بسمه تعالی

گزارش کارآموزی

رشته مهندسی کامپیوتر

تحلیل و پردازش تصویر قطعات موتوری با استفاده از روش بخش بندی تصویر

نگارش: شایان صورتگر

استاد کارآموزی: دکتر محمدرضا رزازی

سرپرست ایپکو: مهندس اشکان موسویان

محل کارآموزی: شرکت تحقیق، طراحی و تولید موتور ایران‌خودرو (ایپکو)

شهریور ۱۴۰۰

چکیده

یکی از موضوعات مهمی که در خطوط تولید موتور به طور مداوم باید در نظر گرفته شود، کنترل فرایندهای همبندی قطعات موتور در خط تولید است. مسلماً هرچه ابزارهای کنترلی بر روی فرایندهای مختلف، بیشتر و دقیق تر باشد، کیفیت نهایی همبندی موتور افزایش می یابد. در حال حاضر، در خط تولید موتور ملی، کنترل فرایندها اغلب با نیروی انسانی صورت می گیرد. استفاده از قوای انسانی همواره با خطا همراه می باشد و این خطا در بسیاری از قسمت های خط تولید، می تواند تبعات مالی و اعتباری زیادی را به کارخانهٔ خودروسازی وارد نماید. لذا وجود یک ابزار کنترلی مستقل از انسان به منظور کنترل دقیق و سریع همبندی صحیح، برای فرایندهای خط تولید خصوصاً ایستگاه های حساس و کلیدی، لازم و ضروری است. کما اینکه در بسیاری از خطوط تولید کارخانجات خودروسازی معتبر دنیا، از چنین ابزارهای کنترلی استفاده می شود. یکی از روش های کنترلی مؤثر که اخیراً در صنایع مختلف به طور گسترده مورد استفاده قرار گرفته است، روش بینایی ماشین است. در این روش عملیات تصمیم گیری و کنترل، برپایهٔ بررسی تصاویر گرفته شده از فرایندها صورت می گیرد. مراحل کار در یک خط تولید موتور بدین صورت است که با استفاده از یک دوربین عکسبرداری، تصویری با کیفیت از قطعهٔ مورد نظر که در حال گذر از مسیر خط همبندی است، گرفته می شود. ویژگی های این قطعه، با استفاده از روش های پردازش تصویر، استخراج می شود. سپس با استفاده از یک مدل هوش مصنوعی، راجع به شرایط مونتاژی این قطعه تصمیم گیری شده و به تکنسین خط، اطلاع داده می شود. در شرایط پیشرفته تر، نتیجهٔ تصمیم گیری هوشمند، به طور خودکار بر فرایند تولید اثرگذار است بدین صورت که اگر ایرادی در قطعه مشاهده شد، می تواند به طور خودکار، خط را متوقف کند. با توجه به توضیحات مذکور، یکی از ارکان اصلی سامانه های بینایی ماشین، فرایند پردازش تصویر است. با استفاده از پردازش تصویر، اطلاعات اصلی مورد نیاز به منظور تصمیم گیری به دست می آید و دیگر ویژگی ها و اطلاعاتی که در یک تصویر خام وجود دارد، حذف می شوند. پردازش تصویر از چند روش عمده و اصلی تشکیل شده است که هر یک مشتمل بر تکنیک های متعدد و بسیاری است. یکی از این روش های اصلی، بخش بندی تصویر است که شامل تکنیک های آستانه گذاری، شناسایی لبه ها و تبدیل هاف برای شناسایی خطوط مستقیم و زوایای آنها می باشد. در این پروژه تئوری روش بخش بندی تصویر فراگیری شده و سپس اقدام به کدنویسی آنها به منظور تحلیل و پردازش تصاویر قطعات موتوری می شود.

واژه‌های کلیدی:

بینایی ماشین ، بخش بندی تصویر ، آستانه گذاری ، لبه یابی ، تبدیل هاف

فهرست مطالب

صفحه

فهرست اشکال

صفحه

فهرست جداول

صفحه

فهرست علائم

علائم لاتین

h	ارتفاع
L	طول موج توربولانس
T	پریود توربولانس
σ	واریانس

فصل اول

مقدمه

مقدمه

مرکز تحقیقات موتور در ۱۶/۱۰/۱۳۷۶ تأسیس شد و در زمینهٔ توسعهٔ قوای محرکه شامل موتور و جعبه دنده، از مرحلهٔ طراحی تا شروع تولید و نظارت بر آن فعال گردید. هم اکنون ایپکو، از دانش حدود سیصد کارشناس خبره در زمینهٔ قوای محرکه استفاده می‌کند. در سال ۱۳۸۱ در راستای ایجاد شرکت رده یکمی در گروه صنعتی ایران‌خودرو، «مرکز تحقیقات موتور» به «شرکت تحقیق، طراحی و تولید موتور ایران خودرو» تغییر نام داد. با این تغییر راهبردی، بخش های تولید، مهندسی محصول و تضمین کیفیت شکل گرفتند. آزمایشگاه موتور و خودروی ایپکو بر اساس دانش فنی و همکاری با آزمایشگاه های موتور معتبر بین‌المللی قادر است تا آزمون‌های دوام، عملکرد، وظیفه‌ای، احتراقی، نگاشت موتور، سنجش آلایندگی و مصرف سوخت را به عنوان یکی از معتبرترین مراجع در سطح کشور انجام دهد. هم اکنون شرکت ایپکو به عنوان پیشرو در این صنعت به تحقیق، طراحی و تولید انواع موتور و جعبه‌دنده (خودرویی و غیر خودرویی) مشغول است .ایپکو در دهه سوم فعالیت خود با توجه به ظهور انقلاب صنعتی چهارم، در حال تعریف و بروز رسانی برنامه‌های راهبردی خود است. استفاده از هوش مصنوعی و یادگیری ماشین در بهبود عملکرد موتورهای متداول، برقی‌سازی خودروها و توسعه قوای محرکه برقی، خودران‌ها و حمل‌ونقل نوین، زیرساخت شبکه اطلاعات محصول (از مواد اولیه، تولید، بکارگیری تا بازیافت)، استفاده از جریان اطلاعات (امکان تشخیص و رفع ایراد از راه دور) و تولیدهای پایدار دوست‌دار محیط زیست از آن جمله است.

فصل دوم

بخش بندی تصاویر

بخش بندی تصاویر

بخش‌بندی تصویر اولین مرحله و بحرانی‌ترین مرحله از آنالیز تصویر می‌باشد که هدفش استخراج اطلاعات داخل تصویرمانند (لبه‌ها ، نماها و هویت هر یک از نواحی) می‌باشدکه از طریق توصیف، ناحیه‌های بدست آمده را برای کاهش آنها به شکل مناسب برای پردازش کامپیوتر و تشخیص هر یک از نواحی آماده می‌کند.نتیجه بخش بندی تاثیر قابل ملاحظه ای بر دقت ارزیابی ویژگی‌هاخواهد داشت .بخش‌بندی اغلب شرح فرآیند تقسیم تصویر به اجزاء اصلی و استخراج قسمتهای مورد علاقه اشیاء می‌باشد. بخش بندی یکی از مشکل ترین مباحث در پردازش تصویراست که در موفقیت عمل تحلیل تصویر بسیار موثر است. برای بخش بندی تصویر روشهای مختلفی وجود دارد که می توان انهارا به دو دسته روشهای مبتنی بر هیستوگرام(Histogram based) و روشهای مبتنی بر خوشه بندی(Clustering-Based) تقسیم کرد. که البته هر کدام از این دو روش دارای زیر مجموعه هایی نیز می باشند. در روشهای مبتنی بر هیستوگرام، بخش بندی تصاویر براساس توزیع پیکسلها صورت می گیرد. قدم اصلی در این روشها یافتن سطح استانه ای مناسب برای اعمال به تصویر میباشد. در روشهای مبتنی بر خوشه بندی برای گروه بندی کردن داده ها از شباهتها و روابط موجود بین آنها استفاده می شود. در این روشها داده ها به نحوی گروه بندی می شوند تا انهایی که در داخل یک بخش قرار می گیرند دارای بیشترین شباهت به هم باشند.

۱-۲- شرایط بخش‌بندی

برای بخش‌بندی هر تصویر باید شرایطی داشته‌باشیم‌ازجمله:مجموع کل قطعات کل پیکسلهای تصویر را تشکیل می‌دهند. نواحی قطعات نباید تداخل داشته باشند. پیکسلهای قطعات یکسان باید خواص یکسانی داشته باشند.پیکسلهای قطعات متفاوت باید خواص متفاوتی داشته باشند. پیکسلهای قطعات یکسان‌مرتبط هستند. در بخش‌بندی تصویر، تصویر به تعدادی نواحی تقسیم می‌شود که این تقسیم بندی با توجه به ویژگی‌های برداری تصویر مثل رنگ تصویر و غیره صورت می‌گیرد .در بخش‌بندی هدف ما بدست آوردن شیء یا اشیاء مورد نظر از تصویر می‌باشد.یکی از راه‌های رسیدن به این هدف‌عمل لبه‌یابی تصویر می‌باشد.

اولین تکنیک‌های گسترش بخش‌بندی تصاویر، به سال ۱۹۶۵برمی‌گردد که یک عملگر برای لبه‌یابی بین قسمتهای مختلف یک تصویر استفاده شد که لبه‌یاب رابرت نامیده شد.این اولین مرحله برای ، گسترش تجزیه تصویر می‌باشد.بر اثر دید منفی و کم‌لطفی محققان ،مدتی بررسی و تحقیق در مورد بخش‌بندی تصاویر به کندی پیش رفت اما جدیداً توجه خاصی به این مبحث می‌شود.

۲-۲- تکنیک های بخش‌بندی مورد استفاده

تکنیک های مورد استفاده در این پروژه به سه گروه کلی آستانه گذاری ، لبه یابی و تبدیل هاف تقسیم می شوند.

روش آستانه گذاری و لبه یابی به صورت مجزا استفاده می شوند و معمولا برای جدا کردن اشیا از پس زمینه ، در صورتی که پس زمینه نسبتا ساده باشد استفاده می شوند. اما تبدیل هاف از خروجی یک تابع لبه یاب برای تشخیص دقیق خطوط مستقیم و دایره ها در تصویر استفاده می کند.

مجموعه خروجی های این تکنیک ها اطلاعات بسیار مفید و قابل استفاده برای دیگر تکنیک های کامپیوتری خوبی را به ما می دهد.

در فصول آتی به بررسی دقیق هر یک از روش ها ، روابط ریاضی آن ها و برنامه نویسی این روش ها می پردازیم و در نهایت مثال های عادی و صنعتی این تکنیک ها را نمایش می دهیم.

فصل سوم

تکنیک آستانه گذاری

تکنیک آستانه گذاری

در آستانه گذاری تصویر، در ابتدا یک مقدار **سطح آستانه** برای پیکسلهای تصویر(که میتواند یک سطح آستانه سراسری برای همه پیکسلها باشد، و یا به ازای هر پیکسل یک سطح آستانه جداگانه ای باشد) مشخص می‌کنیم. سپس مقدار پیکسل‌های تصویر با سطح آستانه مقایسه می‌کنیم و اگر شدت روشنایی پیکسل بزرگتر از حد آستانه باشد به سفید و اگر کمتر باشد به سیاه تبدیل می‌کنیم. که در نتیجه آن، یک تصویر رنگی یا سطح خاکستری به یک تصویر باینری(سیاه و سفید) تبدیل می‌شود. در بیشتر مواقع ما از آستانه‌گذاری به عنوان یک روشی برای انتخاب نواحی مورد نظر تصویر، و حذف نواحی ای که برای ما اهیمت ندارد استفاده می‌کنیم.

به طور کلی روشهای آستانه گذاری تصویربه دو دسته تقسیم می شوند:

- **آستانه گذاری سراسری :**یک سطح آستانه برای همه پیکسل‌های تصویر تعریف می شود.
- **آستانه گذاری محلی :**سطح آستانه به صورت محلی تعریف می شود. به عبارتی به ازای هر پیکسل یک سطح آستانه مجزایی تعریف می‌شود.

۱-۳- آستانه گذاری سراسری تصویر(global thresholding)

در آستانه گذاری سراسری برای کل پیکسل‌های تصویر یک سطح آستانه کلی مشخص می‌شود. سپس مقدار شدت روشنایی تک تک پیکسل‌های تصویر با این سطح آستانه مقایسه می‌شوند. اگر مقدار شدت روشنایی یک پیکسل بزرگتر از سطح آستانه باشد، به مقدار یک (سفید) و اگر کوچکتر باشد به مقدار صفر(سیاه) تبدیل می‌شود. در نتیجه این فرایند یک تصویر باینری (سیاه و سفید) ساخته می‌شود. که محتوای تصویر خروجی از لحاظ شدت روشنایی به دو گروه سیاه و سفید تقسیم می‌شوند.

معروفترین روش برای آستانه گذاری سراسری، روش اتسو هست .این متد به نوعی روشی برای بخش‌بندی تصویر بر اساس یافتن آستانه بهینه t ، به نحوی که تصویر را به دو کلاس مجزا (سیاه و سفید) تقسیم نماید. منظور از آستانه بهینه در اینجا، یافتن مقداری از t، که حداکثر یکنواختی را در تابع شدت، در هردو کلاس ایجاد کند و واریانس تابع توزیع شدت در پیکسل‌ها، مابین دو کلاس را کمینه سازد :

$$\sigma_w^2(t) = \omega_1(t) \cdot \sigma_1^2(t) + \omega_2(t) \cdot \sigma_2^2(t)$$

که در آن وزن‌های w_i، احتمالات وقوع دو کلاس که به‌وسیله آستانه t از هم مجزا گشته‌اند و σ_i^2 ، واریانس این دو کلاس می‌باشد. اتسو کمینه‌سازی واریانس بین‌کلاسی را به‌وسیله احتمال وقوع هریک از کلاس‌ها w_i، و میانگین کلاس‌ها μ_i(t)، تحت شرایط زیر توصیف نمود:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t) \cdot \omega_2(t)\left[\mu_1(t) - \mu_2(t)\right]^2$$

احتمال وقوع کلاس یک برای آستانه موردنظر t که به‌صورت w_1(t) معرفی می‌شود، به شکل زیر محاسبه می‌گردد:

$$\omega_1(t) = \Sigma_0^t p(i)$$

مقدار میانگین برای کلاس یک به‌صورت زیر محاسبه می‌شود:

$$\mu_1(t) = \left[\Sigma_0^t p(i) \cdot x(i)\right] / \omega_1$$

که در آن x(i)، مقدار در مرکز i_th هیستوگرام می‌باشد. به توضیحات شکل‌گرفته، به همین صورت می‌توان مقدار ω_2(t) و μ_2(t) را محاسبه نمود.

