

دانشكده مهندسي برق

طراحی و ساخت سامانه رهگیری اشیای متحرک با چند دوربین

پايان‌نامه براي دريافت درجه کارشناسی

در رشته مهندسي برق گرايش کنترل

محمد عظیمی – علی قره‌خانی

استاد راهنما:

دكتر عباداللهی

شهریور ماه 1401





دانشكده مهندسي برق

طراحی و ساخت سامانه رهگیری اشیای متحرک با چند دوربین

پايان‌نامه براي دريافت درجه کارشناسی

در رشته مهندسي برق گرايش کنترل

محمد عظیمی – علی قره‌خانی

استاد راهنما:

دكتر عباداللهی

شهریور ماه 1401

تشكر و قدرداني:

در ابتدا لازم است تا از زحمات کسانی که من را در پیشبرد این پروژه یاری نمودند، به خصوص دکتر عبادالهی که نقش به‌سزایی در هرجه باکیفیت‌تر شدن پروژه ایفا نمودند، تشکر و قدردانی نموده، و از خداوند متعال برای ایشان، موفقیت و سربلندی آرزومندیم.

چکيده:

امروزه استفاده از سامانه های هوشمند رواج گسترده ای در دنیا پیدا کرده است. به خصوص بعد از معرفی هوش مصنوعی و پردازش تصویر که جزء جدا ناپذیر هوش مصنوعی می باشد تمامی افراد و شرکت‌ها در تلاش هستند تا سامانه‌های خود را مجهز به این دو ابزار قدرتمند بکنند. یکی از موضوعات در حال رشد امروزه رهگیری یک هدف مشخص (tracking) از جمله اشیا و یا حتی انسان ها برای اهداف خاصی می باشد. برای این کار می توان از ارسال و دریافت سیگنال های مختلف استفاده کرد ولی همواره این روش با مشکلات خاص خود روبه‌رو بوده است، از جمله این مشکلات تداخل سیگنال می باشد. برای حل این مشکلات امروزه به کمک پردازش تصویر و هوش مصنوعی وظیفه رهگیری اجسام انجام می گیرد. از جمله کاربرد های این کار می توان به مدیریت انبار داری در کارخانه ها، مدیریت منابع انسانی در محیط کار و یا حتی کاربرد های ورزشی از جمله دنبال کردن ورزشکار ها در حین مسابقه اشاره کرد. در این پروژه به کمک هوش مصنوعی (بخش نرم افزار) و یک ساختار دینامیکی- رباتیکی و یک حلقه کنترلی (بخش سخت افزار) یک ردیاب (tracker) پیاده‌سازی شده‌است که می تواند یک انسان را رهگیری کند.

واژه‌هاي كليدي:

هوش مصنوعی – پردازش تصویر – رهگیری – حلقه کنترلی – رباتیک – هوشمند‌سازی

فهرست مطالب

[فصل 1: مقدمه 1](#_Toc113107347)

[فصل 2: الگوریتم‌ها رهگیری و ردیابی 5](#_Toc113107348)

[2-1- مقدمه 6](#_Toc113107349)

[2-2- الگوریتم های رهگیری هدف 9](#_Toc113107350)

[2-2-1- نحوه بیان و مدل‌سازی یک شئ و یا هدف 9](#_Toc113107351)

[2-2-2- انتخاب ویژگی‌های ساختاری و ظاهری 13](#_Toc113107352)

[2-2-3- الگوریتم های تشخیص22 هدف 18](#_Toc113107353)

[2-2-4- الگوریتم‌های ردیابی 27](#_Toc113107354)

[2-3- نتیجه‌گیری 36](#_Toc113107355)

[فصل 3: هوش مصنوعی و بینایی ماشین 38](#_Toc113107356)

[3-1- چکیده 39](#_Toc113107357)

[3-2- مقدمه 39](#_Toc113107358)

[3-3- تاریخچه 40](#_Toc113107359)

[3-4- ساختار 43](#_Toc113107360)

[3-5- آموزش شبکه 48](#_Toc113107361)

[3-5-2- ابزار های پردازش تصویر 49](#_Toc113107362)

[3-6- نتیجه‌گیری 53](#_Toc113107363)

[فصل 4: سخت‌افزار 55](#_Toc113107364)

[4-1- برد های پردازشی 56](#_Toc113107365)

[4-1-1- آردوینو 56](#_Toc113107366)

[4-1-2- رزبری‌پای 57](#_Toc113107367)

[4-1-3- جتسون نانو 59](#_Toc113107368)

[4-2- عملگرها5: 60](#_Toc113107369)

[4-3- سنسور 62](#_Toc113107370)

[4-4- ساختار دینامیکی 63](#_Toc113107371)

[فصل 5: پیاده‌سازی و نتایج 66](#_Toc113107372)

[5-1- بخش سخت افزار 67](#_Toc113107373)

[5-1-1- پلنت(plant) 67](#_Toc113107374)

[5-1-2- سنسور 68](#_Toc113107375)

[5-1-3- عملگر (actuator) 68](#_Toc113107376)

[5-2- بخش نرم‌افزار 71](#_Toc113107377)

[5-2-1- نویز، اعوجاج و اغتشاش سیستم 71](#_Toc113107378)

[5-2-2- لپتاپ اول؛ خوانش تصویر و انتخاب سوژه 72](#_Toc113107379)

[5-2-3- لپتاپ دوم؛ دریافت اطلاعات شی انتخاب شده، انتخاب و دنبال کردن آن 72](#_Toc113107380)

[5-2-4- نحوه ارتباط لپتاپ اول و دوم 73](#_Toc113107381)

[5-2-5- لپتاپ سوم؛ دریافت خروجی سیستم و تولید سیگنال خطا و فرمان کنترلی 74](#_Toc113107382)

[5-2-6- نحوه ارتباط لپتاپ دوم و سوم 75](#_Toc113107383)

[5-3- شرح یکپارچه پروژه 75](#_Toc113107384)

[5-3-1- ردیاب – نشانه‌های محیطی (LBT) 75](#_Toc113107385)

[5-3-2- ردیاب – ویژگی اشیا (FBT) 77](#_Toc113107386)

[5-3-3- رهگیری شی شناسایی شده بعنوان هدف 78](#_Toc113107387)

[5-3-4- سلام سلام بر تو ای سلام 80](#_Toc113107388)

[5-3-5- فرمان‌دهی عملگرهای سیستم 80](#_Toc113107389)

[فصل 6: نتیجه‌گیری 82](#_Toc113107390)

[مراجع 84](#_Toc113107391)

[پيوست‌ها 1](#_Toc113107392)

فهرست اشکال

فهرست جداول

[جدول(2-1) پارامترهای مداری ماهواره‌ها **Error! Bookmark not defined.**](#_Toc82522355)

[جدول(3-2) فریم UBX **Error! Bookmark not defined.**](#_Toc82522356)

[جدول (3-2) اطلاعات Class های فریم UBX **Error! Bookmark not defined.**](#_Toc82522357)

[جدول(3-3) آیدی های مورد نیاز **Error! Bookmark not defined.**](#_Toc82522358)

1. مقدمه

سیستم‌ها همواره برای کارایی مناسب، نیاز به کنترل‌شدن دارند. از این رو انسان همواره در تلاش بوده است تا سیستم‌های مورد استفاده خود را کنترل کند و تا جایی که می‌تواند سیستم را خودکار کند تا دخالت انسان و احتمال وجود خطاهای انسانی را در سیستم به صفر برساند . کنترل هر سیستمی الزام دارد تا بازخوردی(feedback) از خروجی برگردانده شود. خروجی سیستم را می‌بایست با توجه به شرایط موجود با استفاده از سنسور مناسب خواند. در سیستم‌های ردیاب(tracker)، خروجی سیستم موقعیت و یا سرعت هدف مورد نظر سیستم می‌باشد که البته سرعت نیز پارامتری وابسته به تغییرات موقعیت می‌باشد.

سیستم‌ها از منظری به دو بخش active و passive تقسیم می‌شوند. سیستم active به سیستمی گفته می‌شود که هدف مورد نظر سیستم مجهز به امکاناتی برای عمل کردن در سیستم شده باشد؛ در حالی که سیستم‌ passive سیستمی است که برای اهداف فاقد هرگونه تجهیزات مختص شناسایی شدن طراحی شده است. برای مثال می‌توان گفت سیستم موقعیت‌یاب gps که بر روی تلفن‌های همراه قرار دارد با اتصال به ماهواره‌های اطراف کره زمین یک سیستم active را تشکیل می‌دهند؛ چراکه ما برای انجام عمل موقعیت‌یابی به تجهیزی مانند تلفن همراه (و یا ماژول gps) نیاز داریم. سیستمی که ما طراحی کرده‌ایم مثالی از یک سیستم passive می‌باشد و بدین گونه می‌بایست عمل کند که انسان بدون نیاز به مجهز شدن به افزونه‌ی خاصی بتواند توسط سیستم دنبال(track) شود. قبل از آن که بتوان هدفی را دنبال کرد، می‌بایست آن را ابتدا شناسایی(detect) کرد. هوش مصنوعی ابزاری است که در این بخش می‌تواند بطور قابل ملاحظه‌ای نقش‌آفرینی کند. با توسعه یک شبکه عصبی می‌توان امکان لازم برای تشخیص هدف مورد نظر (که در اینجا، هدف مورد نظر انسان است) را ایجاد کرد.

بطور کلی این سیستم از بخش‌های سخت‌افزاری:

* سازه مکانیکی
* بازوهای مکانیکی
* منبع تغذیه
* میکروکنترلر
* موتورهای DC (سرو موتور)
* دوربین

و از بخش‌های نرم‌افزاری:

* python
* deep learning
* MATLAB/Simulink
* Arduino Programming
* انواع پروتوکول‌های ارتباطی

تشکیل شده ‌است.

سیستم بصورت حلقه بسته طراحی شده و فیدبک آن از نوع فیدبک واحد است. سیستم در هر لحظه با استفاده از شبکه عصبی توسعه داده‌شده قادر است شخص موجود در فریم(frame) دوربین را تشخیص دهد. هدف اصلی سیستم آن است تا قادر باشد در دوربین اول اشیا از قبل تعیین شده(در این پروژه انسان) را تشخیص دهد و سپس کاربر یکی را انتخاب کند. شئ انتخاب شده می‌بایست در دوربین دوم نیز تشخیص داده شده و از اشیای دیگر موجود در تصویر متمایز شود. سپس به کمک بازوهای مکانیکی تعبیه شده برای سیستم، آن را دنبال کند. از این رو اطلاعات خروجی سیستم از طریق فیدبک سیستم بازخورد داده می‌شوند. سپس با تولید سیگنال خطای مناسب و ارسال آن به کنترل‌کننده طراحی‌شده، سیگنال کنترلی مناسب آن توسط کنترل‌کننده تولید می‌شود. کنترل‌کننده مورد استفاده در این سیستم نیز از نوع PID می‌باشد. درنهایت خروجی کنترل‌کننده نیز به عملگرهای سیستم ارسال می‌شوند تا تغییرات بر روی plant اعمال شود. در ادامه، اجزای مختلف مفصلا تشریح خواهند شد.

1. الگوریتم‌های رهگیری و ردیابی
   1. مقدمه

نظارت ویدیویی یکی از موضوعات تحقیقاتی فعال در زمینه بینایی ماشین می‌باشد که سعی در شناسایی ، تشخیص و ردیابی اشیا بر روی یک رشته از تصاویر را دارد و همچنین تلاش دارد تا با جایگزینی روش های قدیمی نظارت که توسط انسان ها انجام می‌شد با دوربین های نظارتی رفتار حرکتی اشیا را درک و بررسی کند . تشخیص اشیا و ردیابی اشیا در بسیاری از کاربردهای بینایی ماشین از جمله نظارت , ناوبری خودرو و هدایت خودکار ربات انجام می‌شود . شناسایی هدف و رهگیری آن شامل پیدا کردن اشیا در قاب توالی فریم های یک ویدئو است . هر روش ردیابی نیازمند یک مکانیزم تشخیص هدف در هر فریم یا زمانی است که هدف ابتدا در ویدئو ظاهر می‌شود . ردیابی هدف , فرآیند پیدا کردن یک شی یا چند شی در طول زمان با استفاده از یک دوربین است . کامپیوترهای با قدرت بالا , قابلیت دسترسی به دوربین‌های ویدئویی با کیفیت بالا و دوربین‌های ویدئویی ارزان و نیاز روزافزون به آنالیز خودکار ویدیو , علاقه مندی های زیادی به الگوریتم ردیابی اشیا ایجاد کرده‌است.

سه مرحله کلیدی در آنالیز ویدیویی وجود دارد , تشخیص اشیا در حال حرکت, رهگیری این اشیا در هر فریم از ویدیو و آنالیز مسیرهای حرکتی شی برای تشخیص رفتار آن‌ها. از این رو استفاده از روش های رهگیری هدف در زمینه های تحقیقاتی مربوط به اشیا متحرک کاربرد دارد . تشخیص خودکار , ردیابی و شمارش تعداد اشیا هدف برای طیف وسیعی از کاربردهای خانگی , کسب‌وکار و کاربردهای صنعتی مثل امنیت , نظارت , مدیریت نقاط دسترسی , برنامه‌ریزی شهری , کنترل ترافیک و غیره بسیار مهم هستند .

با این حال , این کاربردها هنوز نقش مهمی در سطح عملیاتی وسیعی بر عهده ندارند . دلیل اصلی این است که آن‌ها به الزامات قوی برای دستیابی به شرایط کاری رضایت‌بخش , سخت‌افزار تخصصی و گران‌قیمت , تاسیسات پیچیده و رویه‌های راه‌اندازی سخت نیاز دارند . برخی کارها بر روی توسعه روش‌های ردیابی و ردیابی خودکار متمرکز شده‌اند که ضرورت نظارت انسانی را به حداقل می‌رساند . آن‌ها نوعا از یک تابع هدف متحرک استفاده می‌کنند که هر نوع پیکربندی شی فرضی را با کمک مجموعه ای از تشخیص و رهیابی های موجود و از قبل انجام شده و بدون محاسبه مستقیم ارتباط داده‌های آن‌ها ارزیابی می‌کند . بنابراین , یک صرفه‌جویی قابل‌توجه در هزینه محاسباتی انجام می شود . علاوه بر این , تابع هدف به گونه‌ای طراحی شده‌است که برای محیط های دارای اغتشاش و یا حتی در مواردی که شی تشخیص داده نمی شود نیز قابل اعمال کردن باشد . حوزه بینایی ماشین ( رایانه ) با مشکلاتی روبه‌رو می باشد که با محیط اطراف رایانه در ارتباط است. به عنوان مثال یکی از این مشکلات بدین گونه است که سیستم نطارتی وظیفه نظارت بر یک محیط خاص و گزارش اطلاعاتی در مورد فعالیت های مشاهده شده را دارد . در این رابطه , نظارت ویدیویی معمولا ً از حسگرهای نوری ( دوربین‌های ویدیویی ) برای جمع‌آوری اطلاعات از محیط استفاده می‌کند . در یک سیستم مراقبت معمول , این دوربین‌ها در موقعیت‌های ثابت نصب می‌شوند و ویدیو های ضبط شده را به یک مکان خاص که به آن اتاق کنترل می گویند منتقل می‌کنند . سپس ویدیو های ضبط و دریافت‌شده بر روی نمایشگر نشان داده شده و ردیابی توسط اپراتورهای انسانی انجام می‌شوند . با این حال ، اپراتورهای انسانی ممکن است با بسیاری از مشکلات مواجه شوند ، در حالی که آن‌ها این سنسورها را تحت نظر دارند . یک مشکل ناشی از این واقعیت این است که اپراتور باید از طریق دوربین ها مدام اشیا را دنبال کند ، و اگر شی مشخصی بین میدان دید دوربین ها حرکت کند هم زمان با دنبال کردن آن نباید هیچ یک از اشیا دیگر را از دید خود خارج کند. بنابراین ، پایش دوربین ها به چالش بزرگتری تبدیل می‌شود هنگامی که تعداد حسگرهای موجود در چنین شبکه نظارتی افزایش یابد. بنابراین ، سیستم‌های نظارت باید خودکار شوند تا عملکرد را بهبود بخشیده و چنین خطاهایی که ممکن است از اپراتورها سر بزند را حذف کنند .

به طور ایده‌آل ، یک سیستم نظارت خودکار باید تنها برای اهداف یک برنامه مشخص که برای آن تعریف شده است برنامه ریزی شود به گونه ای که که در آن پردازش ها به صورت زمان حقیقی[[1]](#footnote-2) انجام شده و پایداری[[2]](#footnote-3) سیستم بالا باشد. . بنابراین یکی از مهمترین چالش های موجود، ایجاد سیستم‌های نظارت دقیق و بلادرنگ با قیمت مقرون‌به‌صرفه است . با کاهش هزینه‌های سخت‌افزاری برای سنجش و محاسبه ، و افزایش سرعت پردازنده ، سیستم‌های نظارت به صورت تجاری در دسترس قرار گرفته و در حال حاضر به تعدادی از برنامه‌های کاربردی مختلف مانند نظارت ترافیک ، فرودگاه و امنیت بانک‌ها و غیره اعمال می‌شوند . با این حال , الگوریتم ‌های بینایی ماشین ( به خصوص برای یک دوربین ) هنوز هم تحت‌تاثیر بسیاری از نواقص مانند انسداد , سایه‌ها , شرایط آب و هوایی و غیره قرار دارند . اگرچه استفاده از دوربین‌های چندگانه در مقایسه با یک دوربین منجر به کنترل بهتر این مشکلات می‌شود ،متاسفانه سیستم های چند دوربینی هنوز راه‌حل نهایی نیستند .

علاوه بر مشکلات محیطی موجود برخی مشکلات چالش برانگیز در الگوریتم‌های نظارت وجود دارد , مانند مدل‌سازی زمینه تصویر , استخراج ویژگی , رهگیری , جابجایی انسداد و تشخیص رویداد . علاوه بر این , الگوریتم ‌های بینایی ماشین هنوز به اندازه کافی قوی نیستند که سیستم‌های کاملا ً خودکار را اداره کنند و بسیاری از مطالعات تحقیقاتی در مورد این پیشرفت‌ها همچنان در حال انجام هستند . کاری که در این پروژه صورت گرفته است صرفا جزو مراحل اولیه سیستم ها نظارت هوشمند شامل تشخیص اشیا و دنبال کردن مسیر حرکتی آن ها می باشد.

2-2- الگوریتم های رهگیری هدف

1-2-2- نحوه بیان و مدل‌سازی یک شئ و یا هدف

در یک سیستم رهگیری و ردیابی هدف، یک شئ می‌تواند هر چیزی که برای ما اهمیت دارد تعریف شود. به عنوان مثال قایق روی آب، ماشین در جاده و یا آدمی که در حال راه رفتن است می تواند به عنوان یک شئ برای ما تعریف شود. این اهداف و اشیا با توجه به شکل و ظاهری که دارا می‌باشند می‌توانند به روش‌های مختلفی تعریف شده و مدل‌سازی شوند. در ادامه این روش‌ها را بررسی می‌کنیم.

* مدل نقطه‌ای[[3]](#footnote-4): در این روش، شئ یا هدف مورد نظر توسط یک نقطه مرکزی همانند شکل (2-1-الف) و یا توسط تعدادی نقطه پخش شده همانند شکل (2-1-ب) نشان داده شده و بیان می‌شود. در حالت کلی این نوع از بیان شئ زمانی استفاده می شود که هدف مورد نظر بخش کوچکی از تصویر را اشغال کرده باشد.
* مدل شکل و هندسه اصلی[[4]](#footnote-5): در این روش، شئ یا هدف مورد نظر توسط یک مستطیل همانند شکل (2-1-پ) و یا یک بیضی همانند شکل (2-1-ت) نشان داده شده و بیان می شود. در این روش هندسه و شکل کلی هدف مورد نظر داخل مستطیل و یا بیضی قرار دارد. اصطلاحا به مستطیل و بیضی در این روش باندینگ‌باکس[[5]](#footnote-6) گفته می شود.
* مدل سطح داخلی و خارجی[[6]](#footnote-7): در این روش هدف مورد نظر توسط یک منحنی مرزی بیان می‌شود. به این منحنی اصطلاحا کانتور[[7]](#footnote-8)گفته می شود که نمونه‌های آن را در شکل های (2-1-ث) و (2-1-ج) مشاهده می‌کنیم. همچنین نمونه‌ای از سطح داخلی هدف مورد نظر را که توسط این روش به دست می‌آید را در شکل (2-1-چ) مشاهده می‌کنید.
* مدل مفصلی[[8]](#footnote-9) : در این روش هدف مورد نظر که از قسمت های مختلفی تشکیل شده و توسط مفاصلی به هم مرتبط شده اند توسط تعدادی بیضی و یا دایره برای هر قسمت نشان داده می شوند. این بیضی ها و دایره ها نیز همانند مفاصل به هم متصل بوده و با هم زاویه مشخصی را می‌سازند. نمونه این از این روش را در شکل (2-1-ح) مشاهده می‌کنید.
* مدل اسکلتی[[9]](#footnote-10) :در این روش از ساختار اسکلتی هدف مورد نظر برای نشان دادن هدف و بیان آن استفاده می‌شود. برای به دست آوردن ساختار اسکلتی هدف مورد نظر می‌توان از روش های مختلفی مانند تبدیل محور داخلی[[10]](#footnote-11) استفاده کرد. نمونه ای از این روش را در شکل (3-2-خ) مشاهده می‌کنید.

(الف) (ب) (پ) (ت)

(ث) (ج) (چ) (ح) (خ)

شکل((2-1)): روش های مختلف بیان هدف مورد نظر و مدل سازی آن

تا این مرحله روش‌های مدل سازی هدف بر‌اساس شکل هندسی هدف مورد نظر بیان شد. در برخی موارد می توان ویژگی‌های ظاهری را نیز مدل‌سازی کرده و به همراه مدل‌سازی هندسی برای رهگیری و ردیابی استفاده کرد. در ادامه روش های مختلف برای بیان مشخصات ظاهری و هم‌چنین مدل‌سازی مشخصات ظاهری را بیان می‌کنیم.

* مدل چگالی احتمال[[11]](#footnote-12): چگالی احتمال هدف مورد نظر می‌تواند به صورت پارامتری همانند تابع گوسی و یا ترکیبی از توابع گوسی و همچنین به صورت غیر پارامتری همانند نمودار هیستوگرام بیان شود. این توابع که بیان کننده تابع چگالی احتمال ویژگی های ظاهری تصویر می باشند باید بر روی قسمتی از تصویر که در مرحله قبل مدل سازی شده است اعمال شوند. در شکل (2-2) یک نمونه از مدل‌سازی تصویر توسط این روش را مشاهده می‌کنیم.



شکل((2-2)): یک نمونه از مدل‌سازی تصویر توسط مدل چگالی احتمال

همان‌طور که در شکل (2-2)مشاهده می‌کنید، دو تصویر موجود از نظر ظاهری باهم متفاوت می‌باشند و به همین علت نمودار هیستوگرام آن‌ها که یک نمونه از مدل‌سازی چگالی احتمال است نیز متفاوت است. بنابراین از این نوع مدل‌سازی می‌توان به خوبی استفاده کرد تا اجسام و اهداف مختلف را مدل‌سازی کنیم.

