EDUCATIONAL OUTREACH
FLORIDA POLYTECHNIC UNIVERSITY

FLORIDA POLYTECHNIC
COUNTING M&MS
What's Possible with Computer Vision

Overview:

In this lesson we will learn about the python language and OpenCV. You will understand how to create a program that will count the color of M&Ms in a video feed. This will be done by using digital image processing techniques such as color conversion, masking, dilation, and thresholding.

Learning Outcomes

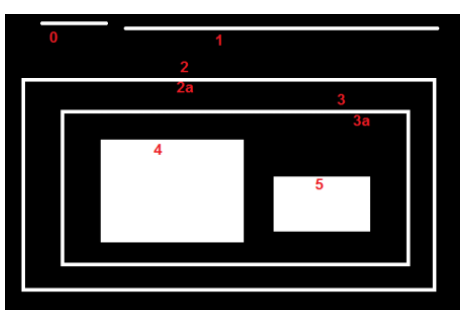
- Identify and recognize the basics concepts of computer vision.
- Understand and demonstrate the basic functions of masking and thresholding.
- Find and test upper and lower bounds for color detection.

Materials
- Browser with Google
- Google Account
- Code and Files from lesson downloads
- Google Colaboratory
- Peko Color Converter Site
- Computer Vision Vocabulary Sheet

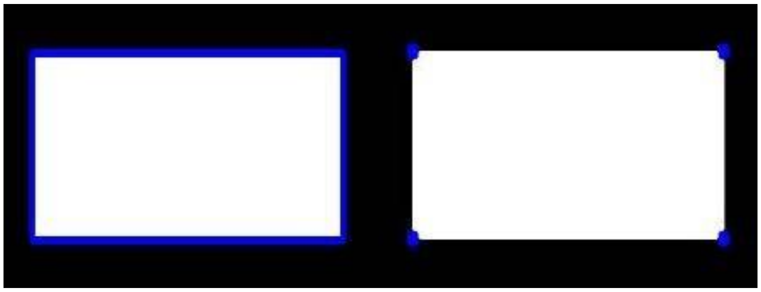
Background
The Contour:

Contours can be explained as a line/curve surrounding points of an image that share similar color and intensity values. This is very helpful when performing obstacle detection and obstacle recognition. In order to use contouring in an efficient way, it is best to use a binary image, as findContour() works better on binary images. The binary image we will be using is our mask, which will be discuss later on in the document.

One important property of contours is hierarchy, this is where a contour is within a contour. The outer contour will be considered the parent contour and the inner contour will be considered the child contour. There are many ways to place the contours in a hierarchy, contours can have the same hierarchy, or they can have different hierarchy. For our example will be using RETR_EXTERNAL, which only returns the parent contours, thus minimizing the contours within a contour.



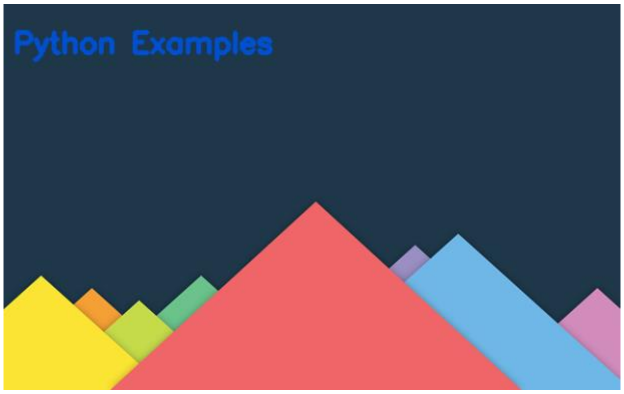
Another property of contours that will help in machines running with less memory is approximation. We can use approximation by only using four points to draw a contour; normally contours are drawn with 734 points. To use no approximation, you would use cv2.CHAIN_APPROX_NONE(figure on the left)and to use four-point approximation you would use cv2.CHAIN_APPROX_SIMPLE(figure on the right).



The putText() Function:

The putText() function is used both in the contourColor() function and counterText() function. The putText() function is used to place text onto an image. The function uses various parameters to place the text of your choice onto the image. The require parameters that were used in our example are: image, text, org, font, font scale, color, and thickness.

