



پروژه درس
سیگنال‌ها و سیستم‌ها
دکتر مجاهدیان

دانشگاه صنعتی شریف، دانشکده مهندسی برق
بهار ۱۴۰۴

فهرست مطالب

۴	۱ مقدمه
۴	۲ الگوریتم‌های تشخیص
۶	۳ تعقیب
۶	۱-۳ الگوریتم CSRT
۷	۲-۳ الگوریتم MOSSE
۸	۳-۳ الگوریتم KCF
۹	۴ الگوریتم شما
۹	۵ خروجی مطلوب

نکات پروژه

۱. نکاتی در مورد تحویل پروژه

(آ) سوالات خود را می توانید از دستیاران آموزش درس بپرسید، آیدی تلگرام دستیاران در زیر آورده شده است:

• [REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

(ب) می توانید از متلب یا پایتون جهت پیاده سازی الگوریتم ها خود استفاده کنید، ولی توصیه اکید ما استفاده از پایتون است!

(ج) تمام فایل های متلب و پایتون و همچنین خروجی های عکس یا ویدیو را در یک فایل زیپ قرار دهید و آپلود کنید. همچنین گزارش نهایی خود را نیز در این فایل قرار دهید.

(د) نام فایل نهایی باید به شکل `Project-#StudentNo1_#StudentNo2_#StudentNo3.zip` باشد.

۲. نوشتن گزارش پروژه با LATEX نمره امتیازی دارد. هم خود فایل PDF گزارش و هم کد LATEX را باید آپلود کنید.

۳. استفاده از `GitHub` برای پروژه اجباری است. فایل های خود را در یک `Repository` قرار دهید و مصحح های پروژه را اضافه کنید. آیدی گیت مصحح ها در زیر آورده شده است:

[REDACTED]
[REDACTED]
[REDACTED]

۱ مقدمه

یکی از مسائل محوری در تحلیل سیگنال‌های ویدیویی، شناسایی و دنبال کردن اشیای متحرک در طول زمان است. از دیدگاه مهندسی برق و به‌ویژه در چارچوب درس سیگنال‌ها و سیستم‌ها، یک ویدیو را می‌توان به‌عنوان یک سیگنال گسسته‌ی دوبعدی زمانی مدل کرد؛ به‌عبارت دیگر، یک توالی از فریم‌ها که هر کدام حاوی اطلاعات مکانی از صحنه هستند و به‌صورت گسسته در زمان نمونه‌برداری شده‌اند. در این بستر، تعقیب^۱ را می‌توان مسئله‌ی تخمین وضعیت یک شیء در طول زمان بر مبنای اطلاعات جزئی، احتمالی و نویزی هر فریم در نظر گرفت، که ماهیتاً یک سیستم تخمین و فیلتر کردن سیگنال زمان پیوسته است.

در این پروژه، هدف طراحی و پیاده‌سازی یک سامانه‌ی ساده‌ی تشخیص و تعقیب^۲ است که با یک‌بار شناسایی شیء هدف در ویدیو، موقعیت آن را در فریم‌های بعدی دنبال کند. ساختار سامانه شامل دو بخش اصلی خواهد بود:

۱. **تشخیص اولیه:** در آغاز ویدیو، با استفاده از یک مدل یادگیری‌محور، موقعیت مکانی شیء هدف (برای مثال، یک انسان یا خودرو) در یکی از فریم‌های اولیه شناسایی می‌شود. این مدل می‌تواند از پیش آموزش‌دیده^۳ باشد و هدف آن تنها تعیین موقعیت اولیه شیء است.

۲. **تعقیب شیء^۴:** پس از تشخیص اولیه، فرآیند تعقیب آغاز می‌شود. هدف از این مرحله، دنبال کردن پیوسته‌ی موقعیت شیء در فریم‌های بعدی ویدیو است، به‌نحوی که بتوان مسیر حرکت آن را با دقت مناسبی بازسازی کرد. در این پروژه، با الهام از روش‌های موجود، یک تعقیب‌کننده^۵ طراحی و پیاده‌سازی خواهید کرد که با استفاده از اطلاعات فریم‌های گذشته و کنونی، موقعیت شیء را در طول زمان تخمین می‌زند.

