مسئله Vehicle Speed

محمد امانلو شماره دانشجویی: 810100084

مبینا مهرآذر شماره دانشجویی: 810100216

محمد امین یوسفی شماره دانشجویی: 810100236

بهار 1404 درس مبانی بینایی کامپیوتر استاد سیفی پور

این پروژه با هدف ساخت یک سیستم بینایی ماشین برای تشخیص، رهگیری و تخمین سرعت خودروها در فیلم‌های ویدیویی ترافیکی طراحی و پیاده‌سازی شده است. فرآیند انجام این پروژه شامل چندین مرحله مهم است که در ادامه به صورت تفصیلی شرح داده می‌شود.

۱. آماده‌سازی محیط و داده‌ها

در گام اول، محیط برنامه‌نویسی آماده شده و ابزارهای مورد نیاز نظیر کتابخانه‌های OpenCV و dlib نصب و تنظیم شدند. سپس مدل پیش‌فرض Haar Cascade برای تشخیص خودرو (که فایل آن به صورت XML وجود داشت) و فایل ویدئویی نمونه (فیلم حرکت خودروها در بزرگراه) در پوشه‌های مشخص قرار گرفتند. به منظور اطمینان از قابلیت اجرا در سیستم‌عامل‌های مختلف، مسیر فایل‌ها با استفاده از تابع `os.path.join` به صورت داینامیک ساخته شد تا مشکلی از لحاظ ناسازگاری مسیرها ایجاد نشود.

۲. بارگذاری مدل تشخیص و آماده‌سازی ویدئو

در این مرحله مدل Haar Cascade با استفاده از تابع `cv2.CascadeClassifier` بارگذاری شد که امکان تشخیص اشیاء (در اینجا خودروها) را فراهم می‌کند. همچنین ویدئو با `cv2.VideoCapture` باز شده و ثابت شدن ابعاد فریم‌ها به ۱۲۸۰ در ۷۲۰ پیکسل انجام شد تا پردازش و تحلیل تصویر به صورت یکنواخت و استاندارد انجام شود.

۳. پیاده‌سازی تابع تخمین سرعت

تابع `estimateSpeed` طراحی و پیاده‌سازی شد تا سرعت خودروها بر اساس تغییر موقعیت آنها در دو فریم متوالی محاسبه شود. این تابع با استفاده از فرمول فاصله اقلیدسی، فاصله پیکسلی بین دو موقعیت را محاسبه و سپس با تبدیل این فاصله به متر (با فرض یک ثابت پیکسل به متر) و در نظر گرفتن نرخ فریم و تبدیل واحدها، سرعت خودرو را بر حسب کیلومتر بر ساعت تخمین می‌زند. این

بخش بسیار حیاتی است چرا که قابلیت ارائه داده‌های کاربردی برای کنترل ترافیک و مدیریت سرعت را فراهم می‌کند.

۴. طراحی و پیاده‌سازی تابع اصلی رهگیری چندگانه

تابع `trackMultipleObjects` به عنوان هسته اصلی پروژه پیاده‌سازی شد. این تابع فرآیند کامل خواندن فریم‌ها، تشخیص خودروها، ساخت و مدیریت رهگیرهای چندگانه، به‌روزرسانی موقعیت‌ها، حذف رهگیرهای با کیفیت پایین و در نهایت نمایش نتایج را انجام می‌دهد.

- برای هر خودرو یک شناسه یکتا در نظر گرفته شد تا رهگیری‌ها به صورت مستقل و سازمان‌یافته مدیریت شوند.
- سیستم به گونه‌ای طراحی شد که هر ۱۰ فریم یک بار مدل Haar Cascade خودروها را تشخیص داده و خودروهای جدید به لیست رهگیری افزوده شوند.
- رهگیرهای dlib برای خودروهای تشخیص داده شده ساخته شدند و کیفیت آنها در هر فریم ارزیابی می‌شد تا رهگیرهای نامناسب حذف شوند.
- موقعیت خودروها در دو فریم متوالی ذخیره شده و با استفاده از تابع تخمین سرعت، سرعت خودرو محاسبه می‌گردید.
- در نهایت فریم‌ها با کادرهای کشیده شده روی خودروها نمایش داده می‌شدند و امکان ذخیره ویدئو خروجی نیز فراهم شده بود.

5. خلاصه روند پیاده‌سازی

در مورد نحوه پیاده‌سازی باید گفت که برنامه به طور هوشمندانه‌ای طراحی شده است تا هم سرعت اجرای مناسبی داشته باشد و هم دقت قابل قبولی ارائه دهد. مسیر فایل مدل Haar Cascade و مسیر ویدئو به صورت داینامیک با استفاده از `os.path.join` ساخته شده است تا قابلیت اجرا روی سیستم‌عامل‌های مختلف بدون نیاز به تغییر مسیرها فراهم شود. مدل تشخیص با `cv2.CascadeClassifier` بارگذاری شده و ویدئو به کمک `cv2.VideoCapture` خوانده می‌شود. ابعاد فریم‌ها نیز به صورت ثابت و استاندارد (1280x720) تعریف شده است تا پردازش در اندازه یکسان و پایدار انجام شود.

