
بسم الله الرحمن الرحيم

دانشگاه فنی و حرفه ای
دانشکده انقلاب اسلامی تهران

پایان نامه کارشناسی

تشخیص پلاک خودرو با استفاده از متلب

ارائه دهندگان :
محمد پورمندنی
هادی محمدی

انستیتو برق
تیر ماه ۱۳۹۲

چکیده

مکان یابی پلاک به عنوان اولین گام در جهت شناسایی پلاک وسا یل نقلیه مطرح است .در این مقاله روشی جهت یافتن مکان پلاک خودرو در تصاویر با ویژگی ها و پس زمینه های متنوع و پیچیده ارائه می گردد . الگوریتم پیشنهادی ، یکبار بر روی تصویر اصلی ، اجرا می شود و از مفاهیم ساده ای مانند آستانه گذاری، عملیات مورفولوژی و تشخیص لبه برای قطعه بندی تصویر استفاده مینماید . سپس با بررسی برخی معیارهای هندسی پلاک مانند مساحت، نسبت طول به عرض، چگالی لبه ها و ... نواحی کاندید پلاک مشخص میگردد .در نهایت ناحیه کاندید انتخاب شده جدا میشود و تک تک ارقام ان با یک تکنیک مناسب از هم جدا شده و با یک معیار اعتماد جهت تشخیص میزان شباهت بین ارقام جدا شده پلاک خودرو با مقایسه ارقام ذخیره شده در سیستم شماره های پلاک تعیین می گردند . این روش برای تعدادی از تصاویر واقعی که دارای شرایط تصویربرداری متفاوت می باشند، نتایج مناسبی داشته است

کلید واژه

پلاک خودرو ، پردازش تصویر ، لبه یابی

فهرست مطالب

عنـــوان **صفحه**

فصل اول

مقدمه

۱-۱) مقدمه

به مجموعه عملیات و پردازش هایی که در راستای انالیز تصویر در زمینه های مختلف انجام شده است، علم پردازش تصویر گویند.

از سال ۱۹۶۴ تاکنون، موضوع پردازش تصویر، رشد فراوانی کرده است. علاوه بر برنامه تحقیقات فضایی، اکنون از فنون پردازش تصویر، در موارد متعددی استفاده می شود. گر چه اغلب این مسائل با هم نامرتبط هستند، اما عموما نیازمند روش هایی هستند که قادر به ارتقای اطلاعات تصویری برای تعبیر و تحلیل انسان باشد. برای نمونه در پزشکی شیوه های رایانه ای کنتراست تصویر را ارتقا می دهند یا این که برای تعبیر آسانتر تصاویر اشعه ایکس یا سایر تصاویر پزشکی، سطوح شدت روشنایی را با رنگ، رمز می کنند. متخصصان جغرافیایی نیز از این روش ها یا روش های مشابه برای مطالعه الگوهای آلودگی هوا که با تصویر برداری هوایی و ماهواره ای بدست آمده است، استفاده می کنند. در باستان شناسی نیز روش های پردازش تصویر برای بازیابی عکس های مات شده ای که تنها باقی مانده آثار هنری نادر هستند، مورد استفاده قرار می گیرد. در فیزیک و زمینه های مرتبط، فنون رایانه ای بارها تصاویر آزمایش های مربوط به موضوعاتی نظیر پلاسماهای پرانرژی و تصاویر ریزبینی الکترونی را ارتقا داده اند. کاربردهای موفق دیگری از پردازش تصویر را نیز می توان در نجوم، زیست شناسی، پزشکی هسته ای، اجرای قانون، دفع و صنعت بیان کرد.

در اوایل دهه ۶۰ سفینه فضایی رنجر ۷ متعلق به ناسا شروع به ارسال تصاویر تلویزیونی مبهمی از سطح ماه به زمین کرد. استخراج جزئیات تصویر برای یافتن محلی برای فرود سفینه آپولو نیازمند اعمال تصمیماتی روی تصاویر بود. این کار مهم به عهده لابراتوار (JPL) نیروی جت قرار داده شد. بدین ترتیب زمینه تخصصی پردازش تصاویر دیجیتالی آغاز گردید و مثل تمام تکنولوژی های دیگر سریعاً استفاده های متعدد پیدا کرد.

ابتدایی ترین کاربردهای پردازش تصاویر دیجیتالی در دهه ۶۰ و۷۰ جنبه های نظامی و جاسوسی بود که باعث شد نیاز به تصاویر با کیفیت بالاتر بوجود آید. پس از آن مصارف دیگری برای تصاویر دیجیتالی سطح زمین پیدا شد که کاربرد تصاویر چند طیفی در کشاورزی و جنگل داری از آن جمله است. همچنین با استفاده از تصاویر دیجیتالی عملیاتهایی مثل کنکاش نفت در سرزمین های دور افتاده و یا ردیابی منابع

آلودگی شهری از داخل دفتر کار متخصصین آنها انجام شد.

بزودی کاربردهای زمینی زیادتری برای پردازش تصاویر دیجیتالی پیدا شد . از اواسط دهه ۷۰ تا اواسط دهه ۸۰ اختراع اسکنر ها ی CAT و اسکنر های MRI پزشکی را متحول کردند. صنعت چاپ استفاده کننده بعدی بود. در اواخر دهه ۸۰ پردازش تصاویر دیجیتالی وارد دنیای سرگرمی شد بطوریکه امروزه این نقش به امر عادی تبدیل شده است. بهمین ترتیب دنیای صنعت با روباتهایی که عملا می بینند یعنی در واقع با ظهور تکنولوژی بینایی ماشین متحول شد و هنوز هم در حال تحول است.

هر ساله با سریعتر و ارزانتر شدن کامپیوتر ها و ایجاد امکان پخش تصاویر با استفاده از تکنولوژی ارتباطات، افراد بیشتری به این تصاویر دسترسی پیدا می کنند. کنفرانس های ویدئویی یک روش زنده برای انجام کسب و کار شده اند و کامپیوترها ی خانگی توانایی نمایش و مدیریت تصاویر را به خوبی پیدا کرده اند. خوشبختانه با بالاتر رفتن سرعت پردازش و فضای حافظه کامپیوترها دیگر از بابت امکانات پردازش تصاویر نگرانی ها کمتر شده است و روز به روز این روند رو به رشد ادامه پیدا می کند.

علم پردازش تصویر، از علوم پرکاربرد و مفید در فنون مهندسی می باشد و از دیر باز مطالعات و تحقیقات گسترده ای در این زمینه صورت گرفته و پیشرفت های فراوانی حاصل شده است.

سرعت گسترش این پیشرفت ها به حدی بوده است، که هم اکنون و پس از گذشت مدت زمان کوتاهی می توان تأثیر پردازش تصویر را در بسیاری از علوم و صنایع به وضوح مشاهده نمود.

در حالی که برخی از این کاربردها، ان گونه به پردازش تصویر وابسته است، که بدون ان اساساً قابل استفاده نمی باشد.

نظر به این که علم پردازش تصویر به صورت جامع و تخصصی در دنیای امروزی، روز به روز نقش اساسی و مهم تری پیدا می کند و در کشور ما نیز تقریباً در اغاز راه است.

مسئله بزرگی داده های تصویری و تلاش جهت حذف نویز و اختلالات تصویری نظیر پارامترهای حاصل از منابع نوری نامناسب، عدم تناسب ترکیب رنگ ها و عوامل متعدد دیگر در تصاویر دریافتی، از موضوعات بسیار مهم در کار با تصاویر و پردازش انها می باشد.

امروزه با پیشرفت های متعددی که در روش های اخذ اطلاعات گسسته مانند اسکنرها و دوربین های دیجیتالی به وجود امده است، پردازش تصویر کاربرد فراوانی یافته است.

تصاویر حاصل از این اطلاعات همواره در حد قابل توجهی دارای نویز و یا تیرگی محسوس بوده است و در مواردی نیز دارای مشکل محو شدگی مرزهای نمونه های داخل تصویر می باشد، که باعث کاهش وضوح تصویر دریافتی می گردد.

پردازش تصاویر امروزه بیشتر به موضوع پردازش تصویر دیجیتال گفته می‌شود که شاخه‌ای از دانش رایانه است که با پردازش سیگنال دیجیتال که نماینده تصاویر برداشته شده با دوربین دیجیتال یا پویش شده توسط پویشگر هستند سر و کار دارد.

پردازش تصاویر دارای دو شاخه عمده ی **بهبود تصاویر** و **بینایی ماشین** است. بهبود تصاویر دربرگیرنده روشهایی چون استفاده از فیلتر محوکننده و افزایش تضاد برای بهتر کردن کیفیت دیداری تصاویر و اطمینان از نمایش درست آنها در محیط مقصد(مانند چاپگر یا نمایشگر رایانه)است، در حالی که بینایی ماشین به روشهایی می‌پردازد که به کمک آنها می‌توان معنی و محتوای تصاویر را درک کرد تا از آنها در کارهایی چون رباتیک و محور تصاویر استفاده شود.

در معنای خاص آن پردازش تصویر عبارتست از هر نوع پردازش سیگنال که ورودی یک تصویر است مثل عکس یا صحنه‌ای از یک فیلم.خروجی پردازشگر تصویر می‌تواند یک تصویر یا یک مجموعه از نشانهای ویژه یا متغیرهای مربوط به تصویر باشد.اغلب تکنیک‌های پردازش تصویر شامل برخورد با تصویر به عنوان یک سیگنال دو بعدی و بکاربستن تکنیک‌های استاندارد پردازش سیگنال روی آنها می‌شود. پردازش تصویر اغلب به پردازش دیجیتالی تصویر اشاره می‌کند ولی پردازش نوری و آنالوگ تصویر هم وجود دارند.این مقاله در مورد تکنیک‌های کلی است که برای همه آنها به کار می‌رود.

پردازش تصویر چیست؟

به مجموعه عملیاتی که یک ماشین الکترونیکی(مثلا کامپیوتر) به منظور ویرایش تصاویر انجام میدهد پردازش تصویر گفته می شود.

مثال: برنامه معروف شرکت ادوبی می تواند مثال خوبی برای این تعریف باشد.(به تمامی عملیاتی که در برنامه فوتوشاپ بر روی تصاویر انجام میشود پردازش تصویر گفته میشود)

بینایی ماشین چیست ؟

سیستم گرافیکی بشر قادر به تفسیر و شناسایی شیء از پس زمینه میباشد که کامپیوتر با استفاده از الگوریتم ها و راه حل های بینایی ماشین میتواند به این قابلیت برسد. این یک تعریف و مثالی از بینایی ماشین بود

تفاوت پردازش تصویر و بینایی ماشین چیست؟

توجه داشته باشید که در پردازش تصویر کامپیوتر از محتوای مفهومی و اشیایء داخل تصویر اطلاعی ندارد و فقط به عنوان مثال رنگ نقطه مشخصی را تغییر میدهد ولی در بینایی ماشین هدف این است که محتوای معنایی تصویر توسط کامپیوتر تفسیر و درک شود

۱-۲) *کاربرد پردازش تصویر*

چشم به عنوان یکی از حیاتی ترین حسگرهای انسان نقش بسزایی در زندگی ما دارد. امروزه با پیشرفت چشمگیری که در ساخت پردازنده ها صورت گرفته است، این امکان نیز برای ما فراهم شده تا در ساخت روبات ها و سیستم های کنترلی از دوربین به عنوان یک چشم مصنوعی استفاده کنیم.

رد پای پردازش تصویر در بسیاری از علوم و صنایع مشاهده می شود و بعضی از این کاربردها آنچنان به پردازش تصویر وابسته هستند که بدون آن، اساساً قابل استفاده نمی باشند. کاربرد پردازش تصویر در هر یک از زمینه هایی که بحث شد، بسیار گسترده است

پنج کاربرد عمده پردازش تصویر را می توان رباتیک ، سیستم های دفاعی ، مهندسی پزشکی ، کنترل صنعتی و گرافیک کامپیوتری عنوان کرد. در سیستم های رباتیک معمولا از پردازش تصویر برای هدایت ربات و تشخیص اشیا استفاده می شود. در سیستم های دفاعی برای یافتن هدف و یا رهگیری یک هدف متحرک پردازش تصویر یکی از قابل اعتمادترین روش های موجود می باشد. در مورد کاربردهای گرافیکی نیز یکی از معروفترین نرم افزارهای مبتنی بر پردازش تصویر فتوشاپ می باشد که همگی با کاربردهای این نرم افزار آشنا هستیم. تشخیص نوع بیماری نیز رایج ترین کاربرد پردازش تصاویر پزشکی است. در نهایت ، امروزه خطوط تولید صنعتی برای کنترل کیفیت محصولات تولید شده و همچنین کنترل حرکات خط تولید از سیستم های مبتنی بر پردازش تصویر بهره جسته اند.

اما دیگر زمینه های کاربرد پردازش تصویر عبارتند از صنعت، هواشناسی، شهرسازی، کشاورزی، علوم نظامی و امنیتی، نجوم و فضا نوردی، پزشکی، فناوری های علمی، باستان شناسی، تبلیغات، سینما، اقتصاد، روانشناسی و زمین شناسی که در ادامه درباره هر کدام مختصراً بحث شده است.

۱-۲-۱) صنعت

امروزه کمتر کارخانه پیشرفته ای وجود دارد که بخشی از خط تولید آن توسط برنامه های هوشمند بینایی ماشین کنترل نشود. خطای بسیار کم، سرعت زیاد، هزینه نگهداری بسیار پایین، عدم نیاز به حضور اپراتور ۲۴ ساعته و خیلی مزایای دیگر باعث شده که صنایع و کارخانه ها به سرعت به سمت پردازش تصویر و بینایی ماشین روی بیاورند. دستگاهی ساخته شده که قادر است کیک های پخته را از کیک هایی که نیاز به پخت مجدد دارند، تشخیص دهد و آنها را به صورت اتوماتیک به بسته بندی بفرستد و کیک هایی که نیاز به پخت دارند را دوباره برای پختن ارسال کند.

یکی دیگر از دلایل استفاده از بینایی ماشین قابلیت دیدن و اندازه گیری محصولاتی است که دیدن یا اندازه گیری آنها با چشم غیر مسلح غیر ممکن است. عناصر تشکیل دهنده یک سیستم بینایی ماشین نرم افزار هوشمند بینایی است که ورودی خود را از دوربین های نصب شده در بخش های مختلف خط تولید می گیرد و بر اساس تصاویر دریافتی دستورات لازم برای کنترل ماشین های صنعتی را صادر می کند.

پردازش تصویر در تشخیص دمای کوره هایی که هیچ وسیله ی مکانیکی و الکترونیکی تحمل دمای آنها را ندارد، کاربرد دارد. دوربین های حرارتی می توانند مشکل بخشی از سازه ی مورد نظر را تشخیص دهند.

۱-۲-۲) صنایع چوب

صنایع چوب یکی از پر کاربرد ترین صنایع در عصر حاضر است.

این صنعت قدیمی روز به روز در حال پیشرفت در زمینه های مختلف آن می باشد.

اکنون دیگر صنایع چوب به یک صنعت آمیخته با هنر تبدیل شده است.

همان طور که می دانیم برش و حالت دهی از جمله مهمترین و کلیدی ترین کار های صنعت چوب می باشد.

