



Object Scope

Real-Time and Image-Based Object Detection with COCO-SSD and TensorFlow.js

Project Documentation & Proposal

Prepared by: Fouad Salehi
Supervisor: Ms. Eng. Bahareh Hashemzadeh



دانشگاه آزاد اسلامی واحد مشهد

عنوان پروژه: **تشخیص بلادرنگ و مبتنی بر تصویر اشیا با استفاده از COCO-SSD و** **TensorFlow.js**

تهیه‌کننده: فؤاد صالحی
استاد راهنما: سرکار خانم مهندس بهاره هاشم‌زاده

سال ۱۴۰۵

فهرست مطالب

چکیده

۱. مقدمه

۲. بیان مسئله

۳. اهداف پروژه

۳-۱. اهداف اصلی پروژه

۳-۲. اهداف توسعه آینده

۴. مروری بر بینایی ماشین (Computer Vision)

۴-۱. معرفی بینایی ماشین

۴-۲. فرآیند پردازش تصویر در سیستم‌های بینایی ماشین

۴-۳. کاربردهای بینایی ماشین

۴-۴. ارتباط بینایی ماشین با پروژه Object Scope

۵. هوش مصنوعی و کاربرد آن در تشخیص اشیا

۵-۱. یادگیری ماشین و یادگیری عمیق

۵-۲. مفهوم تشخیص اشیا (Object Detection)

۶. فناوری‌ها و ابزارهای استفاده شده

۶-۱. TensorFlow.js

۶-۲. مدل COCO-SSD

۶-۳. JavaScript و HTML، CSS

۷. بررسی داده‌ها و محدودیت‌های مدل

۷-۱. معرفی مجموعه داده COCO

۷-۲. محدودیت‌های مدل COCO-SSD

۸. مقایسه با مدل‌های مشابه تشخیص اشیا

۸-۱. مدل YOLO (You Only Look Once)

۸-۲. مدل Faster R-CNN

۸-۳. مدل EfficientDet

۸-۴. مدل COCO-SSD

۸-۵. مقایسه نهایی مدل‌ها

۸-۶. دلیل انتخاب COCO-SSD در پروژه Object Scope

۹. معماری و طراحی سیستم

۹-۱. ساختار کلی سیستم

- ۹-۲. بخش تشخیص تصویر (Image Detection)
 - ۹-۳. بخش تشخیص زنده (Real-Time Detection)
 - ۹-۴. ساختار فایل‌های پروژه
 - ۹-۵. مزایای معماری انتخاب‌شده
 - ۱۰. مراحل پیاده‌سازی پروژه
 - ۱۰-۱. بخش تشخیص تصویر (Image Detection)
 - ۱۰-۲. بخش تشخیص دوربین (Camera Detection)
 - ۱۱. تست و ارزیابی عملکرد سیستم
 - ۱۲. چالش‌ها و راهکارهای حل مشکلات
 - ۱۳. بهبودها و توسعه‌های آینده
 - ۱۴. نتیجه‌گیری
- منابع

چکیده

امروزه هوش مصنوعی و به خصوص حوزه بینایی ماشین (Computer Vision) نقش مهمی در تحلیل و پردازش داده‌های تصویری ایفا می‌کنند. سیستم‌های تشخیص اشیا (Object Detection) یکی از کاربردهای مهم این حوزه هستند که امکان شناسایی و مکان‌یابی اشیای مختلف در تصاویر و ویدئوها را فراهم می‌کنند. این فناوری در زمینه‌هایی مانند سیستم‌های نظارتی هوشمند، خودروهای خودران، پزشکی، رباتیک و برنامه‌های تحت وب مورد استفاده قرار می‌گیرد.

هدف از این پروژه طراحی و پیاده‌سازی یک سامانه تشخیص اشیا تحت وب با نام **Object Scope** است که بتواند بدون نیاز به نصب نرم‌افزارهای جانبی، تصاویر و جریان زنده دوربین را تحلیل کرده و اشیای موجود در آن‌ها را شناسایی کند. این سیستم با استفاده از فناوری **TensorFlow.js** امکان اجرای مدل‌های یادگیری ماشین را مستقیماً در محیط مرورگر فراهم می‌کند و از مدل آماده **COCO-SSD** برای تشخیص اشیا استفاده می‌نماید.

در این پروژه دو قابلیت اصلی توسعه داده شده است. بخش اول، تشخیص اشیا از تصاویر ثابت (Image Detection) است که کاربر می‌تواند یک تصویر را بارگذاری کرده و نتایج شناسایی شامل نام اشیا و میزان اطمینان مدل (Confidence Score) را مشاهده کند. بخش دوم، تشخیص لحظه‌ای از طریق دوربین (Camera Detection) است که با استفاده از جریان ویدئویی دوربین، اشیا را به صورت Real-Time شناسایی می‌کند.

مدل **COCO-SSD** به دلیل سرعت مناسب، قابلیت اجرا در مرورگر و پشتیبانی از تعداد زیادی کلاس از اشیا برای این پروژه انتخاب شده است. با این حال، این مدل دارای محدودیت‌هایی مانند تعداد محدود کلاس‌های قابل تشخیص و وابستگی به داده‌های آموزشی عمومی است که باعث می‌شود برای کاربردهای تخصصی نیاز به آموزش مدل اختصاصی وجود داشته باشد.

نتایج این پروژه نشان می‌دهد که می‌توان با استفاده از فناوری‌های مدرن وب و مدل‌های آماده هوش مصنوعی، یک سیستم تشخیص اشیا با قابلیت اجرای مستقیم در مرورگر ایجاد کرد. **Object Scope** یک نمونه کاربردی از ترکیب هوش مصنوعی، پردازش تصویر و توسعه وب است که مسیر استفاده از مدل‌های یادگیری ماشین در برنامه‌های تحت وب را بررسی می‌کند.

کلیدواژه‌ها:

هوش مصنوعی، بینایی ماشین، تشخیص اشیا، **COCO-SSD**، **TensorFlow.js**، یادگیری عمیق، پردازش تصویر

۱. مقدمه

در سال‌های اخیر، پیشرفت سریع فناوری‌های مرتبط با هوش مصنوعی (Artificial Intelligence) باعث ایجاد تحول گسترده‌ای در نحوه پردازش و تحلیل اطلاعات شده است. یکی از مهم‌ترین شاخه‌های هوش مصنوعی، حوزه بینایی ماشین (Computer Vision) است که هدف آن توانمند کردن سیستم‌های کامپیوتری برای دریافت، پردازش و درک اطلاعات بصری مشابه توانایی مشاهده و تحلیل انسان است.

تصاویر و ویدئوها امروزه بخش بزرگی از داده‌های تولید شده در جهان را تشکیل می‌دهند. دوربین‌های دیجیتال، تلفن‌های هوشمند، سیستم‌های نظارتی و تجهیزات هوشمند به طور مداوم حجم زیادی از داده‌های تصویری تولید می‌کنند. تحلیل دستی این حجم از اطلاعات، زمان‌بر و در بسیاری از موارد غیرممکن است. به همین دلیل، استفاده از الگوریتم‌های هوش مصنوعی برای شناسایی، دسته‌بندی و تحلیل محتوای تصاویر اهمیت زیادی پیدا کرده است.

یکی از مهم‌ترین کاربردهای بینایی ماشین، تشخیص اشیا (Object Detection) است. در این فرآیند، سیستم هوشمند قادر است اشیای موجود در یک تصویر یا ویدئو را شناسایی کرده، نوع آن‌ها را مشخص کند و موقعیت تقریبی آن‌ها را در تصویر تعیین نماید. تفاوت اصلی تشخیص اشیا با دسته‌بندی تصویر (Image Classification) این است که در تشخیص اشیا، علاوه بر شناسایی نوع شیء، محل قرارگیری آن نیز مشخص می‌شود.

برای مثال، یک سیستم تشخیص اشیا می‌تواند در یک تصویر چندین شیء مختلف مانند انسان، خودرو، حیوانات یا وسایل روزمره را به صورت همزمان شناسایی کند و میزان اطمینان خود نسبت به هر تشخیص را نمایش دهد. این قابلیت باعث شده است که Object Detection در زمینه‌هایی مانند خودروهای خودران، سیستم‌های امنیتی هوشمند، رباتیک، پزشکی و نرم‌افزارهای هوشمند کاربرد گسترده‌ای پیدا کند.

با وجود پیشرفت مدل‌های قدرتمند هوش مصنوعی، بسیاری از سیستم‌های تشخیص اشیا برای اجرا نیازمند سخت‌افزارهای قدرتمند مانند پردازنده‌های گرافیکی (GPU) یا سرورهای پردازشی هستند. این موضوع باعث محدود شدن دسترسی کاربران عادی به فناوری‌های هوش مصنوعی می‌شود. یکی از راهکارهای جدید برای کاهش این محدودیت، اجرای مدل‌های یادگیری ماشین در محیط مرورگر است.

فناوری **TensorFlow.js** امکان اجرای مدل‌های یادگیری ماشین را مستقیماً در مرورگر فراهم می‌کند. این فناوری به توسعه‌دهندگان اجازه می‌دهد بدون نیاز به ارسال اطلاعات به سرور، پردازش‌های هوش مصنوعی را در سمت کاربر انجام دهند. این ویژگی علاوه بر کاهش وابستگی به زیرساخت‌های پردازشی، باعث افزایش سرعت پاسخ‌دهی و حفظ بیشتر حریم خصوصی کاربران می‌شود.

پروژه **Object Scope** با هدف بررسی قابلیت اجرای یک سیستم تشخیص اشیا در محیط وب طراحی و پیاده‌سازی شده است. در این پروژه از مدل **COCO-SSD** به عنوان مدل تشخیص اشیا استفاده شده که قابلیت شناسایی کلاس‌های مختلفی از اشیا را دارد. این مدل به دلیل سرعت مناسب و سازگاری با TensorFlow.js برای اجرای Real-Time در مرورگر انتخاب شده است.

در این سامانه، دو قابلیت اصلی پیاده‌سازی شده است. قابلیت اول، تشخیص اشیا از تصاویر ثابت است که کاربر می‌تواند یک تصویر را وارد سیستم کرده و نتایج تحلیل را مشاهده کند. قابلیت دوم، تشخیص لحظه‌ای از طریق دوربین است که امکان شناسایی اشیا در جریان ویدئویی زنده را فراهم می‌کند.

هدف اصلی این پروژه ایجاد یک نمونه عملی از ترکیب هوش مصنوعی و فناوری‌های وب است تا نشان داده شود که می‌توان بدون نیاز به نرم‌افزارهای پیچیده یا سخت‌افزارهای گران‌قیمت، قابلیت‌های پردازش تصویر را در اختیار کاربران قرار داد. همچنین این پروژه محدودیت‌ها و توانایی‌های مدل‌های آماده تشخیص اشیا را بررسی کرده و زمینه‌ای برای توسعه نسخه‌های پیشرفته‌تر در آینده فراهم می‌کند.

۲. بیان مسئله

با گسترش استفاده از سیستم‌های هوشمند و افزایش حجم داده‌های تصویری، نیاز به روش‌هایی برای تحلیل خودکار تصاویر و استخراج اطلاعات از آن‌ها روزبه‌روز بیشتر شده است. تصاویر و ویدئوها شامل اطلاعات ارزشمندی هستند که در بسیاری از کاربردها نیاز است به صورت خودکار پردازش و تحلیل شوند. انجام این فرآیند به صورت دستی علاوه بر زمان‌بر بودن، امکان خطاهای انسانی را نیز افزایش می‌دهد.

