

پیاده‌سازی الگوریتم‌های پردازش تصویر روی یک سامانه نهفته کارا با ابزارهای توسعه منبع باز

بهنام محمدی^۱، سید محمدرضا موسوی میرکلائی^۲، مهدی نصیری^۳

M_Mosavi@iust.ac.ir

۱- دانشجوی کارشناسی ارشد دانشکده مهندسی برق، دانشگاه علم و صنعت ایران

۲- استاد دانشکده مهندسی برق، دانشگاه علم و صنعت ایران

۳- استادیار دانشکده فناوری اطلاعات و ارتباطات، دانشگاه جامع امام حسین (ع)

چکیده

پردازش تصویر دیجیتال، یک زمینه در حال گسترش با کاربردهایی در زندگی روزمره است. در بسیاری از کاربردهای عملی برای پیاده‌سازی الگوریتم‌ها به سخت‌افزارهایی با هزینه و توان مصرفی پایین، قدرت پردازش مناسب، سادگی در توسعه الگوریتم‌ها و حداکثر انعطاف‌پذیری نیاز است. در این مقاله، به عنوان سخت‌افزاری کم هزینه، کم وزن و کم توان برای پیاده‌سازی روش‌های آشکارسازی اهداف در تصویر مادون قرمز مورد بررسی و تحلیل قرار می‌گیرد. پیاده‌سازی مناسب این روش‌ها و انجام عملیات بصورت بلادرنگ برای سامانه‌های دفاعی از اهمیت ویژه‌ای برخوردار است. نتایج نشان می‌دهند که Raspberry Pi 2 دارای قدرت محاسباتی کافی برای پیاده‌سازی الگوریتم آشکارسازی هدف در تصاویر مادون قرمز است. قدرت پردازش سخت‌افزار پیشنهادی با استفاده از الگوریتم آشکارسازی هدف تصاویر مادون قرمز روی محیط توسعه نرم‌افزاری Qt و توابع کتابخانه پردازش تصویر OpenCV با PC روی محیط توسعه نرم‌افزاری Qt و توابع کتابخانه OpenCV و همچنین با نرم‌افزار سطح بالای MATLAB مقایسه می‌شود. نتایج به دست آمده نشان می‌دهند که پیاده‌سازی روی Raspberry Pi 2 نسبت به MATLAB سرعت اجرای الگوریتم را ۶٫۵ برابر افزایش می‌دهد. همچنین زمان اجرای پیاده‌سازی الگوریتم آشکارسازی هدف در تصاویر مادون قرمز (به زبان ++C) با استفاده از کتابخانه OpenCV روی PC تقریباً ۸ برابر اجرای آن با Raspberry Pi 2 است. همچنین با مقایسه Raspberry Pi 2 و PC از نظر توان مصرفی، وزن و هزینه مشاهده می‌شود که Raspberry Pi 2 کارایی بسیار بهتری را از نظر توان مصرفی، وزن و هزینه نسبت به PC دارد. نتایج نشان می‌دهند که هر چند استفاده از نرم‌افزارهای سطح بالا مثل MATLAB دارای شاخص‌های ارزیابی ضرایب تضعیف پس-زمینه ((SCR و نسبت سیگنال به نویز (BSF بالاتری نسبت به استفاده از کتابخانه OpenCV است، اما نتایج زمان اجرا نشان می‌دهد که سخت‌افزار پیشنهادی زمان اجرا را نسبت به نرم‌افزارهای سطح بالا مثل MATLAB بهبود می‌دهد. برای بهینه‌سازی و کاهش زمان اجرا از برنامه‌نویسی چندرشته‌ای روی Raspberry Pi 2 (که شامل پردازنده ۴ هسته‌ای ARM Cortex-A7 است) و خاصیت افزایش فرکانس (برای افزایش سرعت سخت‌افزار Raspberry Pi 2) استفاده می‌شود. نتایج پیاده‌سازی حاصل از بهینه‌سازی روی Raspberry Pi 2 نشان می‌دهند که سرعت اجرای الگوریتم بیش از ۴۰ درصد بهبود می‌یابد.

واژگان کلیدی: آشکارسازی هدف، Raspberry Pi 2، OpenCV، توان پایین، هزینه پایین، وزن کم.

تاریخ دریافت مقاله: ۹۷/۰۵/۰۱

تاریخ پذیرش مقاله: ۹۷/۰۹/۱۲

۱- مقدمه

در طول دهه‌های گذشته، آشکارسازی اهداف کوچک مادون قرمز در بسیاری از زمینه‌های نظامی و غیرنظامی مثل هدایت دقیق مادون قرمز، هشداردهنده‌ها، نظارت ویدئویی، جستجو و ردیابی به طور گسترده مورد توجه قرار گرفته‌اند. در این کاربردها، اهداف مادون قرمز آن‌چنان از سامانه‌های تصویربرداری دور هستند که در تصاویر خیلی کوچک و بی‌شکل می‌باشند و پیکسل‌های کمی را در تصاویر اشغال می‌کنند. به علاوه، پس‌زمینه اهداف کوچک مادون قرمز معمولا به وسیله نویز ناشناخته آلوده می‌شود و کنتراست (تضاد) بین هدف و پس‌زمینه خیلی کم است. بنابراین آشکارسازی هدف کم‌نور مادون قرمز تحت پس‌زمینه نویزی نسبتا مشکل است [۲۰].

در بسیاری از کاربردهای عملی و نظامی، سادگی الگوریتم با قابلیت آشکارسازی بالا و اجرای بلادرنگ روش‌های آشکارسازی از اهمیت بالایی برخوردار است. برای رسیدن به عملکرد بلادرنگ روش‌های آشکارسازی هدف در تصاویر مادون قرمز از پیاده‌سازی این الگوریتم‌ها بر روی سخت‌افزار استفاده می‌شود.

پیاده‌سازی الگوریتم‌های پردازش تصویر روی رایانه‌های همه منظوره و نرم‌افزارهای سطح بالا (با کاربری آسان) مثل MATLAB می‌تواند ساده‌ترین راه‌حل باشد. اما به دلیل محدودیت‌هایی مثل نیاز به توان مصرفی کم، حجم سخت‌افزار کم، هزینه پایین و قدرت پردازش بسیار بالا نمی‌توان از رایانه‌های همه منظوره برای کاربردهای عملی استفاده کرد. به همین دلیل استفاده از سامانه‌های نهفته و نرم‌افزارهای کاربردی و سطح پایین مثل کتابخانه پردازش تصویر OpenCV برای کاربردهای عملی بخصوص در بخش دفاعی- نظامی ضروری است. OpenCV به دلیل قابلیت انتقال کد و پشتیبانی از سیستم عامل‌های مختلف، به طور گسترده در سامانه‌های نهفته استفاده می‌گردد.

به منظور بالا بردن تحقیقات در زمینه بینایی کامپیوتر، بسیار الزامی بود تا کتابخانه‌ای با توابع بهینه، قابل حمل و رایگان ایجاد شود. زمانیکه OpenCV رسماً راه‌اندازی شد، این هدف اصلی گروه Intel در سال ۱۹۹۹ بود. کتابخانه OpenCV یکی از گسترده‌ترین کتابخانه‌های منبع باز استفاده شده برای پردازش تصویر است که در ابتدا به وسیله تیمی به سرپرستی Gary Bradski توسعه داده شد [۳].

OpenCV شامل بیش از ۵۰۰ الگوریتم بهینه برای تحلیل تصویر و ویدئو است که به میزان زیادی از سوی جامعه محققان و توسعه‌دهندگان بینایی ماشین به عنوان ابزار توسعه پذیرفته شده است. اولین نسخه آلفا OpenCV در سال ۲۰۰۰، پنج نسخه بتای آن بین سال‌های ۲۰۰۱ تا ۲۰۰۵ و نسخه ۱٫۰ OpenCV در سال ۲۰۰۶ منتشر شده‌اند. در اواسط سال ۲۰۰۸، OpenCV حمایت همکاری Willow Garage را به دست آورد و در اکتبر همان سال نسخه ۱٫۱ معرفی شد. نسخه دوم [۴] OpenCV (OpenCV2) که شامل تغییرات زیادی در ارائه واسطی جدید برای زبان ++C بود، در سال ۲۰۰۹ منتشر گردید [۵]-۱۰. آخرین نسخه موجود، OpenCV3 است که در سال ۲۰۱۵ معرفی شده است.

در دهه گذشته، رشد نمایی سرعت پردازنده و ظرفیت حافظه منجر به گسترش چشمگیر تحقیقات در زمینه پیاده‌سازی الگوریتم‌های بینایی کامپیوتر شده است [۱۱]. نوع سخت‌افزارهای استفاده شده برای پردازش تصویر شامل تراشه‌های ASIC، پردازنده DSP، پردازنده ARM، FPGA و غیره می‌باشند. طراحی ASIC بالاترین کارایی را دارد، اما پیچیدگی و هزینه آن‌ها در طراحی خیلی بالا است. بسیاری از بسترهای پردازش تصویر روی سخت‌افزارهای قابل برنامه‌نویسی مثل FPGAها و DSPها [۱۲-۱۵] پایه‌ریزی شده‌اند. چنین بسترهای سخت‌افزاری می‌توانند به بهبودهای پردازشی و زمانی لازم برسند، اما برنامه‌نویسی DSPها و مخصوصاً FPGAها [۱۶-۱۸] می‌تواند کار سختی باشد و برنامه‌نویسان FPGA و DSP باید برنامه‌نویسی با معماری این سخت‌افزارها را یاد بگیرند. به علاوه این بسترهای سخت‌افزاری می‌توانند گران باشند، به طوری که هزینه لازم برای تهیه سخت‌افزارهای FPGA و DSPها جهت انجام عملیات پردازش تصویر چندین برابر هزینه لازم برای خرید PC است [۱۹].

