گزارش کار پروژه سیگنال فاز یک

فاطمه مداح زاده 810195472

در این پروژه قصد داریم تا خطوط جاده را در عکس ها و ویدیوها پیدا کنیم

عکس اصلی:

تابع select_white_yellow :

برای پیدا کردن خطوط جاده ها ابتدا رنگ های سفید و زرد عکس را جدا می کنیم.(چون خطوط جاده ها در عکس به رنگ های سفید و زرد هستند!) این کار را با استفاده از cv2.inrange و بازه ی rgb که به آن می دهیم تا جدا کند، انجام می دهیم. سپس رنگ های سفید و زرد جدا شده را با هم ترکیب می کنیم. عکس شماره 1

برای پیدا کردن خطوط جاده ابتدا لبه های عکس را پیدا می کنیم تا بتوانیم خط های صاف موجود در عکس را تشخیص دهیم.

برای پیدا کردن لبه های عکس از canny edge detector استفاده می کنیم که از سه مرحله تشکیل شده است :

- **تابع grayscale :** با استفاده از cv2.cvtColor عکس را خاکستری می کنیم. این کار در تابع grayscale انجام شده است که خروجی آن در عکس شماره 2 نشان داده شده است.
- **تابع gaussian_blur :** با استفاده از `cv2.GaussianBlur` لبه های ناهموار(تیز) موجود درعکس را صاف می کنیم. چون لبه های تیز به علت ایجاد کردن نویز باعث می شوند لبه های اشتباه شناسایی شوند. این تابع یک پارامتر `kernel_size` نیز می گیرد که باید مثبت و فرد باشد و هرچه بیشتر باشد عکس محوتر می شود ما در اینجا عدد 15 را به عنوان `kernel_size` می دهیم. اما هرچه `kernel_size` بیشتر باشد، به زمان بیشتری برای `process` نیاز دارد. عکس شماره 3
- **تابع canny :** با استفاده از cv2.Canny لبه های موجود در عکس را پیدا می کنیم که دو ورودی high threshold و low threshold را می گیرد. پیکسل هایی که شیب آن ها ازhigh threshold بیشتر باشد، به عنوان لبه پذیرفته می شوند. پیکسل هایی که شیب آن ها از low threshold کمتر باشد، رد می شوند. پیکسل هایی که شیب آن ها بین high threshold و low threshold باشد، تنها در صورتی پذیرفته می شوند که به یک پیکسل که شیب آن از high threshold بیشتر بوده ، متصل باشند. مقدار high threshold و low threshold را با آزمون و خطا پیدا می کنیم. ابتدا low threshold را صفر می گذاریم و مقدار high threshold را تغییر می دهیم. مشاهده می کنیم که وقتی مقدار آن خیلی زیاد می شود هیچ لبه ای شناسایی نمی شود و اگر مقدار آن کم باشد، میزان زیادی لبه در عکس پیدا می شود. زمانی که مقدار high threshold را پیدا کردیم ، مقدار low threshold را تنظیم می کنیم تا لبه های ضعیفی (نویز) که به عنوان لبه شناسایی شده اند حذف شوند. عکس شماره 4

تابع select_region :

چون می خواهیم خطوط جاده را پیدا کنیم، به عنوان مثال نیازی به بررسی آسمان نداریم. پس یک ناحیه (مثلا یک مثلث) را انتخاب می کنیم تا فقط آن را بررسی کنیم. بدین منظور ابتدا با استفاده از تابع select_region یک چند گوشه را با آرایه vertices می سازیم سپس آن را به تابع filter_region می دهیم که با استفاده از تابع cv2.fillPoly ناحیه ی چندگوشه را فیلتر می کند.
عکس شماره 5

تابع hough_lines :

اکنون ما خطوط جاده را پیدا کرده ایم ، اما نیاز داریم که آن ها را به عنوان خط شناسایی کنیم. (هر خط جاده یک خط سمت چپ و یک خط سمت راست دارد)

