Code Review Yize Liu

I think the whole project is great and amazing. I watch the ouput video and I think the track detection result is accurate. However, there are still some suggestions I want to make to your project.
First of all, it's hard for me to run your project on my own computer, the environment is hard to set up and I don't know which program I need to start with. I think it will be better for you to make clarification in your readme file to show the steps of running your program. What's more, if this is a mobile application (Ios or Android), I hope you can paste the link in APP store so that I could download your app and try to use it by myself. If this is a web application, I think you could upload your project to a cloud platform to get a public server (like Amazaon EC2 or GCE), in this way, I could visit your website.

Next, I notice that there are many functions and files included in your projects, but it's hard to figure out which one is the main function and how all those files are connected and integrated, you may be better to illustrate this in readme file.

Last but no the least, I review each file in your Our_main_code folder. I guess this folder contains all your codes. And here are some suggestions which I think could improve the conciseness and maybe the efficiency of your program. Hope they are useful to improve your projects.

Calibrate_camera.py:
1: From line 12 to 32: since the value from 2 to 20 are all set to (9,6), it'll be better if you write a for loop to assign the value of the dictionary,
eg: for I in range (2,21):
        objp_dict[i] = (9,6)
In this way, the code could be get shorter and cleaner.

Line.py：
1： from line 19 to line 25, I think in python, you could assign values to multiple variables at the same time. Therefore, you don't have to write a few lines to assign the same values to different variables. Instead, it may be better to write like this, self.A = self.B = self.C = []
self.A_avg = self.B_avg = self.V_avg = 0
2.
I think the way you define the Line() class could be improved. I think it would be better to cancle the extra variables self.A_avg, self.B_avg, self.C_avg, which could decrease the space complexity and decrease the time consuming to some extend.
For example, the codes for add_fit(from line 50 to 54) could be adjusted to just one line:
return(np.mean(self.A), np.mean(self.B), np.mean(self.C))

combined_thresh.py:
1:

Code Style problem: There are some extra spaces in this programe, eg(line 89,94, 104,109 …), make sure that the extra spaces and blank lines are as few as possible. In this way, the codes will look cleaner.
2:
In main function, from line 110 to line 121, I think the codes could be simplified. Since it could be found many replicate syntaxes like: plt.imshow(…). To avoid repetition, I think the codes could be optimized in this way:

```
 count i = 1 #declare a count variable, which will change from 1 to 6:
imgset=[abs_bin, mag_bin, dir_bin, hls_bin, img, combined]
 while (i <= 5):
      plt.subplot(2,3,i+1)
      if (i == 4):
         plt.imshow(imgset[i],vmin=0, vmax=1)
      else:
      plt.imshow(imgset[i], cmap = 'gray', vmin=0, vmax=1)
```

In this way, the whole 12 lines codes could be shorten into just 6 lines.

Dilation_try.py:
1:
I notice that the setmin is declares as a global variable which is set to 200. However, in the while loop, it's just used once in line 39(change > setmine), Therefore, it's redundant to declare a variable to represent an Integer, the line 39 could just be written as if(change > 200) and there is no need to declare the variable setmin.
2:
In the while loop, there are two variables lower_red = np.array([0,10,50]), upper_red = np.array([100,25,200]). Since in each iteration, both of them don't change, I think it may be a waste of time and space to declare such two variables in each iteration, you can declare both of them as global variables or just simply use ([0,10,50]) and ([100,25,200]), instead.
3:
There are still some blank lines in this program, it'll be better to notice the code styles.

Gen_example_images.py

1:
The numpy library imported in the 4th line is never used.

2:
I notice that the variables as_bin, mag_bin, dir_bin, hls_bin in line 33 and binary_unwarped and m in line 38 are declared but never used later. Maybe it could be replaced by one variable.

Line_fit.py:

1:
Imported Library problem:

All the libraries listed below are imported in the program but never used:
Line 4: import matplotlib.image as mpimg
Line 5: import pickle
Line 6: from combined_thresh import combined_thresh
Line 7: from perspective_transform import perspective_transform
2:
From line 125 to line 126:
The variables left_fitx and right_fitx are declared and assigned but never used, could consider to delete both of them.
3:
I notice that in this program, the assignment between ret dictionary and variables happens many times, which is written in this way:

```
left_fit = ret['left_fit']
right_fit = ret['right_fit']
nonzerox = ret['nonzerox']
nonzeroy = ret['nonzeroy']
left_lane_inds = ret['left_lane_inds']
right_lane_inds = ret['right_lane_inds']
```

I think it might be better to declare those variables as global in a list or an ENUM. In this way, each time when you want to make assignments, you just need to write a for loop and do not have to make an assignments for each variables.

Line_fit_video.py:
1:
numpy as np, matplotlib.pylot as plt as well as matplotlib.image as mpimg imported but never used.
2:
Line 18-20:
The list_line and right_line are all assigned to the same value. Therefore, the two lines could be merge into one line: left_line = right_line = Line(n-winow_size).
Since the variable window_size is just used once in this function, it could be replaced by 5. So the whole line 18-20 could be adjusted into: left_line = right_line = Line(n-5).
Line 22-23: I think maybe you could assign the values in this way:
Left_curve = right_curve = 0
Left_lane_inds = right_lane_inds = None

3:
Line 32-33:
Since the variables like right_lane_inds, right_curve… are all declared outside the function, which means that they are actually the global variables, I think it's not necessary to declare them as global again inside the functions, those two lines can simply be deleted to shorten the lines of codes.

4:
Still the same suggestions as the line_fit.py above, better to use list or enum to simply the process of assignments between the dictionary and variables.

Perspective_transform.py:

1:

Lack of comments in this file, which makes it hard to understand the main function of this program.

2:

There are still lots of blank lines in this file like above, it will be better to delete them and make the codes look shorter.

In a word, it’s an excellent project, can’t wait to see your final presentation.