در کاربردهای عملی که نیاز به سخت‌افزارهایی با توان و هزینه پایین است، استفاده از رایانه‌های تک بردی می‌تواند بهترین انتخاب باشد. بزرگ‌ترین فایده رایانه‌های تک بردی قیمت پایین آن‌ها است. رایانه‌های تک بردی، بردهای کوچکی هستند که از نظر معماری با رایانه‌های همه‌منظوره مشابه و شامل پردازنده، RAM، واحد گرافیکی و I/O می‌باشند. به علت آنکه درگاه‌های USB و HDMI در این

پردازش تصویر OpenCV پرداخته شده است. بخش ششم، کارهای صورت گرفته برای بهینه‌سازی روی سخت‌افزار Raspberry Pi 2 را معرفی می‌کند. در بخش هفتم عملکرد سخت‌افزار پیشنهادی ارزیابی خواهد شد. بخش هشتم، نتیجه‌گیری مقاله را بیان می‌کند.

۲- کارهای قبلی

در طول دهه‌های گذشته، کارهای زیادی در زمینه آشکارسازی اهداف کوچک مادون قرمز انجام شده است و پیشرفت‌هایی در این زمینه حاصل شده است. روش‌های مختلفی برای طراحی الگوریتم‌های آشکارسازی اهداف در تصویر مادون قرمز وجود دارد که از جمله مهم‌ترین آن‌ها عبارتند از: اعمال فیلتر باتروث، استفاده از تبدیل موجک، استفاده از شبکه‌های عصبی، ریخت‌شناسی، الگوریتم فرکتال، مدل توجه بصری انسان و غیره [۲۰-۳۲]. آشکارسازی اهداف کوچک در تصویر مادون قرمز با کنتراست پایین پس‌زمینه به وسیله اعمال روش فیلتر بالاگذر باتروث انجام می‌شود [۲۰]. در مراجع [۲۱ و ۲۲] از الگوریتم‌هایی بر مبنای تبدیل موجک برای آشکارسازی اهداف کوچک در تصاویر مادون قرمز استفاده می‌گردد. روش‌هایی برای آشکارسازی خودکار اهداف بر مبنای شبکه‌های عصبی در مراجع [۲۳ و ۲۴] در تصاویر مادون قرمز ارائه شده است. در روش‌هایی بر پایه ریخت‌شناسی [۲۵ و ۲۶] از اطلاعات شکل اهداف کوچک، برای بهبود اهداف و فرونشاندن کلاتر پس‌زمینه استفاده می‌شود. الگوریتم بعدی فرکتال در مراجع [۲۷ و ۲۸]، برای آشکارسازی هدف کوچک کم‌نور در تصاویر مادون قرمز دو باند معرفی شده است. Itti و همکارانش در مرجع [۲۹] مدل توجه بصری را ارائه دادند که این مدل به طور گسترده در زمینه آشکارسازی هدف مورد توجه قرار گرفت. پس از آن تلاش زیادی صورت گرفت تا هدف به وسیله مکانیزم توجه بصری [۳۰-۳۲] آشکارسازی شود. این مقاله از یک الگوریتم بر مبنای توجه بصری جدید استفاده کرده تا اهداف مادون قرمز کم‌نور را آشکارسازی کند. در ابتدا، روش فیلترهای اختلاف گوسی را انتخاب می‌نماید تا نقشه برجستگی را محاسبه کند. سپس در نواحی برجسته که اهداف پنهانی وجود دارند، اهداف به وسیله جستجو از طریق نقشه برجستگی با روش کنترلی WTA و IOR استخراج می‌گردد.

دستگاه‌ها رایج هستند، دستگاه‌های جانبی معمولی مثل صفحه کلید، ماوس و صفحه نمایش می‌توانند به رایانه‌های تک بردی متصل شوند و برد می‌تواند مثل رایانه استفاده گردد. آن‌ها معمولاً به وسیله یک سامانه روی تراشه شامل پردازنده ARM اجرا می‌شوند. چندین مدل از رایانه‌های تک بردی در بازار وجود دارند که به وسیله تولیدکنندگان مختلفی ارائه شده‌اند و مشهورترین نوع آن‌ها، Raspberry Pi است.

این مقاله بستر سخت‌افزاری کم وزن، کم توان و با هزینه پایین را معرفی، بررسی و تحلیل می‌کند که قدرت پردازش مناسبی برای پیاده‌سازی الگوریتم آشکارسازی هدف در تصاویر مادون قرمز را دارد و نتایج استفاده از آن برای آشکارسازی اهداف در تصویر مادون قرمز مورد بررسی و تحلیل قرار می‌گیرد. برای توسعه الگوریتم آشکارسازی هدف روی Raspberry Pi 2 از کتابخانه پردازش تصویر OpenCV2 استفاده شده است. OpenCV روی رایانه‌های تک بردی در زمینه‌های بسیاری مثل کاربردهای نظامی، سامانه‌های امنیتی و جاسوسی، کالیبراسیون دوربین، پردازش تصویر پزشکی، پردازش تصاویر ماهواره‌ای و کاربردهای نقشه‌برداری به طور گسترده استفاده می‌شود. برای ارزیابی قدرت پردازش سخت‌افزار پیشنهادی، با نرم‌افزار سطح بالای MATLAB و با PC روی محیط توسعه نرم‌افزاری Qt همراه با کتابخانه پردازش تصویر OpenCV مقایسه انجام می‌گیرد. همچنین سخت‌افزار پیشنهادی از نظر توان مصرفی، هزینه، وزن و اندازه با PC بررسی و ارزیابی می‌گردد. برای بهینه‌سازی و کاهش زمان اجرا روی سخت‌افزار پیشنهادی از برنامه‌نویسی چندریسه‌ای و خاصیت افزایش فرکانس استفاده می‌شود.

ادامه مقاله به صورت زیر تقسیم‌بندی می‌شود: در بخش دوم به کارهایی که در چندین سال قبل در زمینه آشکارسازی اهداف تصاویر مادون قرمز و همچنین به کارهایی که در زمینه پیاده‌سازی الگوریتم‌های پردازش تصویر روی سامانه‌های نهفته صورت گرفته، پرداخته خواهد شد. در بخش سوم روند کلی پیاده‌سازی الگوریتم آشکارسازی هدف مورد بررسی قرار می‌گیرد. در بخش چهارم الگوریتم آشکارسازی هدف کوچک و کم‌نور تصاویر مادون قرمز معرفی می‌شود. بخش پنجم به بررسی سخت‌افزار پیشنهادی، چگونگی راه‌اندازی آن‌ها و معرفی کتابخانه

۳- روند کلی پیاده‌سازی

شکل (۱) روش پیاده‌سازی الگوریتم آشکارسازی هدف را نشان می‌دهد. اولین مرحله، اخذ تصویر ورودی مادون قرمز است. بعد از این که تصویر ورودی اخذ شد، باید الگوریتم آشکارسازی هدف در تصاویر مادون قرمز را با استفاده از کتابخانه پردازش تصویر OpenCV و زبان برنامه‌نویسی ++C برای پیاده‌سازی روی سخت‌افزار توسعه داد. الگوریتم، روشی بر پایه مکانیسم توجه بصری انسان برای آشکارسازی هدف کم نور است که از فیلترهای اختلاف گوسی برای ایجاد نقشه برجستگی استفاده می‌کند.

بعد از طراحی الگوریتم، باید Raspberry Pi 2 برای پیاده‌سازی الگوریتم آشکارسازی هدف راه‌اندازی شود که برای پیاده‌سازی الگوریتم باید سیستم عامل Raspbian را روی کارت میکرو SD نوشته و Raspberry Pi 2 از روی کارت میکرو SD راه‌اندازی گردد. از SSH برای تبادل اطلاعات بین رایانه و Raspberry Pi 2 روی یک کانال امن استفاده می‌شود که SSH اجازه می‌دهد تا از راه دور به Raspberry Pi متصل شده و دستورات مختلف را اجرا نماید. با استفاده از SSH و دسترسی به خط فرمان Linux، می‌توان کتابخانه‌ها و نرم‌افزارهای مورد نیاز برای پیاده‌سازی را روی Raspberry Pi نصب و الگوریتم آشکارسازی هدف را روی Raspberry Pi 2 پیاده‌سازی کرد. برای مشاهده تصویر خروجی روی PC از پروتکل VNC استفاده می‌شود که اجازه مشاهده و کنترل یک دسکتاپ را از راه دور و از طریق شبکه می‌دهد.

۴- الگوریتم

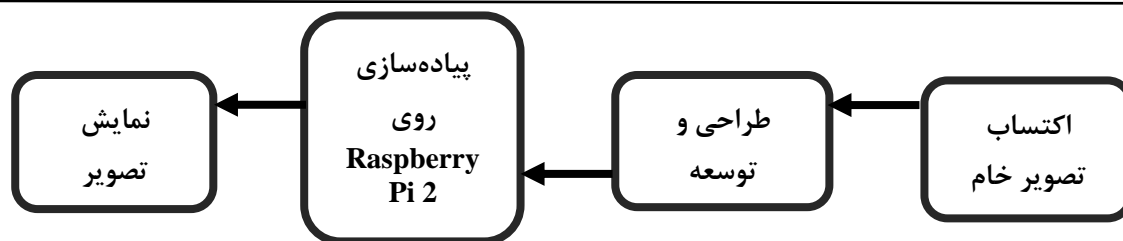
این بخش توسعه الگوریتمی که قابلیت آشکارسازی اهداف کوچک را در تصویر مادون قرمز دارد، نشان می‌دهد. الگوریتم، روشی بر پایه مکانیسم توجه بصری انسان برای آشکارسازی هدف کم نور را ارائه می‌دهد که از فیلترهای اختلاف گوسی برای ایجاد نقشه برجستگی استفاده می‌کند [۲]. شکل (۲) مراحل مختلف پیاده‌سازی الگوریتم را نشان می‌دهد.