بدین منظور از cv2.HoughLinesP استفاده می کنیم که با استفاده از hough transform خطوط را شناسایی می کند. با پارامترهایی که در کد داده شده است حدود 5 الی 15 خط برای هر عکس شناسایی می شود و سپس با تابع draw_lines خطوط را می کشیم. عکس شماره 6

تابع average_slop_intercept :

برای هر خط جاده فقط یک خط شناسایی نشده است و تعداد آن ها بیشتر است. به همین دلیل نیاز داریم تا بین این خط ها میانگین بگیریم. از طرفی بعضی خط ها نیز به طور کامل شناسایی نشده اند و ما نیاز داریم تا خط را کامل کنیم تا تمام خط جاده را در بر بگیرد.

همان طور که گفته شد هر خط جاده از دو خط چپ و راست تشکیل شده است که خط های راست شیب منفی و خط های چپ شیب مثبت دارند. پس خط های با شیب مثبت را جداگانه جمع آوری کرده و بین آن ها میانگین می گیریم. همین کار را برای خط های با شیب منفی نیز انجام می دهیم. (در عکس داده شده محور y برعکس است به همین دلیل خطوط سمت چپ شیب منفی و خطوط راست شیب مثبت دارند)

تابع average_slop_intercept :

به ازای هر خط چپ و راست یک شیب و عرض از مبدا برمی گرداند.

تابع make_line_points :

این تابع شیب و عرض از مبدا را به نقاط پیکسل تبدیل می کند و مطمئن می شود که همه حاصل ها int است چون cv2.line به int نیاز دارد.

تابع lane_lines :

خروجی های مرحله ی make_line_points را به خط های چپ و راست تبدیل می کند.

تابع draw_lane_lines :

آرگومان اول آن عکس و آرگومان ورودی دوم این تابع یک آرایه از خط هاست که هر خط از چهار نقطه ی x1,y1 , x2,y2 تشکیل شده است. که همان خط هایی است که خروجی مرحله ی lane_lines هستند. عکس شماره 7

برای ویدیو ها نیز یک کلاس می سازیم که در متد آن از همین توابعی که برای پردازش عکس استفاده کردیم، استفاده می کنیم.

اگر می خواهیم خروجی ویدیو ها به این صورت باشد از دور خطوط جاده عکس قرمز بکشد، از این تیکه کد ها استفاده می کنیم :