- **Image:** this is the image that we are placing the text in
- **Text:** this is the text that we are placing onto the image
- **Org:** this is the bottom left corner of the text box and can be used to place the text anywhere on the image.
- **Font:** this is the font style of the text. FONT_HERSHEY_PLAIN, for example.
- **Font Scale:** this is the font size of the text.
- **Color:** this is the color of the text. For our example we passed (0, 0, 0) for RGB black.
- **Thickness:** this is how thick the text is.



Thresholding and Masking:

Two important techniques that were used for color detection were thresholding and masking. Thresholding is used in conjunction with masking to segment parts of an image, such as color. This is done by applying limits onto the image and applying a light color (white) to areas that fall within the limits and applying a dark color (black) to areas that fall outside the limits. In our example we use HSV color limits to separate pixels that fell within the specific color range, we then masked the image and applied a dark blue to areas outside the HSV range and a yellow color to areas that fell within the color range; this were then passed to the contourColor() function to create a contour.

The Activity

In this activity you will learn the basics of contours and placing text on an image. You will also learn to apply thresholding limits to detect specific colors and your results on a test image that has been specifically calibrated for the threshold values.

Setting Up Google Colab

Download the files associated with this lesson and save them in a place of your choice. Once downloaded go to [Google Colab,](https://colab.research.google.com) and sign in with your google account. Go to "File" on the top left and choose "Open notebook", from there click the "Upload" option, then click "choose file", and choose the file named "Counting_M&Ms_Simple_Fill_In.ipynb". Once open, click the file icon on the left side bar of your colab notebook (Figure A); your screen should now look like Figure B. Once in the files menu choose the upload file icon (Figure C) and upload all the images under the "M&M images" folder in the

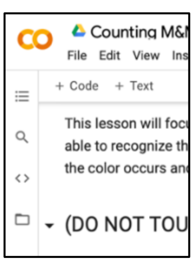


Figure A.

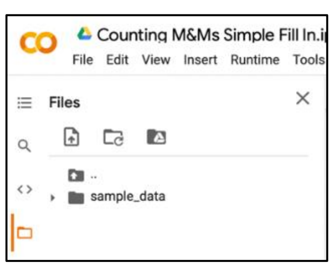


Figure B.

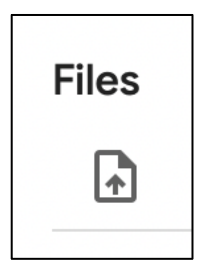


Figure C.

downloaded files. Finally, go to “Runtime” on the top left and click “Run all”(Figure D), you can also. Individually run each line by clicking the play icon when you hover over the “[ ]” icon on each code segement(Figure E). When finished, your result should look like Figure F; where the green contour is not placed correctly, this where our activity begins!!

Figure D.

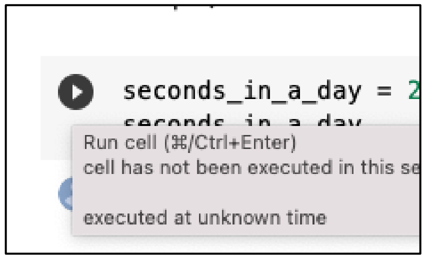


Figure E.

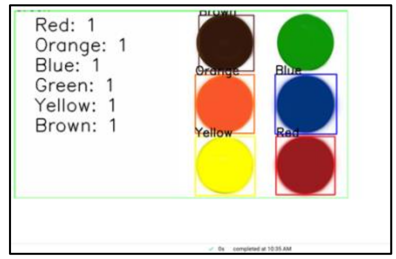


Figure F.

Detecting Green!

Navigate through the code until you find the “Try it yourself” section. There you will find a section of code that defines the lower and upper limits of the HSV scale for green; we will be working with the lower limits – which are set to [0, 0, 0], where the first 0 is the place holder for Hue(H), the second is the place holder for Saturation(S), and the third is the place holder for Value(V). To help us find the correct limits, we will be using [Peko-Step](), an online website that converts RGB to HSV and allows us to manipulate HSV values. HSV is a color scale that uses Hue which represents the color, Saturation which represent the lightness of the color, and Value which measures the darkness of the color. This scale is widely used in color detection because of the number of shades you can achieve with each color.

