Project Title

Colour based object detector with C++ and openCV (Ubuntu 16.04)

Developed by: MSc. Shiva Agrawal

Place: Germany

Date: November 2018

Content

List of Figures

1. Introduction

1.1. Project description

As each image can be represented with RGB values, depending on the values of each color, we can also detect the objects in an image and then it can be further used in computer vision application.

The aim of this project is to develop such application which allow user to select the lower and upper threshold values of each RGB colors and then to detect the object based on those values.

For the development, I have used Eclipse CDT as IDE, OpenCV, C++ programming language and Ubuntu 16.04 (Linux OS).

We need to configure the eclipse in order to work with opencv3 and this task can be confusing, if all the required settings are not properly known. These configurations are available from the official website of opencv.org but are not complete. I had faced problems while working and hence I thought to make a short document on this part. This document is available separately in the repository under computer vision folder for reference.

1.2. Outline

- Chapter 1: Project description and outline of the project report
- Chapter 2: OpenCV and Eclipse CDT
- Chapter 3: Color Object detector app
- Chapter 4: Test and results
- Chapter 4: Conclusion
- Chapter 5: References

2. OpenCV and Eclipse CDT

2.1. OpenCV

[1] **"OpenCV (Open Source Computer Vision Library)** is an open source library for image processing and computer vision. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV is designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform".

2.2. Eclipse CDT

[2] "The CDT Project provides a fully functional C and C++ Integrated Development Environment based on the Eclipse platform. Features include: support for project creation and managed build for various toolchains, standard make build, source navigation, various source knowledge tools, such as type hierarchy, call graph, include browser, macro definition browser, code editor with syntax highlighting, folding and hyperlink navigation, source code refactoring and code generation, visual debugging tools, including memory, registers, and disassembly viewers."

3. Color Object detector app

As we know that an image is represented as 2D matrix containing values of each pixel. If the image is RGB and represented using uint8 then it is saved as three 2D matrix of each color channel where each pixel value is between 0 and 255 representing the intensity of that color for the pixel.

If we filter the pixels depending on the color values for specific range than that color part of the image can be highlighted and rest can be suppressed. This is the basic concept behind this application.

It takes the image as input and convert into color image. Then depending on the user selected (using sliders) threshold values of each color, the image is filtered and then displayed.

The complete application is developed in OOPs with C++. Following files are the source code of the applications. (available in **src** folder of the project repository)

1. ColourObjectDetector.cpp
2. ColourObjectDetector.h
3. ColourObjectDetectorApp.cpp (like main.cpp)

This project report is also available in **doc** folder of the project repository.

4. Test and results

In order to test the application, an image with preselected colors is created using Gnome Paint in Ubuntu as shown below in Fig 1. (ignore the borders)

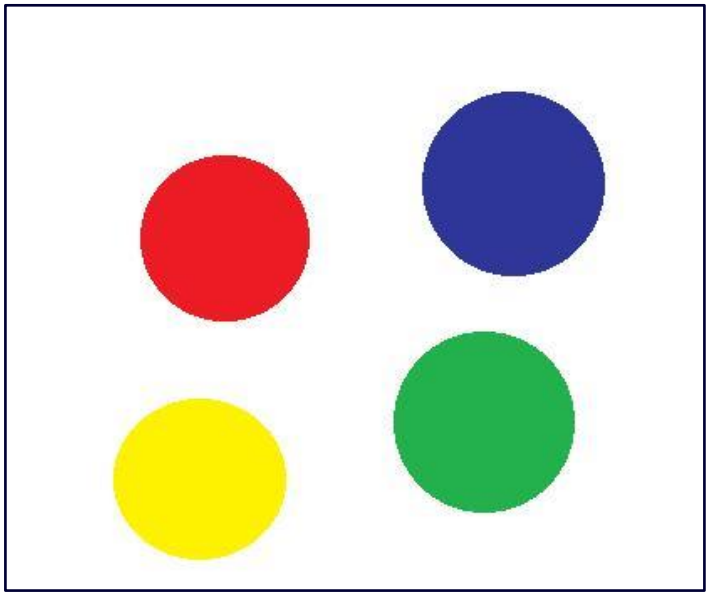

Figure 1 - Original RGB image

Then each color object is tested for functionality by changing the values on the slider.

