Hash match for similar images

Team Project #2

Course Title	Image Processing
Professor	Choi Kwang Nam
Major	Software
Date	2021. 12. 05
Team Name	YSWL
(KOR)20150413	Yoo Seungshin
(KOR)20151224	Seo Jeonghyeon
(CHN)20191335	Wang ShengPeng
(KOR)20192455	Lee JooYoung

1. Introduction

Data is accessible easily in everyone's life and that amount is also large. We are now live-in period of data and even at this moment, a lot of data is being generated and disappeared. When you search for data in Engine such as "Google" or "Naver", you might have experienced that a lot of junk data is with necessary data.

So, we related this with Image Processing. We tried to find similar images with original image we have. The goal of this project is filter data images to obtain what we need by "Average Hash Matching" and print it.

2. Survey

The most important function used in this project is image matching. Image matching helps determine whether the two images have similar characteristics or are the same by comparing two different images. Image matching is divided into Template Matching and Average Hash Matching. Find the same image using templates, which is prepared images is Template Matching. The function adopted by the project is Average Hash Matching.

The principle of average hash matching is simple. First, calculates the average value of all pixels of the image and if the value of each pixel is greater than the average, it is changed to 1, and if less than the average, changed to 0. Similarity measurements use a method of measuring the distance between two points, and measurements include Euclidean distance and Hamming distance.

Euclidean distance is a representative method of measuring distance. This is often the case for measuring the distance between points and points learned in the mathematical class.

Hamming distance is quite different from the concept generally known. Hamming distance is to check whether the two values of comparison match exactly. For this reason, the condition that the two digits of the two values must be the same in order to use the Hamming distance follows.

Without above two methods, there is also Manhattan distance as a typical method of measuring distance. This is a measurement method that calculates the sum of the difference of each dimension's absolute number.

3. Project process

As mentioned in the introduction, this project is carried out using "Average Hash Matching". Set the specified input image and select the necessary images among various images with junk images.

The order of the project is as follows.

```
int main()
{
    Mat orgImg = imread("C:/OrgImg/grape.jpg");
    imshow("Input image", orgImg);
    moveWindow("Input image", 1000, 0);
    waitKey(2000);

    Mat orgHash = img2hash(orgImg);
    cout << orgHash << endl;

    vector<string> path;
    glob("C://Img/*.jpg", path, false);

    vector<Mat> Img;
```

Read the graph.jpg file to use as input image from the specified path. Data images are not the only one, so use glob function to read images in directory.

```
Mat imgHash = img2hash(Img[i]);

Mat Color;
resize(Img[i], Color, Size(16, 16));
```

```
Mat img2hash(Mat orgImg)
{
    Mat Hash;
    resize(orgImg, Hash, Size(16, 16));

    Mat gray(Hash.rows, Hash.cols, CV_8UC1);
    for (int i = 1; i < Hash.rows; i++) {
        for (int j = 1; j < Hash.cols; j++) {
            gray.at<uchar>(i, j) = (Hash.at<Vec3b>(i, j)[0] + Hash.at<Vec3b>(i, j)[1] + Hash.at<Vec3b>(i, j)[2]) / 3;
        }
    }

    Scalar avg = mean(gray);

    Mat averageHash(gray.rows, gray.cols, CV_8UC1);
    for (int i = 0; i < gray.rows; i++) {
        for (int j = 0; j < gray.cols; j++) {
            if (gray.at<uchar>(i, j) > avg[0]) {
                averageHash.at<uchar>(i, j) = 1;
            }
            else {
                averageHash.at<uchar>(i, j) = 0;
            }
        }
    }
    return averageHash;
}
```

