

Nama : Indah Hamidah

NIM : G64140055

Nama Dosen : Dr. Yeni Herdiyeni, SSi MKom

Nama Asisten : Kak Hana & Bang

## Lembar Kerja Praktikum 5

1. Download citra car.png pada LMS
2. Baca citra car.png dalam format grayscale

```
#include <iostream>

#include <cv.h>

#include <highgui.h>

using namespace std;

using namespace cv;

int main()
{
    Mat gambar = imread ("car.png");

    Mat imGray(gambar.rows, gambar.cols, CV_8UC1,Scalar(0,0,0));

    //buat grey

    for(int row=0; row<gambar.rows; row++)
    {
        for(int col=0; col<gambar.cols; col++)
        {
            Vec3b intensity = gambar.at<Vec3b>(row,col);

            uchar blue = intensity.val[0];

            uchar green = intensity.val[1];

            uchar red = intensity.val[2];

            uchar Gray = (0.114 * blue)+(0.587 *green) + (0.299 *red);
```

```

        imGray.at<uchar>(row,col)=Gray;
    }
}

imshow("Original", gambar);

imshow("Gray",imGray);

waitKey(0);

return 0;
}

```

3. Buatlah sebuah fungsi contrastStretching tanpa menggunakan fungsi OpenCV pada car.png (Fungsi opencv yang diperbolehkan: imread, waitKey, imshow, dan membaca citra grayscale).
4. Buatlah sebuah fungsi histogramEqualization tanpa menggunakan fungsi openCV dan lakukan proses histogram equalization pada car.png (Fungsi opencv yang diperbolehkan: imread, waitKey, imshow, dan membaca citra grayscale).
5. Algoritme histogramEqualization:
  - Hitung jumlah kemunculan piksel setiap nilai derajat keabuan
  - Hitunglah peluang nilai kemunculan setiap nilai derajat keabuan (normalized histogram)
  - Hitung histogram kumulatif
  - Lakukan histogram equalization dengan rumus sebagai berikut:
6. Tampilkan histogram hasil contrast stretching, histogram citra original, normalized histogram, cumulative histogram, dan equalized histogram
7. Berikan penjelasan singkat terhadap citra hasil histogram equalization dan bandingkan terhadap hasil contrast stretching.

```

#include <cv.h>

#include <highgui.h>

#define BIN 256

using namespace cv;

using namespace std;

//fungsi mencari nilai maksimum pada array

float maxVal(float data[], int n)
{
    float outp = 0;

```

```

    for(int i = 0; i < n; i++)
    {
        if(data[i] > outp) outp = data[i];
    }
    return outp;
}

//fungsi untuk menampilkan histogram
void drawHistogram(float data[], string title, int xScale, int yScale)
{
    //find max value
    float yMax = maxVal(data, BIN);

    //set margin
    float yMargin = 1;

    //set y size
    int ySize = 128*yScale;

    //set x size
    int xSize = 256*xScale;

    //create Mat to draw histogram
    Mat histogram = Mat(ySize,xSize,CV_8UC1,double(0));

    //iterating
    for(int x = 0; x < histogram.cols; x++)
    {
        //histogram view scaling
        int histValue = ySize - ceil(data[x/xScale]*(ySize-yMargin)/yMax);

        for(int y = histogram.rows-1; y >= histValue; y--)
        {

```

```

        histogram.at<uchar>(y,x) = 255;
    }
}
imshow(title+" histogram", histogram);
}

//fungsi untuk ekualisasi histogram
void histogramEqualization(Mat image)
{
    //deklarasi array 1 dimensi dengan ukuran BIN
    float hist[BIN] = {0.0};
    int i,j;
    //menghitung histogram
    for(i=0; i<256; i++){
        hist[i]=0;
    }
    for(i=0; i<image.rows; i++){
        for(j=0; j<image.cols; j++){
            hist[(int)image.at<uchar>(i,j)]++;
        }
    }

    //draw histogram
    drawHistogram(hist, "Histogram", 1, 1);
    //menghitung normalized histogram
    int ukuran=image.rows*image.cols;
    float awal[BIN];

```

```

for(i=0; i<256; i++){
    awal[i]=(double) hist[i]/ukuran;
}

//draw normalized histogram
drawHistogram(hist, "normalized", 1, 1);

//menghitung cumulative histogram
float cumulative[BIN];
cumulative[0]=hist[0];
for(i=0; i<256; i++){
    cumulative[i]=hist[i]+cumulative[i-1];
}

//draw cumulative histogram
drawHistogram(hist, "cumulative", 1, 1);

//menghitung equalized histogram
float equa[BIN], alf=255.0/ukuran;
for(i=0; i<256; i++){
    equa[i]=cvRound((double) cumulative[i] * alf);
}

float equa2[BIN]={0.0};
for(int i=0; i<256; i++){
    equa2[(int)equa[i]]+=awal[i];
}

float equalisasi[BIN];
for(int i=0; i<256; i++){
    equalisasi[i]=cvRound(equa2[i]*255);
}

