資工所碩一　林群崴

Histogram Equalization

```
#include <iostream>
#include <opencv2/opencv.hpp>
#include <vector>

using namespace cv;
using namespace std;

int main()
{
	//讀圖，以灰階讀入
	Mat lena = imread("lena.BMP", CV_LOAD_IMAGE_GRAYSCALE);

	//先將原圖調暗1/3
	for (int i = 0; i < lena.cols; i++) {
```

```
            uchar *data = lena.ptr<uchar>(i);
            for (int j = 0; j < lena.cols; j++) {
                data[j] = data[j] / 3;
            }
        }

        vector <int> v(256, 0);  //v = {0,0,...0} 用來記錄0~255每個itensity的個數

        for (int i = 0; i < lena.rows; i++) {
            for (int j = 0; j < lena.cols; j++) {
                v[lena.at<uchar>(i, j)]++;
            }
        }

        vector<int> s_k(256, 0);

        //計算Histogram Equalization
        double n = lena.rows*lena.cols;
        double temp = 0;
        for (int k = 0; k <= 255; k++) {
            temp += v[k] / n;
            s_k[k] = temp * 255;
        }

        Mat he(lena.rows, lena.cols, CV_8U, Scalar(0));

        for (int height = 0; height < he.cols; height++) {
            uchar *ref = lena.ptr<uchar>(height);
            uchar *data= he.ptr<uchar>(height);
            for (int width = 0; width < he.cols; width++) {
                data[width] = s_k[ref[width]];
            }
        }

        //存圖
        imwrite("HE.bmp", he);
        return 0;
}
```

首先，先將 lena.BMP 讀入名為 lena 的 Mat 變數，然後循訪每個 pixel，將其原本的亮度調整為 1/3，即把每個 pixel 的值乘除以 3，作為前置準備，使 Histogram Equalization 的效果更好。
接著開一個名為 v 的 256 維 vector，將 0~255 的每個 itensity 分別有幾個記錄下來之後，再另開一個 s_k 的 vector 紀錄套用 Histogram Equalization transformation 的計算結果。
最後將途中每個 pixel 對應的 idensity，藉由查表找到經過轉換的結果，改寫 pixel 的強度，在巡訪所有 pixel 後完成。