All input and data images have preprocessing, resize to a predetermined size, 16 * 16, and gray scaled. Images which gray scaled, changed to "Hash" in this function. Calculate Average pixel value, and if each pixel value is larger than average value, It mapped to 1, or it mapped to 0. This is main function of hash matching.

```
int distance = hamming_distance(orgHash, imgHash);
if (float(distance) / 256 < 0.25) {
    cout << path[i] << "->" << float(distance) / 256 << endl;
    cout << imgHash << endl;
    cout << float(distance) / 256 << endl;
    imshow(path[i], Img[i]);
    moveWindow(path[i], 0, 0);

    for (int i = 0; i < imgHash.rows; i++) {
        for (int j = 0; j < imgHash.cols; j++) {
            if (imgHash.at<uchar>(i, j) == 0) {
                Color.at<Vec3b>(i, j)[0] = 255;
                Color.at<Vec3b>(i, j)[1] = 0;
                Color.at<Vec3b>(i, j)[2] = 255;
            }
            else {
                Color.at<Vec3b>(i, j)[0] = 255;
                Color.at<Vec3b>(i, j)[1] = 255;
                Color.at<Vec3b>(i, j)[2] = 255;
            }
        }
    }
```

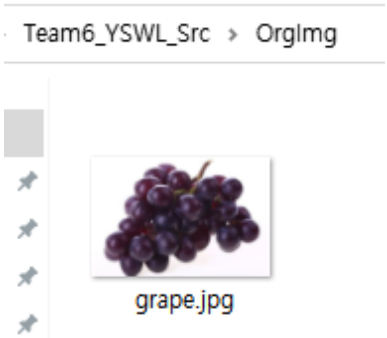
```
int hamming_distance(Mat orgHash, Mat imgHash) {
    int distance = 0;
    for (int i = 0; i < orgHash.rows; i++) {
        for (int j = 0; j < orgHash.cols; j++) {
            if (orgHash.at<uchar>(i, j) != imgHash.at<uchar>(i, j)) {
                distance += 1;
            }
        }
    }
    return distance;
}
```

After "Img2Hash" original input image, grape.jpg and data images in directory "Img" calculate Hamming Distance. We resized size of image to 16 * 16, so one image has 256 pixels. Add for 256 pixels' distance and divide to 256. If it is smaller than 0.25 which means more than 75% pixels have same value. We judged that two images are similar.

And similar images which determined with hamming distance. Input image is grape so, we set new images. If hash matched mapped to purple or mapped to white to show how it changed.

Collecting Images

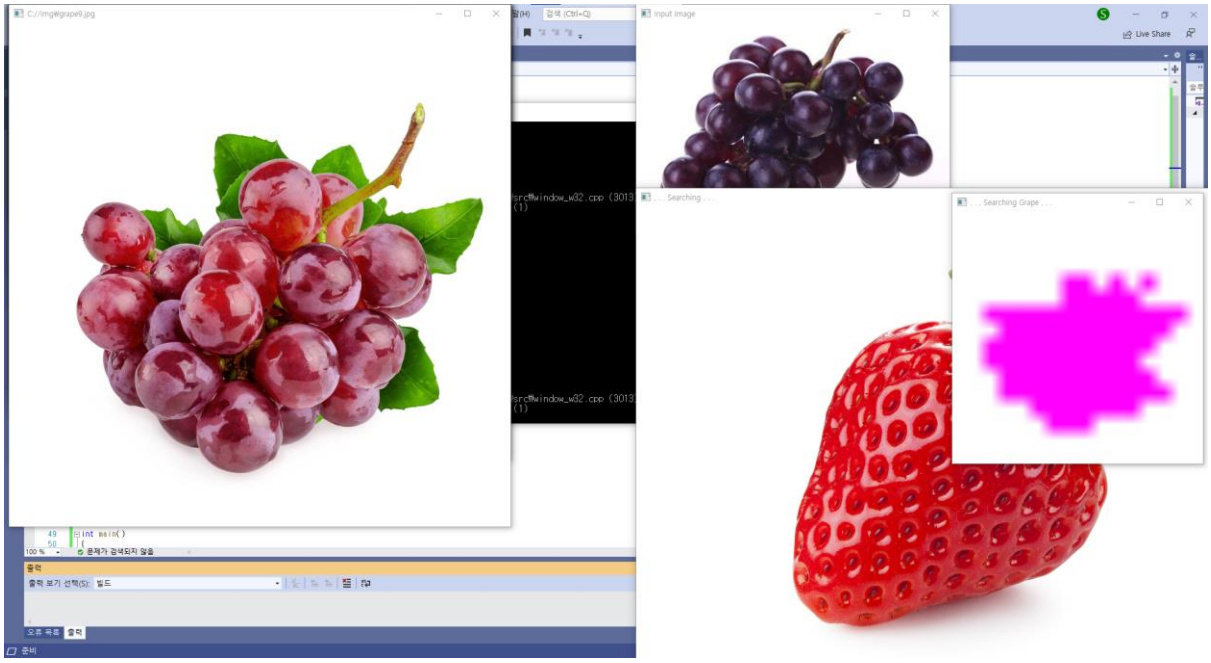
The images required to execute the code is as follows.