اما همیشه یک مشکل اساسی در برش صحیح چوب وجود داشت و آن هم این بود که چگونه چوب به حالتی برش شود که کمترین میزان اتلاف چوب را داشته باشد و نیز بعد از برش چگونه می توان صحیح بودن برش را کنترل کرد.

این مشکل نیز به راحتی توسط پردازش تصویر قابل حل است.

بعد از این که برش یک قسمت از چوب تمام شد ، با استفاده از یک دوربین آن قسمت را کنترل می کنیم تا نقصی از لحاظ برش وجود نداشته باشد.

۱-۲-۳) شمارش گر

بحث شمارش، جزء لاینفک بسته بندی کالاهای مختلف می باشد.

زمانی که تعداد بسته بندی ها بالا رود ، این کار یک کار خسته کننده و طاقت فرسا برای انسان به نظر می آید.

اما شاید ساده ترین کار در بحث پردازش تصویر ، شمارش باشد.

شمارش تعداد به خودی خود شامل چندین موضوع می شود؛ از جمله : شمارش اجزای داخل بسته بندی ، شمارش اجزای روی نوار نقاله و ... .

۱-۲-۴) اتوماسیون صنعتی

کنترل ماشین آلات و تجهیزات صنعتی یکی از وظایف مهم در فرآیندهای تولیدی است. بکارگیری کنترل خودکار و اتوماسیون روزبه روز گسترده تر شده و رویکردهای جدید با بهره گیری از تکنولوژی‌های نو امکان رقابت در تولید را فراهم می‌سازد. لازمه افزایش کیفیت و کمیت یک محصول، استفاده از ماشین آلات پیشرفته و اتوماتیک می‌باشد. ماشین آلاتی که بیشتر مراحل کاری آنها به طور خودکار صورت گرفته و اتکای آن به عوامل انسانی کمتر باشد. امروزه استفاده از تکنولوژی ماشین بینایی و تکنیک‌های پردازش تصویر کاربرد گسترده‌ای در صنعت پیدا کرده‌است و کاربرد آن به ویژه در کنترل کیفیت محصولات تولیدی، هدایت روبات و مکانیزم‌های خود هدایت شونده روز به روز گسترده تر می‌شود.

با استفاده از تکنیکهای پردازش تصویر می‌توان دگرگونی اساسی در خطوط تولید ایجاد کرد. بسیاری از پروسه‌های صنعتی که تا چند دهه پیش پیاده سازیشان دور از انتظار بود، هم اکنون با بهرگیری از پردازش هوشمند تصاویر به مرحله عمل رسیده‌اند. از جمله منافع کاربرد پردازش تصویر به شرح زیر است.

۱. افزایش سرعت و کیفیت تولی
۲. کاهش ضایعات
۳. اصلاح روند تولید
۴. گسترش کنترل کیفیت

عدم اطلاع کافی مهندسین از تکنولوژی ماشین بینایی و عدم آشنایی با توجیه اقتصادی بکارگیری آن موجب شده‌است که در استفاده از این تکنولوژی تردید و در بعضی مواقع واکنش منفی وجود داشته باشد. علی رغم این موضوع، ماشین بینایی روز به روز کاربرد بیشتری پیدا کرده و روند رشد آن چشمگیر بوده‌است عملیات پردازش تصویر در حقیقت مقایسه دو مجموعه عدد است که اگر تفاوت این دو مجموعه از یک محدوده خاص فراتر رود، از پذیرفتن محصول امتناع شده و در غیر این‌صورت محصول پذیرفته می‌شود. در زیر پروژه‌هایی که در زمینه پردازش تصاویر پیاده سازی شده‌است، توضیح داده می‌شود. این پروژه‌ها با استفاده از پردازش تصویر، شمارش و اندازه گیری اشیا، تشخیص عیوب، تشخیص ترک، دسته بندی اشیا و عملیات بیشمار دیگری را انجام می‌دهند:

۱. اندازه گیری و کالیبراسیون
۲. جداسازی پینهای معیوب
۳. بازرسی لیبل و خواندن بارکد
۴. بازرسی عیوب چوب
۵. بازرسی قرص
۶. بازرسی و دسته بندی زعفران
۷. درجه بندی و دسته بندی کاشی
۸. بازرسی میوه

۹. کالیبراسیون و ابزار دقیق

اندازه گیری دقیق و سنجش فواصل کوچک یکی از دغدغه‌های اصلی در صنایع حساس می‌باشد.دوربینهای با کیفیت امکان کالیبراسیون با دقت بسیار بالا در حد میکرون را فراهم آورده‌اند.

۱-۲-۵) حمل و نقل

۱. شخیص شماره پلاک خودرو

۲. نرم افزار شمارش خودروهای عبوری از عرض خیابان

۳.....

بی شک یکی از مؤثر ترین مولفه‌ها در مدیریت و برنامه ریزی دسترسی به آمار دقیق می‌باشد. درصورت وجود آمار دقیق و سریع می‌توان از روشهای کنترل بهینه استفاده کرد و بهره وری را افزایش داد. به عنوان مثال اگر آمار دقیقی از میزان مصرف یک محصول غذایی وجود داشته باشد با برنامه ریزی مناسب می‌توان زمینه تولید و عرضه اصولی آن را فراهم کرد. لذا احتمال نابسامانی در بازار و متضرر شدن کشاورز و مصرف کننده کاهش می‌یابد. چنان که بیان شد مهمترین فاکتور در برنامه ریزی دسترسی به آمار مناسب است اما تهیه آمار فرایند پیچیده و وقت گیر است و معمولا هزینه زیادی را در بر دارد. به عنوان مثال به دلایلی از جمله کنترل ترافیک یا کنترل میزان روشنایی خیابان باید خودروهای عبوری از خیابان شمارش شوند. این کار اگر به صورت دستی یا انسانی انجام شود، هزینه زیادی نیاز دارد، امکان سهل انگاری انسانی نیز وجود دارد پس استفاده از یک دستگاه مناسب که توانایی شمارش خودروهای عبوری را داشته باشد تنها گزینه ممکن است. با توجه به نیاز فوق نرم افزاری تهیه شده‌است که با استفاده از تصاویر گرفته شده از عرض خیابان خودروهای عبوری را تشخیص می‌دهد و تعداد آنها را شمارش می‌کند. این نرم افزار امکان استفاده در روز یا شب را دارا می‌باشد

۱-۲-۶) هواشناسی

از آنجایی که در علم هواشناسی تشخیص و پیش بینی آب و هوا اکثراً از طریق تصاویر هوایی و ماهواره ای انجام می گیرد، پردازش تصویر در این علم کاربرد زیادی دارد و دقت و سرعت پیش بینی آب و هوا و طوفان ها را بسیار بالا می برد. جبهه های پرفشار، کم فشار، گردبادها و گرداب های بوجود آمده در سطح کره زمین را می توان مشاهده کرد.

۱-۲-۷) شهرسازی

با مقایسه عکس های مختلف از سال های مختلف یک شهر می توان میزان گسترش و پیشرفت آن را مشاهده کرد.

کاربرد دیگر پردازش تصویر می تواند در کنترل ترافیک باشد. با گرفتن عکس های هوایی از زمین ترافیک هر قسمت از شهر مشخص می شود.

قبل از ساختن یک شهر می توان آن را توسط کامپیوتر شبیه سازی کرد که به صورت دو بعدی از بالا و حتی به صورت سه بعدی از دید های مختلف، یک شهرک چطور ممکن است به نظر برسد. تصاویر ماهواره ای که از شهرها گرفته می شود، می تواند توسط فیلترهای مختلف پردازش تصویر فیلترشود و اطلاعات مختلفی از آن استخراج شود. به طور مثال این که شهر در چه قسمت هایی دارای ساختمان ها، آب ها یا راه های بیشتری است و همین طور می توان جاده هایی که داخل یا خارج از شهر کشیده شده اند را تحلیل کرد.

۱-۲-۸) کشاورزی

این علم در بخش کشاورزی معمولاً در دو حالت کاربرد دارد. یکی در پردازش تصاویر گرفته شده از ارتفاعات بالا مثلاً از هواپیما و دیگری در پردازش تصاویر نزدیک به زمین.

در تصاویر دور به عنوان مثال می توان تقسیم بندی اراضی را تحلیل کرد. همچنین می توان با مقایسه تصاویر دریافتی در زمان های متفاوت میزان صدمات احتمالی وارد به محیط زیست را دید. به عنوان مثال می توان برنامه ای نوشت که با توجه به محل رودخانه ها و نوع خاک مناطق مختلف، به صورت اتوماتیک

بهترین نقاط برای کشت محصولات مختلف را تعیین می کند.

تصاویر نزدیک هم در ساخت ماشین های هرز چین اتوماتیک کاربرد دارد. امروزه ماشین های بسیار گران قیمت کشاورزی وجود دارند که می توانند علف های هرز را از گیاهان تشخیص بدهند و به صورت خودکار آنها را نابود کنند.

برای مثال یکی از پروژه های جالب در بخش کشاورزی، تشخیص خودکار گل زعفران برای جداسازی پرچم قرمز رنگ آن بوده است.

۱-۲-۹) علوم نظامی و امنیتی

پردازش تصویر بخصوص بینایی هوشمند، کاربردهای بسیاری را در علوم نظامی و امنیتی دارند و این کاربرد برای دولت اکثر کشورها بسیار مهم است. به عنوان مثال موشک هدایت شونده خودکاری وجود دارد که می تواند روی در یک ساختمان قفل کند و حتی می تواند به درز بین در و دیوار آن ساختمان که حساس ترین جای ساختمان است به راحتی نفوذ کند. این موشک به صورت اتوماتیک این قسمت را شناسایی کرده و به سمت آن حمله می کند.

در مسائل امنیتی هم کاربرد پردازش تصویر کاملاً در زندگی ما مشهود است. دوربین های که به صورت اتوماتیک از ماشین هایی که تخلف رانندگی انجام می دهند عکس برداری می کند.

از سیستم های امنیتی دیگر می توان سیستم تشخیص اثر انگشت اتوماتیک را نام برد. در لپ تاپ های جدید قابلیت اثر انگشت به آنها اضافه شده و می تواند صاحب لپ تاپ را توسط اثر انگشت شناسایی کند.

کد امنیتی دیگری که همیشه همراه انسان حمل می شود، چشم انسان است. دانشمندان ثابت کرده اند که پترن های موجود در مردمک چشم هر انسان منحصر به فرد است و هیچ دو فردی در دنیا وجود ندارند که پترن هایی که در مردمک چشم آنها وجود دارد دقیقاً مثل هم باشد. از همین روش برای شناخت افراد و سیستم های امنیتی استفاده می شود.

در کل این خواص بیومتریک در انسان بسیار زیاد است. عرض و طول صورت، فاصله بین انگشتان دست، طول و عرض انگشت ها، فاصله ی بندها از یکدیگر و حتی خط های کشیده شده کف دست و هزاران خاصیت دیگر، تماماً خصوصیاتی هستند که برای انسان ها منحصر به فرد هستند.دوربین هایی وجود دارند

که به صورت دید در شب، قادر هستند چیزهایی را که ما نمی بینیم، ببینند و پردازش کنند.اسلحه های خودکاری ساخته شده اند که به صورت اتوماتیک و دقیق نشانه گیری می کنند.

پردازش تصویر همینطور با پردازش تصاویر گرفته شده از فاصله های دور هم می تواند در علوم نظامی و امنیتی کمک کند.به عنوان مثال دوربینی قادر است با سرعت بسیار زیاد یک توپ را دنبال کند.این مسئله کاربرد بسیار زیادی در مسائل نظامی دارد.

کاربردهای امنیتی دیگر مانند تشخیص حرکت، تشخیص اثر انگشت، تشخیص چهره و تشخیص دست خط یا امضا و ازکاربردهای نظامی مانند تشخیص و هدف یابی خودکار اهداف متحرک یا ثابت توسط موشک های هوا به زمین میتوان نام برد

۱-۲-۱۰) نجوم و فضا نوردی

ساخت دستگاه های اتوماتیک رصد آسمان و ثبت وقایع آسمانی به صورت خودکار از کاربردهای پردازش تصویر است که امروزه روی آن کار می شود. از پروژه های جدید در بخش نجوم که بخشی از آن توسط سیستم پردازش تصویر انجام می شود، تهیه نقشه سه بعدی از کل عالم کائنات است

پردازش تصویر در فضانوردی هم کاربرد زیادی دارد. در تصاویر دور می توان سطح سیارات و همچنین سطح قمرها را اسکن کرده و اطلاعات بسیار ریزی از آنها استخراج کنیم.

کاربرد دیگر پردازش تصویر در فیلترکردن عکس هایی است که توسط تلسکوپ های فضایی مختلف از جمله هابل از فضا گرفته می شود.

کاربرد دیگر آن حذف گرد و خاک و جو سیاره ها از تصاویر به کمک تصویربرداری IR و X-RAY به صورت همزمان و ترکیب این تصاویر است.

در تصاویر نزدیک هم کاربرد دارد، از جمله هدایت مریخ نوردها، فرود فضاپیماهای بدون سرنشین و الصاق تجهیزات جدید به ایستگاههای فضایی به صورت خودکار.

از امکانات سایت گوگل، امکاناتی است به نام Google Mars که این برنامه دقیقاً مانند Google Earth عمل می کند با این تفاوت که Google Earth سطح زمین را در هر زمان که بخواهید و در هر نقطه ای از زمین و از ارتفاع های بسیار پائین هم نشان می دهد ولی Google Mars دقیقاً همین کار را برای سطح

سیاره مریخ انجام می دهد.

۱-۲-۱۱) پزشکی

یکی از مهمترین کاربردهای پردازش تصویر در علم پزشکی است. در جایی که ما نیاز داریم تمام عکس ها با نهایت شفافیت و وضوح گرفته شوند زیرا دیدن تمام جزئیات لازم است. جراحی های ریز با ایجاد یک سوراخ کوچک و فقط دیدن محل جراحی توسط پزشک، از راه دور و توسط بازوهای رباتیک بسیار دقیق انجام می شوند.

کاربردهای پزشکی دیگر مانند ارتقای ویژگی های تصاویر اشعهX ، تولید تصاویر MRI از مغز و یا تصاویر مربوطه بهCTScan و... است

۱-۲-۱۲) فناوری های علمی

پردازش تصویر در افزایش سرعت پیشرفت های علمی تأثیر فوق العاده داشته است. اولین و مشخص ترین تأثیر آن را می توان در علم عکاسی یا هنر دید. شکار لحظه های شگفت آوری که در کسری از ثانیه اتفاق می افتد، بالا بردن وضوح عکس های گرفته شده و ایجاد افکت های خیره کننده، از دستاوردهای پردازش تصویر است. همچنین در توسعه تکنولوژی پیشرفته gps کمک زیادی داشته و تهیه نقشه های سه بعدی از جاده ها در تمام نقاط جهان، از کاربردهای دیگر آن است. با به وجود آمدن این علم، مسابقات ربات های فوتبالیست به صورت جدی دنبال شد. این علم در پیشرفت علوم پایه فیزیک ، شیمی و مخصوصاً تحقیقات فیزیکی و مکانیکی، کمک فراوانی کرده است. به عنوان مثال وسیله ای برای حمل و نقل کالاها در مسیرهای صعب العبور ساخته شده است. قبل از ساخت آن، رفتار چهارپایان در حالت های مختلف توسط کامپیوتر تحلیل و عیناً به دستگاه آموزش داده شده است.در کل پردازش تصاویر به علت سرعت زیاد آن، در ساخت وسایل مکانیکی پر سرعت، کاربرد زیادی دارد. وسیله ای وجود دارد که قادر است ، توپی که با سرعت بسیار زیاد به سمت پائین می آید را مهار کند.

