B space, will be 64 channels. Each channel will count the number of red, green, blue (between corresponding intensities) pixels and store them as a 64-dimensional vector into a CSV file as in the figure above. Each row represents each image in the database and each column represents for a number of pixels counted in corresponding channels.

3.4 K-mean Clustering

K-mean Algorithm is used because it is an unsupervised algorithm using a data vector without the need for labeling.[10]

The indexing methods are using a K-mean clustering algorithm to assign the features extracted into a different cluster. Each cluster can be acknowledged as a different category.

The K-mean clustering Algorithm separated into two different steps:

- Step 1: Assign the observation to different Clusters. The centroid of the Clusters will be randomly assigned the space. Each observation will be assigned to the nearest centroid and become an element of the corresponding cluster.

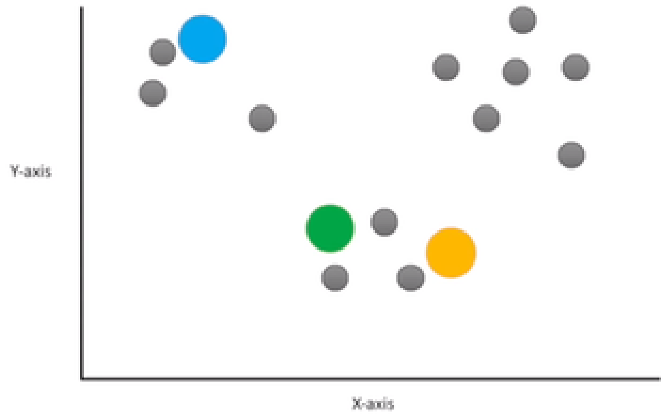
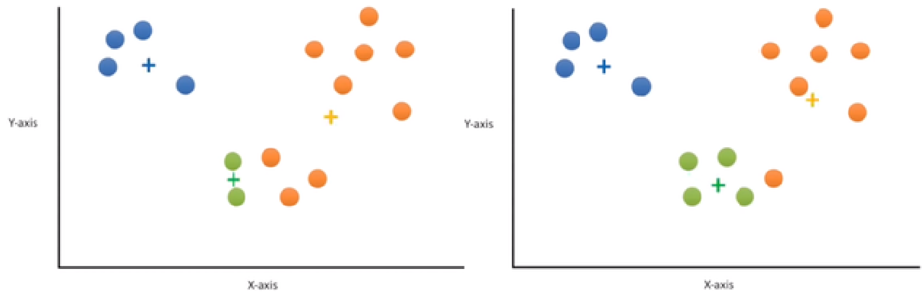


Figure 16: random created centroid of cluster [15]

- Step 2: Recalculate the centroid of the Cluster. After each iteration, clusters will have different number observations. For that, the centroid of the clusters will recalculate. And the number of iteration continues until there will be no more chance of the centroid of all clusters.



(a) Step 2.1

(b) Step 2.2

Figure 17: Recalculate centroid of the cluster [15]

Based on the number of categories of images stored in U-lake, the number of Clusters will be decided by the admin. As USTH is a university, the data of the U-lake may divide images categories depend on the subject and department.

At first, each Vector represents for the Histogram of each image will be acknowledged as a point. We will randomly choose a number Vector to be the centroid of each cluster. Using the Euclidean distance, points can be assigned to the clusters.

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

Figure 18: Euclidean distance [16]

After all the Points has been Assigned, the new centroid of each cluster will calculate again.

After several iterations, the output will bring out as vector type of the color histogram and which cluster that image belongs to and stored in a CSV file.

4 Implementation

4.1 Practical implementation

We will go through what we need did to create a CBIR system, and the main requirements are listed as below:

- Dataset: Consist of images crawled and being stored in U-lake data.
- Dataset used to test the system: https://www.kaggle.com/piyushkumar18/animal-image-classification-dataset
- Language : Java - Main Programming Language.
- Knowledges: basic understand in computer vision, machine learning.
- Methods : Color histogram extraction and K-mean Algorithms.
- Input : Dataset of images with .jpg, .jpeg, .png. This is input for feature extraction part which have output are the list of vector of images in the dataset which also used as the input for the index part of the project.
- Output: CSV file contain all the images histogram and which cluster it belong to.

