

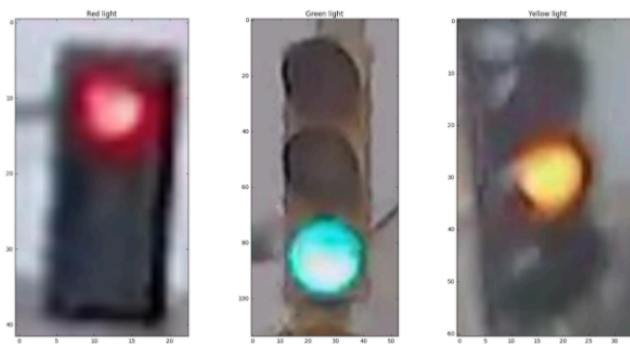
PROJECT: Traffic Light Classifier

Build a classification pipeline that takes in an image of a traffic and outputs a label that classifies the image as a: red, green, or yellow traffic light.

1. Project Overview

Traffic Light Classifier

In this project, you'll use your knowledge of computer vision techniques to build a classifier for images of traffic lights! You'll be given a dataset of traffic light images in which one of three lights is illuminated: red, yellow, or green.



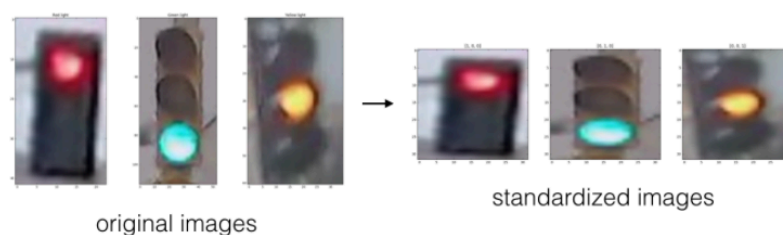
Images from the dataset. Left to right: red, green, and yellow traffic lights.

Classification Steps

In the provided notebook, you'll pre-process these images, extract features that will help distinguish the different types of images, and use those features to classify the traffic light images into three categories: red, yellow, or green. The tasks will be broken down into a few sections:

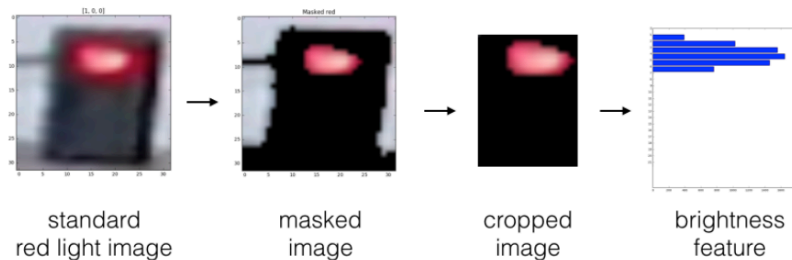
1. Loading and visualizing the data. The first step in any classification task is to be familiar with your data; you'll need to load in the images of traffic lights and visualize them!

2. Pre-processing. The input images and output labels need to be standardized; that is, all the input should be of the same type of data and of the same size, and the output should be a numerical label. This way, you can analyze all the input images using the same procedures, and you know what output to expect when you eventually classify a new image.



Pre-processed, standardized images

3. Feature extraction. Next, you'll extract some features from each image that will be used to distinguish and classify these images. This is where you have a lot of creativity; features should be 1D vectors or even single values that provide some information about an image that can help classify it as a red, yellow, or green traffic light.



An example of feature extraction steps

4. Classification and visualizing error. Finally, you'll write one function that uses your features to classify *any* traffic light image. This function will take in an image and output a label. You'll also be given code to classify a test set of data, compare your predicted label with the true label, and determine the accuracy of your classification model.

5. Evaluate your model. To pass this project, your classifier must be >90% accurate and never classify any red lights as green; it's likely that you'll need to improve the accuracy of your classifier by changing existing features or adding new features. I'd also encourage you to try to get as close to 100% accuracy as possible!

Next, read through the instructions and get ready to build a classifier!

2. Project Instructions

The Notebook

You'll be tasked with completing a project notebook `Traffic_Light_Classifier.ipynb`.

In this notebook, some template code has already been provided for you, but you'll need to implement additional code steps to successfully complete this project. Any code that is required to pass this project is marked with **'(IMPLEMENTATION)'** in the header. There are also a couple of questions about your thoughts as you work through this project, which are marked with **'(QUESTION)'** in the header.

Make sure to answer all questions and to check your work against the [project rubric](#) to make sure you complete the necessary classification steps!

Helper functions and testing

Also included are some additional Python files: `helpers.py` and `test_functions.py`

These provide helper functions (that load in data) and test functions that will let you test your code as you go! You do not need to change these files, but you may add to them if you want. It is especially encouraged to look at the test functions and see how they evaluate your code.

A complete classification model

In the project notebook, you will be tasked with building a classification model step-by-step. Your complete classification code should be able to take in an RGB image and output a predicted label for that image.

Criteria

You should submit your project once you meet the following two criteria:

1. Greater than 90% accuracy
2. *Never* classify red lights as green

This project is fairly open-ended and a good portfolio project that you can show off to friends, enemies, and potential employers. For this reason, you are encouraged to get as close to 100% accuracy as possible!

Evaluation

Once you have completed your project, use the [Project Rubric](#) to review the project. If you have covered all of the points in the rubric, then you are ready to submit! If you see room for improvement in any category in which you do not meet specifications, keep working!

Your project will be evaluated by a Udacity reviewer according to the same Project Rubric. Your project must "meet specifications" in each category in order for your submission to pass.

Submission

Once you have completed the notebook, make sure you've run every cell, and then **submit your project from your workspace**. Pressing the submit button zips all the files in the workspace and submits them for grading!