۱-۲-۱۳) باستان شناسی

در علم باستان شناسی تنها مدارک باقی مانده از دوران باستان، دست نوشته ها، نقاشی ها و غارنگاری های قدیمی است. تهیه تصاویر از بناهای گذشته و بازسازی مجازی این بناهای تاریخی یکی از کاربردهای پردازش تصویر در این علم است. همچنین می توان نقاشی ها و غارنگاری ها را مورد پردازش دقیق قرار داد و شکل آنها را همان طور که در ابتدا بوده اند، شبیه سازی کرد. حتی می توان مکانهای باستانی را از زوایایی که تصاویر مستندی از آنها وجود ندارد، شبیه سازی کرد.

امروزه یکی از پروژه های پر سر و صدای بازسازی بناهای باستانی، بازسازی شهر روم باستان توسط دانشمندان ایتالیایی است. هم اکنون توریست ها با زدن عینک های مخصوص می توانند در خیابان های شهر روم باستان قدم بزنند.

۱-۲-۱۴) تبلیغات

از مقایسه تبلیغات دهه ی ۷۰ و ۸۰ میلادی با تبلیغات امروزی می توان تأثیر تکنولوژی را در تبلیغات کاملاً درک کرد. تغییر شکل تبلیغات از اشکال مربع و زاویه دار به شکل های دایره ای، تغییر رنگ تبلیغات و هزاران تغییر دیگر. یکی از مهمترین فاکتورهای فروش و دلایل بالا رفتن یا پایین آمدن فروش، شکل و نحوه ی بسته بندی کالاست. پردازش تصویر می تواند به ما کمک کند تا قبل از تولید یک بسته بندی آن را شبیه سازی کنیم. با ادغام کردن علم الگوریتم ژنتیک با پردازش تصویر می توان برنامه ای را نوشت که به صورت اتوماتیک به ساختن بسته بندی های مختلف بپردازد و آنهایی که از نظر کاربران زیباتر و جالب تر به نظر خواهند آمد را به ما معرفی نماید.

۱-۲-۱۵) سینما

اولین علمی که پردازش تصویر در آن مورد استفاده قرار گرفت، هنر و سینما بود. یکی از تکنولوژی های برتر دنیا حرکت عکس است که در آن یک کاراکتر انیمیشنی قادر است حرکات دست انسان را تقلید کند. امروزه این سیستم جهت ساخت فیلم ها و بازی های کامپیوتری مورد استفاده قرار می گیرد.

در پردازش تصویر قابلیتی به نام هیستوگرام وجود دارد که با آن قادرند تصاویر را شفاف یا تیره تر کرده و

یا هر تغییر مورد نیاز دیگری را روی تصاویر با توجه به منحنی ها و نمودارهای هیستوگرام بدهند.

در سینما برای اینکه تصویری شفاف به نظر آید، با استفاده از یک کره ی نقره ای رنگ، تصاویر اطراف دوربین را هم ثبت می کنند. بنابراین تصویر نسبت به محیط اطراف خود شفافیت غیر قابل تصوری پیدا می کند.

۱-۲-۱۶) اقتصاد

در دنیای امروز تمام نوآوری ها، به نوعی مستقیم یا غیر مستقیم باعث تغییراتی در اقتصاد گروهی از کشورها و یا کل دنیا می شوند. پردازش تصویر هم، به صورت مستقیم و غیر مستقیم در اقتصاد تأثیر گذار است. در تبلیغات، سیاست، فضانوردی، کشاورزی، شهرسازی، سینما، پزشکی و علوم نظامی می تواند تأثیر غیر مستقیمی در اقتصاد کشورها داشته باشد. همچنین از تأثیر مستقیم آن در اقتصاد، می توان به وجود شعبه های بانک بدون کارمند اشاره کرد. این شعبه ها قادرند به صورت خودکار سریال چک ها و قبوض پرداختی را بخوانند، نوع اسکناس ها را تشخیص دهند و تا حد زیادی از کارهای یک بانک عادی را انجام دهند.

۱-۲-۱۷) روانشناسی

بحث تاثیر رنگ در روحیه انسان اهمیت بسیار زیادی دارد به طوری که در روانشناسی گرایشی به نام روانشناسی رنگ وجود دارد. در این علم در مورد رنگ ها و تأثیر هر یک بر روح و جسم انسان صحبت می شود. به عنوان مثال رنگ قرمز بیشتر تأثیر را در چشم انسان دارد. در حالی که رنگ سبز بیشترین تأثیر را در مغز انسان دارد.

همچنین رنگ آبی باعث ایجاد حس آرامش و اطمینان در انسان می شود. به همین دلیل در سخنرانی های اکثر سیاستمداران دنیا از پرده آبی رنگ در پشت سر آن ها استفاده می شود.

با پردازش تصویر می توان به راحتی تصاویر ثابت و متحرک را ویرایش کرد. به طور مثال رنگ آبی را برای ایجاد حس اطمینان یا رنگ سبز را برای حس زیبایی و قرمز را برای ایجاد هیجان در تصاویر پر رنگ تر

کرد.

۱-۲-۱۸) زمین شناسی

با پردازش تصویر می توان کانی های مختلف را از روی رنگ و اندازه آن ها شناسایی و دسته بندی کرد. همچنین در زمین شناسی برای پی بردن به مواد تشکیل دهنده کانی ها از روش پرتونگاری استفاده می کنند و پردازش تصویر در این بخش می تواند سرعت و دقت این روش را بسیار بالا ببرد. کاربرد دیگر آن این است که دانشمندان با مقایسه کردن ارتفاع آب در سال های مختلف، در واقع روند تند شدن یا کند شدن کاهش آب در سطح زمین را مورد بررسی قرار می دهند.

۱-۳) *عملیات اصلی در پردازش تصویر*

یک تصویر از لحظه ورود به سیستم پردازش تصویر تا تولید تصویر خروجی، به ترتیب مراحل زیر را طی می کند :

(آنجه که در پردازش تصویر اهمیت بسیاری دارد، تسلط کامل بر مفاهیم تکنیک های پردازش تصویر است. به عنوان مثال خواهیم دید که چگونه تنها با چند تکنیک بسیار ساده پردازش تصویر می توانیم یک سیستم دوربین امنیتی ایجاد کنیم.)

۱. تبدیلات هندسی: همانند تغییر اندازه، چرخش و...
۲. رنگ: همانند تغییر روشنایی، وضوح و یا تغییر فضای رنگ
۳. ترکیب تصاویر : ترکیب دو و یا چند تصویر
۴. فشرده سازی تصویر : کاهش حجم تصویر
۵. قطعه بندی تصویر : تجزیه ی تصویر به قطعات با معنی
۶. تفاوت تصاویر : به دست آوردن تفاوت‌های تصویر
۷. میانگین گیری : به دست آوردن تصویر میانگین از دو تصویر

۱-۳-۱) فشرده‌سازی تصاویر

برای ذخیره‌سازی تصاویر باید حجم اطلاعات را تا جایی که ممکن است کاهش داد و اساس تمام روش‌های فشرده‌سازی کنار گذاردن بخش‌هایی از اطلاعات و داده‌ها است.

ضریب یا نسبت فشرده‌سازی است که میزان و در صد کنار گذاشتن اطلاعات را مشخص می‌کند. این روش ذخیره‌سازی و انتقال اطلاعات را آسان‌تر می‌کند و پهنای‌باند و فرکانس مورد نیاز کاهش می‌یابد.

امروزه روش‌هایی متعدد و پیشرفته برای فشرده‌سازی وجود دارد. فشرده‌سازی تصویر از این اصل مهم تبعیت می‌کند که چشم انسان حد فاصل دو عنصر تصویری نزدیک به هم را یکسان دیده و تمایز آنها را نمی‌تواند تشخیص دهد. همچنین اثر نور و تصویر برای مدت زمان معینی در چشم باقی مانده و از بین نمی‌رود که این ویژگی در ساخت تصاویر متحرک مورد توجه بوده‌است.

۱-۳-۱-۱) روش *JPEG*

از این روش در فشرده‌سازی عکس و تصاویر گرافیکی ساکن استفاده می‌شود JPEG اولین و ساده‌ترین روش در فشرده‌سازی تصویر است به همین دلیل در ابتدا سعی شد برای فشرده‌سازی تصاویر متحرک مورد استفاده قرار گیرد. برای این منظور تصاویر به صورت فریم به فریم مانند عکس فشرده می‌شدند وبا ابداع روشJPEG حرکتی برای ارتباط دادن این عکس‌ها به هم تلاش شد که با مشکلاتی همراه بود.

۱-۳-۱-۲) روش *MPEG*

این روش در ابتدای سال ۱۹۹۰ابداع شد و در آن اطلاعات تصویر با سرعت حدود ۱/۵ مگابیت بر ثانیه انتقال پیدا می‌کرد که در تهیه تصاویر ویدئویی استفاده می‌شد. با این روش امکان ذخیره حدود ۶۵۰ مگابایت اطلاعات معادل حدود ۷۰ دقیقه تصویر متحرک در یک دیسک به وجود آمد. در MPEG بیت‌های اطلاعات به صورت سریال ارسال می‌شوند و به همراه آنها بیت‌های کنترل و هماهنگ‌کننده نیز ارسال می‌شوند که موقعیت و نحوه قرارگیری بیت‌های اطلاعاتی را برای انتقال و ثبت اطلاعات صدا و تصویر تعیین می‌کند.

۱-۳-۱-۳) روش *MP3*

MP3نیز روشی برای فشرده سازی اطلاعات صوتی به ویژه موسیقی است که از طریق آن حجم زیادی از اطلاعات صوتی در فضای نسبتاً کوچکی ذخیره می‌شود.

۱-۳-۱-۴) روش *MPEG2*

در روش MPEG2از ضریب فشرده‌سازی بالاتری استفاده می‌شود و امکان دسترسی به اطلاعات ۳ تا ۱۵ مگابیت بر ثانیه‌است از این روش در دی‌وی‌دی‌های امروزی استفاده می‌شود در اینجا نیز هر فریم تصویری شامل چندین سطر از اطلاعات دیجیتالی است.

۱-۳-۱-۵) روش *MPEG4*

از این روش برای تجهیزاتی که با انتقال سریع یا کند اطلاعات سرو کار دارند استفاده می‌شود. این روش توانایی جبران خطا و ارائه تصویر با کیفیت بالا را دارد. مسئله خطا و جبران آن در مورد تلفن‌های همراه و کامپیوترهای خانگی و لپ‌تاپ‌ها و شبکه‌ها از اهمیت زیادی برخوردار است. در شبکه‌های کامپیوتری باید تصویر برای کاربرانی که از مودم‌های سریع یا کند استفاده می‌کنند به خوبی نمایش داده شود، در چنین حالتی روشMPEG4 مناسب است. از این روش در دوربین‌های تلویزیونی نیز استفاده می‌شود. ایده اصلی این روش تقسیم یک فریم ویدئویی به یک یا چند موضوع است که مطابق قاعده خاصی کنار هم قرار می‌گیرند مانند درختی که از روی برگ‌های آن بتوان به شاخه تنه یا ریشه آن دست یافت. هر برگ می‌تواند شامل یک موضوع صوتی یا تصویری باشد. هر کدام از این اجزا به صورت مجزا و جداگانه قابل کپی و یا انتقال هستند. این تکنیک را با آموزش زبان می‌توان مقایسه کرد.

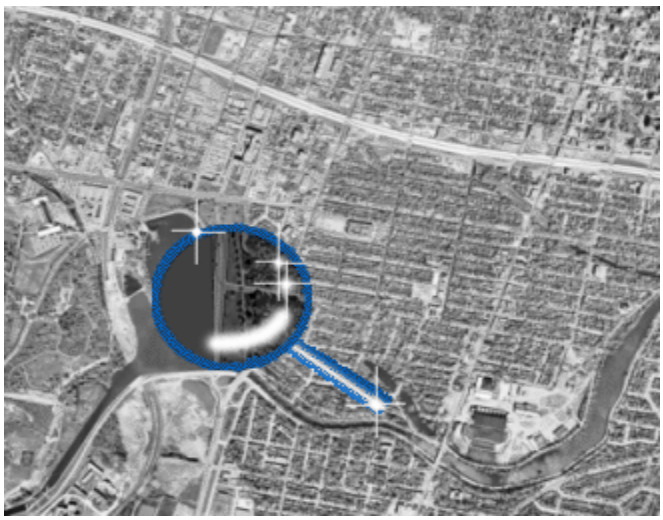
همان‌طوری‌که در آموزش زبان کلمات به صورت مجزا و جداگانه قرار داده می‌شوند و ما با مرتب کردن آن جملات خاصی می‌سازیم و می‌توانیم در چند جمله، کلمات مشترک را فقط یک‌بار بنویسیم و هنگام مرتب کردن آن‌ها به کلمات مشترک رجوع کنیم، در اینجا هم هر یک از این اجزا یک موضوع خاص را مشخص می‌کند و ما می‌توانیم اجزا مشترک را فقط یک‌بار به کار ببریم و هنگام ساختن موضوع به آنها رجوع کنیم. هر یک از موضوعات هم می‌توانند با موضوعات دیگر ترکیب و مجموعه جدیدی را بوجود آورند. این

مسئله باعث انعطاف‌پذیری و کاربرد فراوان روشMPEG4 می‌شود. برای مثال به صحنه بازی تنیس توجه کنید. در یک بازی تنیس می‌توان صحنه را به دو موضوع بازیکن و زمین بازی تقسیم کرد زمین بازی همواره ثابت است بنا بر این بعنوان یک موضوع ثابت همواره تکرار می‌شود ولی بازیکن همواره در حال حرکت است و چندین موضوع مختلف خواهد بود. این مسئله سبب کاهش پهنای باند اشغالی توسط تصاویر دیجیتالی می‌شود. توجه داشته باشید که علاوه بر سیگنال‌های مربوط به این موضوعات سیگنال‌های هماهنگ کننده‌ای هم وجود دارند که نحوه ترکیب و قرارگیری صحیح موضوعات را مشخص می‌کند.

۱-۳-۲) انواع تصاویر

۱-۳-۲-۱) *تصاویر آنالوگ*

تصاویری مانند عکس های هوایی که توسط سیستم های عکس برداری (دوربین) به دست میآیند.از آنجایی که در این عکس ها از فیلم عکاسی استفاده شده است،پس هیچ پردازشی نیاز ندارد.