نکته‌ی کلیدی در این پروژه، بهره‌گیری از وابستگی‌های زمانی میان فریم‌هاست، که به‌عنوان یکی از اصول بنیادی در تحلیل سیگنال‌های پویا شناخته می‌شود. تعقیب، برخلاف تشخیص مستقل در هر فریم، با تکیه بر پیوستگی زمانی و پویایی سیستم، تخمین بهتری از وضعیت شیء ارائه می‌دهد و از منظر مصرف منابع محاسباتی نیز بسیار بهینه‌تر است.

۲ الگوریتم‌های تشخیص

تشخیص شیء^۶ یکی از مسائل اساسی در بینایی ماشین و پردازش تصویر است که هدف آن، شناسایی محل و نوع اشیای موجود در یک تصویر یا فریم ویدیویی می‌باشد. برخلاف دسته‌بندی تصویر که تنها وجود یا عدم وجود یک کلاس خاص را مشخص می‌کند، تشخیص شیء ترکیبی از دو وظیفه‌ی اساسی است: **مکان‌یابی^۷** و **تشخیص کلاس^۸** به‌صورت همزمان. در این مسئله، خروجی الگوریتم معمولاً شامل مختصات ناحیه مستطیلی محاط‌کننده^۹ به‌همراه برجسب کلاس پیش‌بینی شده است.

Tracking^۱

Detection-Tracking^۲

Pre-trained^۳

Object Tracking^۴

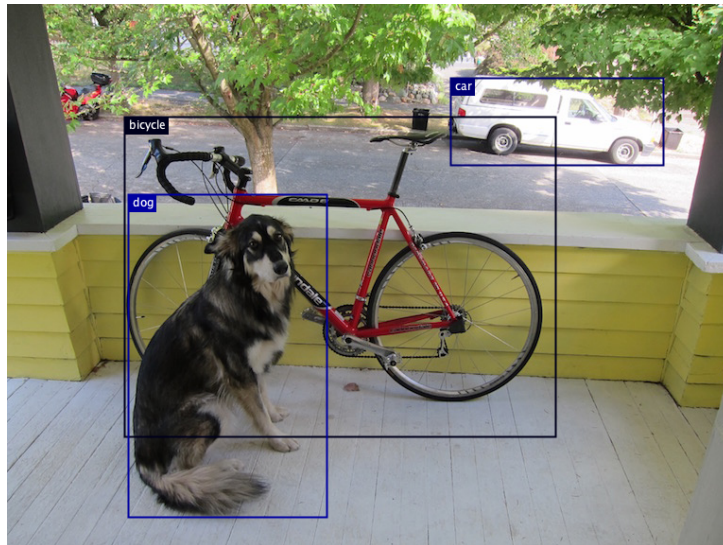
Tracker^۵

Object Detection^۶

Localization^۷

Classification^۸

Bounding Box^۹



شکل ۱: نمونه‌ای از خروجی الگوریتم تشخیص شیء

در سال‌های اخیر، مدل‌های یادگیری عمیق، به‌ویژه شبکه‌های عصبی کانولوشنی^{۱۰}، پیشرفت‌های قابل‌توجهی در دقت و سرعت تشخیص شیء به‌وجود آورده‌اند. این مدل‌ها معمولاً از شبکه‌های از پیش آموزش‌دیده بهره می‌برند که روی پایگاه‌داده‌های بزرگی مانند COCO یا ImageNet آموزش دیده‌اند و می‌توانند برای وظایف خاص به‌صورت جزئی بازآموزی^{۱۱} شوند.

در این پروژه، برای انجام تشخیص اولیه، می‌توان از یکی از مدل‌های رایج و آماده‌ی تشخیص شیء استفاده کرد. در ادامه، به‌صورت خلاصه چند مدل پرکاربرد معرفی می‌شوند:

- **YOLO**: یک خانواده از مدل‌های سریع تشخیص شیء است که کل تصویر را در یکباره از طریق شبکه بررسی می‌کند. معماری YOLO برای کاربردهای بلادرنگ بسیار مناسب است. نسخه‌های مختلفی مانند YOLOv5، YOLOv8 و YOLOv11 تاکنون توسعه یافته‌اند.