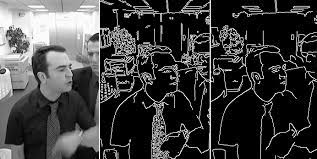
* مدل قالب[[12]](#footnote-13): قالب هدف مورد نظر به کمک یک سری اشکال هندسی ساده تشکیل می شود. مزیت این نوع از مدل سازی این است که هم اطلاعات مکانی و هم اطلاعات ظاهری هدف مورد نظر را در بر می‌گیرد. این نوع از مدل سازی برای اهدافی که جا‌به‌جایی زیادی حین رهگیری و ردیابی ندارند مناسب می باشد، علت این امر نیز این است که مشخصات در این مدل‌سازی صرفا توسط یک چشم‌انداز[[13]](#footnote-14) از هدف رمزگذاری می‌شود.
* مدل ظاهری فعال[[14]](#footnote-15): این روش به صورت پیوسته مدلی از ساختار تصویر و همچنین ظاهر تصویر را مدل‌سازی می‌کند. در حالت کلی ساختار تصویر به کمک یک سری علائم نشانه‌گذاری همانند مدل سطح داخلی که در قسمت قبل بررسی کردیم مدل‌سازی می‌شود. برای هر ساختار مدل‌سازی شده یک آرایه از مشخصات ظاهری هدف مورد نظر همانند رنگ، ابعاد و غیره ذخیره‌سازی می‌شود. این نوع از مدل‌سازی نیاز به آموزش دارد تا بتواند ساختار هدف مورد نظر و همچنین مشخصات ظاهری متنایب با آن را تشخیص دهد.
* مدل ظاهر چندوجهی[[15]](#footnote-16): در این مدل‌سازی چشم‌انداز های متفاوتی از هدف مورد نظر ذخیره شده و به کمک الگوریتم هایی همانند آنالیز مولفه های اساسی[[16]](#footnote-17) یک زیرفضا[[17]](#footnote-18) از تصاویر ذخیره شده تشکیل می‌شود که بیان کننده ویژگی های ظاهری هدف مورد نظر می‌باشد.

تا این مرحله روش‌های مختلف مدل‌سازی یک شئ و یا یک هدف را مورد بررسی قرار دادیم. در حالت کلی بین نحوه مدل‌سازی یک هدف و الگوریتم رهگیری آن ارتباط قوی و معنا‌دای وجود دارد. نحوه مدل‌سازی یک هدف مشخص در یک سیستم رهگیری به این بستگی دارد که از آن سیستم در چه زمینه ای، برای چه هدفی و برای چه کاربردی استفاده می‌کنیم. برای درک بیشتر موضوع فرض کنید که هدف از ایجاد یک سیستم رهگیری، دنبال کردن و یا ردیابی یه شئ با اندازه‌ی بسیار کوچک باشد، در این حالت بهترین انتخاب ما برای مدل‌سازی هدف، مدل نقطه‌ای می‌باشد. به عنوان مثال ردیابی پرنده‌های در حال پرواز که با دوربین فاصله بسیار زیادی داشته و ابعاد آن ها در تصویر ضبط شده بسیار کوچک است. به عنوان مثالی دیگر فرض کنید که هدف مورد رهگیری دارای ابعادی باشد که بتوانیم آن را به کمک یک مستطیل و یا یک بیضی مدل‌سازی کنیم. در این حالت بهترین گزینه برای مدل‌سازی هدف، مدل شکل و هندسه اصلی می‌باشد. به عنوان مثال برای رهگیری ماشین های در حال حرکت در جاده ها می توانیم از این نوع مدل‌سازی استفاده کنیم. یا فرض کنید که هدف مورد نظر دارای ابعاد و ساختار پیچیده‌تری باشد. در این حالت بهترین گزینه برای مدل‌سازی، مدل سطح خارجی و یا داخلی می‌باشد. به عنوان مثال برای رهگیری یک انسان می توان از این نوع مدل‌سازی استفاده کرد. البته دقت کنیم که مدل شکل و هندسه اصلی نیز در این مورد کارساز بوده و می‌تواند مورد استفاده قرار بگیرد. در قسمت یکی دیگر از مراحل رهگیری هدف با عنوان انتخاب ویژگی های ساختاری و ظاهری هدف را مورد بررسی قرار می‌دهیم.

2-2-2- انتخاب ویژگی‌های ساختاری و ظاهری

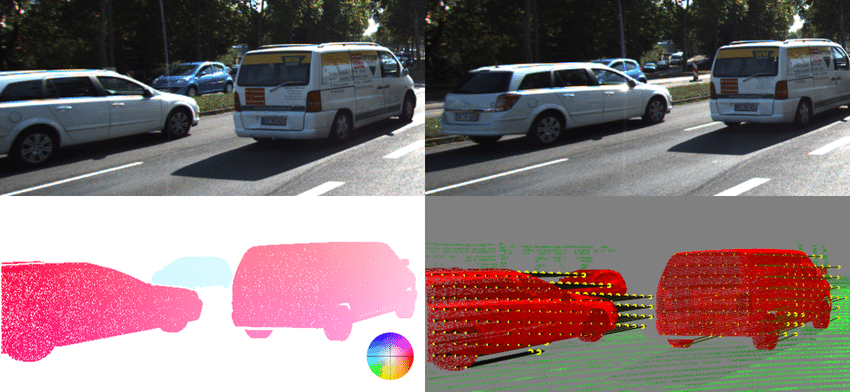
انتخاب ویژگی‌های ظاهری و ساختاری مناسب از هدف مورد نظر نقش بسیار مهم و حیاتی در رهگیری هدف ایفا می‌کند. در حالت کلی می‌توان گفت که ویژگی‌های مطلوب یه هدف و یا یک شئ بیان‌کننده منحصر‌به‌فرد بودن آن هدف می‌باشد. در نتیجه به کمک ویژگی‌های مطلوب و منحصر‌به‌فردی که هدف مورد نظر دارد می‌توان آن را از سایر اشیا تشخیص و تمیز داد. انتخاب ویژگی های یک شئ ارتباط بسیار نزدیکی با نحوه مدل‌سازی آن شئ دارد. به عنوان مثال اگر هدف مورد نظر به کمک مدل سطح داخلی و یا خارجی مدل‌سازی شده باشد، لبه‌های هدف مورد رهگیری یک ویژگی حیاتی و بسیار مهم است که باید انتخاب شود. و یا به عنوان مثالی دیگر اگر از مدل‌سازی چگالی احتمال-نمودار هیستوگرام برای مدل‌سازی استفاده شود، رنگ هدف مورد نظر می‌تواد یک نقش بسیار حیاتی برای رهگیری آن بازی کند. در حالت کلی الگوریتم های رهگیری ترکیبی از انواع ویژگی های ساختاری و ظاهری استفاده می‌کنند. در ادامه دسته‌های مختلف ویژگی‌های یک شئ را بررسی می‌کنیم.

* رنگ: رنگ ظاهری یک شئ تحت تاثیر دو پارامتر اصلی و فیزیکی قرار دارد: پارامتر اول چگالی طیفی از روشنایی هدف مورد نظر میباشد و پارامتر دوم ویژگی های بازتابی سطح هدف مورد رهگیری می باشد. در پردازش تصویر فضاهای رنگی متفاوتی وجود دارد. یکی از معروف ترین فضاهای رنگی مورد استفاده در پروژه های پردازش تصویری فضای رنگی RGB می باشد که یکی از ویژگی های این فضای رنگی یونیفرم[[18]](#footnote-19)  نبودن این فضای رنگی می باشد به این معنی که تفاوت مقادیر رنگی در پیکسل های فضای RGB با مقادیر این رنگ ها که توسط چشم انسان مشاهده می شود متفاوت است. دو نمونه دیگر از فضاهای رنگی مورد استفاده L.u.v و L.a.b میباشند.این فضاهای رنگی برعکس فضای رنگی RGB یونیفرم می باشند. دو فضای رنگی اخیر حساسیت بالاتری به نویز دارند. بنابراین در حالت کلی نمی‌توان گفت کدام فضای رنگی مناسب ترین برای پردازش تصویر است و با توجه به کاربرد سیستم رهگیری و سامانه پردازش تصویری که وجود دارد یکی از این فضاهای رنگی می تواند مناسب ترین انتخاب برای سیستم باشد.
* کناره‌های تصویر[[19]](#footnote-20): کناره های یک جسم مشخص در یک تصویر و یا لبه های مرزی آن جسم عموما تغییرات شدیدی در شدت تصویر ایجاد می کنند از این واقعیت برای پیاده سازی الگوریتم های متفاوت تشخیص لبه استفاده می شود. یکی از مهمترین ویژگی‌های لبه تصویر این است که به تغییرات میزان شدت روشنایی در تصویر حساسیت پایینی دارد.  الگوریتمهای رهگیری و یا ردیابی که هدف آنها دنبال کردن و یا ردیابی یک جسم توسط لبه های مرزی آن است از این ویژگی به عنوان یک ویژگی پایه و اساسی استفاده می کنند. علت استفاده از این ویژگی نیز سادگی آن و دقت بالای آن می باشد. مشهورترین الگوریتم تشخیص لبه الگوریتم Canny می‌باشد. یک نمونه از این ویژگی را در تصویر (3-2) مشاهده می‌کنید.

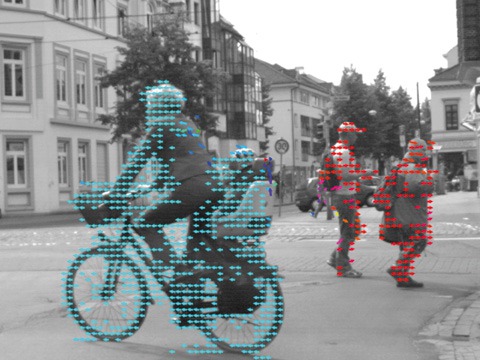


شکل((2-3)): نمونه ای از ویژگی لبه تصویر

* جریان نوری[[20]](#footnote-21):جریان نوری به معنای بیان بردار جابجایی در یک تصویر می باشد. به بیانی دیگر جریان نوری بیان کننده میزان جابجایی و انتقال هر پیکسل از تصویر در یک محیط مشخص می باشد. این ویژگی از تصویر به کمک میزان روشنایی نور و میزان ثبات روشنایی نور در هر پیکسل محاسبه می شود. برای روشن شدن موضوع فرض کنید تعدادی فریم از یک ویدیو ضبط شده را پشت سر هم از نظر مقدار روشنایی پیکسل ها مقایسه میکنیم، به کمک این مقادیر و میزان تغییرات شدت روشنایی پیکسل ها و ارتباطی که این پیکسل ها با هم دارند میزان جریان نوری محاسبه می‌شود که عملاً مشخص کننده میزان جابجایی و جهت جابجایی یک قسمت از تصویر می باشد. از جابجایی نوری در کاربرد هایی همانند تشخیص حرکت در یک تصویر و همچنین رهگیری و ردیابی اجسام استفاده میشود. در شکل های (2-4)و (2-5) دو نمونه از نتایج استفاده از ویژگی جریان نوری را مشاهده می‌کنید.



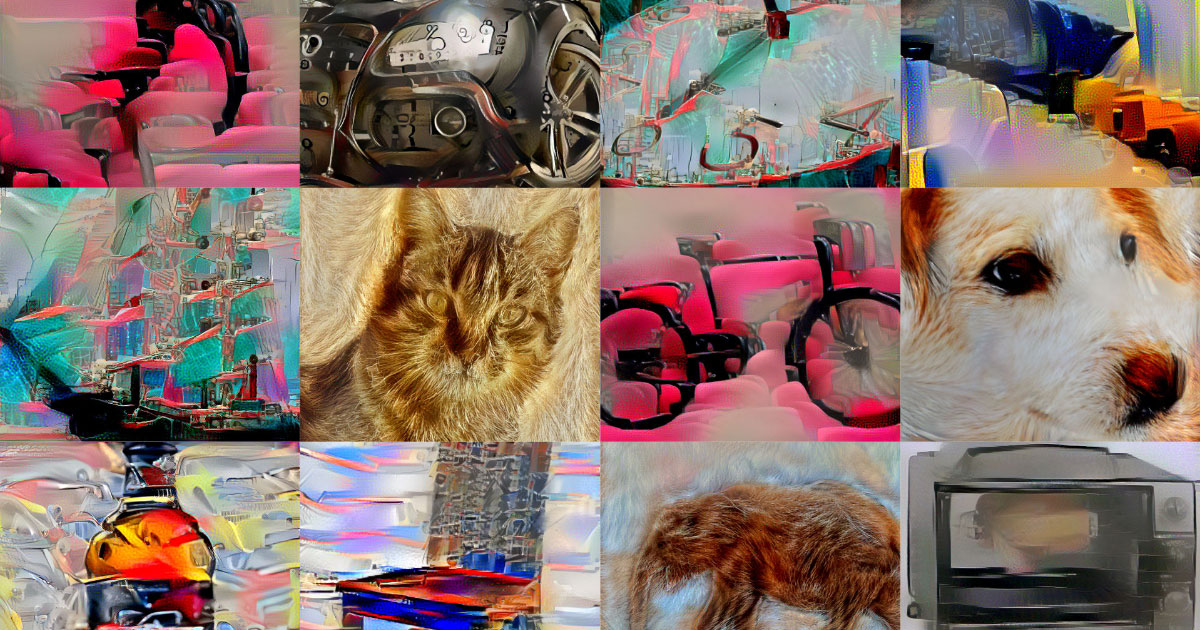
شکل((2-4)): نمونه ای از نتایج استفاده از ویژگی جریان نوری



شکل((2-5)): نمونه ای از نتایج استفاده از ویژگی جریان نوری

همان طور که در شکل های (2-4) و (2-5) مشاهده می‌کنید، در هر دو تصویر به کمک ویژگی جریان نوری، بردار جا‌به‌جایی به دست آمده و اجسام در حال حرکت تشخیص داده شده‌اند. همان‌طور که می‌دانیم تشخیص اجسام محرک یکی از کابرد‌های جریان نوری است.

* بافت[[21]](#footnote-22): بافت یک تصویر بیان کننده تغییرات شدت ویژگی های سطح یک تصویر می باشد به طوری که ویژگی هایی مانند صافی و نظم اجسام را با مقادیر عددی بیان می کند. در حالت کلی بافت یک جسم مربوط به ویژگی‌های سطح آن جسم و ویژگی‌های ظاهری سطح آن جسم از جمله ابعاد، ساختار، شدت نظم و ترتیب و غیره می باشد. عموما بافت توست صاف بودن و یا ناصاف بودن، سخت و یا نرم بودن و منظم و یا منظم نبودن سطح جسم مورد نظر بیان می شود. در شکل (2-6) یک نمونه از ویژگی های بافت که به صورت تصویر بیان شده است را مشاهده می‌کنیم. به معنای دیگر این شکل بیان می‌کند که کدام ویژگی‌های تصویر به عنوان ویژگی‌های بافت تصویر شناخته می‌شوند.



شکل(2-6): ویژگی های بافت یک تصویر

تا این قسمت چهار مورد  از مهمترین ویژگی های ساختاری و ظاهری یک جسم که می تواند به عنوان یک ویژگی پایه و اساسی در سیستم ردیابی و رهگیری استفاده شود را معرفی کردیم. به این نکته توجه کنیم که ویژگی رنگ یک جسم یکی از مهمترین ویژگی هایی است که در اکثر سیستم های رهگیری و ردیابی استفاده می شود. ویژگی ها به صورت دستی و توسط افراد، متناسب با کاربردی که آن سیستم ردیابی دارد انتخاب میشوند. هرچند امروزه انتخاب خودکار و اتوماتیک ویژگی ها بدون دخالت انسان پیشرفت گسترده ای داشته و در کاربرد های متفاوت استفاده می شود.یکی از بهترین و کارآمدترین روش های استخراج خودکار ویژگی ها استفاده از شبکه های عصبی و شبکه های عمیق می باشد. تا این مرحله نحوه مدل‌سازی هدف مورد نظر و همچنین ویژگی‌هایی که می‌توان از آن‌ها در رهگیری هدف استفاده کرد بررسی کردیم، در ادامه به بررسی انواع روش‌های تشخیص هدف و یا شئ می‌پردازیم.

3-2-2- الگوریتم های تشخیص[[22]](#footnote-23) هدف

هر الگوریتم ردیابی نیاز به یک مکانیزم تشخیص هدف دارد. از این مکانیزم تشخیص هدف در هر دو حالت فریم اول و همه فریم ها با توجه به کاربردی که از سیستم ردیابی داریم می توانیم استفاده کنیم. متداول ترین روش های تشخیص هدف، استفاده از اطلاعات پیکسل ها در هر فریم جداگانه است هرچند برخی از این الگوریتم های تشخیص هدف از تعدادی فریم پشت سرهم و همچنین اطلاعات موقتی برای کم کردن خطای تشخیص استفاده می کنند. در حالت کلی این اطلاعات موقتی شامل تفاوت مقدار پیکسل ها در فریم های متوالی می باشد. بعد از تشخیص هدف مورد نظر وظیفه الگوریتم ردیابی هست که این هدف را فریم به فریم دنبال کند. در ادامه تعدادی از الگوریتم های تشخیص هدف را بررسی می‌کنیم.

1-3-2-2- تشخیص دهنده‌های نقطه‌ای[[23]](#footnote-24)

تشخیص دهنده های نقطه ای برای پیدا کردن نقاط محتمل در تصویر که دارای ساختار ظاهری بسیار مشخص و همچنین متفاوت با دیگر نقاط می باشند استفاده می شوند. از این نوع تشخیص دهنده در کاربردهای متفاوت  همچون حرکت و ردیابی استفاده می شود. مهمترین مشخصه نقاط محتمل بدین صورت می باشد که این نقاط در برابر تغییرات روشنایی و همچنین زاویه دید دوربین ثابت بوده و تغییرات زیادی ندارد. در حالت کلی الگوریتم های که از این روش برای پیدا کردن اشیا و نقاط محتمل استفاده می کنند با محاسبه میزان تغییرات شدت تصویر به کمک روش هایی همچون مشتق گیری نقاطی از تصویر که دارای تغییرات بالایی می باشد و احتمال وجود یک هدف و یا یک شئ و یا به اصطلاح یک نقطه محتمل در آن محل وجود دارد را تشخیص می‌دهند. به عنوان مثال دو نمونه از این نوع الگوریتم ها شامل الگوریتم تشخیص KLT و همچنین الگوریتم تشخیص Harris می باشد. هر دو روش به کمک محاسبه ماتریس موجود در فرمول (3-1) نقاط محتمل را تشخیص میدهند.

(2-1) 

تشخیص دهنده Harris با محاسبه تابع تغییرات که در فرمول شماره (3-1) به آن اشاره شده است و همچنین مقایسه آن با یک مقدار آستانه[[24]](#footnote-25) نقاط را پیدا می کند و تشخیص دهنده KLT به کمک رابطه موجود در فرمول (3-2) با محاسبه مقادیر ویژه ماتریس تعریف شده و مقایسه آن‌ها با یک مقدار آستانه این کار را انجام میدهد.

(2-2) 

در حالت کلی تمامی الگوریتم های موجود در این گروه به کمک محاسبه تغییرات موجود در تصویر و مقایسه آن با یک مقدار آستانه تشخیص می‌دهند که آیا این نقطه، نقطه محتمل می‌باشد و یا خیر. در شکل (2-7) سه نمونه از نتایج الگوریتم های مختلف بر روی یک تصویر را مشاهده می‌کنیم.



شکل(2-7): نتایج تشخیص دهنده های نقطه‌ای

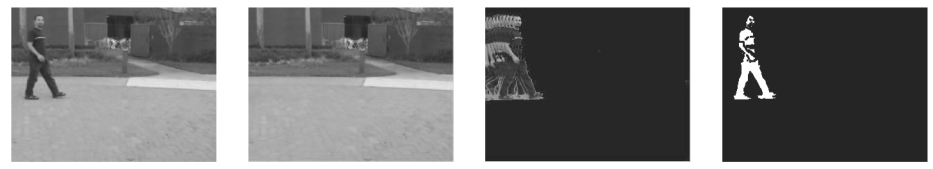
2-3-2-2- اختلاف پس‌زمینه[[25]](#footnote-26)

تشخیص هدف مورد نظر در یک تصویر و یا یک ویدیو می تواند به کمک ایجاد یک مدل از تصویر به نام مدل پس زمینه انجام شود. در این نوع از الگوریتم ها پس از محاسبه مدل پس زمینه در هر فریم اختلاف این مدل در فریم های متوالی اطلاعاتی را به ما میدهد که می توانیم هدف مورد نظر خود را تشخیص دهیم.تشخیص هدف در این الگوریتم‌ها بدین صورت است که هر تغییر مهم در مقادیر پیکسل مدل پس زمینه، یک هدف در حال حرکت را بیان می کند پیکسل هایی از تصویر که شامل این اختلاف در فریم های متوالی می باشند برای محاسبات بعدی علامت گذاری می شوند. عموماً در این نوع الگوریتم ها پس از به دست آوردن پیکسل های جسم محرک یک الگوریتم برای ارتباط دادن تک تک پیکسل های جدا از هم به یکدیگر و ایجاد یک ناحیه مشخص از تصویر که بیان کننده یک جسم کامل میباشد استفاده میشود. تمامی این مراحل با عنوان مدل اختلاف پس‌زمینه شناخته می‌شود. عموماً در این روش ها تعدادی فرم متوالی به الگوریتم تشخیص شئ و یا هدف داده میشود. با مقایسه مقدار پیکسل ها در این فریم های متفاوت و به دست آوردن میزان تغییرات درهر تصویر مکان هایی از تصویر و یا پیکسل هایی از تصویر که دچار تغییرات در طول زمان یا به عبارتی دیگر در پیکسل های متوالی شده‌اند به دست می آیند. اگر میزان این تغییرات از مقدار مشخصی بیشتر باشد آن پیکسل از تصویر را با مقدار رنگی مشکی و یا سفید به عنوان تصویر پیش زمینه که شامل جسم محرک است مشخص میکنیم و قسمتهایی از تصویر که تغییرات آن ها از یک مقدار مشخص کمتر است با رنگ مخالف رنگ قبلی با عنوان پس زمینه مشخص میکنیم. عملا می‌توانیم این روش را به صورت رابطه (3-3) فرموله‌سازی کنیم.

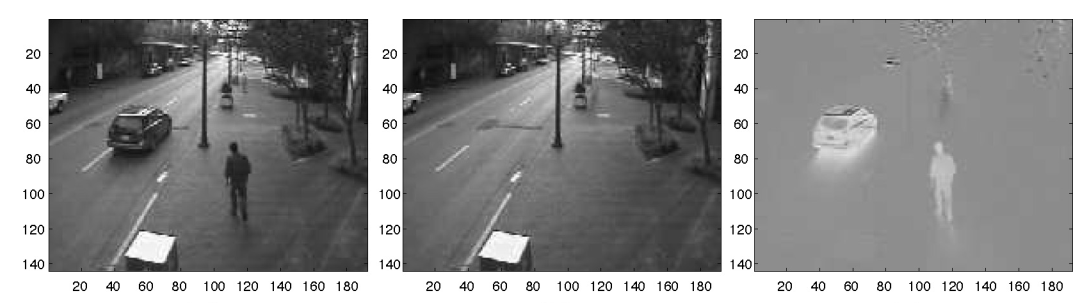
(2-3) 

در رابطه (3-3) تابع P مشخص کننده مقدار پیکسل بوده و پارامتر t مشخص کننده زمان می‌باشد. حال اگر اختلاف این تابع در دو زمان متوالی از مقدار مشخصی کم‌تر و یا بیشتر باشد مشخص می‌کند که این پیکسل متعلق به مدل پس زمینه است و یا پیش‌زمینه.