یکی از مهم‌ترین چالش‌ها در حوزه بینایی ماشین، توانایی سیستم‌های کامپیوتری در شناسایی اشیا موجود در تصاویر و محیط واقعی است. تشخیص اشیا (Object Detection) برخلاف روش‌های ساده دسته‌بندی تصویر، علاوه بر تشخیص نوع شیء، قادر است موقعیت تقریبی آن را نیز مشخص کند. این قابلیت باعث شده است که سیستم‌های تشخیص اشیا در حوزه‌هایی مانند نظارت هوشمند، خودروهای خودران، رباتیک، پزشکی و سامانه‌های امنیتی مورد استفاده قرار گیرند.

با وجود پیشرفت چشمگیر مدل‌های یادگیری عمیق، بسیاری از سیستم‌های تشخیص اشیا برای اجرا به سخت‌افزارهای قدرتمند و زیرساخت‌های پردازشی مانند GPU و سرورهای اختصاصی نیاز دارند. این موضوع باعث می‌شود استفاده از چنین سیستم‌هایی برای کاربران عادی یا در محیط‌هایی که دسترسی به منابع پردازشی محدود است، دشوار باشد.

از طرف دیگر، توسعه فناوری‌هایی مانند TensorFlow.js امکان اجرای مدل‌های یادگیری ماشین را در محیط مرورگر فراهم کرده است. این فناوری اجازه می‌دهد پردازش‌های مربوط به هوش مصنوعی بدون نیاز به ارسال داده‌ها به سرور و مستقیماً در سمت کاربر انجام شود. با این حال، استفاده از مدل‌های آماده در محیط وب نیازمند بررسی توانایی‌ها، محدودیت‌ها و میزان عملکرد آن‌ها در شرایط واقعی است.

در این پروژه، مسئله اصلی بررسی امکان پیاده‌سازی یک سیستم تشخیص اشیا تحت وب با قابلیت پردازش تصاویر ثابت و جریان زنده دوربین است. هدف این است که مشخص شود آیا می‌توان با استفاده از مدل **COCO-SSD** و فریم‌ورک **TensorFlow.js** یک سیستم Real-Time Object Detection ایجاد کرد که در مرورگر اجرا شده و توانایی شناسایی اشیای مختلف را داشته باشد.

مدل **COCO-SSD** اگرچه سرعت مناسب و قابلیت اجرای آسان در محیط وب دارد، اما به دلیل استفاده از یک مجموعه داده عمومی دارای محدودیت‌هایی است. این مدل تنها قادر به تشخیص تعداد مشخصی از کلاس‌های موجود در مجموعه داده **COCO** است و برای کاربردهای تخصصی یا تشخیص اشیای خاص، ممکن است دقت کافی نداشته باشد. بنابراین در این پروژه علاوه بر پیاده‌سازی سیستم تشخیص، عملکرد و محدودیت‌های مدل نیز مورد بررسی قرار می‌گیرد.

بر این اساس، پروژه **Object Scope** با هدف ایجاد یک نمونه عملی از تشخیص اشیا در محیط وب طراحی شده است تا قابلیت‌های مدل‌های آماده هوش مصنوعی در پردازش تصویر و ویدئوی زنده بررسی شده و امکان توسعه نسخه‌های پیشرفته‌تر در آینده فراهم شود.

۳. اهداف پروژه

هدف اصلی پروژه **Object Scope** طراحی و پیاده‌سازی یک سیستم تشخیص اشیا مبتنی بر وب است که بتواند با استفاده از فناوری‌های هوش مصنوعی، تصاویر و جریان زنده دوربین را تحلیل کرده و اشیای موجود در محیط را شناسایی کند. این پروژه با تمرکز بر استفاده از **TensorFlow.js** و مدل **COCO-SSD** تلاش می‌کند قابلیت‌های تشخیص اشیا را بدون نیاز به نصب نرم‌افزارهای اختصاصی یا استفاده از زیرساخت‌های پردازشی سنگین در اختیار کاربران قرار دهد.

اهداف این پروژه شامل بررسی قابلیت اجرای مدل‌های یادگیری ماشین در محیط مرورگر، تحلیل عملکرد یک مدل آماده تشخیص اشیا و ایجاد یک نمونه عملی از ترکیب فناوری‌های هوش مصنوعی و توسعه وب است.

۳-۱. اهداف اصلی پروژه

۱. پیاده‌سازی سیستم تشخیص اشیا در محیط وب

یکی از اهداف اصلی پروژه، ایجاد بستری است که کاربران بتوانند از طریق مرورگر به قابلیت‌های تشخیص اشیا دسترسی داشته باشند. برخلاف بسیاری از سیستم‌های سنتی که نیازمند نصب نرم‌افزار یا اتصال به سرور پردازشی هستند، Object Scope تلاش می‌کند پردازش‌های مربوط به هوش مصنوعی را مستقیماً در سمت کاربر انجام دهد.

۲. تشخیص اشیا از تصاویر ثابت (Image-Based Object Detection)

در بخش Image Detection، هدف ایجاد قابلیت است که کاربر بتواند یک تصویر را وارد سیستم کرده و مدل هوش مصنوعی اشیای موجود در آن را شناسایی کند.

این بخش شامل موارد زیر است:

- دریافت تصویر از کاربر
- پردازش تصویر توسط مدل COCO-SSD
- شناسایی اشیای موجود
- نمایش نام اشیا
- نمایش میزان اطمینان مدل (Confidence Score)

۳. تشخیص لحظه‌ای اشیا از طریق دوربین (Real-Time Object Detection)

یکی دیگر از اهداف مهم پروژه، پیاده‌سازی قابلیت تشخیص زنده از طریق دوربین است. در این بخش، سیستم به صورت Real-Time فریم‌های دریافت شده از دوربین را پردازش کرده و اشیای موجود در محیط را نمایش می‌دهد.

هدف از این بخش بررسی توانایی مدل‌های سبک هوش مصنوعی در پردازش مداوم داده‌های تصویری و اجرای آن‌ها در محیط مرورگر است.

۴. بررسی قابلیت‌های TensorFlow.js در اجرای مدل‌های هوش مصنوعی

این پروژه به بررسی امکان استفاده از TensorFlow.js برای اجرای مدل‌های یادگیری ماشین در محیط وب می‌پردازد.

اهداف این بررسی:

- کاهش وابستگی به سرورهای پردازشی
- اجرای مدل در مرورگر کاربر
- استفاده از JavaScript به عنوان بستر توسعه هوش مصنوعی
- بررسی محدودیت‌های پردازشی مرورگر

۵. بررسی عملکرد و محدودیت‌های مدل COCO-SSD

یکی از اهداف مهم پروژه، تنها استفاده از مدل آماده نیست، بلکه بررسی عملکرد واقعی آن در شرایط مختلف است.

در این پروژه موارد زیر بررسی می‌شوند:

- میزان دقت تشخیص اشیا
- سرعت پردازش مدل
- تعداد کلاس‌های قابل شناسایی
- تاثیر کیفیت تصویر بر نتیجه
- محدودیت Dataset مورد استفاده

از آنجا که COCO-SSD بر پایه مجموعه داده عمومی COCO آموزش داده شده است، ممکن است در تشخیص اشیای خارج از کلاس‌های آموزش دیده یا کاربردهای تخصصی عملکرد محدودی داشته باشد.

۶. ایجاد رابط کاربری ساده و قابل استفاده

یکی دیگر از اهداف پروژه، طراحی یک رابط کاربری مناسب برای تعامل آسان کاربران با سیستم است.

رابط طراحی شده باید امکان:

- انتخاب تصویر
- مشاهده نتایج تشخیص
- فعال‌سازی دوربین
- مشاهده خروجی Real-Time

را بدون نیاز به دانش تخصصی هوش مصنوعی فراهم کند.

۲-۳. اهداف توسعه آینده

علاوه بر اهداف فعلی، پروژه قابلیت توسعه در آینده را نیز دارد. برخی از مسیرهای پیشنهادی عبارت‌اند از:

- استفاده از مدل‌های دقیق‌تر و پیشرفته‌تر
- آموزش مدل اختصاصی با Dataset شخصی‌سازی شده
- افزایش تعداد کلاس‌های قابل تشخیص
- ذخیره نتایج تشخیص کاربران
- توسعه نسخه موبایل
- استفاده از پردازش ابری برای مدل‌های سنگین‌تر

جمع‌بندی اهداف

در مجموع، هدف پروژه **Object Scope** ایجاد یک سیستم عملی برای نمایش قابلیت‌های تشخیص اشیا مبتنی بر هوش مصنوعی در محیط وب است. این پروژه علاوه بر پیاده‌سازی یک سیستم کاربردی، به بررسی محدودیت‌ها و توانایی‌های مدل‌های آماده مانند COCO-SSD پرداخته و نشان می‌دهد چگونه می‌توان فناوری‌های یادگیری ماشین را با برنامه‌های تحت وب ترکیب کرد.

۴. مروری بر بینایی ماشین (Computer Vision)

۴-۱. معرفی بینایی ماشین

بینایی ماشین (Computer Vision) یکی از شاخه‌های مهم هوش مصنوعی (Artificial Intelligence) است که هدف آن ایجاد توانایی در سیستم‌های کامپیوتری برای دریافت، پردازش و تفسیر اطلاعات بصری از محیط اطراف است. به بیان ساده، بینایی ماشین تلاش می‌کند قابلیت مشاهده و درک تصاویر را که به صورت طبیعی در انسان وجود دارد، برای ماشین‌ها شبیه‌سازی کند.

انسان با استفاده از سیستم بینایی خود می‌تواند اشیا، افراد، محیط‌ها و ارتباط میان اجزای مختلف یک تصویر را به سرعت تشخیص دهد. اما برای کامپیوتر، یک تصویر تنها مجموعه‌ای از اعداد و مقادیر پیکسلی است که بدون استفاده از الگوریتم‌های پردازشی، معنای مشخصی ندارند. وظیفه بینایی ماشین تبدیل این داده‌های خام تصویری به اطلاعات قابل فهم برای سیستم‌های کامپیوتری است.

با پیشرفت قدرت پردازشی سیستم‌ها و ظهور الگوریتم‌های یادگیری عمیق (Deep Learning)، بینایی ماشین از روش‌های سنتی پردازش تصویر فاصله گرفته و به یکی از سریع‌ترین حوزه‌های توسعه هوش مصنوعی تبدیل شده است.

۲-۴. فرآیند پردازش تصویر در سیستم‌های بینایی ماشین

یک سیستم بینایی ماشین معمولاً شامل چند مرحله اصلی است:

۱. دریافت داده تصویری (Image Acquisition)

در اولین مرحله، سیستم داده تصویری را از یک منبع دریافت می‌کند. این منبع می‌تواند شامل موارد زیر باشد:

- دوربین دیجیتال
- وب‌کم
- تصاویر ذخیره شده
- سنسورهای تصویری

در پروژه Object Scope، ورودی سیستم از دو روش دریافت می‌شود:

- تصویر بارگذاری شده توسط کاربر
- جریان زنده دوربین (Camera Stream)

۲. پیش‌پردازش تصویر (Image Preprocessing)

قبل از تحلیل تصویر، معمولاً عملیات‌هایی برای آماده‌سازی داده انجام می‌شود. این عملیات می‌توانند شامل موارد زیر باشند:

- تغییر اندازه تصویر (Resizing)
- تنظیم کیفیت تصویر
- حذف نویز
- تبدیل فرمت داده

هدف این مرحله آماده کردن ورودی مناسب برای مدل هوش مصنوعی است.

۳. استخراج ویژگی‌ها (Feature Extraction)

در روش‌های قدیمی پردازش تصویر، ویژگی‌های خاصی مانند لبه‌ها، رنگ‌ها و اشکال هندسی به صورت دستی استخراج می‌شدند. اما در روش‌های جدید مبتنی بر یادگیری عمیق، شبکه‌های عصبی قادر هستند ویژگی‌های مهم تصویر را به صورت خودکار یاد بگیرند.