گام های اول و دوم محاسبه نقشه برجستگی و استخراج ناحیه برجسته برای اهداف کوچک در تصاویر مادون قرمز است. گام سوم آشکارسازی هدف کم‌نور است که در ادامه هر کدام از گام‌های اول تا سوم بررسی می‌شوند.

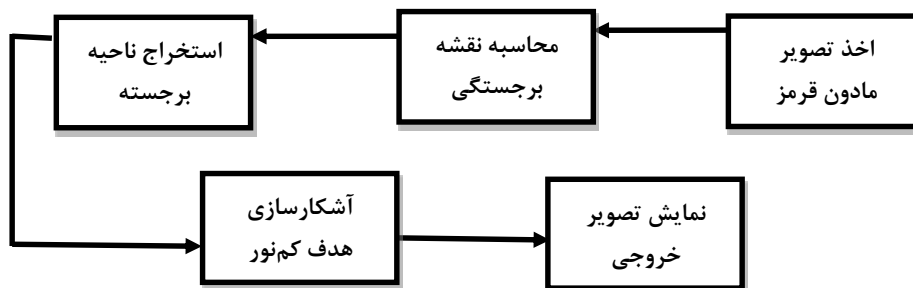
در گذشته این نواحی به وسیله مشخصه‌های اهداف مادون قرمز کم‌نور شناسایی می‌شدند، به طوری که اهداف واقعی شناسایی و هشدارهای کاذب رد می‌شدند.

در چند سال اخیر کارهای زیادی در زمینه پیاده‌سازی الگوریتم‌های پردازش تصویر به وسیله محققان مختلف انجام گرفته است. تحقیقات قبلی معمولاً بر مبنای نوع سخت‌افزار و الگوریتم پردازش تصویر پیاده‌سازی شده تقسیم‌بندی می‌شوند. مرجع [۱۹] سامانه ردیابی انسان را با استفاده از یک بستر نهفته پیاده‌سازی کرده است. بستر غیرنهفته پردازنده دو هسته‌ای اینتل با کلاک ۲،۱۳ گیگاهرتز و حافظه ۴ گیگابایت است و بستر نهفته BeagleBoard xM است که دارای پردازنده DM3730 با کلاک ۱ گیگاهرتز از شرکت TI بوده و ۵۱۲ مگابایت حافظه LPDDR RAM POP را داراست.

مرجع [۳۳] یک پردازنده تصویر بر مبنای ARM Cortex-A9 را معرفی می‌کند که این واحد پردازشی توانایی بالایی در پیاده‌سازی الگوریتم‌های پروتز بینایی دارد. بعد از بهینه‌سازی نرم‌افزاری، پردازنده تصویر می‌تواند تصویر ۶۴۰×۴۸۰ را در مدت ۵۰ میلی ثانیه پردازش کند. مرجع [۳۴] الگوریتم تشخیص عنیبه را با استفاده از کتابخانه بینایی ماشین OpenCV روی برد Beagle Bone-Black که بر مبنای پردازنده ARM کم هزینه و کم توان است، پیاده‌سازی می‌کند و زمان پردازش را کاهش می‌دهد. مرجع [۳۵] الگوریتم همبستگی را روی پردازنده OMAP 3530 که شامل معماری دو هسته‌ای شامل ARM و DSP است، پیاده‌سازی می‌کند که برای توسعه الگوریتم همبستگی از برخی توابع کتابخانه OpenCV استفاده کرده است. مرجع [۳۶] الگوریتمی را روی Raspberry Pi پیاده‌سازی می‌نماید که الگوریتم، تصاویر با کنتراست پایین و تاریک را که به وسیله دوربین Raspberry Pi گرفته است، بهبود می‌دهد تا بخش‌های خاصی از تصویر را شناسایی کند. مرجع [۳۷] پیاده‌سازی سامانه آشکارسازی و ردیابی هدف را روی Raspberry Pi به عنوان یک سامانه نهفته توصیف می‌کند. مرجع [۳۸] روش‌های موجود برای الگوریتم‌های تشخیص چهره و حرکت و کاربردهای آن‌ها را با Raspberry Pi 3 مورد بررسی قرار می‌دهد. الگوریتم‌ها با توابع OpenCV پیاده‌سازی می‌شوند تا عملکرد آن‌ها برای Raspberry Pi 3 بهینه گردد.



شکل (۱) بلوک دیاگرام مراحل مختلف پیاده‌سازی الگوریتم‌های پردازش تصویر.



شکل (۲) مراحل مختلف آشکارسازی هدف در تصاویر مادون قرمز.

DoG ترکیب شده با تصویر مادون قرمز اصلی I کانولوشن شده و نقشه برجستگی S را نتیجه می‌دهد:

$$S(i,j) = |I_{flow}(i,j) - I_{fhigh}(i,j)| \quad (3)$$

که در آن، I_{flow} با روش میانگین‌گیری همسایگی محاسبه می‌شود و I_{fhigh} نسخه بلور (مات) شده گوسی (با استفاده از هسته دو جمله ای 5×5 شکل (۳)) از تصویر برای حذف نویز است.

$$1/256 \times$$

۱	۴	۶	۴	۱
۴	۱۶	۲۴	۱۶	۴
۶	۲۴	۳۶	۲۴	۶
۴	۱۶	۲۴	۱۶	۴
۱	۴	۶	۴	۱

شکل (۳) هسته دو جمله ای 5×5 [۲].

۴-۲- استخراج ناحیه برجسته

روشی بهبود یافته برای استخراج نواحی برجسته در این بخش پیشنهاد شده است که گام‌های تشکیل آن به قرار زیر است: یک پنجره با اندازه $W \times H$ بر روی نقشه برجستگی از چپ به راست و از بالا به پایین حرکت می‌کند و سپس جمع همه مقادیر پیکسل‌ها در پنجره به عنوان "درجه برجستگی" پنجره محاسبه می‌شود. مجموعه پیکسل‌ها در

۴-۱- محاسبه نقشه برجستگی بر پایه فیلتر میان‌گذر DoG ترکیب‌شده

توجه بصری یکی از مشخصه‌های مهم سامانه بصری انسان است که یک هدف یا پیکسل را نسبت به همسایه‌هایش برجسته و توجه انسان را نسبت به آن جلب می‌کند. کاربرد آشکارسازی هدف، ناحیه مورد علاقه یعنی مکانی که هدف در آن وجود دارد به صورت ناحیه برجسته است. آشکارسازی هدف مادون قرمز کم‌نور برای الگوریتم‌های محاسبه نقشه برجستگی [۲۰-۲۱] چالش برانگیز است. برای غلبه بر مشکلات این الگوریتم‌ها در محاسبه نقشه برجستگی تصاویر مادون قرمز، از اختلاف فیلترهای گوسین استفاده شده است. تصویر مادون قرمز را می‌توان به صورت ذیل تقسیم بندی نمود [۲۵]:

$$I(i,j) = I_t(i,j) + I_b(i,j) + I_n(i,j) \quad (1)$$

که در آن، (i,j) مختصات یک پیکسل، I_t مولفه هدف، I_b و I_n به ترتیب مولفه کلاتر و نویز پس‌زمینه را نشان می‌دهند. فیلتر میان‌گذر DoG به صورت رابطه زیر تعریف می‌شود [۲]:

$$\sum_{n=1}^N DoG(i,j, \sigma_1^n, \sigma_2^n) = G(i,j, \sigma_1^n) - G(i,j, \sigma_2^n) \quad (2)$$

که در آن، σ_1 و σ_2 انحراف معیارهای گوسین و $\sigma_1 > \sigma_2$ است. نسبت $k = \sigma_1 / \sigma_2$ باند گذر فیلتر DoG را کنترل می‌کند و n تعداد فیلترهای DoG است. فیلتر میان‌گذر

برای M_P ، M_B و C_{PB} برخی روابط مختلف وجود دارد:

$$M_P > M_B, C_{PB} > C_{Th} \quad (9)$$

$$M_P > M_B, C_{PB} < C_{Th} \quad (10)$$

$$M_P < M_B, C_{PB} > C_{Th} \quad (11)$$

$$M_P < M_B, C_{PB} < C_{Th} \quad (12)$$

که در آن‌ها، C_{Th} آستانه تباین است و با احتمال آشکارسازی و احتمال هشدار کاذب مشخص می‌شود.

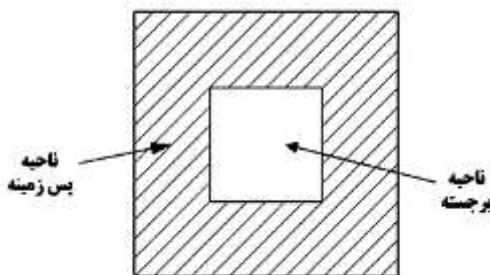
در آزمایشات مقاله مورد بررسی [۲]، نتایج آشکارسازی با C_{Th} برابر ۰,۰۵ رضایت بخش بوده است. در حالت اول، مقدار میانگین خاکستری ناحیه برجسته بزرگ‌تر از ناحیه پس‌زمینه است و تباین خاکستری بین آنها همچنین بزرگ‌تر از آستانه تباین می‌باشد که این امر با مشخصات هدف IR کم‌نور مطابق است.