یکی دیگر از روش هایی که در این نوع از تشخیص دهنده های هدف می توانیم استفاده کنیم این است که به جای محاسبه تغییرات هر پیکسل در دو فریم متوالی از تعدادی فریم متوالی استفاده کرده و با تشکیل یک ماتریس که سطرهای آن را فریم های مختلف تشکیل میدهند و همچنین با محاسبه مقادیر ویژه ماتریس و انتخاب مهمترین مقادیر ویژه مدل پس زمینه و پیش زمینه تصویر را به دست آوریم. در شکلهای( 2-8 )و (2-9) دو نمونه از نتایج این نوع از الگوریتم را مشاهده می‌کنیم که شکل شماره (2-8) متعلق به روش اول یعنی محاسبه اختلاف مقادیر پیکسل ها در دو فریم متوالی و همچنین شکل شماره (2-9) متعلق به روش دوم یعنی تشکیل ماتریسی از پیکسل های متوالی و به دست آوردن مدل پیش زمینه و پس زمینه می باشد.



شکل(2-8): نتیجه الگوریتم با دو فریم متوالی



شکل(2-9): نتیجه الگوریتم با تعدادی فریم متوالی

همان‌طور که در تصاویر (2-3) و (2-9) مشاهده می‌کنید اجسام متحرک که در تصویر وجود دارند به خوبی تشخیص داده شده و به کمک رنگ سفید و یا مشکی، متناسب با انتخاب کاربر، مشخص شده است. امروزه استفاده از این روش در سیستم های ردیابی با دوربین‌ثابت گسترش زیادی پیدا کرده است.علت این امر نیز این می‌باشد که این نوع از الگوریتم ها امروزه توانایی مدل کردن میزان تغییرات روشنایی و همچنین نویز و یا اغتشاش را دارند، بنابراین می‌توانند به خوبی در مقابل این نوع از تغییرات و اغتشاش ها از خود مقاومت نشان دهند.

3-3-2-2- قسمت‌بندی[[26]](#footnote-27)

هدف از تقسیم بندی تصویر جدا کردن تصویر به قسمتهای مختلف می باشد که از نظر ظاهری شبیه هم هستند. هر الگوریتم تقسیم بندی تصویر با دو چالش مهم روبرو می باشد اولین چالش انتخاب معیار مناسب برای تقسیم بندی تصویر می باشد و چالش دوم رسیدن به یک دقت خوب در تقسیم بندی تصویر می باشد.

برای پیاده‌سازی این مدل روش های مختلفی از جمله روش Mean-Shift Clustering و همچنین روش Graph-Cuts معرفی شده‌اند. در روشMean-Shift Clustering هر کدام از پیکسل ها به کمک فضای رنگی و مختصات آن ها در تصویر مدل می‌شوند. درواقع هر پیکسل مبین یک بردار با 5 درایه است که درایه های آن را مقادیر فضای رنگی و مختصات پیکسل تشکیل می‌دهند. نحوه این مدل‎‌سازی تصویر را در رابطه (3-4) مشاهده می‌کنیم.

(2-4) 

همان‌طور که در رابطه (3-4) مشخص است مقادیر l, u, v که بیان‌کننده مقادیر فضای رنگی مثلا RGB هستند به همراه مقادیر x, y که بیان کننده مختصات تصویر هستند یک مدل‌ برداری از پیکسل را به ما می‌دهند.بعد از مدل‌سازی برداری پیکسل ها تعدادی مرکز به صورت اختیاری در تصویر تعریف می‌شوند. دقت کنیم که تعداد این مراکز وابسته به تعداد قسمت‌ها و یا خوشه‌های[[27]](#footnote-28) مختلفی است که ما می‌خواهیم در تصویر ایجاد کنیم. بعد از انتخاب مراکز، فاصله هر کدام از پیکسل ها با هر یک از مراکز محاسبه شده و پیکسل مورد نظر به خوشه ای که کم‌ترین فاصله را با آن دارد تعلق می‌گیرد. بعد از این که تمامی پیکسل ها گروه بندی شدند، مقادیر مراکز تمامی خوشه ها بروزرسانی می‌شود. برای این کار میانگین تمامی پیکسل‌های متعلق با آن خوشه محاسبه شده و به عنوان مرکز جدید خوشه مورد نظر پذیرفته می‌شود. در رابطه (5-3) نحوه گروه‌بندی پیکسل ها و در رابطه (6-3) نحوه بروزرسانی مراکز را مشاهده می‌کنیم.

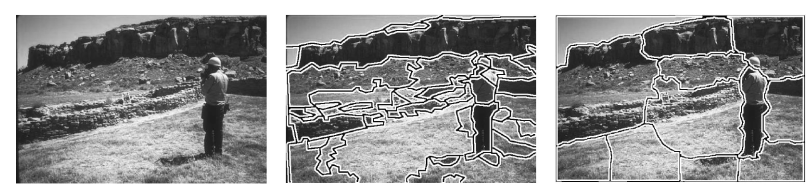
(2-5) 

همان‌طور که در رابطه (5-3) مشاهده می‌کنیم،مرکز پیکسل شماره i برابر است با مرکزی که با آن کمترین فاصله را دارد. دقت کنیم که در این رابطه P بیانگر پیکسل و C بیانگر مرکز می‌باشد.

(2-6) 

همان‌طور که در رابطه (6-3) مشاهده می‌کنیم، مرکز جدید هر خوشه به کمک میانگین گرفتن از تمامی پیکسل های متعلق به آن خوشه به دست می‌آید. دو مرحله قبلی را تا جایی ادامه می‌دهیم که دیگر تغییراتی در مراکز خوشه‌ها نداشته باشیم. بعد از این که مراکز خوشه‌ها ثابت شدند می‌توانیم تصویر را به قسمت هایی تقسیم کنیم که هر یک از این قسمت ها بیانگر یک ناحیه مشخص با ویژگی‌های مشخص می‌باشد. هر کدام از این ناحیه ها می‌تواند بیانگر هدف و یا شئ مورد نظر ما در سیسم ردیابی باشد. نحوه تقسیم بندی تصویر بعد از ثبات مراکز نیز بدین‌گونه است که پیکسل‌های متعلق به هر مرکز از بقیه پیکسل ها جدا ‌می‌شوند و بدین صورت نواحی مختلفی در تصویر ایجاد می‌شود.

از دیگر روش هایی که می‌توانیم برای تقسیم‌بندی تصویر از آن استفاده کنیم بدین صورت است که ابتدا تصویر را با یک گراف مدل می‌کنیم که هر پیکسل از تصور بیانگر یک نقطه ار آن گراف بوده و خطوط ارتباطی در آن گراف بیانگر مشخصات ظاهری از جمله رنگ، شدت تصویر و میزان روشنایی می‌باشد. سپس این گراف را به کمک روش‌های تقسیم گرافی که به دنبال کمینه کردن یک تابع هزینه در راستای خواسته مورد نظر(تقسیم گراف) می‌باشند، به قسمت هایی تقسیم می‌کنیم. به علت وجود تابع هزینه در این حالت نیز مانند حالت قبل، قسمت های به دست آمده در تصویر بیانگر یک ناحیه مشخص با ویژگی‌های مشخص می‌باشد. در شکل (2-10) دو نمونه از نتایج این نوع از روش تشخیص هدف(قسمت‌بندی) را مشاهده می‌کنیم.



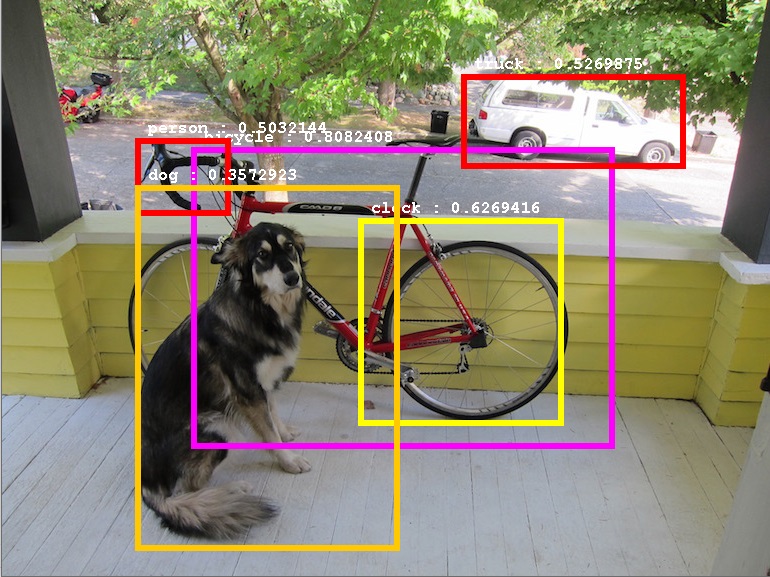
شکل(2-10): تشخیص شئ به کمک روش تقسیم‌بندی

4-3-2-2- یادگیری با نظارت[[28]](#footnote-29)

برای تشخیص یک هدف می توانیم بدین صورت عمل کنیم که یک تابع را تعریف می‌کنیم و سپس با کمک تعدادی نمونه از شیء مورد نظر که از زوایای مختلف در تصویر وجود دارند این تابع را آموزش می دهیم تا یاد بگیرد که این شی را در تصاویر مختلف تشخیص دهد. برای انجام این کار به تعدادی نمونه از هدف مورد نظر نیاز داریم. به کلیت این نوع از تشخیص هدف، یادگیری با نظارت می گویند. عملکرد در این روش بدین گونه است که به کمک داده هایی که برای آموزش در دست داریم یک تابع تولید می‌شود که می‌تواند ورودی را که یک فریم از تصویر است به خروجی که در آن فریم هدف مورد نظر مشخص شده است تبدیل کند. به خاطر اینکه در این روش هدف یادگیری یک تابع است و یا به معنای دیگر هدف تولید یک تابع است که بتواند وظیفه خواسته شده را انجام دهد نمونه‌های آموزش این تابع باید به صورت جفت داده های ورودی و خروجی باشد. داده های ورودی که برای ما در یک سیستم رهگیری مورد نیاز است فریم های یک تصویر و خروجی مورد نظر ما همان فریم ولی با این تفاوت که در آن هدف مورد نظر مشخص شده است. به این نوع از روش ها، کلاس‌بندی[[29]](#footnote-30)و در حالت تخصصی تر تشخیص هدف گفته می‌شود.

انتخاب ویژگی های مناسب نقش اساسی در میزان عملکرد یک سیستم تشخیص هدف در این روش دارد. اینکه کدام یک از ویژگی ها انتخاب شوند تا به کمک آنها بین هدف مورد نظر و بقیه اجسام موجود در شکل تمایز ایجاد شود مهم است. اگر ویژگی های مناسب از اشیا و اهداف مورد نظر انتخاب شوند در زوایای مختلف و در شرایط مختلف این سیستم به خوبی می تواند هدف مورد نظر را تشخیص دهد. این نوع از روش‌ها شامل شبکه های عصبی، درخت تصمیم، بردار ماشین پشتیبان و غیره می باشد. روش های یادگیری با نظارت  یک منبع داده مناسب و به اندازه کافی بزرگ نیاز دارند که در هر کدام از این داده های فردی هدف مورد نظر باید مشخص شود و به اصطلاح باید لیبل گذاری شود. وقتی که این داده ها وارد الگوریتم مدل یادگیری می شوند قسمتهایی از تصویر که لیبل زده شده است و با یک شکل هندسی نظیر مستطیل مشخص شده است به عنوان هدف در نظر گرفته و تابع یاد می گیرد که با استفاده از ویژگی هایی که در این ناحیه وجود دارد هدف مورد نظر را تشخیص دهد.

شاید بتوان گفت که امروزه شبکه های عصبی و شبکه‌های عمیق بهترین و کاربردی‌ترین روش در این گروه از الگوریتم‌های تشخیص هدف می‌باشند. نحوه عملکرد شبکه‌های عصبی در فصل مربوط به هوش مصنوعی مورد بحث و بررسی قرار گرفت و کلیت کار بدین صورت بود که یک‌سری پارامتر های یادگیری در شبکه تعریف می‌شود. سپس این پارامتر ها با مقادیر دلخواه اولیه مقدار دهی می‌شوند. در ادامه با مشتق گیری از اختلاف خروجی شبکه و خروجی دلخواه مقادیر پارامتر های یادگیری متناسب با مقدار مشتق و یا گرادیان تغییر کرده و به سمت مقدار بهینه حرکت می‌کنند. باقی روش‌های نام‌برده که جزو شبکه های عصبی محسوب نمی‌شوند نیز از این روش کلی برای تشخیص هدف استفاده می‌کنند. بدین صورت که تمامی الگوریتم ها تعدادی پارامتر یادگیری دارند و همچنین یک تابع هزینه برای آن‌ها تعریف می‌شود. سپس به کمک مقدار این تابع هزینه(اختلاف بین خروجی الگوریتم و خروجی واقعی) پرامتر ها بروزرسانی می‌شوند تا به مقدار بهینه دست یابند. تابع بروزرسانی نیز عموما به کمک گرادیان تابع هزینه تعریف می‌شود، چرا که به کمک گرادیان تابع هزینه می‌توان به مقدار کمینه دست پیدا کرد که این مقدار کمینه یعنی نزدیک‌ترین حالت به خروجی واقعی. در شکل (2-11) یک نمونه از خروجی این نوع از روش ها را مشاهده می‌کنیم.



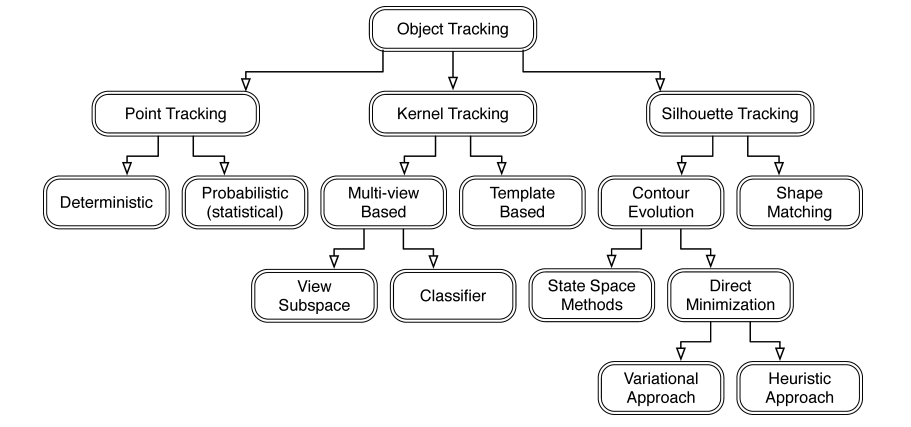
شکل(2-11): خروجی یکی از الگوریتم های یادگیری بانظارت

همان طور که در شکل (2-11) مشاهده می‌کنید الگوریتم به خوبی توانسته است اهداف و اشیا مورد نظر را که الگوریتم بر اساس آن آموزش دیده است تشخیص دهد. الگوریتم استفاده شده در این تصویر یک شبکه عصبی به نام YOLO می‌باشد که در فصل هوش مصنوعی به آن اشاره شده‌است. امروزه شبکه های عصبی پرکاربردترین روش برای تشخیص هدف می‌باشد ولی این بدان معنا نیست که مابقی روش‌ها کاربردی ندارند. بسته به کاربری که از سیستم رهگیری انتظار می‌رود هر کدام از روش‌های تشخیص هدف می‌تواند مورد استفاده قرار گیرد. تا این مرحله روش‌های مختلف تشخیص اهداف را بررسی کردیم. در مرحله بعد در مورد الگوریتم‌های رهگیری و ردیابی که در واقع مرحله آخر در سیستم رهگیری هدف می‌باشند بحث می‌کنیم.

4-2-2- الگوریتم‌های ردیابی

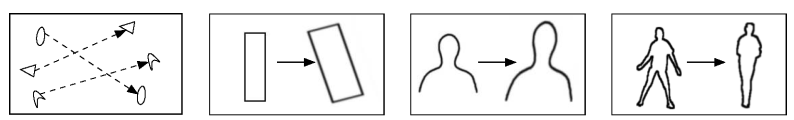
هدف از رهگیری یک شئ و یا یک هدف به دست آوردن و تولید کردن مسیر حرکتی آن هدف در طول زمان با استفاده از مکان یابی محل آن هدف و یا شی در هر فریم ویدیو می باشد. تشخیص هدف و تولید مسیر ارتباطی بین فریم های مختلف که مرتبط با شی تشخیص داده شده می باشد می‌توانند به صورت همزمان و جدا از هم صورت گیرند. در مورد اول در هر فریم مکان  هدف مورد نظر توسط الگوریتم های تشخیص اشیا مشخص می شود سپس با کنار هم قرار دادن فریم های متفاوت و محل اشیا در فریمها مسیر ارتباطی و مسیر حرکتی هدف مورد نظر به دست می آید. در مورد دوم تشخیص شئ و همچنین مکان یابی و مسیر ارتباطی در فریم های متفاوت به صورت همزمان و به کمک اطلاعات فریم های قبلی به دست می‌آید. به معنای دیگر در  یک فریم مشخص محل هدف مورد نظر و مسیر ارتباطی به وجود آمده، توسط اطلاعات فریم های قبلی تولید می شود. در هر دو مورد بیان شده هدف و شی مورد نظر به وسیله ساختار هندسی و مشخصات ظاهری  که در بخش انتخاب ویژگی های ساختاری و ظاهری در مورد آنها بحث شد مشخص می گردد.

روش‌های مختلف و الگوریتم‌های متفاوتی برای رهگیری معرفی و تولید شده‌اند که در ادامه به بررسی هر کدام از آن‌ها می‌پردازیم. در شکل (2-12) تمامی الگوریتم های پیاده‌سازی شده تا الان را مشاهده می‌کنید. ابتدا توضیحات کلی در مورد هر گروه داده می‌شود و سپس به صورت جزئی‌تر هر کدام مورد بحث قرار می‌گیرد.



شکل(2-12): دسته‌بندی انواع روش‌ها و الگوریتم‌های ردیابی

* ردیابی نقطه‌ای[[30]](#footnote-31): در این روش هدف مورد  نظر در فریم های متوالی به کمک یک نقطه مدل سازی میشود و ارتباط این نقاط بر پایه حالت های قبلی نقطه مورد نظر در فریم های قبلی می باشد که شامل محل هدف مورد نظر و همچنین جهت حرکت آن است. این نوع از روشهای ردیابی به یک مکانیزم خارجی برای تشخیص دادن هدف مورد نظر در فریم نیاز دارند. یک نمونه از نحوه بیان حرکت و رهگیری در این نوع از روش‌ها را در شکل (2-13-الف) مشاهده می‌کنیم.
* ردیابی کرنل[[31]](#footnote-32) :کرنل بیان کننده ساختار و ظاهر یک شئ می باشد. به عنوان مثال یک کرنل می تواند یک مستطیل با یک نمودار هیستوگرام مرتبط با آن باشد بدین صورت که هدف مورد نظر داخل آن مستطیل قرار گرفته و هیستوگرام مشخص کننده مشخصات ظاهری هدف مورد نظر می باشد. رهگیری هدف به کمک به دست آوردن حرکت این کرنل مستطیل شکل در فریم های متوالی به دست می آید. نمونه‌ای از نحوه بیان حرکت و رهگیری در این نوع روش را در شکل (2-13-ب) مشاهده می‌کنید.
* ردیابی به کمک ساختار داخلی[[32]](#footnote-33): این روش بسیار شبیه روش ردیابی کرنل می باشد. با این تفاوت که در این روش هدف مورد نظر به کمک مدل ساختار داخلی که آن را در بخش نحوه مدلسازی هدف مورد بحث و بررسی قرار دادیم مدل سازی میشود. همانند روش کرنل در این روش نیز  علاوه بر مشخص کردن ناحیه هدف  مورد نظر در تصویر یک سری اطلاعات ظاهری نیز که به کمک مشخصات ناحیه داخلی مدل ساختار داخلی به دست می آید مدل سازی می شوند. مزیتی که این روش نسبت به روش کرنل دارد در این است که در این روش یک سری اطلاعات اضافی نیز در رابطه با ساختار هدف مورد نظر که همان ساختار داخلی و یا مکمل آن ساختار خارجی میباشد در اختیار داریم. دو نمونه از نحوه بیان حرکت و همچنین ردیابی هدف به کمک این روش را در شکل‌های (2-13-پ)و (3-13-ت)مشاهده می‌کنید.



(ت) (پ) (ب) (الف)

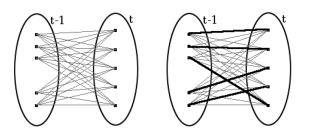
شکل(2-13): نمونه‌هایی از نحوه بیان حرکت و رهگیری به کمک الگوریتم ها و روش‌های مختلف

تا این مرحله کلیات انوان روش‌ها و الگوریتم‌های ردیابی را توضیح دادیم. در ادامه به صورت جزئی‌تر به بررسی هر کدام می‌پردازیم.

1-4-2-2- ردیابی نقطه‌ای

ردیابی می تواند به صورت ارتباط اشیا و اهداف تشخیص داده شده که به صورت نقطه مدلسازی شده اند تعریف شود. ارتباطات نقطه‌ای یکی از پیچیده ترین مسائل موجود در زمینه رهگیری می‌باشد، به خصوص زمانی که  با مسائلی همچون از دست رفتن هدف و وجود مانع در تصویر رو به رو هستیم. در حالت کلی  روش های مبتنی بر ارتباطات نقطه‌ای به دو دسته کلی تقسیم می شوند. روشهای قطعی[[33]](#footnote-34) و روشهای آماری[[34]](#footnote-35). در روش های قطعی به کمک کیفیت و نحوه حرکت جسم تشخیص داده شده هدف را ردیابی می کنیم و در روش های آماری به کمک اطلاعات به دست آمده از فریم های قبلی و تعریف الگوریتم های نظیر فضای حالت هدف مورد نظر را رهیگیری می کنیم. در ادامه به بررسی هر کدام از این روش‌ها می‌پردازیم.

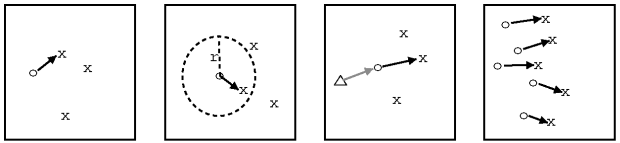
* روش قطعی: در روش قطعی برای تعریف ارتباطات نقطه‌ای یک تابع هزینه بین هر هدف تشخیص داده شده در فریم قبلی و فریم موجود تعریف می شود.(شکل( 2-14-الف)) که این تابع هزینه بر اساس یک سری محدودیت هایی قابل تعریف شدن میباشد.به دنبال تعریف تابع هزینه، هدف حداقل کردن مقدار این تابع است و مرحله آخر در این روش پیدا کردن کمترین مقدار تابع هزینه بین یک هدف مشخص در فریم قبلی و مجموعه ای از اهداف تشخیص داده شده در فریم حاضر می‌باشد که مشخص میکند هدف تشخیص داده شده در فریم قبلی به کدام هدف و یا شی تشخیص داده شده در فریم حاضر مرتبط است (شکل (2-14-ب)).