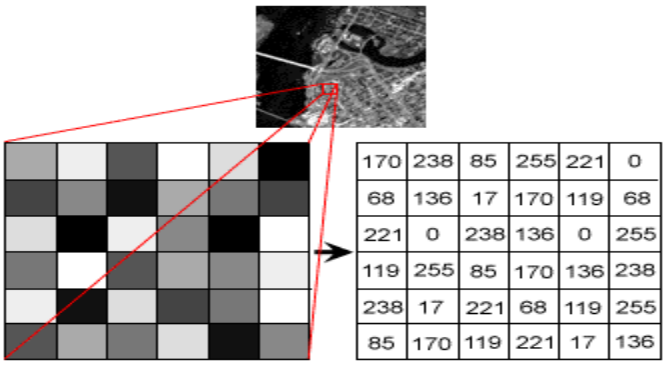


تصویر آنالوگ (عکس هوایی که نیاز به اصلاح و پردازش ندارد)

۱-۳-۲-۲) *تصاویر دیجیتالی*

تصاویر سنجش شده که از تعداد زیادی مربعات کوچک(پیکسل) تشکیل شده اند. هر پیکسل دارای یک شماره رقمی میباشد که بیانگر میزان روشنایی آن پیکسل است.

به این نوع تصاویر ، تصاویر رستری هم میگویند.تصاویر رستری دارای سطر و ستون میاشند.



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

تصویر بالا(دیجیتالی) .پایین و سمت چپ(پیکسلها).سمت راست و پایین(شماره های هر پیکسلDN ِ)

۱-۳-۲-۳) *مقادیر پیکسلها*

مقدار انرژی مغناطیسی که یک تصویر دیجیتالی به هنگام تصویر برداری کسب میکند،رقم های دوتایی یا بیت هارا تشکیل میدهند که از قوه صفر تا ۲ ارزش گذاری شده است.هر بیت ، توان یک به قوه ۲ (۱بیت=۲۱)میباشد. حداکثر تعداد روشنایی بستگی به تعداد بیت ها دارد. بنابراین ۸ بیت یعنی ۲۵۶ شماره دیجیتالی که دامنه ای از ۰ تا ۲۵۵ دارد.به همین دلیل است که وقتی شما تصویر رستری از سنجنده خاصی مانند TM را وارد نرم افزاری میکنید تغییرات میزان روشنایی را بین ۰ تا ۲۵۵ نشان میدهد.

۱-۳-۲-۴) *دقت تصویر*

دقت تصویر بستگی به عدد پیکسل ها دارد.با یک تصویر ۲ بیتی ، حداکثردامنه روشنایی ۲۲ یعنی ۴ میباشد که دامنه آن از ۰ تا ۳ تغییر میکند.در این حالت تصویر دقت (تفکیک پذیری لازم) را ندارد.تصویر

۸ بیتی حداکثر دامنه ۲۵۶ دارد و تغییرات آن بین ۰ تا ۲۵۵ است .که دقت بالاتری دارد.

دقت تصویر ۳ بیتی دقت تصویر ۸ بیتی

۱-۳-۲-۵) ***ترمیم تصویر***

در بیشتر تصاویری که توسط ماهواره ها یا رادار ها ثبت میگردند ، اختلالاتی در تصویر به وجود میاید که به دلیل خش میباشد.

دو اختلال مهم در تصاویر چند باندی ، نواری شدن و خطوط از جا افتاده میباشد.

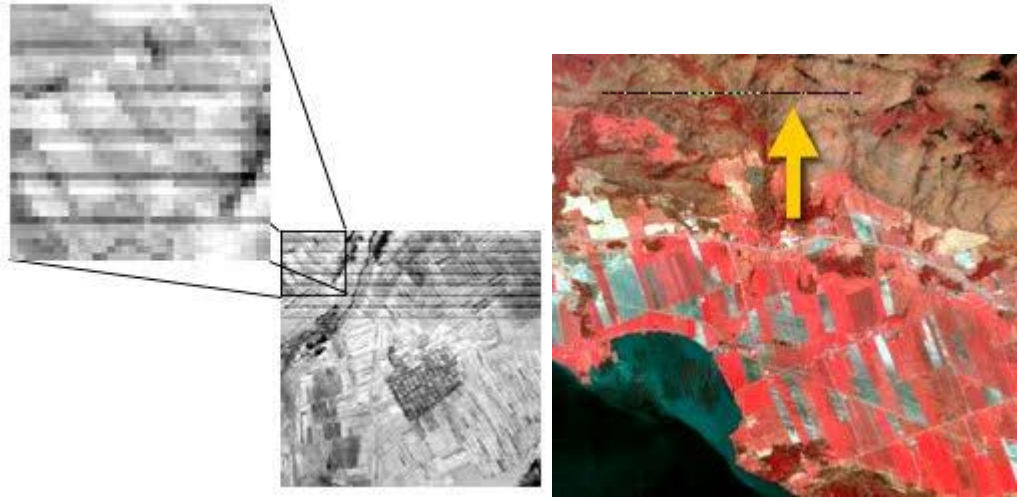
نواری شدن(باندی شدن)

اشتباهی که توسط سنجنده ، در ثبت و انتقال داده ها روی میدهد.و یا تغییر پیکسل در بین ردیف ها میتواند باعث ایجاد چنین اشتباهی گردد.

خطوط از جا افتاده (خطا در تصویر)

اشتباهی که در ثبت و انتقال داده ها روی میدهد و در نتیجه، یک ردیف پیکسل در عکس از بین میرود.

باندی شدن یک ردیف پیکسل در تصویر است



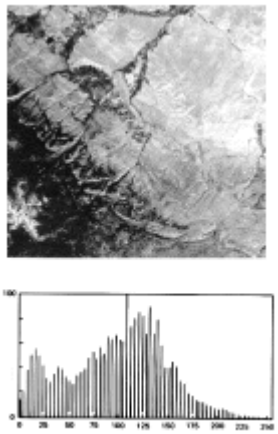
باندی شدن یک ردیف پیکسل در تصویر

۱-۳-۳) بالا بردن دقت عکس

یکی از کار های مهمی که در پردازش تصویر انجام میگردد، بالا بردن دقت عکس به منظور دید و تفسیر چشمی دقیق تر میباشد.روش های بسیاری برای نیل به این هدف وجود دارد ولی مهمترین آنها ، افزایش تباین تصویر و عملیات فیلتر کردن میباشد.

۱-۳-۳-۱) *هیستوگرام تصویر*

در هر تصویر دیجیتالی ، مقادیر پیکسل ها بیانگر خصوصیات آن تصویر(مانند میزان روشنایی تصویر و وضوح آن) میباشد.هیستوگرام تصویر در حقیقت بیان گرافیکی میزان روشنایی تصویر میباشد. مقادیر روشنایی(برای مثال ۰-۲۵۵) در طول محور X بیان شده و میزان فراوانی هر مقدار در محور Y بیان میگردد.



تصویر ۸ بیتی(۰-۲۵۵) در بالا و هیستوگرام مقادیر پیکسل تصویر در پایین .

محور افقی بین ۰-۲۵۵ و محور قائم ، تعداد پیکسل ها میباشد.

۱-۴) *تشخیص خودکار شماره پلاک خودرو*

۱-۴-۱) دوربین سامانه ی تشخیص پلاک خودرو

شماره پلاک خودرو یکی از مناسب‌ترین اقلام اطلاعاتی جهت احراز هویت خودروها می‌باشد. سامانه تشخیص پلاک خودرو یک سیستم کاملاً مکانیزه است که با استفاده از پردازش تصویر خودروهای عبوری از یک مکان، شماره پلاک آنها را استخراج می‌کند. برای استفاده از این سامانه، نیازی به نصب و تجهیز خودروها به وسیلهی دیگری (مانند GPS یا برچسب‌های رادیویی) وجود ندارد. این سامانه با استفاده از دوربین‌های مخصوص، تصویری از خودرو در حال عبور اخذ می‌کند و آن تصویر را جهت پردازش توسط نرم‌افزار تشخیص پلاک خودرو به رایانه ارسال می‌کند. از این سامانه می‌توان در زمینه‌های امنیتی و ترافیکی بسیار بهره گرفت.

۱-۴-۲) کاربردهای سامانه ی تشخیص پلاک

۱-۴-۲-۱) *کنترل و اخذ عوارض ورود به محدوده طرح ترافیک*

امروزه شهرهای بسیاری (از جمله تهران) ورود خودروها به منطقه مرکزی شهر را به منظور کنترل ترافیکِ آن محدود ساخته‌اند. از آنجا که استفاده از روش‌های سنتی (قرار دادن نیروهای پلیس در تمامی مبادی محدوده) هم پر هزینه و هم کم دقت است، راه حل‌های جدیدی برای کنترل و اخذ عوارض ورود به محدوده پرتردد شهرها پیشنهاد شده است. یکی از این راه حل‌ها (که برای مثال در استکهلم و لندن استفاده می‌شود) استفاده از فناوری تشخیص پلاک خودرو است. در این راه حل، دوربین‌های تشخیص پلاک خودرو در تمامی مبادی طرح نصب می‌شوند و ورود هر خودرو به محدوده طرح ثبت می‌شود. سپس مانند روش اخذ عوارض، فرصتی به راننده داده می‌شود تا عوارض ورود به طرح را تا زمان مقرر پرداخت کند. در غیر اینصورت، راننده طبق قانون ملزم به پرداخت جریمه خواهد بود.

۱-۴-۲-۲) *اخذ عوارض جاده‌ها و بزرگراه‌ها به صورت خودکار*

از آنجا که وجود مانع بر سر راه خودروها در عوارضی‌ها باعث کند شدن حرکت، ایجاد ترافیک، و به تبع آن آلودگی محیط زیست می‌شود، راه‌های مختلفی برای حذف موانعِ موجود در عوارضی‌ها پیشنهاد شده است. یکی از این راه‌ها استفاده از سامانه ی تشخیص پلاک خودرو می‌باشد. در این راه حل، خودروها بدون نیاز به توقف از عوارضی‌ها عبور می‌کنند و سامانه ی تشخیص پلاک خودرو شماره پلاک آنها را ثبت می‌کند. بر اساس شماره پلاک، عوارض مربوطه محاسبه می‌شود و راننده ملزم به پرداخت عوارض در زمان مشخصی خواهد بود. در صورت عدم پرداخت عوارض در زمان مقرر، خودرو طبق قانون جریمه خواهد شد.

۱-۴-۲-۳) *محاسبه مدت سفر*

تخمین مدتِ زمان سفر یکی از کاربردهای مهم سیستم‌های ترافیک هوشمند می‌باشد. در این کاربرد، مسافران می‌توانند پیش از سفر به آمارها و اطلاعات مربوطه مراجعه کنند و تخمینی از مدتِ زمان سفر میان مبدا و مقصد خود داشته باشند. سامانه تشخیص پلاک خودرو یکی از راه حل‌های مناسب جهت این کاربرد به شمار می‌رود. در این راه حل، سامانه ی تشخیص پلاک خودرو در نقاط مختلف یک جاده نصب

می‌شود (برای مثال در مبدا و مقصد) و بنابراین مدت زمان سفر را به صورت تفکیک شده برای هر خودرو محاسبه می‌کند. با تحلیل آماری این مدت برای تمامی خودروها می‌توان با دقتِ مطلوبی، متوسط و تغییراتِ آن در زمان‌های مختلف روز و هفته را در جاده اندازه گرفت و برای تصمیم‌گیری در اختیار عموم قرار داد.

۱-۴-۲-۴) *اندازه‌گیری سرعت متوسط خودروها*

علاوه بر روش‌های معمولِ اندازه‌گیری سرعت که در یک نقطهی خاص سرعت خودروها را محاسبه می‌کنند، روش‌هایی نیز جهت محاسبه سرعت متوسط خودروها در یک مسیر وجود دارد. جهت اندازه‌گیری سرعتِ متوسط نیاز به تشخیص هویت خودروها در ابتدا و انتهای مسیر می‌باشد. تشخیص پلاک خودرو یکی از راه‌های مناسب جهت تشخیص هویت خودروها و به تبعِ آن اندازه‌گیری سرعت متوسط آنها می‌باشد. در این راه حل، دوربین‌های تشخیص پلاک در چندین نقطه از مسیر نصب می‌شوند و با ثبتِ زمان تردد خودرو از مقابل هر یک از آنها، امکان محاسبه سرعت متوسط خودرو میان هر دو نقطه متوالی وجود دارد. در این راه حل، حتی اگر رانندگان در مقابل این دوربین‌ها ترمز کنند تأثیر چندانی در سرعت متوسط محاسبه شده در مسیر نخواهند گذاشت و بنابراین تا حدی در مقایسه با روش‌های مبتنی بر سرعتِ نقطه‌ای برتری دارد.

۱. دیده بانی معابر، گلوگاه‌ها و مرزها و گزارش سریع خودروهای سرقتی عبور کرده از آنها
۲. ثبت اطلاعات ترافیکی دقیق و جامع از تردد خودروها در معاب

فصل دوم

دستورات نرم افزارمتلب

۲-۱) *مفاهیم رنگ وتصویر*

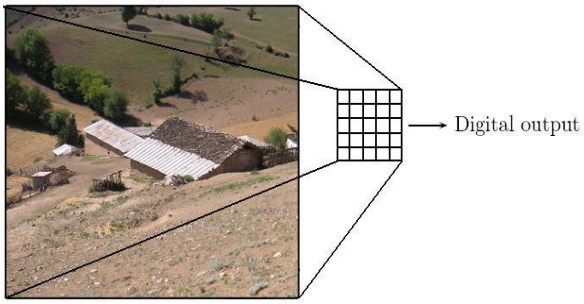
در علم پردازش تصویر،رنگ وتصویر اساس کار با سیستم ها و ماشین های بینایی را تشکیل می دهند. هر سیستم کنترلی برای خود ورودی خاصی دارد که ان را توسط الگوریتم های خود و نسبت به نیاز کاربر،کنترل می کند. در سیستم های پردازش تصویر ورودی رنگ وتصویر است که بعد از تبدیل انها به سیگنال های دیجیتال با الگوریتم ها و فیلتر های متنوع انها را قابل کنترل می نماید. صنعت پردازش با انالیز تصویر در حال توسعه بوده و در اینده اساس کار روبات ها را تشکیل خواهند داد.ناگفته نماند که این علم در تمامی زمینه ها از جمله پزشکی و روباتیک ویا تشخیص جرم با اثر انگشت و در زمینه های فوق پیشرفته هوا و فضا و کاربردهای گسترده نظامی و هدف یابی های خود کار موشک ها و دهها کاربرد دیگر وارد شده است و نمی توان ان را نادیده گرفت

.

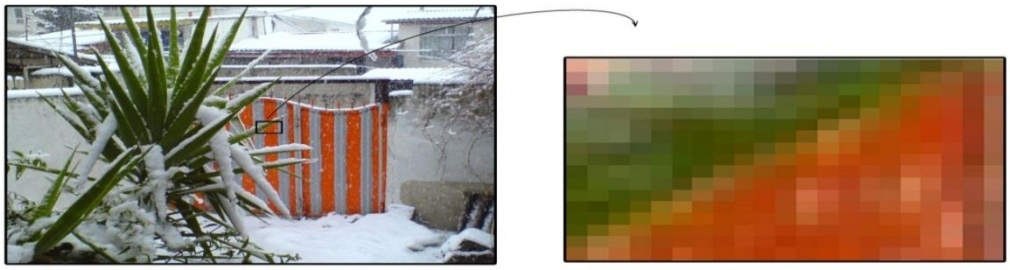
۲-۱-۱) پیکسل

تصاویری که یک دوربین دیجیتال دریافت می کند به صورت شکل زیر می باشد.