۴. تحلیل و تشخیص (Detection and Recognition)

در این مرحله، مدل هوش مصنوعی داده ورودی را بررسی کرده و اطلاعات مورد نیاز را استخراج می‌کند.

در سیستم‌های تشخیص اشیا، خروجی معمولاً شامل موارد زیر است:

- نام شیء شناسایی شده
- موقعیت شیء در تصویر
- میزان اطمینان مدل (Confidence Score)

برای مثال:

Object: Car
Confidence: 94%

یعنی مدل با احتمال ۹۴ درصد تشخیص داده است که شیء موجود در تصویر خودرو است.

۳-۴. کاربردهای بینایی ماشین

بینایی ماشین امروزه در حوزه‌های مختلف مورد استفاده قرار می‌گیرد:

خودروهای خودران

خودروهای هوشمند با استفاده از دوربین‌ها و مدل‌های بینایی ماشین قادر هستند:

- تشخیص خودروهای دیگر
- تشخیص عابر پیاده
- تشخیص علائم رانندگی

را انجام دهند.

سیستم‌های امنیتی و نظارتی

در سیستم‌های نظارتی هوشمند، بینایی ماشین برای:

- تشخیص افراد
- شناسایی رفتارهای غیرعادی
- کنترل دسترسی

استفاده می‌شود.

پزشکی

در حوزه پزشکی، مدل‌های بینایی ماشین می‌توانند برای تحلیل تصاویر پزشکی مانند:

- تصاویر X-Ray
- MRI
- تصاویر میکروسکوپی

به منظور کمک به تشخیص بیماری‌ها استفاده شوند.

رباتیک

ربات‌ها برای تعامل با محیط اطراف خود نیازمند درک اطلاعات بصری هستند. بینایی ماشین به ربات‌ها کمک می‌کند:

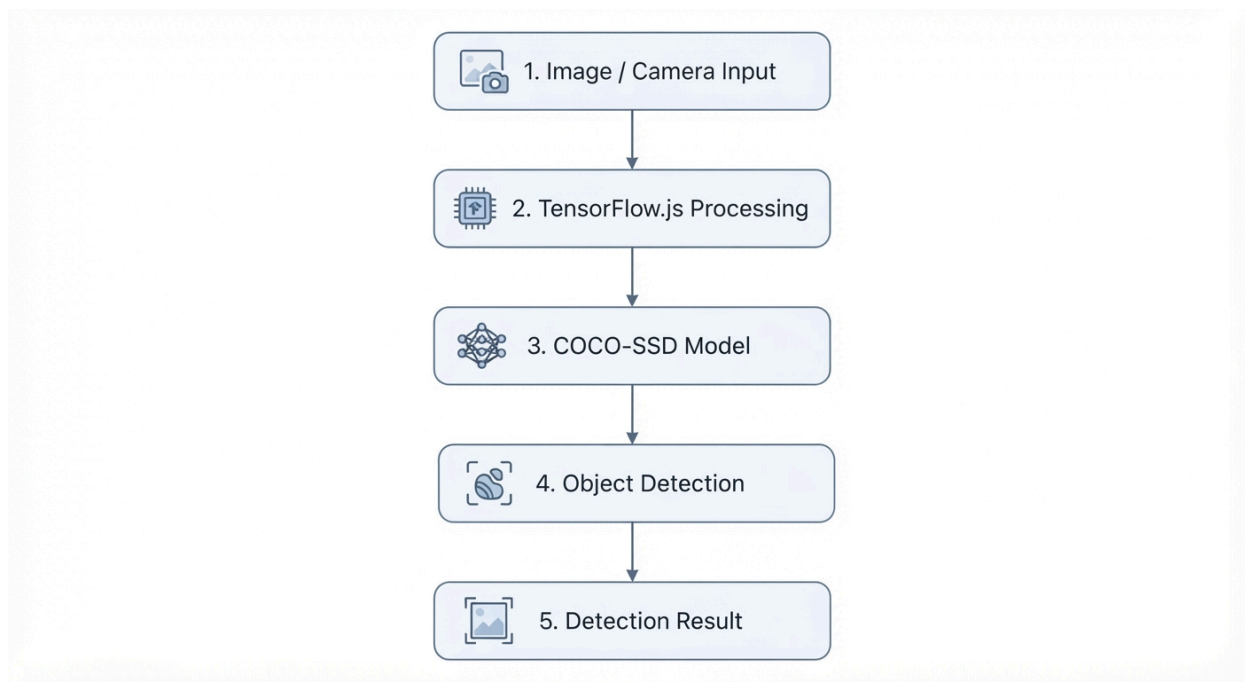
- اشیا را پیدا کنند
- مسیر حرکت را تشخیص دهند
- با محیط تعامل داشته باشند

۴-۴. ارتباط بینایی ماشین با پروژه Object Scope

پروژه Object Scope در حوزه بینایی ماشین و به صورت مشخص در زمینه **Object Detection** قرار می‌گیرد. هدف این پروژه استفاده از مدل‌های هوش مصنوعی برای شناسایی اشیای مختلف در تصاویر و ویدئوهای زنده است.

در این سیستم، داده ورودی ابتدا از تصویر یا دوربین دریافت شده، سپس توسط مدل **COCO-SSD** پردازش می‌شود. مدل پس از تحلیل داده، اشیای موجود را شناسایی کرده و نتیجه شامل نام شیء و میزان اطمینان تشخیص را نمایش می‌دهد.

ساختار کلی فرآیند در Object Scope به صورت زیر است:



بنابراین این پروژه نمونه‌ای عملی از استفاده از مفاهیم بینایی ماشین در یک محیط تحت وب است که نشان می‌دهد چگونه می‌توان قابلیت‌های هوش مصنوعی را با فناوری‌های مدرن وب ترکیب کرد.

۵. هوش مصنوعی و کاربرد آن در تشخیص اشیا

هوش مصنوعی (Artificial Intelligence) یکی از مهم‌ترین شاخه‌های علوم کامپیوتر است که با هدف طراحی سیستم‌هایی توسعه یافته است که بتوانند وظایفی را انجام دهند که معمولاً به هوش انسانی نیاز دارند. این وظایف شامل یادگیری، تصمیم‌گیری، استدلال، حل مسئله، تشخیص الگو و درک اطلاعات محیط است.

در سال‌های اخیر، پیشرفت چشمگیر در توان پردازشی رایانه‌ها، توسعه الگوریتم‌های یادگیری ماشین و افزایش حجم داده‌های آموزشی، باعث شده است هوش مصنوعی به یکی از فناوری‌های کلیدی در صنایع مختلف تبدیل

شود. امروزه از هوش مصنوعی در زمینه‌هایی مانند پزشکی، حمل‌ونقل، امنیت، تجارت الکترونیک، پردازش زبان طبیعی، رباتیک و بینایی ماشین استفاده می‌شود.

یکی از مهم‌ترین کاربردهای هوش مصنوعی، تحلیل داده‌های تصویری است. تصاویر حاوی حجم زیادی از اطلاعات هستند که استخراج آن‌ها به صورت دستی علاوه بر صرف زمان زیاد، احتمال بروز خطا را نیز افزایش می‌دهد. الگوریتم‌های هوش مصنوعی قادر هستند این اطلاعات را با سرعت بالا تحلیل کرده و الگوهای موجود در تصاویر را شناسایی کنند.

پروژه **Object Scope** نیز بر پایه همین قابلیت توسعه یافته است. در این پروژه از فناوری‌های هوش مصنوعی برای تشخیص اشیای موجود در تصاویر و ویدئوهای زنده استفاده شده است. هدف اصلی، بررسی امکان اجرای مدل‌های هوش مصنوعی در محیط مرورگر و ارائه یک سامانه تحت وب برای تشخیص اشیا است.

هوش مصنوعی به تنهایی یک فناوری واحد نیست، بلکه مجموعه‌ای از روش‌ها و الگوریتم‌های مختلف را در بر می‌گیرد. در حوزه تشخیص اشیا، مهم‌ترین فناوری مورد استفاده، یادگیری ماشین (Machine Learning) و به طور خاص یادگیری عمیق (Deep Learning) است که در ادامه به آن‌ها پرداخته می‌شود.

۱-۵. یادگیری ماشین و یادگیری عمیق

یادگیری ماشین (Machine Learning) یکی از زیرشاخه‌های اصلی هوش مصنوعی است که به سیستم‌های کامپیوتری این امکان را می‌دهد تا بدون برنامه‌نویسی مستقیم برای هر حالت، از داده‌ها الگوها را یاد گرفته و بر اساس آن‌ها تصمیم‌گیری کنند.

در روش‌های سنتی برنامه‌نویسی، تمامی قوانین توسط برنامه‌نویس تعریف می‌شوند. اما در یادگیری ماشین، سیستم با مشاهده حجم زیادی از داده‌های آموزشی، روابط میان داده‌ها را فرا گرفته و پس از پایان آموزش، قادر خواهد بود داده‌های جدید را نیز تحلیل کند.

به عنوان مثال، اگر هزاران تصویر از خودرو، انسان و حیوانات در اختیار یک مدل قرار گیرد، مدل به مرور ویژگی‌های مربوط به هر دسته را یاد گرفته و در آینده می‌تواند تصاویر جدید را نیز با احتمال مناسبی طبقه‌بندی یا شناسایی کند.

با افزایش پیچیدگی مسائل، روش‌های سنتی یادگیری ماشین با محدودیت‌هایی مواجه شدند. به همین دلیل شاخه‌ای پیشرفته‌تر با عنوان **یادگیری عمیق (Deep Learning)** توسعه یافت.

یادگیری عمیق از شبکه‌های عصبی مصنوعی (Artificial Neural Networks) با لایه‌های متعدد تشکیل شده است. این شبکه‌ها توانایی استخراج ویژگی‌های پیچیده تصاویر را بدون نیاز به طراحی دستی ویژگی‌ها دارند. به

همین دلیل امروزه تقریباً تمامی مدل‌های پیشرفته تشخیص تصویر و تشخیص اشیا بر پایه یادگیری عمیق توسعه داده می‌شوند.

یکی از مهم‌ترین مزایای یادگیری عمیق، توانایی آن در یادگیری خودکار ویژگی‌های مهم تصویر است. به عنوان نمونه، در تشخیص یک خودرو، مدل نیازی به تعریف دستی ویژگی‌هایی مانند چرخ، شیشه یا چراغ ندارد، بلکه این ویژگی‌ها را در فرآیند آموزش به صورت خودکار فرا می‌گیرد.

مدل **COCO-SSD** که در پروژه Object Scope مورد استفاده قرار گرفته است نیز بر پایه شبکه‌های عصبی عمیق طراحی شده و پس از آموزش روی مجموعه داده COCO، قادر به تشخیص ده‌ها کلاس مختلف از اشیا شده است.

۲-۵. مفهوم تشخیص اشیا (Object Detection)

تشخیص اشیا (Object Detection) یکی از مهم‌ترین کاربردهای بینایی ماشین است که هدف آن شناسایی اشیای موجود در یک تصویر و تعیین موقعیت آن‌ها است. برخلاف دسته‌بندی تصویر (Image Classification) که تنها نوع کلی تصویر را مشخص می‌کند، در تشخیص اشیا علاوه بر تعیین نوع شیء، محل قرارگیری هر شیء نیز مشخص می‌شود.

به عنوان مثال، اگر تصویری شامل یک خودرو، دو انسان و یک گربه باشد، یک سیستم دسته‌بندی تصویر تنها ممکن است اعلام کند که تصویر شامل چند شیء مختلف است. اما یک سیستم تشخیص اشیا قادر است هر یک از این اشیا را به صورت جداگانه شناسایی کرده، محدوده قرارگیری آن‌ها را با استفاده از کادر (Bounding Box) مشخص کند و میزان اطمینان خود نسبت به هر تشخیص را نمایش دهد.