- **Faster R-CNN**: یکی از الگوریتم‌های دقیق تشخیص شیء است که ابتدا نواحی پیشنهادی^{۱۲} را تولید کرده و سپس هر ناحیه را به‌صورت جداگانه تحلیل می‌کند. اگرچه از نظر دقت بسیار مناسب است، ولی از لحاظ محاسباتی سنگین‌تر از YOLO است.

انتخاب مدل مناسب بستگی به الزامات پروژه، مانند دقت مورد نیاز، محدودیت‌های زمانی، و قابلیت اجرا در محیط‌های پردازشی ساده دارد. در این پروژه، مدل تشخیص تنها برای شناسایی اولیه شیء به‌کار خواهد آمد و پس از آن، تعقیب مستقل از تشخیص ادامه می‌یابد. بنابراین، مدل انتخاب‌شده باید قادر باشد در فریم‌های آغازین، موقعیت دقیق و قابل اعتمادی از شیء هدف ارائه دهد.

^{۱۰}Convolutional Neural Networks – CNN

^{۱۱}Fine-tuning

^{۱۲}Region Proposals

۳ تعقیب

در این بخش با بعضی از الگوریتم‌های تعقیب آشنا می‌شویم. الگوریتم‌های مد نظر سه الگوریتم CSRT^{۱۳}، MOSSE^{۱۴} و KCF^{۱۵} هستند. در ادامه‌ی پروژه قرار است این الگوریتم‌ها را پیاده کنید و از آنها به عنوان مبنای مقایسه استفاده کنید.

۱-۳ الگوریتم CSRT

الگوریتم CSRT یکی از روش‌های پیشرفته در زمینه ردیابی اشیاء در تصاویر و ویدئوها است. این الگوریتم، از دسته الگوریتم‌های مبتنی بر فیلترهای همبستگی تفکیک‌پذیر است. هدف اصلی آن، دنبال کردن شیء در فریم‌های متوالی با دقت بالا، حتی در شرایطی مانند انسداد جزئی، چرخش یا تغییر اندازه است.

در ابتدای ردیابی، از ناحیه‌ای که شامل شیء هدف است، چند کانال ویژگی استخراج می‌شود؛ برای مثال، گرادیان تصویر یا شدت روشنایی. این ویژگی‌ها به صورت $\{x_k\}_{k=1}^K$ مدل می‌شوند که در آن K تعداد کانال‌ها و x_k تصویر ویژگی کانال k ام است. این ویژگی‌ها به عنوان ورودی برای یادگیری یک فیلتر $\{h_k\}_{k=1}^K$ استفاده می‌شوند. هدف یادگیری این فیلترها، بیشینه‌سازی پاسخ همبستگی در محل شیء هدف است. به طور خاص، پاسخ نهایی الگوریتم از طریق مجموع همبستگی‌ها به دست می‌آید:

$$R = \sum_{k=1}^K x_k \star h_k,$$

که در آن $\star$ نشان‌دهنده عمل همبستگی (یا کانولوشن دوری) است. در الگوریتم CSRT، علاوه بر استفاده از چند کانال ویژگی، از یک نگاشت اطمینان فضایی (spatial reliability map) به صورت یک تابع وزن‌دار $w(p)$ استفاده می‌شود که اهمیت هر پیکسل p را در ناحیه آموزش نشان می‌دهد. این وزن‌ها با استفاده از مدل‌های آماری برای تمایز بین ناحیه پیش‌زمینه (شیء) و پس‌زمینه (محیط) به دست می‌آیند، معمولاً با فرض مدل‌سازی هر کلاس با توزیع گاوسی چندمتغیره. یادگیری فیلترها با در نظر گرفتن این نگاشت اطمینان به صورت حل یک مسئله بهینه‌سازی انجام می‌شود، مانند:

$$\min_{h_k} \sum_p w(p) |(x_k \star h_k)(p) - y(p)|^2 + \lambda \|h_k\|^2,$$

که در آن $y(p)$ پاسخ هدف (معمولاً یک تابع گاوسی حول مرکز شیء)، و λ یک پارامتر تنظیم است که مانع بیش‌برازش می‌شود.