این تصاویر از نقاط مربع شکل بسیار کوچکی تشکیل شده که هر کدام از این نقاط دارای رنگی خاص



است که با قرار گیری در کنار یکدیگر،تشکیل یک تصویر را می دهند.اگر به تصویر زیر نگاه کنید،در صورتی که تصویر را چندین برابر بزرگ کرده و به گوشه ای از ان توجه نمایید این موضوع را خواهید دید



این نقاط رنگی پیکسل نامیده می شود که کوچکترین عنصر تشکیل دهنده یک تصویر دیجیتال است.

۲-۱-۲) ابعاد و اندازه تصاویر:

یک دوربین هر یک از این پیکسل ها را توسط حسگرهای خود دریافت کرده و به اعداد دیجیتال تبدیل می کندوتعداداین پیکسل ها در یک تصویر اندازه تصویر را مشخص می کند و برای اندازه یک تصویر از

تعداد پیکسل های موجود در تصویر،به صورت ضرب سطر در ستون یا طول در عرض استفاده می شود.برای مثال اگر یک تصویر دارای ۸۰۰ پیکسل در طول و ۶۰۰ پیکسل در عرض باشد این به معنی وجود ۴۸۰۰۰۰عدد پیکسل(۶۰۰*۸۰۰) در ان تصویر می باشد،و می توان با استفاده از همین پیکسل ها و تبدیل انها به یک ماتریس،تصاویر را پردازش کرد.در این روش هر ارایه ماتریس برابر با یک پیکسل در تصویر مورد نظر خواهد بود.هر چه تعداد این پیکسل در یک تصویر بیشتر باشد،تصویر بزرگتر و همین طور دارای کیفیت بالاتری خواهد بود.البته این عامل تنها دلیل بالایی کیفیت در یک تصویر نیست و عامل اصلی مقدار بیت موجود در هر پیکسل است.

۲-۱-۳) تعداد بیت های یک تصویر:

هر پیکسل باید به صورت یک عدد بیان شود تا قابل پردازش باشد.در صورتی که تنها از اعداد ۰ و۱ برای نمایش پیکسل ها بیان کنیم،تصویر ما تنها دارای دو رنگ سیاه و سفید خواهد بود که بدون هیچ سایه و تغییر روشنایی خواهد بود.به این نوع تصویر باینری گویند که در ادامه به ان اشاره خواهد شد.حال در صورتی که هر پیکسل دارای تعداد بیشتری بیت باشد،می توان تصاویر را با کیفیت و سایه ها و روشنی های بیشتری دید.درتصاویر زیر تفاوت تعداد پیکسل با رزولوشن و تعداد بیت های یک تصویر را ملاحظه می کنید.

تصویر دو رنگ (0 – 1)

تصویر اصلی تعداد بیت بالا

تصویر دو رنگ با کاهش پیکسل

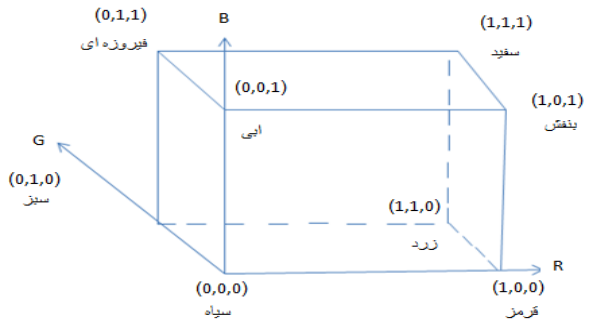
تصویر با کاهش پیکسل

۲-۱-۴) روش شناسایی رنگ در یک پیکسل

برای شناسایی رنگ یک پیکسل به نرم افزار روش ها و مدل های مختلفی وجود دارد.اما برای نرم افزار بهترین نوع مدل رنگ، مدل سه رنگ می باشد که به دلیل سادگی بیشتر استفاده شده است .

۲-۱-۴-۱) *مدل سه رنگ*

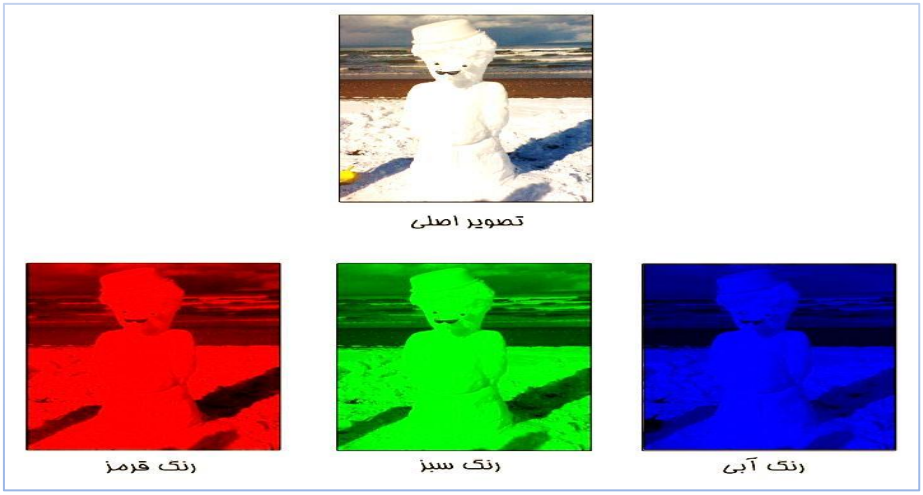
سه رنگ که کوتاه شده ی کلمات ابی و سبز و قرمز است و این سه رنگ اصلی برای تشکیل رنگ های موجود در دنیا کافی می باشد.در دوربین ها و تلوزیون ها هم کاملا از این نوع روش یعنی از سه رنگ اصلی برای تولید رنگ ها استفاده می شود.برای نمایش یک سیستم سه رنگ از مکعب رنگ استفاده می شود.مکعب رنگی را در شکل زیر مشاهده می کنید.



برای هر یک از رنگ های ابی و سبز و قرمز ،عددی ۸ یا ۱۶یا ۳۲بیتی استفاده می شود.برای مثال در صورت استفاده از ۸بیت(برابر با عدد ۲به توان ۸)برای هر یک از رنگ های سه گانه،سیستم می تواند تا بیش از تعداد۱۶ میلیون رنگ را تشخیص دهد(۱۶۵۸۱۳۷۵=۲۵۵*۲۵۵*۲۵۵).حال ببینید با ۳۲ بیت برای هر یک از رنگ های سه گانه،چه تعداد رنگ قابل بررسی می شود.در مطلب مقدار بیت هر پیکسل را با نام و کلاس uint8نمایش می دهند و به همین صورت تصاویر ۱۶و۳۲ را با کلاس uint16-doubleبه نمایش در می اید. در نرم افزار هر یک از این سه عدد ۸تا۳۲ بیتی به یک پیکسل اختصاص دارد که تشکیل رنگ یک تصویر را می دهد.این سه رنگ اصلی به نسبت های مشخص در صورتی که با همدیگر ترکیب شوند،رنگ سفید را تشکیل می دهند.به تصویر ادم برفی زیر نگاه کنید.

این تصویر در حالت معمولی سفید است حال هر یک از رنگ های قرمز و سبز و ابی را از این تصویر جدا می کنیم.دلیل استفاده از این تصویر تنها وجود رنگ سفید در تمامی تصویر می باشد و به شما نشان می دهد که چگونه با ترکیب سه رنگ اصلی می توان رنگ سفید را تولید نمود.

مشاهده می کنید که هر یک از رنگ های جدا شده دارای سایه و روشنی های مخصوص به خود را دارند که با ترکیب با همدیگر،تشکیل یک تصویر واحد را می دهند.



۲-۱-۵) شناسایی تصویر برای متلب

برای انالیز و پردازش یک تصویر و شناسایی ان برای نرم افزار متلب باید تصویر را قابل فهم برای سیستم تبدیل نمود.تنها زبان قابل فهم یک سیستم کامپیوتری اعداد می باشد.به همین دلیل تصاویر به صورت ماتریسی از اعداد تبدیل می شود که نرم افزار تنها با این اعداد در ارتباط است.البته برای تصاویر غیر رنگی تنها به یک ماتریس نیاز است،اما برای تصاویر رنگی به سه ماتریس جدا برای هر یک از سه رنگ اصلی سه رنگ نیازمی باشد و نرم افزار باید هر ماتریس را جداگانه مورد برسی قرار دهد.نرم افزار مطلب چهار نمونه از این تصاویر را تولید و بررسی می کند که عبارتند از:

۱- تصاویر باینری
۲- تصاویر رنگی با شاخص
۳-تصاویر غیر رنگی با شدت نور
۴- تصاویر رنگی سه رنگ

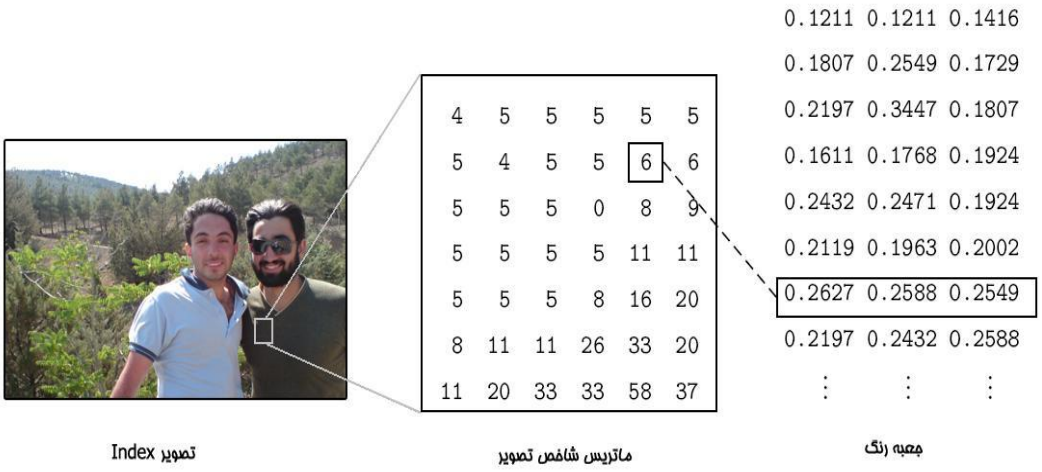
۲-۱-۵-۱) *تصاویر باینری*

تصاویری که تنها دارای دو رنگ سیاه و سفید می باشند و در ماتریس انها تنها اعداد ۰و۱ قرار گرفته است.این نوع تصاویر کاربرد فراوانی را در زمینه پردازش تصویر دارند و اکثر برنامه ها تصاویر خود را در نهایت به این نوع تصاویر تبدیل کرده و بر روی ان پردازش را انجام می دهند. ساده ترین پردازش در مورد این نوع تصاویرصورت می گیرد و این تنها به دلیل تک بیت بودن پیکسل های این نوع تصاویر

است.(اعداد ۰ به معنی رنگ سیاه و اعداد ۱ به معنی رنگ سفید می باشد).

۲-۱-۵-۲) *تصاویر رنگی با شاخص*

تصاویر رنگی که بعد از ورود به محیط مطلب ،تبدیل به یک ماتریس به اندازه ابعاد تصویر شده که ارایه های ان دارای اعدادی می باشد. در این نوع تصاویر،خود عدد ارایه و ماتریس مشخص کننده رنگ پیکسل نیست بلکه عدد ارایه،شاخص یا شماره عددی از یک جعبه رنگ می باشد.این جعبه رنگ به صورت ۳ ماتریس برای سبز و ابی و قرمز تعبیه شده است و در حافظه نرم افزار و با فراخوانی تصویر و یا جعبه های رنگ اماده همراه می باشد که توسط اعداد درون ماتریس تصویر شاخص فراخوانی شده و به نمایش در می ایند.در شکل زیر نمونه ای از این نوع تصاویر را ملاحظه می کنید.



۲-۱-۵-۳) *تصاویر غیر رنگی با شدت نور*

این نوع تصاویر غیر رنگی با وضوح روشنی و تیرگی های تصویر به صورت تک ماتریس در ابعاد تصویر می باشد.به این نوع تصاویر،تصاویر خاکستری گفته می شود.ارایه های ماتریس این تصویر تنها نشان دهنده میزان روشنایی و یا تیرگی و سایه های تصویر هستند. در قدیم ارایه های این نوع تصاویر عددی بین ۰ تا ۱ و تا سه رقم اعشار بود،اما در حال حاضر این نوع تصاویر به صورت بیشتر تا ۸بیتی و شبیه به اعداد ماتریس سه رنگ می باشد.در حقیقت این نوع تصویر تنها تفاوتی که با تصاویر آر جی بی دارند این است که نبود رنگ در تصاویر می باشد.در شکل زیر نمونه ای از این نوع تصاویر را ملاحظه می کنید.

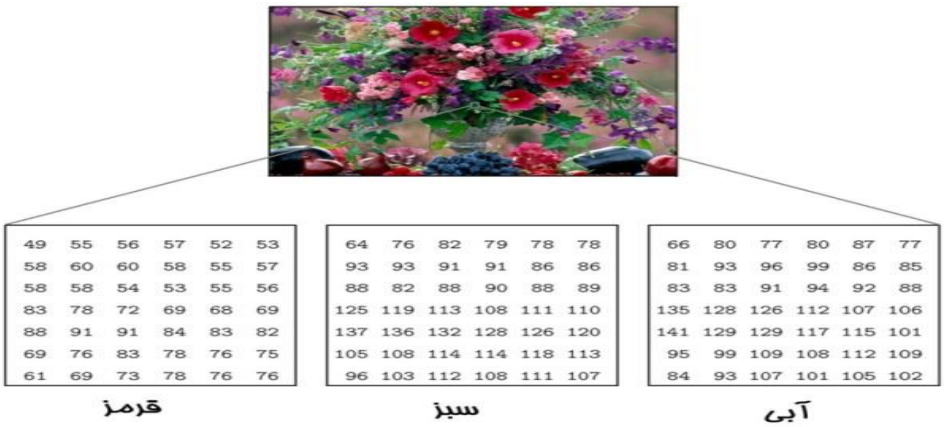
۲-۱-۵-۴) *تصاویر سه رنگ*

این نوع تصاویر رنگی که تشکیل شده از ۳ ماتریس جداگانه برای رنگ های سبز و ابی و قرمز،که هر درایه از سه ماتریس بعد از ترکیب شده ،تبدیل به رنگ کامل یک پیکسل از تصویر می شود.در تصویر زیر یک نمونه از این تصاویر را به همراه ماتریس های ان مشاهده می نمایید.

۲-۲) *پردازش اولیه تصویر*

۲-۲-۱) ورود تصاویر به نرم افزار مطلب

همانطور که قبلا گفته شد نرم افزار مطلب تمامی تصاویر را به صورت ماتریس شناسایی میکند .اولین دستور مورد نیاز در این زمینه،دستور خواندن تصویر توسط نرم افزار است



A=imread(‘ نام فایل و مسیر جاری تصویر ’)

توسط این دستور تصویر مورد نظر شما که در مسیر جاری ذخیره شده است را فراخوانی کرده و در متغییر A و به صورت یک ماتریس ذخیره می کند.بهتر است که در پایان دستور از علامت (;) استفاده شود،چون نرم افزار می خواهد بعد از زدن کلید ورود مقدار قرار گرفته در متغییر را نمایش دهد و این متغییر ارایه های بسیار زیادی دارد و زمان را از دست خواهید داد و میزان درگیری CPU را افزایش می دهید.به مثال زیر توجه نمایید.

```
>> P=imread('D:\pictures\CHAMAHCO\DSC01271.jpg');
```

نکته:در صورتی که تصویر را در جایی که برنامه ذخیره شده است قرار دهید،دیگر نیازی به نوشتن مسیر جاری نخواهد بود و نام و پسوند ان تصویر کفایت می کند.