خروجی یک سیستم تشخیص اشیا معمولاً شامل سه بخش اصلی است:

- نام شیء شناسایی‌شده (Object Class)
- موقعیت شیء در تصویر (Bounding Box)
- میزان اطمینان مدل (Confidence Score)

در پروژه Object Scope، پس از دریافت تصویر یا جریان زنده دوربین، داده‌ها توسط TensorFlow.js به مدل COCO-SSD ارسال می‌شوند. مدل تصویر را تحلیل کرده و برای هر شیء شناسایی‌شده، اطلاعات مربوط به کلاس، مختصات کادر تشخیص و میزان اطمینان را تولید می‌کند. سپس این اطلاعات در رابط کاربری به کاربر نمایش داده می‌شود.

یکی از ویژگی‌های مهم سیستم طراحی‌شده در این پروژه، قابلیت اجرای فرآیند تشخیص اشیا به صورت **Real-Time** است. در حالت استفاده از دوربین، هر فریم ویدئو به طور مداوم توسط مدل تحلیل شده و نتایج تقریباً به صورت لحظه‌ای نمایش داده می‌شوند. در حالت استفاده از تصاویر ثابت نیز، سیستم پس از بارگذاری تصویر توسط کاربر، تنها یک بار عملیات تشخیص را انجام داده و نتیجه را ارائه می‌کند.

اگرچه مدل COCO-SSD از نظر سرعت عملکرد بسیار مناسبی دارد، اما مانند سایر مدل‌های مبتنی بر داده‌های آموزشی، محدودیت‌هایی نیز دارد. دقت تشخیص به عواملی مانند کیفیت تصویر، نور محیط، زاویه دید، فاصله شیء از دوربین، اندازه شیء و همچنین کلاس‌های موجود در مجموعه داده آموزشی وابسته است. بنابراین انتظار نمی‌رود این مدل بتواند تمامی اشیای موجود در محیط را با دقت کامل شناسایی کند.

با وجود این محدودیت‌ها، مدل COCO-SSD به دلیل سرعت بالا، حجم مناسب، قابلیت اجرا در مرورگر و سازگاری با TensorFlow.js، گزینه‌ای مناسب برای پروژه‌های آموزشی، نمونه‌های اولیه (Prototype) و برنامه‌های تحت وب محسوب می‌شود. به همین دلیل در پروژه **Object Scope** این مدل به عنوان هسته اصلی سیستم تشخیص اشیا انتخاب شده است.

۶. فناوری‌ها و ابزارهای استفاده شده

در توسعه پروژه **Object Scope** از مجموعه‌ای از فناوری‌های مدرن وب و ابزارهای هوش مصنوعی استفاده شده است. انتخاب این فناوری‌ها با هدف ایجاد یک سامانه سبک، سریع، قابل اجرا در مرورگر و بدون نیاز به نصب نرم‌افزارهای جانبی انجام شده است.

در این پروژه تلاش شده است تا تمامی عملیات تشخیص اشیا در سمت کاربر (Client-Side) انجام شود. به همین دلیل از فناوری‌هایی استفاده شده است که امکان اجرای مدل‌های یادگیری ماشین را مستقیماً در مرورگر فراهم می‌کنند.

مهم‌ترین فناوری‌های مورد استفاده در این پروژه عبارت‌اند از:

- TensorFlow.js
- COCO-SSD
- HTML5
- CSS3
- JavaScript

در ادامه، هر یک از این فناوری‌ها به صورت جداگانه بررسی می‌شوند.

۶-۱. TensorFlow.js

TensorFlow.js یکی از کتابخانه‌های متن‌باز شرکت Google است که امکان اجرای مدل‌های یادگیری ماشین و یادگیری عمیق را در محیط JavaScript فراهم می‌کند. این کتابخانه نسخه‌ای از فریم‌ورک TensorFlow محسوب می‌شود که به طور ویژه برای برنامه‌های تحت وب طراحی شده است.

مهم‌ترین ویژگی TensorFlow.js این است که مدل‌های هوش مصنوعی را مستقیماً در مرورگر اجرا می‌کند و نیازی به ارسال تصاویر یا داده‌ها به سرور ندارد. این ویژگی باعث کاهش زمان پاسخ‌گویی، افزایش سرعت پردازش و حفظ بهتر حریم خصوصی کاربران می‌شود.

TensorFlow.js قادر است از قابلیت‌های سخت‌افزاری مرورگر مانند WebGL و در برخی محیط‌ها WebGPU برای افزایش سرعت محاسبات استفاده کند. به همین دلیل اجرای مدل‌های نسبتاً پیچیده نیز در بسیاری از مرورگرهای مدرن امکان‌پذیر است.

در پروژه TensorFlow.js، Object Scope وظیفه بارگذاری مدل COCO-SSD، آماده‌سازی داده‌های ورودی، اجرای عملیات تشخیص اشیا و مدیریت ارتباط میان رابط کاربری و مدل هوش مصنوعی را بر عهده دارد.

مزایای استفاده از TensorFlow.js

- اجرای مدل‌های هوش مصنوعی بدون نیاز به سرور
- متن‌باز و رایگان
- سازگاری با اکثر مرورگرهای مدرن
- امکان استفاده از شتاب‌دهنده‌های سخت‌افزاری مرورگر
- نصب و راه‌اندازی آسان
- مناسب برای پروژه‌های آموزشی و تحقیقاتی

محدودیت‌های TensorFlow.js

- عملکرد ضعیف‌تر نسبت به نسخه Python در مدل‌های بسیار بزرگ
- وابستگی به توان سخت‌افزاری دستگاه کاربر
- محدودیت حافظه در مرورگر
- کاهش سرعت در دستگاه‌های قدیمی یا تلفن‌های همراه ضعیف

با وجود این محدودیت‌ها، TensorFlow.js یکی از بهترین گزینه‌ها برای توسعه برنامه‌های هوش مصنوعی مبتنی بر وب محسوب می‌شود و به همین دلیل در این پروژه انتخاب شده است.

۲-۶. مدل COCO-SSD

COCO-SSD یکی از مدل‌های آماده تشخیص اشیا است که برای اجرای سریع و سبک طراحی شده است. این مدل بر پایه معماری **(Single Shot Detector (SSD** توسعه یافته و با استفاده از مجموعه داده معروف **COCO (Common Objects in Context)** آموزش دیده است.

مجموعه داده COCO شامل بیش از ۳۳۰ هزار تصویر و میلیون‌ها نمونه برچسب‌گذاری‌شده است و یکی از پرکاربردترین Dataset‌های حوزه بینایی ماشین به شمار می‌رود. مدل COCO-SSD توانایی تشخیص حدود ۸۰ کلاس مختلف از اشیای رایج را دارد؛ از جمله:

- Person
- Car
- Bicycle
- Dog
- Cat
- Chair
- Cell Phone
- Laptop
- Bottle
- Book
- Bus
- Truck
- Bird
- Cup
- Keyboard

در پروژه Object Scope از نسخه **Lite MobileNet V2** این مدل استفاده شده است. دلیل انتخاب این نسخه، حجم کمتر، سرعت بالاتر و مناسب بودن آن برای اجرای مستقیم در مرورگر است.

نحوه عملکرد مدل

پس از دریافت تصویر یا فریم دوربین، مدل مراحل زیر را انجام می‌دهد:

1. دریافت تصویر ورودی
2. استخراج ویژگی‌های تصویر توسط شبکه عصبی
3. تشخیص اشیای موجود
4. تعیین مختصات Bounding Box
5. محاسبه میزان اطمینان (Confidence Score)
6. ارسال نتیجه به رابط کاربری

در نهایت، سیستم برای هر شیء، نام کلاس، مختصات محل قرارگیری و درصد اطمینان را نمایش می‌دهد.

مزایای COCO-SSD

- سرعت اجرای بالا
- حجم مناسب مدل
- سازگاری کامل با TensorFlow.js
- مناسب برای پردازش Real-Time
- نصب آسان
- مناسب برای نمونه‌های اولیه و پروژه‌های آموزشی

محدودیت‌های COCO-SSD

اگرچه این مدل عملکرد مناسبی دارد، اما محدودیت‌هایی نیز دارد که باید در نظر گرفته شوند:

- قابلیت تشخیص تنها حدود ۸۰ کلاس از اشیا
- عدم توانایی در تشخیص اشیای تخصصی یا سفارشی
- کاهش دقت در تصاویر با نور نامناسب
- کاهش عملکرد در تصاویر تاریک یا کم‌کیفیت
- کاهش دقت برای اشیای کوچک یا دور از دوربین
- نیاز به آموزش مجدد برای کاربردهای تخصصی

به همین دلیل، COCO-SSD بیشتر برای نمونه‌های اولیه، آموزش، تحقیقات و پروژه‌های تحت وب مناسب است و در سامانه‌های صنعتی معمولاً از مدل‌های پیشرفته‌تر مانند YOLO، Faster R-CNN یا EfficientDet استفاده می‌شود.

۳-۶. HTML، CSS و JavaScript

برای توسعه رابط کاربری و پیاده‌سازی بخش‌های مختلف سامانه Object Scope از فناوری‌های استاندارد توسعه وب استفاده شده است.

HTML5

HTML5 ساختار اصلی صفحات وب را تشکیل می‌دهد. تمامی اجزای رابط کاربری شامل منوها، دکمه‌ها، فرم بارگذاری تصویر، بخش نمایش نتایج، صفحه معرفی پروژه و صفحات مربوط به تشخیص تصویر و دوربین با استفاده از HTML طراحی شده‌اند.

CSS3

CSS3 مسئول طراحی ظاهری سامانه است. در این پروژه از CSS برای طراحی رابط کاربری مدرن، ایجاد ساختار Responsive، مدیریت رنگ‌بندی، انیمیشن‌ها، نمایش مناسب در دستگاه‌های مختلف و افزایش تجربه کاربری استفاده شده است.

JavaScript

JavaScript هسته اصلی منطق پروژه محسوب می‌شود. تمامی عملیات مربوط به تعامل کاربر با سیستم، بارگذاری مدل، دریافت تصاویر، پردازش داده‌ها، ارتباط با TensorFlow.js، اجرای عملیات تشخیص اشیا و نمایش نتایج توسط JavaScript انجام می‌شود.

همچنین بخش‌هایی مانند:

- مدیریت دوربین
- بارگذاری تصاویر
- رسم Bounding Box روی Canvas
- نمایش نتایج تشخیص
- مدیریت وضعیت بارگذاری مدل
- کنترل منوی واکنش‌گرا

همگی با استفاده از JavaScript پیاده‌سازی شده‌اند.

جمع‌بندی

در این پروژه، ترکیب فناوری‌های **HTML5، CSS3 و JavaScript** با ابزارهای هوش مصنوعی **TensorFlow.js** و **COCO-SSD** امکان توسعه یک سامانه تشخیص اشیا کاملاً مبتنی بر وب را فراهم کرده است. این ترکیب نشان می‌دهد که بدون نیاز به نرم‌افزارهای جانبی یا پردازش سمت سرور، می‌توان یک سیستم کاربردی برای تشخیص اشیا در تصاویر و ویدئوهای زنده ایجاد کرد. انتخاب این فناوری‌ها بر اساس معیارهایی مانند سرعت اجرا، سادگی توسعه، قابلیت اجرا در مرورگر و مناسب بودن برای پروژه‌های آموزشی و تحقیقاتی انجام شده است.

۷. بررسی داده‌ها و محدودیت‌های مدل

هر مدل یادگیری عمیق برای دستیابی به عملکرد مناسب، نیازمند آموزش بر روی حجم زیادی از داده‌های برچسب‌گذاری‌شده است. کیفیت، تنوع و حجم این داده‌ها تأثیر مستقیمی بر دقت و قابلیت تعمیم مدل دارند. به همین دلیل، انتخاب مجموعه داده (Dataset) مناسب یکی از مهم‌ترین مراحل توسعه سیستم‌های مبتنی بر هوش مصنوعی محسوب می‌شود.