در هر فریم جدید، الگوریتم با اعمال فیلترهای آموخته‌شده بر روی ناحیه جستجو، نقشه پاسخ جدید را محاسبه کرده و مکان شیء را با یافتن بیشینه R به‌روزرسانی می‌کند. سپس در صورت نیاز، فیلتر نیز با توجه به ظاهر جدید شیء با یک نرخ یادگیری مشخص به‌روزرسانی می‌شود. این ویژگی به الگوریتم کمک می‌کند که با تغییرات تدریجی

^{۱۳} Discriminative Correlation Filter with Channel and Spatial Reliability
^{۱۴} Minimum Output Sum of Squared Error
^{۱۵} Kernelized Correlation Filters

در ظاهر یا موقعیت شیء سازگار شود.

در نهایت، CSRT دقت بالایی در مقایسه با الگوریتم‌هایی مانند KCF یا MOSSE دارد، اما سرعت اجرای آن پایین‌تر است. به همین دلیل، برای کاربردهایی مناسب است که دقت از سرعت مهم‌تر باشد، مانند تحلیل ویدئویی آفلاین یا سیستم‌های نظارتی دقیق.

۲-۳ الگوریتم MOSSE

الگوریتم MOSSE یکی از اولین و سریع‌ترین روش‌های مبتنی بر فیلترهای همبستگی برای ردیابی اشیاء در ویدئو است. این الگوریتم به گونه‌ای طراحی شده که بتواند با نرخ فریم بسیار بالا (حتی بیش از ۶۰۰ فریم بر ثانیه) اشیاء را دنبال کند، در حالی که مصرف محاسباتی بسیار پایینی دارد.

در MOSSE، ایده اصلی این است که فیلتر h به گونه‌ای آموزش داده شود که وقتی روی تصویر f اعمال می‌شود، خروجی حاصل نزدیک به پاسخ دلخواه g باشد. به بیان دقیق‌تر، فیلتر به گونه‌ای انتخاب می‌شود که مجموع خطاهای مربعی بین پاسخ واقعی و پاسخ دلخواه به حداقل برسد:

$$\min_h \sum_i |f_i * h - g_i|^2$$

برای افزایش سرعت محاسبات، این بهینه‌سازی در حوزه فرکانس انجام می‌شود، با استفاده از تبدیل فوریه. در این حالت، پاسخ همبستگی به صورت زیر نوشته می‌شود:

$$H = \frac{\sum_i G_i \odot F_i^*}{\sum_i F_i \odot F_i^*}$$

که در آن F_i تبدیل فوریه تصویر آموزش، G_i تبدیل فوریه پاسخ دلخواه، و $\odot$ ضرب عضو به عضو است. F_i^* نیز مزدوج مختلط F_i است.

در حین ردیابی، فیلتر H روی ناحیه جستجو در فریم جدید اعمال می‌شود تا پاسخ به‌دست‌آید، و موقعیت شیء در نقطه‌ای تعیین می‌شود که مقدار پاسخ بیشینه باشد. سپس، اگر الگوریتم تشخیص دهد که موقعیت ردیابی معتبر است، فیلتر نیز به شکل تطبیقی با استفاده از اطلاعات جدید به‌روزرسانی می‌شود.

الگوریتم MOSSE بسیار سریع است و در برابر برخی تغییرات مانند روشنایی، نویز و مقیاس‌های کوچک مقاوم می‌باشد. با این حال، چون تنها از ویژگی‌های خاکستری سطح پایین تصویر استفاده می‌کند و کانال‌های ویژگی یا اطلاعات فضایی پیچیده‌ای را در نظر نمی‌گیرد، در شرایط پیچیده مانند انسداد یا تغییرات زیاد در ظاهر شیء عملکرد کمتری دارد.

در مجموع، MOSSE الگوریتمی سبک و مناسب برای کاربردهای ردیابی بلادرنگ و سخت‌افزارهای ضعیف‌تر مانند ربات‌های سبک و سیستم‌های نهفته است.

۳-۳ الگوریتم KCF

الگوریتم KCF (Kernelized Correlation Filters) یکی از روش‌های مؤثر و سریع در ردیابی اشیاء است که بهبود یافته‌ی الگوریتم MOSSE محسوب می‌شود. تفاوت اصلی آن، استفاده از فیلترهای همبستگی در فضای ویژگی غیرخطی به کمک کرنل‌ها است. این ویژگی به الگوریتم اجازه می‌دهد تا شباهت‌های پیچیده‌تری را بین ویژگی‌های ناحیه هدف مدل کند.

