



به نام خدا
دانشکده مهندسی برق و کامپیوتر

فاز اول پروژه

تجزیه و تحلیل سیستم‌ها

تاریخ تحویل: ۹۷/۹/۱۵



دانشگاه تهران

پیدا کردن خطوط سفید خیابان به کمک پردازش تصویر

لینک زیر را مطالعه کنید:

<https://github.com/udacity/CarND-LaneLines-P1>

در این لینک همه اطلاعات لازم برای نوشتن این پروژه قرار داده شده است. با این حال خواندن ادامه صورت پروژه بد نیست! در این پروژه همان‌طور که قبلاً گفته شد قرار است خطوط سفید خیابان را به کمک python و OpenCV تشخیص بدهید.

۱- نصب محیط اجرایی:

برای انجام این پروژه صرفاً به پایتون و OpenCV نیاز دارید. فرض می‌کنیم که پایتون را نداریم که فرض غلطی است! (اگر از ابونتو یا هر سیستم‌عامل دیگری از شاخه ابونتو استفاده می‌کنید، پایتون را دارید!). اگر هم ندارید، سرچ کنید که چگونه نصب کنید! برای نصب OpenCV هم می‌توانیم سرچ کنیم، یا دستور زیر را بزنیم:

```
pip install opencv-python
```

۱-۱- Jupyter:

استفاده از جویپتر اجباری نیست ولی استفاده از آن برای ما بسیار خوش‌حال‌کننده است + نمره امتیازی هم می‌توانیم بابت این کار بگیریم. Jupyter رو گوگل کنید و بقیه کارهایش واضح است.

<http://jupyter.org>

۲- ایده‌ها و الگوریتم‌ها

در شروع کدتان حتما از همین لینکی که اول صورت پروژه گذاشته شده استفاده کنید. لازم است یک سری تابع را در آن بنویسید و البته یک سری توابع را هم خودش دارد که می‌توانید آن‌ها را مطالعه و از آن‌ها استفاده کنید. مطالب پیشنهادی که می‌توانید مطالعه کنید به صورت زیر است:

۲-۱- سیاه سفید کردن عکس

سیاه سفید کردن معمولا باعث کاهش نویز تصویر می‌شود؛ تا بتوانید فانکشن‌های بهتری روی آن اعمال کنید. می‌توانید بحث زیر را بخوانید تا بفهمید چرا باید این کار را بکنید. ولی اگر می‌خواهید در این موضوع عمیق‌تر شوید، می‌توانیم موقع تحویل بیشتر صحبت کنیم.

https://www.reddit.com/r/MachineLearning/comments/5h2l09/d_cnn_object_recognition_grayscale_vs_rgb/

۲-۲- ناحیه مورد علاقه

با توجه به جای دوربین لازم نیست همه محیط را به دنبال خطوط بگردیم برای همین فقط ناحیه مورد علاقه خودمان را جست‌وجو خواهیم کرد. (در این تمرین می‌توانید فقط یک مثلث در نظر بگیرید.)

https://en.wikipedia.org/wiki/Region_of_interest

۲-۳- تشخیص لبه‌های عکس

برای این قسمت، از الگوریتم Canny استفاده می‌کنیم. اگر بتوانید ایده دیگری بزنید خیلی خوب می‌شود. به BFS و به صورت کلی به الگوریتم‌های جستجوی گراف فکر کنید.

https://en.wikipedia.org/wiki/Canny_edge_detector

۲-۴- تشخیص نقاط کلیدی خطوط

بعد از تشخیص لبه‌های عکس باید نقاطی که خطوط زیادی از آن‌ها می‌گذرد را پیدا کنیم. این قسمت جذاب‌ترین قسمت این پروژه است. در واقع داریم Heat map نقاط کلیدی را پیدا می‌کنیم که مطمئن هستیم لبه‌هایی که در آن قرار می‌گیرند می‌توانند کاندیدهایی باشند برای پیدا کردن نقاط کلیدی خطوط. برای این منظور از Hough Transform استفاده می‌کنیم. در لینک زیر می‌توانید اطلاعات جامع‌تری در این مورد ببینید:

https://en.wikipedia.org/wiki/Hough_transform

دقت داشته باشید که اینجا هم لازم نیست لزوماً این کار را بکنید. همچنان ایده‌های الگوریتمی ساده‌تری مثل DFS می‌توانند به شما کمک کنند، این نقاط حساس را پیدا کنید.

۲-۵- رگرسیون

بعد از پیدا کردن نقاط کلیدی می‌توانید با یک رگرسیون ساده خطوط سفید جاده را پیدا کنید.

۳- بهتر کردن خروجی‌ها

۳-۱- از بین بردن نویز تصویر

در پردازش تصویر کوچکترین واحد کارایی پیکسل است. چون یک پیکسل خیلی کوچک است، اطلاعاتش خیلی خوب منتقل نمی‌شود. برای همین کارهای زیادی انجام می‌شود که اثر سه رنگ را از بین ببرند (RGB). مثلاً اطلاعات زیر را در نظر بگیرید:

$R = 120, G = 50, B = 10$

$R = 124, G = 44, B = 15$

این دو نقطه می‌توانند دو نقطه کنار هم در یک تصویر باشند. این تفاوت اعداد می‌تواند باعث شود که الگوریتم‌هایمان روی یک رنج خاص بایاس بشوند و یا اصلاً درست کار نکنند. برای همین، تا جای ممکن سعی می‌کنیم این‌ها را به هم نزدیک کنیم (بررسی ناحیه‌ای). که مثلاً ساده‌ترین روش، همان سیاه سفید کردن تصویر است.

یک راه دیگر، استفاده از الگوریتم‌های از بین بردن فاصله بین نقاط به کمک تبدیل‌های مختلف است. یکی از معروف‌ترین این تبدیل‌ها، تبدیل گاوسی است که در لینک زیر می‌توانید اطلاعات بیشتری در مورد آن کسب کنید:

https://en.wikipedia.org/wiki/Gaussian_blur

۴- نکات نهایی

در این پروژه کد خیلی زیادی را نباید بزنی (مگر این که ایده خودتان را بخواهید پیاده‌سازی کنید!). همه توابع بالا در OpenCV وجود دارد و در کد سمپل پروژه هم هست. تنها کاری که باید بکنید ساخت یه pipeline است که همه کارهایی که گفته شده را، روی تصویر انجام دهد. اگر بتوانید محیط خود پروژه را بالا بیارین (راحت‌تر این که با anaconda این کار را بکنید) خیلی کارتان راحت‌تر می‌شود اگه حال هم نداشتید که هیچی (فقط کافیه OpenCV را نصب کنید).

اگر سوالی داشتید به آدرس زیر ایمیل بزنین:

alimohammad1995@gmail.com

اگر سوالاتتان زیاد بود یک روز دیگر هم می‌توانیم بیاییم دانشگاه که سوالاتتان را رفع کنیم. یا اگر حوصله داشتید بیایید شرکت!