بنابراین، ناحیه برجسته به عنوان ناحیه هدف واقعی در نظر گرفته می‌شود. در حالت دوم، اگرچه مقدار میانگین خاکستری ناحیه برجسته بزرگ‌تر از ناحیه پس‌زمینه است، لیکن اختلاف بین آن‌ها کافی نیست. در این مورد، این دو ناحیه همگن در نظر گرفته می‌شوند. در حالت‌های سوم و چهارم، مقدار خاکستری میانگین ناحیه برجسته کم‌تر از ناحیه پس‌زمینه است. در این دو مورد، بر طبق مشخصات هدف IR کم‌نور، ناحیه برجسته به عنوان ناحیه غیرهدف در نظر گرفته می‌شود.

۵- سخت‌افزار و نرم‌افزار

۵-۱- سخت‌افزار پیشنهادی

سخت‌افزار کم هزینه، کم توان و کم وزن که برای پیاده‌سازی الگوریتم‌ها مورد استفاده قرار می‌گیرد، Raspberry Pi 2 است. Raspberry-Pi [۳۹-۴۱]، یک SBC یا رایانه تک بردی است و به اندازه کارت اعتباری بوده که توسط بنیاد Raspberry Pi و به قصد ارتقای آموزش مفاهیم پایه رایانه در مدارس تهیه شده است.



شکل (۴) شناسایی ناحیه هدف واقعی [۲].

پنجره با θ مشخص می‌گردد. درجه برجستگی S_θ به صورت زیر مشخص می‌شود:

$$S_\theta = \sum_{(i,j) \in \theta} S(i,j) \quad (4)$$

که در آن، $S(i,j)$ مقدار برجستگی پیکسل در موقعیت (i,j) را نشان می‌دهد. در نتیجه، از نقشه برجستگی S ، r مقدار برجستگی $S_{\theta_1}, \dots, S_{\theta_r}$ بدست می‌آید. کم‌ترین و بیشترین مقدار S_θ از r درجه برجستگی یافت می‌شود و به عنوان مرجع برای تنظیم آستانه Th در نظر گرفته می‌شود:

$$Th = \frac{K \times S_{\theta max} + S_{\theta min}}{K+1} \quad (5)$$

که در آن، k یک عدد ثابت و $k > 1$ است. انتخاب k در محدوده [۲-۷] برای تصاویر مادون قرمز دارای هدف کم‌نور مناسب است. مقادیر درجه برجستگی $S_{\theta_1}, \dots, S_{\theta_r}$ توسط آستانه Th قطعه‌بندی می‌شود. اگر مقدار هر کدام از درجه برجستگی‌ها بزرگ‌تر از Th بود، ناحیه متناظر یک ناحیه برجسته است و به عنوان ناحیه مورد علاقه در نظر گرفته می‌شود. به عبارت دیگر، احتمال وجود اهداف کم‌نور در آن ناحیه زیاد است.

۴-۳- آشکارسازی هدف کم‌نور

بعد از اینکه یک ناحیه برجسته انتخاب شد، بایستی تصمیم گرفته شود که این ناحیه وابسته به ناحیه هدف است یا خیر؟ فرض می‌شود که ناحیه برجسته یک ناحیه هدف است. P مجموعه پیکسل‌های آن را نشان می‌دهد. N_P تعداد پیکسل‌ها و M_P میانگین مقادیر پیکسل‌های ناحیه برجسته را مشخص می‌کنند.

$$M_P = \frac{1}{N_P} \sum_{(i,j) \in P} I(i,j) \quad (6)$$

همانطور که در شکل (۴) نشان داده شده است، پنجره کوچک ناحیه برجسته و پنجره بزرگ ناحیه بسط داده شده را نشان می‌دهند. ناحیه بین این دو پنجره به عنوان پس‌زمینه در نظر گرفته می‌شود و B مجموعه پیکسل‌های پس‌زمینه است. میانگین پیکسل‌های ناحیه پس‌زمینه به طریق زیر محاسبه می‌شود که در آن N_B تعداد پیکسل‌های پس‌زمینه است:

$$M_B = \frac{1}{N_B} \sum_{(i,j) \in B} I(i,j) \quad (7)$$

که در آن، C_{PB} تباین خاکستری بین هدف و پس‌زمینه را مشخص می‌کند و به صورت زیر تعریف می‌گردد:

$$C_{PB} = \frac{|M_P - M_B|}{|M_P + M_B|} \quad (8)$$

۴-۵- معرفی کتابخانه پردازش تصویر OpenCV

OpenCV یک کتابخانه چندسکویی منبع باز برای پردازش تصویر بلادرنگ است که اولین نسخه آن در سال ۲۰۰۰ تحت مجوز BSD منتشر شد. OpenCV شامل الگوریتم‌های بینایی کامپیوتر زیادی است که توسط سازمان Intel با استفاده از زبان‌های برنامه‌نویسی C و C++ توسعه داده شده است. توسعه‌های روی رابط‌های Python، Ruby، MATLAB و دیگر زبان‌ها وجود دارد. OpenCV برای بسیاری از سیستم عامل‌ها مثل GNU/Linux، OS X، Windows، Android، iOS و غیره در دسترس است. دلیل اصلی ایجاد کتابخانه OpenCV این است که به کاربردهای تجاری بینایی کامپیوتر در زمینه‌های رابط انسان رایانه، رباتیک، نظارت، بیومتریک و امنیت از طریق فراهم کردن زیرساخت رایگان و باز و همچنین به برنامه‌های بینایی نسبتاً پیچیده کمک کند تا به سرعت ایجاد شوند. OpenCV یک ساختار مدولار دارد که شامل مدول‌های زیادی است:

- Core: شامل ساختمان داده‌هایی مثل آرایه‌های چندبعدی و توابع پایه است که به وسیله ماژول‌های دیگر مورد استفاده قرار می‌گیرد.
- Imgproc: ماژول پردازش تصویر که شامل فیلترینگ تصویر خطی و غیرخطی، تغییر هندسه تصویر، تبدیل فضای رنگی، هیستوگرام و غیره است.
- Video: ماژول تحلیل ویدئویی شامل تخمین حرکت، کاهش پس‌زمینه و ردیابی اشیاء است.
- Calib3d: ماژول شامل کالیبراسیون دوربین، تخمین هندسه دو‌نمایی و توابع استریو است.
- Features2d: شامل تشخیص گره‌ها و توصیف‌گرهای ویژگی نقطه‌ای و چارچوب‌های انطباق ویژگی نقطه‌ای است.
- Objdetect: شامل توابع تشخیص‌دهنده شی از قبیل تشخیص‌دهنده‌های صورت، انسان، ماشین، لیوان و غیره است.

• highGUI: شامل توابعی برای خواندن و نوشتن ویدئو و تصویر همراه با توابع واسط کاربری ساده است.

و همچنین ماژول‌های کمکی دیگر مثل FLANN، اتصالات python، توابع یادگیری ماشین و غیره در کتابخانه وجود دارند. در این مقاله از توابع کتابخانه OpenCV2 برای توسعه الگوریتم آشکارسازی هدف روی Raspberry Pi 2 و PC استفاده شده است [۱۰، ۹، ۴].

۶-۶- بهینه‌سازی

در این قسمت به معرفی کارهایی که در زمینه کاهش زمان حاصل از پیاده‌سازی الگوریتم آشکارسازی هدف روی Raspberry Pi 2 پرداخته می‌شود. بهینه‌سازی در اجرای هم‌زمان کدها روی هسته‌های مختلف پردازنده با استفاده از کتابخانه Pthread و استفاده از خاصیت افزایش فرکانس انجام می‌گردد.

۶-۱- استفاده از کتابخانه Pthread

Raspberry Pi 2 دارای پردازنده ۴ هسته‌ای ARM Cortex-A7 است. برای همین منظور می‌توان از برنامه‌نویسی چندریسه‌ای با استفاده از کتابخانه Pthread استفاده کرد تا زمان اجرا را تا حدودی کاهش داد. در فرآیند چندریسه‌ای با محیط پردازنده چند هسته‌ای، هر ریسره در برنامه می‌تواند به صورت هم‌زمان روی یک هسته جدا اجرا شود.

چندریسه‌ای همچنین شامل مفهوم هم‌زمانی نیز است تا جریان برنامه را کنترل کند. چهار مدل هم‌زمانی شامل قفل mutex، قفل‌های خواندن/نوشتن، متغیرهای شرطی و سمافورها وجود دارند. قفل‌های mutex فقط به یک ریسره در یک زمان اجازه می‌دهد تا یک بخش مشخصی از کد را اجرا کند یا به داده مخصوصی دسترسی داشته باشد. قفل خواندن/نوشتن اجازه خواندن هم‌زمان و نوشتن منحصربفرد با منابع اشتراکی محافظت شده را می‌دهد. متغیرهای شرطی ریسره‌ها را مسدود می‌کنند تا یک شرایط خاصی صحیح گردد. شمارش سمافورها معمولاً دسترسی به منابع را هماهنگ می‌کند. شمارش محدود به تعداد ریسره‌هایی است که می‌تواند به وسیله سمافور در دسترس باشد. وقتی که شمارش حاصل شد، سمافور مسدود می‌گردد.

