**Use this link to convert document to markdown text:** [**https://word-to-markdown.herokuapp.com/**](https://word-to-markdown.herokuapp.com/)

**Markdown cheat sheet:** [**https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet**](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)

**Inserting Images:** [**https://stackoverflow.com/questions/14494747/add-images-to-readme-md-on-github**](https://stackoverflow.com/questions/14494747/add-images-to-readme-md-on-github)

# Blob Detection

The Blob Detection repo is a personal project of mine that deals identifying color similarity within an image that is captured in real-time via a webcam. It uses the Blob Detection library developed by [Julien ‘v3ga’ Gachadoat](http://www.v3ga.net/blog2/). The inspiration comes from a saying my drawing professor used, “draw what you see not what you think you see”. The intention of this project is to draw profiles around similar colors in the canvas emphasizing light, shadow, and color. This repo was developed using Processing version 3.2.3, on Windows PC. Newer versions of Processing might need some minor modifications to the code.

## Intro:

The focus of this project is to identify color similarities within images. An [average human can distinguish about 10 million different colors] (<https://en.wikipedia.org/wiki/Color_vision>). This project attempts to highlight those different colors and finds their seminaries within images taken from a web cam attached to your PC. Check out the [color similarity page on my website](https://www.charlesportelli.com/play-1/#/color-sampling/)to see the results from this app. For a list of all my projects check out: [www.CharlesPortelli.com](https://www.charlesportelli.com/)

## Requirements:

* Processing 3.2.3 or later
* Web cam (either from laptop or USB webcam)
* [Blob detection library](https://processing.org/reference/libraries/) by [Julien ‘v3ga’ Gachadoat](http://www.v3ga.net/blog2/)
* Canvas size must match an aspect ratio that the webcam can support

## Results:

The resultant images can be seen below.