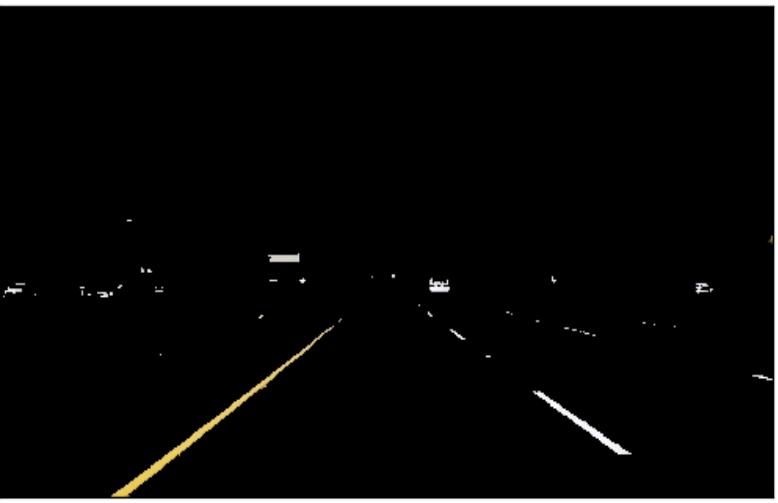
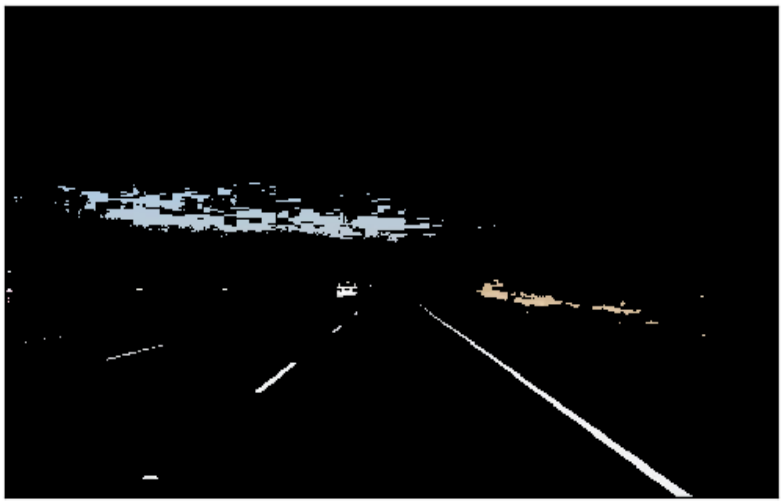
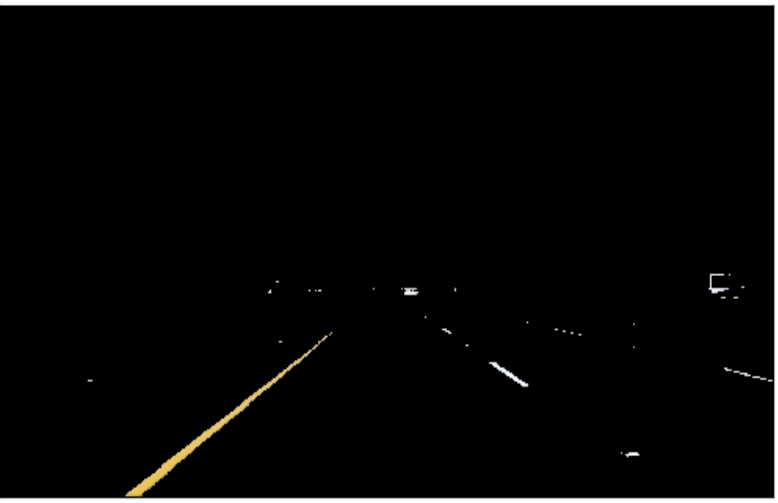
```
line_images = []
for image0, lines0 in zip(images, lines_selected):
    line_images.append(draw_lines(image0, lines0))
```

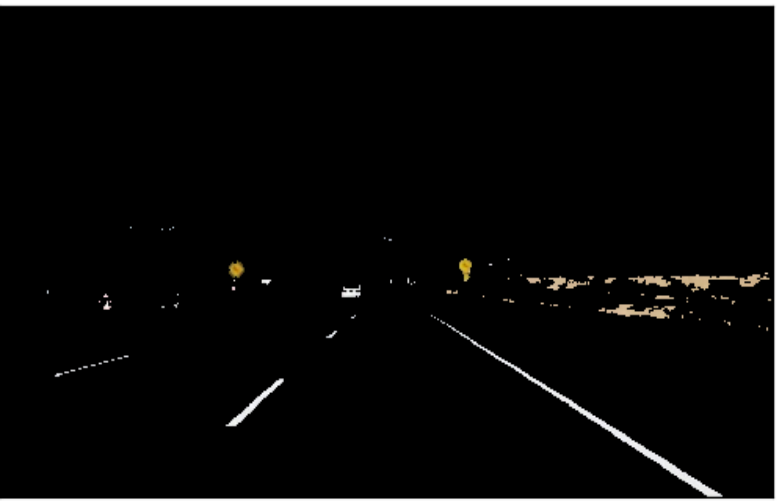
