UDACITY

‹ Return to Classroom

Traffic Light Classifier

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

CONGRATULATIONS. Your project passed all the requirements in the rubric.
Thanks for your time and efforts. Keep working hard and stay motivated. Good luck for your next projects.

If this is your final project, congratulations again! You are about to graduate.
Go to the main page of this nanodegree and press the GRADUATE button.

Notebook Questions

✓

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

The 2 questions are answered in a complete and correct way. Thanks.

(QUESTION 1): How do the features you made help you distinguish between the 3 classes of traffic light images? Answer: I looked at an approach to consider that all traffic lights are vertical and that the brightness will be largest in the area that has a light active. A red light would be always be at the top, yellow light in the middle and a red light would be at the bottom. Using RGB converted to HSV and masking the Value channel of a histogram feature vector. A histogram is a graphical display of data that shows bars of different heights. Each bar groups data (in this case, pixel values) into ranges and the height of each bar indicates the number of times the data falls into that range. So, a taller bar show that more data falls in that specific range. This would offer the best indication when splcing and image of where the light would be. To confirm this I also wanted to use an RGB summation feature. This would only be available for the red and green light, but I felt it was important to get another feature to confirm either green or red as they have the highest impact.

(Question 2): After visualizing these misclassifications, what weaknesses do you think your classification algorithm has? Please note at least two.
Answer: Weakness includes the fact that it assumes that the image is always formatted for a traffic light, i.e. is always zoomed in and the image contains a traffic light. This is an issue where traffic lights are not vertical, i.e. in Quebec, Canada where traffic lights are horizonal and have red lights at both ends, with yellow and green lights in the middle. This classifier would need to improve to reconize this. I would also like to use a sobel operator to determine if an arrow is present in a traffic light, potentially if I can recongize the direction of the arrow also output the command of say "Green light, right turning lane"

output the command or say "Green light, right turning lane".

Pre-processing

✓

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

All input images are pre-processed so that they are the same size.

✓

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

All labels are one-hot encoded vectors of length 3. TEST PASSED.

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 can help classify any traffic light image. A feature can be a list, array, or a single value.

Three brightness features were created based on the location in the image:

1. Upper region = red light.
2. Middle region = yellow light.
3. Lower region = green light.

The most illuminated region wins the classification.

(QUESTION 1): How do the features you made help you distinguish between the 3 classes of traffic light images? Answer: I looked at an approach to consider that all traffic lights are vertical and that the brightness will be largest in the area that has a light active. A red light would be always be at the top, yellow light in the middle and a red light would be at the bottom. Using RGB converted to HSV and masking the Value channel of a histogram feature vector. A histogram is a graphical display of data that shows bars of different heights. Each bar groups data (in this case, pixel values) into ranges and the height of each bar indicates the number of times the data falls into that range. So, a taller bar show that more data falls in that specific range. This would offer the best indication when splcing and image of where the light would be. To confirm this I also wanted to use an RGB summation feature. This would only be available for the red and green light, but I felt it was important to get another feature to confirm either green or red as they have the highest impact.

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.

You successfully created the function `estimate_label` that takes an RGB image and tells the color of a traffic light.

Model Evaluation

✓

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

You have obtained an accuracy of `91.91%`.
And only `24` images were misclassified, out of 297 images. Congratulations.

✓

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

Red lights are never mistakenly labeled as green. TEST PASSED.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

☆☆☆☆☆