۶-۲- استفاده از خاصیت افزایش فرکانس

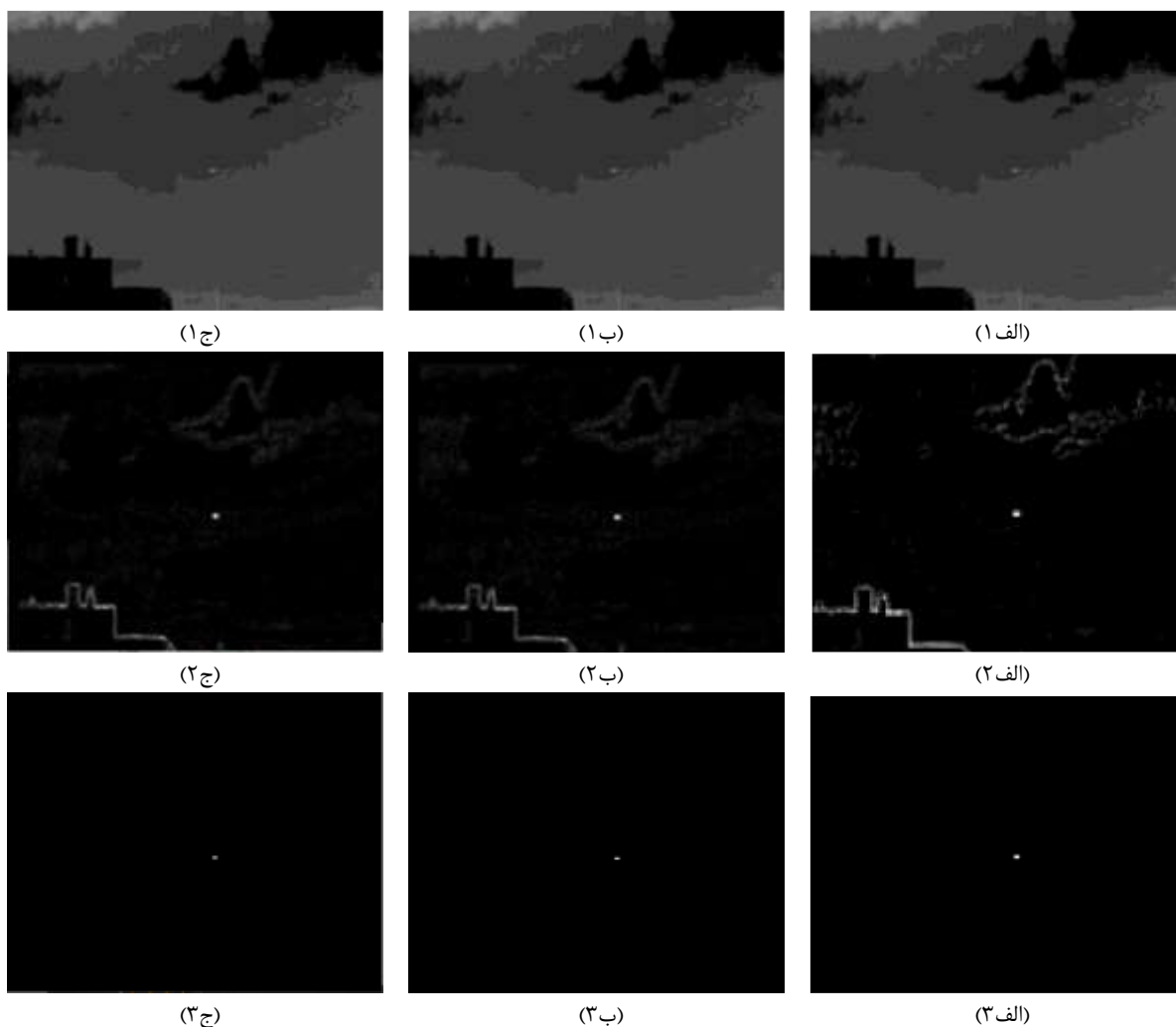
افزایش فرکانس روشی است که منجر به بالا رفتن سرعت اجزای سخت‌افزار Raspberry Pi می‌شود و باعث می‌گردد پردازنده Raspberry-Pi سریع‌تر از فرکانس سازنده کار کند. هدف از افزایش فرکانس افزایش سرعت و عملکرد سخت‌افزار است. با بالا بردن کلاک تعداد عملیات در هر ثانیه نیز افزایش می‌یابد و سرعت انجام عملیات بالا می‌رود. افزایش فرکانس ممکن است باعث کاهش طول عمر برد

کتابخانه پردازش تصویر OpenCV همراه با زبان برنامه‌نویسی C++ و روی برد Raspberry Pi 2 با محیط توسعه نرم‌افزاری Qt و توابع کتابخانه پردازش تصویر OpenCV همراه با زبان برنامه‌نویسی C++ پیاده‌سازی می‌شود. شکل‌های (۶)، (۷) و (۸) نتایج اعمال الگوریتم آشکارسازی هدف را برای یک فریم مادون قرمز از سه صحنه متفاوت هدف در حال پرواز با ابعاد یکسان 501×731 پیکسل، نشان می‌دهند. این تصاویر با همکاری دانشگاه جامع امام حسین (ع) تهیه شده‌اند. زمان اجرا برای الگوریتم آشکارسازی هدف تصویر مادون قرمز برای شکل‌های (۶)، (۷) و (۸) روی PC با محیط توسعه نرم‌افزاری در جدول (۱) نشان داده شده است.

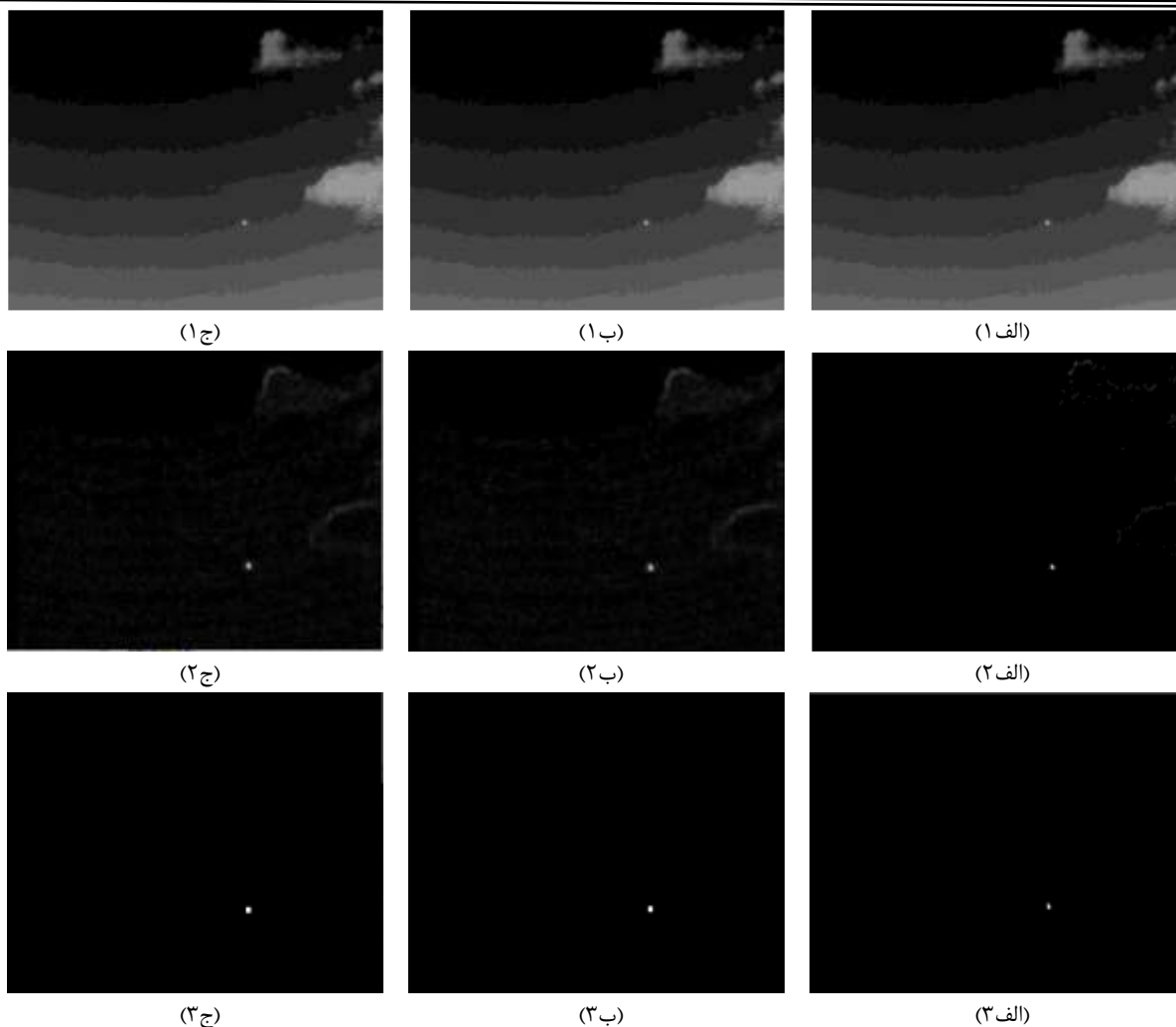
Raspberry Pi شود. اگر افزایش فرکانس در یک سطح خاص منجر به عدم ثبات سامانه شود از افزایش فرکانس با فرکانس کمتر باید استفاده شود.