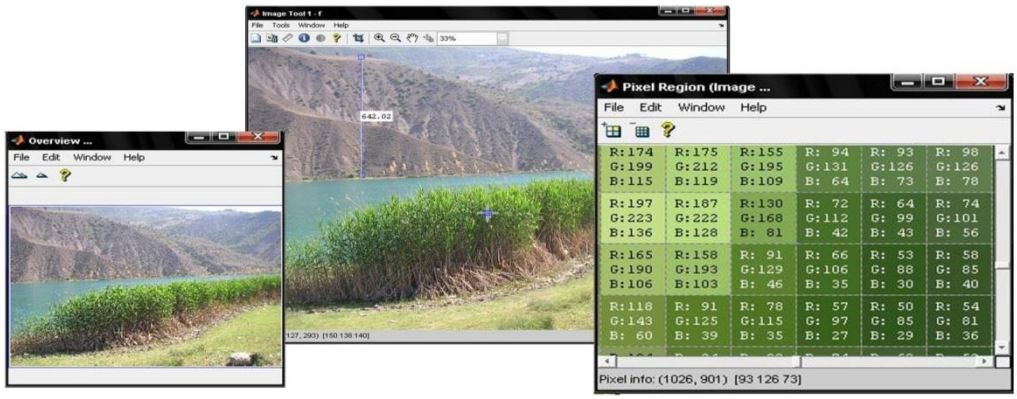
۲-۲-۲) دریافت اطلاعاتی کامل از یک تصویر

imfinfo('نام فایل و مسیر جاری تصویر')

اطلاعات کاملی را از یک تصویری که در مسیر جاری موجود می باشد را در اختیار کاربر قرار می دهد.

۲-۲-۳) نمایش تصویر فراخوانی شده با ابزار ویرایش:

imshow(pic)



بعد از دستور فراخوانی تصویر شما برای دیدن تصویر در محیط نرم افزار از این دستور استفاده نمایید.البته ابزاری برای ترسیم و نوشتن متون در این پنجره وجود دارد.این دستور متغییر pic که در ان تصویر ذخیره شده است را نمایش می دهد.

نکته: فراموش نکنید که نرم افزار متلب به حروف بزرگ و کوچک حساس است.

۲-۲-۴) نمایش تصویر فرخوانی شده با ابزار رنگ پیکسل:

imtool(pic)

این دستور هم مانند دستور قبل تصویر مورد نظر شما را نمایش می دهد،اما به همراه نمایش تصویر دو پنجره دیگر هم باز می شود که امکاناتی نظیر دیدن رنگ های سه رنگ هر پیکسل را به شما نشان خواهد داد.در پنجره دیگر یکی از امکانات بسیار مناسب،استفاده از خط کش برای اندازه گیری می باشد.این دستور متغییر pic که در ان تصویر ذخیره شده است را نمایش می دهد.

مثال

۲-۲-۵) نمایش هر نوع اطلاعاتی در پنجره های نمایشی:

(title(type

اگر نیاز داشته باشید که در هر پنجره نمایشی مانند نمایش تصویر و یا نمایش بردارها و در کل پنجره های شکل،متنی یا عددی و یا مقدار متغییری را قرار دهید،از این تابع استفاده کنید.

برای نمایش متون کافی است متن مورد نظر خود را در علامت ' 'قرار دهید.

برای نمایش اعداد باید از دستور num2str استفاده نمایید تا مقادیر به رشته تبدیل شود.

نکته:اگر از چند مقدار و متن برای نمایش می خواهید استفاده نمایید هر کدام را با علامت کاما جدا کرده و همگی را در داخل دراکت[] قرار دهید.

مثال:

```
>> p=imread('D:\pictures\CHAMAHCO\DSC00994.jpg');
>> x=23;
>> imshow(p);title(['image processing',num2str(x)])
```

۲-۲-۶) انواع تصاویر:

قبلا در مورد انواع تصاویر در محیط نرم افزار مطلب توضیح کامل داده شد و انواع تصاویر باینری شاخص وخاکستری وسه رنگ مورد بررسی قرار گرفت.در اینجا به بررسی توابع این نوع تصاویر و نحوه تبدیل انها به یکدیگر ،می پردازیم.به خاطر داشته باشید که هر تصویری را که نرم افزار مطلب فراخوانی می کنید،به صورت یک تصویر سه رنگ ذخیره می شود.برای تبدیل این تصاویر به انواع دیگر،از توابعی استفاده می شود که در ادامه به بررسی انها می پردازیم.بیشتر توابع مربوط به پردازش تصویر تنها بک نوع از انواع تصاویر را پشتیبانی می کنند.به همین دلیل نیازی شدید به تبدیل این نوع تصاویر را دارید.

۲-۲-۶-۱) *تبدیل تصویر به باینری:*

(im2bw(pic,n

توسط این دستور هر نوع تصویری (شاخص وشدت نور وسه رنگ) را می توان به تصویر از نوع باینری تبدیل نمود.به مثال زیر توجه کنید.در این مثال تصویر فراخوانی شده که همانطور که گفته شده،سه رنگ می باشد،به تصوی باینری تبدیل می شود.n عددی بین ۰ تا ۱ که تعیین کننده میزان مرز بین سیاه و سفید می باشد.اگر از n استفاده نکنید،به صورت پیش فرض مقدار 0.5را دارد.

```
>> A=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> imshow(A);
>> x=im2bw(A,0.6);
>> imshow(x);
```

تصویر باینری

تصویر اصلی

پیدا کردن مقدار استانه رنگ (مقدارn) در تبدیل تصویر به باینری:

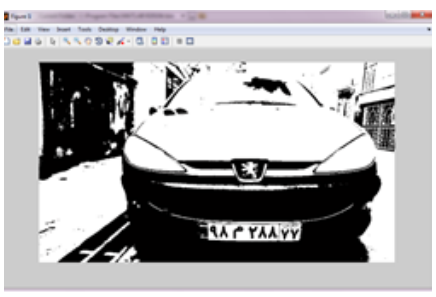
همیشه نمی توان یک تصویر را به خوبی به یک تصویر باینری تبدیل نمود.برای اینکه با تغییر مقدار n در تابع im2bw کیفیت تصویر خروجی تغییر اساسی می کند.برای اینکه بهترین تصویر را از این تابع بدست بیاوریم،دستوری است که مقدارn را نسبت به تصویر اصلی ،به صورت اتوماتیک بدست می اورد و دیگر نیازی به تغییر مقدار n نداریم.

تعیین مقدار استانه:

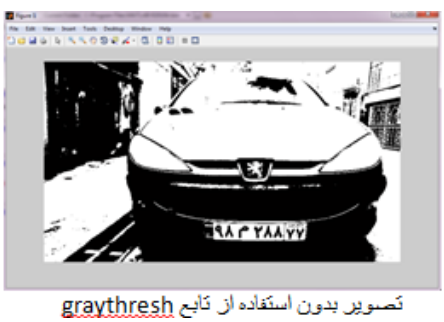
(graythresh(pic

این دستور به صورت زیر استفاده شده و در تابع im2bw قرار داده می شود. در مثال زیر تفاوت استفاده کردن و یا نکردن از این تابع مورد بررسی قرار گرفته است.تفاوت کاملا قابل احساس است.

```
>> A=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> imshow(A);
>> x=im2bw(A);
>> imshow(x);
>> x3=im2bw(A,graythresh(A));
>> imshow(x3);
```



تصویر باینری با استفاده از تابع graythresh



تصویر بدون استفاده از تابع graythresh

تصویر اصلی

۲-۲-۶) *تبدیل تصویر سه رنگ به شدت نور:*

rgb2gray

توسط این دستور تصویر سه رنگ را می توان به تصویر از نوع شدت نور تبدیل کرد.به مثال زیر توجه کنید:

```
>> A=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> imshow(A);
>> x=rgb2gray(A);
>> imshow(x);
```

تصویر غیر رنگی Gray

تصویر اصلی

۲-۲-۷) ذخیره تصاویر:

در مورد ورود و نحوه نمایش تصاویر به صورت کاملا کاربردی و کافی توضیح داده شد.حال نوبت به نحوه ذخیره یک تصویر ایجاد شده رسیده است. در صورتی که نیاز به ذخیره کردن یک تصویر دارید از دستور زیر استفاده کنید.

pic)imwrite , ’ مسیر ذخیره و نام تصویر به همراه پسوند دلخواه‘);

این دستور تصویری که در متغییر pic قرار دارد را در مسیری که تعیین کرده اید را با نام و پسوند دلخواه شما،ذخیره می کند.در صورتی که مسیر را برای ذخیره انتخاب نکنید و تنها نام وپسوند تصویر را بنویسید،نرم افزار تصویر را در مسیر جاری که در بالای پنجره فرمان نوشته شده است،ذخیره می کند.

به مثال توجه کنید.

```
>> x=imread('D:\pictures\wallpaper(2)\01010.jpg');
>> imwrite(x,'D:\01010.png');
```

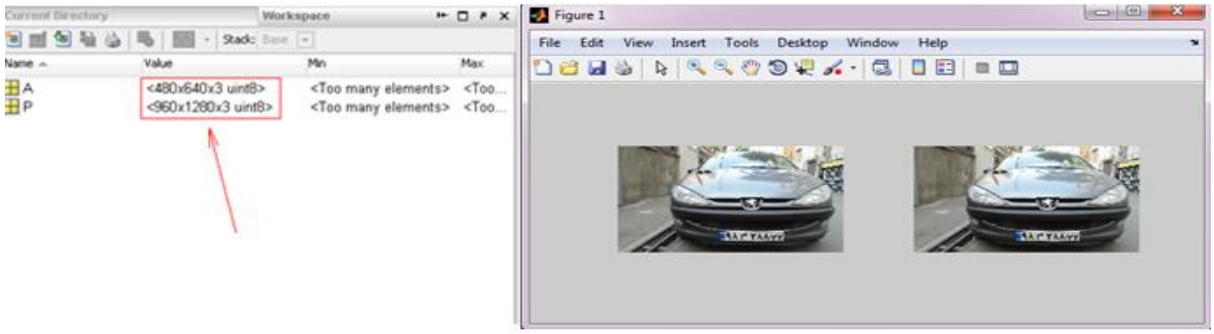
نکته:نرم افزار تمامی تصاویری را که برای فراخوانی پشتیبانی می کند،در زمان ذخیره هم پشتیبانی خواهد کرد.شما می توانید حتی فرمت تصاویر را هم تغییر دهید.

۲-۲-۸) تغییر اندازه تصویر:

(pic2=imresize(pic,x

این تابع اندازه تصویری در متغییر pic را به اندازه نسبت مقدار x تغییر داده و در متغییر pic2 قرار می دهد.مقدار ۱ برای x به این معنی می باشد که تصویر جدید برابر با تصویر اصلی می باشد.در مثال زیر اندازه تصویر را ۵۰ درصد کوچک تر می کنیم:

```
>> p=imread('C:\Users\parviz\Desktop\New folder\126(01).png');
>> a=imresize(p,.5);
>> subplot(1,2,1),imshow(p)
>> subplot(1,2,2),imshow(a)
```



البته می توانید اندازه تصویر خروجی خود را به اندازه دلخواه هم تنظیم نمایید مانند زیر

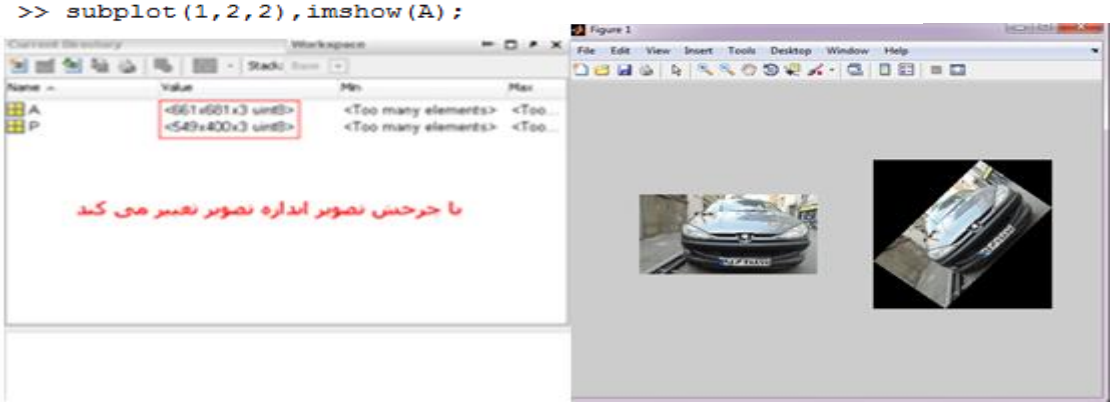
در این مثال فوق تصویر به اندازه۵۰ سطر در ۱۰۰ ستون در می اید.

۲-۲-۹) چرخش تصویر:

(pic2=imrotate(pic,x

این تابع تصویری در متغییر pic را به اندازه مقدار x در جهت ساعت گرد و مقیاس درجه ، چرخش داده و در متغییر pic2 قرار می دهد.به مثال توجه کنید:

```
>> p=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> A=imrotate(p,50);
>> subplot(1,2,1),imshow(p);
>> subplot(1,2,2),imshow(A);
```




```
>> p=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> A=imresize(P,[50 100]);
```

۲-۲-۱۰) چیدن و جدا کردن بخشی از تصویر:

یکی از پرکاربردترین ابزار برای پردازش تصویر،جدا کردن بخشی از تصویر می باشد.برای مثال شما در یک تصویر تنها نیاز به بخشی از تصویر را دارید و با این روش ناحیه مورد نظر خود را جدا می کنید. این کار باعث افزایش سرعت پردازش نیز خواهد شد، چون هر چه تصویر کوجکتر باشد تصویر دارای پیکسل های کمتری خواهد بود و در زمان درگیری cpu کمتر خواهد شد و در نتیجه انجام عملیات از سرعت بالاتری برخوردار خواهد بود.