I suggest to place the Google Colab tab to the left and the Peko-Step tab to the right(Figure G), this way you are able to quickly access both pages. In Peko-Step, before we begin using the HSV slide bars we need to ensure that the range for S and V is set to 0…255. To do this, go on the drop down menu for “range(S,V)” and select 0…255(Figure H). Another thing to note is that the H range is set to 0…360, however python measures hue from 0…180. Because there is no dropdown menu to set H range to 0…180 in Peko-Step, we need to divide the H value by two when we place it onto Google Colab; for example – if your H is 96 in Peko-Step, then you will put 48 as your H value in the code.

Figure G.

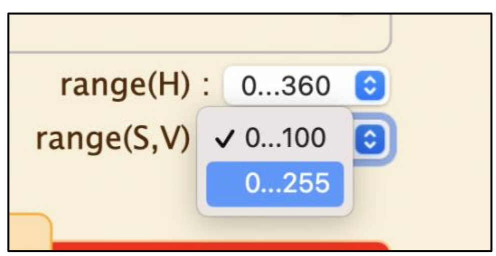


Figure H.

To begin, it is good to have a start point, and thus the best start point is to begin by displaying green. To do this, in Peko-Step, slide the RGB values(Figure I) so that Red(R) is set to 0, Green(G) is set to 255, and Blue(B) is set to 0. Now look at the HSV values, you can now notice that the values are set as followed: 120 for H, 255 for S, and 255 for V. These values are also the same as our upper limits in the code: 60 (120 / 2) for H, 255 for S, and 255 for V. Now that we have starting point, first begin by lowering H until the color represented is no longer a type of green(this is when the color starts looking more yellow), then start lowering S and V value, but still keeping a green shade. Continue to slide the scales, while maintaining a shade of green. Run the code multiple times until the green contour is correctly place around the green M&M, as shown in Figure J. If you are experiencing difficulty, feel free to open the “Counting M&Ms Answer Key.ipynb”, using the same first steps from the Setting Google Colab section.tt

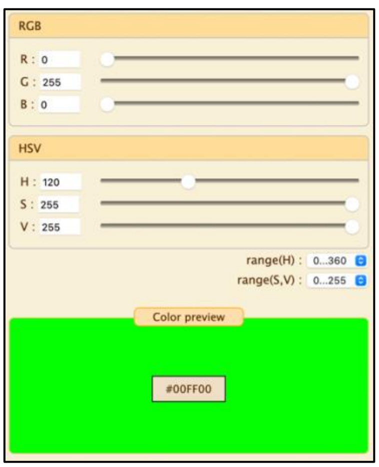


Figure I.

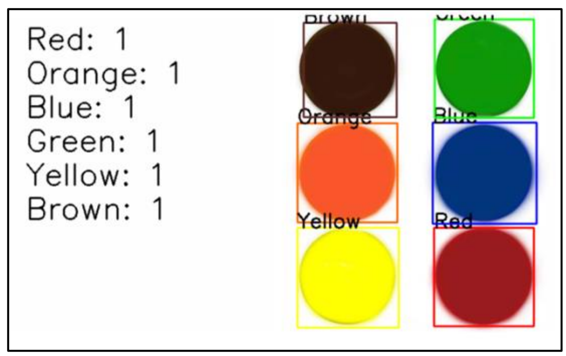


Figure J.

Assessment

1. Why are masking and thresholding so important to computer vision?

2. What does each component of the HSV represent?

3. How is HSV use to detect colors, such as green?

4. Could you use other color scales, like RGB, to detect colors? Are there pros and cons?

5. Were you able to detect green? What parameters did you use?

Additional Resources

1. Counting M&Ms Library Imports (Code)
2. Images (3)
3. Counting M&Ms Vocabulary list

Created and Designed by Juan Forero, Computer Engineering class of 2021

Thank you for downloading this lesson, please take a moment to complete our survey