

توجه: در انجام پروژه استفاده از اینترنت و مشورت مجاز میباشد اما کپی کردن مجاز نمی باشد و در صورت مشاهده نمره کل پروژه صفر در نظر گرفته خواهد شد. برای پروژه نیاز است که گزارش کامل همراه کدها تحویل داده شود و گزارش نیز نمره قابل توجهی دارد. در کدها هم به میزان نیاز حتما کامنت گذاری انجام شود. لطفا در نوشتن گزارش دقت شود با توجه به شرایط فعلی به احتمال زیاد نمره ی تحویل در گزارش اعمال می شود.

در صورتی که سوالی در مورد پروژه داشتید، ترجیحا آن را در قسمت "پرسش و پاسخ" پروژه مطرح نمایید تا سوال دانشجویان دیگر نیز برطرف شود. در صورت نیاز با ایمیل های [aref.einizade@yahoo.com](mailto:aref.einizade@yahoo.com) یا [parsa.rahimi.n@gmail.com](mailto:parsa.rahimi.n@gmail.com) نیز می توانید مکاتبه نمایید.

اگر از IPython استفاده می کنید می توانید گزارش خود را به صورت Markdown در notebook بیاورید. در غیر این صورت گزارش خود را به صورت PDF همراه با کدهای خود ارسال نمایید. با توجه به احتساب زمان در پروژه بهتر است از notebook استفاده نکنید.

اگر غیر از package های متداول در ویژن مانند (OpenCV, Numpy, torchvision, tf, tqdm, ...) از package دیگری استفاده می کنید در فایل requirements.txt در directory پروژه خود نام package مربوطه را ذکر کنید (اسم هر package در یک خط با توجه به PyPI).

لطفا حتما از اجرای برنامه ارسالی خود اطمینان حاصل کنید.

#### خلاصه سازی ویدیو (Video Synopsis)

امروزه تعداد بسیار زیادی دوربین در اماکنی همچون مراکز حمل و نقل، ATM، نظامی، تجاری، خانگی و آموزشی مورد استفاده قرار گرفته اند. تصاویر این دوربین ها به صورت 24 ساعته در حال ضبط است و باعث استفاده از حجم عظیمی از حافظه شده است در حالی که تنها بخش بسیار اندکی از آنها توسط انسان مشاهده می شود. هنگامی که شخصی بخواهد در ویدیویی جستجو کند (مانند تعقیب یک شخص خاص با دوربین های نظارتی) بایستی مدت بسیار زیادی صرف کند تا شخص مورد نظر را بیابد و اگر اهمال ورزد ممکن است شخص مورد نظر را از دست بدهد. بدیهی است که این امر صرفا با استفاده از نیروی انسانی غیر قابل انجام است.

یکی از روش های موجود خلاصه سازی ویدیو به صورت مکانی و زمانی هست که در نهایت باعث Dense تر شدن اشیا در تصویر و کوتاه شدن زمان کلی ویدیو می شود. در این روش ابتدا اشیا متحرک را از پس زمینه جدا می کنیم. سپس این اشیا را در طول وجود خود در ویدیو Track می کنیم تا Tube های زمانی- مکانی از آنها بسازیم. در نهایت این Tube ها را در ویدیو خلاصه شده ای که می خواهیم آن را تولید کنیم طوری قرار می دهیم که زمان و همپوشانی میان آنها کمینه شود (یک Trade-off بین زمان و همپوشانی داریم. یعنی اگر مثلا همپوشانی را در نظر نگیریم طول ویدیو نهایی برابر با طول بلندترین Tube می شود). با این حال، در این پروژه در سطح پایه آن، نیازی به کمینه کردن همپوشانی بین Tube ها وجود ندارد و ویدیو خلاصه شده شما می تواند طولی برابر بلندترین Tube داشته باشد.

برای مثال می توانید با مراجعه به [لینک ۱](#)، [لینک ۲](#) و [لینک ۳](#) نمونه هایی از این کار را مشاهده کنید.

حال با جستجویی که خود انجام می دهید روش مناسبی برای تولید ویدیو خلاصه شده ارائه دهید. توجه کنید که متد ذکر شده یکی از روش ها برای رسیدن به مقصود این پروژه است و متد های دیگری نیز وجود دارند؛ مثلا به جای بخش بندی مسأله می توان از یک شبکه End-to-End استفاده کرد اما این روش بخاطر پیچیدگی زیاد توصیه نمی شود.

در ویدیو نهایی مانند شکل زیر زمان اصلی حضور شی را در تمام طول مسیر نمایش دهید. به طور مثال زمان اصلی (در ویدیو خلاصه نشده) که ماشین سفید در مکان کنونی خود قرار دارد 38.0 است ولی زمانی که تصویر زیر گرفته شده (در ویدیو خلاصه شده) 5.0 ثانیه است.