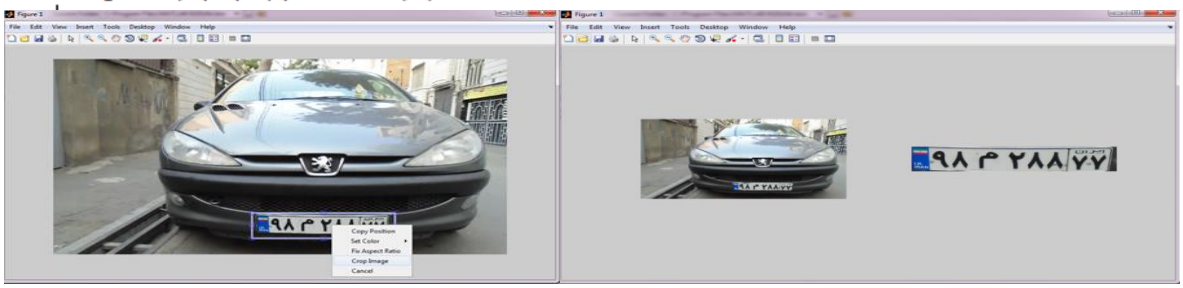
۲-۲-۱۰-۱) *چیدن و جدا کردن تصویر توسط موس:*

;(pic2=imcrop(pic

در این روش کاربر با انتخاب ناحیه مورد نظر خود توسط موس می تواند به این مقصود خود برسد.
حال دستور چیدن تصویر را نوشته و توسط کلیک چپ ماوس ناحیه مورد نظر خود را انتخاب کنید.حال با کلیک راست کردن بر روی ناحیه انتخاب شده در تصویر اصلی و گزینه crop image ، ناحیه مورد نظر جدا شده و در درون متغییر A قرار می گیرد:

;(A=imcrop(p

```
>> subplot(1,2,1),imshow(p)
>> subplot(1,2,2),imshow(A)
```



۲-۲-۱۰-۲) *چیدن وجدا کردن تصویر با استفاده از مختصات ناحیه:*

;([pic2=imcrop(pic,[Y1 X1 Y2 X2

در این روش با وارد کردن مختصات سطر ها و ستون های ماتریس تصویر می تواند به این مقصود خود برسید.مقادیر y1وX1 تعیین کننده مختصات سطر وستون نقطه برش است و مقادیر y2وX2 تعیین کننده طول و عرض بخش مورد نظر می باشد.برای درک بیشتر دستور به مثال زیر توجه کنید:
.حال می خواهیم سطر ۲۰۰ و ستون ۲۰تا و به مقدار ۸۰۰ پیکسل به سمت راست و ۷۰۰ پیکسل به سمت پایین تصویر را جدا کنیم:

```
>> A=imcrop(p,[20 200 800 700]);
>> imshow(A)
```

نکته: اعداد سطر و ستون همان اندازه تصویر می باشد که در پنجره workspace مشخص می شود. برای نمونه تصویر اصلی و جدا شده که در مثال وجود دارد دارای مشخصات زیر می باشد.

۲-۲-۱۱) نمایش مستطیل بر روی تصویر توسط ورود مختصات:

imrect(gca,[Y X width height)

در صورتی که بخواهید مستطیل که مختصات X و Y ان را از قبل تعیین کرده اید را بر روی تصویر نمایش دهید،از این روش استفاده می شود. اگر کادر اطراف مستطیل را در نظر بگیرید؛Xو Y مختصات نقطه شروع مستطیل (بال به سمت چپ) است و width اندازه عرض مستطیل و height اندازه ارتفاع مستطیل می باشد. برای اشنایی بیشتر به مثال زیر توجه کنید.

```
>> p=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> a=rgb2gray(p);
>> subplot(1,2,1),imshow(a);
>> subplot(1,2,2),imhist(a);
```

۲-۳) *فیلترها و بهینه سازی های تصاویر*

.

۲-۳-۱) نمودار هیستوگرام:

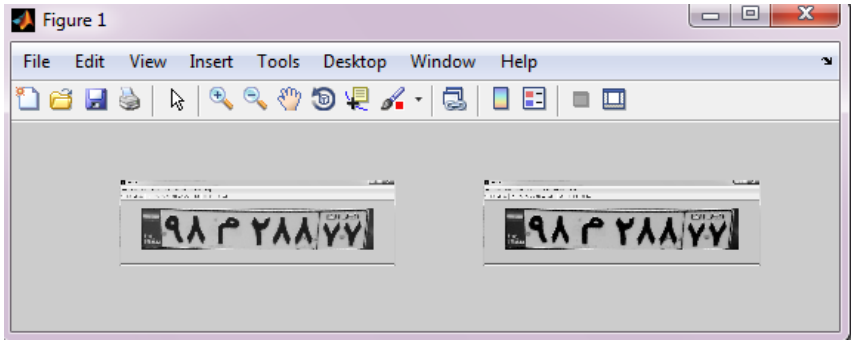
تنها در تصویرهایی از نوع شدت نور برای نمایش میزان کنتراست تصویر(محور افقی) نسبت به شدت نور تابیده شده به تصویر(محور عمودی) می باشد. این نمودار برای تصاویر دو بعدی میزان دامنه شدت نور را در مقادیر مختلف کنتراست که هم به صورت رقمی و هم به صورت رنگ در محور افقی نمایش داده شده است را بیان می کند.این نمودار بهترین روش برای مقایسه دو تصویراز تار نما خواهد بود. برای نمایش این تابع از دستور زیر استفاده می شود.

A	<281x621x3 uint8>	<Too ...	<Too ...
p	<480x640x3 uint8>	<Too ...	<Too ...

نمایش نمودار هیستوگرام تصویر:

imhist(pic)

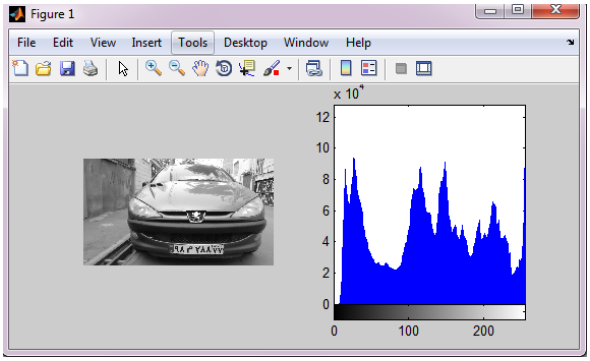
به مثال زیر برای اشنایی ببیشتر توجه نمایید:



۲-۳-۲) تنظیم کننده اتوماتیک تصویر:

imadjust

در دنیای واقعی اکثر تصاویری که از دوربین ها به نرم افزار فرستاده می شوند دارای کیفیت های نا مناسبی می باشد. از جمله مشکلات یک تصویر، میزان تعادل کنتراست و شدت رنگ می باشد. این دستور این عملیات را برای ما انجام می دهد و وضوح رنگ را برای ما افزایش می دهد. این تابع به دو صورت برای تمایز تصاویر سه رنگ و تصویر شدت نور استفاده می شود.



۲-۳-۲-۱) *تنظیم کننده اتوماتیک کنتراست تصویر از نوع شدت نور:*

;(Imadjust(pic

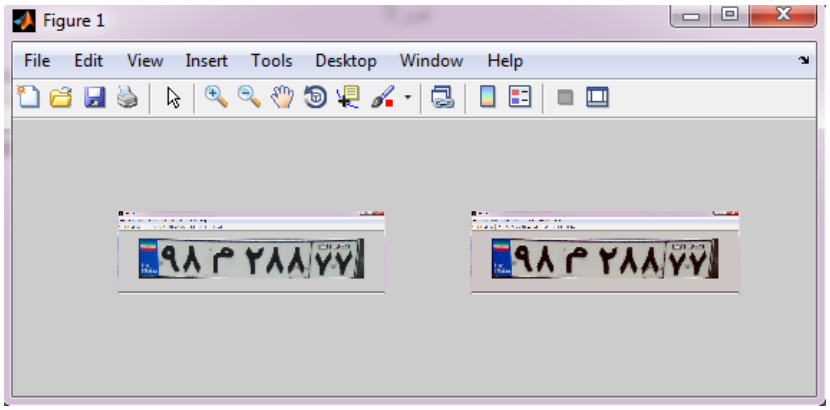
در صورتی که تصویر شما بدون رنگ و از نوع شدت نور باشد،می توانید استفاده کنید،تا تابع به صورت اتوماتیک میزان نور و کنتراست تصویر را بهینه و تنظیم کند. برای اشنایی به مثال زیر توجه نمایید.

۲-۳-۲-۲) *تنظیم کننده اتوماتیک کنتراست تصویر از نوع سه رنگ:*

Imadjust (pic , stretchlim (pic) , [ ]) ;

ر صورتی که تصویر شما رنگی و از نوع سه رنگ باشد،می توانید از این حالت استفاده نمایید،تا تابع به صورت اتوماتیک میزان نور و کنتراست تصویر را بهینه و تنظیم کند. در حقیقت تابع به کمک یک تابع دیکر عمل کرده است. تابع حد کشیدگی شناسایی حد مجاز برای کنتراست تصویر می باشد و تصویر pic را از نظر کمترین و بیشترین میزان کنتراست رنگ بررسی کرده و اطلاعات را در اختیار تابع تنظیم کننده تصویرقرار دهد. برای اشنایی به مثال زیر توجه کنید.

```
>> a=imread('C:\Users\parviz\Desktop\New folder\126(01).png');
>> a1=rgb2gray(a);
>> a2=imadjust(a1);
>> subplot(1,2,1),imshow(a1);
>> subplot(1,2,2),imshow(a2);
```

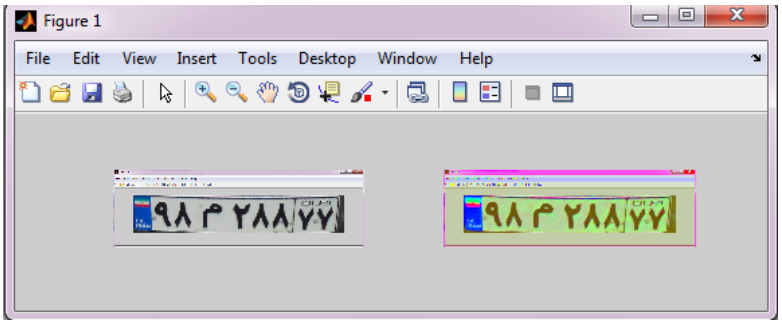


۲-۳-۳) اشباع رنگ:

;(Pic2=decorrstretch(pic1 , ‘ tol ‘ , x

توسط این دستور رنگ های درون تصویر به میزان x به صورت اشباع در امده و باعث همبستگی بین رنگ های درون تصویر می شود. برای درک بیشتر به مثال زیر توجه نمایید.

```
>> a=imread('C:\Users\parviz\Desktop\New folder\126(01).png');
>> a1=decorrstretch(a,'tol',0.01);
>> subplot(1,2,1),imshow(a);
>> subplot(1,2,2),imshow(a1);
```



۲-۳-۴) فیلتر حذف نویز

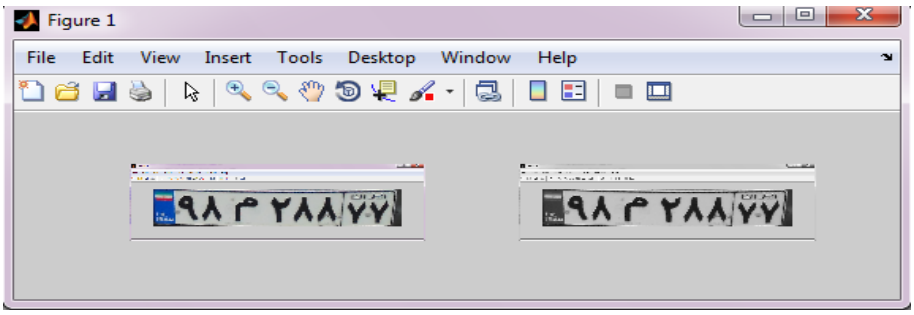
۲-۳-۴-۱) *فیلتر حذف نویز از تصویر:*

Medfid2

حال که با ایجاد نویز در تصویر اشنایی کامل پیدا کرده اید، نوبت به حذف نویز از یک تصویر رسیده است.این بهینه ساز و فیلتر باعث می شود که انواع نویز های موجود در یک تصویر را تا حد قابل قبول حذف کرده و تصویری شفاف تر را در اختیار ما قرار می دهد.روش از بین بردن نویز در این دستور به این صورت است که سیستم چند پیکسل در کنار یکدیگر را بررسی کرده و رنگ میانگین را بدست می اورد.بیشترین رنگ درون تصویر، رنگ اصلی تصویر است و نویز همیشه کمترین میزان رنگ را در یک تصویر به خود اختصاص داده است،به همین دلیل رنگ پیکسل هایی که با رنگ میانگین بدست امده در بخشی از تصویر فاصله زیادی داشته،همان رنگ نویز است و نرم افزار انها را به رنگ میانگین تغییر می دهد. به مثال زیر توجه کنید.

نکته: این دستور تنها برای تصاویر از نوع شدت نور می باشد.

```
>> a=imread('C:\Users\parviz\Desktop\New folder\126(01).png');
>> a1=rgb2gray(a);
>> a2=medfilt2(a1);
>> subplot(1,2,1),imshow(a);
>> subplot(1,2,2),imshow(a1);
```



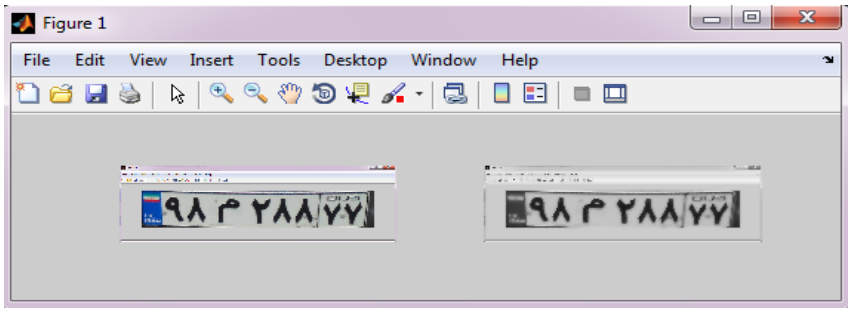
۲-۳-۴-۲) *فیلتر حذف نویز تطابقی:*

(wiener2(pic , [x ,y]

فیاتر هم برای حذف نویز موجود در تصاویر روشی دیگر را استفاده می کند.روش انجام عمل فیاترینگ در این تابع به این صورت است که کاربر تعدادپیکسل های شبکه بندی کردن یک تصویر را در نظر می گیرد.(مقادیر x و.(y با این روش تصویر به صورت یک شبکه تنظیم می شود.که هر شبکه به صورت یک ماتریس که شما سطر و ستون ان را تعیین نموده اید. تابع هر ماتریس یا شبکه را مورد بررسی قرار می دهد و به صورت تطابق رنگ ،یک رنگ را برای پیکسلی که دارای رنگ بسیار متفاوت با پیکسل های اطراف خود داردرا تغییر داده و به رنگ پیکسل های اطراف خود در می اورد.با این روش پیکسل نویز دارای رنگی مطابق با رنگ پیکسل های اطراف خود می شود.از جمله مشکلات این فیلتر، از بین رفتن لبه های تیز در تصویر است.در ادامه به مثالی در این مورد می پردازیم.

نکته:این فیلتر هم تنها برای تصاویر از نوع شدت نور کاربرد دارد،اما می توانید به روش تک رنگ تصاویر رنگی را نیز فیلتر نمایید.

```
>> a=imread('C:\Users\parviz\Desktop\New folder\126(01).png');
>> a1=rgb2gray(a);
>> a2=wiener2(a1,[15,15]);
>> subplot(1,2,1),imshow(a);
>> subplot(1,2,2),imshow(a2);
```



نکته:افزایش اندازه ماتریس تطابق(مقدار X و Y)، باعث از بین رفتن لبه های تصویر و کاهش کیفیت تصویر را منجر خواهد شد

۲-۳-۵) ایجاد فیلتر دلخواه:

شاید هیچ نوع فیلتری نتواند شما را در رسیدن به هدف مورد نظر یاری کند. اینجاست که شما باید خود یک فیلتر طراحی کنید تا به مقصود مورد نظر خود در یک تصویر برسید.تابع زیر برای طراحی و ایجاد یک فیلتر شما را یاری می کند. اما برای استفاده از این فیلتر باید از دستوراتی استفاده نمایید.

