

Coding up a Color Selection

Let's code up a simple color selection in Python.

No need to download or install anything, you can just follow along in the browser for now.

We'll be working with the same image you saw previously.



Check out the code below. First, I import `pyplot` and `image` from `matplotlib`. I also import `numpy` for operating on the image.

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
```

I then read an image and print out some stats. I'll grab the x and y sizes and make a copy of the image to work with. NOTE: Always make a copy of arrays or other variables in Python. If instead, you say `"a = b"` then all changes you make to `"a"` will be reflected in `"b"` as well!

```
# Read in the image and print out some stats
image = mpimg.imread('test.jpg')
print('This image is: ', type(image), 'with dimensions:', image.shape)

# Grab the x and y size and make a copy of the image
ysize = image.shape[0]
```

```
xsize = image.shape[1]
# Note: always make a copy rather than simply using "="
color_select = np.copy(image)
```

Next I define a color threshold in the variables `red_threshold`, `green_threshold`, and `blue_threshold` and populate `rgb_threshold` with these values. This vector contains the minimum values for red, green, and blue (R,G,B) that I will allow in my selection.

```
# Define our color selection criteria
# Note: if you run this code, you'll find these are not sensible values!!
# But you'll get a chance to play with them soon in a quiz
blue_threshold = 0
green_threshold = 0
red_threshold = 0
rgb_threshold = [blue_threshold, green_threshold, red_threshold]
```

Next, I'll select any pixels below the threshold and set them to zero.

After that, all pixels that meet my color criterion (those above the threshold) will be retained, and those that do not (below the threshold) will be blacked out.

```
# Identify pixels below the threshold
thresholds = (image[:, :, 0] < rgb_threshold[0]) \
             | (image[:, :, 1] < rgb_threshold[1]) \
             | (image[:, :, 2] < rgb_threshold[2])
color_select[thresholds] = [0,0,0]

# Display the image
plt.imshow(color_select)
plt.show()
```

The result, `color_select`, is an image in which pixels that were above the threshold have been retained, and pixels below the threshold have been blacked out.

In the code snippet above, `red_threshold`, `green_threshold` and `blue_threshold` are all set to 0, which implies all pixels will be included in the selection.

In the next quiz, you will modify the values of `red_threshold`, `green_threshold` and `blue_threshold` until you retain as much of the lane lines as possible while dropping everything else. Your output image should look like the one below.