مدل **COCO-SSD** که در پروژه **Object Scope** مورد استفاده قرار گرفته است، پیش از انتشار توسط پژوهشگران آموزش داده شده و نیازی به آموزش مجدد در این پروژه ندارد. با این حال، شناخت ساختار داده‌های آموزشی و محدودیت‌های آن برای تحلیل نتایج و ارزیابی عملکرد مدل ضروری است.

در این فصل، ابتدا مجموعه داده COCO معرفی شده و سپس محدودیت‌های مدل COCO-SSD مورد بررسی قرار می‌گیرد.

۷-۱. معرفی مجموعه داده COCO

COCO که مخفف **Common Objects in Context** است، یکی از مشهورترین و پرکاربردترین مجموعه داده‌های حوزه بینایی ماشین محسوب می‌شود. این مجموعه داده با هدف آموزش و ارزیابی مدل‌های تشخیص اشیا، بخش‌بندی تصویر (Segmentation)، تشخیص افراد و سایر وظایف مرتبط با بینایی ماشین طراحی شده است.

یکی از ویژگی‌های مهم COCO، استفاده از تصاویر واقعی به جای تصاویر ساده یا آزمایشگاهی است. در بسیاری از تصاویر این مجموعه داده، چندین شیء به طور هم‌زمان حضور دارند و اشیا در شرایط مختلفی مانند زوایای

متفاوت، نورهای متنوع، اندازه‌های مختلف و پس‌زمینه‌های پیچیده دیده می‌شوند. این موضوع باعث می‌شود مدل‌هایی که با این داده‌ها آموزش می‌بینند، توانایی بهتری در مواجهه با تصاویر واقعی داشته باشند.

نسخه‌های مختلف این مجموعه داده در طول سال‌های گذشته منتشر شده‌اند و شامل صدها هزار تصویر و میلیون‌ها شیء برچسب‌گذاری شده هستند. در نسخه‌های رایج COCO، مدل‌ها بر روی **حدود ۸۰ کلاس اصلی** از اشیای پرکاربرد آموزش می‌بینند.

برخی از کلاس‌های موجود در این مجموعه داده عبارت‌اند از:

- Person •
- Car •
- Bicycle •
- Motorcycle •
- Bus •
- Truck •
- Dog •
- Cat •
- Bird •
- Chair •
- Bottle •
- Cup •
- Cell Phone •
- Laptop •
- Keyboard •
- Book •
- Backpack •
- Handbag •
- TV •
- Clock •

تمامی این اشیای دارای برچسب (Label) و مختصات دقیق محل قرارگیری در تصویر هستند. این اطلاعات به مدل اجازه می‌دهد علاوه بر تشخیص نوع شیء، موقعیت آن را نیز مشخص کند.

در پروژه **Object Scope**، مدل COCO-SSD از همین مجموعه داده از پیش آموزش دیده استفاده می کند و بنابراین تنها قادر به شناسایی کلاس هایی است که در Dataset COCO وجود دارند.

ویژگی های مجموعه داده COCO

مجموعه داده COCO دارای ویژگی های متعددی است که آن را به یکی از استانداردهای اصلی حوزه بینایی ماشین تبدیل کرده است:

- حجم بسیار بالای تصاویر آموزشی
- تنوع زیاد اشیا و محیط ها
- تصاویر واقعی از محیط های روزمره
- وجود چندین شیء در یک تصویر
- برچسب گذاری دقیق موقعیت اشیا
- استفاده گسترده در تحقیقات دانشگاهی و پروژه های صنعتی

به دلیل همین ویژگی ها، بسیاری از مدل های مطرح تشخیص اشیا از جمله YOLO، COCO-SSD و Faster R-CNN برای آموزش اولیه از این مجموعه داده استفاده کرده اند.

۷-۲. محدودیت های مدل COCO-SSD

اگرچه COCO-SSD یکی از مناسب ترین مدل ها برای اجرای سریع در مرورگر است، اما مانند هر مدل یادگیری ماشین دارای محدودیت هایی است که باید در هنگام استفاده مورد توجه قرار گیرند.

۱. محدود بودن کلاس های قابل تشخیص

یکی از مهم ترین محدودیت های این مدل، وابستگی کامل آن به Dataset COCO است. از آنجا که این مجموعه داده تنها شامل حدود ۸۰ کلاس است، مدل نیز فقط همین اشیا را تشخیص می دهد.

به عنوان مثال، اگر تصویری شامل یک وسیله صنعتی خاص، تجهیزات پزشکی، قطعات الکترونیکی تخصصی یا اشیای سفارشی باشد، احتمال زیادی وجود دارد که مدل قادر به شناسایی آن ها نباشد.

۲. وابستگی به کیفیت تصویر

عملکرد مدل تا حد زیادی به کیفیت تصویر ورودی بستگی دارد. عواملی مانند:

- نور کم
- تصاویر تار
- کیفیت پایین دوربین
- نویز زیاد
- زاویه نامناسب

می‌توانند باعث کاهش دقت تشخیص شوند.

۳. کاهش دقت در اشیای کوچک

مدل COCO-SSD در شناسایی اشیایی که اندازه بسیار کوچکی در تصویر دارند، عملکرد ضعیف‌تری نسبت به مدل‌های پیشرفته‌تر دارد. هرچه شیء کوچک‌تر یا دورتر باشد، احتمال عدم تشخیص یا تشخیص اشتباه افزایش می‌یابد.

۴. کاهش عملکرد در تصاویر شلوغ

در تصاویری که تعداد زیادی شیء در کنار یکدیگر قرار دارند یا اشیای بخشی از یکدیگر را پوشانده‌اند (Occlusion)، مدل ممکن است برخی از اشیای تشخیص ندهد یا آن‌ها را با یکدیگر اشتباه بگیرد.

۵. محدودیت در کاربردهای تخصصی

COCO-SSD برای کاربردهای عمومی طراحی شده است و برای حوزه‌های تخصصی مانند:

- پزشکی
- صنایع نظامی
- کشاورزی هوشمند
- کنترل کیفیت صنعتی
- تصاویر ماهواره‌ای

به تنهایی مناسب نیست. در چنین کاربردهایی معمولاً مدل باید با استفاده از داده‌های اختصاصی آموزش مجدد (Fine-Tuning) ببیند.

۶. وابستگی به توان سخت‌افزاری کاربر

در پروژه Object Scope تمامی پردازش‌ها در مرورگر کاربر انجام می‌شوند. بنابراین سرعت اجرای مدل به عواملی مانند قدرت پردازنده (CPU)، کارت گرافیک (GPU)، حافظه سیستم و مرورگر مورد استفاده وابسته است.

در دستگاه‌های ضعیف‌تر ممکن است سرعت تشخیص کاهش یابد یا نرخ پردازش فریم‌ها کمتر شود.

۷. احتمال تشخیص اشتباه (False Detection)

مانند سایر مدل‌های یادگیری عمیق، COCO-SSD نیز در برخی شرایط ممکن است دچار خطا شود. برای مثال، اشیایی که از نظر شکل یا رنگ شباهت زیادی به یکدیگر دارند، ممکن است به اشتباه در یک کلاس قرار گیرند. همچنین گاهی مدل برای یک شیء واقعی چند تشخیص متفاوت ثبت می‌کند یا شیئی را تشخیص می‌دهد که در واقع وجود ندارد.

به همین دلیل، مقدار **Confidence Score** همراه با هر نتیجه نمایش داده می‌شود تا میزان اطمینان مدل نسبت به تشخیص مشخص باشد.

جمع‌بندی

مدل COCO-SSD به دلیل سرعت بالا، حجم مناسب و قابلیت اجرا در مرورگر، گزینه‌ای بسیار مناسب برای پروژه‌های آموزشی، تحقیقاتی و نمونه‌های اولیه محسوب می‌شود. با این حال، محدودیت‌های آن از جمله وابستگی به مجموعه داده COCO، تعداد محدود کلاس‌های قابل تشخیص و کاهش دقت در شرایط خاص، نشان می‌دهد که برای کاربردهای صنعتی یا تخصصی، استفاده از مدل‌های پیشرفته‌تر یا آموزش مدل اختصاصی ضروری است.

در پروژه **Object Scope** نیز این مدل با هدف بررسی عملی فناوری تشخیص اشیا در محیط وب انتخاب شده است. نتایج حاصل از اجرای پروژه نشان می‌دهد که COCO-SSD می‌تواند در بسیاری از سناریوهای عمومی عملکرد مناسبی ارائه دهد، اما شناخت محدودیت‌های آن برای تفسیر صحیح نتایج و توسعه نسخه‌های آینده اهمیت زیادی دارد.

۸. مقایسه با مدل‌های مشابه تشخیص اشیا

امروزه مدل‌های متعددی برای تشخیص اشیا توسعه یافته‌اند که هر یک با هدف ایجاد تعادل میان سرعت، دقت و میزان منابع پردازشی طراحی شده‌اند. انتخاب مدل مناسب به عواملی مانند نوع کاربرد، سخت‌افزار در دسترس، سرعت مورد نیاز و میزان دقت بستگی دارد.

در پروژه **Object Scope** با توجه به اینکه هدف، اجرای مدل در محیط مرورگر و بدون نیاز به سرورهای پردازشی است، از مدل **COCO-SSD** استفاده شده است. با این حال، برای انتخاب این مدل، بررسی و مقایسه آن با سایر مدل‌های شناخته‌شده ضروری است.

۸-۱. مدل YOLO (You Only Look Once)

YOLO یکی از مشهورترین مدل‌های تشخیص اشیا است که با هدف افزایش سرعت پردازش توسعه یافته است. این مدل تصویر را تنها یک بار پردازش کرده و به همین دلیل سرعت بسیار بالایی دارد.

مزایا

- دقت بالا
- سرعت بسیار مناسب
- مناسب برای سیستم‌های Real-Time
- پشتیبانی از نسخه‌های متعدد مانند YOLOv8، YOLOv5 و YOLOv11

معایب

- نیاز به سخت‌افزار قوی‌تر
- اجرای دشوارتر در مرورگر
- مصرف حافظه بیشتر نسبت به COCO-SSD

۸-۲. مدل Faster R-CNN

Faster R-CNN یکی از دقیق‌ترین مدل‌های تشخیص اشیا محسوب می‌شود و در بسیاری از تحقیقات علمی مورد استفاده قرار می‌گیرد.

مزایا

- دقت بسیار بالا
- عملکرد مناسب در تصاویر پیچیده
- تشخیص بهتر اشیای کوچک

معایب

- سرعت پایین تر
- نیازمند GPU
- مناسب نبودن برای اجرای مستقیم در مرورگر

۸-۳. مدل EfficientDet

EfficientDet یکی از مدل‌های جدید تشخیص اشیا است که تلاش می‌کند تعادل مناسبی میان سرعت و دقت ایجاد کند.

مزایا

- دقت بالا
- مصرف منابع کمتر نسبت به Faster R-CNN
- مقیاس‌پذیری مناسب

معایب

- پیچیدگی بیشتر
- اجرای دشوارتر در پروژه‌های سبک وب

۸-۴. مدل COCO-SSD

COCO-SSD مدلی سبک و بهینه‌شده برای اجرای سریع است که از معماری SSD و مجموعه داده COCO استفاده می‌کند.

