Adjust contrast in an Image using Histogram Equalization.

CODE:

#include <opencv2/opencv.hpp>

#include <iostream>

using namespace cv;

using namespace std;

int main(int argc, char\*\* argv)

{

// Read the image file

Mat image = imread("D:/My OpenCV Website/fly-agaric.jpg");

// Check for failure

if (image.empty())

{

cout << "Could not open or find the image" << endl;

cin.get(); //wait for any key press

return -1;

}

//change the color image to grayscale image

cvtColor(image, image, COLOR\_BGR2GRAY);

//equalize the histogram

Mat hist\_equalized\_image;

equalizeHist(image, hist\_equalized\_image);

//Define names of windows

String windowNameOfOriginalImage = "Original Image";

String windowNameOfHistogramEqualized = "Histogram Equalized Image";

// Create windows with the above names

namedWindow(windowNameOfOriginalImage, WINDOW\_NORMAL);

namedWindow(windowNameOfHistogramEqualized, WINDOW\_NORMAL);

// Show images inside created windows.

imshow(windowNameOfOriginalImage, image);

imshow(windowNameOfHistogramEqualized, hist\_equalized\_image);

waitKey(0); // Wait for any keystroke in one of the windows

destroyAllWindows(); //Destroy all open windows

return 0;

}

CODE EXPLANATION:

// Read the image file

Mat image = imread("D:/My OpenCV Website/fly-agaric.jpg");

// Check for failure

if (image.empty())

{

cout << "Could not open or find the image" << endl;

cin.get(); //wait for any key press

return -1;

}

**Explanation:** The above code segment will load the image from the specified file. The program will exit if the image load-up is failed.

//change the color image to grayscale image

cvtColor(image, image, COLOR\_BGR2GRAY);

**Explanation:** The above function converts the image in BGR color space to grayscale color space.

//equalize the histogram

Mat hist\_equalized\_image;

equalizeHist(image, hist\_equalized\_image);

**Explanation:** The above function equalizes the histogram of the grayscale *image*and store the output in the *hist\_equalized\_image.*

//Define names of windows

String windowNameOfOriginalImage = "Original Image";

String windowNameOfHistogramEqualized = "Histogram Equalized Image";

// Create windows with the above names

namedWindow(windowNameOfOriginalImage, WINDOW\_NORMAL);

namedWindow(windowNameOfHistogramEqualized, WINDOW\_NORMAL);

// Show images inside the created windows.

imshow(windowNameOfOriginalImage, image);

imshow(windowNameOfHistogramEqualized, hist\_equalized\_image);

**Explanation:** The above code segment will create windows and show images in them. As windows are created passing the flag WINDOW\_NORMAL, they can be resized freely.

waitKey(0); // Wait for any keystroke in the window

destroyAllWindows(); //destroy all open windows

return 0;

**Explanation:** The program will wait until any key is pressed. After a key is pressed, all created windows will be destroyed and the program will exit.

**IMAGE INPUT:**

****

**IMAGE OUTPUT:**

****