This is an input image, "grape.jpg", this image was used as a criterion for determining whether the data images are grapes or not.

Next is data images which are in "Img" Directory. We found some grape images and other fruit images also we googled "grape" and gather images which is not grape shape, and we named these grape_junk.

4. Project Result



It is implemented by comparing the input image graph.jpg at the top right with the data image underneath it while searching one by one, and if it is judged to be similar, there is an output at the top left. The image judged to be similar was resized to large and show purple mapped as output on the right.

There are a total of 29 images used as datasets in the directory, of which there are a total of 11 grapes. It succeeded in detecting 8 out of 11 grapes, and one Dragon fruit was derived as a result of 18 images unrelated with grape.

5. Spending time with minutes(Kakao Talk)

Keep in touch with KakaoTalk anytime whenever we have something to share.

Use English and Korean together for communication between Chinese and Korean.

- 10/28 Started Kakao talk

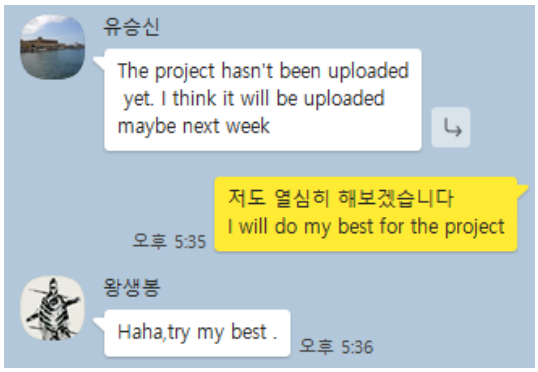
- 11/7 Selected topic as average hash match by using vote

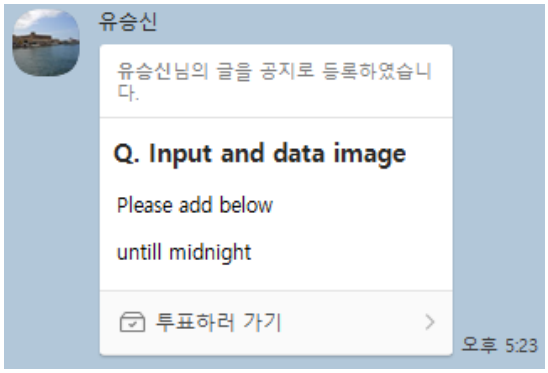
- 11/16 Selected input and data image as fruit by using vote