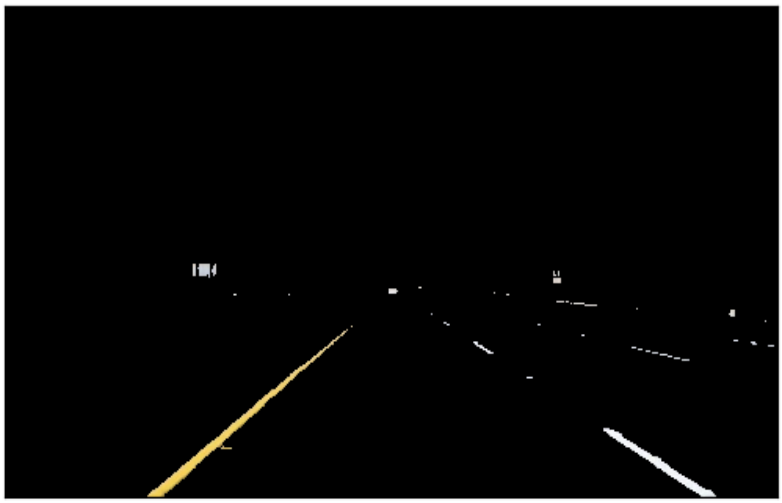
و برای اینکه خطوط جاده ها را هایلایت کند، از تکه کد زیر استفاده می کنیم :

```
lane_images = []
for image1, lines1 in zip(images, lines_selected):
    lane_images.append(draw_lane_lines(image1, lane_lines(image1, lines1)))
```