تابع تخمین سرعت با نام `estimateSpeed` بر اساس فاصله پیکسلی بین موقعیت خودرو در دو فریم متوالی کار می‌کند. ابتدا فاصله اقلیدسی بین دو موقعیت محاسبه شده و سپس با فرض یک

مقدار ثابت برای تبدیل پیکسل به متر (ppm=8.8)، به مسافت واقعی تبدیل می‌شود. با در نظر گرفتن نرخ فریم و تبدیل واحدها، سرعت خودرو بر حسب کیلومتر بر ساعت محاسبه می‌گردد. این روش اگرچه ساده است اما با کالیبراسیون مناسب می‌تواند سرعت خودروها را با دقت قابل قبولی تخمین بزند.

تابع اصلی سیستم به نام `trackMultipleObjects` وظیفه پردازش ویدئو و انجام تشخیص، رهگیری و تخمین سرعت را بر عهده دارد. در این تابع ابتدا متغیرهای ضروری مثل شمارنده فریم، دیکشنری‌های ذخیره رهگیرها و موقعیت‌ها تعریف شده و یک شیء `VideoWriter` برای ذخیره خروجی با کادرهای ترسیم شده ایجاد می‌شود. در حلقه پردازش ویدئو در هر فریم، ابتدا تصویر خوانده شده و اندازه آن ثابت می‌شود. سپس کیفیت رهگیری هر خودرو بررسی و در صورت پایین بودن، آن خودرو حذف می‌گردد. هر ۱۰ فریم، تشخیص خودرو توسط مدل Haar Cascade انجام شده و خودروهای جدید به لیست رهگیری افزوده می‌شوند. رهگیری‌ها به روز شده و کادرهای سبز روی خودروها رسم می‌شوند. نرخ فریم سیستم نیز برای پایش عملکرد محاسبه می‌شود. نهایتاً سرعت خودروها بر اساس تغییر موقعیت آنها در دو فریم متوالی تخمین زده شده و می‌تواند روی تصویر نمایش داده شود. نمایش فریم نهایی در پنجره‌ای به نام "result" انجام شده و با فشردن کلید Esc حلقه متوقف می‌شود.

استفاده از ترکیب تشخیص و رهگیری باعث شده سیستم هم از نظر سرعت و هم دقت عملکرد مناسبی داشته باشد. تشخیص مکرر در هر فریم پردازش را کند می‌کند اما اجرای آن هر ۱۰ فریم و استفاده از رهگیری برای فریم‌های میان‌فریم تعادل خوبی ایجاد کرده است. اختصاص شناسه یکتا به هر خودرو این امکان را می‌دهد که داده‌ها و رهگیری هر خودرو به صورت مستقل مدیریت شود و از خطاهای تداخل جلوگیری شود. ساختار دیکشنری‌ها نیز به شکل بهینه‌ای داده‌ها را ذخیره و مدیریت می‌کند و حذف رهگیری‌های ضعیف باعث جلوگیری از بروز خطاهای انباشته و افزایش دقت کلی شده است.

6. بررسی خروجی ها

در ابتدا مسیرها به شکل سازگار با سیستم‌عامل‌های مختلف تعریف و مدل Haar Cascade بارگذاری شد. سپس ویدئو باز و آماده پردازش شد. یک حلقه اصلی نوشته شد که در هر بار اجرای آن یک فریم از ویدئو خوانده می‌شود و پردازش‌های مربوط به تشخیص، رهگیری و تخمین سرعت روی آن انجام می‌گیرد. تشخیص خودروها هر ۱۰ فریم یک بار انجام می‌شد و بقیه فریم‌ها تنها توسط رهگیرها تحلیل

می‌شدند تا سرعت پردازش بهینه باشد. رهگیری‌ها با شناسه‌های یکتا مدیریت شده و خودروهایی که کیفیت رهگیری‌شان پایین می‌آمد از لیست حذف می‌شدند. سرعت خودروها بر اساس تغییر موقعیت آنها در فریم‌ها تخمین زده می‌شد و در صورت نیاز روی تصویر نمایش داده می‌شد. خروجی نهایی با کادرهای سبز رنگ دور خودروها به کاربر نمایش داده می‌شد و امکان ذخیره ویدئو خروجی نیز وجود داشت.

خروجی موجود در پوشه videos نمایانگر عملکرد یک سیستم است که هدف آن تشخیص و رهگیری خودروها در یک فیلم ویدیویی از یک بزرگراه چند خطه است. در این سیستم برای شناسایی خودروها ابتدا از مدل Haar Cascade استفاده شده که یک روش سنتی اما کارآمد برای تشخیص اشیاء است و سپس برای رهگیری خودروها در فریم‌های بعدی از الگوریتم dlib بهره گرفته شده است. در نهایت سیستم امکان تخمین سرعت خودروها را نیز فراهم می‌کند که البته در خروجی ویدئویی که مشاهده می‌کنید، نمایش سرعت روی خودروها به صورت عددی ارائه نشده است اما این قابلیت به صورت کامل در کد برنامه وجود دارد و می‌توان آن را فعال کرد.