4.2 Implementation

4.2.1 Image color histogram

As the dataset was stored in a folder, the Histogram class read all the images inside the folder based on the extension to ensure only image files were being read.

Color histogram was a bar graph counting the number of pixels inside each bin of the images representing a fixed intensity value range. The images will be separated into 4 bins and the output will return each bin to have pixels counted corresponding to it.

```
int[][][] ch = new int[4][4][4];
BufferedImage img = null;
try {
    img = ImageIO.read(f);
    for (int x = 0; x < img.getWidth(); x++)
        for (int y = 0; y < img.getHeight(); y++) {
            int color = img.getRGB(x, y);
            int red = (color & 0x00ff0000) >> 16;
            int green = (color & 0x0000ff00) >> 8;
            int blue = color & 0x000000ff;
                ch[red/64][green/64][blue/64]++;
            }
        }
```

After acquired the image color histogram, the histogram stored the data consist of all numbers of pixels counted in 64 bins as a vector in a CSV file to make easy access for indexing.

4.2.2 Indexing

To implement the indexing part, I used the K-mean clustering algorithm, for the first part, then read the histogram data stored in the CSV above.

I decided to separate the clustering into 2 parts, which the first part is only the first iteration of the process. At this step, randomly assign a number of vectors to a number of clusters. As the vector assigned acts as a centroid of a cluster, the system will compare the distance of vectors to each centroid and the vector will become the cluster with the nearest centroid.

```
private static void FirstIteration () {
    for (int k = 0; k < K; k++) {
        Vector<Point> newClusters = new Vector<Point>();
        Point p = new Point();
        Random rand = new Random();
        p = Points.get(rand.nextInt(Points.size()));
        newClusters.add(p);
        Clusters.add(newClusters);

    }
    for (int i = K; i < Points.size(); i++) {
        Point p = new Point();
        p = Points.get(i);
        int index = -1;
        double dist = Double.MAX_VALUE;
        for (int k = 0; k < K; k++) {
            Point centre = Clusters.get(k).get(0);
            double distant = Euclidean(p, centre);
            if (distant < dist) {
                dist = distant;
                index = k;
            }
        }
        Clusters.get(index).add(p);
    }
}
```

After the first iteration, all the vectors have been assigned to the clusters. To continue, the clusters will start re-calculate their centroid again.

```
    for (int i = 0; i < Clusters.size(); i++) {
        Vector<Point> newClusters = Clusters.get(i);
        Point points = new Point();
```

```
            Vector<Double> sumA = new Vector<Double>();
            Double NumberofPoints = Double.valueOf(newClusters.size());
            Point temppoint = newClusters.get(0);
            int Nd = temppoint.getA().size();
            System.out.println(Nd);
            for (int k = 0; k < Nd; k++) {
                double x = 0.0;
                for (int j = 0; j < NumberofPoints; j++) {
                    Point p = newClusters.get(j);
                    Vector<Double> p1 = p.getA();
                    x += p1.get(k);
                } x = x/NumberofPoints;
                sumA.add(x);
            }
            points.setA(sumA);
        }
```

After recalculate the centroid, All the vectors will be compared with all the clusters again to find which clusters they belong to.

The final output will bring out the vector corresponding to each image and the cluster to which that image belongs.

5 Result

K-mean clustering is an unsupervised using non labeled image so to be able to evaluated, figures below used to show the examples of images which belong to the same Cluster. Since the number of Clusters based on the number of categories, Sum of square error(SSE) had been excluded for this project. The SSE in this project is the sum of the distance of all vectors to each cluster.

From the dataset, 100 images was taken to evaluate the accuracy of the System. 100 images was equally taken from 3 categories : panda, chicken and cat. After testing the System, with each categories chosen as each clusters the number of Images of same categories have same clusters are:

- Panda: 16/33 images was in the same cluster with accuracy equals 48.48 %
- Chicken: 22/34 images was in the same cluster with accuracy equals 67.66 %
- Cats: 10/33 images was in the same cluster with accuracy equals 30.30 %

(a) A (b) B (c) C

Figure 19: Panda images of same cluster

(a) A (b) B (c) C

Figure 20: Chicken images of same cluster

(a) A (b) B (c) C

Figure 21: Cat images of same cluster

6 Conclusion

6.1 Conclusion

After the session of the Internship, the system able to finish the purpose of the topic. with the images have been assign to a number of clusters, with understanding about the data stored in the metadata, each cluster can be classified as different content of the images based on the need of the user.