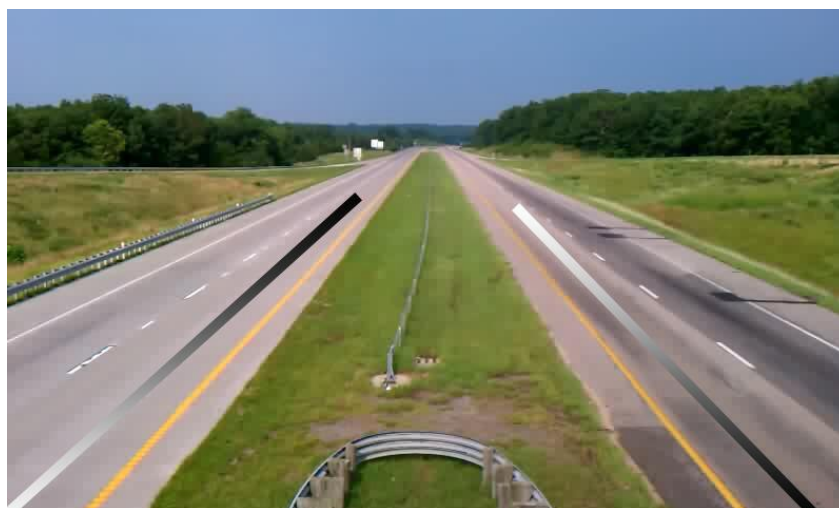


شکل 1. زمانهای اصلی

### موارد امتیازی:

(\*) تشخیص مسیر حرکت و ترسیم آن:

مشابه آنچه در شکل ۱ نشان داده شده، مسیر حرکت را با Gradient Line نمایش دهید. Gradient Line خطی است با روشنایی متغیر. روشنایی خط را برای هر شی طوری تنظیم کنید که رنگ سیاه مشخص کننده اولین مکانی باشد که شی در ویدیو دیده شده و رنگ سفید مشخص کننده آخرین مکان دیده شدن آن باشد. (از آستانه گذاری روی سائز شی می توانید به عنوان معیاری برای دیده شدن یا نشدن استفاده کنید). برای مکانهای مابین هم روشنایی رابه صورت خطی تغییر دهید. را در نظر بگیرید. (راهنمایی: اول سعی کنید که Centroid های اشیاء را به صورت زمانی Cluster کنید تا هر مسیر مشخص شود. می توانید پارامترهای خود را روی ویدیو ها داده شده Fine-Tune کنید. توجه کنید که OpenCV نمی تواند Gradient Line رسم کند. لذا لازم است خودتان آن را پیاده سازی نمایید. همچنین نیازی نیست که مانند شکل ۱، خط Anti Alias شده باشد. هر چند که این کار هم نمره امتیازی محسوب می شود. (راهنمایی: Rasterization را جستجو کنید).



شکل 2. مسیرهای حرکت به صورت Gradient Line

(\*) تشخیص رنگ خودروها:

در این مورد می‌توانید به روش‌های مختلفی نتیجه را نمایش دهید که به چند مورد اشاره می‌شود: ترسیم مربعی کوچک به رنگ تشخیص داده شده برای هر خودرو در هر فریم (مثلاً در گوشه‌ی BBox خودرو)، انداختن یک مستطیل دور هر خودرو در هر فریم که رنگ این مستطیل همرنگ خودروی مورد نظر باشد و ... .

(\*) شمارش تعداد خودروهایی که در طول ویدیو از مسیرهای یکسانی (شمال به جنوب و ...) عبور می‌کنند.

(\*) محاسبه و نمایش کمیتی مربوط به هر خودرو که به نوعی مقایسه درستی از سرعت‌های خودروهای مختلف باشد. مثلاً اگر در ویدیوی اصلی، سرعت واقعی خودرویی 2 برابر خودرویی دیگر باشد، کمیت و عدد محاسبه شده توسط شما تقریباً این نسبت را بین این دو خودرو رعایت کند (دقت شود لزوماً نیازی نیست سرعت واقعی خودروها محاسبه شود چون این محاسبه در اکثر مواقع نیاز به روش‌های پیشرفته و داده‌های بیشتری دارد).

(\*) کمینه کردن همپوشانی بین اشیاء مختلف و در عین حال Dense بودن ویدیو (لازمه Dense بودن ویدیو کاهش زمان بین Tube هاست).

نمره دهی پروژه بر اساس Trial-and-Failure های انجام شده، کیفیت انجام کار و از همه مهم تر **زمان اجرای برنامه** می‌باشد. برای ارزیابی Trial-and-Failure های انجام شده توسط شما، لازم است که مثلاً الگوریتم‌هایی که استفاده کرده اید اما نتایج مناسبی از آنها نگرفته اید، در گزارش بیان و توضیح مختصری راجع به دلیل عدم موفقیت آنها ارائه نمایید.

در این پروژه فرض کنید که پس زمینه در طول ویدیو ثابت است. به همین دلیل استفاده از متد های Deep توصیه نمی‌شود؛ چرا که علاوه بر افزودن پیچیدگی به مسأله، از لحاظ زمان اجرای برنامه نیز مناسب نیستند. به طور مثال اگر از YOLOv3 استفاده شود، نمره‌ی Performance تعلق نمی‌گیرد چرا که روی GPU 1080 Ti با نرخ حدود 4 فریم بر ثانیه خروجی می‌دهد (برای تصاویر 720p).

خلاصه‌سازی را بر روی دو ویدیوی داده شده (Video1.avi و Video2.avi) انجام داده و کدها، گزارش، ویدئوهای خلاصه شده و موارد امتیازی پیاده شده بر روی ویدیوها را آپلود نمایید.