مزیت روشهای آستانه گذاری سراسری

روشهای آستانه گذاری سراسری پیچیدگی محاسباتی کمتری دارند و زمان اجرای برنامه در مقایسه با روشهای آستانه گذاری محلی بسیار کمتری دارند.

ایراد روشهای آستانه گذاری سراسری

آستانه گذاری سراسری زمانی درست عمل می‌کند، که پیکسلهای تصویر به طور کلی از دو گروه شدت روشنایی تشکیل شوند، به عبارتی این روشها زمانی خوب عمل می‌کنند که **هیستوگرام تصویر شامل**

دو قله باشد. در غیراینصورت با این رویکرد حد آستانه مناسبی انتخاب نشده و در نتیجه تصویر بـه طـور بهنیه به باینری تبدیل نمی‌شود.

به عبارت دیگر، زمانی که پیش زمینه (background)تصویر ورودی **یکنواخت** نباشد، انتخاب یک حد آستانه سراسری برای کل پیکسل‌های تصویر امکان‌پذیر نخواهد بود. چرا کـه در ایـن حالـت حـد آسـتانه انتخاب شده، ممکن است برای یک ناحیه از تصویر مناسب باشد و یک ناحیـه مناسـب نباشـد. در نتیجـه همه پیکسلهای تصویر به درستی به باینری تبدیل نشوند.

۲-۳- آستانه گذاری محلی تصویر(local/adaptive thresholding)

در آستانه گذاری محلی، به جای اینکه برای کل تصویر یک حد آستانه تعریف شود، برای **هر پیکسل** بـر اساس اطلاعات **پیکسل‌های همسایه** آن یک حد آستانه مجزایی تعریـف می‌شـود.در ایـن حالـت دیگـر مشکل روشهای سراسری ایجاد نخواهد شد، چرا برای هـر ناحیـه از تصـویر یـک حـد آسـتانه مناسـب و منحصر به فردی تعریف می‌شود و در نتیجه آن تمام بخشهای تصویر به طور مناسـب بـه بـاینری تبـدیل می‌شوند .

برای محاسبه استانه محلی مراحل زیر طی می شود:

۱- یک بلوک یا ناحیهbxb ، اطراف مکان پیکسل توسط کاربر انتخاب می شود...b=3,5,7,.

۲- میانگین وزنی ناحیه bxb را محاسبه می کنیم Opencv .دو روش پیشنهاد می دهد:

- محاسبه میانگین ناحیه پیکسل ناحیهbxb
- محاسبه میانگین وزنی گوسی مکان پیکسل ناحیهbxb

۳- برای محاسبه مقدار **آستانه**، میانگین از یک مقدر ثابت (C)کم می شود:

$$T= M- C$$

توابع ساده و سریع عبارتند از:

میانگین (mean) توزیع شدت محلی:

$$T = mean$$

مقدار میانه(median)

$$T = median$$

یا میانگین حداقل و حداکثر مقادیر

$$T = \frac{max + min}{2}$$

اندازه همسایگی، باید به اندازه کافی بزرگ باشد تا پیکسل های شی و پس زمینه کافی را پوشش دهد، در غیر این صورت یک آستانه ضعیف انتخاب میشود.از سوی دیگر، انتخاب مناطق که بیش از حد بزرگ هستند می تواند فرض تقریبا یکنواخت را نقض کند.

۳-۳-آستانه گذاری تصاویر رنگی

عکس های رنگی با فضاهای رنگی متفاوتی به وجود می آیند که قابل تبدیل شدن به یکدیگر نیز هستند. به صورت کلی در یک عکس رنگی، برخلاف یک عکس سیاه وسفید ،چندین کانال رنگی وجود دارد که ترکیب اعداد آن ها رنگ آن پیکسل را نشان می دهد. مثلا معروف ترین فضای رنگی RGB می باشد که در آن اعداد هر پیکسل به ترتیب بیانگر مقدار رنگ قرمز ، سبز و آبی هستند و اگر هر سه در حالت مینیم یعنی صفر باشند آن پیکسل رنگ سیاه مطلق و در صورت ماکسیمم بودن آن پیکسل سفید مطلق است . نمونه ای از رنگ ها را در جدول زیر مشاهده می کنید:

جدول ۳-۱ رنگ ها در فضای RGB

Color	RGB value
Red	255, 0, 0
Orange	255, 128, 0
Pink	255, 153, 255

در آستانه گذاری رنگی تصاویر ، ابتدا محدوده ی رنگ مورد نظرمان (اندازه ی محدوده با توجه به کیفیت ، نور ، زاویه عکس و غیره ممکن است تغییر کند) در آن فضای رنگی را پیدا کرده و سپس پیکسل های قرار گرفته در آن محدوده را به عنوان شی در نظر می گیریم(ماسک می کنیم). طبیعتا برای اشیا دو یا چند رنگ می توان ماسک های هررنگ از آن شی را با یکدیگر جمع کرد. روش دیگر برای تشخیص اشیا در پس زمینه تقریبا یکنواخت ، تشخیص و ماسک کردن پس زمینه و سپس کم کردن ماسک از تصویر کلی است.

بعد از تشخیص دقیق پیکسل های یک شی ، می توان محدوده ی رنگی را با توجه به کمینه و بیشینه ی مقادیر عددی پیکسل های به دست آمده بهینه کرد و از آن برای تشخیص شی در عکس ها و محیط های دیگر استفاده کرد. البته بهتر است برای جلوگیری از اثرپذیری این روش از عوامل ناخواسته مانند کیفیت ، نور و غیره از کمینه و بیشینه ی دقیق استفاده نشود و یک عدد حاشیه بسته به شرایط عکس اولیه به آن ها اضافه یا از آن ها کم شود.

۳-۴- بخش برنامه نویسی

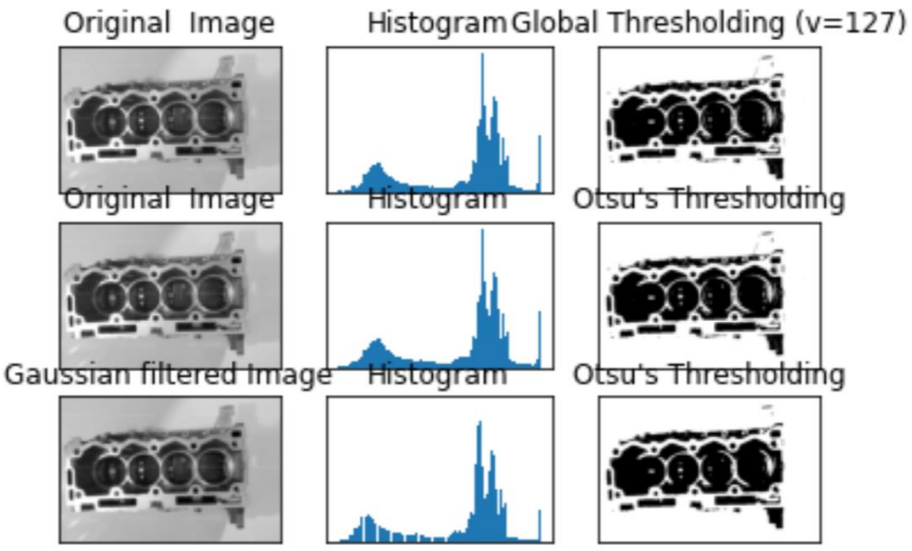
برای پیاده سازی تکنیک آستانه گذاری از کتابخانه ی CV2 استفاده شده است.

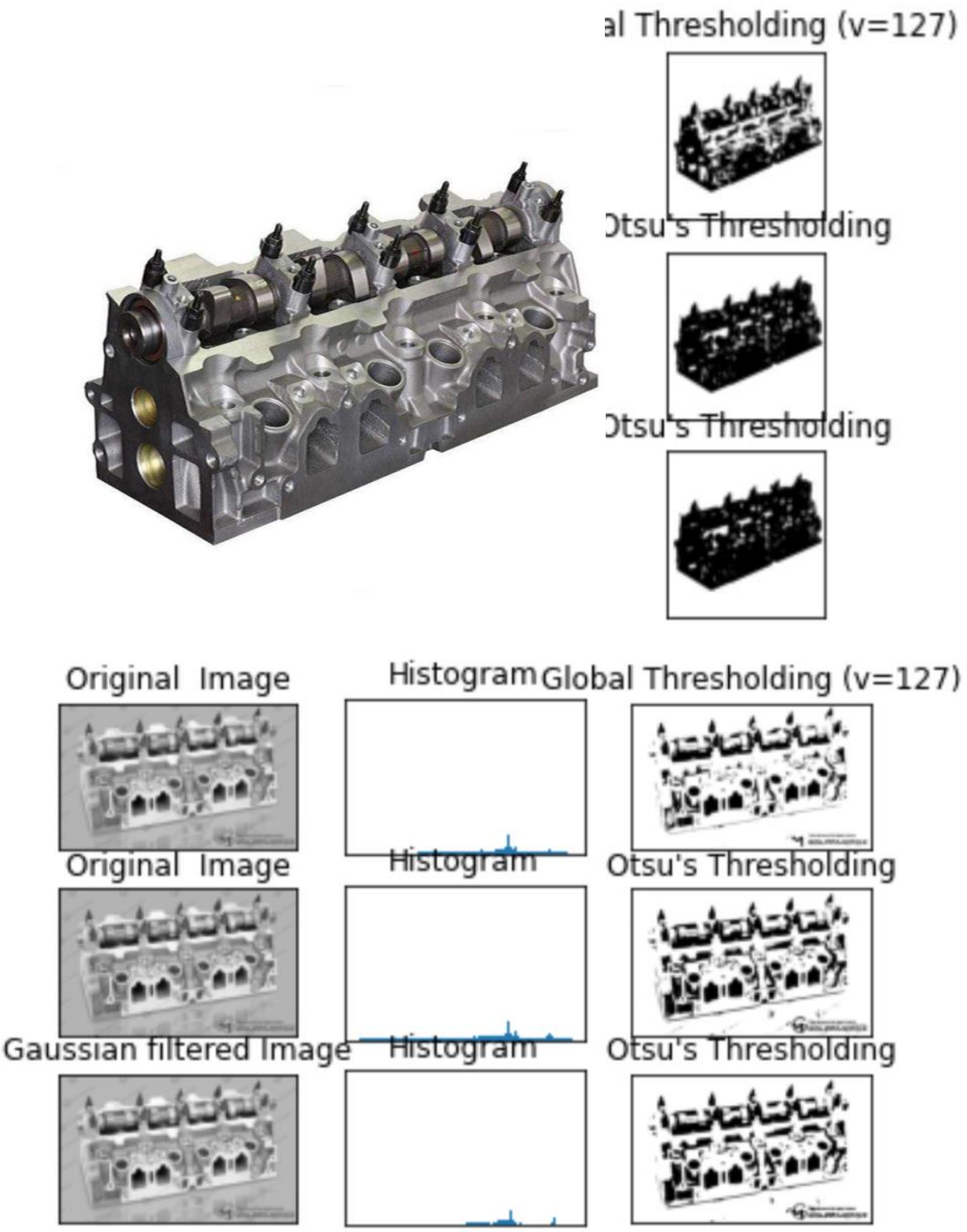
این کتابخانه تابعی به نام threshold دارد برای استانه گذاری سراسری دارد که چهار ورودی می گیرد و ورودی چهارم آن نوع آستانه گذاری است. برای آستانه گذاری اتسو متغیر چهارم باید cv.THRESH_BINARY+cv.THRESH_OTSU باشد.

برای فیلتر کردن نویز ها قبل از استفاده از روش اتسو می توان از یک فیلتر گاوسی نیز استفاده کرد.

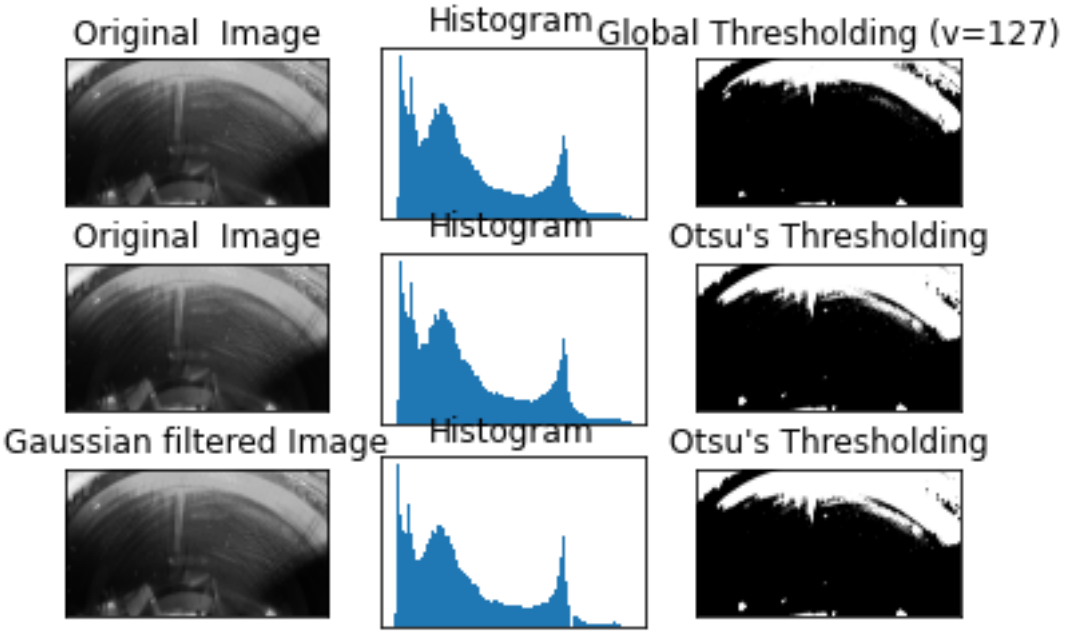
خروجی های زیر تکنیک استانه گذاری سراسری با مقدار ثابت ، استانه گذاری اتسو و آستانه گذاری اتسو پس از گذراندن عکس ها از یک فیلتر گاوسی را به همراه هیستوگرام آن ها نشان می دهد.