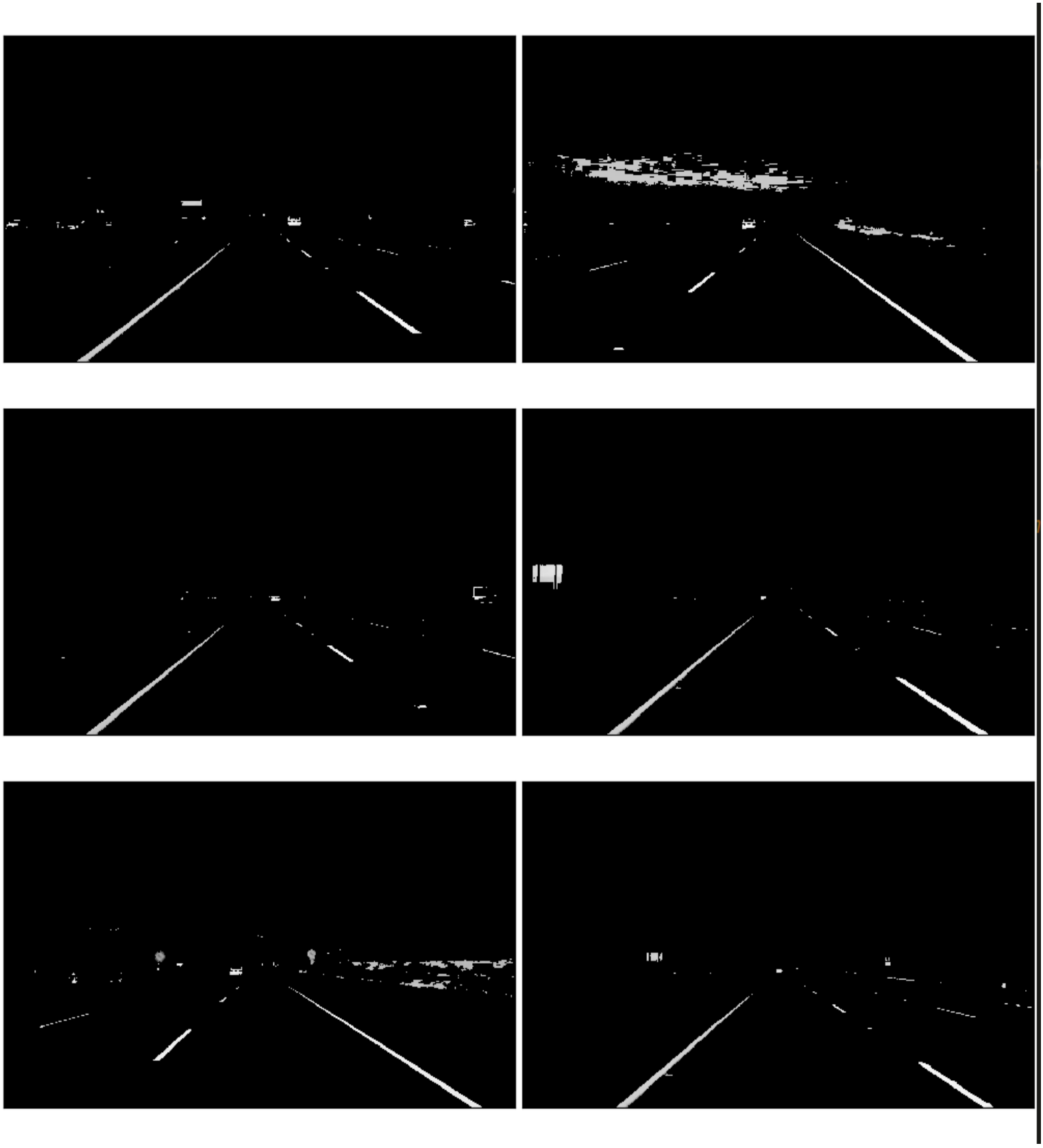
خروجی ویدیو ها به هر دو فرمت گفته شده، در فولدر output_videos ایجاد می شوند.

