# روند و اجرای پروژه

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"

## بخش سخت افزار

### 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.

### 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.

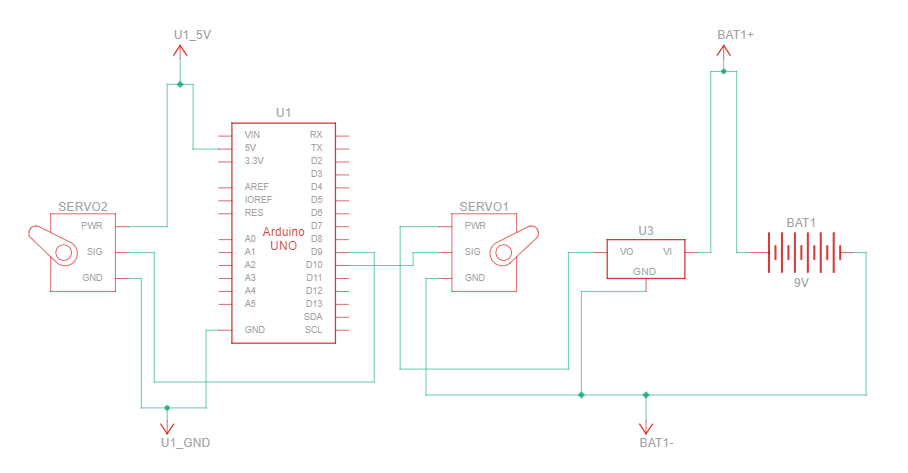
### عملگر (actuator)

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.

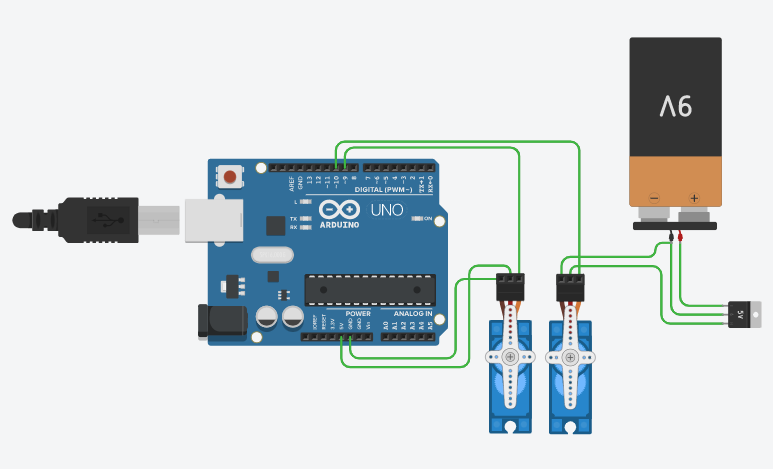
"تصویر 3 – سرو موتورهای متصل به بازوهای گیمبال"

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"

## 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.

### 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.

### لپتاپ اول؛ خوانش تصویر و انتخاب سوژه

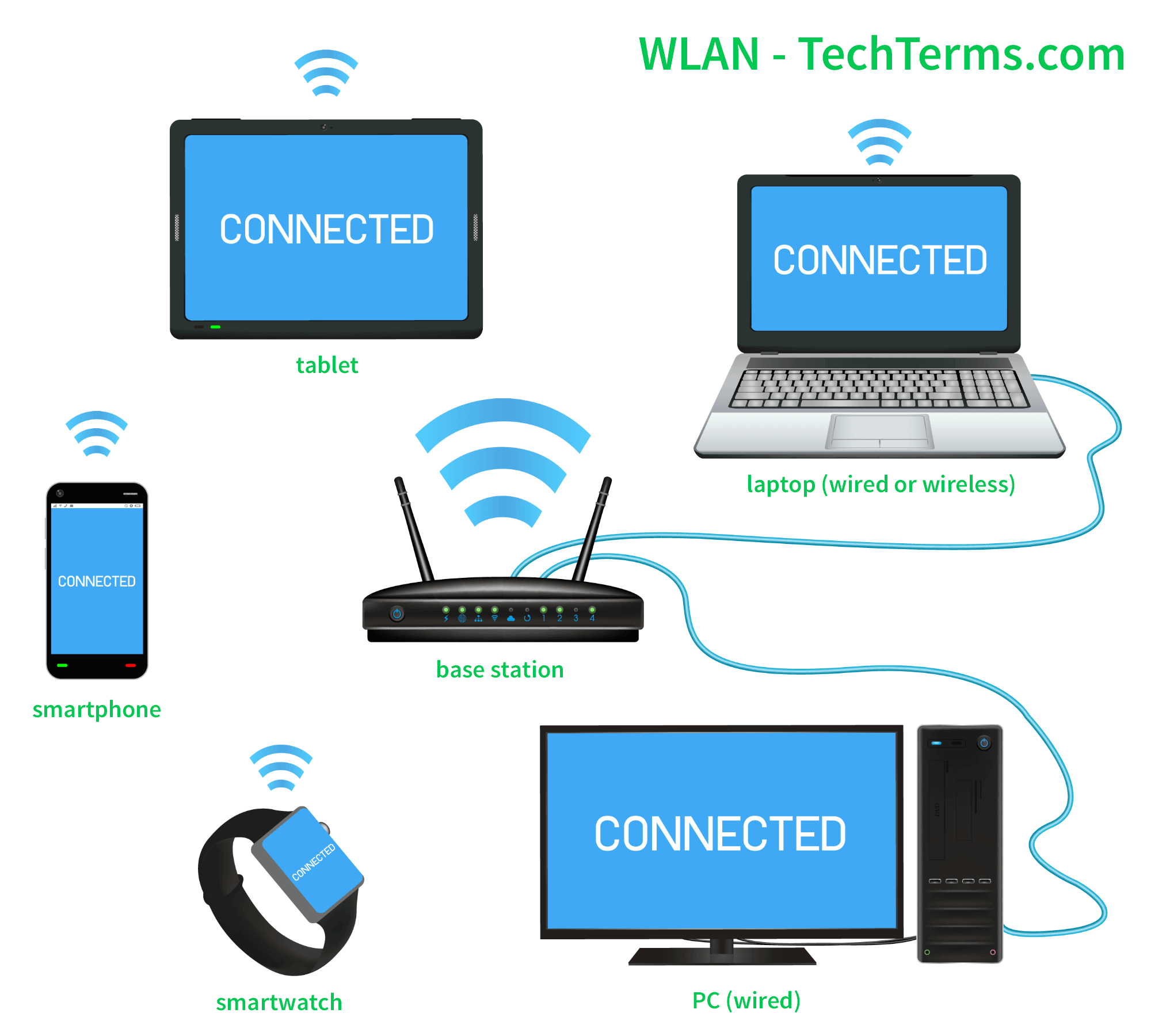
دوربین متصل به پایه گیمبال به لپتاپ اول متصل شده و تصویر برداشتی خود به آن می‌فرستد. لپتاپ اول با اجرای شبکه عصبی یولو (YOLO)، اشیا (objects) با کلاس‌های تعریف شده برایش را تشخیص داده و سپس نتایج خود را به شبکه عصبی اواس‌نت (OSNet) تحویل داده تا این شبکه به اشیا تشخیص داده شده بر اساس ویژگی‌های استخراج 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.

### 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.

### 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"

### 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"

### 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.

## 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.

### 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 "

#### 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"

#### 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"

### 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"

### رهگیری شی شناسایی شده بعنوان هدف

#### 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.

#### 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 "

#### 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"