

# The goals / steps of this project are the following:

- 1- Compute the camera calibration matrix and distortion coefficients given a set of chessboard images.
- 2- Apply a distortion correction to raw images.
- 3- Use color transforms, gradients, etc., to create a thresholded binary image.
- 4- Apply a perspective transform to rectify binary image ("birds-eye view").
- 5- Detect lane pixels and fit to find the lane boundary.
- 6- Determine the curvature of the lane and vehicle position with respect to center.
- 7- Warp the detected lane boundaries back onto the original image.
- 8- Output visual display of the lane boundaries and numerical estimation of lane curvature and vehicle position.

The images for camera calibration are stored in the folder called camera\_cal.

The images in test\_images are for testing your pipeline on single frames.

If you want to extract more test images from the videos, you can simply use an image writing method like cv2.imwrite(), i.e., you can read the video in frame by frame as usual, and for frames you want to save for later you can write to an image file.

To help the reviewer examine your work, please save examples of the output from each stage of your pipeline in the folder called output\_images, and include a description in your writeup for the project of what each image shows.

The video called project\_video.mp4 is the video your pipeline should work well on.

The challenge\_video.mp4 video is an extra (and optional) challenge for you if you want to test your pipeline under somewhat trickier conditions.

The harder\_challenge.mp4 video is another optional challenge and is brutal!

If you're feeling ambitious (again, totally optional though), don't stop there! We encourage you to go out and take video of your own, calibrate your camera and show us how you would implement this project from scratch!