عکس شماره 1 :

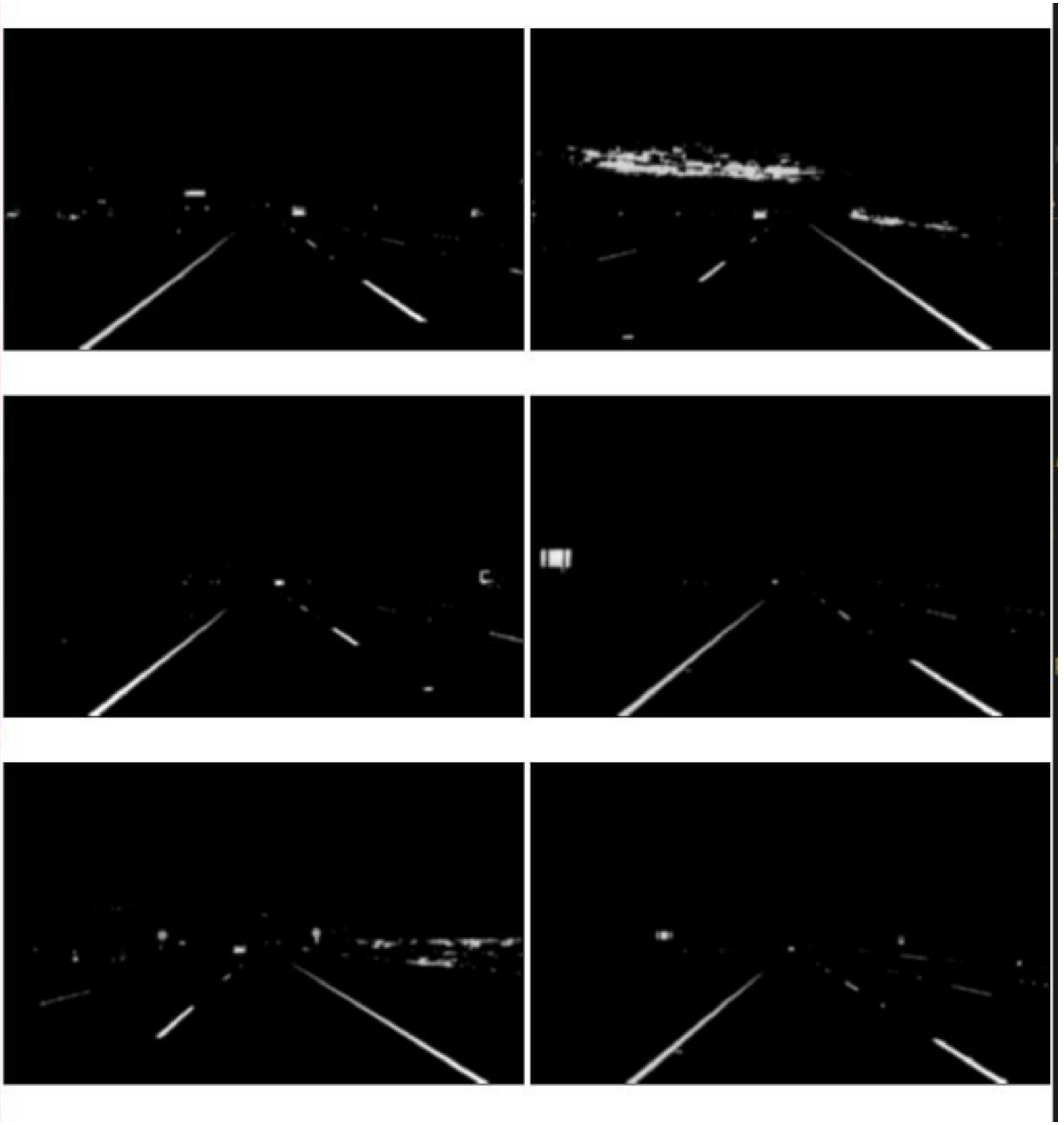




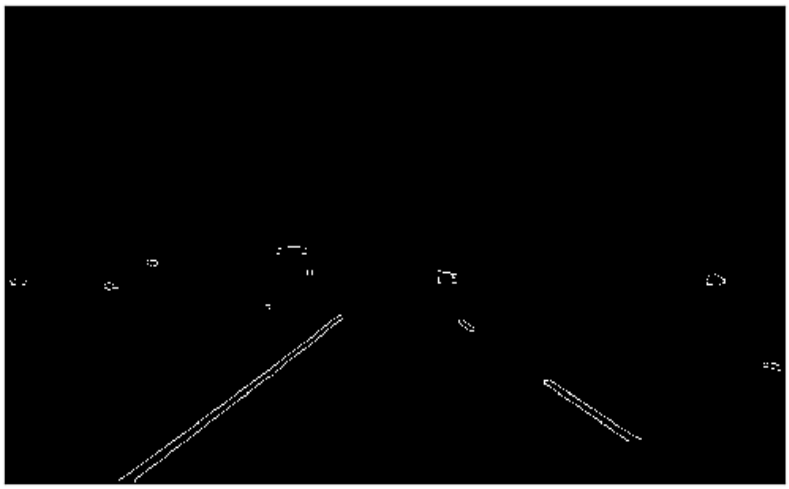
عکس شماره 2 :