(ب) (الف)

شکل(2-14): تعریف تابع هزینه بین هر هدف در فریم قبلی و فرم حاضر و انتخاب مقدار بهینه

محدودیت هایی که برای تعریف تابع هزینه در این روش تعریف می شوند عبارتند از: مختصات مکانی هدف تشخیص داده شده نباید بیشتر از یک مقدار معین تغییر کند(شکل (2-15 – الف))، حداکثر سرعتی که هدف در فریم های متوالی دارد باید از یک مقدار بیشینه کمتر باشد(شکل (2-15 – ب))، سرعت و جهت شئ نباید به شدت تغییر کند(شکل (2-15 – پ)) و سرعت نقاط مشخص در یک ناحیه و همسایگی کوچک نباید تفاوت بالایی با هم داشته باشند(شکل (2-15 – ت)) که این مورد برای مدل سازی که در آن هدف به کمک چند نقطه مدل سازی شده است مناسب میباشد. دقت کنیم که این محدودیت ها صرفاً برای روش ردیابی قطعی نمی‌باشد و از آنها در روش های آماری نیز می‌توانیم استفاده کنیم.



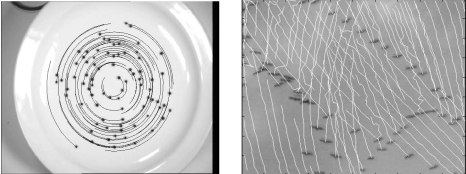
(ت) (پ) (ب) (الف)

**شکل(2-15): محدودیت‌هایی که از آن‌ها برای تعریف تابع هزینه در روش قطعی استفاده می‌کنیم.**

* روش آماری: تصاویری که از دوربین ها ثبت می شوند به احتمال زیاد ممکن است دارای اختلال و نیز باشندو همچنین در بعضی مواقع ممکن است که مشکلاتی از جمله آشفتگی در تصاویر به وجود آید. روش های آماری  این چنین مشکلات را حل کرده‌اند. این نوع از روش‌ها یک حالت برای یک هدف مشخص تعریف می‌کنند.روش های آماری به کمک تعریف فضای حالت برای مشخصات هدف مورد نظر نظیر مختصات هدف، سرعت هدف و شتاب هدف عمل رهگیری را انجام می دهند. در فرمول موجود در رابطه (2-7) نحوه بیان فضای حالت در حالت کلی بیان شده است.

(2-7) 

در فرمول بیان‌شده در رابطه (2-7) متغیر X بیان‌کننده حالت سیستم بوده که یکی از مولفه‌های آن مختصات هدف مورد نظر است. مشاهده می‌کنید که در این حالت در یک فریم مشخص مختصات یک هدف که بیان کننده یک حالت از سیستم است به کمک اطلاعات فریم‌های قبلی و فضای حالت سیستم به دست می‌آید. همچنین متغیر W بیان‌کننده نویز سفید می‌باشد. دو نمونه از نتایج این روش را در شکل (2-16) مشاهده می‌کنید.



شکل(2-16): دو نمونه از نتایج روش آماری در ردیابی نقطه‌ای

2-4-2-2-ردیابی کرنل

در این نوع از روش های ردیابی و رهگیری هدف پیدا کردن مسیر حرکتی و یا دنبال کردن حرکت هدف است که توسط ناحیه ای در فریم موردنظر مشخص شده است.  در این نوع از روش ها الگوریتم های متفاوتی وجود دارند که از جنبه های نحوه مدل‌سازی مشخصات ظاهری یک هدف، تعداد هدف‌های مورد رهگیری و همچنین روش مورد استفاده برای رهگیری با‌هم متفاوت می‌باشند. این نوع از روش ها را می توانیم به دو دسته کلی  بر اساس نحوه مدلسازی مشخصات ظاهری آنها تقسیم کنیم. دسته اول به کمک یک قالب که مشخص کننده هدف مورد نظر است رهگیری را انجام می دهد و دسته دوم بر اساس یادگیری یک الگوریتم که از زوایای مختلف هدف مورد نظر را مشاهده کرده است رهگیری را انجام می دهد. در حالت کلی همه الگوریتم های موجود در این روش ها بر اساس مشخصات ظاهری که از هدف مورد نظر به دست می‌آید رهگیری را انجام می دهند.

* رهگیری به کمک قالب[[35]](#footnote-36): در این روش ابتدا یک‌سری اطلاعات از هدف مورد نظر که توسط یک قالب مستطیل شکل و یا بیضی شکل مشخص شده‌است استخراج می‌شود. سپس میزان شباهت این اطلاعات ذخیره‌شده با قسمت‌های مختلف فریم بعدی که ابعاد هر کدام از این قسمت ها برابر ابعاد قالب اولیه هدف مورد نظر است محاسبه می‌شوند. در نهایت قسمتی از تصویر که شباهت بیشتری به قالب اولیه و اطلاعات موجود در آن دارد به عنوان هدف شناخته‌شده و عملیات رهگیری صورت می‌پذیرد. در شکل‌های (2-17)و (2-18) دو نمونه از نتایج این الگوریتم‌ها را مشاهده می‌کنیم.





شکل(2-17): نتیجه رهگیری هدف به روش کرنل-قالب



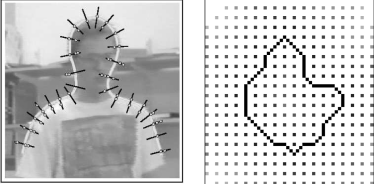
شکل(2-18): نتیجه رهگیری هدف به روش کرنل-قالب

همان‌طور که در شکل (2-17) و (2-18) مشاهده می‌کنید هدف مورد نظر به کمک ویژگی‌های ظاهری که در قالب اولیه هدف وجود دارد به خوبی تشخیص داده شده و رهگیری شده‌است. در شکل (2-18) مشخصات داخل قالب اولیه هدف را نیز که به صورت یک تصویر کد شده‌است، در کنار هر تصویر مشاهده می‌کنید. این واقعیت مبین آن است که چه قدر نحوه مدل‎سازی هدف و مشخصات ظاهری آن اهمیت بالایی در سیستم‌های رهگیری دارد.

* رهگیری به کمک یادگیری هدف از زوایا مختلف[[36]](#footnote-37): در روش قبلی اطلاعات هدف مورد نظر به صورت برخط و آنی به دست می‌آید. به همین علت این احتمال وجود دارد که در فریم بعدی مشخصات ظاهری هدف مورد نظر به خاطر تغییر مشخصات محیطی تغییر کند و به دنبال آن این مشکل به وجود می‌آید که هدف مورد نظر در فریم بعدی تشخیص داده نمی‌شود. راه‌حل این مشکل استفاده از الگوریتم های یادگیری است که در بخش تشخیص هدف به کمک یادگیری با نظارت بحث شد. الگوریتم هایی مانند ماشین بردار پشتیبان و یا حتی شبکه‌های عصبی و عمیق که به کمک مجموعه‌ای از داده‌های گوناگون مورد آموزش قرار می‌گیرند و از آن‌ها می‌توانیم برای تشخیص هدف به کمک مشخصات ظاهری و به دنبال آن رهگیری هدف استفاده کنیم. نمونه‌ای از این کار را در شکل (2-11) مشاهده کردیم. البته این شکل صرفا بخش تشخیص هدف را بیان می‌کند ولی خب مشخص است که با دنبال کردن همین هدف در فریم‌های متوالی عمل رهگیری نیز انجام می‌پذیرد، همان‌طور که ماهیت روش ردیابی کرنل همین است.

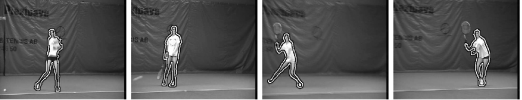
3-4-2-2-ردیابی سطح داخلی و خارجی[[37]](#footnote-38)

اشیا در یک تصویر ممکن است که دارای ساختارهای پیچیده باشند مانند دست یک انسان، سر یک انسان و یا حتی شانه‌های آن. به همین علت نمیتوانیم این نوع از اشیا و اهداف را صرفاً با یک هندسه ساده مدل سازی کنیم(شکل(2-19)). هدف از این نوع رهگیری این است که ناحیه ای که مشخص کننده هدف ما میباشد را در فریم های مختلف با توجه به مدل تولید شده در فریم قبلی تشخیص دهیم. مدل هدف تشخیص داده شده می‌تواند در قالب نمودار هیستوگرام رنگ سطح داخلی هدف باشد و یا اینکه شامل اطلاعات لبه‌های سطح خارجی از آن تصویر باشد. این نوع از روش ها را می توانیم به دو دسته کلی شامل ردیابی سطح خارجی[[38]](#footnote-39) و  تطابق هدف[[39]](#footnote-40) تشخیص داده شده تقسیم کنیم.



شکل(2-19): نحوه بیان هدف به کمک مدل سطح داخلی

* تطابق هدف تشخیص داده‌شده: این روش همانند روش رهگیری به کمک قالب که در بخش رهگیری کرنل داشتیم عمل میکند. به این ‌صورت که سطح داخلی مبین هدف تشخیص داده شده در فریم بعدی به کمک یک تابع هزینه معرفی شده جستجو می شود و قسمت‌هایی از تصویر و یا قسمتی از تصویر که بیشترین  مشابهت را با هدف موردنظر دارد به عنوان هدف تشخیص داده شده در نظر گرفته میشود و عمل رهگیری کامل می گردد.
* ردیابی سطح خارجی:در این روش بر خلاف روش قبلی که در جستجوی هدف مورد نظر با توجه به مدل قبلی بودیم، در هر فریم مجزا هدف مورد نظر به صورت سطح داخلی و یا خارجی تشخیص داده میشود و سپس مختصات های تشخیص داده شده در فریم های متوالی به عنوان مسیر حرکتی و یا حرکت جسم مورد نظر در نظر گرفته می شوند. دو نمونه‌ از نتایج این روش را در شکل (2-20) مشاهده می‌کنیم.





شکل(2-20): دو نمونه از نتایج روش ردیابی سطح داخلی

3-2- نتیجه‌گیری

در این فصل به صورت کامل روش‌های مختلف برای پیاده‌سازی سیستم‌های رهگیری و مراحلی که مورد نیاز است تا یک سیستم رهگیری پیاده‌سازی شود را با مورد بحث و بررسی قرار دادیم. به صورت کلی راه‌اندازی یک سیستم رهگیری را به چهار قسمت اصلی تقسیم کردیم. این مراحل شامل نحوه مدل‌سازی هدف مورد نظر، انتخاب ویژگی‌های ساختاری و ظاهری مهم برای هدف مورد‌نظر، تشخیص هدف در یک تصویر و در نهایت رهگیری‌آن بر اساس اطلاعات مراحل قبلی. در مرحله مدل‌سازی می‌توانستیم از مدل نقطه‌ای، هندسی و یا سطح داخلی استفاده کنیم. در مرحله انتخاب ویژگی‌ها، ویژگی‌هایی از جمله ساختار، رنگ، اطلاعات کناره تصویر و جریان نوری را می‌توانستیم انتخاب کنیم که مشخصات رنگی از جمله مهم‌ترین آن‌ها بود. در مرحله تشخیص هدف که یک مرحله بسیار ضروری می‌باشد می‌توانستیم از روش‌های پس‌زمینه، قسمت‌بندی،نقطه‌ای و روش‌های هوش مصنوعی استفاده کنیم. در نهایت و در مرحله آخر نوبت آن بود که هدف را رهگیری کنیم که این امر به کمک روش‌های ردیابی نقطه‌ای، ردیابی کرنل و ردیابی سطح داخلی و خارجی انجام می‌شوند که البته روش ردیابی کرنل امروزه از مهمترین و معروف‌ترین روش‌ها می‌باشد که در سیستم‌های مبتنی بر هوش‌مصنوعی نیز از این مورد استفاده می‌شود.

1. هوش مصنوعی و بینایی ماشین

1-3- مقدمه

پردازش تصویر دیجیتال شامل دست‌کاری تصاویر با استفاده از رایانه‌های دیجیتال است. در دهه‌های گذشته استفاده از آن به‌طور تصاعدی افزایش‌یافته است. کاربردهای پردازش تصویر دیجیتال از پزشکی گرفته تا سرگرمی و سنجش‌ازدور است. رشتۀ پردازش‌تصویر دیجیتال یک رشتۀ گسترده است که شامل تکنیک‌های پردازش سیگنال دیجیتال و همچنین تکنیک‌های خاص تصاویر است.یک تصویر را می‌توان به‌عنوان یک تابع در نظر گرفت f(x,y) از دو متغیر پیوسته x و y . برای پردازش دیجیتالی که باید از آن نمونه‌برداری کرد و به ماتریسی از اعداد تبدیل شود. ازآنجاکه رایانه، اعداد را با استفاده از دقت محدود نشان می‌دهد، برای نمایش دیجیتالی، این اعداد باید کمی شوند. پردازش‌تصویر دیجیتال شامل دست‌کاری اعداد دقیق محدود است. پردازش تصاویر دیجیتالی را می‌توان به چنددسته تقسیم کرد: بهبود تصویر، ترمیم تصویر، تجزیه‌وتحلیل تصویر و فشرده سازی تصویر. پردازش تصویر دارای روش های مختلفی می باشد که آن هار را می توانیم در دو گروه اصلی تقسیم بندی بکنیم. گروه اول روش هایی می باشند که به کمک یک سری روابط ریاضی پیاده سازی شده اند. در این روش ها الگوریتم های ریاضی را بر روی تصویر پیاده سازی کرده و خروجی مورد نظر را دریافت می کنند. به عنوان مثال به کمک این روش ها می توان فضای رنگی تصویر را عوض کرد و یا نویز[[40]](#footnote-41) تصویر را حذف کرد و یا حتی نویز به آن اضافه کرد. یکی از معروف ترین کتاب‌خانه ها در این زمینه که به زبان پایتون نوشته شده است، کتاب‌خانه open-cv می باشد. گروه دوم روش های مبتنی بر یادگیری عمیق می باشند که از آن ها با عنوان شبکه های عصبی نیز یاد می شود. تفاوت روش های این گروه در ساختار شبکه عصبی می باشد که استفاده شده است. امروزه شبکه های عصبی فراوانی برای پردازش تصویر معرفی شده است که هر کدام دارای ویژگی خاص می باشد. به عنوان مثال شبکه های Retina Net و YOLO مثال هایی از شبکه های عصبی برای پردازش تصویر می باشند که برای تشخیص اشیا[[41]](#footnote-42) از آن ها استفاده می شود. به علت این که در این پروژه از شبکه های عصبی برای قسمت پردازش تصویر استفاده شده است در ادامه تاریخچه ای از پردازش تصویر و شبکه های عصبی بیان می شود.

2-3- تاریخچه

اواخر سال 1980 بود که شبکه­ های عصبی در علم پزشکی ظهور کرده و به کار گرفته شدند. شبکه­ های عصبی که دارای دو نوع یادگیری با معلم[[42]](#footnote-43)و بدون معلم[[43]](#footnote-44) می­باشند، هر دو به عنوان راه حل ­هایی موفقیت آمیز در پزشکی مورد استفاده قرار گرفتند.تشخیص بیماری ای که امکان نتیجه­ گیری از داده­های فراوان به دست آمده از آن به صورت دستی وجود ندارد، از مهم­ترین کمک ­های شبکه عصبی به علم پزشکی می­باشد. شبکه عصبی از جمله تکنولوژی­ هایی است که به شکل­های مختلف به پشتیبانی از خدمات پزشکی می پردازد. شناخت تصویر در شبکه عصبی علاوه بر کمک به پیشرفت درمان، موجب کاهش هزینه­ های ناشی از درمان، ارتباط موثر تر میان اطلاعات به دست آمده از شرایط بیمار، ارتقاء سطح کیفی خدمات بهداشتی و… می­گردد.حوزه پردازش تصویر و شناخت تصویر در شبکه عصبی در علم پزشکی بازه گسترده ­ای از کاربرد های گوناگون نظیر تشخیص دیابت چشمی با استفاده از تصاویر گرفته شده از شبکیه چشم، بخش بندی تصویرهای ام آر آی تهیه شده به منظور تشخیص وجود یا عدم وجود تومور در مغز انسان و… را در بر می­گیرد.اگر چه در گذشته برخی از روش­های تشخیص بیماری بر اساس استخراج ویژگی­ ها و اطلاعات از روی تصاویر پزشکی و به صورت کاملا دستی و با صرف وقت و انرژی فراوان به وسیله متخصصان صورت می پذیرفت، اما امروزه روشی جدید با هدف تشخیص و طبقه بندی تصاویر پزشکی بدون آن که نیازمند استخراج ویژگی ­ها به صورت دستی باشد و با به کارگیری شناخت تصویر در شبکه عصبی پایه گذاری شده است. مهم­ترین مزیت شناخت تصویر در شبکه­ عصبی، توانایی این شبکه­ ها در استخراج خودکار ویژگی­ های تصویر با به کارگیری مفهوم یادگیری عمیق می­باشد. همین مزیت مهم موجب استفاده از شناخت تصویر در شبکه عصبی در سال­ های اخیر شده است.به عنوان مثال در تشخیص بیماری رتینوپاتی دیابتی که از آن تحت عنوان دیابت چشمی نیز یاد می­شود، رگ­ های شبکیه موجود در چشم بیمار در اثر بیماری دیابت به منظور خون رسانی بهتر نسبت به حالت معمولی خود گشادتر شده و تغییر شکل می­دهند. شناخت تصویر در شبکه عصبی به منظور استخراج ویژگی­ های شبکیه چشم مورد استفاده قرار گرفته و مراحل پیشرفت بیماری با استفاده از این تصاویر معین می­گردد. تشخیص درست پولیپ­ ها در طی تصویر برداری کلونوسکپی، تشخیص خودکار رگ گرفتگی­ های ریوی، تصاویر سی تی اسکن، تشخیص سرطان سینه، تشخیص غده­های لنفاوی با استفاده از تصاویر سی تی اسکن، تشخیص غده­های لنفاوی شکمی، تشخیص خودکار آناتومی بدن و… از دیگر مزایای استفاده از شناخت تصویر در شبکه عصبی می­باشد.شناخت تصویر در شبکه عصبی علاوه بر موارد فوق الذکر، در زمینه­ اندازه ­گیری و بخش ­بندی تصاویر موجود در زمینه پزشکی نظیر بخش بندی لوزالمعده در تصاویر به دست آمده از اسکن توموگرافی، بخش بندی تصاویر گرفته شده از مغز کودکان، بخش بندی غشاء عصبی در تصاویر الکترومیکروسکوپی گرفته شده، بخش بندی تصاویر گرفته شده از غضروف­ های زانو در اسکن­ های ام آر آی، اندازه ­گیری ضخامت لایه عروق و رگ­ ها و…. نیز بسیار کاربرد دارد.

همان گونه که پیش­تر نیز اشاره کردیم، شبکه عصبی و شناخت تصویر در شبکه عصبی ایده­ای جذاب و در عین حال قدیمی است. مفهوم اولیه شبکه عصبی را نخستین بار Frank Rosenblatt در مقاله­ای که در سال 1957 به چاپ رساند، تحت عنوان پرسپترون معرفی نمود. وی در سال 1958 موفق به طراحی سیستم اولیه آنالیز یک تصویر ساده و شناسایی اشکال هندسی شد. اصلی ­ترین هدف Rosenblatt ساخت سیستمی پر کاربرد به منظور بخش بندی یا کلاس بندی تصاویر بود. این در حالی است که او علاقه فراوانی به درک بهتر کارکرد مغز از این راه بود. بعدها این علاقه Rosenblatt طرفداران زیادی را به خود جذب نمود! هر نورون موجود در یک شبکه عصبی تابعی ریاضیاتی است که مجموعه­ای از ورودی­ های دارای وزن را پردازش می­کند. لازم به ذکر است که در صورت بالا بودن وزن سیناپس، آن سیناپس خروجی نورون را به مقدار قابل توجه ­تری تحت تاثیر قرار می­دهد. خروجی نورون به تابعی غیر خطی تحت عنوان تابع فعال ارسال می­شود. این تابع توانایی مدل سازی پدیده­ های غیر خطی پیچیده را به شبکه می­دهد.یک شبکه با استفاده از تنظیم نمودن وزن سیناپس­ ها و بازخورد دریافت شده از عملکرد سیستم نسبت به ورودی­ های گوناگون موفق به یادگیری مدل می­گردد. کلاس بندی صحیح تصاویر توسط شبکه موجب افزایش مقادیر وزن­ های موثر در جواب و کاهش دیگر وزن­ها می شود. در صورتی که کلاس بندی تصویر به اشتباه صورت بپذیرد، کلیه وزن ­ها برعکس خواهند شد. نخستین شبکه ­های عصبی بدین شکل موفق به یادگیری شدند. در ادامه ساختار کلی شبکه های عصبی را بررسی کرده و روند آموزش و استفاده از این شبکه ها را بیان می کنیم.

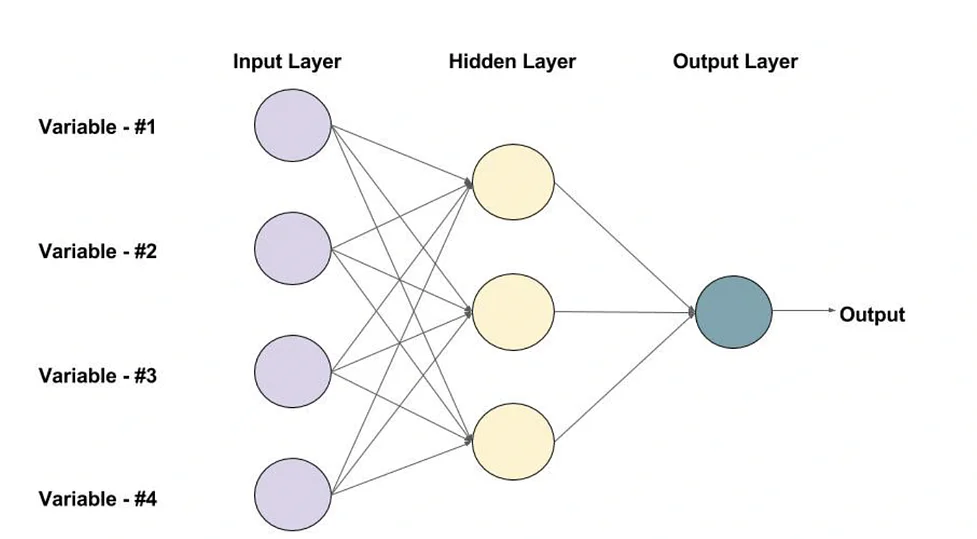
3-3- ساختار

همان طور که در قسمت تاریخچه نیز به آن اشاره شد ساختار شبکه های شبیه به ساختار مغز انسان می باشند و همانند آن از یک سری سلول5 تشکیل شده اند که هر کدام از سلول ها در واقع بیانگر یک رابطه ریاضی می باشد. در شکل (3-1) ساده ترین حالت یک سلول شبکه عصبی را مشاهده می کنیم.