در ابتدا باید اشاره کرد که وجود کادرهای سبز رنگ که دور هر خودرو در هر فریم کشیده شده‌اند یکی از شاخصه‌های مهم و قابل توجه این سیستم است. این کادرها نه تنها موقعیت خودروها را به صورت دقیق نمایش می‌دهند، بلکه به خوبی با حرکت واقعی خودروها همگام شده و به طور پیوسته و بدون هیچ گونه پرش یا ناپایداری جابجا می‌شوند. این موضوع نشان می‌دهد که سیستم توانسته به خوبی خودروها را در هر فریم شناسایی کند و همچنین به درستی در فریم‌های بعدی نیز آنها را رهگیری نماید. دقت کادرها اهمیت زیادی دارد زیرا اگر خودرو به درستی شناسایی نشود یا اگر رهگیری در فریم‌های بعدی قطع شود، عملکرد کلی سیستم مختل می‌شود.

از دیگر نکات مهمی که در خروجی قابل مشاهده است، قابلیت سیستم در رهگیری چند خودرو به طور همزمان است. یعنی سیستم توانسته در هر لحظه تعداد زیادی خودرو را با شناسه‌های متفاوت شناسایی و رهگیری کند، به گونه‌ای که هر خودرو با شناسه یکتای خود رصد می‌شود و وضعیت آن به صورت مستقل حفظ می‌گردد. این امر به واسطه مدیریت هوشمند داده‌ها و استفاده از دیکشنری‌ای است که برای هر خودرو یک رهگیر جداگانه تعریف کرده است. این ساختار اجازه می‌دهد که رهگیری‌های مختلف به طور همزمان و بدون تداخل با یکدیگر به خوبی انجام شوند. همچنین مدیریت هوشمند شناسه‌ها باعث می‌شود که سیستم بتواند خودروهایی را که از تصویر خارج شده‌اند یا دیگر قابل رهگیری نیستند، حذف کند و خودروهای جدیدی که وارد میدان دید می‌شوند را به لیست رهگیری‌ها اضافه نماید.

یکی دیگر از ویژگی‌های بسیار مهم که در خروجی مشاهده می‌شود، ثبات و پایداری کادرهای کشیده شده روی خودروها است. هیچ گونه پرش ناگهانی، گم شدن خودرو یا جابه‌جایی ناهماهنگ در کادرها مشاهده نمی‌شود که این مسئله به معنای کیفیت بالای الگوریتم رهگیری است. این الگوریتم به گونه‌ای طراحی شده که خودروها را حتی با تغییر اندازه، تغییر نور و حرکت سریع نیز به خوبی دنبال کند و اطلاعات رهگیری را به صورت پیوسته به‌روزرسانی نماید. این موضوع بسیار حائز اهمیت است چرا که خودروها در یک فیلم واقعی در ابعاد مختلف، با سرعت‌های متغیر و در شرایط نوری متفاوت در حال حرکت هستند و سیستم باید بتواند به صورت پایدار این تغییرات را مدیریت کند.

نکته قابل توجه دیگر، انطباق هماهنگ بین تشخیص‌های اولیه و رهگیری‌های پیوسته است. در سیستم به این صورت عمل می‌شود که هر ۱۰ فریم یک بار، مدل Haar Cascade دوباره خودروها را در تصویر شناسایی می‌کند. این تشخیص‌های جدید باعث می‌شود خودروهای تازه وارد به تصویر شناسایی و به لیست رهگیری اضافه شوند و همچنین خودروهایی که دیگر در تصویر نیستند یا کیفیت رهگیری آنها به شدت پایین آمده حذف شوند. این مکانیزم ترکیبی تشخیص و رهگیری باعث می‌شود سیستم همواره به‌روز و دقیق باقی بماند و از انباشته شدن خطاهای ناشی از رهگیری طولانی مدت جلوگیری شود.

شرایط محیطی و ویژگی‌های صحنه نیز تاثیر بسیار مهمی روی عملکرد سیستم دارند. نور روز و روشنایی مناسب، خطوط راهنمای ترافیکی واضح و پس‌زمینه نسبتاً ثابت از جمله عواملی هستند که به افزایش دقت تشخیص و رهگیری کمک شایانی کرده‌اند. علاوه بر این، زاویه دید بالا که از طریق دوربین ثابت نصب شده بر روی یک سازه مرتفع فراهم شده است، باعث شده خودروها به شکل واضح و با حداقل انسداد در تصویر ظاهر شوند و سیستم بتواند همه آنها را به خوبی پوشش دهد.