UDACITY

PROJECT

Traffic Light Classifier

A part of the Intro to Self-Driving Cars Program

PROJECT REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Well done! Your classifier works **very** well! Please see the detailed feedback below.

Notebook Questions

In the project notebook, all questions are answered. (There are two questions total.)

Good choices on features! Using similar characteristics as what humans use is a great first choice. It may be good to play with a colorspace calculator to really find the limits of each color. Maybe you can use GIMP or Photoshop to pick colors in various pixels and observe the HSV values?

Nice analysis of the weak points of your classifier! Would it be possible to use spatial positioning to refine the search for colors? That is, we know that in the vertical traffic lights green will always be at the bottom so there's no sense in using pixels at the top.

As for finding the traffic lights in a video stream, this topic is covered near the end of term 1 of SDCND, using computer vision techniques. However, it is also possible to use deep learning techniques.

Pre-processing

All input images (before they are classified) should be processed so that they are the same size.

You created a function to resize all your images to 32 by 32 pixels, great!

All labels should be a one-hot encoded vector of length 3. Ex. 'yellow' becomes: [0, 1, 0].

Your one_hot_encode function returns a length three vector with the element 1 in the correct position for the label.

Create a brightness feature

Using HSV colorspace, extract a feature from a traffic light image that represents the level(s) of brightness in an image. This feature should help classify any traffic light image. A feature can be a list, array, or a single value.

You used the HSV colorspace to create a histogram of brightness feature, good choice!

I noticed you also checked which color was most present in the image as an additional feature. Good call!

Model Evaluation

In the given test set, red traffic lights can never be mistakenly labeled as green.

No red lights are labeled as green, so it won't cause cars to crash by driving out of turn (although sudden stops could be an issue if the car slows down on a green light).

The model must have greater than 90% accuracy on the given test set.

Wow! 98% accuracy is very good! I guess if humans had 32x32 resolution in their eyes, we would probably get a similar accuracy! It does seem though that for the first 3 misclassified images you may be able to adjust the detection to classify them correctly. The last one is barely visible and was probably included just to see what the students' algorithms do.

Classification Model

Using any created features, write a classification function that takes in a standardized RGB image and outputs whether a traffic light is red, yellow, or green as a one-hot encoded label.

I like how your classification model uses both colors and a brightness histogram to make your choice, and gives importance to a color that is very strong.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

Student FAQ