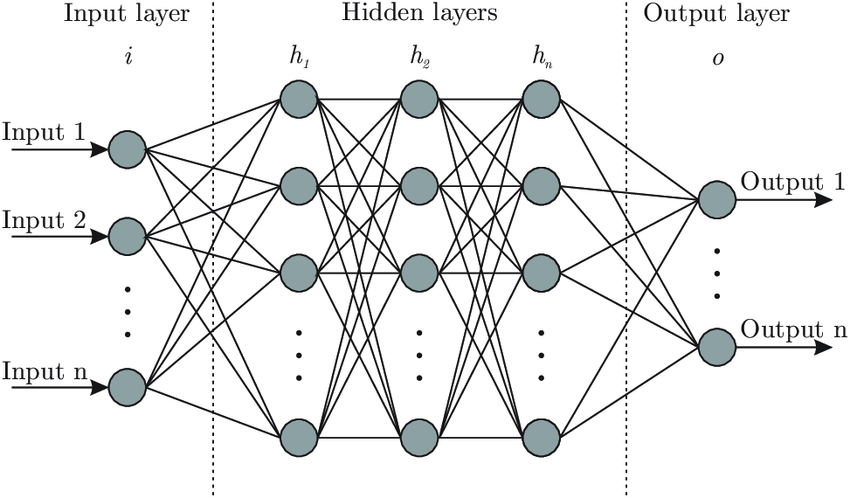
شکل(3-1): تصویر نمونه از نورون‌های شبکه عصبی

همانطور که در شکل (3-1) مشاهده می کنیم هر سلول عصبی از سه قسمت اصلی تشکیل شده است: وزن های ورودی یا همان پارامتر های یادگیری (این وزن ها یا پارامتر ها حین آموزش شبکه تغییر می کنند)، عملگر جمع کننده و تابع فعال ساز. عملکرد هر سلول بدین گونه است که ورودی شبکه در وزن های سلول ضرب شده و وارد سلول می شود، سپس در سلول مقادیر با هم جمع شده و در نهایت از یک تابع فعال ساز عبور داده می شود. هدف از تابع فعال ساز ایجاد یک رابطه غیر خطی می باشد، همان طور که میدانیم داده های ورودی می توانند دارای ویژگی های خطی و یا غیر خطی باشند. حال برای تشکیل یک شبکه عصبی باید تعداد زیادی از تک سلول ها را با یک ساختار مناسب کنار هم قرار دهیم. در شکل (3-2) ساده ترین حالت کنار هم قرار گرفتن سلول ها را مشاهده می کنیم.



شکل(3-2): ساده‌ترین حالت شبکه‌های عصبی

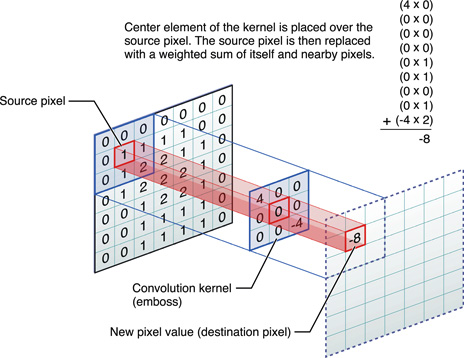
با توجه به شکل (3-2) هر شبکه عصبی از یک لایه ورودی تشکیل شده است که بیان گر داده های ورودی است. با توجه به ابعاد داده های ورودی تعداد سلول های لایه ورودی متغیر می باشد. دقت کنیم که سلول های لایه ورودی دارای پارامتر یادگیری نمی باشند و فقط مبین داده های ورودی می باشند. بعد از لایه ورودی لایه های پنهان شبکه را مشاهده می کنیم که وظیفه بسیار مهم استخراج ویژگی ها را بر عهده دارند. سلول های این لابه دارای پارامتر های یادگیری(وزن های ورودی) می باشند که این پارامتر ها حین آموزش شبکه تغییر می کنند. تعداد لایه های پنهان با توجه به ساختار و کاربرد شبکه می تواند متغیر باشد. در نهایت بعد از لایه های پنهان لایه خروجی شبکه را داریم که با توجه به نوع خروجی مورد نظر تعداد سلول های آن می تواند متفاوت باشد. در شکل (3-3) ساختار یک شبکه عصبی با تعداد لایه های پنهان بیشتر از یک را مشاهده می کنیم.



شکل(3-3): ساختار کامل شبکه‌های عصبی

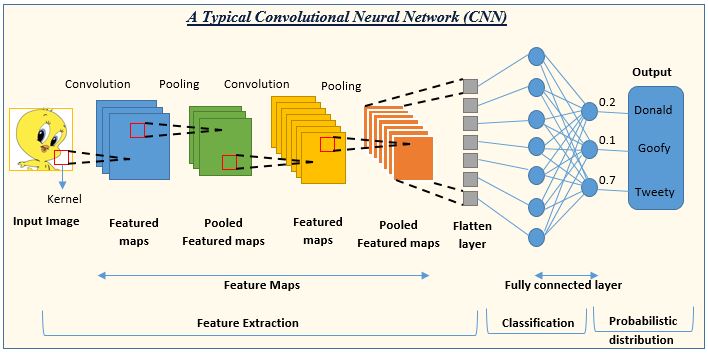
با توجه به شکل (3-3) و توضیحات قبلی در مورد سلول ها و لایه های شبکه می توان عملکرد یک شبکه عصبی را بدین گونه توصیف کرد: در ابتدای هر شبکه یک لایه مربوط به داده های ورودی داریم که متناسب با بعد[[44]](#footnote-45) داده های ورودی، تعداد سلول های آن متفاوت می باشد. یک نکته بسیار مهم در مورد سلول های لایه ورودی این است که این سلول ها فقط بیانگر مقدار داده ورودی می باشند و نماینده هیچ گونه رابطه ریاضی نیستند و همچنین فاقد پارامتر های یادگیری می باشند. در ادامه لایه های پنهان قرار دارند که میتوان گفت مهمترین بخش شبکه های عصبی را تشکیل می دهند.در هر کدام از این لایه ها متناسب با ساختار شبکه تعداد سلول های متفاوتی قرار می گیرند. وظیفه این لایه ها استخراج ویژگی های داده های ورودی می باشد. می توان گفت اساس کار شبکه های عصبی، همین لایه های پنهان و ویژگی هایی است که در این لایه ها استخراج می شوند و بر اساس آن ها خروجی مورد نظر در لایه آخر ساخته می شود. بعد از لایه های پنهان لایه خروجی قرار دارد که همانند لایه های پنهان دارای پارامتر های یادگیری می باشند ولی تعداد سلول های موجود در آن وابسته به بعد داده های خروجی متغیر می باشد.تا این مرحله ساختار کلی شبکه های عصبی را مورد بررسی قرار دادیم، ولی با توجه به این که داده های مورد استفاده ما در این پروژه از نوع تصویر می باشند در ادامه ساختار شبکه های عصبی که متناسب با وظایف[[45]](#footnote-46) پردازش تصویر می باشد را بررسی می کنیم.

همان طور که می دانیم هر تصویر از سه بعد طول، عرض و کانال تشکیل شده است. هر تصویر دارای یک طول و عرض مشخصی می باشد و تعداد کانال های آن عموما سه می باشد که اصطلاحا به آن RGB می گویند. برای پردازش داده های تصویری در شبکه های عصبی به جای لایه های تک بعدی از سلول ها(همانند شکل(3-3)) از لایه های با ساختار دو بعدی که سلول ها در دو بعد قرار گرفته اند استفاده می شود.ماهیت این سلول ها دقیقا مشابه ماهیت سلول های ساختار تک بعدی می باشد.اصطلاحا به این لایه های سلولی دو بعدی، کرنل[[46]](#footnote-47) می گویند.این کرنل ها بر روی تک تک پیکسل های تصویر ورودی اعمال می شوند و یک تصویر جدید را تولید می کنند در شکل (3-4) یک نمونه از این کرنل ها را مشاهده می کنیم.



شکل(3-4): نمونه‌ای از کرنل‌های مورد استفاده در پردازش تصویر

با استفاده از این کرنل ها می توان تصویر با ابعاد و کانال های متفاوتی تولید کرد. البته دقت کنیم معمولا تغییر ابعاد تصویر اهمیت چندانی ندارد و در شبکه های عصبی تعداد کانال های تصویر را به کمک تعداد کرنل هایی که تعریف می کنند تغییر می دهند و بدین گونه ویژگی های داده های ورودی را استخراج می کنند.مهم ترین نکته در مورد کرنل ها این است که مقادیر درایه های این کرنل ها به عنوان پارامتر یادگیری در شبکه های عصبی تعریف می شوند و حین آموزش شبکه این پارامتر ها تغییر می کنند تا به بهترین مقدار خود دست پیدا کنند و بدین گونه است که با بهینه شدن مقادیر این کرنل ها میتوانیم ویژگی های مهم داده های ورودی را استخراج کرده و از آن ها استفاده کنیم. به این گونه لایه ها که از کرنل ها ساخته شده اند لایه های کانوولوشنی[[47]](#footnote-48) می گویند و همچنین به شبکه های عصبی که از این لایه ها ساخته شده اند، شبکه های عصبی کانوولوشنی[[48]](#footnote-49) گفته می شود. در شکل (3-5) یک نمونه کامل از این شبکه های عصبی کانوولوشنی را مشاهده می کنیم.



شکل(3-5): نمونه کامل شبکه عصبی کانوولوشنی

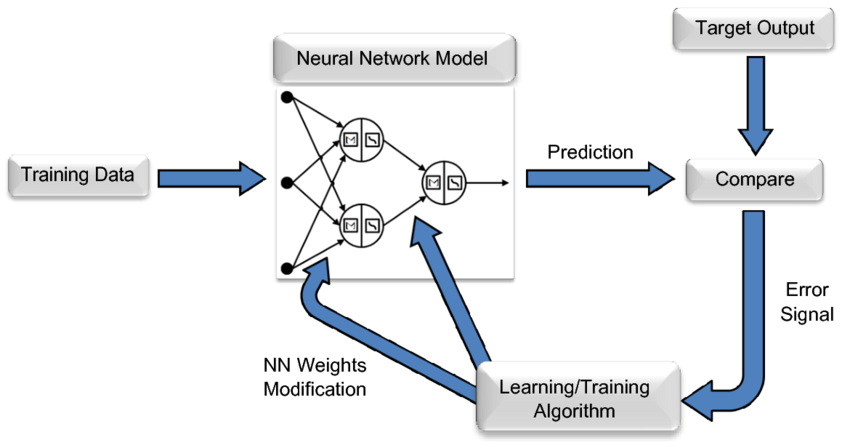
همان طور که در شکل (3-5) مشاهده می‌کنیم به کمک لایه های کانوولوشنی، لایه های پنهانی را تشکیل می دهیم تا ویژگی های داده ورودی که تصویر می باشد استخراج شوند. لایه ورودی همانند لایه ورودی شبکه های غیر کانوولوشنی همان داده ورودی(تصویر) می باشد و فقط متناسب با بعد تصویر تعداد سلول های آن می تواند متغیر باشد. در مورد لایه خروجی نکته ای که در شبکه های کانوولوشنی وجود دارد این است که با توجه به خروجی مطلوب و مورد نظر لایه خروجی می تواند شامل کرنل برای خروجی تصویر و یا شامل لایه های تک بعدی شبکه های معمولی برای خروجی عدد و یا غیر تصویر باشد. در ادامه با نحوه آموزش شبکه های عصبی آشنا می شویم و توضیح داده می شود که چگونه پارامتر های یادگیری تنظیم می شوند.

4-3- آموزش شبکه

در مورد آموزش شبکه های عصبی دو قسمت اصلی وجود دارد که شامل ساختار شبکه و داده می باشد.در مورد ساختار شبکه در قسمت های قبل توضیح داده شد.داده هایی که از آن ها برای آموزش استفاده می شود شامل دو قسمت می باشند: داده های ورودی و داده های خروجی. ابتدا داده های ورودی وارد شبکه می شوند و شبکه یک خروجی تولید می کند، خروجی تولید شده با خروجی واقعی مقایسه شده و بر اساس یک الگوریتم ریاضی به نام پس انتشار خطا[[49]](#footnote-50) و خطای تولیدی شبکه پارامتر های شبکه تنظیم می شوند.اساس کار الگوریتم پس انتشار خطا این می باشد که از خطای شبکه نسبت به هر کدام از پارامتر های شبکه که به آن ها اصطلاحا وزن های شبکه می گویند مشتق می گیریم و گرادیان آن را حساب می کنیم، سپس با یک ضریب کوچک(بین 0 و 1) این مقدار را از مقدار قبلی پارامتر کم می کنیم. بدین گونه پارامتر های شبکه به گونه ای تنظیم می شوند که به کمترین مقدار خطا در شبکه دست پیدا کنیم. رابطه (3-1) بیانگر رابطه ریاضی الگوریتم پس انتشار خطا می باشد.

(1-3) 

در شکل (3-6) ساختار بلوک دیاگرامی آموزش شبکه را مشاهده می کنیم. با توجه به این شکل ماهیت الگوریتم پس انتشار خطا را در روند آموزش شبکه به خوبی مشاهده می کنیم.



شکل(3-6): ساختار بلوک‌دیاگرامی آموزش شبکه

تا این قسمت ساختار شبکه های عصبی را بررسی کرده و روند آموزش آن را توضیح دادیم. در ادامه دو مورد از ابزار های پردازش تصویری که در روند پروژه از آن استفاده کردیم را توضیح خواهیم داد.

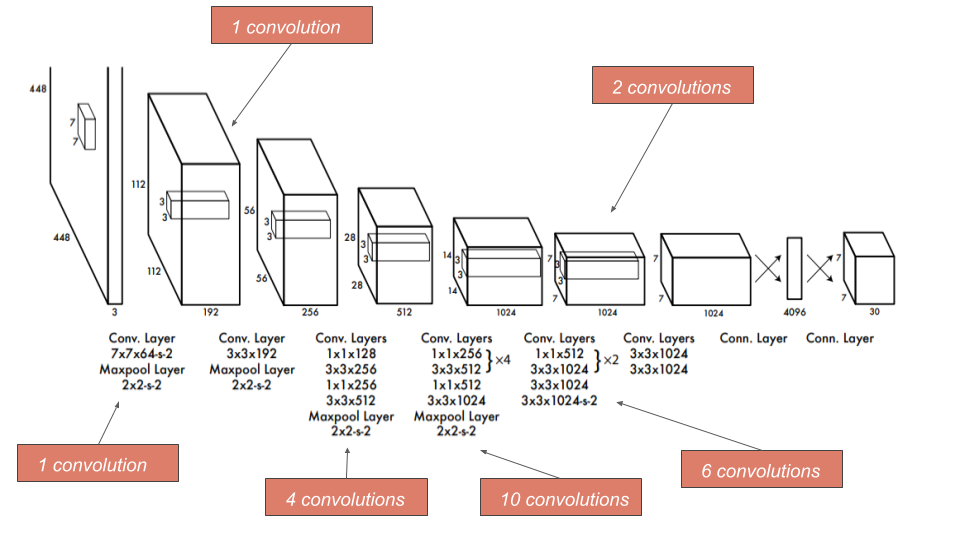
با توجه به موضوع و هدف پروژه ما باید از ابزار پردازش تصویری استفاده کنیم که قادر به انجام خواسته های ما باشد. در این راستا به دو نوع ابزار پردازش تصویر نیازمندیم. اولین مورد ابزاری است که قادر به تشخیص یک شئ خاص باشد که اصطلاحا به آن Object Detection گفته می شود. و دومین مورد ابزار پردازشی که بتواند ویژگی‌های ظاهری موجود در تصویر را استخراج کرده و رمزگذاری کند که اصطلاحا به آن Feature Extraction گفته می‌شود. بر همین اساس 2 ابزار پردازش تصویری شامل شبکه عصبی YOLO و الگوریتم HAAR Cascade در کتابخانه OpenCV استفاده شده‌است که وظیفه تشخیص هدف را دارا می‌باشند که در نهایت به علت نتیجه بسیارخوبی که شبکه عصبی YOLO داشت از این شبکه عصبی برای پروژه استفاده شد. و همجنین از شبکه عصبی OS Net برای استخراج ویژگی های یک تصویر استفاده شده است. در ادامه به معرفی این ابزار‌ها و شبکه‌های عصبی می‌پردازیم.

* الگوریتم HAAR Cascade در کتابخانه OpenCV

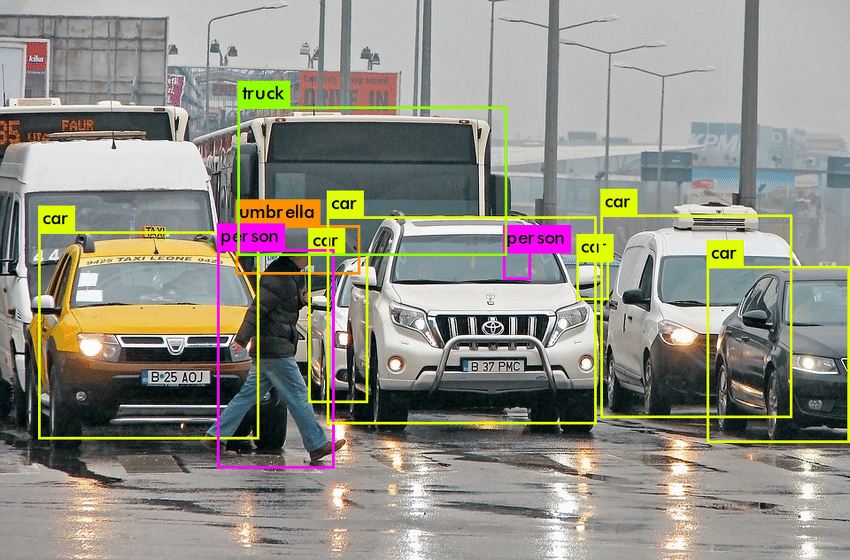
تشخیص اجسام با استفاده از الگوریتم HAAR Cascade، یک روش بسیار موثر است که اولین بار توسط Paul Viola و Michael Jones در سال ۲۰۰۱ مطرح شد. این الگوریتم بر پایه یادگیری ماشین بوده و بدین صورت عمل می کند که با استفاده از تصاویری که چهره در آنها وجود دارد (تصاویر مثبت) و تصاویری که چهره در آنها وجود ندارد (تصاویر منفی) تابعی را آموزش می دهد تا بتواند مولفه هایی مانند وجود فاصله بین چشم ها را پیدا کند. این الگوریتم منحصر به تشخیص چهره نیست؛ و بیشتر یک آموزش دهنده است و می توان با آموزش دادن تابع دلخواه خودتان هر شی مانند ماشین، میز، مداد و … را شناسایی کنید. OpenCV مجموعه ای از توابع از پیش آموزش دیده را برای تشخیص چهره، چشمان، لبخند و … در خود دارد که می توان به سادگی از آنها استفاده کرد. مشکل اصلی این الگوریتم این بود که برای تشخیص شئ مورد نظر حتما باید شئ بدون هیچ مانعی(هر چند کوچک) در مقابل دوربین قرار می گرفت و اگر یک مانع بسیار کوچک در مقابل آن باشد دیگر قادر به تشخیص نیست و این مورد باعث کم شدن دقت انجام کار میشد. با استفاده از شبکه عصبی YOLO توانستیم مشکل موجود در الگوریتم HAAR را حل کنیم. در ادامه به توضیح شبکه عصبی YOLO می پردازیم.

* شبکه عصبی YOLO

شبکه عصبی YOLO یک شبکه عصبی عمیق با لایه های کانوولوشنی می باشد. همان طور که در قسمت ساختار شبکه های عصبی آشنا شدیم از لایه های کانوولوشنی برای پردازش داده های تصویری استفاده می شود. همچنینی YOLO یک شبکه عمیق[[50]](#footnote-51) می باشد.اصطلاحا به شبکه های عصبی که از لایه های زیادی تشکیل شده و دارای پارامتر های بسیار زیادی می باشند شبکه های عمیق می‌گویند. در شکل (3-7) ساختار شبکه YOLO را مشاهده می‌کنیم. این شبکه عصبی عمیق بر روی مجموعه داده‌های بسیار بزرگی به نام COCO آموزش دیده‌است و می‌تواند گستره بسیار بزرگی از اشیا را تشخیص دهد. البته با توجه به نوع پروژه و نوع هدفی که مایل به تشخیص آن هستیم می‌توانیم این شبکه را دوباره آموزش دهیم تا بتواند شئ مورد نظر ما را تشخیص دهد. مزایای بسیار خوب این شبکه سریع بودن آن و دقت بالای آن می‌باشد. یک نمونه از نتایج این شبکه در تشخیص هدف را در شکل (3-8) مشاهده می‌کنید.



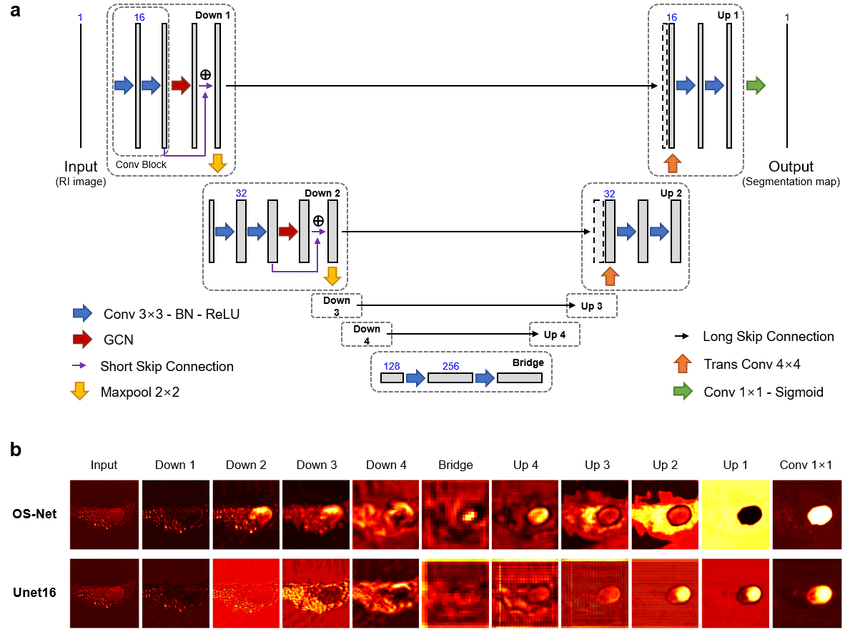
شکل(3-7): ساختار شبکه عصبی YOLO



شکل(3-8): خروجی شبکه YOLO برای تشخیص اهداف

* شبکه عصبی OS Net

در روند پروژه یکی از مهم‌ترین بخش‌های موجود استخراج ویژگی از تصویر می‌باشد. علت این امر نیز این است که به کمک استخراج ویژگی‌ها می‌توانیم هدف مورد نظر را در هر مرحله به عنوان هدف مورد نظر از بقیه اهداف تمییز دهیم. این عمل نه تنها در سیستم‌های رهگیری بلکه در بسیاری دیگر از سیستم‌ها از جمله سیستم‌های تشخیص هویت و امنیتی نیز کاربرد دارد. OS Net شبکه عصبی می‌باشد که به کمک آن می‌تواینم یک سری اطلاعات و ویژگی ها از تصویر استخراج کنیم. لایه‌های آخر این شبکه عصبی به گونه‌ای قرار گرفته‌اند که یک ماتریس با ابعاد مشخص را که مبین اطلاعات تصویر است در اختیار ما قرار می‌دهد. توجه کنیم که این شبکه عصبی نیز همانند بقیه شبکه‌ها بر روی مجموعه داده‌های بسیار بزرگی آموزش داده شده است و بنابراین دقت بالایی دارد. در شکل (3-9) ساختار این شبکه را مشاهده می‌کنیم.