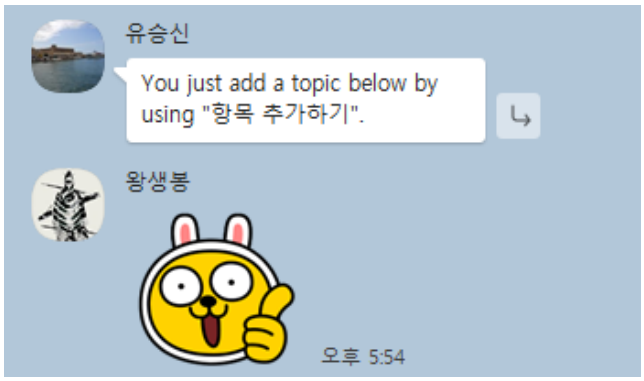
- 11/18 Divided work by using ladder game

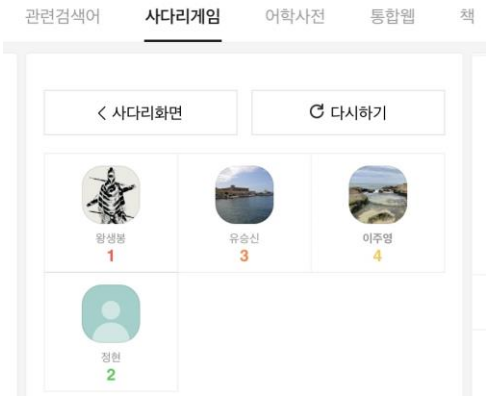
- 11/21 Shared code

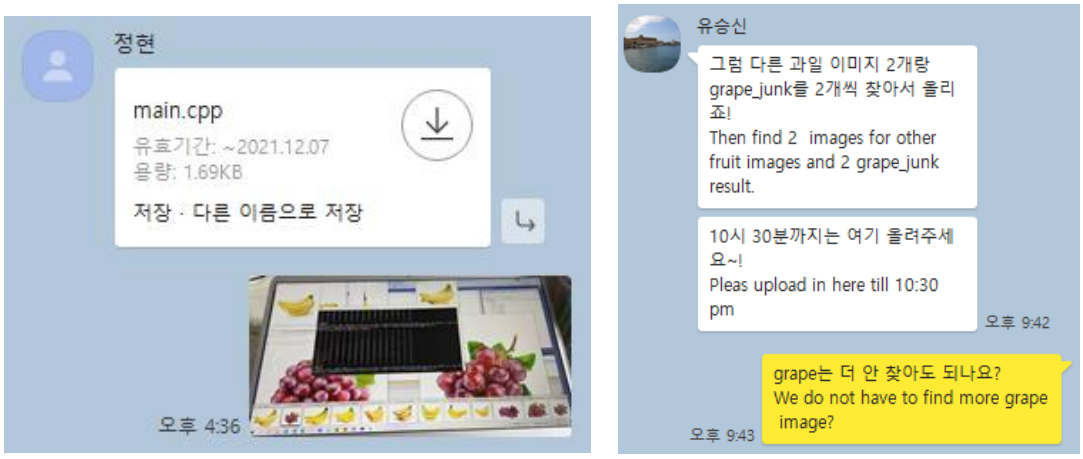
- 11/27 Combined all codes and upgraded and added more images











(KakaoTalk chat captured by Lee Jooyoung)

6. Team's thought

Yoo Seungshin: During project, it was fun to change the results little by little depending on our code. Above all, even though, the team was formed suddenly, I liked this project, which is an interesting topic, with good team members. To be honest, I was a little worried because I was a team leader, but really thank to teammates for participating well.

Seo Jeonghyeon: It was a meaningful time to get to know more about image processing, and it was fun to solve problems together.

Wang ShengPeng: I am very happy to finish this project with my Korean friends. I have learned a lot from this project, and I can experience the power of teamwork. I learned the process of a project from everyone publishing their own ideas, division of labor, and finally publishing. This is very helpful for me personally. Thank my teammates.

Lee JooYoung: It was my first time using English when I do team project. I could learn importance of English and feel globalization. It was surprising when average hash match was good at distinguishing even though the shape of banana and grape were similar. I want to experience more image processing projects next time.