4.1. Test 1: Red Object detection

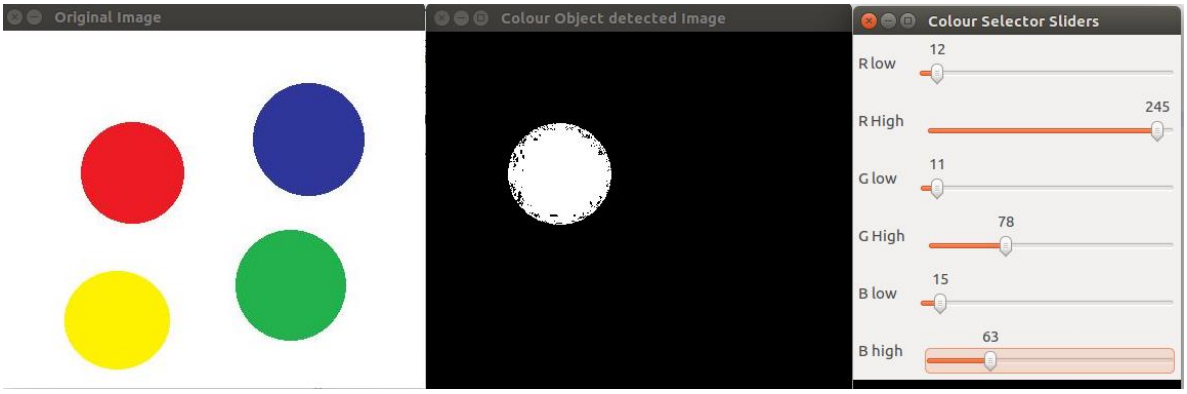

Figure 2 - (right) Original Image, (middle) Red color Object highlighted, (right) sliders

4.2. Test 2: Green Object Detection

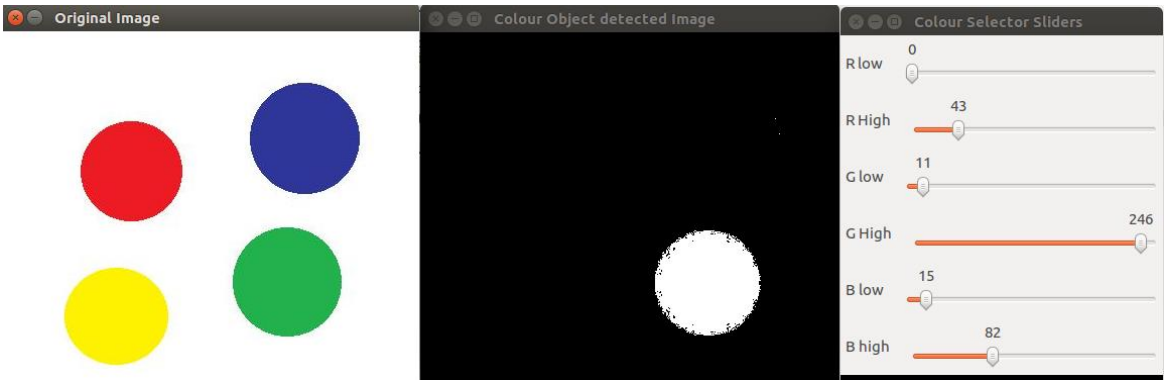

Figure 3 - (right) Original Image, (middle) Green color Object highlighted, (right) sliders

4.3. Test 3: Blue Object Detection

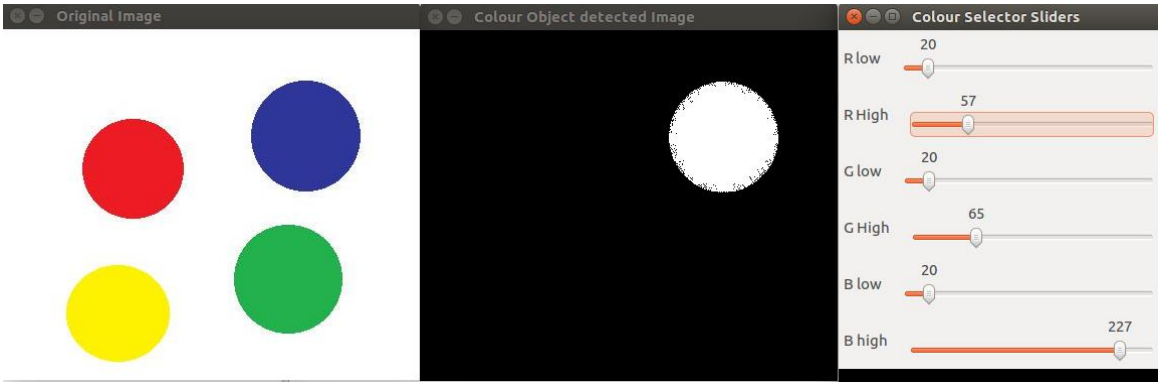

Figure 4 - (right) Original Image, (middle) Blue color Object highlighted, (right) sliders

These Screenshots are also available in **test** folder of the project repository.

5. Conclusion

The color-based object detection application is developed and tested.

6. References

[1] "CV," openCV, [Online]. Available: https://opencv.org/.

[2] "Eclipse CDT," Eclipse Foundation Inc., [Online]. Available: https://www.eclipse.org/cdt/.