شکل(3-9): ساختار شبکه عصبی OS Net

در شکل (3-9) ساختار شبکه OS Net و لایه هایی که در آن استفاده شده‌است را مشاهده می‌کنید. همچنین در قسمت پایینی تصویر یک نمونه از خروجی شبکه را نشان می‌دهد. قسمت پایین سمت چپ ورودی شبکه بوده و قسمت پایین سمت راست خروجی شبکه است که به صورت تصویر(ماتریس) اطلاعات و ویژگی های تصویر ورودی در آن ذخیره شده است.

5-3- نتیجه‌گیری

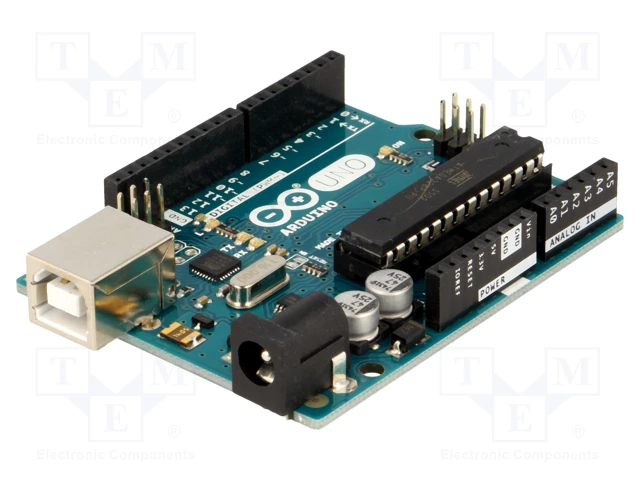
در این فصل کلیت پردازش تصویر و شبکه‌های عصبی را بررسی کردیم. از ساده‌ترین حالت شبکه‌های عصبی که شامل فقط یک لایه تا شبکه‌های عصبی مخصوص پردازش تصویر که شامل لایه‌های کانوولوشنی هستند را مورد بحث و بررسی قرار دادیم. و در نهایت ابزار‌های پردازشی مورد استفاده در این پروژه که شامل روش‌های پردازش تصویری ریاضی و همچنین شبکه‌های عصبی بود را معرفی کردیم.

1. سخت‌افزار
   1. برد های پردازشی

در این قسمت به بررسی برد های پردازشی می پردازیم که برای اجرای این پروژه از آن ها استفاده شده است و یا می توانیم از آن ها در قسمت های مختلف پروژه استفاده کنیم. متداول ترین برد های پردازشی که از آن ها در پروژه های مختلف استفاده می شود شامل برد های آردوئینو1، برد های رزپری پای2 و برد های جتسون نانو3 می باشد. این برد ها از نظر توان پردازشی و قابلیت های پردازشی متفاوت هستند. در ادامه با هر کدام از برد ها بیشتر آشنا شده و توضیح خواهیم داد که هر کدام از برد ها در کدام قسمت پروژه قابلیت استفاده دارند.

* + 1. آردوینو

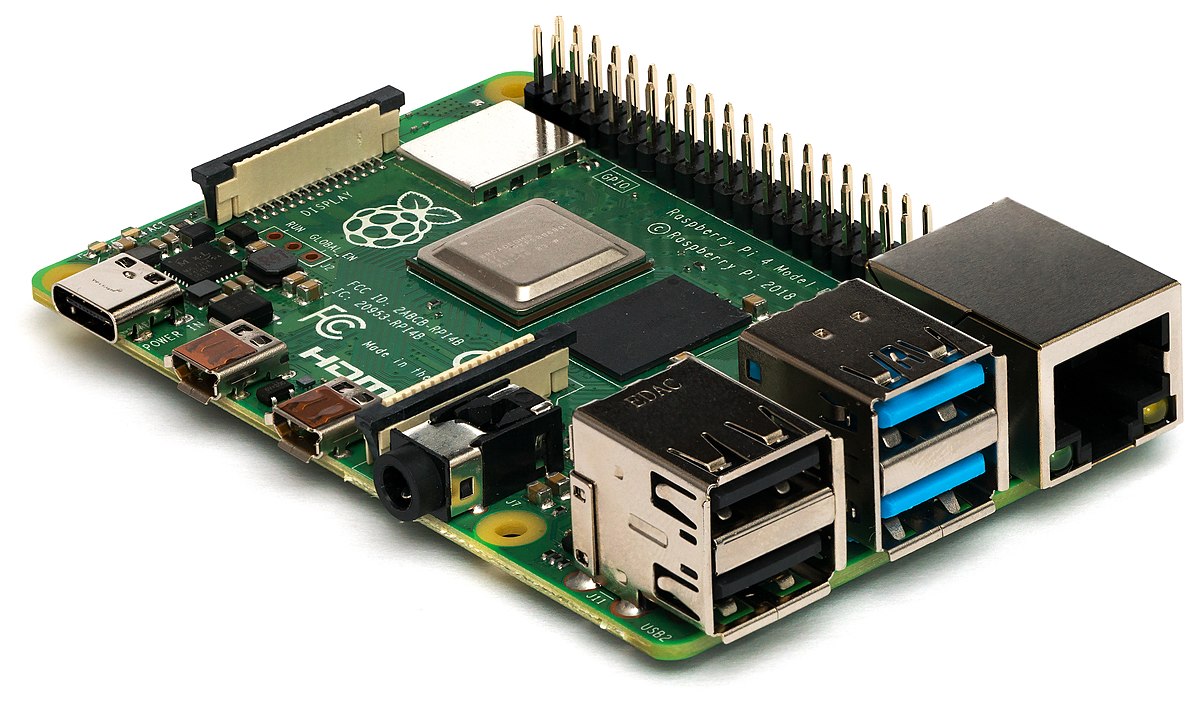
آردوینو یک برد متن‌باز برای نمونه‌سازی می‌باشد و بر اساس سخت‌افزار و نرم‌افزار منعطف و ساده پایه‌ریزی و طراحی شده است. آردوینو می‌تواند محیط اطراف را با استفاده از سنسورهای مختلف حس کند. آردوینو  می‌تواند بر محیط اطراف خود با استفاده از لامپ‌ها، موتورها و سایر محرک‌ها اثر بگذارد. میکروکنترل بکار رفته بر روی برد آردوینو بر اساس زبان برنامه‌نویسی آردوینو و محیط ویژه کدنویسی آن برنامه‌ریزی شده است و برای کدنویسی به هیچ نرم‌افزار جانبی و یا کامپایلر دیگری نیاز نیست. پروژهای آردوینو می‌توانند به صورت مستقل و یا مرتبط با سایر نرم‌افزارهای کامپیوتر شخصی باشد. برای مثال می‌توان فرمان‌ها را برای کنترل موتورها از طریق پورت یواس‌بی ارسال نموده و یا داده‌های سنسورها را نیز از همین پورت دریافت کرد.توان پردازشی آردوئنو به گونه ای است که نمیتوان تمام پردازش های پروژه را بر روی آن انجام داد و با توجه به قابلیت بالایی که آردوئنو در تعامل با سنسور ها و عملگر های متفاوت دارد از این برد می توانیم برای کنترل موتور های به کار رفته در پروژه استفاده کنیم. عموما در پروژه های متوسط به بالا برای پردازش های سنگین از برد های پردازشی قوی تر استفاده می‌کنند و از برد آردوئینو صرفا به عنوان یک واسط برای ارتباط با سنسور ها و عملگر ها استفاده میکنند. در قسمت های بعدی خواهیم دید که چگونه در این پروژه نیز ما از برد آردوئینو برای کنترل سروو موتور ها استفاده کرده ایم. در بین گونه های متفاوت برد های آردوئینو، آردوئینو یونو4 به علت قابلیت های بیشتر و گوناگون دارای محبوبیت بیشتری می باشد. در شکل (1-4) برد یونو را مشاهده می کنیم.



برد آردئوینو یون

* + 1. رزبری‌پای

رزبری پای رایانه کوچکی است که از سال 2006 در حال توسعه است و قطعات آن روی یک مادربرد به اندازه کارت بانکی سوار شده و Raspbian را اجرا می‌کند که یک نسخه اختصاصی از سیستم عامل لینوکس است که اختصاصاً برای این رایانه طراحی شده است.رزبری پای کاربردهای محاسباتی ابتدایی اداری، بازی‌های سطح پایین، دسترسی به اینترنت و ایمیل، بازپخش ویدئو و بسیاری قابلیت‌های دیگر دارد که به طور معمول از یک رایانه در قرن بیست و یکم انتظار می‌شود. رزبری پای همه این امکانات را با تعداد بسیار کمی از قطعات از جمله یک پردازنده ARM و قیمت بسیار پایین عرضه می‌کند.در مقایسه با برد آردوئینو، برد رزپبری‌پای دارای توان پردازشی بسیار بالایی می باشد. همچنین این نوع از برد ها به علت دارا بودن سیستم عامل توانایی این را دارند که برنامه های مختلف به زبان های برنامه نویسی مختلف از جمله پایتون را اجرا کنند. بنابراین می توانیم قسمت هایی از پروژه که دارای پردازش های سنگین می باشند را بر روی این برد انجام دهیم. در شکل (2-4) نمونه ای از برد های رزبری‌پای را مشاهده می کنیم.

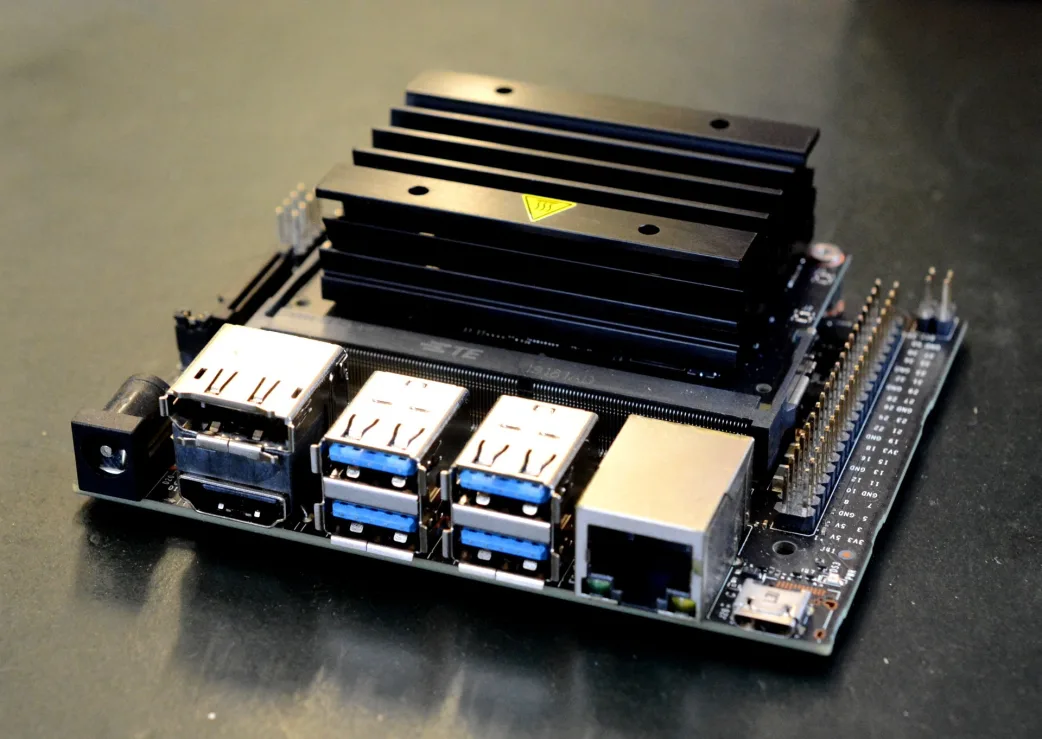


برد رزپبری‌پای

با این که برد رزپبری‌پای توان پردازشی بالایی دارد ولی بیشتر از یک حد معینی نمیتواند پاسخگوی پردازش ها باشد، به خصوص پردازش های سنگینی همانند پردازش تصویر در شبکه های عصبی و عمیق. به همین دلیل در ادامه برد های قوی تری از جمله جتسون نانو را معرفی می کنیم.

* + 1. جتسون نانو

این نوع از برد ها، کامپیتور های در اندازه کوچک و قدرتمند هستند که دارای کارت گرافیکی بوده و برای پردازش های شبکه های عصبی به خصوص پردازش های تصویری ساخته شده اند. همانند برد های رزپرب‌پای برو روی این برد ها نیز می توان سیستم عامل پیاده سازی کرد و به کمک آن برنامه های مختلف به زبان های مختلف اجرا کرد.در واقع این نوع از برد ها بهترین گزینه برای این پروژه می باشند که به کمک آن می توان تمام پردازش های مورد نیاز از جمله بخش هوش مصنوعی و بخش کنترلی متلب را بر روی آن اجرا کرد.مهمترین مشکلی که برای استفاده از این برد ها وجود دارد قیمت بالای آن می باشد. در شکل (3-4) یک نمونه از این برد ها را مشاهده می کنیم.

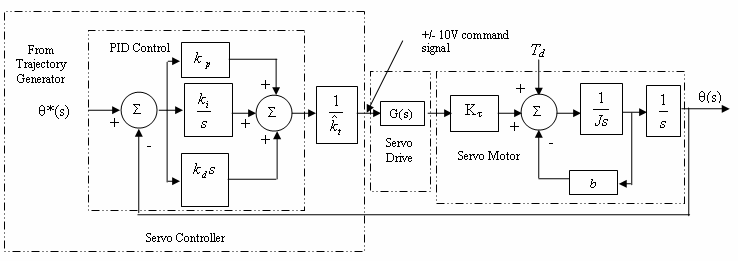


برد جتسون نانو

تا این مرحله برد های پردازشی که می توان در پروژه استفاده کرد و یا از آن ها استفاده شده و یا حتی تست هایی به کمک این برد ها صورت گرفته معرفی شدند. در ادامه کار و توضیحات اصلی پروژه خواهیم دید که به علت مشکلاتی از جمله قیمت بالای بعضی برد ها و یا توان پردازشی کم بعضی دیگر از برد ها قسمت هایی از پردازش های مورد نیاز بر روی لپتاپ صورت می گیرد.

* 1. عملگرها5:

در این قسمت در مورد عمگری که در پروژه از آن استفاده شده است توضیح داده و مشخصات فنی آن را بیان می کنیم.با توجه به نیاز پروژه برای کنترل زاویه دید دوربین نیاز بود تا از موتور های DC برای حرکت ساختار دینامیکی استفاده کرد. یکی از بهترین گزینه ها در این زمینه موتور های سروو6 می باشند. با کمک این نوع از موتور های DC می توان میزان حرکت موتور را کنترل کرده و در نتیجه می توانیم زاویه دید دوربین را به کمک آن کنترل کنیم. مزیتی که سروو موتور دارا می باشد این است که داخل آن کنترل کننده PID موجود می باشد و یک حلقه کنترلی بسته شده است که به کمک آن میتوان به صورت دقیقی موقعیت موتور و میزان چرخش آن را تنظیم کرد. در شکل (4-4) حلقه داخلی سروو موتور ار مشاهده می کنیم.



حلقه داخلی سروو موتور

* MG-995: نوع سروویی که در این پروژه از آن استفاده شده است MG-995 می باشد که در جدول (4-1) مشخصات این سروو موتور را مشاهده می کنیم.

جدول((3-1)): مشخصات سرووموتور MG-995

|  |  |  |  |
| --- | --- | --- | --- |
| زاویه چرخش | گشتاور (kg-cm)stall | سرعت چرخش(sec/60') | ولتاژ کاری(v) |
| 180 | 13 | 0.2 | 7.2 – 4.8 |

* S3003 : نوع سروویی که در این پروژه از آن استفاده شده است S3003 می باشد که در جدول (4-2) مشخصات این سروو موتور را مشاهده می کنیم.

مشخصات سرووموتور S3003

|  |  |  |  |
| --- | --- | --- | --- |
| زاویه چرخش | گشتاور (kg-cm)stall | سرعت چرخش(sec/60') | ولتاژ کاری(v) |
| 180 | 3.2 | 0.23 | 7.2 – 4.8 |

* SG-5010 : نوع سروویی که در این پروژه از آن استفاده شده است SG-5010 می باشد که در جدول (4-3) مشخصات این سروو موتور را مشاهده می کنیم.

جدول((3-2)): مشخصات سرووموتور sg-5010

|  |  |  |  |
| --- | --- | --- | --- |
| زاویه چرخش | گشتاور (kg-cm)stall | سرعت چرخش(sec/60') | ولتاژ کاری(v) |
| 180 | 6.5 – 5.5 | 0.20 – 0.16 | 6.6 – 4.8 |

سرووموتور مورد نظر دارای سه سیم می باشد که دو سیم از این سه سیم برای تغذیه و زمین مدار سرووموتور بوده و سیم سوم برای ارسال دستورات6 به سرووموتور می باشد. در شکل (5-4) شکل این سرووموتور را مشاهده می کنیم.



سرووموتور sg-5010

* 1. سنسور

می دانیم که در هر حلقه کنترلی نیاز به وجود حداقل یک سنسور هست تا به کمک آن مقدار پارامتر مورد‌نظر سنجیده شود و به همراه نقطه مرجع7 به کنترل کننده داده شود تا سیگنال خطا تولید شده و به دنبال آن سیگنال کنترلی تولید شود. با توجه به این که هدف پروژه دنبال کردن یک شیئ می باشد و داده های موجود در حلقه کنترلی از جنس تصویر می باشند، بهترین و مناسب‌ترین گزینه برای انتخاب سنسور، دوربین می باشد.باید دوربینی انتخاب شود که حداقل کیفیت لازم را برای پردازش های بعدی در شبکه عصبی را داشته باشد. انتخاب ما برای این پروژه دوربین وبکم GP2135 می باشد که مشخصات آن را در جدول (4-4) مشاهده می کنیم. با توجه به این که درگاه ارتباطی این دوربین USB می باشد به راحتی می تواند به برد های پردازشی معرفی شده متصل شود و یا اگر در صورت نیاز پردازش ها به وسیله لپتاپ انجام شود، یه لپتاپ متصل شود.

مشخصات دوربین GP2135

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| حسگر تصویر | رزولوشن تصویر | رزولوشن فیلم | کیفیت فیلم برداری | درگاه ارتباطی |
| CMOS | 3 MP | 1080p Full HD | Full HD | USB |

همچنین اندازه دوربین انتخاب شده به گونه ای می باشد که به راحتی می توان آن را بر روی ساختار دینامیکی که در مرحله بعد معرفی خواهیم کرد سوار کرد.

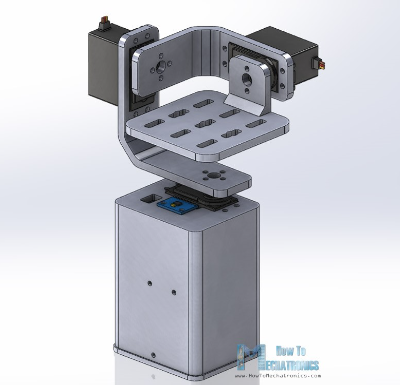
* 1. ساختار دینامیکی

تا این قسمت در مورد برد های پردازشی، سنسور و عملگر‌ها توضیحات لازم را داده ایم. آخرین قسمت از سخت افزار که سایر قسمت ها را در کنار هم قرار می دهد و یک سیستم جامع به دست می آید، ساختار دینامیکی می‌باشد. با توجه به نیاز پروژه که لازم هست یک ساختار دینامیکی دو درجه آزادی داشته باشیم لازم بود تا یک ساختار دینامیکی ایجاد می شد که این قابلیت را به ما میداد. بر این اساس از یک ساختار معروف به گیمبال8 استفاده شده است. در حالت کلی، گیمبال ساختاری است که به کمک آن می توان بازوهای مکانیکی ایجاد کرد که با قرار دادن تعدادی از این بازو ها در کنار هم می توان ساختار چند درجه آزادی ایجاد کرد.گیمبال دارای ساختار های متفاوتی می باشد که با توجه به نیاز پروژه می توان از هر کدام از این ساختار ها استفاده کرد. در شکل(6-4) ساختار یک گیمبال 3 درجه آزادی را مشاهده می کنیم که در سه محور Pitch , Yaw, Roll قابلیت حرکت دارد.



ساختار یک گیمبال 3 درجه آزادی

ساختاری که ما در این پروژه از آن استفاده کردیم نیز دارای سه درجه آزادی می باشد، اما با توجه به این که ما در این پروژه صرفا به دو درجه آزادی نیاز داشتیم، فقط از دو محور آن استفاده کرده ایم. در شکل (7-4) تصویر گرافیکی ساختار دینامیکی را مشاهده می کنیم.



تصویر گرافیکی ساختار دینامیکی پروژه

برای ایجاد این ساختار دینامیکی ازپرینت سه بعدی کمک گرفتیم. بدین صورت که فایل های STL(فایل های مربوط به نرم افزار CAD که برای طراحی سه بعدی استفاده می شود) را در اختیار شرکت دارای پرینتر سه بعدی قرار دادیم و قطعات پرینت شده را تحویل گرفتیم. در شکل (8-4) قطعات پرینت شده را مشاهده می کنید.



قطعات پرینت شده ساختار دینامیکی

1. پیاده‌سازی و نتایج

دیاگرام کلی سیستم طراحی شده در تصویر 0 مشاهده می‌شود.پیاده‌سازی این سیستم به دو بخش کلی سخت‌افزار و نرم‌‌افزار تقسیم می‌شود. در ادامه به شرح جزییات هر یک از بخش‌ها می‌پردازیم.

  
"تصویر 0 – دیاگرام کلی سیستم"

* 1. بخش سخت افزار
     1. پلنت(plant)

همان‌طور که می‌دانیم، پلنت عنصر اصلی و اساسی کل مجموعه می‌باشد؛ چرا که تمامی اهتمام‌های صورت گرفته در تمامی بخش‌های دیگر برای بدست آوردن رفتاری مناسب در این بخش است.

پلنت مورد استفاده در این پروژه در تصویر 1 مشاهده می‌‎شود. این پلنت خود شامل دو بخش دوربین و پایه است. درمورد دوربین مورد استفاده در پلنت در بخش‌های قبل توضیح داده شده است؛ اما پایه مورد استفاده برای دوربین، پایه گیمبال نام دارد.

"تصویر 1 – تصویر پایه گیمبال و دوربین"

پایه گیمبال، نوعی مکانیسم است که می‌تواند دو درجه آزادی را فراهم کند. این پایه دو بازوی مستقل از هم دارد که هریک از آن‌ها برای درجه‌های آزادی مربوط به محورهای x و y کاربرد دارند. تصویر شماره 2، نمایی از بازوهای موجود در این مکانیسم را نشان می‌دهد.

"تصویر 2 – بازوهای گیمبال"

در واقع بازوهای گیمبال در نقش پیوندگاه پلنت سیستم و عملگرهای سیستم () عمل می‌کنند؛ به طوری که با اتصال عملگرهای سیستم به این بازوها، تغییر وضعیت پلنت نسبت به محورهای x و y معنا پیدا می‌کند.

* + 1. سنسور

بخش سنسور در این سیستم را می‌توان از هر دو بعد سخت‌افزار و نرم‌افزار بررسی می‌کرد؛ چرا که رابطه تنگاتنگی بین این دو بخش در این زمینه وجود دارد. دوربین موجود در سیستم، به نوعی، قسمتی از عملیات تشخیص (sensing) را انجام می‌دهد. عملیات تشخیص شامل ثبت تصویر و تحلیل تصویر ثبت شده می‌باشد؛ به این صورت که فرایند ثبت تصویر مربوط به دوربین و بخش سخت‌افزاری بوده و مرحله تحلیل این تصویر بر عهده نرم‌افزار است.