در این روش، ابتدا ویژگی‌هایی مانند Histogram of Oriented Gradients (HOG) از تصویر استخراج می‌شوند. سپس هدف، یادگیری فیلتری w است که در فضای کرنل، خروجی آن با پاسخ هدف y که معمولاً به صورت یک نگاشت گاوسی است بیشترین هم‌خوانی را داشته باشد. مدل یادگیری به صورت زیر فرمول‌بندی می‌شود:

$$\min_w \sum_i |\phi(x_i)^\top w - y_i|^2 + \lambda \|w\|^2$$

که در آن:

- $\phi(x_i)$: نگاشت ویژگی از نمونه x_i به فضای کرنل
 - y_i : مقدار پاسخ هدف
 - λ : پارامتر تنظیم‌کننده برای جلوگیری از بیش‌برازش
- با استفاده از نظریه Representer Theorem، راه‌حل به صورت ترکیبی از کرنل بین نمونه‌ها به دست می‌آید:

$$\alpha = (K + \lambda I)^{-1} y$$

که در آن K ماتریس کرنل با مؤلفه‌های $K_{ij} = k(x_i, x_j)$ است. برای کاهش هزینه محاسباتی، از ساختارهای دوری (circulant) و تبدیل فوریه استفاده می‌شود:

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda}$$

در هر فریم، فیلتر یادگرفته‌شده بر ناحیه جستجو اعمال شده و مکان شیء با یافتن بیشینه پاسخ مشخص می‌شود. برای سازگاری با تغییرات ظاهری شیء، مدل به صورت تدریجی به‌روزرسانی می‌گردد:

$$\hat{\alpha}_t = (1 - \eta) \hat{\alpha}_{t-1} + \eta \hat{\alpha}_{\text{new}}$$

که η نرخ یادگیری و $\hat{\alpha}_{\text{new}}$ پاسخ حاصل از ویژگی‌های جدید است. معمولاً از کرنل گاوسی برای تعریف شباهت بین نمونه‌ها استفاده می‌شود:

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{\sigma^2}\right)$$

الگوریتم KCF با بهره‌گیری از محاسبات در حوزه فرکانس و کرنل‌های غیرخطی، سرعت بالا و دقت مناسبی ارائه می‌دهد و برای کاربردهای بلادرنگ مانند ردیابی اشیاء در ویدئوهای زنده بسیار مناسب است.

۴ الگوریتم شما

در این بخش، لازم است با بهره‌گیری از ایده‌ی خلاقانه‌ی خود و با الهام از الگوریتم‌های پیشنهادی، یک الگوریتم برای تشخیص و تعقیب شیء طراحی کنید. برای بخش تشخیص می‌توانید از مدل‌های از پیش آموزش دیده‌شده استفاده کنید، اما پیاده‌سازی بخش تعقیب باید توسط خودتان انجام شود.

در ابتدا، روشی برای استخراج وابستگی میان فریم‌های متوالی ارائه دهید. سپس توضیح دهید که در صورت تغییر اندازه‌ی جسم در تصویر (مثلاً با نزدیک یا دور شدن به دوربین)، چگونه اندازه‌ی جعبه‌ی مرزی را به‌طور دینامیک به‌روز می‌کنید تا همواره منطبق با ابعاد واقعی شیء باقی بماند. همچنین، نحوه‌ی برخورد الگوریتم با شرایطی که شیء موردنظر پشت یک جسم دیگر قرار می‌گیرد یا به‌طور موقت از میدان دید خارج می‌شود را تشریح نمایید.

برای جلوگیری از خطا در چنین موقعیت‌هایی، می‌توانید از **فیلتر کالمن**^{۱۶} جهت تخمین سرعت و شتاب جسم و پیش‌بینی مکان بعدی آن استفاده کنید.

به منظور افزایش سرعت اجرا، توصیه می‌شود بخشی از پردازش‌ها از جمله کانولوشن را در حوزه‌ی فرکانس انجام دهید؛ این روش می‌تواند با کاهش پیچیدگی محاسباتی، به بهبود نرخ پردازش کمک کند.