مزایا

- حجم پایین مدل
- سرعت بالا
- اجرای مستقیم در مرورگر
- سازگاری کامل با TensorFlow.js
- مناسب برای پروژه‌های آموزشی و نمونه‌های اولیه

معایب

- محدود بودن به کلاس‌های Dataset COCO
- دقت کمتر نسبت به مدل‌های سنگین‌تر
- کاهش عملکرد در شرایط پیچیده

۸-۵. مقایسه نهایی مدل‌ها

مدل	سرعت	دقت	اجرا در مرورگر	مناسب برای پروژه
COCO-SSD	بسیار بالا	متوسط	بله	بسیار مناسب
YOLO	بالا	بالا	محدود	مناسب
Faster R-CNN	پایین	بسیار بالا	خیر	خیر
EfficientDet	بالا	بالا	محدود	نسبتاً مناسب

۸-۶. دلیل انتخاب COCO-SSD در پروژه Object Scope

هدف اصلی این پروژه، توسعه یک سامانه تشخیص اشیا کاملاً مبتنی بر وب بوده است. بنابراین معیارهای زیر در انتخاب مدل اهمیت زیادی داشتند:

- امکان اجرا در مرورگر
- عدم نیاز به سرور
- سرعت مناسب در تشخیص لحظه‌ای
- حجم پایین مدل
- سازگاری کامل با TensorFlow.js
- نصب و راه‌اندازی ساده

با توجه به این معیارها، مدل **COCO-SSD** بهترین گزینه برای پروژه حاضر محسوب می‌شود. اگرچه مدل‌هایی مانند YOLO یا Faster R-CNN دقت بیشتری دارند، اما نیاز آن‌ها به منابع پردازشی بالاتر باعث می‌شود برای اجرای مستقیم در مرورگر گزینه مناسبی نباشند.

۹. معماری و طراحی سیستم

معماری پروژه **Object Scope** به گونه‌ای طراحی شده است که تمامی عملیات پردازش تصویر و تشخیص اشیا در سمت کاربر (Client-Side) انجام شود. این معماری باعث می‌شود نیازی به ارسال تصاویر به سرور وجود نداشته باشد و تمامی پردازش‌ها مستقیماً در مرورگر کاربر انجام شوند.

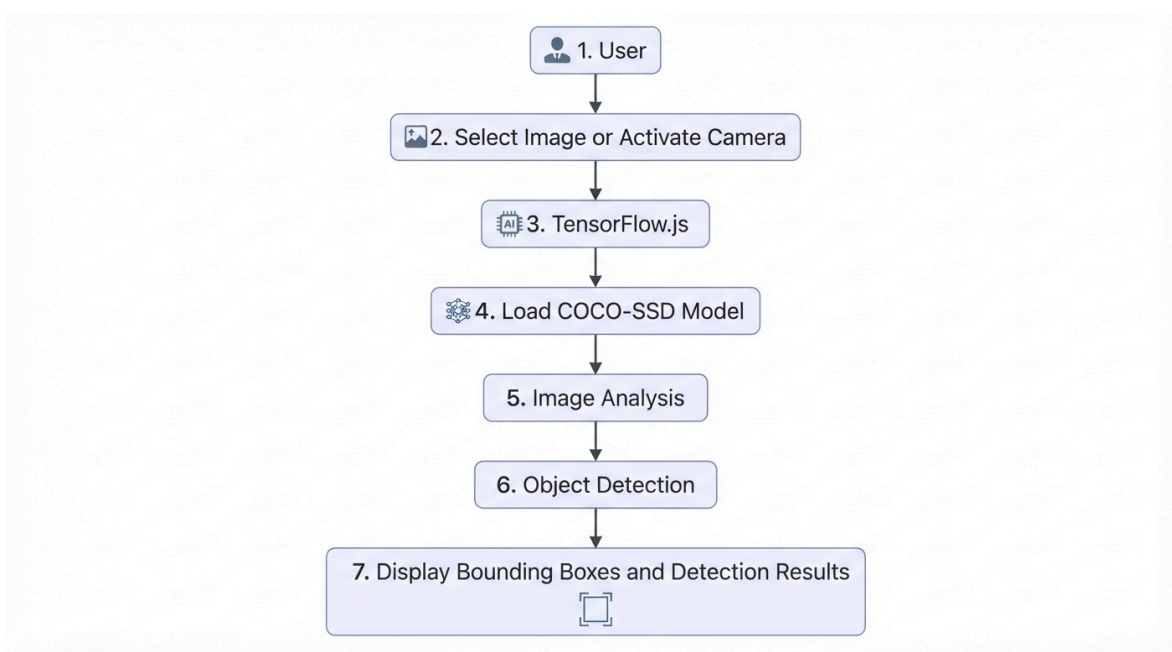
این روش علاوه بر افزایش سرعت پاسخ‌گویی، باعث حفظ حریم خصوصی کاربران نیز می‌شود؛ زیرا تصاویر از دستگاه کاربر خارج نمی‌شوند.

۹-۱. ساختار کلی سیستم

معماری کلی پروژه از بخش‌های زیر تشکیل شده است:

- رابط کاربری (User Interface)
- دریافت تصویر یا ویدئوی زنده
- بارگذاری مدل COCO-SSD
- پردازش تصویر توسط TensorFlow.js
- نمایش نتایج تشخیص

جریان کلی عملکرد سیستم به صورت زیر است:



۹-۲. بخش تشخیص تصویر (Image Detection)

در این بخش، کاربر یک تصویر را از دستگاه خود انتخاب می‌کند. پس از بارگذاری تصویر، مدل COCO-SSD عملیات تشخیص را انجام داده و اطلاعات زیر نمایش داده می‌شود:

- نام شیء
- میزان اطمینان (Confidence Score)
- کادر اطراف شیء (Bounding Box)

۹-۳. بخش تشخیص زنده (Real-Time Detection)

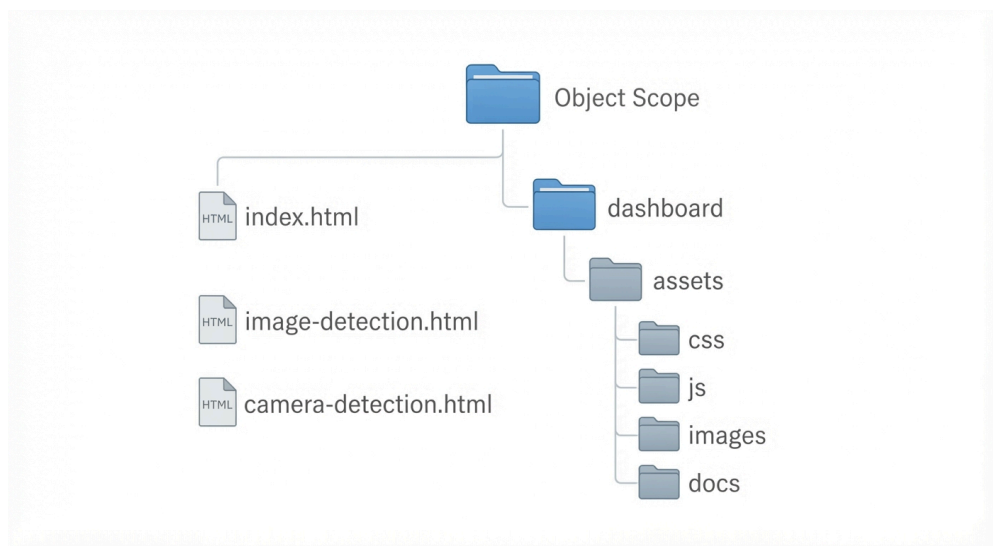
در حالت تشخیص زنده، تصاویر به صورت پیوسته از دوربین دریافت می‌شوند. هر فریم توسط مدل تحلیل شده و اشیای شناسایی شده به همراه کادرهای مشخص‌کننده روی تصویر نمایش داده می‌شوند.

این قابلیت مهم‌ترین بخش پروژه محسوب می‌شود و توانایی اجرای هوش مصنوعی به صورت لحظه‌ای در محیط مرورگر را نشان می‌دهد.

۹-۴. ساختار فایل‌های پروژه

پروژه Object Scope از ساختار پوشه‌ای منظم برای جداسازی بخش‌های مختلف استفاده می‌کند.

نمونه ساختار پروژه:



این ساختار باعث افزایش خوانایی پروژه، سهولت نگهداری و توسعه آسان‌تر آن در آینده می‌شود.

۵-۹. مزایای معماری انتخاب شده

معماری پروژه مزایای متعددی دارد، از جمله:

- اجرای کامل در مرورگر
- عدم نیاز به سرور برای پردازش
- حفظ حریم خصوصی کاربران
- سرعت مناسب در تشخیص اشیا
- قابلیت اجرا روی سیستم‌عامل‌های مختلف
- امکان استقرار آسان روی سرویس‌هایی مانند **Firebase Hosting** یا سایر سرویس‌های میزبانی فایل‌های استاتیک

جمع‌بندی

معماری پروژه **Object Scope** بر پایه پردازش سمت کاربر طراحی شده است و با بهره‌گیری از **TensorFlow.js** و **COCO-SSD**، امکان تشخیص اشیا از تصاویر و ویدئوهای زنده را بدون نیاز به پردازش سمت سرور فراهم می‌کند. این معماری علاوه بر سادگی و سرعت، قابلیت توسعه در آینده و استفاده از مدل‌های پیشرفته‌تر را نیز دارد.

۱۰. مراحل پیاده‌سازی پروژه

پس از بررسی مبانی نظری، انتخاب فناوری‌های مناسب و مطالعه مدل‌های تشخیص اشیا، مرحله پیاده‌سازی پروژه **Object Scope** آغاز شد. هدف اصلی این مرحله، توسعه یک سامانه تحت وب بود که بتواند بدون نیاز به نصب نرم‌افزارهای جانبی، عملیات تشخیص اشیا را مستقیماً در مرورگر کاربر انجام دهد.

در این پروژه، از زبان‌های **HTML5**، **CSS3** و **JavaScript** برای طراحی رابط کاربری و پیاده‌سازی منطق برنامه استفاده شده است. همچنین کتابخانه **TensorFlow.js** به عنوان بستر اجرای مدل یادگیری ماشین و مدل **COCO-SSD** به عنوان موتور اصلی تشخیص اشیا انتخاب شد.

فرآیند توسعه پروژه به صورت مرحله‌ای انجام شد. ابتدا ساختار صفحات وب طراحی گردید، سپس رابط کاربری برای تعامل کاربران ایجاد شد. در ادامه، مدل هوش مصنوعی بارگذاری شده و قابلیت تشخیص تصویر و

تشخیص زنده از طریق دوربین به سامانه اضافه شد. در نهایت، نتایج تشخیص به همراه کادرهای مشخص‌کننده (Bounding Box) و میزان اطمینان مدل در اختیار کاربر قرار گرفت.

سامانه نهایی شامل دو بخش اصلی است که در ادامه معرفی می‌شوند.

۱-۱۰. بخش تشخیص تصویر (Image Detection)

بخش تشخیص تصویر به کاربران این امکان را می‌دهد که یک تصویر را از حافظه دستگاه خود انتخاب کرده و پس از بارگذاری، اشیای موجود در تصویر را شناسایی کنند.

در این بخش، پس از انتخاب تصویر، فایل توسط مرورگر دریافت شده و برای پردازش به مدل COCO-SSD ارسال می‌شود. مدل با تحلیل تصویر، تمامی اشیای قابل تشخیص را شناسایی کرده و اطلاعات مربوط به هر شیء را استخراج می‌کند.