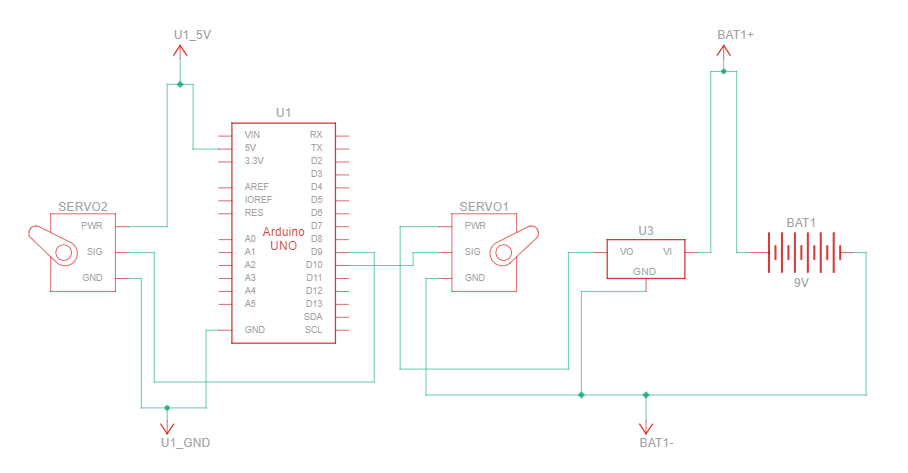
* + 1. عملگر (actuator)

همان‌طور که بیان شد، عملگرهای سیستم با اتصال به بازوهای موجود در پایه‌ی گیمبال، می‌توانند امکان تغییر وضعیت سیستم در راستای محورهای x و y را فراهم کند. این عملگرهای از نوع سرو موتور (servo motor) مدل ### بوده و دارای محدوده‌ی عملکردی حدودا 180 درجه‌ای هستند. تصویر مربوط سرو موتورهای مورد استفاده در سیستم در تصویر 3 ثابل مشاهده است.

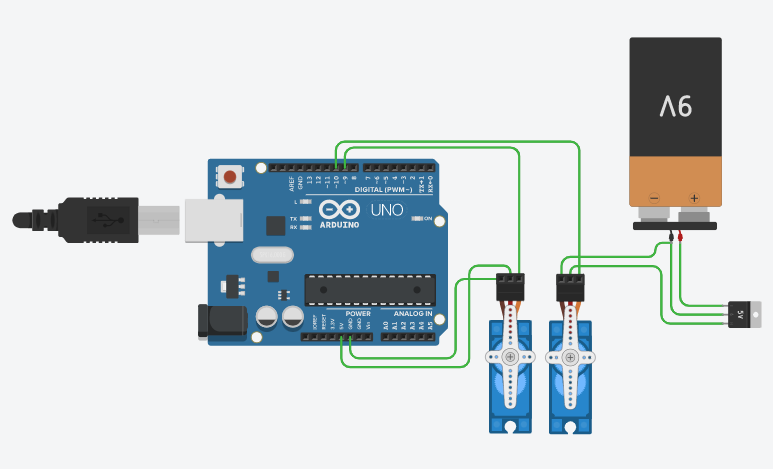
"تصویر 3 – سرو موتورهای متصل به بازوهای گیمبال"

فرمان عملکردی این عملگرها می‌بایست از سمت کنترل‌کننده‌ی سیستم صادر شود و نهایتا به عملکرها منتقل شود. می‌دانیم که فرمان کنترل‌کننده، فرمانی حاصل از بازخورد مقدار خروجی سیستم و تنظیم کنترل‌کننده‌ی سیستم است. راه‌اندازی سرو موتورها از طریق سه سیمی که در اختیار کاربر قرار گرفهت باید صورت بگیرد. بصورت معمول، این سیم‌ها دارای سه رنگ قرمز، قهوه‌ای و نارنجی است. سیم قرمز رنگ مربوط به پایه مثبت تغذیه و سیم قهوه‌ای بعنوان پایه زمین (پایه منفی تغذیه) است. همجنین سیم نارنجی رنگ نیز مربوط به پایه فرمان سرو موتور است که می‌بایست به یک منبع pwm متصل شود. این سرو موتورها برای تعمین تغذیه به منبع انرژی 5 ولت نیاز دارد. از این رو برای تعمین این منبع، از پایه‌های خروجی برد آردوینو یونو (Arduino UNO) استفاده شده است. فرمان ورودی به سرو موتورها نیز از جنس pwm (pulse width modulation) می‌باشد. این فرمان کنترلی نیز از طریق پایه‌های pwm موجود در برد آردوینو یونو صورت گرفته است.

شماتیک اتصالات انجام شده بین سرو موتورها و برد آردوینو در تصویر 4 نمایش داده شده است.

  
"تصویر 4 – شماتیک اتصال سرو موتورها به برد آردوینو"

هر سرو موتور برای تغذیه، به منبع 5 ولتی نیاز دارد. از همین رو تغذیه یکی از سرو موتورها را از پایه خروجی 5 ولتی خود آردوینو تعمین کرده و برای سرو موتور دوم، از یک باتری کتابی 9 ولت استفاده شده است. قطعه LM7805 یک تنظیم‌کننده‌ی ولتاژ () است که ولتاژ ورودی اعمال شده را می‌تواند بر روی مقدار خروجی 5 ولت تنظیم کند. از همین رو با استفاده از این قطعه و اعمال ولتاژ 9 ولتی باتری کتابی به ورودی این قطعه، در خروجی آن، ولتاژ تنظیم شده‌ با مقدار 5 ولت دریافت می‌کنیم. سپس از این مقدار ولتاژ برای تعمین تغذیه‌ی سرو موتور دوم استفاده می‌شود. همچنین پایه‌های کنترلی سرو موتورها نیز بترتیب به پایه‌های 9 و 10 دیجیتال آردوینو متصل شده‌اند. شایان ذکر است که پایه‌های 3، 5، 6، 9، 10 و 11 دیجیتال آردوینو اونو از قابلیت pwm پشتیبانی می‌کنند. نمایی از نحوه اتصالات بیان‌شده در تصویر 5 آورده شده است.

  
"تصویر 5 – نمایی از اتصالات سرو موتورها به برد آردوینو"

* 1. بخش نرم‌افزار

تا این جا ساختار سخت‌افزاری سیستم مورد بررسی قرار داده شد. اما مغز تصمیم‌گیر سیستم در بخش نرم‌افزاری قرار داده شده است. در این قسمت برای پردازش‌های مختلف سیستم از سه لپتاپ مختلف استفاده شده است و با شبکه کردن این لپتاپ‌ها، داده‌های آن‌ها منتقل می‌شوند. در این بخش ابتدا به موضوع نویز و موارد موثر بر عملکرد سیستم پرداخته و سپس سه بخش نرم‌افزاری را بررسی می‌کنیم.

* + 1. نویز، اعوجاج و اغتشاش سیستم

ما خروجی کلی سیستم را در خروجی دوربین مورد استفاده در سیستم مشاهده می‌کنیم. لذا هر نوع مزاحمت و یا کمبودی برای دوربین‌های مورد استفاده در سیستم منتج به تغییر کیفیت عملکرد سیستم می‌شود. اولین مورد کیفیت تصویربرداری خود دوربین است. واضح است که هرچه کیفیت تصویربرداری دوربین مورد استفاده بالاتر باشد، امکان پردازش‌های با دقت بالاتری را برا سیستم فراهم می‌کند. به بیان دیگر، تصویر در دوربین‌های با کیفیت بالاتر، تصویر صاف‌تر و شفاف‌تر بوده و در دوربین‌های با کیفیت پایین‌تر، دارای نویزهای زیاد در تصویر است. پس نوع دوربین و کیفیت آن بسیار بر عملکرد سیستم موثر است. برای مورد دوم نیز می‌توان به نور محیط اشاره کرد. نور محیطی که سیستم در آن قرار دارد با کیفیت تصویربرداری رابطه مستقیمی دارد؛ بدین صورت که با کم کردن نور محیط، عملا جزییات تصویر کاهش پیدا کرده و پردازش‌های مورد نظر سخت‌تر شده و یا غیرممکن می‌شود. از طرف دیگر، افزایش میزان نور محیط تا حدی می‌تواند باعث بهبود شرایط شود؛ چراکه اگر نور شدیدا زیاد شود و از حد مشخصی عبور کند، بازهم جزییات تصویر از بین رفته و پردازش‌های مورد اتظار سیستم سخت‌تر و یا غیرممکن می‌شوند. مورد سوم که اهمیت آن دست کمی از موارد قبلی ندارد، نرخ فریم بر ثانیه (fps) سیستم است. دقت کنید که نرخ فریم بر ثانیه سیستم موضوع مستقلی از نرخ فریم بر ثانیه‌ی خود دوربین است. نرخ فریم بر ثانیه سیستم در ماکسیموم خود می‌تواند برابر با نرخ خود دوربین شود؛ اما به دلیل حجم بالای پردازش‌ها اعمالی بر روی هر فریم، این نرخ شدیدا کاهش می‌کند. نرخ سیستم تابع پارامترهای مختلفی در سیستم است. توان پردازشی هسته‌های پردازشی سیستم و حجم داده‌های سیستم از جمله این پارامترها هستند. ضمن این که با توجه به این موضوع که هدف اصلی سیستم، دنبال کردن سوژه مورد نظر خود است، نتیجه گرفته می‌شود که سوژه سیستم در حال حرکت است. لذا سرعت حرکت اشیا در محیط نیز اهمیت بسیار بالایی دارد و اگر سرعت حرکت اشیا از توان پردازشی سیتسم بیشتر باشد، عملا سیستم جا مانده و سوژه را از دست می‌دهد.

* + 1. لپتاپ اول؛ خوانش تصویر و انتخاب سوژه

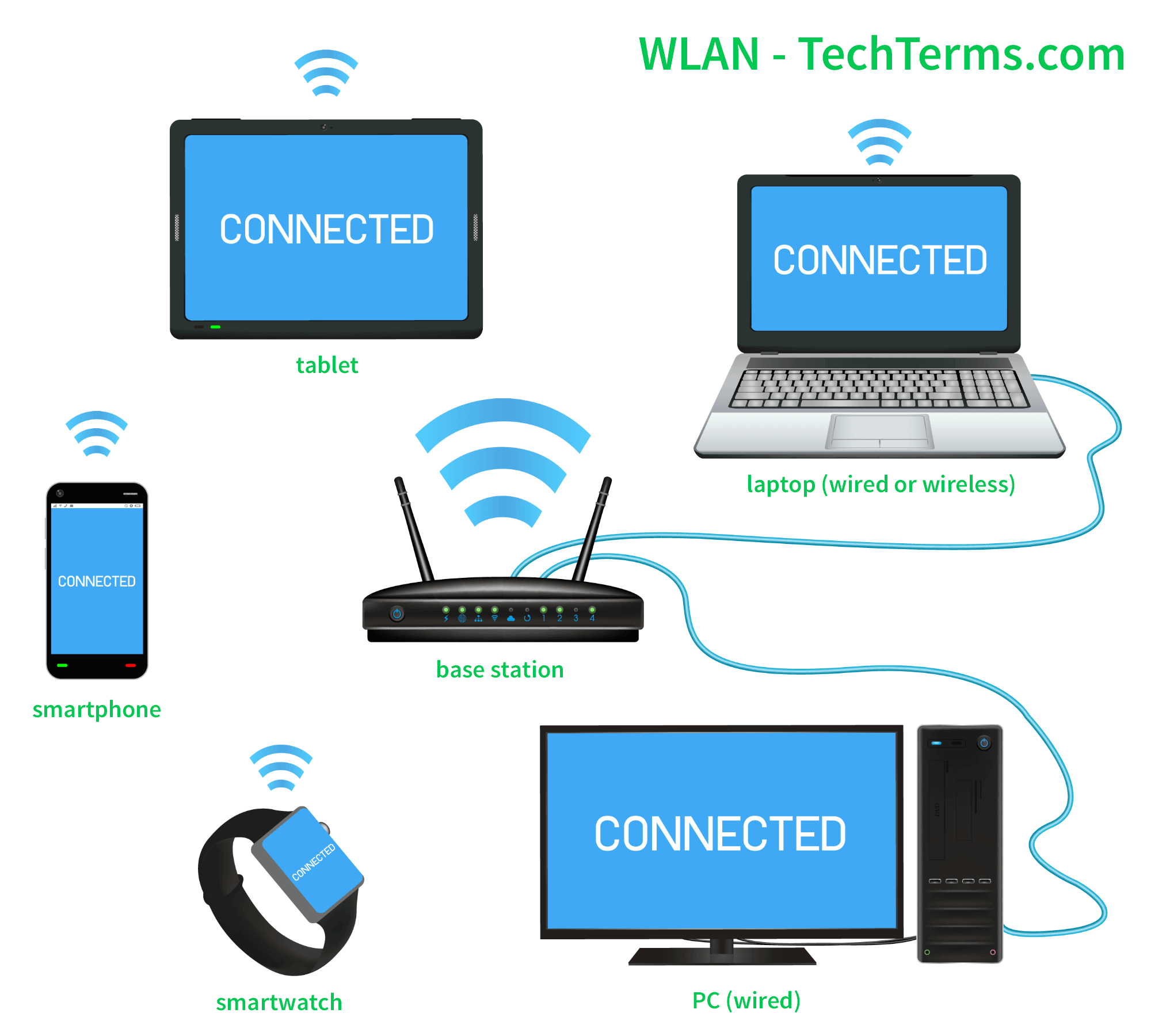
دوربین متصل به پایه گیمبال به لپتاپ اول متصل شده و تصویر برداشتی خود به آن می‌فرستد. لپتاپ اول با اجرای شبکه عصبی یولو (YOLO)، اشیا (objects) با کلاس‌های تعریف شده برایش را تشخیص داده و سپس نتایج خود را به شبکه عصبی اواس‌نت (OSNet) تحویل داده تا این شبکه به اشیا تشخیص داده شده بر اساس ویژگی‌های استخراج شده از آن‌ها (features)، شناسه‌ای اختصاص دهد (sorting). حفظ و پایداری این شناسه‌های اختصاص یافته چالش اصلی این شبکه بوده و بسار تحت تاثیر عوامل نویزی، اعوجاجی و اغتشاشی می‌باشد. پس از اختصاص شناسه به اشیا، یکی از اشیای تشخیص داده شده انتخاب شود و اطلاعات آن به لپتاپ دوم فرستاده شود.

* + 1. لپتاپ دوم؛ دریافت اطلاعات شی انتخاب شده، انتخاب و دنبال کردن آن

فرایند ابتدایی در لپتاپ دوم نیز همانند لپتاپ اول است. ابتدا شبکه عصبی یولو، اشیای با کلاس‌های تعریف شده را تشخیص داده و سپس اواس‌نت به آن‌ها شناسه‌ اختصاص می‌دهد. در مرحله بعد می‌بایست اطلاعات دریافت شده از لپتاپ اول تحلیل شوند و از بین اشیای تشخیص داده شده، آن شی‌ای که با شی انتخاب شده در قسمت قبل مطابقت دارد، انتخاب شود و مابقی اشیای تشخیص داده شده نادیده گرفته شوند. با بدست آوردن سوژه انتخاب شده سیستم، حال نوبت آن است که داده‌های آن به لپتاپ سوم، جایی که کنترل‌کننده‌ی سیستم در آن‌ جا قرار دارد، منتقل شود.

* + 1. نحوه ارتباط لپتاپ اول و دوم

زمانی که صحبت از انتقال داده می‌شود، موضوع پرتوکول‌ ارتباطی مورد استفاده مطرح می‌شود. در این بخش، از آن جایی که ما از لپتاپ اول به لپتاپ دوم داده قصد اناتقال داده داریم، واضح است که این دو لپتاپ باید به نحوی باهم در یک شبکه قرار بگیرند. از متداول‌ترین و مرسوم‌ترین شبکه‌های مورد استفاده، شبکه‌ W-LAN (تصویر 6) است. امروزه این نوع شبکه تقریبا در هر خانه و کاشانه‌ای یافت می‌شود. برای قرارگیری لپتاپ‌ها در یک شبکه کافی است که هر دوی آن‌ها به یک روتر مشترک متصل شوند. در این صورت آی‌پی‌های آنان (IP) برای یکدیگر قابل دسترس خواهند بود. حال اگر لپتاپ دوم را که دریافت‌کننده‌ی داده از لپتاپ اول است، بعنوان سرور (server) در نظر بگیریم و لپتاپ اول را بعنوان کلاینت (client)، می‌توان با استفاده چند خط کد به زبان برنامه‌نوسی پایتون (python)، سرور را راه‌اندازی کرده و داده‌ها را منتقل کرد.

  
"تصویر 6 – شبکه W-LAN"

* + 1. لپتاپ سوم؛ دریافت خروجی سیستم و تولید سیگنال خطا و فرمان کنترلی

لپتاپ سوم جایی است می‌بایست فرمان کنترلی مناسبی با توجه به وضعیت خروجی سیستم محاسبه شده و به پلنت اعمال شود. از این رو در این بخش باید کنترل‌کننده‌ی PID قرار داشته باشد. رابطه‌ی کلی مربوط به کنترل‌کننده‌های PID در بصورت معادله‌ی 1 است. در این مرحله، برای طراحی کنترل‌کننده‌ی مورد نیاز، از نرم‌افزار متلب (MATLAB) و بخش سیمولینک (Simulink) آن استفاده می‌شود.

  
" معادله 1 – رابطه کلی کنترل‌کننده‌ی pid"

* + 1. نحوه ارتباط لپتاپ دوم و سوم

نحوه ارتباط بین این دو لپتاپ نیز همانند مرحله قبل از طریق شبکه مشترک بیت دو لپتاپ است و تنها کافی است آی‌پی مربوط به سیستم‌ها را استفاده کنیم.

* 1. شرح یکپارچه پروژه

عملکرد سیستم طراحی شده از لحاظ تکنیک و الگوریتم به دو بخش زیر تقسیم می‌شود.

* ردیاب به کمک نشانه‌های محیطی (Landmarks’ Based Tracking (LBT))
* ردیاب به کمک ویژگی‌ اشیا (Feature Based Tracking (FBT))

تفاوت این دو مورد نسبت به هم در پروسه‌ی تشخیص و انتخاب شی در سیستم اول و تشخیص و یافتن همان شی در سیستم دوم می‌باشد. از این رو، تا مرحله‌ی یافتن شی انتخاب‌شده در لپتاپ دوم را بصورت جدا شرح داده می‌شود و سپس ادامه‌ی روند عملکرد سیستم که در هر دو حالت یکسان است، ارایه می‌شود.

* + 1. ردیاب – نشانه‌های محیطی (LBT)

همان‌طور که از نام این تکنیک برمی‌آید، سیستم در این حالت برای عملکرد صحیح نیاز به نشانه‌هایی درون محیط دارد. درواقع پس از آن که در تصویر در دوربین اول مشاهده و یکی از اشیای موجود در آن توسط کاربر انتخاب شد، سیستم و دوربین دوم باید قادر باشند تا همان شی انتخاب شده را در تصویر خود نیز انتخاب کنند؛ از این رو می‌بایست تصاویر هر دو دوربین همگام‌(synchronized) شوند. نشانه‌های محیطی مذکور، المان‌هایی هستند که می‌توان مشخصا درون محیط قرار داد تا با استفاده ویژگی‌های مختص آنان تصاویر دوربین‌ها را به همگام‌سازی کرد. به عبارت دیگر، با چنین کاری می‌توان به یک تابع تبدیل از دوربین اول به دوربین دوم دست یافت. سپس با استفاده از آن تابع تبدیل می‌توان تصاویر دوربین‌ها را همگام‌شده فرض کرد.

نشانه‌ی محیطی مورد استفاده ما در این پروژه یک صفحه‌ی مستطیل شکل به رنگ تماما قرمز می‌باشد. در این حالت با توجه به تماما قرمز بودن آن و همینطور شکل منتظم آن، می‌توان با استفاده از تکنیک‌های مربوط به رنگ در حوزه پردازش تصویر، حضور و موقعیت این نشانه را در تصویر تشخیص داد و از اطلاعات آن استفاده کرد. برای مثال در تصویر 7، ورودی و خروجی یک تصویو شامل نشانه‌ مشاهده می‌شود که در آن نشانه‌ قرمز رنگ موجود در تصویر با استفاده از پردازش تصویر تشخیص داده شده و مختصات مرکز آن شی نسبت به مبدا تصویر گزارش شده است.

" تصویر 7 – مثال تشخیص نشانه محیطی قرمز "

* + - 1. همگام‌سازی تصاویر و بدست آوردن تابع تبدیل بین آن‌ها

در تصویر 8 مشاهده می‌شود که وضعیت قرارگیری نشانه‌های قرارداده‌شده در محیط به چه صورت می‌باشند. با ارسال مختصات مرکز نشانه‌ موجود در تصویر اول (در لپتاپ اول) به لپتاپ دوم و همچنین بدست آوردن مختصات مرکز همان نشانه در تصویر دوم (در لپتاپ دوم) می‌توان به تابع تبدیلی مورد نظر دست یافت.

" تصویر 8 – نشانه‌های مورد استفاده در محیط تست"

در نتیجه‎ی این پروسه، تابع تبدیل بدست آمده بصورت معادله‌ی 2 می‌باشد. با استفاده از این تابع تبدیل می‌توان مختصان شی انتخاب‌شده را به مختصات آن شی در تصویر دوم تبدیل کرد.

" معادله 2 – تابع تبدیل بدست آمده برای همگام‌سازی تصاویر"

* + - 1. تشخیص شی انتخاب شده در لپتاپ دوم

پس از بدست آوردن مختصات شی در تصویر دوم با استفاده از تابع تبدیل بدست آمده و سپس مقایسه آن با مختصات اشیای تشخیص داده شده در تصویر، می‌توان همان شی را از بین اشیا تمیز داد و آن را انتخاب کرد. نمایی از این قسمت در تصویر 9 آورده شده است. در این تصویر شی انتخاب شده در لپتاپ اول و همچنین شی یافته‌شده در لپتاپ دوم قابل تشخیص است.

" تصویر 9 – انتخاب شی در لپتاپ اول و یافتن آن در لپتاپ دوم"

* + 1. ردیاب – ویژگی اشیا (FBT)

در این حالت، سیستم احتیاجی به همگام‌سازی تصاویر نداشته و با استفاده از امکاناتی که هوش مصنوعی مورد استفاده در سیستم در اختیار قرار می‌دهد می‌توان سیستم را عملیاتی کرد. می‌دانیم که هر شی تشخیص داده شده در یک تصویر توسط هوش مصنوعی، دارای ویژگی‌های مشخصی می‌باشد که سبب می‌شود آن به عنوان خود شناسایی شود. اما این اشیا هریک ویژگی‌های مختص خود را نیز دارند که می‌توانند هر شی را از اشیای دیگر هم‌کلاس و یا غیر هم‌کلاس خود متمایز کند. با استخراج چنین ویژگی‌هایی، واضح است که چالش سیستم برطرف شده و می‌توان شی انتخاب‌شده در لپتاپ اول را در لپتاپ دوم شناسایی کرد و آن را دنبال کرد.