عملکرد الگوریتم خود را در شرایط نوری مختلف مانند وجود سایه، نور مستقیم خورشید، مه یا دود تحلیل کرده و میزان پایداری و دقت آن را در این شرایط بررسی نمایید.

در طراحی و ارزیابی الگوریتم، معیارهای زیر اهمیت زیادی دارند:

- نرخ پردازش (FPS) الگوریتم تعقیب
- پایداری در تعقیب در مواردی که جسم به‌طور موقت از دید خارج می‌شود یا پشت اجسام دیگر حرکت می‌کند
- به‌روزرسانی تطبیقی مسیر و ویژگی‌های تعقیب با توجه به تغییرات ظاهری یا حرکتی جسم
- قابلیت تعقیب در شرایط نوری متغیر
- به‌روزرسانی اندازه‌ی جعبه مرزی در هنگام تغییر مقیاس جسم (نزدیکی یا دوری نسبت به دوربین)

۵ خروجی مطلوب

یک ویدیو به همراه نوع شیء(های) مورد نظر برای تعقیب به عنوان ورودی به الگوریتم داده می‌شود. الگوریتم شما باید به‌صورت زنده عملیات پردازش را انجام داده و ویدیو استریم شده به همراه مقدار FPS (نرخ پردازش بر ثانیه) در خروجی نمایش داده شود.

تشخیص اولیه توسط یک مدل انجام شده و پس از شناسایی اولین شیء، ادامه‌ی فرآیند بر عهده‌ی الگوریتم تعقیب خواهد بود.

^{۱۶}Kalman filter

شیوهی نمره‌دهی

بخش اجباری

- ۱۵ امتیاز برای پیاده‌سازی صحیح الگوریتم تشخیص شیء
- ۲۰ امتیاز پیاده‌سازی سه الگوریتم CSRT، Moss و KFC (می‌توانید از کتابخانه‌ها موجود استفاده کنید)
- ۲۵ امتیاز برای ایده‌ی نوآورانه و طراحی خلاقانه‌ی بخش تعقیب
- ۱۰ امتیاز برای مقایسه الگوریتم‌ها تعقیب با یکدیگر از منظر دقت و نرخ پردازش
- ۱۰ امتیاز برای توانایی الگوریتم در تطبیق اندازه‌ی جعبه‌ی تعقیب با دوری و نزدیکی جسم نسبت به دوربین
- ۱۰ امتیاز برای نرخ پردازش بالا (FPS) در اجرای زنده‌ی الگوریتم (بیش از ۸۰٪ نرخ پردازشی که در الگوریتم CSRT بدست آورده‌اید).
- ۱۰ امتیاز برای حفظ پیوستگی تعقیب در هنگام خروج موقت جسم از دید یا قرار گرفتن پشت اشیای دیگر

بخش امتیازی

- ۲۰ امتیاز برای تعقیب هم‌زمان چندین شیء متفاوت (اولین شیء‌ها)، در صورتی که نرخ پردازش (FPS) و دقت الگوریتم مناسب باشد.
- ۱۰ امتیاز نوشتن گزارش پروژه با $\text{\LaTeX}$

در کل، می‌توانید ۱۳۰ امتیاز بدست آورید. با بدست آوردن ۱۰۰ امتیاز نمره کامل پروژه را بدست می‌آورید و بقیه امتیازها بصورت نمره اضافی به نمره کل شما اضافه خواهند شد.

یادآوری می‌شود که استفاده از گیت اجباری است، همچنین کامیت‌های^{۱۷} اعضا گروه نباید تفاوت چشمگیری داشته باشد. در صورتی که از گیت استفاده نکنید ۲۵ امتیاز منفی دریافت خواهید کرد. در صورتی که اعضا گروه میزان کامیت نامتقارنی داشته باشند، ۱۵ امتیاز منفی برای افرادی که کم کاری کرده‌اند در نظر گرفته خواهد شد.

الگوریتم شما باید به‌صورت end-to-end کار کند، به این معنا که با دریافت ورودی (ویدیو و نوع اشیاء هدف)، به‌طور کامل پردازش را انجام داده و فایل خروجی را تولید کند.

Commits^{۱۷}