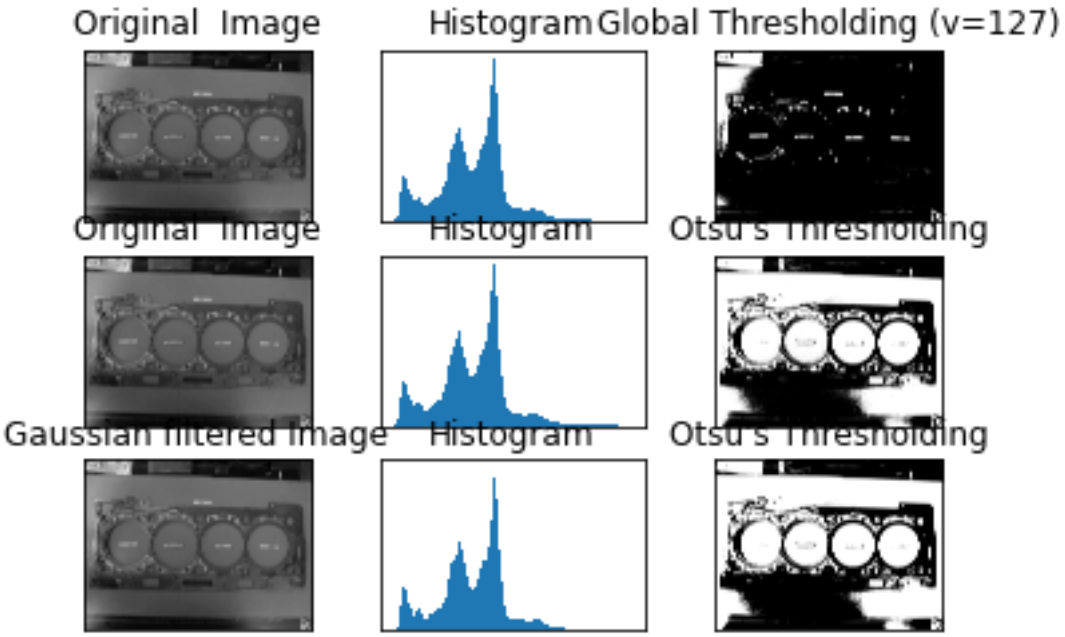
شکل۱-۳ مثال های آستانه گذاری استو

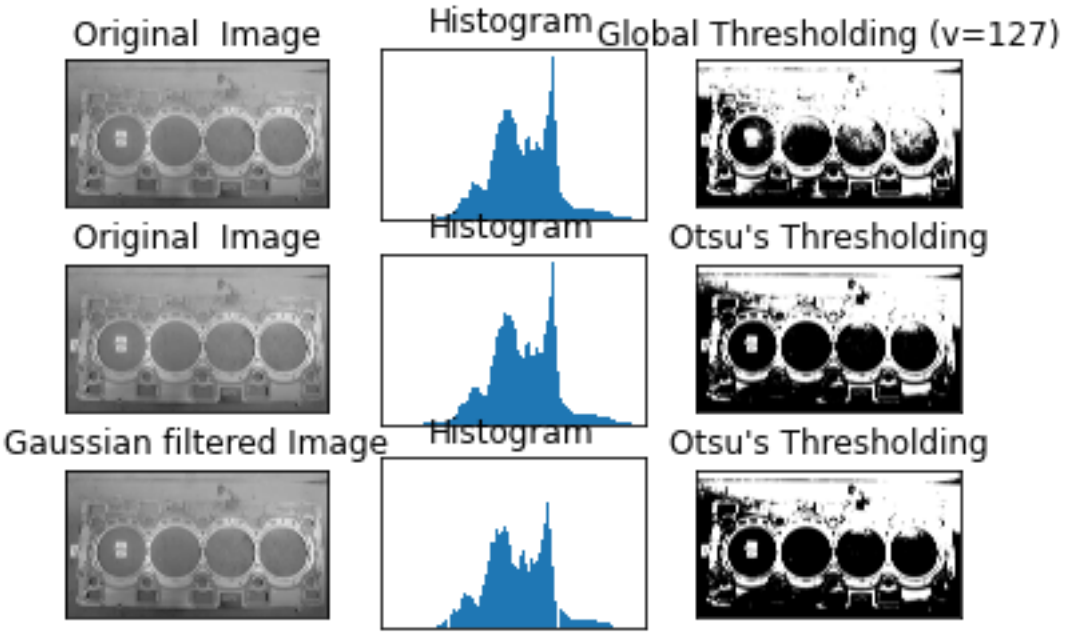


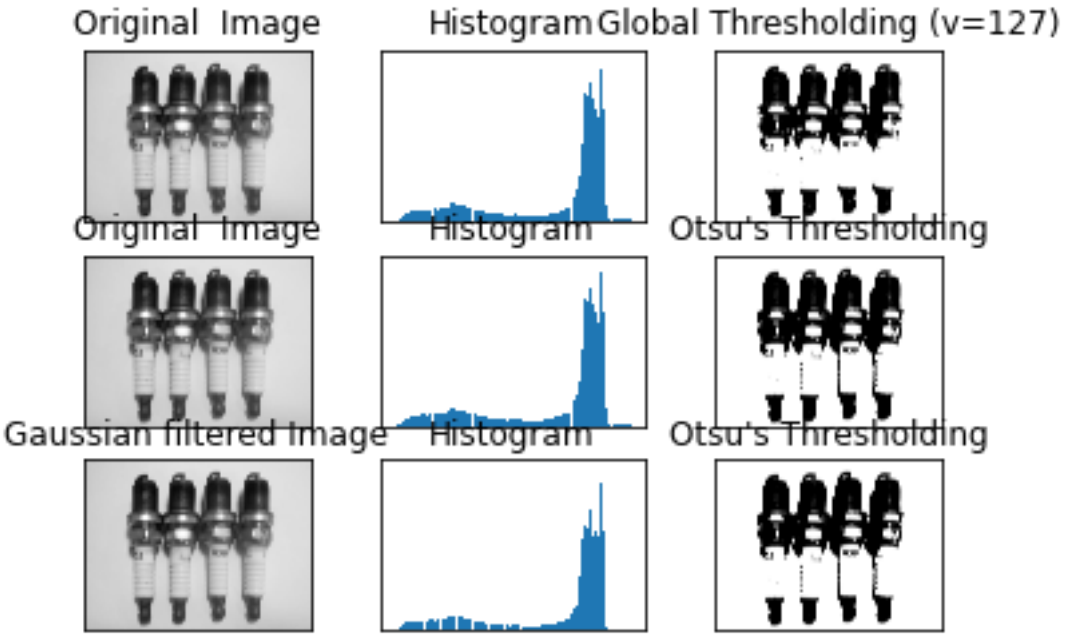

al Thresholding (v=127)
Otsu's Thresholding
Otsu's Thresholding
Original Image
Histogram
Global Thresholding (v=127)
Original Image
Histogram
Otsu's Thresholding
Gaussian filtered Image
Histogram
Otsu's Thresholding

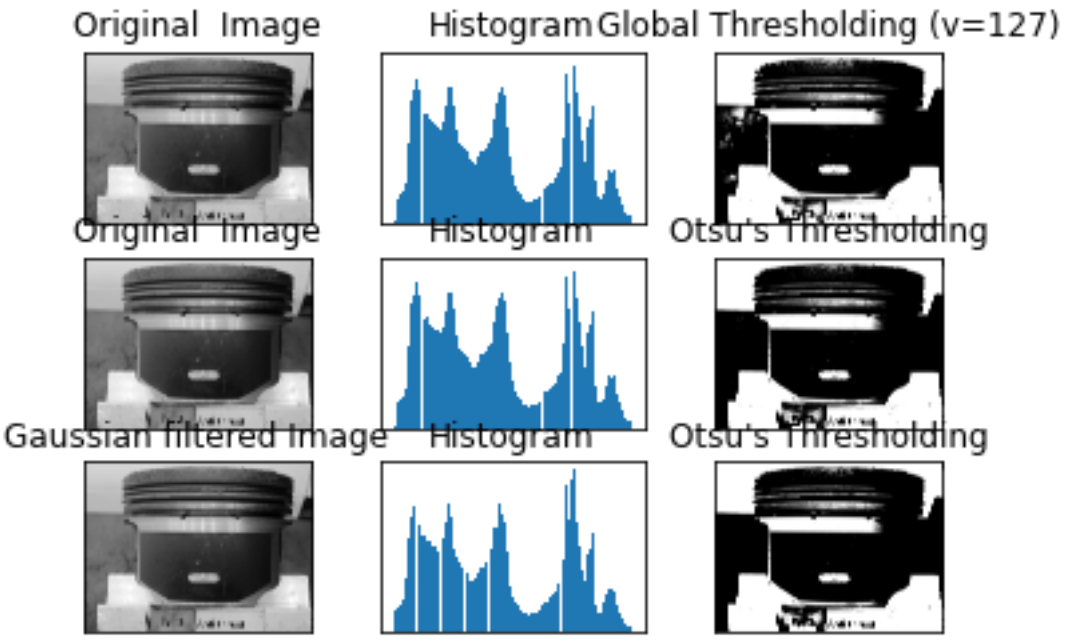
شکل۲-۳ عکس های صنعتی ایپکو (آستانه گذاری اتسو)






Original Image
Histogram
Global Thresholding (v=127)
Original Image
Histogram
Otsu's Thresholding
Gaussian filtered Image
Histogram
Otsu's Thresholding


Original Image
Histogram
Global Thresholding (v=127)
Original Image
Histogram
Otsu's Thresholding
Gaussian filtered Image
Histogram
Otsu's Thresholding



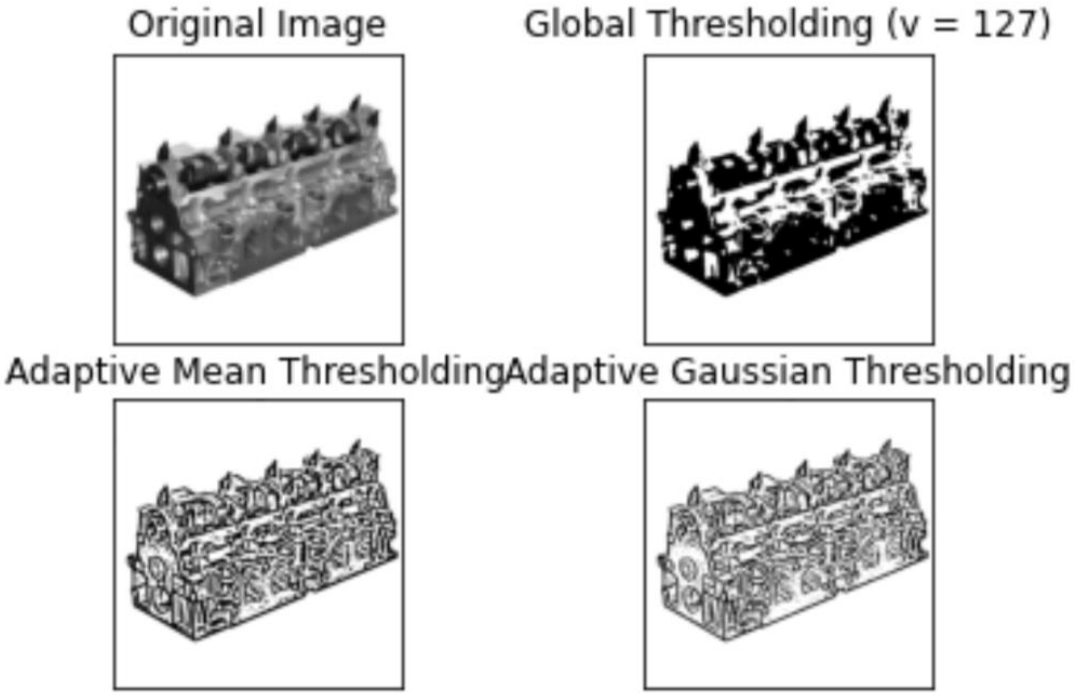
برای استانه گذاری محلی باید از تابع adaptiveThreshold استفاده کرد که سه ورودی بیشتر از تـابع قبل می گیرد.

