COLOUR DETECTION IN OPENCV

-M.Amutha Bharathi

INTRODUCTION:

OpenCv is a Computer Vision library developed by Intel. It is a collection of C functions and a few C++ classes that implement popular Image Processing and Computer Vision algorithms. Some of the basic image processing capabilities include filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms, image pyramids, contour processing, distance transform, moments, template matching, Hough transform, polygonal approximation,etc.

Here in this tutorial, we will see about a very simple and basic colour detection technique.

POINTS TO REMEMBER:

- An Image is a matrix of pixels
- An image structure has various details of the image like size,channels,origin,etc.
- Each and every pixel is itself a matrix holding basic information like color,depth,saturation,etc.
- For example,a 640 x 480 image has 307200 pixels!

COLOR DETECTION:

Here we take a coloured image of 3 channel (RED,GREEN,BLUE). So each pixel has a RED value,GREEN and BLUE value. Our aim is to retrieve these values from an image to determine the color of the pixel.

EXAMPLE: RED pixel means RED =255,GREEN=0, BLUE=0 in a pure red pixel.However in reality there will be green and blue components too.

Some basic commands.

- Create a window:
  ```
  cvNamedWindow("Color", CV_WINDOW_AUTOSIZE);
  ```
- Load an image:
  ```
  IplImage* img;
  img=cvLoadImage(fileName);
  ```
- Display an image:
  ```
  cvShowImage("Color",img);
  ```
- Close a window:
  ```
  cvDestroyWindow("Color");
  ```

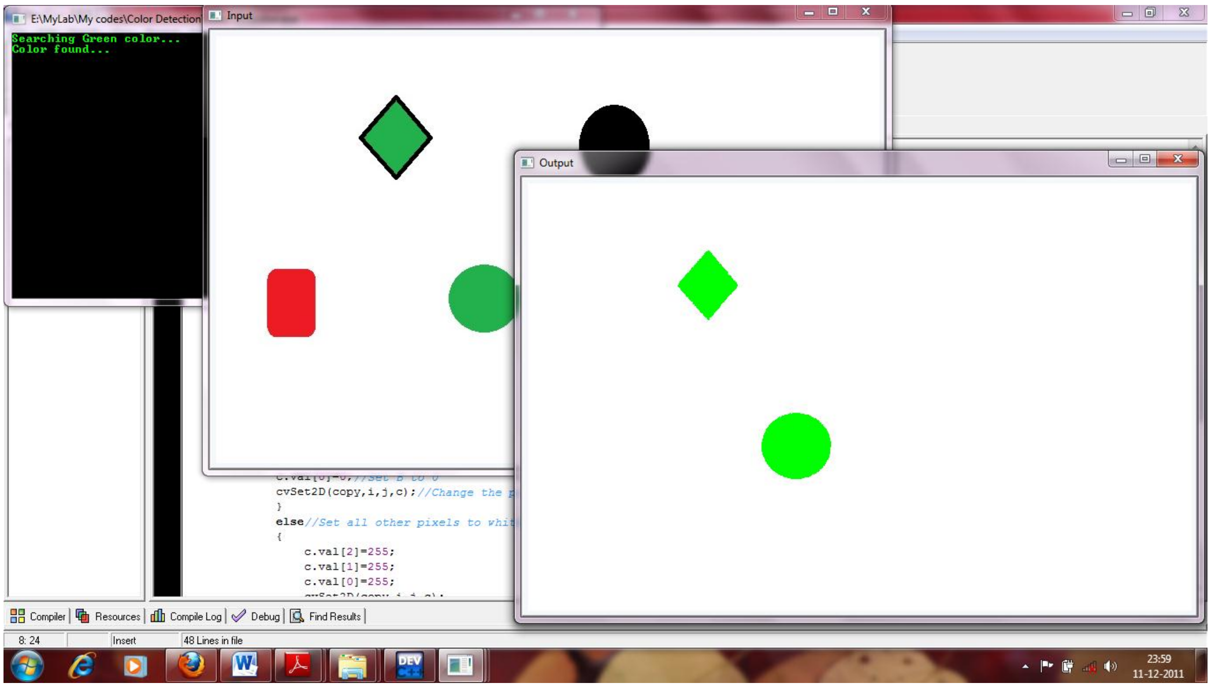
CODE:

```
/*********************************************************************/

Program: Color Detection using BGR values of pixels

Compiler:  DevC++  4.9.9.2 with OpenCV2.0

/*********************************************************************/


#include<highgui.h>

#include<cv.h>

#include<iostream>

using namespace std;

int main()

{

   char name[]="sample.jpg";                // Assign the file name to a character array

   IplImage* src=cvLoadImage(name,1); // Loading the image

   IplImage* copy=cvCreateImage(cvGetSize(src),8,3); //Create a new image of,8 bit,3 channel

   CvScalar s,c;  // create two scalar variables

   cout<<"Searching Green color...\n";

   for(int i=0;i<(src->height);i++)//In the 2D array of the img..count till the vertical pixel reaches the
height of src

   {

          for(int j=0;j<(src->width);j++)//In the 2D array of the img..count till orizontal pixel reaches the
width of src

          {

           s=cvGet2D(src,i,j); //Get the RGB values of src's i,j into a scalar s

           if((s.val[2]<50)&&(s.val[1]>100)&&(s.val[0]<100))// if RGB values (in the order as in code) are
satisfying threshold condn ie. RED<50 & GREEN>100 & BLUE<100

//Remember s.val[2],s.val[1],s.val[0] are RGB correspondingly
```

```
            {                                    //ie. if the pixel is  predominantly Green

            c.val[2]=0;//Set R to 0

            c.val[1]=255;//Set G to 255

            c.val[0]=0;//Set B to 0

            cvSet2D(copy,i,j,c); //Change the pixel value of copy img to pure green(G=255 R=0 B=0)

            }


            else //Set all other pixels in copy to white

            {

               c.val[2]=255; // Red

               c.val[1]=255;// Green

               c.val[0]=255;// Blue

               cvSet2D(copy,i,j,c); // Now set the scalar c(now white) to the pixel in i,j in copy

            }

            }


}


   cout<<"Color found...\n";


cvNamedWindow( "Input", CV_WINDOW_AUTOSIZE ); //Create a window "Input"

cvShowImage( "Input", src ); //Display src in a window named "Input"

cvNamedWindow( "Output", CV_WINDOW_AUTOSIZE ); //Create a window "Output"

cvShowImage( "Output", copy ); //Display copy in a window named "Output"

cvWaitKey(); //Wait till the user presses a key or cvWaitKey(10) will wait for 10ms

cvReleaseImage( &src );

return 0;
```

```
}
```

OUTPUT :



```
Searching Green color...
Color found...
```

```
        c.val[0]=0;//Set B to 0
        cvSet2D(copy,i,j,c);//Change the p
        }
        else//Set all other pixels to whit
        {
            c.val[2]=255;
            c.val[1]=255;
            c.val[0]=255;
            cvSet2D(copy,i,j,c);
```

The original image has green pixels which are changed to PURE GREEN in output. Though this method is little inefficient, it doesn't make big difference in such a simple application.

Now try changing the threshold values to find more colors ☺