At the current state, the system did not find the similar images of the query image. Instead, we upload the image into the dataset then process extract it feature again with other images and assign the cluster for the image.

6.2 Future work

In the future, the system should be able to show out the images which belong to the same cluster of the example image, also can be used data from another content to be more precise.

Feature extraction The system will able do extracted more features from the images. For example texture of the image which used along with color feature might bring more accurate for the indexing.[17]

Indexing method In near future, the system will have apply more indexing methods to bring out different result to compare the efficiency of the indexing methods.[8]

7 References

[1] Gorelik, A (2019). The Enterprise Big Data Lake: Delivering the Promise of Big Data and Data Science 1st Edition.

[2] Structure of Index in dateabase, Geeksforgeeks. Accessed July 2014 from: `https://www.geeksforgeeks.org/indexing-in-databases-set-1/`.

[3] Datta, R. Joshi, D. Li, Jia and Wang, J, Z.(2008) *Image Retrieval: Ideas, Influences, and Trends of the New Age*, ACM Computing Surveys, vol. 40, no. 2, article 5, pp. 1-60, 2008.

[4] Thuy Ho, amp; Ngoc Ly. (2012). *A scene text-based image retrieval system.* 2012 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). `https://doi.org/10.1109/isspit.2012.6621264`.

[5] Bhute, A. N, Meshram, B. B. *Content Based Image Indexing and Retrieval*(2014). IJGIP 2013 Vol 3 issue 4,Jan 2014. 1401.1742.

[6] Sun, H Tian, S.(2011) *Image retrieval based on blocked histogram and Sobel edge detection algorithm.* 2011 International Conference on Computer Science and Service System (CSSS), 2011, pp. 3277-3281, doi: 10.1109/CSSS.2011.5974765.

[7] Kamble, Parshuram Hegadi, Ravindra. (2015). Handwritten Marathi Character Recognition Using R-HOG Feature. Procedia Computer Science. 45. 10.1016/j.procs.2015.03.137.

[8] Bouker, M. A Hervet, E. (2012) *Image retrieval based on mean-shift clustering using color descriptor*, 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), 2012, pp. 1422-1423, doi: 10.1109/ISSPA.2012.6310521.

[9] Udacity. (2020, Feb 24). Mean Shift in Space [Video]. Youtube. `https://www.youtube.com/watch?v=TMPEujQrY70`.

[10] Amory, A. A Sammouda, R Mathkour, H and Jomaa, R. M. *A content based image retrieval using K-means algorithm.*(2012) Seventh International Conference on Digital Information Management (ICDIM 2012), 2012, pp. 221-225, doi: 10.1109/ICDIM.2012.6360121.

[11] How to use TinEye. Tineye.com. `https://tineye.com/how`.

[12] Kincaid,J (June 14, 2011). *Google Search By Image: Use A Snapshot As Your Search Query.* TechCrunch.

[13] Naveena, A. K., Narayanan, N. K. (2016). Image retrieval using combination of color, texture and shape descriptor. 2016 International Conference on Next Generation Intelligent Systems (ICNGIS). doi:10.1109/icngis.2016.7854023.

[14] Decrease number of histogram bins [Online image]. Mathworks. `https://www.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.histogram.fewerbins.html`.

[15] StatQuest with Josh Starmer . (2018, May 24). StatQuest: K-means clustering [Video]. Youtube. https://www.youtube.com/watch?v=4b5d3muPQmA.

[16] Tabak, J (2014), Geometry: The Language of Space and Form, Facts on File math library, Infobase Publishing, p. 150, ISBN 978-0-8160-6876-0

[17] Hiremath, P. S Pujari, J.(2007) *Content Based Image Retrieval Using Color, Texture and Shape Features*, 15th International Conference on Advanced Computing and Communications (ADCOM 2007), 2007, pp. 780-784, doi: 10.1109/ADCOM.2007.21.