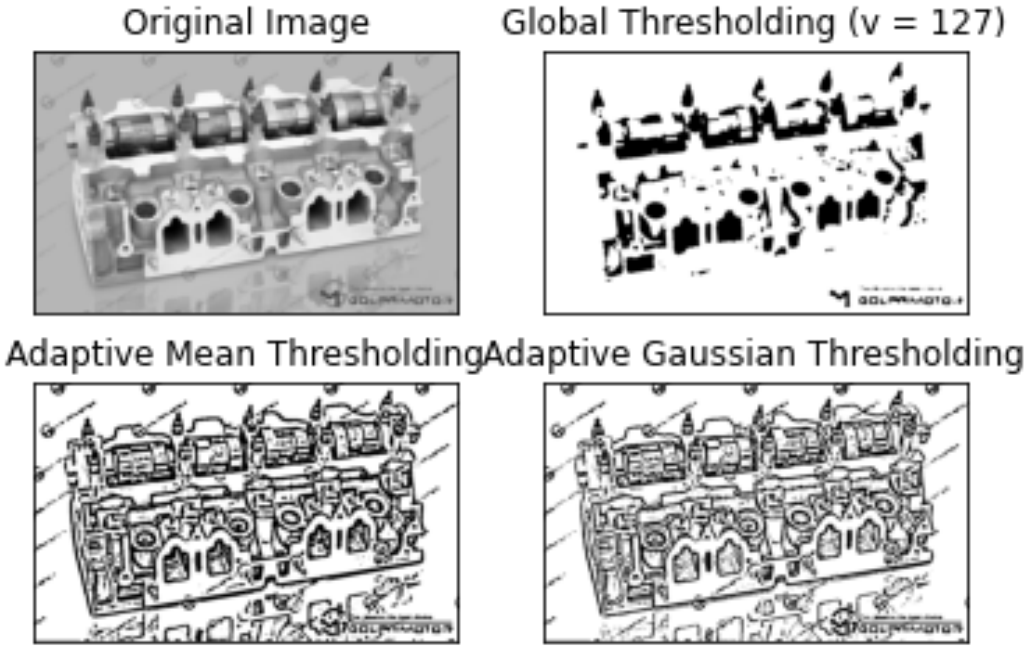
متد محلی (**adaptiveMethod**) روش محاسبه ی حد آستانه ، **blockSize** اندازه ی همسـایگی و c عدد ثابتی که از میانگین یا جمع وزن دار کم می شود.

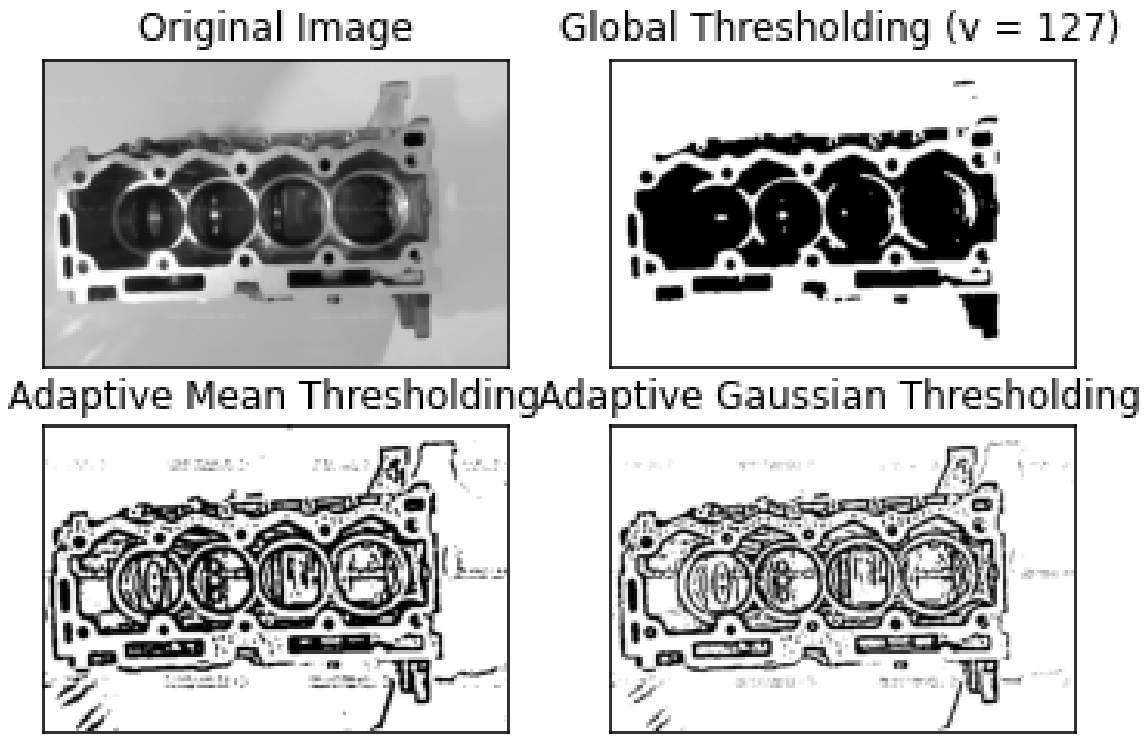
لازم به ذکر است که عکس ورودی باید لزوما خاکستری (**grayscale**) باشد.

خروجی های زیر استانه گذاری سراسری ، استانه گذاری محلی با میانگین و استانه گذاری محلی گاوسـی را نشان می دهد.

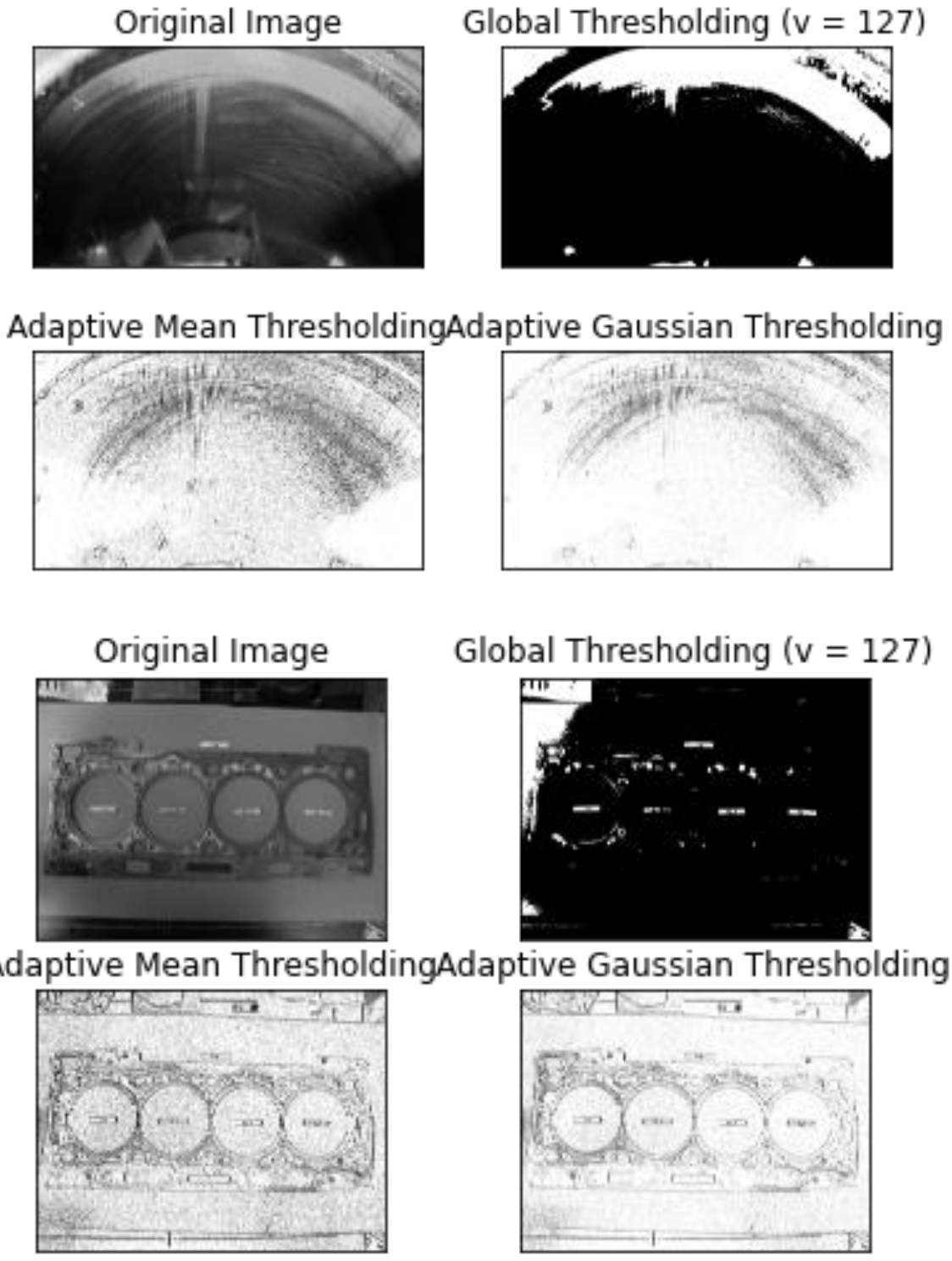
شکل۳-۳ مثال های استانه گذاری محلی

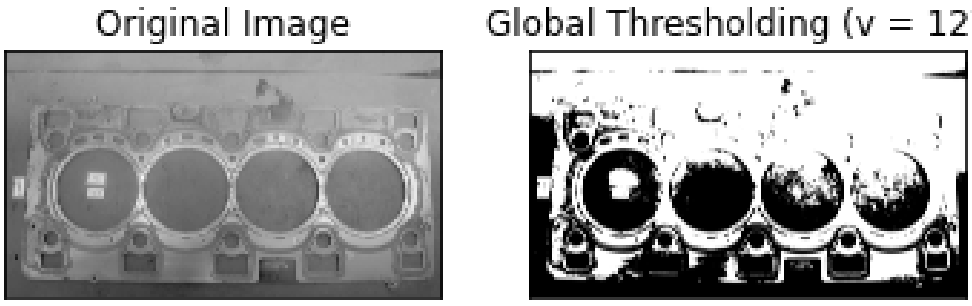


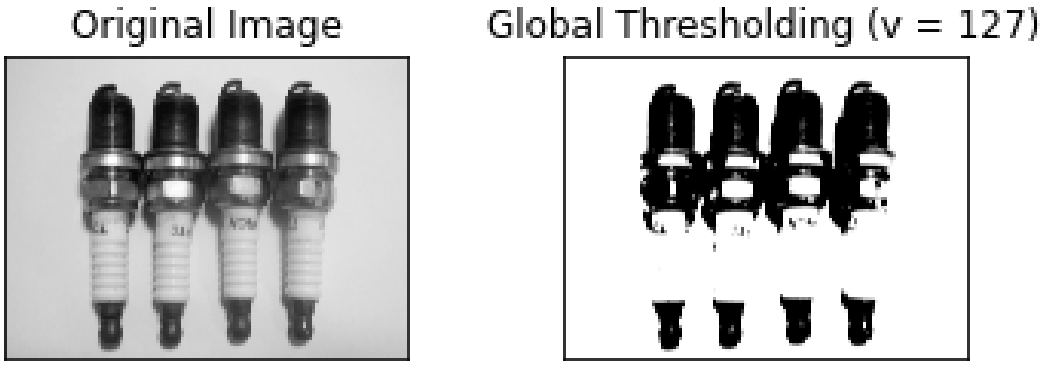


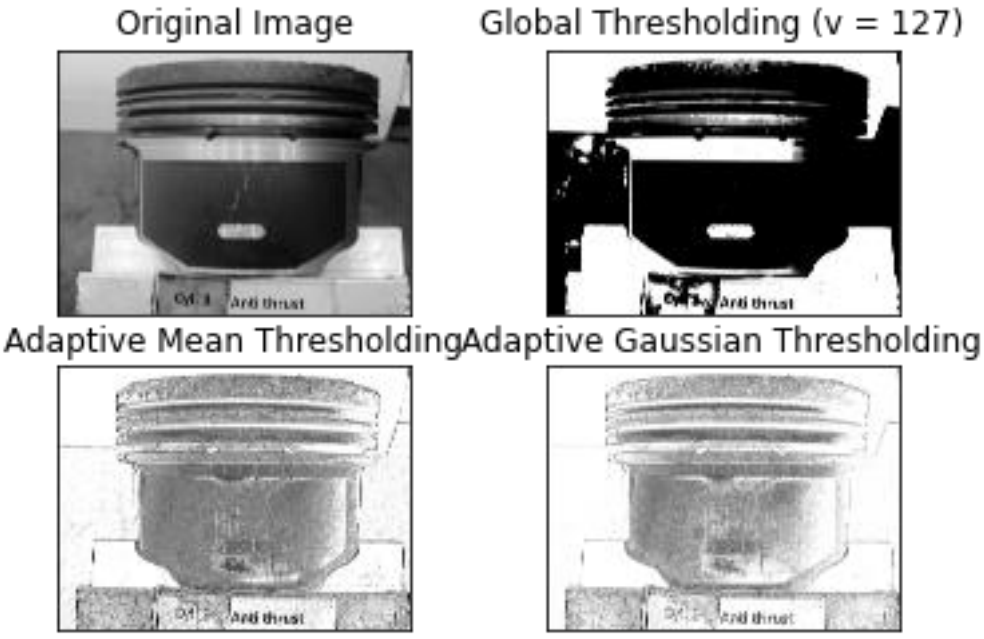

Original Image
Global Thresholding (v = 127)
Adaptive Mean Thresholding
Adaptive Gaussian Thresholding

شکل۴-۳ عکس های صنعتی ایپکو (آستانه گذاری محلی)




Original Image
Global Thresholding (v = 127)
Adaptive Mean Thresholding
Adaptive Gaussian Thresholding


Original Image
Global Thresholding (v = 127)
Adaptive Mean Thresholding
Adaptive Gaussian Thresholding

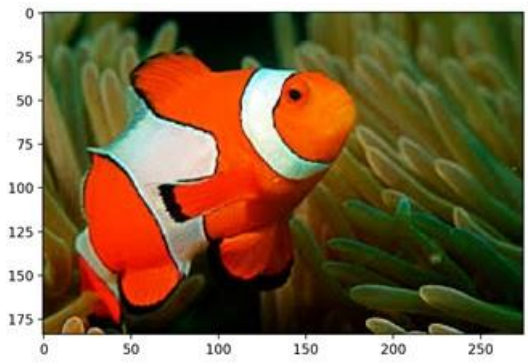

Original Image
Global Thresholding (v = 127)
Adaptive Mean Thresholding
Adaptive Gaussian Thresholding

برای انتقال یک عکس از یک فضای رنگی به یک فضای دیگر می توان از تابع cv2.cvtColor استفاده کرد که ورودی اول آن عکس مورد نظر و ورودی دوم پرچم (flag) انتقال مورد نظر است.