```

```
}
```

```
//draw equalized histogram
```

```
drawHistogram(equalisasi, "equalized", 1, 1);
```

```
}
```

```
void histContrast(Mat image){
```

```
//deklarasi array 1 dimensi dengan ukuran BIN
```

```
float hist[BIN] = {0.0};
```

```
//menghitung histogram
```

```
for(int i=0;i<256;i++){
```

```
    hist[i] = 0;
```

```
}
```

```
for(int i=0;i<image.rows;i++) {
```

```
    for(int j=0;j<image.cols;j++) {
```

```
        hist[(int)image.at<uchar>(i,j)]++;
```

```
    }
```

```
}
```

```
//draw histogram
```

```
drawHistogram(hist, "Histogram Contrast Stretching", 1, 1);
```

```
}
```

```
Mat contrastStretching(Mat img){
```

```
float mini=255, maxi=0;
```

```
for(int i=0;i<img.rows;i++){
```

```
    for(int j=0;j<img.cols;j++){
```

```
        if(img.at<uchar>(i,j)<mini)
```

```
            mini=img.at<uchar>(i,j);
```

```

    }
}

for(int i=0;i<img.rows;i++){
    for(int j=0;j<img.cols;j++){
        if(img.at<uchar>(i,j)>maxi)
            maxi=img.at<uchar>(i,j);
    }
}

cout << mini << endl;
cout << maxi << endl;

Mat dst(img.rows, img.cols, CV_8UC1, double(0));
for(int i=0;i<img.rows;i++){
    for(int j=0;j<img.cols;j++){
        dst.at<uchar>(i,j)=((img.at<uchar>(i,j)-mini)/(maxi-mini))*255;
    }
}

return dst;
}

```

```

int main()
{
    Mat original;

    // ... baca citra car.png dalam format grayscale
    original= imread("car.png",0);

    // ... buat Mat equalized dengan ukuran sama dengan original, tipe data CV_8UC1
    Mat equalized;

```

```

equalizeHist(original, equalized);

Mat hasil=contrastStretching(original);

histogramEqualization(original);

histContrast(hasil);

imshow("Original",original);

imshow("Contrast Stretching",hasil);

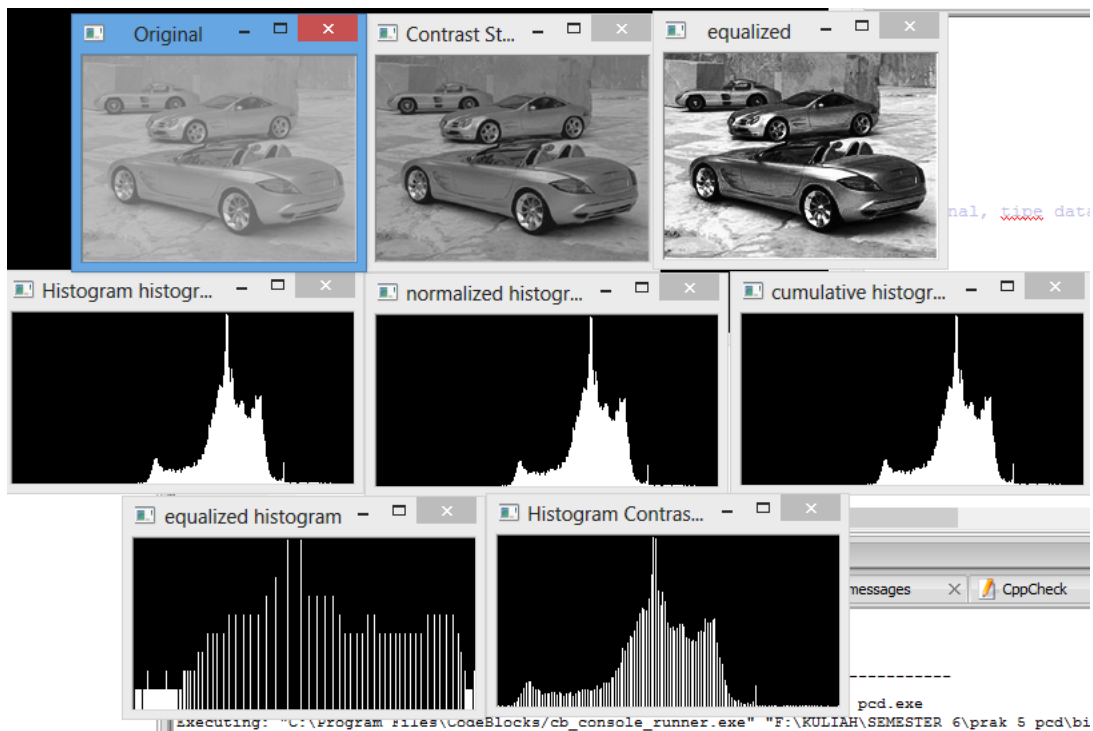
imshow("equalized", equalized);

cvWaitKey(0);

return 0;

}

```



Pada gambar dapat dilihat bahwa equalized lebih tajam, karena equalized menggunakan teknik distribusi nilai derajat keabuan pada suatu citra dibuat rata, sedangkan kontras stretching menambah atau mengurangi kontras dilakukan dengan cara meregangkan atau memampatkan histogram.