با نفوذ به لایه‌ی عمیق‌تر شبکه‌ی عصبی اواس‎نت که پیشتر در مورد آن صحبت شده است، چنین ویژگی‌هایی از تمام اشیای تشخیص داده‌شده‌ی تصویر را می‌توان استخراج کرد و به آنان دسترسی داشت. ویژگی‌های استخراج شده از جنس اعداد در قالب یک ماتریس با 512 سطر و 1 ستون می‌باشند. از این رو می‌توان با ارسال ویژگی شی انتخاب شده در لپتاپ اول از به لپتاپ دوم، و مقایسه‌ی این ویژگی‌ با ویژگی‌های تمامی اشیا موجود در تصویر دوم، آن شی را که کم‌ترین عدم تطابق را با ویژگی دریافت شده دارد و دقت آن از مقدار مشخصی که تعیین شده است بیش‌تر است، بعنوان شی انتخاب شده شناسایی و انتخاب کرد. نمایی از انتخاب شی در لپتاپ اول و شناسایی همان شی در لپتاپ دوم در تصویر 10 آورده شده است.

" تصویر 10 – انتخاب شی در لپتاپ اول و یافتن آن در لپتاپ دوم"

* + 1. رهگیری شی شناسایی شده بعنوان هدف
       1. بستن حلقه‎ی سیستم

در این مرحله، به هر حال، هدف سیستم شناسایی و انتخاب شده است. می‌دانیم برای آن که بتوان آن را رهگیری کرد (بعبارت دیگر، سیستم را کنترل کرد)، می‌بایست سیستم بصورت حلقه‌ بسته باشد. این مرحله دقیقا همان جایی است که حلقه‌ی سیستم باید بسته شود. از این رو خروجی سیستم که مختصات هدف سیستم در تصویر دوربین دوم است را باید بازخورد (feedback) کرد. همچنین می‌‌دانیم که کنترل یک سیستم از طریق مقدار خطای سیستم (error) انجام می‌شود. لذا می‌بایست مقدار خطای سیستم در هر لحظه محاسبه شود. برای محاسبه‌ی خطا نیز به ست‌پوینت نیاز می‌باشد و واضح است که در این سیستم، ست‌پوینت مدنظر مختصات مرکز تصویر، یعنی (50,50) می‌باشد. لازم به ذکر است که بیان مختصات بصورت درصدی می‌باشد.

* + - 1. کنترل‌کننده PID

در هر لحظه، مختصات هدف سیستم از لپتاپ دوم به لپتاپ سوم ارسال می‌شود و در فرایند کنترل سیستم با اطلاعات در دست در لپتاپ سوم صورت می‌گیرد. بدین صورت که با دریافت مختصات هدف سیستم در سیمولینک متلب و محاسبه‌ی سیگنال خطای سیستم که حاصل تفاضل خروجی سیستم و ست‌پوینت آن است، آماده‌ی تولید سیگنال کنترلی می‌شود. سیگنال کنترلی در واقع خروجی بخش کنترل‌کننده‌ی هر سیستم می‌باشد. از این رو در این سیستم، سیگنال خطا ورودی کنترل‌کننده pid و خروجی آن سیگنال کنترلی است. واضح است که کنترل‌کننده‌ی pid نیاز دارد تا با توجه به دینامیک سییستم، تنظیم شود. روش‌های مختلفی برای تنظیم کردن کنترل‌کننده‌ی pid ارایه شده است؛ یکی از شناخته‌شده‌ترین این روش‌ها، روش زیگلر-نیکولز (Ziegler-Nichols) می‌باشد.

بصورت کلی، روش زیگلر-نیکولز به این صورت عمل می‌کند که ابتدا کنترل‌کننده را تنها بصورت تناسبی بر روی سیستم حلقه باز قرار داده و مقدار ضریب تناسبی آن را تا حدی بالا می‌بریم تا رفتار خروجی سیستم به حالت نوسانی پایدار برسد. سپس مقادیر مناسب برای ضرایب کنترل‌کننده را با توجه به جدول ارایه شده در این روش می‌توان بدست آورد.

در این سیستم با اعمال روش زیگلر-نیکولز و بدست آوردن ضرایب، رفتار سیستم تا حدی دارای نقص بود که با کمی تنظیم ضرایب بصورت دستی، خروجی مناسبی از سیستم مشاهده شد. در نهایت، ضرایب مورد استفاده بصورت معادله 3 تنظیم شدند. همچنین در تصویر 11، نمایی از بلوک‌های قرار داده شده در سیمولینک متلب آوره شده‌است.

" معادله 3 – ضرایب بدست آمده برای کنترل‌کننده"

" تصویر 11 – نمایی از بلوک‌های قرار گرفته در سیمولینک "

* + 1. سلام سلام بر تو ای سلام
    2. فرمان‌دهی عملگرهای سیستم

در نهایت امر، کنترل‌کننده با توجه به ضرایب تنظیم‌شده برای آن در هر لحظه، خوجی مناسب را که همان سیگنال کنترلی است تولید می‌کند. این سیگنال می‌بایست از طریق برقراری ارتباط سریال با پلنت، به عملگرها منتقل شود. یکی از روش‌های برقراری چنین ارتباطی، استفاده از بلوک‌های خواندن و نوشتن آردوینو در سیمولینک بصورت واحدهای ورودی/خروجی است. تشریح مطالب مربوط به سیمولینک متلب در بخش ضمایم آمده است. با توجه به این که برای فرمان‌دهی به عملگرهای سییستم از برد آردوینو اونو استفاده شده است، این روش مناسب بوده و نیاز‌های سیستم را برآوره می‌کند. همچنین همواره باید توجه داشت که در امر انتقال دیتا، می‌بایست قراردادی بین واحدهای فرستنده و گیرنده وجود داشته باشد تا قالب داده‌های ارسالی و دریافتی در هر دو یکسان بوده و داده بدرستی تجزیه شده و مورد استفاده قرار بگیرند.

تا این مرحله، داده‌های سیگنال کنترلی به برد آردوینو منتقل شده‌اند. در این جا، هسته‌ی پردازشی آردوینو باید مطابق با قالب تعیین شده برایشف داده‌های دریافتی را تجزیه کرده و آنان را بر سرو موتورهای متصل به پایه‌های خود اعمال کند. در نتیجه‌ی این کار، سرو موتورها که بازوان سیستم هستند، به مقدار فرمان داده شده چرخیده و سیستم را در شرایط جدیدی قرار می‌دهند و می‌بایست حلقه‌ی بسته سیستم بار دیگر برای بروزرسانی بار دیگر آن طی شود. بدین ترتیب با تکرار این چرخه، سیستم همواره در تلاش برای کاهش خطای خود بوده و در نتیجه، هدف انتخاب شده همواره دنبال و رهگیری می‌شود؛ حتی اگر این هدف بصورت متحرک باشد. تصویر 12 به خوبی نشان می‌دهد که سیستم چطور با فرمان‌‌دهی به عملگرها و چرخاندن آنان، خطای سیستم را از بین برده و هدف خود را در وسط تصویر (همان ست‌پوینت) قرار می‌دهد.

" تصویر 12 – خروجی سیستم قبل و بعد از کنترل"

1. نتیجه‌گیری

در این گزارش به بررسی کامل یه سیستم رهگیری پرداختیم که شامل بخش‌های تئوری و بخش‌های عملیاتی و پیاده‌سازی بود. در فصل دوم به بررسی بخش‌های مختلف یک سامانه و سیستم رهگیری پرداختیم و روش‌ها و همچنین الگوریتم‌های مختلفی که در این زمینه وجود دراد را مورد بررسی قرار دادیم. همان‌طور که در فصل اول بررسی کردیم، یک سیستم رهگیری شامل 4 بخش اصلی می‌باشد که این بخش‌ها به ترتیب شامل مدل‌سازی هدف مورد‌نظر، انتخاب ویژگی‌های ساختاری و ظاهری مناسب و مورد‌نیاز، تشخیص هدف در هر تصویر و در نهایت رهگیری هدف معین در فریم‌های متوالی به کمک الگوریتم‌های موجود برای رهگیری. در قسمت مدل‌سازی هدف مورد نظر، با توجه با ساختاری که هدف دارد می‌تواند به کمک روش‌هایی مانند مدل‌سازی نقطه‌ای ،که برای هدف‌های با ابعاد کوچک مناسب است، و یا مدل سطح داخلی، که برای اهداف با ساختار ظاهری پیچیده مناسب است، و یا بقیه روش‌ها استفاده کرد. یکی دیگر از قسمت‌های مهم در بخش مدل‌سازی، مدل‌سازی ویژگی‌های ظاهری هدف مورد‌نظر می‌باشد که به این منظور می‌توان از روش‌هایی مانند مدل چگالی احتمال و یا مدل ظاهر‌چندوجهی، که روش بنیادی در شبکه‌های عصبی می‌باشد، استفاده کرد. بعد از مدل‌سازی هدف و یا شئ مورد‌نظر باید ویژگی‌های ساختاری و ظاهری‌مهم برای هدف را انتخاب کرده و به عنوان ویژگی یکتا برای هدف در نظر گرفت. علت این کار این است که انتخاب ویژگی‌های یکتا برای یک هدف و یا شئ به ما کمک می‌کند که در فریم‌های متوالی در یک ویدیو تشخیص دهیم که هدف مشخص شده همان هدف مورد‌نظر ما می‌باشد. ویژگی‌هایی که می‌توان در نظر گرفت شامل رنگ، چگالی احتمال رنگ، کناره‌های تصویر، جریان‌نوری و بافت هدف می‌باشد. بعد از مشخص کردن مشخصات‌ظاهری باید به کمک الگوریتم‌های موجود بتوانیم اهداف مورد نظر خود را در تصویر تشخیص دهیم. تشخیص هدف مهم‌ترین مرحله در یک سیستم رهگیری می‌باشد. برای تشخیص اهداف می‌توانیم از روش‌هایی مانند تشخیص‌دهنده‌های نقطه‌ای، اختلاف پس‌زمینه‌، قسمت‌بندی و یادگیری با‌نظارت استفاده کرد. در بین این روش‌ها، روش یادگیری با نظارت که روش بنیادی در شبکه‌های عصبی است امروزه کاربرد فراوان‌تری نسبت به بقیه روش‌ها دارد. ما نیز در این پروژه از همین روش‌استفاده کرده‌ایم. بعد از تشخیص هدف در تصویر باید به کمک الگوریتم‌های ردیابی موجود عمل رهگیری انجام شود. الگوریتم‌ انتخابی در این مرحله به مراحل قبل و این که هدف چگونه مدل‌سازی شده‌است و یا کدام ویژگی‌های آن انتخاب شده‌است بستگی دارد. الگوریتم‌های موجود در این قسمت شامل ردیابی نقطه‌ای، ردیابی کرنل و ردیابی به کمک ساختار داخلی می‌باشد. الگوریتم انتخابی ما در این پروژه، ردیابی کرنل می‌باشد و علت این امر نیز آن است که در دریابی کرنل، به کمک ویژگی‌های ظاهری که از یک هدف استخراج شده و به نوعی رمزگذاری شده‌است رهگیری صورت می‌پذیرد و ما نیز در روند پروژه از یک شبکه عصبی برای استخراج ویژگی‌های ظاهری استفاده کرده‌ایم. بنابراین ردیابی کرنل بهترین گزینه برای ما می‌باشد. علت استفاده از شبکه عصبی نیز عملکرد بسیار خوب آن در استخراج ویژگی‌ها می‌باشد. در ادامه و در فصل سوم به بررسی دقیق الگوریتم‌های بینایی ماشین و شبکه‌های عصبی و دقیق پرداختیم. در روند پروژه از دو شبکه عصبی عمیق برای پیاده‌سازی سیستم استفاده شده‌است. یکی از این شبکه‌های عصبی با عنوان YOLO برای تشخیص هدف مورد نظر استفاده شده و شبکه عصبی دیگر با عنوان OS Net برای استخراج ویژگی‌های هدف مورد‌نظر به کار‌گرفته شده‌است. هر دو شبکه عصبی در راستای مراحل بیان شده در فصل دوم یعنی تشخیص هدف و انتخاب و استخراج ویژگی‌های هدف استفاده شده‌اند. در ادامه و در فصل چهارم به بررسی سخت‌افزار های استفاده شده در روند پروژه پرداخته شد که شامل انواع برد‌های پردازش به عنوان کنترل‌کننده، دوربین استفاده شده به عنوان سنسور حلقه کنترلی، سروو‌موتور‌ها به عنوان عملگر حلقه‌کنترلی و ساختار دینامیکی می‌باشد. در فصل پنجم و آخر نیز مراحل پیاده‌سازی سیستم رهگیری را مورد بررسی قرار دادیم. در این فصل حلقه‌کنترلی سیستم بررسی شده و مراحلی که برای پیاده‌سازی عملی مورد نیاز بود بررسی شد. ابتدا در سیستم اول محیط جستو‌جو می‌شود و اهداف تشخیص داده می‌شود، سپس هدف مورد نظر به کمک اپراتور انتخاب شده و به سیستم دوم منتقل می‌شود. در ادامه سیستم دوم وظیفه رهگیری را بر عهده دارد و بدین صورت عمل می‌کند که به کمک دوربین، داده‌های از جنس تصویر را وارد سیستم می‌کند سپس پردازش لازم برای تشخیص هدف و رهگیری آن صورت گرفته و دستور لازم به موتور‌ها ارسال می‌شود و موتور‌ها دوربین متحرک را برای رهگیری و دنبال کردن هدف حرکت می‌دهند.

مراجع

مراجع

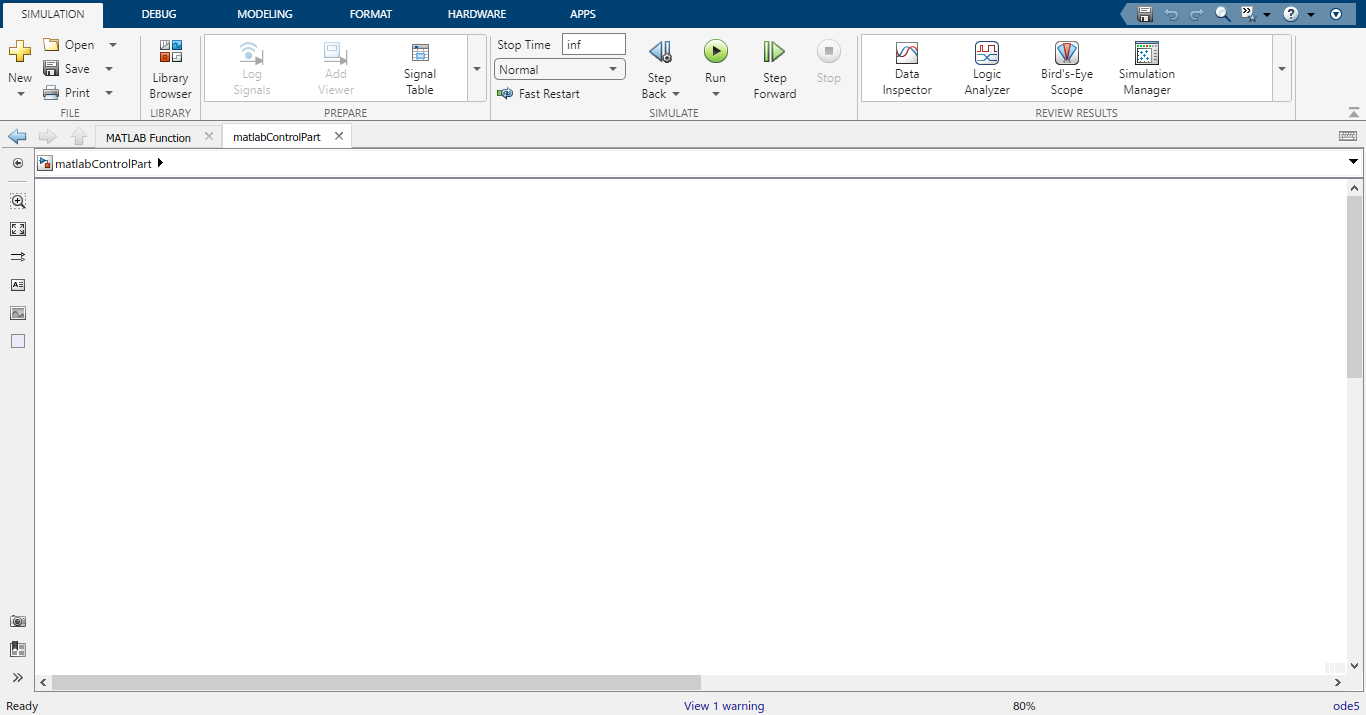
1. S. R. Balaji, Dr. S. Karthikeyan, "A survey on moving object tracking using image processing", 11th International Conference on Intelligent Systems and Control (ISCO), 2017
2. Alper Yilmaz, Omar Javed, Mubarak Shah, "Object tracking: A survey", Association for Computing Machinery, 2006
3. YOLOv5: The friendliest AI architecture you'll ever use, https://ultralytics.com/yolov5
4. Code Implementation of YOLOv5, https://github.com/ultralytics/yolov5
5. Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, Tao Xiang, " Omni-Scale Feature Learning for Person Re-Identification", Conference on Computer Vision (ICCV), 2019
6. Code Implementation of deep person reidentification, https://github.com/KaiyangZhou/deep-person-reid
7. Zhou, Kaiyang and Xiang, Tao, " Torchreid: A Library for Deep Learning Person Re-Identification in Pytorch", arXiv preprint arXiv:1910.10093, 2019
8. Zhou, Kaiyang and Yang, Yongxin and Cavallaro, Andrea and Xiang, Tao, " Learning Generalisable Omni-Scale Representations for Person Re-Identification", TPAMI, 2021
9. YOLOv5 and StrongSORT with OSNet, https://github.com/mikel-brostrom/Yolov5\_StrongSORT\_OSNet

پيوست‌ها

پیوست الف

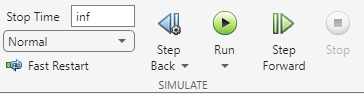
مروری کوتاه بر متلب

**MATLAB یکی از قدرتمندترین ابزارهای حوزه مهندسی می‌باشد. امکانات فراوانی را در اختیار مهندسین قرار می‌دهد. یکی از قابلیت‌هایی که این نرم‌افزار در کنار امکان کدنویسی دارد، بخش Simulink آن است. نمای کلی آن را در شکل 2 مشاهده می‌کنید.**



**در این بخش می‌توان انواع سیستم‌ها را شبیه‌سازی کرد و به تحلیل آنان پرداخت.**

**شکل 3 نشان‌دهنده‌ی بخشی است که سیستم‌های مدل شده در سیمولینک را می‌توان بصورت عادی اجرا و شبیه‌سازی کرد.**



**این شبیه‌سازی بصورت real-time نبوده و صرفا محاسبات ریاضی بر روی یک سیستم انجام شده و نتیجه را نمایش می‌دهد.**

شبیه‌سازی با استفاده از support packages 1

**سیمولینک متلب این قابلیت را دارد تا با انواع بردهای مرسوم کنترلی تعامل کند. برای مثال سیمولینک می‌تواند به انواع بردهای Arduino متصل شود. لذا با نصب support packageهای متناسب با برد کنترلی خود، می‌توان برنامه مورد نظر خود را از طریق سیمولینک بر روی برد برنامه‌ریزی کرده و با آن ارتباط برقرار کرد. شکل 4 مربوط به نحوه اجرا کردن مدل در این حالت است.**

**شکل مربوط به این بخش**

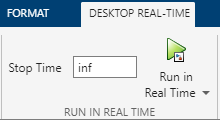
شبیه‌سازی با استفاده از support packages 2

**از نسخه 2022a متلب، قابلیت جدیدی به بخش support packageهای آن اضافه شده است که می‌توان مدل طراحی شده در سیمولینک را بدون برنامه‌ریزی کردن خود برد کنترلی اجرا کرد و در عین حال، با برد کنترلی نیز در حال تعامل بود. تصویر مربوط به این بخش را در شکل 5 مشاهده می‌کنید.**

**شکل مربوط به این بخش**

شبیه‌سازی در حالت Simulink desktop realtime

**با نصب کردن این افزونه بر روی نرم‌افزار متلب، امکان جدیدی به آن اضافه می‌شود که می‌توان بصورت بلادرنگ(real-time)، مدل شبیه‌سازی ‌شده‌ی سیمولینک را طوری اجرا کرد که با دنیای سخت‌افزاری بیرون ارتباط برقرار کرده و تبادل داده انجام دهد. تصویر مرتبط با این بخش سیمولینک در تصویر 6 مشاهده می‌شود.**



1. Real Time [↑](#footnote-ref-2)
2. Reliability [↑](#footnote-ref-3)
3. Points [↑](#footnote-ref-4)
4. Primitive geometric shapes [↑](#footnote-ref-5)
5. Bonding Box [↑](#footnote-ref-6)
6. silhouette and contour [↑](#footnote-ref-7)
7. Contour [↑](#footnote-ref-8)
8. Articulated Shape [↑](#footnote-ref-9)
9. Skeletal models [↑](#footnote-ref-10)
10. Medial Axis [↑](#footnote-ref-11)
11. Probability densities [↑](#footnote-ref-12)
12. Templates [↑](#footnote-ref-13)
13. Single View [↑](#footnote-ref-14)
14. Active appearance [↑](#footnote-ref-15)
15. *Multiview appearance* [↑](#footnote-ref-16)
16. Principal Component Analysis (PCA) [↑](#footnote-ref-17)
17. Subspace [↑](#footnote-ref-18)
18. Uniform [↑](#footnote-ref-19)
19. Edges [↑](#footnote-ref-20)
20. Optical Flow [↑](#footnote-ref-21)
21. Texture [↑](#footnote-ref-22)
22. Object Detection [↑](#footnote-ref-23)
23. Point Detectors [↑](#footnote-ref-24)
24. Threshold [↑](#footnote-ref-25)
25. Background Subtraction [↑](#footnote-ref-26)
26. **Segmentation** [↑](#footnote-ref-27)
27. Clusters [↑](#footnote-ref-28)
28. Supervised Learning [↑](#footnote-ref-29)
29. Classification [↑](#footnote-ref-30)
30. Point Tracking [↑](#footnote-ref-31)
31. Kernel Tracking [↑](#footnote-ref-32)
32. Silhouette Tracking [↑](#footnote-ref-33)
33. Deterministic Method [↑](#footnote-ref-34)
34. Statistical Method [↑](#footnote-ref-35)
35. Tracking Using Template [↑](#footnote-ref-36)
36. Tracking Using Multiview Appearance Models [↑](#footnote-ref-37)
37. Silhouette Tracking [↑](#footnote-ref-38)
38. Contour Tracking [↑](#footnote-ref-39)
39. Shape Matching [↑](#footnote-ref-40)
40. Noise [↑](#footnote-ref-41)
41. Object Detection [↑](#footnote-ref-42)
42. Supervised [↑](#footnote-ref-43)
43. Unsupervised [↑](#footnote-ref-44)
44. Dimension [↑](#footnote-ref-45)
45. Task [↑](#footnote-ref-46)
46. Kernel [↑](#footnote-ref-47)
47. Convolutional Layers [↑](#footnote-ref-48)
48. Convolutional Neural Networks [↑](#footnote-ref-49)
49. Error Back Propagation [↑](#footnote-ref-50)
50. Deep Neural Network [↑](#footnote-ref-51)