برای آستانه گذاری تصاویر رنگی ، بهتر است عکس را از فضای رنگی RGB به HSV ببریم که پرچم آن cv2.COLOR_RGB2HSV است.

سپس با دادن عکس hsv به تابع cv2.inRange پیکسل های در محدوده ی مورد نظرمان را پیدا کرده و با تابع cv2.bitwise_and آن ها را روی عکس مشخص می کنیم. برای نمونه در تصویر زیر می خواهیم ماهی را پیدا کنیم:

شکل۵-۳ مثال آستانه گذاری رنگی



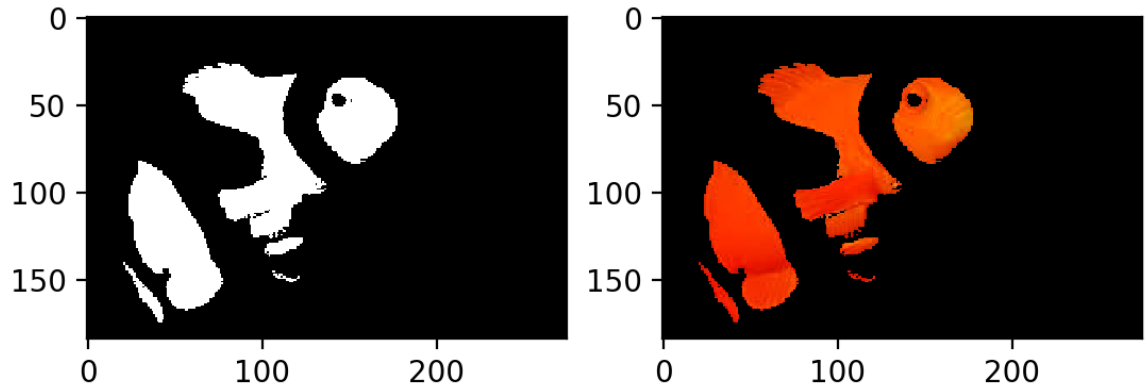
رنگ بخش نارنجی به صورت تقریبی در محدوده ی زیر است :

light_orange = (1, 190, 200)

dark_orange = (18, 255, 255)

که با آستانه گذاری این محدوده بخش هایی از ماهی به دست می آید:

شکل۶-۳ آستانه گذاری با رنگ نارنجی



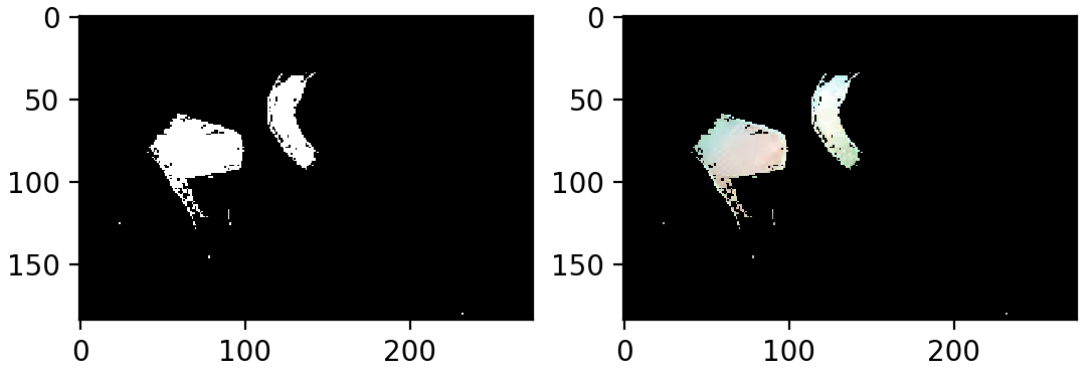
حال برای پیدا کردن بخش های سفید استانه گذاری دوم را انجام می دهیم:

light_white = (0, 0, 200)

dark_white = (145, 60, 255)

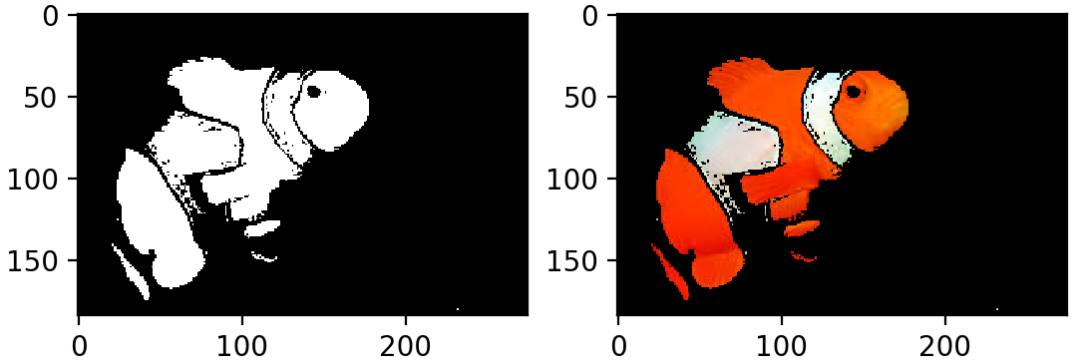
که منجر به شناسایی بخش های زیر می شود:

شکل۷-۳ استانه گذاری با رنگ سفید



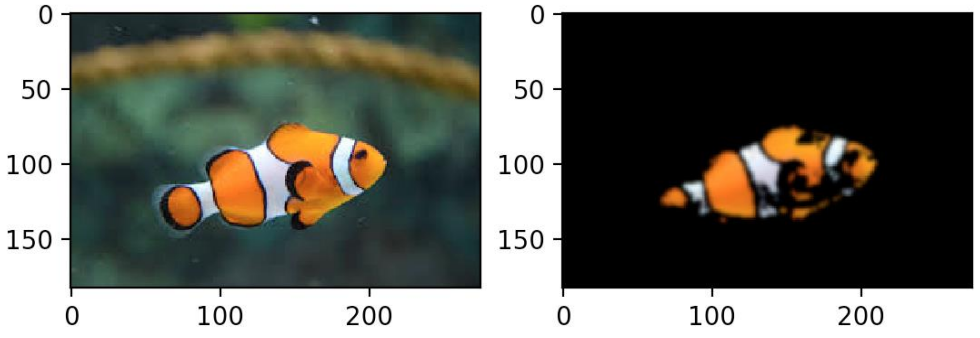
در نهایت با ترکیب دو ماسک به عکس نهایی می رسیم:

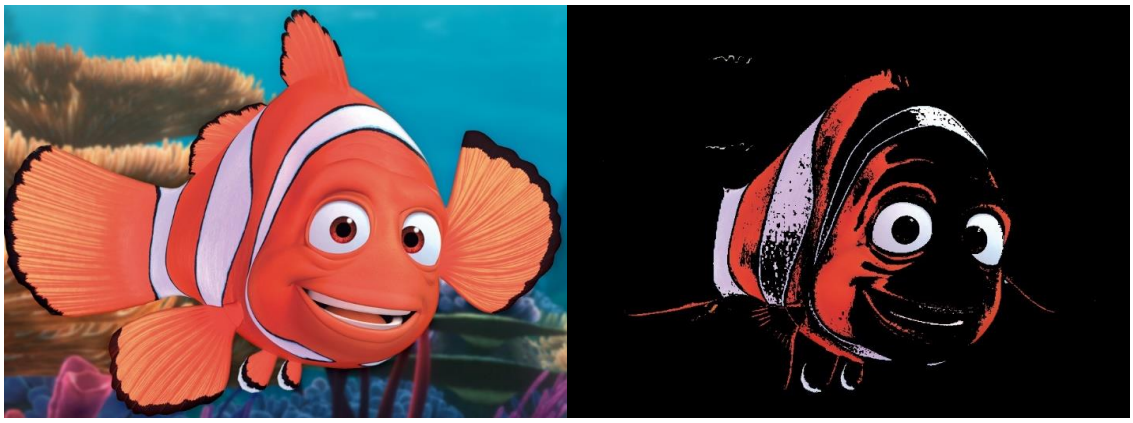
شکل۸-۳ نتیجه استانه گذاری رنگی



محدوده های به دست آمده قابل استفاده برای تشخیص این ماهی در تصاویر دیگر نیز هستند:

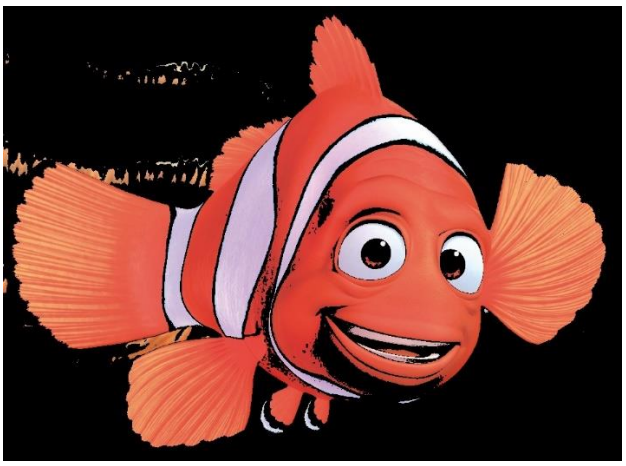
شکل۹-۳ استفاده از فیلتر به دست آمده در عکس های دیگر





البته همانطور که گفته شد برای تشخیص دقیق تر نیاز به بهینه سازی می باشد:

شکل۱۰-۳ بهینه سازی عکس ها بعد از آستانه گذاری رنگی



فصل چهارم

لبه یابی

لبه یابی

آشکارسازی لبه (edge detection) معمولا برای تشخیص لبه های یک شی از بین چند شی دیگر مورد استفاده قرار می گیرد.

لبه مرز بین نواحی با خواص نسبتاً متفاوت سطح خاکستری است. نظریه پایه در بیشتر روش‌های آشکارسازی لبه، محاسبه یک عملگر مشتق محلی است. در این مقطع توجه شود که لبه (گذر از تاریک به روشن) به صورت یک تغییر آرام، نه سریع ، با سطح خاکستری مدل می‌شود. این مدل نشان می‌دهد که معمولاً لبه‌های تصاویر رقمی بر اثر نمونه‌برداری، کمی مات می‌شوند. مشتق اول مقطع سطح خاکستری در لبه جلویی گذر، مثبت است، در لبه عقبی آن منفی است و همان طور که مورد انتظار است، در نواحی با سطح خاکستری ثابت صفر است. مشتق دوم برای قسمتی از گذر که در طرف تیره لبه است، مثبت است، برای قسمت دیگر گذر که در طرف روشن لبه است، منفی است، و در نواحی با سطح خاکستری ثابت، صفر است. بنابراین، از بزرگی مشتق اول می‌توان برای تعیین این که آیا پیکسل در روی لبه قرار دارد، استفاده کرد. مشتق دوم در نقطه وسطی هر گذر سطح خاکستری یک عبور از صفر دارد. عبور از صفرها راهی قوی برای تعیین محل لبه‌های تصویر فراهم می‌آورند. اندازه مشتق اول تصویر در هر نقطه برابر بزرگی گرادیان است. مشتق دوم نیز با استفاده از لاپلاسین به دست می‌آید. اگر یک لبه را به عنوان تغییر در شدت روشنایی که در طول چند پیکسل دیده می‌شود در نظر بگیریم، الگوریتم‌های آشکارسازی لبه به طور کلی مشتقی از این تغییر شدت روشنایی را محاسبه می‌کنند. برای ساده‌سازی، به آشکارسازی لبه در یک بعد می‌پردازیم. در این نمونه، داده‌های ما می‌تواند یک تک‌خط از شدت روشنایی پیکسل‌ها باشد. برای نمونه بین پیکسل‌های چهارم و پنجم در داده‌های ۱بعدی زیر به روشنی می‌توان لبه‌ای را آشکار کرد.

۱۴۹	۱۴۸	۱۵۲	۴	۶	۷	۵

۱-۴-محاسبه مشتق اول

تعداد زیادی از عملگرهای آشکارسازی لبه بر پایه مشتق اول شدت روشنایی کار می‌کنند، یعنی با گرادیان شدت روشنایی داده‌های اصلی سروکار داریم. با این اطلاعات می‌توانیم تصویری را برای قله‌های گرادیان

روشنایی جستجو کنیم. اگر $I(x)$ نماینده شدت روشنایی پیکسل x، و $I'(x)$ نماینده مشتق اول(گرادیان شدت روشنایی) در پیکسل x باشد، بنابراین داریم:

$$I'(x) = -1 \cdot I(x-1) + 0 \cdot I(x) + 1 \cdot I(x+1)$$

۴-۲-محاسبه مشتق دوم

برخی دیگر از الگوریتم‌های آشکارسازی لبه بر اساس مشتق دوم شدت روشنایی کار می‌کنند که درواقع نرخ تغییرات گرادیان شدت روشنایی است و برای آشکارسازی خط‌ها بهترین است. زیرا به‌صورتی که در بالا بیان‌شد هر خط یک لبه دوگانه است، بنابراین در یک سوی خط یک گرادیان روشنایی و در سوی دیگر گرادیان مخالف آن دیده می‌شود. پس می‌توانیم منتظر تغییر بسیار زیاد در گرادیان شدت روشنایی در محل یک خط بود. برای یافتن خط‌ها می‌توانیم گذر از صفرهای تغییر گرادیان را در نتایج جستجو کرد. اگر $I(x)$ نمایشگر شدت نور در نقطه x و $I''(x)$ مشتق دوم در نقطه x باشد:

$$I''(x) = 1 \cdot I(x-1) - 2 \cdot I(x) + 1 \cdot I(x+1)$$

۴-۳-آستانه گیری

هنگامی که مشتق محسابه شد، گام بعدی اعمال یک آستانه برای کشف نقاطی که بخشی از یک لبه هستند، است. هر چه آستانه کمتر باشد، خط‌های بیشتری آشکارسازی می‌گردند و نتایج بیشتر نسبت به نویز، و ویژگی‌های نامرتبط تصویر حساس می‌شوند. از سوی دیگر یک آستانه زیاد ممکن است خط‌های ضعیف یا بخش‌هایی از خط‌ها را از دست بدهد. یک مصالحه معمول، آستانه‌گیری با پسماند است. این روش از چندین آستانه برای جستن لبه‌ها سود می‌جوید. با آستانه بالایی، جستجو برای پیدا کردن ابتدای خط‌ها آغاز می‌شود. هنگامی که یک نقطه آغاز داریم، مسیر لبه را درون تصویر پیکسل به پیکسل با نشانه‌گذاری پیکسل‌هایی که از آستانه پایینی بالاترند پیگیری می‌شوند و تنها هنگامی که مقدار از آستانه پایینی، پایین‌تر رود، فرایند خاتمه می‌یابد. این رهیافت بر اساس این گمان است که لبه‌ها به احتمال زیاد در مسیرهای پیوسته قرار دارند و دنبال کردن بخش ضعیفی از لبه‌ای که از پیش مشخص شده‌اند را ممکن می‌کند، بدون آن‌که پیکسل‌های نویزی به عنوان لبه نشانه‌گذاری شوند.