۷- نتایج

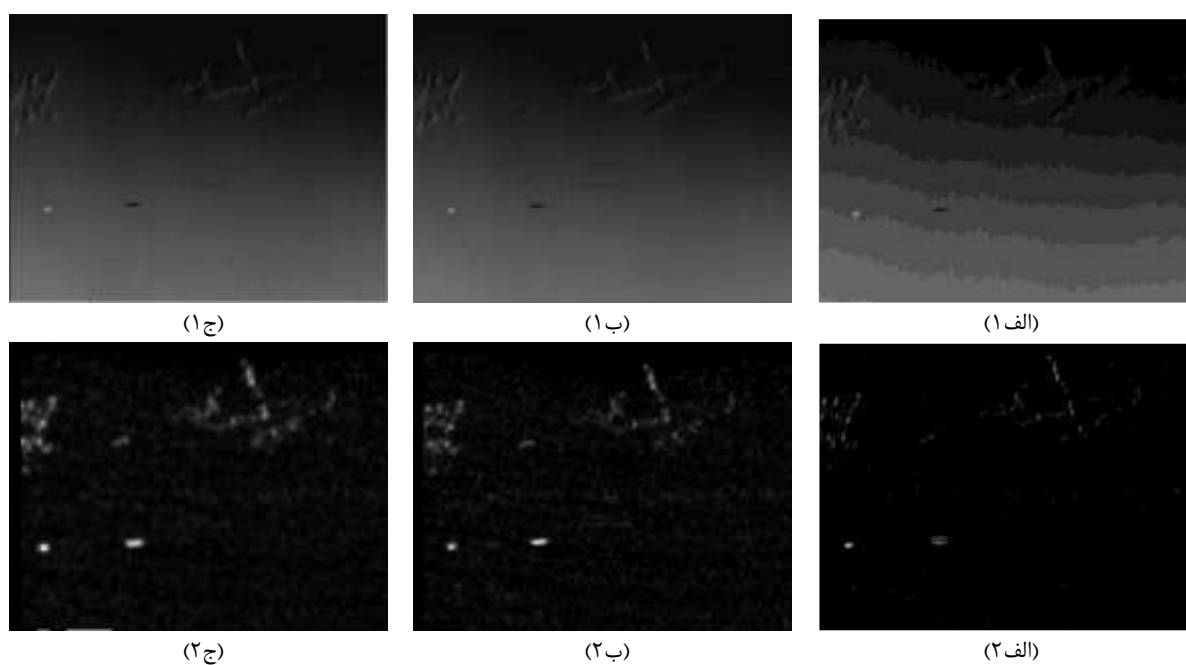
این بخش، سخت‌افزار پیشنهادی را از نظر زمان اجرا، توان مصرفی، وزن، هزینه با PC مورد مقایسه قرار می‌دهد. برای مقایسه زمان اجرا، PC با دو محیط نرم‌افزاری و کتابخانه‌های پردازش تصویر مختلف با برد Raspberry Pi 2 مورد مقایسه قرار می‌گیرد. الگوریتم آشکارسازی هدف تصویر مادون قرمز روی PC با محیط توسعه نرم‌افزاری و توابع کتابخانه پردازش تصویر MATLAB، روی PC با محیط توسعه نرم‌افزاری Qt و توابع

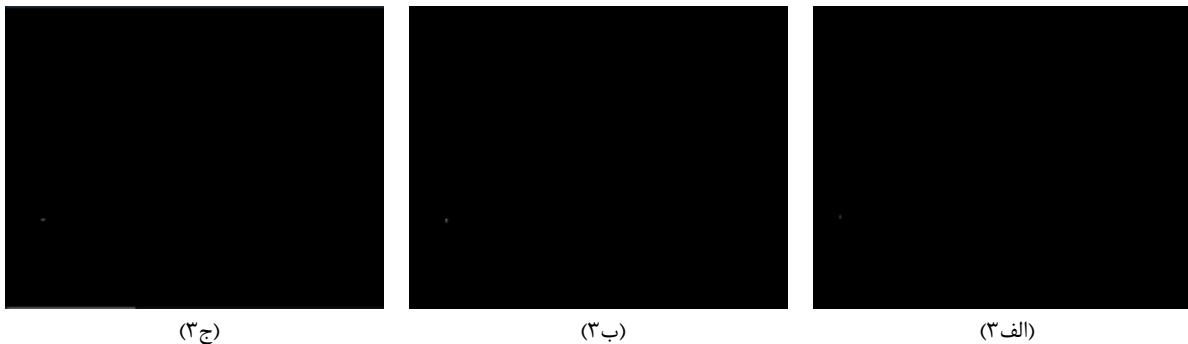


شکل (۶) نتایج پیاده‌سازی الگوریتم آشکارسازی هدف تصویر مادون قرمز روی سه محیط: الف) پیاده‌سازی روی PC با محیط نرم‌افزاری MATLAB، ب) پیاده‌سازی روی PC با محیط نرم‌افزاری Qt و ج) پیاده‌سازی روی Raspberry Pi 2 (اندیس ۱ تصویر ورودی، اندیس ۲ نتیجه اعمال الگوریتم آشکارسازی هدف قبل از آستانه گذاری و اندیس ۳ نتیجه الگوریتم آشکارسازی هدف بعد از آستانه گذاری).



شکل (۷) نتایج پیاده‌سازی الگوریتم آشکارسازی هدف تصویر مادون قرمز روی سه محیط: الف) پیاده‌سازی روی PC با محیط نرم‌افزاری MATLAB، ب) پیاده‌سازی روی PC با محیط نرم‌افزاری Qt و ج) پیاده‌سازی روی Raspberry Pi 2 (اندیس ۱ تصویر ورودی، اندیس ۲ نتیجه الگوریتم آشکارسازی هدف قبل از آستانه‌گذاری و اندیس ۳ نتیجه الگوریتم آشکارسازی هدف بعد از آستانه‌گذاری).





شکل (۸) نتایج پیاده‌سازی الگوریتم آشکارسازی هدف تصویر مادون قرمز روی سه محیط: الف) پیاده‌سازی روی PC با محیط نرم‌افزاری MATLAB، ب) پیاده‌سازی روی PC با محیط نرم‌افزاری Qt و ج) پیاده‌سازی روی Raspberry Pi 2 (اندیس ۱ تصویر ورودی، اندیس ۲ نتیجه الگوریتم آشکارسازی هدف قبل از آستانه‌گذاری و اندیس ۳ نتیجه الگوریتم آشکارسازی هدف بعد از آستانه‌گذاری).

جدول (۱) زمان اجرا برای سه تصویر ورودی.

زمان اجرا تصویر سوم (برحسب میلی ثانیه)	زمان اجرا تصویر دوم (برحسب میلی ثانیه)	زمان اجرای تصویر اول (برحسب میلی ثانیه)	زمان اجرا تصاویر بسترها
۱۶۷۱،۵۴	۱۶۷۱،۳۷	۱۶۷۰،۲۱	MATLAB
۲۹،۶۷	۲۸،۸۸	۲۹،۶	PC (با محیط توسعه نرم‌افزاری Qt)
۲۶۰،۷۹	۲۵۵،۵۲	۲۵۴،۴۳	Raspberry Pi 2

می‌گیرد. علاوه بر این توابع، کتابخانه OpenCV برای پردازش تصویر بلادرنگ ایجاد شده‌اند.

در نتیجه برنامه نوشته شده بر روی OpenCV سریع‌تر از MATLAB اجرا می‌شوند. زمان اجرای PC با محیط توسعه نرم‌افزاری Qt دارای بهبود ۸،۵ برابری نسبت به برد Raspberry Pi 2 است که دلیل این اختلاف در قدرت پردازش بالای PC نسبت به Raspberry Pi 2 و همچنین بهینه بودن کتابخانه OpenCV برای معماری Intel است. جدول (۲) دو سخت‌افزار Raspberry Pi 2 و PC را از نظر فرکانس کاری، قیمت، وزن، توان مصرفی و غیره مقایسه می‌کند. همان‌طور که از جدول (۲) مشخص است از نظر قدرت پردازش PC قدرت پردازش بیشتری نسبت به برد Raspberry Pi 2 دارد، ولی اگر سخت‌افزارها از نظر توان، وزن و قیمت مقایسه شوند، مشاهده می‌گردد که برد Raspberry Pi 2 از نظر توان ۶،۵ برابر توان کمتری نسبت به PC مصرف می‌کند. از نظر وزن برد Raspberry Pi 2 تقریباً ۴۵ برابر دارای وزن کمتری نسبت به PC است. همچنین قیمت برد Raspberry Pi 2، ۱۷ برابر کمتر از PC می‌باشد. پس بنابراین در مکان‌هایی که نیاز به توان پایین، وزن کم، اندازه کوچک و صرفه جویی در هزینه است، قطعاً

همچنین توابع کتابخانه پردازش تصویر MATLAB، روی PC با محیط توسعه نرم‌افزاری Qt و توابع کتابخانه پردازش تصویر OpenCV با زبان برنامه‌نویسی C++ و روی برد Raspberry Pi 2 با محیط توسعه نرم‌افزاری Qt و توابع کتابخانه پردازش تصویر OpenCV با زبان برنامه‌نویسی C++ در جدول (۱) نشان داده شده است.

عملکرد الگوریتم آشکارسازی هدف تصویر مادون قرمز روی محیط نرم‌افزاری و سخت‌افزاری مختلف نشان می‌دهد که متوسط زمان اجرا برای MATLAB، PC و برد Raspberry Pi 2 به ترتیب ۱۶۷۰، ۲۹،۴ و ۲۵۷ میلی‌ثانیه است. PC با محیط نرم‌افزاری Qt تقریباً ۵۷ برابر زمان اجرا را نسبت به MATLAB بهبود داده است. همچنین Raspberry Pi 2 دارای بهبود زمانی ۶،۵ برابری نسبت به MATLAB است. دلیل اختلاف زمانی بالای MATLAB نسبت به OpenCV این است که MATLAB از Java و C ساخته شده است. بنابراین زمانی که برنامه MATLAB اجرا می‌شود، رایانه کدها را جمع‌آوری کرده و کدها را به Java تبدیل و اجرا می‌کند. در صورتی که OpenCV کتابخانه‌ای از توابع نوشته شده با C/C++ است، بنابراین در OpenCV پردازش بیشتری در سیکل پردازشی رایانه بر روی داده‌ها انجام

بهرتر است از برد Raspberry Pi 2 استفاده شود. برای ارزیابی عملکرد الگوریتم و کتابخانه‌های پردازش تصویر MATLAB و OpenCV از دو شاخص ارزیابی یعنی نسبت سیگنال به کلتر (SCR) و ضریب حذف پس‌زمینه (BSF) استفاده می‌شود که به صورت زیر تعریف می‌شوند:

عملکرد الگوریتم و کتابخانه‌های پردازش تصویر MATLAB و OpenCV از دو شاخص ارزیابی یعنی نسبت سیگنال به کلتر (SCR) و ضریب حذف پس‌زمینه (BSF) استفاده می‌شود که به صورت زیر تعریف می‌شوند:

$$SCR = \frac{(S/C)_{out}}{(S/C)_{in}} \quad (13)$$

$$BSF = \frac{C_{in}}{C_{out}} \quad (14)$$

در روابط (۱۳) و (۱۴)، S دامنه سیگنال و C انحراف استاندارد از کلتر را مشخص می‌کنند و زیرنویس‌های in و out تصاویر قبل و بعد از آشکارسازی را بیان می‌کنند [۴۸]. جدول (۳) نتایج مربوط به محاسبه شاخص‌های ارزیابی

جدول (۲) مقایسه سخت‌افزارها.