نتایج نمایش داده‌شده شامل موارد زیر هستند:

- نام شیء شناسایی‌شده
- درصد اطمینان مدل (Confidence Score)
- محل قرارگیری شیء با استفاده از Bounding Box

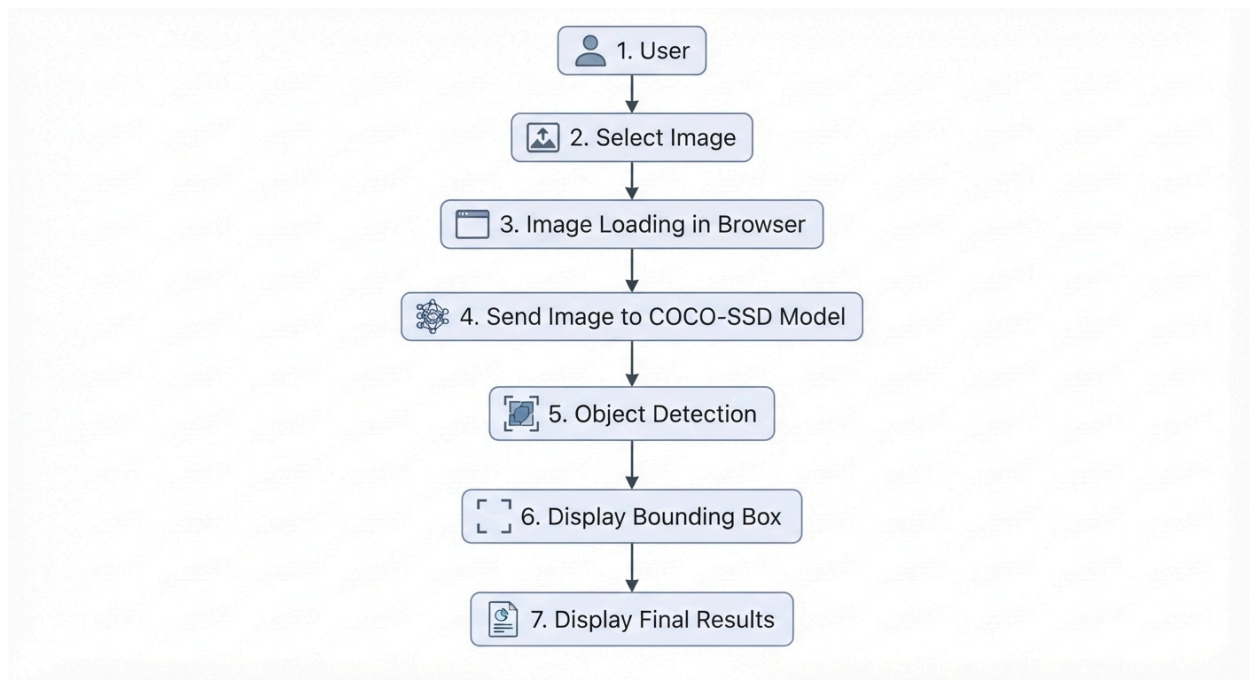
برای نمایش کادرهای تشخیص، از عنصر **Canvas** استفاده شده است. مختصات ارائه‌شده توسط مدل روی Canvas رسم شده و هم‌زمان تصویر اصلی به کاربر نمایش داده می‌شود. این روش باعث می‌شود کاربر علاوه بر مشاهده نتایج متنی، محل دقیق هر شیء را نیز در تصویر مشاهده کند.

یکی از قابلیت‌های این بخش، نمایش وضعیت بارگذاری مدل هوش مصنوعی است. هنگام ورود به صفحه، سامانه وضعیت بارگذاری مدل را بررسی کرده و پس از آماده شدن کامل مدل، امکان انجام عملیات تشخیص را فراهم می‌کند. این ویژگی از بروز خطاهای ناشی از اجرای مدل قبل از آماده‌سازی جلوگیری می‌کند.

همچنین در صورتی که تصویر فاقد اشیای قابل تشخیص باشد یا مدل نتواند شیئی را شناسایی کند، پیام مناسب به کاربر نمایش داده می‌شود.

مراحل اجرای بخش تشخیص تصویر

فرآیند کلی این بخش به صورت زیر انجام می‌شود:



۲-۱۰. بخش تشخیص دوربین (Camera Detection)

بخش تشخیص زنده، مهم‌ترین قسمت پروژه **Object Scope** محسوب می‌شود. در این بخش، کاربر با فعال کردن دوربین دستگاه، امکان مشاهده نتایج تشخیص اشیاء به صورت لحظه‌ای را خواهد داشت.

برای دسترسی به دوربین، از رابط برنامه‌نویسی **MediaDevices API** استفاده شده است. پس از دریافت مجوز از کاربر، جریان ویدئویی دوربین در عنصر **Video** نمایش داده می‌شود.

مدل **COCO-SSD** به صورت مداوم فریم‌های دریافتی را پردازش کرده و در هر لحظه اشیاء موجود را شناسایی می‌کند. سپس اطلاعات تشخیص شامل نام شیء، کادر تشخیص و میزان اطمینان روی تصویر نمایش داده می‌شود.

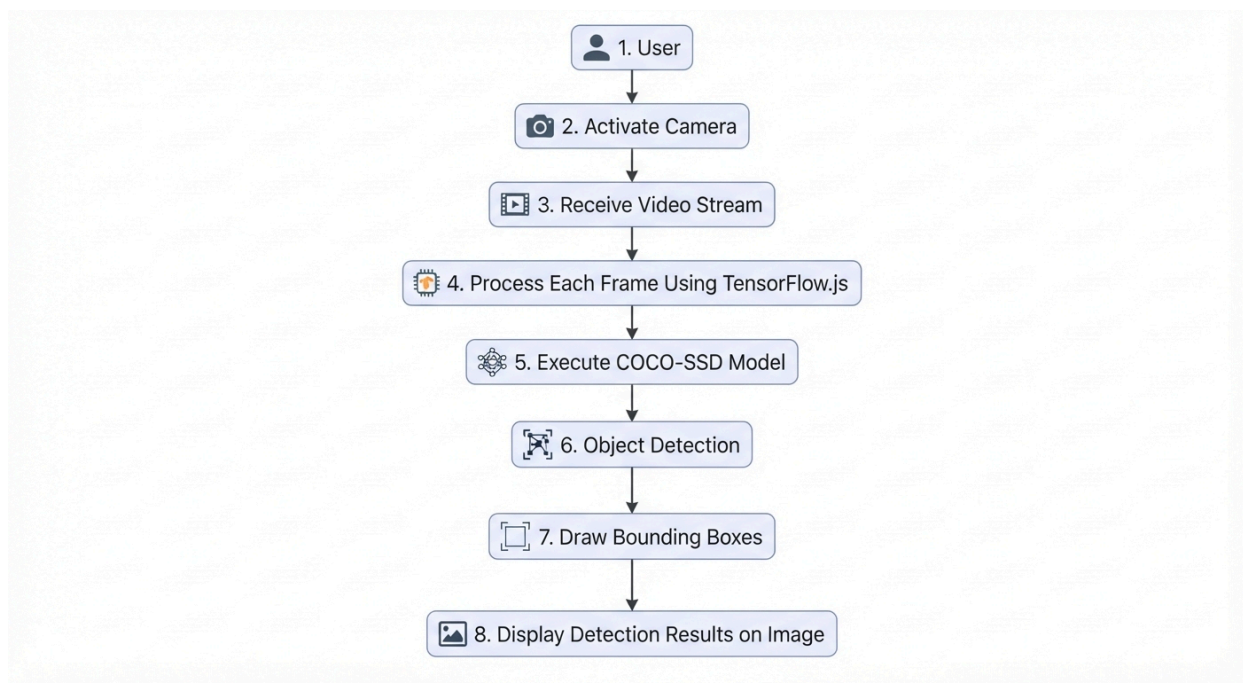
به دلیل اینکه عملیات تشخیص برای هر فریم به صورت پیوسته انجام می‌شود، سرعت اجرای مدل اهمیت زیادی دارد. در این پروژه از نسخه **Lite MobileNet V2** استفاده شده است تا پردازش‌ها با سرعت مناسب و بدون فشار بیش از حد به سخت‌افزار کاربر انجام شوند.

در این بخش، دو دکمه برای کنترل عملکرد سیستم در نظر گرفته شده است:

- **Start Camera** برای فعال‌سازی دوربین
- **Stop Camera** برای توقف جریان ویدئویی و آزادسازی منابع سیستم

با متوقف شدن دوربین، پردازش مدل نیز متوقف شده و منابع سخت‌افزاری آزاد می‌شوند.

مراحل اجرای بخش تشخیص زنده



قابلیت‌های پیاده‌سازی شده در پروژه

در فرآیند توسعه **Object Scope** قابلیت‌های زیر پیاده‌سازی شده‌اند:

- طراحی رابط کاربری واکنش‌گرا (Responsive)
- بارگذاری خودکار مدل COCO-SSD
- نمایش وضعیت بارگذاری مدل
- تشخیص اشیاء از تصاویر بارگذاری شده
- تشخیص لحظه‌ای از طریق دوربین
- نمایش Bounding Box روی Canvas
- نمایش نام اشیاء و میزان اطمینان مدل

- کنترل شروع و توقف دوربین
- مدیریت خطاهای مربوط به بارگذاری مدل و دسترسی به دوربین
- طراحی ساختار منظم فایل‌ها و پوشه‌های پروژه
- آماده‌سازی پروژه برای استقرار روی سرویس‌های میزبانی مانند Firebase Hosting

جمع‌بندی

در این فصل، مراحل عملی توسعه پروژه **Object Scope** تشریح شد. سامانه طراحی‌شده قادر است تصاویر ثابت و جریان زنده دوربین را با استفاده از **TensorFlow.js** و **COCO-SSD** تحلیل کرده و نتایج تشخیص اشیا را مستقیماً در مرورگر نمایش دهد. پیاده‌سازی کامل پروژه در سمت کاربر، علاوه بر کاهش وابستگی به سرور، باعث افزایش سرعت اجرا و حفظ حریم خصوصی کاربران شده است.

۱۱. تست و ارزیابی عملکرد سیستم

پس از تکمیل مراحل توسعه، پروژه **Object Scope** در شرایط مختلف مورد آزمایش قرار گرفت تا عملکرد سیستم از نظر دقت، سرعت و پایداری بررسی شود. آزمایش‌ها بر روی تصاویر ثابت و همچنین جریان زنده دوربین انجام شدند و نتایج نشان دادند که مدل **COCO-SSD** قادر است بسیاری از اشیای رایج را با سرعت مناسب و دقت قابل قبول شناسایی کند.

در بخش تشخیص تصویر، تصاویر شامل اشیای مختلف مانند انسان، خودرو، حیوانات، صندلی، تلفن همراه، کتاب، لپ‌تاپ و سایر اشیای موجود در مجموعه داده COCO مورد استفاده قرار گرفتند. مدل توانست در اکثر موارد اشیا را به درستی شناسایی کرده و همراه با درصد اطمینان نمایش دهد.

در بخش تشخیص زنده نیز عملکرد سیستم در شرایط مختلف مانند نور مناسب، نور کم، فاصله‌های متفاوت و زوایای مختلف بررسی شد. نتایج نشان داد که سرعت اجرای مدل برای استفاده آموزشی و نمونه‌های اولیه مناسب است و پردازش تصاویر به صورت لحظه‌ای در مرورگر امکان‌پذیر است.

همچنین عملکرد سیستم روی مرورگرهای مختلف مورد بررسی قرار گرفت و مشخص شد که مرورگرهای مبتنی بر Chromium عملکرد بهتری نسبت به برخی مرورگرهای دیگر در اجرای TensorFlow.js دارند.

به طور کلی، ارزیابی انجام‌شده نشان داد که پروژه **Object Scope** توانسته است اهداف تعیین‌شده را محقق کند و یک نمونه عملی از اجرای فناوری تشخیص اشیا در محیط وب را ارائه دهد.

۱۲. چالش‌ها و راهکارهای حل مشکلات

در طول توسعه پروژه، چالش‌های مختلفی در بخش‌های فنی و پیاده‌سازی وجود داشت که برای هر یک راهکار مناسبی در نظر گرفته شد.