۴-۴-عملگرهای آشکارسازی لبه

مرتبه نخست :رابرتز، پرویت، سوبل، کنی، اسپیسک

مرتبه دوم :لاپلاسی، مار-هیلدرث

در اینجا دو الگوریتم سوبل و کنی پیاده سازی شده است.

۴-۵-بخش برنامه نویسی

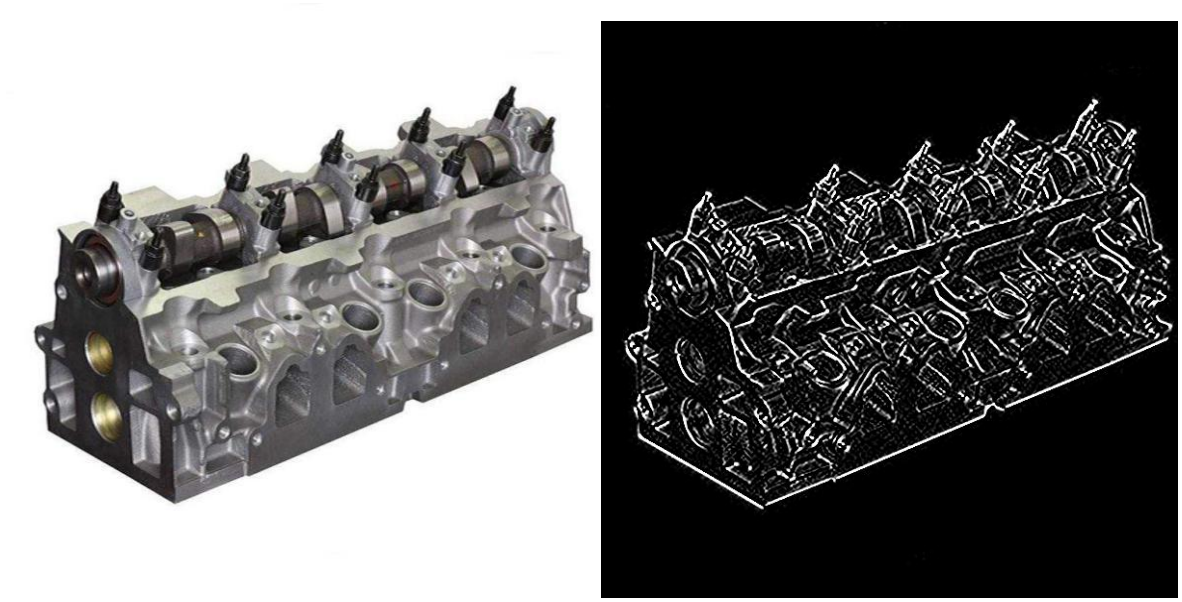
کتابخانه ی cv2 از هر دو تکنیک پشتیبانی می کند. ابتدا عکس را با استفاده از خود کتابخانه می خوانیم و یک img object به وجودا می آوریم و سپس آن را خاکستری (grayscale) می کنیم.

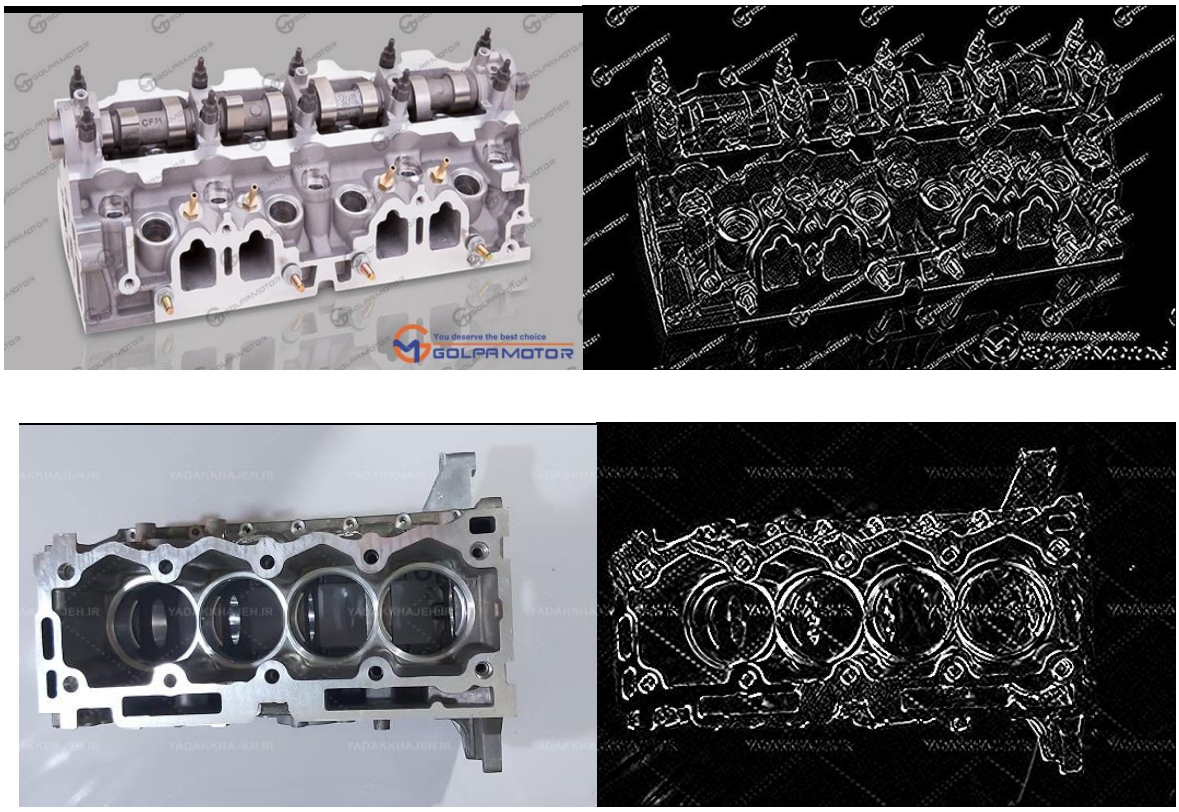
برای استفاده از الگوریتم سوبل از تابع Sobel استفاده می کنیم که ۵ ورودی می گیرد. ورودی اول عکس ورودی ، بعدی عمق عکس خروجی ، دو ورودی بعد محور های مد نظر و ورودی آخر سایز کرنل است.

منظور از محور های مد نظر محورهاییست که نسبت به آن ها مشتق گرفته می شود.

خروجی عکس های سمت چپ با مشتق گرفتن نسبت به هر دو محور، به صورت عکس های سمت راست است:

شکل۱-۴ مثال لبه یاب سوبل



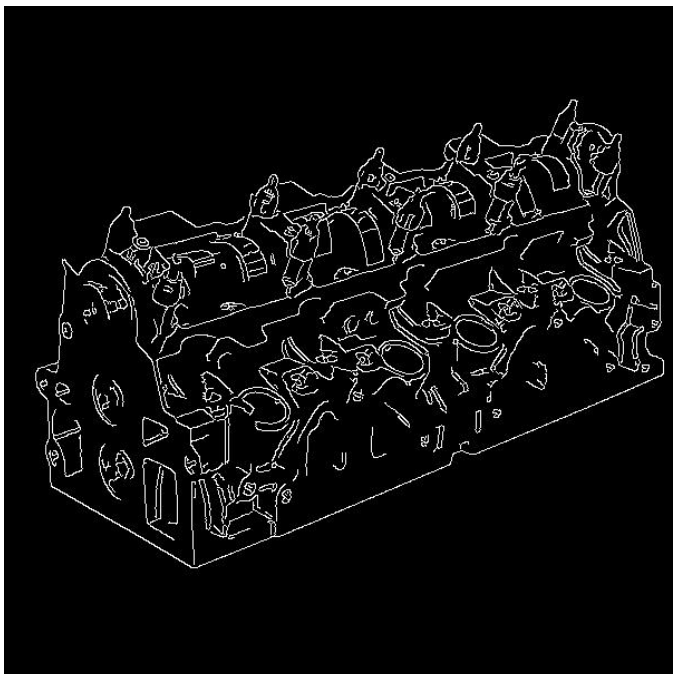


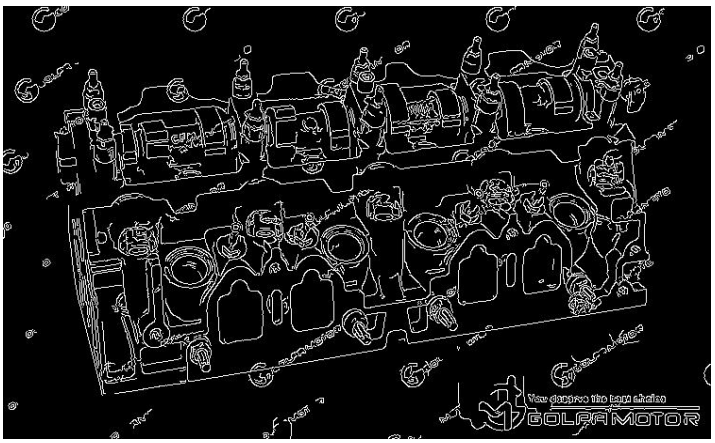
برای پیاده سازی تکنیک کنی نیز مانند تکینک قبل ، بعد از خاکستری کردن از تابع Canny استفاده مـی کنیم. که سه ورودی می گیـرد. ورودی اول آن عکـس ورودی و دو ورودی بعـدی آسـتانه ی بـالا و پـایین است.

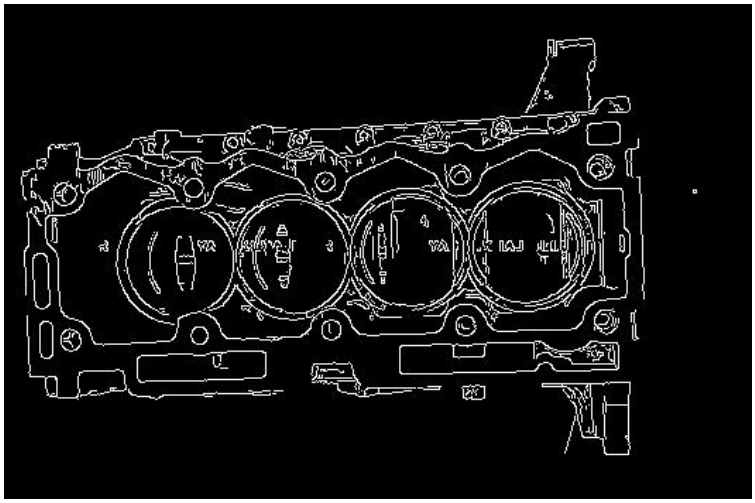
در Canny از روشی به نام hysteresis استفاده میشه. بـرای ایـن منظـور دو آسـتانه بـالا و پـایین تعریـف می‌کنیم هر پیکسل که دارای gradient بیشتر از حد بالا باشه به عنوان لبه پذیرفتـه می‌شـه و در صـورتی که دارای مقدار کمتر از حد پایین باشه رد می‌شه و در صورتی که دارای مقداری بین این دو حـد باشـه در صورتی پذیرفته می‌شه که یکی از همسایه های آن پذیرفته شده باشه.