Asus X450LD	Raspberry Pi 2	مشخصات سخت‌افزار
1.6 GHz up to 2.3 GHz Intel Core i5 4200U	900 MHz Quad-Core ARM Cortex-A7	واحد پردازش مرکزی
NVIDIA GeForce 820M	Broadcom VideoCore IV @ 250 MHz	واحد پردازش گرافیکی
8 GB	1 GB	حافظه (SDRAM)
2 × USB3.0 1 × USB2.0	4 × USB2.0	درگاه USB
10/100/1000 Mbit/s Ethernet	10/100 Mbit/s Ethernet	شبکه روی تراشه
19 V, 3.42 A, 65 W	5 V, 2A, 10W	توان
2.1 Kg	45 g	وزن
34.8 cm × 24.1 cm × 3.17 cm	85.6 mm × 56.5 mm × 17 mm	ابعاد
~ 600\$	35\$	قیمت

جدول (۳) شاخص‌های ارزیابی SCR و BSF.

تصویر ۳	تصویر ۲	تصویر ۱	شاخص‌ها	تصاویر بسترها
۶,۶۹۰۲	۹,۴۲۰۱	۵,۲۳۴۳	SCR	MATLAB
۴,۱۰۳۰	۷,۵۰۶۷	۲,۸۴۲۱	BSF	
۴,۶۰۹۲	۶,۵۲۶۵	۴,۵۳۰۱	SCR	PC (با محیط توسعه نرم‌افزاری Qt)
۲,۸۲۶۸	۵,۲۰۰۸	۲,۴۵۹۷	BSF	
۴,۶۰۹۲	۶,۵۲۶۵	۴,۵۳۰۱	SCR	Raspberry Pi 2
۲,۸۲۶۸	۵,۲۰۰۸	۲,۴۵۹۷	BSF	

Pi 2 بالا رود. با افزایش فرکانس کلاک، زمان اجرای مطابق جدول (۴)، ۱۵ درصد کاهش پیدا می‌کند. با اعمال همزمان بهینه‌سازی برنامه‌نویسی چندریسه‌ای و افزایش فرکانس روی برد Raspberry Pi 2، بهبود ۴۰ درصدی در زمان حاصل می‌شود (مطابق جدول (۴)). بعد از استفاده از برنامه‌نویسی چندریسه‌ای و استفاده از تکنیک افزایش فرکانس زمان اجرا روی Raspberry Pi 2 تقریباً به بهبود ۱۱ برابری نسبت به نرم‌افزار MATLAB می‌رسد. همچنین تقریباً ۵ برابر نسبت به PC با محیط نرم‌افزاری Qt و کتابخانه پردازش تصویر OpenCV عملکرد زمانی بهتری ارائه می‌دهد.

با اعمال بهینه‌سازی‌های مورد نظر به الگوریتم آشکارسازی هدف و سخت‌افزار Raspberry Pi می‌توان زمان اجرا برد Raspberry Pi 2 را کاهش داد. به علت آنکه Raspberry Pi 2 دارای پردازنده ۴ هسته‌ای ARM Cortex-A7 است می‌توان از برنامه‌نویسی چندریسه‌ای برای کاهش زمان اجرا استفاده کرد. همان طور که در جدول (۴) نشان داده شده است زمان اجرا برای برنامه‌نویسی از ۲ ریشه تا ۴ ریشه نسبت به ۱ ریشه کاهش پیدا می‌کند که برای ۴ ریشه تقریباً به ۳۲ درصد بهبود در زمان اجرا رسیده است. همچنین به دلیل آنکه برد توانایی افزایش فرکانس را دارد، می‌توان سرعت کلاک Raspberry Pi 2 را از ۹۰۰ مگاهرتز به ۱ گیگاهرتز افزایش داد تا سرعت پردازنده Raspberry

جدول (۴) زمان اجرای (بر حسب میلی ثانیه) Raspberry Pi 2 بعد از بهینه‌سازی.

افزایش فرکانس + چندریسه‌ای	افزایش فرکانس	۴ ریشه	۳ ریشه	۲ ریشه	۱ ریشه	بهینه‌سازی ورودی
۱۵۰٫۶۶	۲۱۹٫۲۰	۱۷۰٫۸۷	۲۰۰٫۸۲	۲۱۶٫۳۵	۲۵۴٫۴۳	تصویر ۱
۱۵۱٫۹۳	۲۲۰٫۵۷	۱۷۲٫۱۶	۲۰۱٫۷۶	۲۱۷٫۶۰	۲۵۵٫۵۲	تصویر ۲
۱۵۲٫۱۸	۲۲۰٫۹۸	۱۷۲٫۲۰	۲۰۱٫۵۹	۲۱۷٫۷۸	۲۶۰٫۷۹	تصویر ۳
۱۵۱٫۲۹	۲۱۹٫۸۸	۱۷۱٫۵۱	۲۰۱٫۲۹	۲۱۶٫۹۷	۲۵۴٫۹۷	میانگین

مقایسه شود، مشاهده می‌گردد که برد Raspberry Pi 2 از نظر توان، وزن و قیمت بهبود قابل توجه‌ای نسبت به PC دارد. همچنین استفاده از برنامه‌نویسی چندریسه‌ای و خاصیت افزایش فرکانس به عنوان روش‌های بهینه‌سازی روی Raspberry Pi 2، بهبود ۴۰ درصد در زمان اجرا را نتیجه می‌دهد. زمان اجرا روی Raspberry Pi 2 بعد از بهینه‌سازی تقریباً ۱۱ برابر نسبت به نرم‌افزار MATLAB بهبود می‌یابد.

برای کارهای آینده می‌توان برای بالا بردن دقت آشکارسازی از الگوریتم‌هایی با دقت آشکارسازی بالاتر استفاده شود که افزایش دقت پیچیده شدن الگوریتم را به دنبال خواهد داشت. همچنین با توجه به اختلاف زمان اجرای بین رایانه تک بردی پیشنهادی مقاله و PC می‌توان برای کاهش زمان اجرا از رایانه‌های تک بردی با قدرت پردازش بالاتر مثل Raspberry Pi 3، Beagle Bone و غیره استفاده گردد که با بالا رفتن قدرت پردازش، هزینه سخت‌افزار افزایش خواهد یافت.

۸- نتیجه‌گیری

در این مقاله، یک سخت‌افزار با وزن پایین، کم هزینه و کم توان برای پیاده‌سازی الگوریتم پردازش تصویر معرفی و مورد بررسی و تحلیل قرار گرفت. برای مقایسه عملکرد سخت‌افزار پیشنهادی با PC و MATLAB از الگوریتم آشکارسازی هدف کوچک و کم‌نور در تصاویر مادون قرمز که با استفاده از فیلترهای DoG و بر پایه سامانه بصری انسان، نقشه برجستگی اهداف را ایجاد می‌کند، استفاده شده است. نتایج نشان می‌دهد که هر چند استفاده از نرم‌افزارهای سطح بالا مثل MATLAB دارای شاخص‌های ارزیابی SCR و BSF بالاتری نسبت به استفاده از کتابخانه OpenCV است، اما نتایج زمان اجرا نشان می‌دهد که سخت‌افزار پیشنهادی زمان اجرا را نسبت به نرم‌افزارهای سطح بالا مثل MATLAB بهبود می‌دهد. PC با محیط نرم‌افزاری Qt و توابع کتابخانه OpenCV زمان اجرا سریع‌تری نسبت به سخت‌افزار پیشنهادی دارد. ولی اگر سخت‌افزار پیشنهادی مقاله با PC از نظر توان، وزن و قیمت