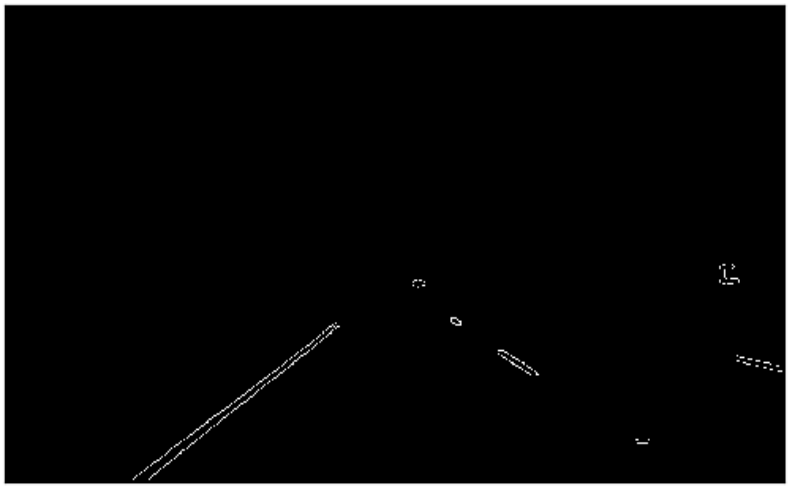
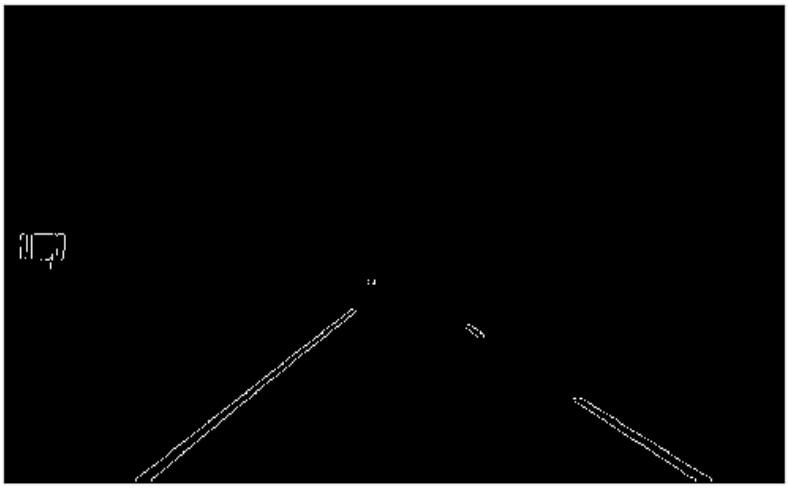
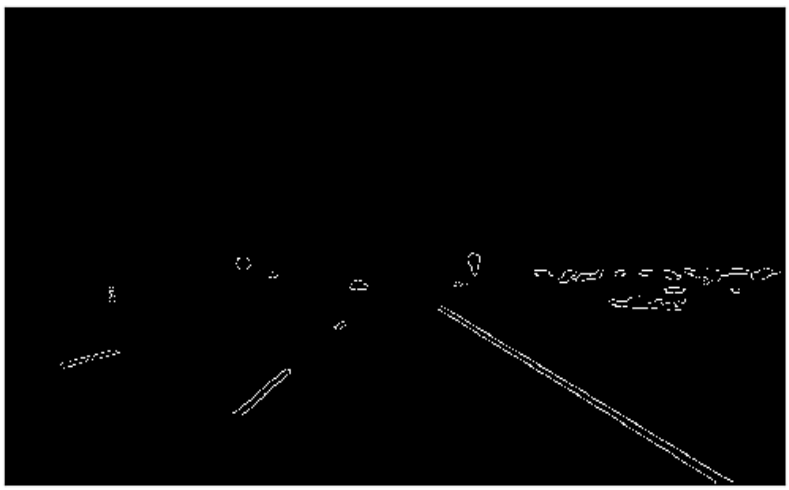
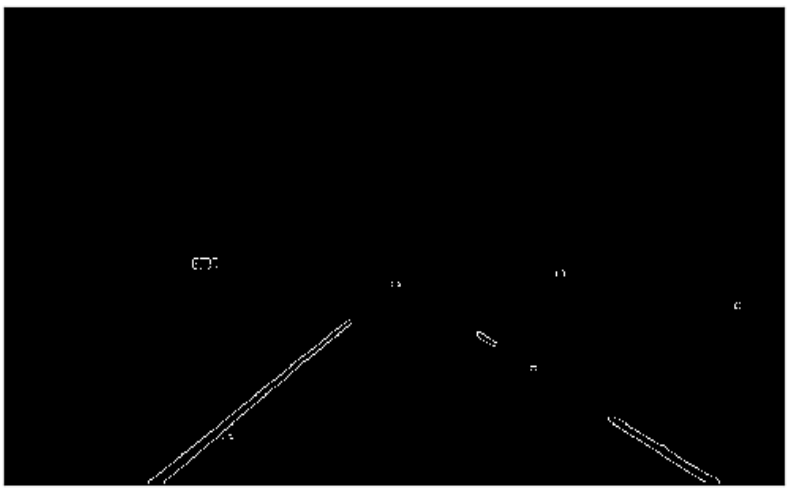


عکس شماره 3 :