خروجی عکس های مرحله قبل استفاده از لبه یابی کنی ، عکس های زیر است:

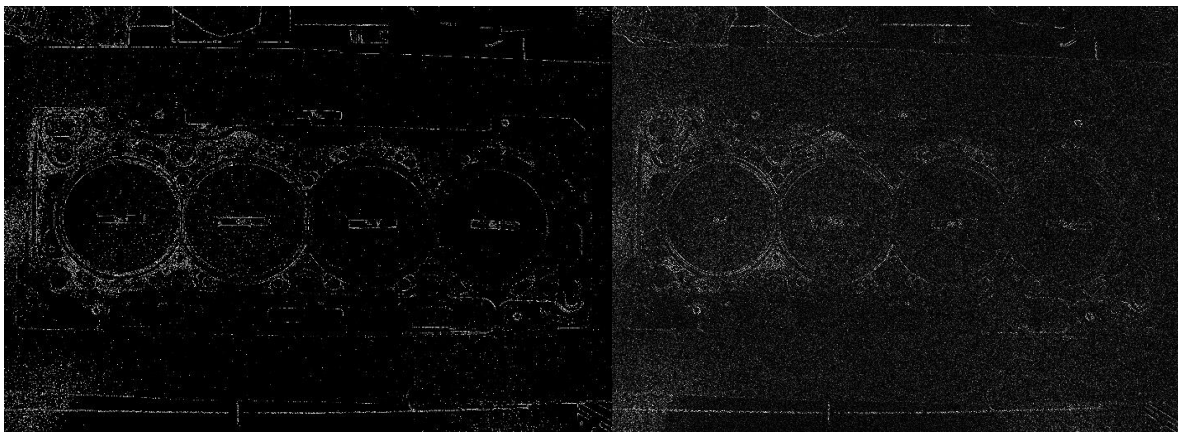
شکل۲-۴ مثال لبه یاب کنی

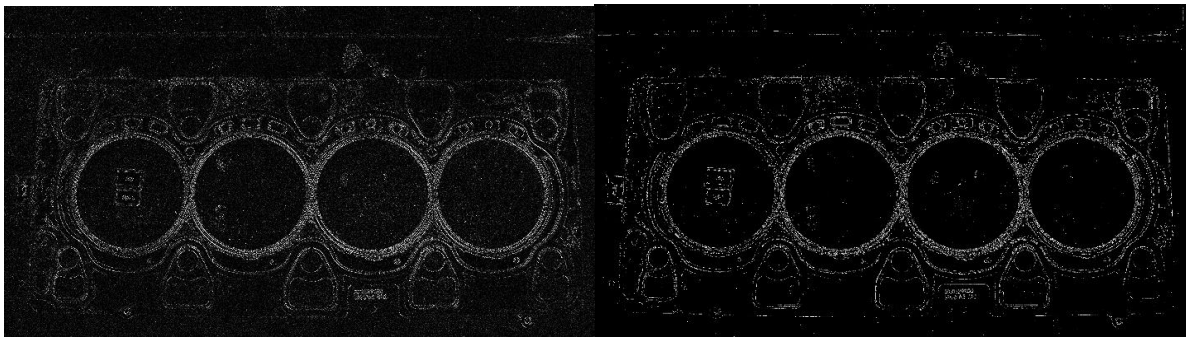


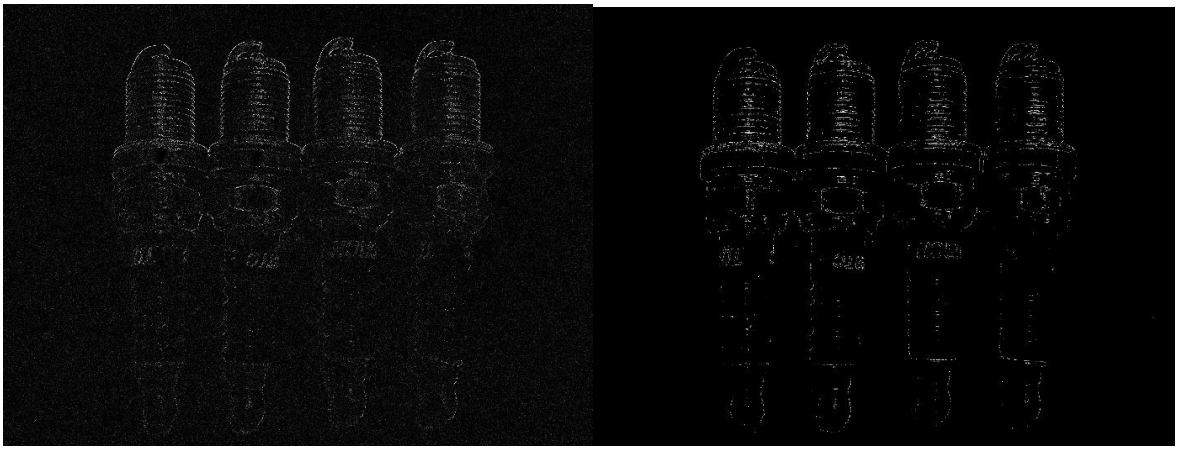




شکل۳-۴ عکس های صنعتی ایپکو (لبه یابی)








Anti shock
Anti shock

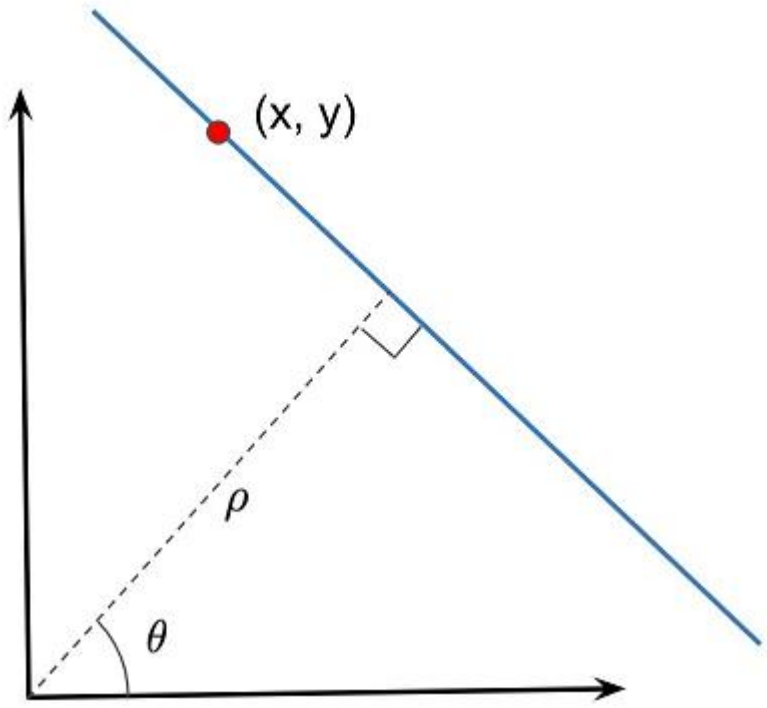
فصل پنجم

تبدیل هاف

تبدیل هاف

تبدیل هاف (Hough Transform) تکنیکی است که به وسیله آن می‌توان خطوط راست و حتی اشکال دایره‌ای را در یک تصویر تشخیص داد. در واقع می‌توان تبدیل هاف را یک شیوه استخراج ویژگی دانست که به وسیله آن شکل‌های ساده مانند خط و دایره در یک تصویر تشخیص داده می‌شوند. توجه کنید که منظور از یک شکل ساده، شکلی است که بتوان آن را تنها با استفاده از تعداد کمی پارامتر نشان داد. به عنوان مثال، یک خط را می‌توان فقط به کمک دو پارامتر شیب و عرض از مبدا نشان داد. همچنین دایره با کمک ۳ پارامتر قابل نشان دادن است که عبارتند از مختصات X و Y و نیز شعاع R دایره. تبدیل هاف در یافتن این شکل‌ها در یک تصویر به صورت فوق‌العاده عمل می‌کند. یکی از مهم‌ترین مزایای تبدیل هاف در یافتن خط و دایره در تصویر این است که نسبت به همپوشانی حساس نیست.

شکل۱-۵ یک خط راست در مختصات قطبی



معادله یک خط در **مختصات قطبی** به صورت زیر نوشته می‌شود:

$$\rho=x\cos(\theta)+y\sin(\theta)$$

در این رابطه، ρ برابر با فاصله عمودی خط از مبدا بر حسب پیکسل و θ **زاویه** خط با مبدا است که بر حسب رادیان اندازه گرفته می‌شود. این موارد در تصویر فوق نیز به خوبی نشان داده شده است. ممکن است این سوال پیش بیاید که چرا از معادله خط در **مختصات دکارتی** که به صورت زیر است، استفاده نمی‌کنیم:

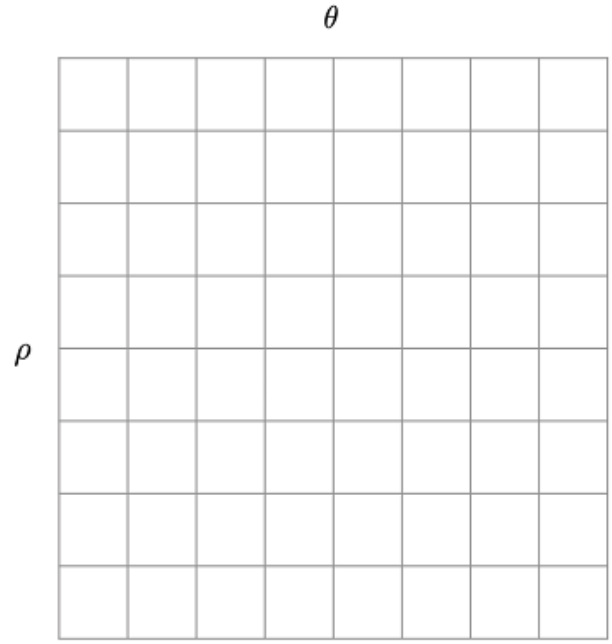
$$y=mx+cy=mx+c$$

دلیل عدم استفاده از مختصات دکارتی این است که مقدار شیب خط یا m ، می‌تواند مقادیر بین $-\infty$ تا ∞ را به خود اختصاص دهد، در حالی که در تبدیل هاف مقادیر پارامترها باید محدود باشند. همچنین سوال دیگری که ممکن است در مورد تبدیل هاف پیش بیاید این است که در معادله مربوط به خط در مختصات قطبی، مقدار θ محدود است، اما آیا پارامتر ρ در بازه مقادیر بین ۰ تا ∞ نیست؟ در پاسخ باید گفت این مقادیر تنها از لحاظ تئوری درست هستند، اما در عمل پارامتر ρ نیز محدود است؛ زیرا خود تصویر محدود است.

۱-۵-انباشتگر (Accumulator)

زمانی که می‌گوییم یک خط در فضای دو بعدی با دو پارامتر ρ و θ مشخص می‌شود، به این معنی است که اگر دو مقدار تصادفی برای پارامترهای ρ و θ انتخاب کنیم، در این صورت یک خط در فضای دو بعدی به دست می‌آید. یک آرایه دو بعدی را در نظر بگیرید که محور x دارای تمام مقادیر ممکن برای θ و محور y دارای تمام مقادیر ممکن برای ρ باشد. هر عضو این آرایه دو بعدی متناظر با یک خط در فضا است. به این آرایه دو بعدی انباشتگر می‌گویند. در تصویر زیر نمایی از این مفهوم نشان داده شده است.

شکل۲-۵ انباشتگر



از مقادیر موجود در آرایه دو بعدی انباشتگر برای جمع‌آوری اطلاعات درباره این امر استفاده می‌کنیم که کدام خطوط در تصویر وجود دارند. سلول بالا سمت چپ، متناظر با $(-R,0)$ و سلول پایین سمت راست متناظر با (R,π) است. هر چقدر مدارک بیشتری درباره حضور یک خط با پارامتر ρ و θجمع‌آوری شود، خواهیم دید که مقادیر درون سلول‌های این آرایه دو بعدی کم کم (ρ,θ) افزایش خواهند یافت. برای تشخیص یک خط در تصویر باید گام‌های زیر را انجام دهیم.

۲-۵-گام اول: مقداردهی اولیه (Initialize) آرایه دو بعدی

ابتدا لازم است که یک آرایه انباشتگر بسازیم. تعداد سلول‌هایی که در شبکه حضور دارند، یک پارامتر طراحی است که باید آن را مشخص کرد. حال فرض کنید یک شبکه انباشتگر ۱۰ در ۱۰ در انتخاب کرده‌ایم. این امر بدین معنی است که ρ فقط ۱۰ مقدار متمایز می‌تواند بگیرد، همچنین θ نیز فقط ۱۰ مقدار متفاوت را می‌تواند به خود اختصاص دهد. بنابراین در حالت کلی ما قادر به تشخیص ۱۰۰ خط متفاوت هستیم. البته اندازه انباشتگر که انتخاب می‌کنیم به رزولوشن تصویر نیز بستگی دارد.

۵-۳-گام دوم: لبه یابی (Edge Detection)

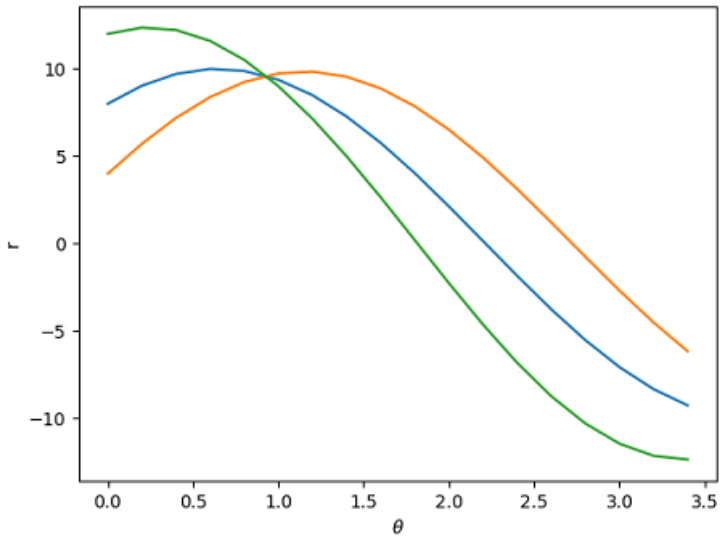
حال که ابعاد شبکه را انتخاب کردیم و انباشتگر نیز تنظیم شد، می‌خواهیم برای هر سلول در انباشتگر مدارک کافی جمع‌آوری کنیم؛ زیرا هر سلول در این شبکه متناظر با یک خط است. اما ایده‌ای که در پس جمع‌آوری مدارک وجود دارد این است که اگر یک خط مرئی در تصویر وجود داشته باشد، **الگوریتم تشخیص لبه** در مرزهای خط فعال (Fire) می‌شود و این مرزها را نشان دهد. پیکسل‌های لبه تصویر می‌تواند مدرک کافی برای حضور یک خط در تصویر را فراهم کنند. خروجی الگوریتم تشخیص لبه یک آرایه از پیکسل‌های لبه تصاویر به صورت زیر است:

$$[(x_1,y_1),(x_2,y_2)...(x_n,y_n)]$$

۵-۴-گام سوم: انتخاب پیکسل‌های لبه

برای هر پیکسل لبه (x,y) در آرایه فوق، مقدار θ را در بازه ۰ تا π تغییر می‌دهیم و آن را در معادله خط در مختصات قطبی جایگزین می‌کنیم تا یک مقدار برای ρ به دست آید. در تصویر زیر مقادیر θ را در این بازه برای سه پیکسل تغییر داده‌ایم و مقادیر ρ را با استفاده از معادله خط به دست آورده‌ایم. این سه پیکسل توسط سه منحنی رنگی نشان داده شده‌اند.