۹- مراجع

- [15] D. Jinghong, D. Yaling and L. Kun, "Development of Image Processing System Based on DSP and FPGA", The Eight International Conference on Measurement and Instruments, pp.791-794, 2007.
- [16] M. Ali, E. Stotzer, F. D. Igual and R. A. Geijn, "Level-3 BLAS on the TI C6678 Multi-Core DSP", IEEE Conference on Computer Architecture and High Performance Computing, pp.179-186, 2012.
- [17] L. Yan, T. Zhang and S. Zhong, "A DSP/FPGA-Based Parallel Architecture for Real-Time Image Processing", The Sixth World Congress on Intelligent Control and Automation, pp.10022-10025, 21-23 June 2006.
- [18] B. Ramesh, A. Bhardwaj, J. Richardson, A. D. George and H. Lam, "Optimization and Evaluation of Image and Signal Processing Kernels on the TI C6678 Multi-Core DSP", 2014 IEEE High Performance Extreme Computing Conference, pp.1-6, 9-11 September 2014.
- [19] S. K. Teoh, V. V. Yap, C. S. Soh and P. Sebastian, "Implementation and Optimization of Human Tracking System using ARM Embedded Platform", 4th International Conference on Intelligent and Advanced Systems, pp.353-356, 2012.
- [20] L. Yang, J. Yang and K. Yang, "Adaptive Detection for Infrared Small Target under Sea-Sky Complex Background", Electron Letters, Vol.40, No.17, pp.1083-1085, 2004.
- [21] F. A. Sadjadi, "Infrared Target Detection with Probability Density Functions of Wavelet Transform Sub-bands", Applied Optics, Vol.43, No.2, pp.315-323, 2004.
- [22] X. Wang and Z. M. Tang, "Combining Wavelet Packets with Higher-Order Statistics for Infrared Small Targets Detection, Infrared Laser Eng, Vol.38, No.5, pp.915-920, 2009.
- [23] J. F. Khan, M. S. Alam and S. M. Bhuiyan, "Automatic Target Detection in Forward-Looking Infrared Imagery via Probabilistic Neural Networks", Applied Optics, Vol.48, No.3, pp.464-476, 2009.
- [24] P. Zhang and J. Li, "Neural-Network-Based Single-Frame Detection of Dim Spot Target in Infrared Images", Optical Engineering, Vol.46, No.7, pp.076401-076401, 2007.
- [25] J. F. Khan and M. S. Alam, "Target Detection in Cluttered Forward-Looking Infrared Imagery", Optical Engineering, Vol.44, No.7, pp.076404-076404, 2005.
- [26] M. Zeng, J. X. Li and Z. Peng, "The Design of Top-Hat Morphological Filter and Application to Infrared Target Detection", Infrared Physics & Technology, Vol.48, No.1, pp.67-76, 2006.
- [27] Y. Q. Sun, J. W. Tian and J. Liu, "Novel Method on Dual-Band Infrared Image Fusion for Dim Small Target Detection", Optical
- [1] H. Deng, X. Sun, M. Liu, C. Ye and X. Zhou, "Entropy-Based Window Selection for Detecting Dim and Small Infrared Targets", Pattern Recognition, Vol.61, pp.66-71, 2017.
- [2] X. Wang, G. Lv and L. Xu, "Infrared Dim Target Detection Based on Visual Attention", Infrared Physics & Technology, Vol.55, No.6, pp.513-521, 2012.
- [3] H. Sugano and R. Miyamoto, "Highly Optimized Implementation of OpenCV for the Cell Broadband Engine", Computer Vision and Image Understanding, Vol.114, No.11, pp.1273-1281, 2010.
- [4] R. Laganière, "OpenCV 2 Computer Vision Application Programming Cookbook", Packt Publishing Ltd, 2011.
- [5] H. Jina, "Review Paper on Industrial Automation based on OpenCV", International Journal of Emerging Trends in Electrical and Electronics, Vol.2, Issue.1, pp.92-95, 2013.
- [6] S. Matuska, R. Hudec and M. Benco, "The Comparison of CPU Time Consumption for Image Processing Algorithm in Matlab and OpenCV", In ELEKTRO, pp.75-78, 2012.
- [7] M. Marengoni and D. Stringhini, "High Level Computer Vision using OpenCV", 24th SIBGRAPI Conference on In Graphics, Patterns and Images Tutorials, pp.11-24, 2011.
- [8] P. N. Druzhkov, V. L. Erukhimov, N. Y. Zolotykh, E. A. Kozinov, V. D. Kustikova, I. B. Meerov and A. N. Polovinkin, "New Object Detection Features in the OpenCV Library", Pattern Recognition and Image Analysis, Vol.21, No.3, pp.384-386, 2011.
- [9] G. B. García, O. D. Suarez, J. L. E. Aranda, J. S. Tercero, I. S. Gracia and N. V. Enano, "Learning Image Processing with OpenCV", Packt Publishing Ltd, 2015.
- [10] A. Kaehler and G. Bradski, "Learning OpenCV", O'Reilly Media, 2014.
- [11] Q. Yu, H. H. Cheng, W. W. Cheng and X. Zhou, "Ch OpenCV for Interactive Open Architecture Computer Vision", Advances in Engineering Software, Vol.35, No.8, pp.527-536, 2004.
- [12] X. Yuanfang and S. Xia, "System Design for Real-Time Image Processing Based on Multi-Core DSP", Journal of Networks, Vol.9, No.11, pp.3143-3150, 2014.
- [13] W. Wasfy and H. Zheng, "General Structure Design for Fast Image Processing Algorithms Based upon FPGA DSP Slice", International Conference on Medical Physics and Biomedical Engineering, pp.690-697, 2012.
- [14] M.S. Kumar and D. Nedumaran, "Development of Image Enhancement Algorithm for Fingerprint Images in TMS320C6416 DSK", IEEE Conf. on Computational Intelligence in Biometrics and Identity Management, pp.1-6, 2013.

- Computers & Electrical Engineering, Vol.44, pp.153-71, 2015.
- [41] D. S. Bölsche and A. M. Schön, "A Raspberry in Sub-Saharan Africa? Chances and Challenges of Raspberry Pi and Sensor Networking in Humanitarian Logistics", *Procedia Engineering*, Vol.107, pp.263-272, 2015.
- [42] A. D. Deshmukh and U. B. Shinde, "A Low Cost Environment Monitoring System using Raspberry Pi and Arduino with Zigbee", *International Conference on Inventive Computation Technologies*, Vol.3, pp.1-6, 2016.
- [43] V. S. Tomar and V. Bhatia, "Low Cost and Power Software Defined Radio using Raspberry Pi for Disaster Effected Regions", *Procedia Computer Science*, Vol.58, pp.401-407, 2015.
- [44] J. Bermúdez-Ortega, E. Besada-Portas, J. A. López-Orozco, J. A. Bonache-Seco and J. M. Cruz, "Remote Web-Based Control Laboratory for Mobile Devices Based on EJS, Raspberry Pi and Node.js", *International Federation of Accountants*, pp.158-163, 2015.
- [45] B. Qureshi, Y. Javed, A. Koubâa, M. F. Sriti and M. Alajlan, "Performance of a Low Cost Hadoop Cluster for Image Analysis in Cloud Robotics Environment", *Symposium on Data Mining Applications*, pp.90-98, 2016.
- [46] N. Hossain, M. T. Kabir, T. R. Rahman, M. S. Hossen and F. Salauddin, "A Real-Time Surveillance Mini-Rover Based on OpenCV-Python-Java using Raspberry Pi 2", *International Conference on Control System, Computing and Engineering*, pp.476-481, 2015.
- [47] Q. He, B. Segee and V. Weaver, "Raspberry Pi 2 B+ GPU Power, Performance and Energy Implications", *International Conference on Computational Science and Computational Intelligence*, pp.163-167, 2016.
- [48] C. I. Hilliard, "Selection of a Clutter Rejection Algorithm for Real Time Target Detection from an Airborne Platform", *International Society for Optics and Photonics*, Vol.4048, pp.74-84, 2000.
- Engineering, Vol.46, No.11, pp.116402-116402, 2007.
- [28] X. Wang, L. Liu and Z. M. Tang, "Infrared Dim Target Detection Based on Fractal Dimension and Third-Order Characterization", *Chinese Optics Letters*, Vol.7, No.10, pp.931-933, 2009.
- [29] L. Itti, C. Koch and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.20, No.11, pp.1254-1259, 1998.
- [30] B. C. Ko and J. Nam, "Object-of-Interest Image Segmentation Based on Human Attention and Semantic Region Clustering", *Journal of the Optical Society of America A*, Vol.23, No.10, pp.2462-2470, 2006.
- [31] Y. Xu, Y. Zhao, C. Jin, Z. Qu, L. Liu and X. Sun, "Salient Target Detection Based on Pseudo-Wigner-Ville Distribution and Rényi Entropy", *Optics letters*, Vol.35, No.4, pp.475-477, 2010.
- [32] W. Li, C. Pan and L. X. Liu, "Saliency-Based Automatic Target Detection in Forward Looking Infrared Images", *IEEE International Conference on Image Processing (ICIP)*, pp. 957-960, 2009.
- [33] L. Zhao, K. Wu, X. Chai and C. Gu, "Image Processor for Visual Prosthesis based on ARM", *7th International Conference on Biomedical Engineering and Information (BMEI 2014)*, pp.592-596, 2014.
- [34] D. F. Vera, D. M. Cadena and J. M. Ramírez, "Iris Recognition Algorithm on Beagle Bone Black", In *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Vol.1, pp.282-286, 2015.
- [35] P. Poudel and M. Shirvaikar, "Optimization of Computer Vision Algorithms for Real Time Platforms", *42nd South Eastern Symposium on System Theory (SSST)*, pp.51-55, 2010.
- [36] K. S. Shilpashree, H. Lokesha and H. Shivkumar, "Implementation of Image Processing on Raspberry Pi", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol.4, No.5, pp.199-202, 2015.
- [37] B. E. Gamal, A. N. Ouda, Y. Z. Elhalwagy and G. A. Elnashar, "Embedded Target Detection System Based on Raspberry Pi System", *International Conference on Computer Engineering*, pp.154-157, 2016.
- [38] G. Arva and T. Fryza, "Embedded Video Processing on Raspberry Pi", *International Conference on Radioelektronika*, pp.1-4, 2017.
- [39] R. Dudas, C. VandenBussche, A. Baras, S. Z. Ali and M. T. Olson, "Inexpensive Telecytology Solutions That Use the Raspberry Pi and the iPhone", *Journal of the American Society of Cytopathology*, Vol.3, No.1, pp.49-55, 2014.
- [40] V. Vujovic and M. Maksimovic, "Raspberry Pi as a Sensor Web Node for Home Automation",