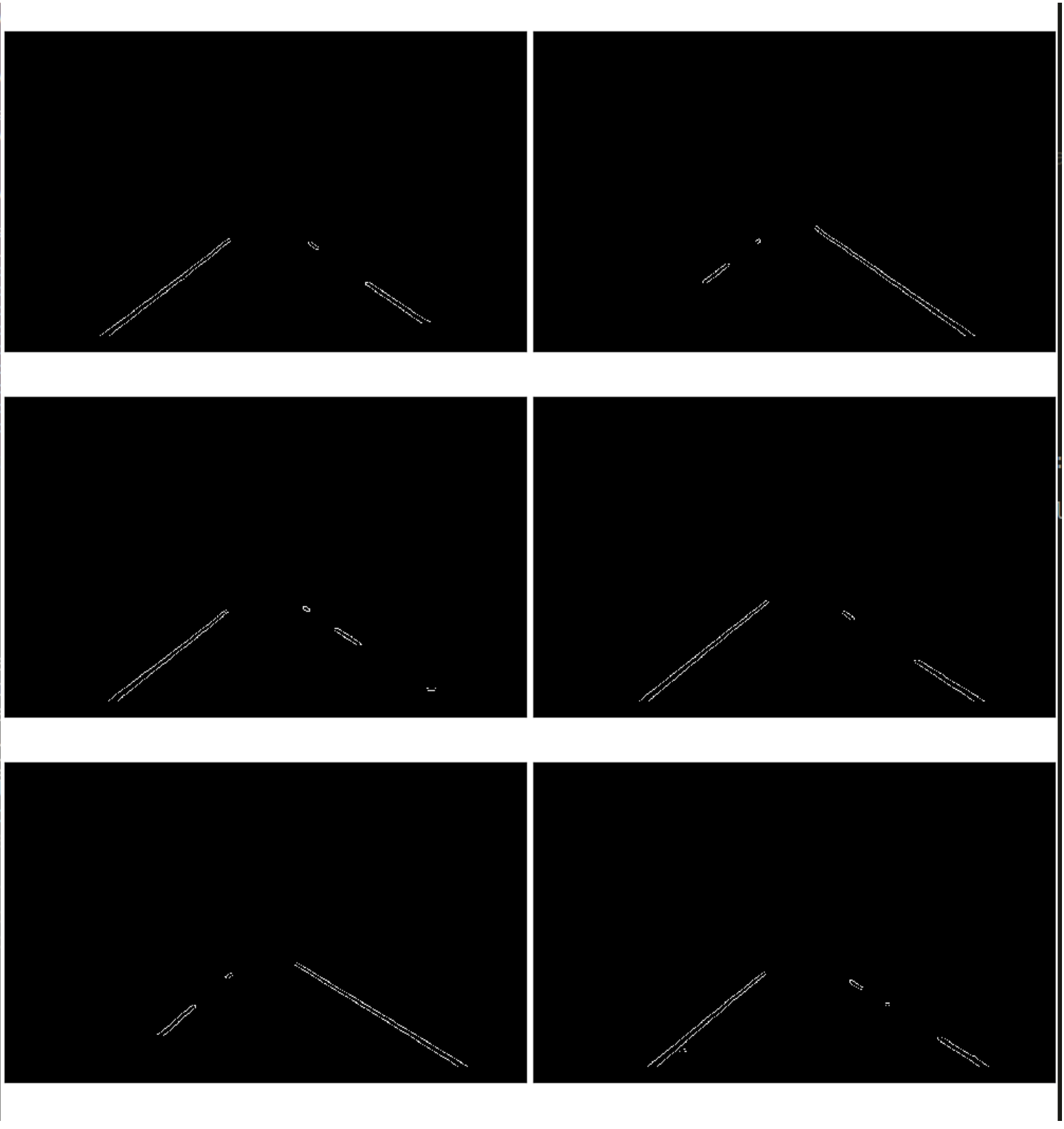


عکس شماره 4 :



عکس شماره 5 :



عکس شماره 6 :

عکس شماره 7 :