شکل۳-۵ انتخاب پیکسل های لبه



همان طور که در تصویر بالا دیده می‌شود، منحنی‌ها در یک نقطه با همه برخورد می‌کنند که نشان دهنده این است که یک خط با پارامترهای $\theta=1$ و $\rho=0.95$ از آن نقطه عبور می‌کند. معمولا در یک تصویر صدها پیکسل مربوط به لبه داریم و از انباشتگر برای یافتن تقاطع تمام منحنی‌های ایجاد شده توسط پیکسل‌های لبه استفاده می‌کنیم. در ادامه به بررسی نحوه انجام این کار می‌پردازیم.

فرض کنید که انباشتگر سایز ۲۰ در ۲۰ داشته باشد. بنابراین ۲۰ مقدار متمایز برای θ وجود دارد و در نتیجه برای هر پیکسل لبه (x,y) نیز ۲۰ جفت (ρ,θ) را با استفاده از معادله قطبی خط خواهیم داشت. مقادیر سلول‌های انباشتگر متناظر با این ۲۰ مقدار (ρ,θ) به تدریج افزایش می‌یابند. برای هر پیکسل لبه این محاسبات را انجام می‌دهیم و در نتیجه یک انباشتگر خواهیم داشت که دارای اطلاعات کافی درباره تمام خطوط موجود در تصویر است.

می‌توانیم به سادگی تمام سلول‌های انباشتگر که بالاتر از یک حد آستانه هستند را انتخاب کنیم تا خطوط تصویر را پیدا کنیم. هر چقدر میزان آستانه بالاتر باشد، تعداد خطوط قوی کمتری را پیدا خواهیم کرد و هر چه آستانه را مقداری پایین‌تر قرار دهیم، در این صورت تعداد خطوط بیشتری را به دست می‌آوریم که حتی شامل خطوط ضعیف نیز هستند.

۵-۵-بخش برنامه نویسی

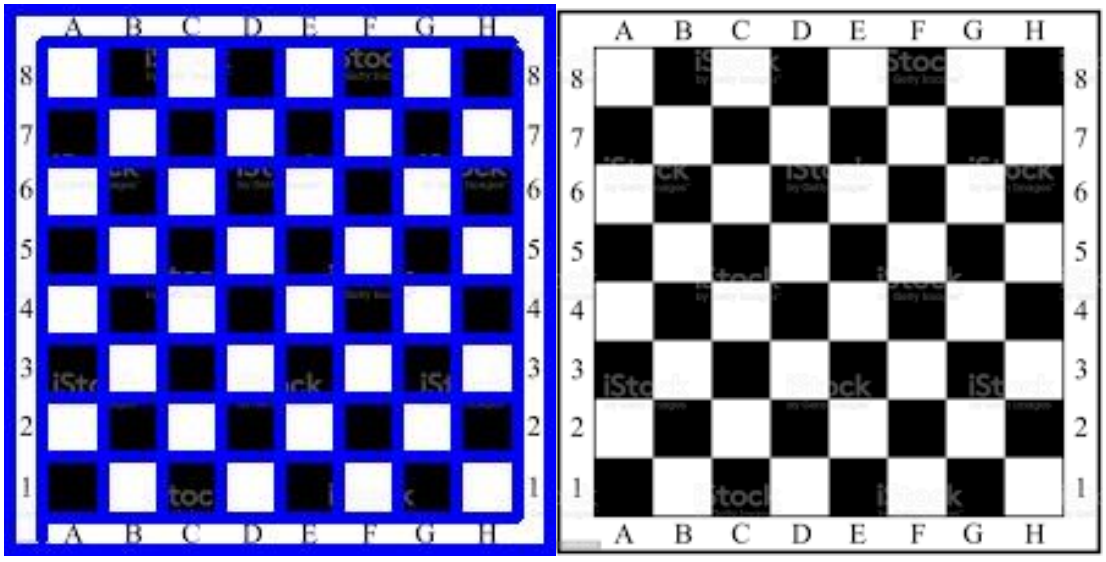
در کتابخانه OpenCV تشخیص خط با استفاده از تبدیل هاف تابع HoughLines یا HoughLinesP پیاده‌سازی کرد HoughLines .مربوط به نوع دیگری از تبدیل هاف است که تبدیل هاف احتمالات نام دارد. این تابع آرگومان‌های زیر را به عنوان ورودی دریافت می‌کند.

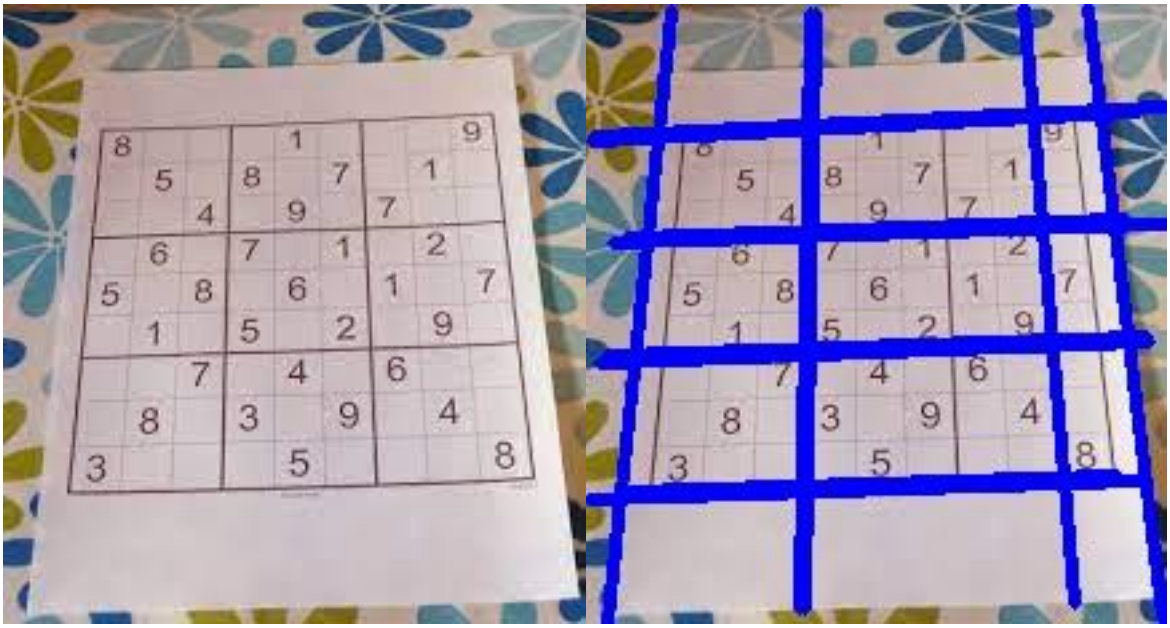
- **edges:** خروجی تابع لبه یاب (در اینجا از لبه یاب کنی که در قسمت قبل آورده شد استفاده شده است.)
- **lines:** یک بردار برای ذخیره‌سازی مختصات شروع و پایان خطوط.
- **theta:** پارامتر رزولوشن ρ در پیکسل‌ها.

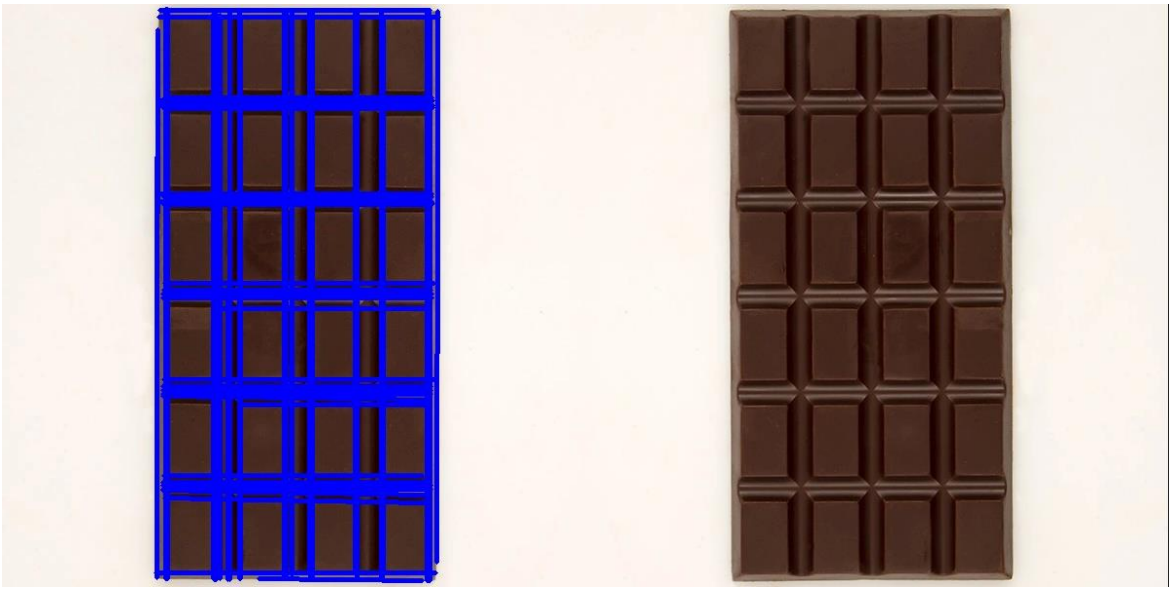
- **threshold:** تعداد کمینه نقاط تقاطع برا تشخیص یک خط.

به عنوان مثال ، در شکل سمت چپ خطوط آبی ، خطوط تشخیص داده شده در شکل سمت راست هستند:

شکل۴-۵ مثال تبدیل هاف







فصل ششم

جمع‌بندی و نتیجه‌گیری و پیشنهادات

جمع‌بندی و نتیجه‌گیری

در این پروژه بعد از معرفی بخش بندی تصویر به عنوان یکی از مهمترین ارکان پردازش تصویر ، سه تـا از مهمترین و اصلی ترین روش های آن یعنی آستانه گذاری ، لبه یابی و تبدیل هاف از نظر تئوری ، ریاضی و برنامه نویسی بررسی شد و سپس کارآیی آن ها در مجموعـه متنـوعی از مثـال هـای عـادی و صـنعتی نشان داده شد.

در بخش آستانه گذاری دیدیم که با استفاده از این حقیقت که یک عکس مجموعه ای از پیکسل هاسـت که عدد هر پیکسل به معنای روشنایی آن بخش از تصویر است و می توان این گونه با استفاده از تکنیـک های مختلف و گذاشتن یک حد آستانه اشیا را در آن تصویر تشخیص داد. همچنین دیدیم که با دانسـتن نمایش عددی یک طیف رنگی می توان پیکسل هایی که در آن محدوده ی عددی قرار دارند را پیدا کـرد و بخش رنگی مختلف یک تصویر را از هم جدا کرد.

در بخش لبه یابی نیز روابط پیکسل ها با یکدیگر در یک محدوده را بررسی شد و با اسـتفاده از ابزارهـای ریاضی مثل مشتق اول و دوم مرز های اشیا تشخیص داده شد و دو تکنیک معروف در ایـن زمینـه ، لبـه یاب های کنی و سوبل را معرفی شد.

در بخش بعدی یعنی تبدیل هاف با استفاده از خروجی بخش قبل و با تکیه بر ابزارهـای ریاضـی ، روشـی برای تشخیص خطوط مستقیم تصویر ارائه شد.

انتقادات و پیشنهادات

در مجموع خروجی های این تکنیک ها اطلاعات بسیار مفید و قابل اسـتفاده بـرای دیگـر تکنیـک هـای کامپیوتری خوبی را به ما می دهد که می توان از آن ها برای استخراج اطلاعات از تصویر استفاده کرد اما برای کارآیی بهتر پیشنهاد به استفاده از عکس هایی با کیفیت بالا و تنظیم دقیق عملگـر هـا و همچنـین همکاری آن ها با دیگر تکنیک های سامانه پردازش تصویر می باشد.

منابع و مراجع

- Golnabi, H., & Asadpour, A. (2007). Design and application of industrial machine vision systems. Robotics and Computer-Integrated Manufacturing, 23(6), 630-637.
- Zhou, Q., Chen, R., Huang, B., Liu, C., Yu, J., & Yu, X. (2019). An automatic surface defect inspection system for automobiles using machine vision methods. Sensors, 19(3), 644.
- Sathiyamoorthy, S. (2014). Industrial application of machine vision. International Journal of Research in Engineering and Technology (IJRET), 3(7), 678-682.
- Kochan, A. (2002). Machine vision guides the automotive industry. Sensor Review.
- Capela, S., Silva, R., Khanal, S. R., Campaniço, A. T., Barroso, J., & Filipe, V. (2020, July). Engine Labels Detection for Vehicle Quality Verification in the Assembly Line: A Machine Vision Approach. In Portuguese Conference on Automatic Control (pp. 740-751). Springer, Cham.
- Mizushima, A., & Lu, R. (2013). An image segmentation method for apple sorting and grading using support vector machine and Otsu's method. Computers and electronics in agriculture, 94, 29-37.
- Wu, J., Zeng, P., Zhou, Y., & Olivier, C. (2006, November). A novel color image segmentation method and its application to white blood cell image analysis. In 2006 8th international Conference on Signal Processing (Vol. 2). IEEE.
- Li, F., Ng, M. K., Zeng, T. Y., & Shen, C. (2010). A multiphase image segmentation method based on fuzzy region competition. SIAM Journal on Imaging Sciences, 3(3), 277-299.