یکی از مهم‌ترین چالش‌ها، اجرای مدل هوش مصنوعی در محیط مرورگر بود. از آنجا که تمامی پردازش‌ها در سمت کاربر انجام می‌شوند، عملکرد سیستم به توان سخت‌افزاری دستگاه وابسته است. برای کاهش این مشکل، از نسخه **Lite MobileNet V2** مدل COCO-SSD استفاده شد که حجم کمتر و سرعت بیشتری نسبت به نسخه‌های سنگین‌تر دارد.

چالش دیگر، مدیریت بارگذاری مدل بود. در صورتی که کاربر پیش از آماده شدن کامل مدل عملیات تشخیص را آغاز می‌کرد، امکان بروز خطا وجود داشت. برای رفع این مشکل، وضعیت بارگذاری مدل در رابط کاربری نمایش داده شد و اجرای عملیات تنها پس از آماده شدن کامل مدل امکان‌پذیر شد.

مدیریت دسترسی به دوربین نیز یکی دیگر از مشکلات پروژه بود. با استفاده از MediaDevices API و مدیریت مناسب خطاها، امکان کنترل صحیح فعال‌سازی و غیرفعال‌سازی دوربین فراهم شد.

همچنین محدودیت‌های مدل COCO-SSD از جمله تعداد محدود کلاس‌های قابل تشخیص و کاهش دقت در برخی شرایط، از جمله چالش‌های ذاتی پروژه بودند که در تحلیل نتایج مورد توجه قرار گرفتند.

۱۳. بهبودها و توسعه‌های آینده

پروژه Object Scope قابلیت توسعه در بخش‌های مختلف را دارد و می‌توان امکانات جدیدی به آن اضافه کرد.

یکی از مهم‌ترین مسیرهای توسعه، استفاده از مدل‌های پیشرفته‌تر مانند YOLO یا EfficientDet است که دقت بالاتری در تشخیص اشیا دارند. همچنین می‌توان با آموزش مدل بر روی مجموعه داده اختصاصی، امکان شناسایی اشیای سفارشی را فراهم کرد.

افزودن قابلیت ذخیره تصاویر، ثبت تاریخچه نتایج، پشتیبانی از فایل‌های ویدئویی، توسعه نسخه چندزبانه، افزایش تعداد قابلیت‌های رابط کاربری و اتصال به سرویس‌های ابری برای اجرای مدل‌های بزرگ‌تر از دیگر مسیرهای توسعه این پروژه محسوب می‌شوند.

همچنین امکان استقرار پروژه بر روی سرویس‌هایی مانند Firebase Hosting و توسعه نسخه موبایل می‌تواند دسترسی کاربران به سامانه را افزایش دهد.

با توجه به ساختار ماژولار پروژه، اضافه شدن قابلیت‌های جدید بدون ایجاد تغییرات اساسی در معماری سیستم امکان‌پذیر خواهد بود.

۱۴. نتیجه‌گیری

پروژه **Object Scope** با هدف طراحی و پیاده‌سازی یک سامانه تشخیص اشیا مبتنی بر فناوری‌های نوین هوش مصنوعی و تحت وب توسعه داده شد. در این پروژه تلاش شد تا با بهره‌گیری از **TensorFlow.js** و مدل **COCO-SSD**، سیستمی ایجاد شود که بتواند بدون نیاز به نصب نرم‌افزارهای جانبی یا استفاده از سرورهای پردازشی، عملیات تشخیص اشیا را مستقیماً در مرورگر کاربر انجام دهد. این رویکرد علاوه بر افزایش سرعت پردازش، موجب حفظ حریم خصوصی کاربران نیز می‌شود؛ زیرا تمامی عملیات تحلیل تصاویر در سمت کاربر (Client-Side) انجام می‌گیرد.

در ابتدای این گزارش، مفاهیم پایه‌ای مرتبط با هوش مصنوعی، یادگیری ماشین، یادگیری عمیق و بینایی ماشین مورد بررسی قرار گرفت تا چارچوب نظری پروژه مشخص شود. سپس فناوری‌های مورد استفاده شامل **TensorFlow.js**، مدل **COCO-SSD** و فناوری‌های استاندارد توسعه وب معرفی شدند و نحوه عملکرد هر یک در ساختار پروژه توضیح داده شد. همچنین مجموعه داده **COCO**، ویژگی‌ها و محدودیت‌های آن بررسی شد تا مشخص شود مدل مورد استفاده بر چه مبنایی آموزش دیده و چه توانایی‌ها و محدودیت‌هایی دارد.

در ادامه، مدل **COCO-SSD** با سایر مدل‌های مطرح حوزه تشخیص اشیا مانند **Faster R-CNN**، **YOLO** و **EfficientDet** مقایسه شد. این مقایسه نشان داد که اگرچه برخی از این مدل‌ها دقت بالاتری دارند، اما نیازمند منابع پردازشی بیشتری هستند و اجرای آن‌ها در محیط مرورگر با محدودیت‌هایی همراه است. از این رو، با توجه به اهداف پروژه و نیاز به اجرای مستقیم در مرورگر، **COCO-SSD** به عنوان مناسب‌ترین گزینه انتخاب شد.

در بخش پیاده‌سازی، معماری سیستم و مراحل توسعه پروژه به صورت کامل تشریح شد. ساختار پروژه به گونه‌ای طراحی گردید که علاوه بر سادگی، قابلیت توسعه و نگهداری مناسبی داشته باشد. دو بخش اصلی سامانه شامل **تشخیص اشیا از تصویر و تشخیص لحظه‌ای از طریق دوربین** پیاده‌سازی شدند و کاربران قادر هستند تنها با استفاده از مرورگر، تصاویر یا جریان زنده دوربین را تحلیل کرده و نتایج تشخیص را به همراه کادرهای مشخص‌کننده (Bounding Box) و میزان اطمینان مدل مشاهده کنند.

در مرحله ارزیابی، عملکرد سیستم در شرایط مختلف مورد آزمایش قرار گرفت. نتایج نشان داد که سامانه در تشخیص بسیاری از اشیای متداول عملکرد قابل قبولی دارد و از سرعت مناسبی برای کاربردهای آموزشی، پژوهشی و نمونه‌های اولیه برخوردار است. همچنین مشخص شد که کیفیت تصاویر، نور محیط، توان سخت‌افزاری دستگاه و محدودیت‌های ذاتی مدل می‌توانند بر دقت تشخیص تأثیرگذار باشند. با وجود این موارد، عملکرد کلی سامانه رضایت‌بخش بوده و اهداف اصلی پروژه با موفقیت محقق شده است.

یکی از مهم‌ترین دستاوردهای این پروژه، اثبات امکان اجرای مدل‌های یادگیری عمیق در محیط مرورگر بدون وابستگی به پردازش سمت سرور است. این موضوع نشان می‌دهد که با پیشرفت فناوری‌های وب، توسعه سامانه‌های هوشمند و مبتنی بر هوش مصنوعی بیش از گذشته در دسترس توسعه‌دهندگان قرار گرفته است. علاوه بر این، ساختار ماژولار پروژه امکان جایگزینی مدل‌های پیشرفته‌تر، افزودن قابلیت‌های جدید و توسعه سامانه برای کاربردهای تخصصی را در آینده فراهم می‌کند.

در مجموع، پروژه **Object Scope** نمونه‌ای عملی از به‌کارگیری فناوری‌های هوش مصنوعی، بینایی ماشین و توسعه وب در قالب یک سامانه یکپارچه است. این پروژه نه تنها اهداف آموزشی و پژوهشی تعیین‌شده را محقق کرده، بلکه می‌تواند به عنوان پایه‌ای برای توسعه سامانه‌های پیشرفته‌تر در زمینه‌هایی مانند سیستم‌های نظارتی، رباتیک، خودروهای هوشمند، پزشکی، صنایع تولیدی، کشاورزی هوشمند و سایر کاربردهای مرتبط با بینایی ماشین مورد استفاده قرار گیرد. با توجه به قابلیت توسعه، سادگی استفاده و استقلال از زیرساخت‌های پردازشی پیچیده، Object Scope می‌تواند نقطه شروع مناسبی برای پژوهش‌ها و پروژه‌های آینده در حوزه تشخیص اشیا و هوش مصنوعی مبتنی بر وب باشد.

منابع

منابع علمی:

- [1] Google. **TensorFlow.js Documentation**. Available at: <https://www.tensorflow.org/js>
- [2] TensorFlow Team. **COCO-SSD Model Documentation**. Available at: <https://github.com/tensorflow/tfjs-models/tree/master/coco-ssd>
- [3] Microsoft. **COCO – Common Objects in Context Dataset**. Available at: <https://cocodataset.org>
- [4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). **You Only Look Once: Unified, Real-Time Object Detection**. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [5] Ren, S., He, K., Girshick, R., & Sun, J. (2015). **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks**. Advances in Neural Information Processing Systems (NeurIPS).
- [6] Tan, M., Pang, R., & Le, Q. V. (2020). **EfficientDet: Scalable and Efficient Object Detection**. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
- [7] Goodfellow, I., Bengio, Y., & Courville, A. (2016). **Deep Learning**. MIT Press.
- [8] Szeliski, R. (2022). **Computer Vision: Algorithms and Applications (Second Edition)**. Springer.
- [9] Russell, S., & Norvig, P. (2021). **Artificial Intelligence: A Modern Approach (Fourth Edition)**. Pearson.
- [10] Mozilla Developer Network (MDN). **HTML, CSS and JavaScript Documentation**. Available at: <https://developer.mozilla.org>

منابع نرم‌افزاری و فناوری‌های مورد استفاده:

در فرآیند توسعه پروژه **Object Scope** از مجموعه‌ای از نرم‌افزارها، کتابخانه‌ها و فناوری‌های مدرن استفاده شده است. محیط توسعه اصلی پروژه **Visual Studio Code** بوده است که برای نوشتن، مدیریت و سازمان‌دهی فایل‌های پروژه مورد استفاده قرار گرفته است.

ساختار صفحات وب پروژه با استفاده از **HTML5** پیاده‌سازی شده و برای طراحی رابط کاربری، ایجاد ظاهر مناسب، مدیریت بخش‌های بصری و واکنش‌گرایی صفحات از **CSS3** استفاده شده است. همچنین زبان برنامه‌نویسی **(JavaScript (ES6** به عنوان بخش اصلی منطق برنامه، وظیفه مدیریت تعاملات کاربر، پردازش داده‌ها، ارتباط با مدل هوش مصنوعی و اجرای قابلیت‌های تشخیص اشیا را بر عهده داشته است.

در بخش هوش مصنوعی، از کتابخانه **TensorFlow.js نسخه 4.17.0** برای اجرای مدل‌های یادگیری ماشین در محیط مرورگر استفاده شده است. مدل اصلی تشخیص اشیا در این پروژه **COCO-SSD نسخه 2.2.2** بوده که با استفاده از معماری **Lite MobileNet V2** امکان تشخیص سریع و سبک اشیا را در محیط وب فراهم می‌کند.

برای توسعه و آزمایش عملکرد پروژه، مرورگر **Google Chrome** به عنوان محیط اصلی اجرای برنامه مورد استفاده قرار گرفته است، زیرا سازگاری مناسبی با فناوری‌های مورد استفاده مانند **TensorFlow.js** و **WebGL** دارد.

همچنین برای مدیریت نسخه‌های پروژه و نگهداری کدها از **Git** استفاده شده و مخزن پروژه جهت ذخیره‌سازی، مدیریت تغییرات و اشتراک‌گذاری کدها بر روی **GitHub** قرار گرفته است.

به طور کلی، ترکیب این فناوری‌ها امکان توسعه یک سامانه تشخیص اشیا تحت وب را فراهم کرده است که بدون نیاز به نصب نرم‌افزارهای جانبی و تنها با استفاده از مرورگر قادر به پردازش تصاویر و ویدئوهای زنده است.