۲-۳-۵-۱) *طراحی فیلتر:*

fspecial(type)

برای استفاده از این فیلتر بر روی یک تصویر باید فرم و الگوریتم استفاده از این فیلتر از پیش معرفی شود. برای مثال شما باید تعیین کنید که الگوریتم استفاده شما از این فیلتر در تصویر به صورت شبکه های مربعی است یا دایره و یا نوع عملیاتی که می خواهید در تصویر انجام شود،چیست.این تابع برای هدف های مختلف دارای فرم های مختلفی می باشد. در ادامه فقط به فیلتر شدت نور اشاره خواهیم کرد.

type	توصیف عملکرد
‘average’	فیلتر با روش میانگین ماتریس با شبکه مربعی(محو کننده)
‘disk’	فیلتر با روش میانگین ماتریس با شبکه گرد(محو کننده)
‘gaussain’	فیلتر پایین گذر گوس

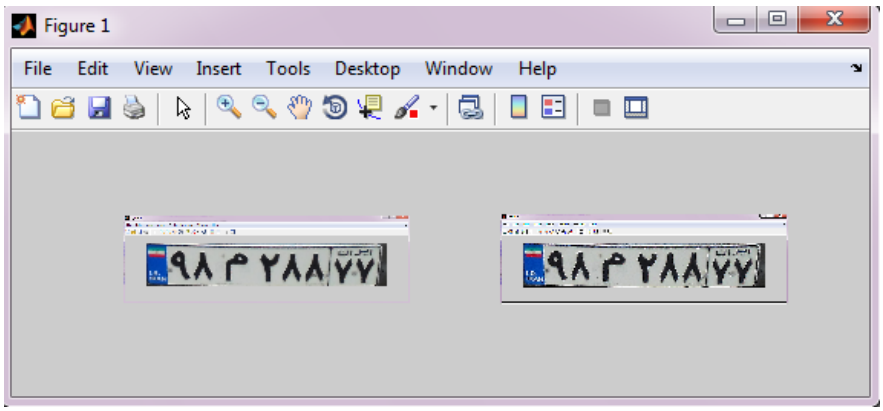
‘laplacian’	فیلتر لاپلاس
‘log’	فیلتر لاپلاس با روش حذف گوس
‘motion’	فیلتر شوک دهنده(حرکت دهنده)
‘prewitt’	فیلتر تقویت لبه
‘sobel’	فیلتر لبه افقی و عمودی
‘unsharp’	فیلتر افزایش شدت نور

۲-۳-۵-۲) *طراحی افزایش دهنده شدت نور لبه ها:*

(fspecial(‘ unsharp ‘, a

این فیلتر عکس الگوریتم میانگین عمل می کند و باعث افزایش نور در تصویر و لبه های ان می شود.میزان شدت تیزی و روشن شدن تصویر را توسط مقدار a می توان تعیین کرد.این مقدار بین ۰ تا ۱ قابل تغییر است. مقدار a به صورت پیش فرض برابر ۰,۲ می باشد.

```
>> a=imread('C:\Users\parviz\Desktop\New folder\126(01).png');
>> f=fspecial('unsharp',0.9);
>> x=imfilter(a,f);
>> subplot(1,2,1),imshow(a);
>> subplot(1,2,2),imshow(x);
```



۲-۴) *شناسایی اشیا*

هدف کلی شناسایی اشیا جدا سازی و پیاده کردن یک هدف در تصویر است.

۲-۴-۱) ارتباط پیکسل در تصویر:

قبل از هر چیزی به این نکته توجه نمایید که از این به بعد به اشیاء درون تصویر ابجکت گفته می شود.هر یک از این ابجکت ها توسط اتصال پیکسل ها به همدیگر در یک تصویر به وجود امده است.اتصال و ارتباط این پیکسل ها به یکدیگر در حقیقت مهم ترین بخش پردازش تصویر و ارتباط با ابجکت ها در یک تصویر است.پیکسل ها در یک تصویر نسبت به بعد تصویر دارای همسایگی و شیوه اتصال به یکدیگر هستند.در حقیقت چیزی که پیکسل را از پیکسل همسایه و کناری خود مجزا می کند،تنها مقداری است که ارایه پیکسل به خود اختصاص داده است.اکثر کسانی که در زمینه پردازش تصویر به فعالیت مشغول هستند،اخرین پردازش خود را بر روی تصاویر باینری انجام می دهند.این تنها به دلیل ساده تر شدن عملیات پردازش در این تصاویر است.چون پیکسل های این نوع تصاویر دارای مقادیر 0 و1 است و تفکیک و جدا سازی ابجکت ها بسیار ساده تر می باشد.

نکته: در اکثر توابع این فصل که با object ها و تصاویر باینری درگیر هستند، ابجکت به رنگ سفید و زمینه به رنگ سیاه در نظر گرفته و شناسایی می شوند.

۲-۴-۲) انواع همسایگی پیکسل ها:

در تصاویر دو نوع همسایگی وجود دارد که عبارتند از:

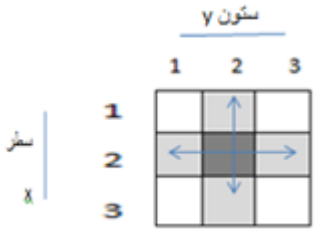
همسایگی ۴ تایی

همسایگی ۸ تایی

البته در تصاویر سه بعدی و یا بیشتر که موضوع بحث ما نمی باشد،این نوع اتصالات و همسایگی به نسبت بعد تصویر افزایش خواهد یافت. برای مثال تصویری که دارای دو بعد است،در همسایگی ۴تایی از بالا و پایین و چپ و راست دارای پیکسل های همسایه است،اما همین پیکسل در تصویر سه بعدی از ۶ طرف دارای همسایگی پیکسلی است.اما بحث ما در زمینه دو بعدی می باشد و نیازی به فراگیری تصاویر سه بعدی و یا بیشتر نمی باشد.

۲-۴-۲-۱) *همسایگی ۴ تایی:*

اگر به تصویر زیر که تنها دارای ابعاد ۳*۳ پیکسل می باشد توجه کنید، خواهید دید که پیکسل وسط از بالا و پایین و چپ و راست دارای همسایه هایی در خط افقی و عمودی می باشد. به این ۴ پیکسل همسایگی ۴ تایی گویند.

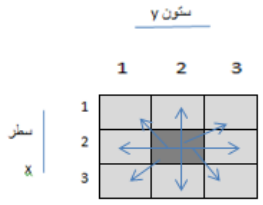


برای معرفی پیکسل های همسایه ۴ تایی به حالت زیر سطر و ستون های انها نمایش می دهیم.

(x,y)(=(1,2) , (2,1) , (2, 3) , (3,2

۲-۴-۲-۲) *همسایگی ۸ تایی:*

اگر دوباره به تصویر زیر که تنها دارای ابعاد ۳*۳ پیکسل می باشد توجه کنید،خواهید دید که پیکسل وسط از بالا و پایین و چپ و راست دارای همسایه هایی در خط افقی و عمودی می باشد. اما در زاویه ۴۵ درجه،از هر سمت خود هم باز دارای اتصالی است.به این ۸ پیکسل که به پیکسل وسط در ارتباط هستند،همسایگی ۸ تایی گویند.



برای معرفی پیکسل های همسایه ۸تایی به حالت زیر سطر و ستون های انها نمایش می دهیم.

(x , y)=(1 ,1), (1, 2), (1 ,3), (2, 1), (2 ,3), (3, 1), (3, 2), (3 ,3)

نکته: همسایگی ۴ و ۸ تایی هر دو دو بعدی می باشند و بیشترین کاربردی ندارند.همسایگی ها در بعضی از توابع که به تصاویر باینری و object ها مربوط است،کاربرد فراوانی دارند.

۲-۴-۳) **اماده سازی تصویر برای تبدیل به تصویر باینری مطلوب:**

roipoly

اخرین مرحله ای که در بحث فیلتر کردن و بهینه سازی یک تصویر بررسی می شود، در حقیقت نا خاصی های یک تصویر را حذف می کنیم تا از بروز اشتباه کاسته شود. نا خالصیی یعنی وجود ابجکت هایی که از نظر کاربر غیر قابل استفاده و مزاحم می باشد.در ادامه با توابعی در این زمینه اشنا خواهید شد.

۲-۴-۳-۱) *انتخاب بخشی از تصویر برای انجام عملیات:*

همان طور که می دانید برای انجام پردازش تصویر نیاز به میان حافظه و cpu قابل قبولی در سیستم می باشد، اما شما هم باز انجام روش هایی برر روی تصویر می توانید به عملیات پردازش تصویر خود سرعت بخشید. در صورتی که شما محدوده مورد نظر خود را در یک تصویر بدانید، و بخواهید که عملیاتی مانند فیاتر کردن ر روی ان نجام دهید،دیگر نیازی نیست که تمام تصویر را فیلتر نمایید،فیلتر کردن تمام تصویر باعث کاهش سرعت پردازش می شود. اما با روش شما سرعت را بهینه خواهید کرد.البته این تنها یکی از کاربرد های این تابع می باشد و نسبت به خلاقیت شما ،از تابع بهره های زیادی را می توان برد.برای اشنایی با نحوه عملکرد این تابع مثالی اورده شده است .برای دستیابی و استفاده از این تابع دو روش وجود دارد:

۱-با استفاده از موس

۲-با استفاده از ورود مختصات

۲-۴-۳-۲) *انتخاب ناحیه با استفاده از ماوس:*

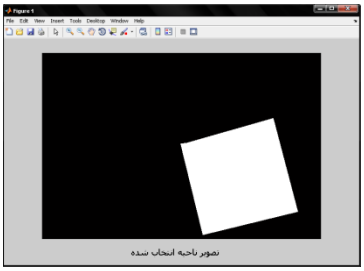
(N=roipoly(pic

در این حالت با استفاده از موس می توان ناحیه مورد نظر خود را انتخاب نمایید.در این مثال ناحیه کاری شما تنها در اطراف یکی از طاس ها می باشد.

```
>> p=imread('D:\pictures\wallpaper(2)\Wallpaper (25).jpg');
>> n=roipoly(p);
```

این عملیات تصویر نشان داده می شود و شما توسط کلیک چپ ماوس ناحیه اطراف طاوس را انتخاب نمایید، بعد از انتخاب ناحیه مورد نظر ، بر روی ناحیه کلیک راست کرده و گزینه ایجاد ماسک را انتخاب نمایید تا ناحیه مورد نظر در متغییر n قرار گیرد.

در درون متغییرn تصویر زیر قرار می گبرد که ناحیه انتخاب شده به رنگ سفید و بقیه به رنگ سیاه می باشد(تصویر نوع باینری)



تا اینجا مراحل استفاده از این تابع پایان یافته است.حال برای مثال می خواهیم تنها در ابن ناحیه فیلتر خاص را اجرا نماییم. اما برای استفاده از این فیلتر در ناحیه انتخاب شده تصویر،نیاز به استفاده از توابعی دیگر می باشد.

قبلا برای اجرای فیلتر های فیلتر خاص بر روی تصاویر، از تابعی به نام فیلتر عکس استفاده می کردیم. این تابع فیلتر را بر روی کل تصویر عملی می کرد. اما حال برای اجرای انواع فیلتر فیلتر خاص و یا دیگر فیلتر ها بر روی ناحیه ای از تصویر،نیاز به تابعی دارید که فیلتر را با استفاده از مختصات ناحیه مورد نظر کاربر طراحی کند و بر روی ناحیه قرار دهد.

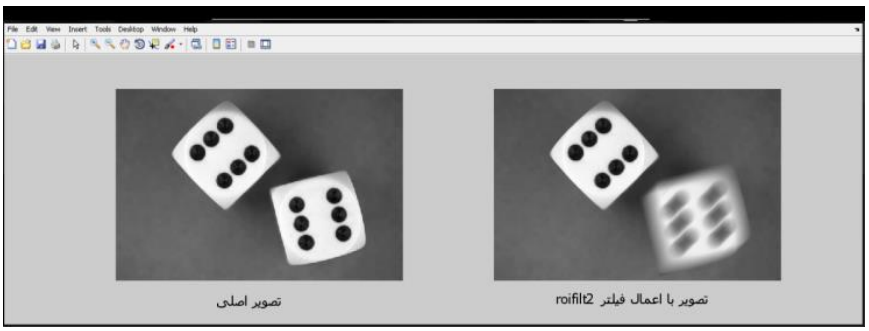
۲-۴-۳-۳) *اجرای عملیات بر روی ناحیه انتخاب شده تصویر:*

(x=roifilt2 (filter name , pic , n

از این تابع برای اجرای فیلتر ها بر روی ناحیه انتخاب شده،استفاده می شود. بعد از انتخاب ناحیه مورد نظر خود در یک تصویرو طراحی فیلتر مورد نظر خود،از این تابع برای اجرای فیلتر بر روی ناحیه انتخاب شده استفاده می شود. مراحل ساخت فیلتر فیلتر خاص مانند قبل می باشد. و تنها به جای استفاده از تابع فیلتر عکس، از این تابع استفاده می شود. این مثال ادامه قبل است و ناحیه مورد نظر در متغییر n قرار می گیرد.

نکته: برای اجرای دستور باید تصویر از نوع شدت نور باشد یا اینکه از روش تک رنگ استفاده شده باشد.

```
>> p=imread('C:\Users\parviz\Desktop\New folder\126(01).png');
>> p=rgb2gray(p);
>> f=fspecial('motion',50,45);
>> x=roifilt(f,p,n);
>> subplot(1,2,1),imshow(p)
>> subplot(1,2,2),imshow(x)
```



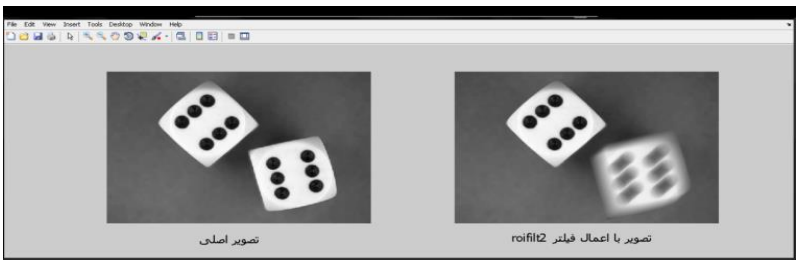
در تصویر مشاهده می کنید که فیلتر حرکت تنها بر روی ناحیه اطراف یکی از طاس ها اجرا شده است. حال نوبت به استفاده از روش دوم در انتخاب ناحیه ای از تصویر رسیده است.

۲-۴-۳-۴) *انتخاب ناحیه با استفاده از ورود مختصات:*

(n=roipoly(pic , y, x

اگر در زمان استفاده از ماوس برای انتخاب ناحیه ای در تصویر دقت کرده باشید، خواهید دید که با هر کلیک چپ ماوس بر روی تصویر، نقطه ای نمایان می شود. در حقیقت نرم افزار از مختصات همین نقاط است که ناحیه را تشخیص می دهد.حال شما با این تابع می توانید این مختصات را وارد نموده تا نرم افزار خود ناحیه را ترسیم نماید.این مختصات نقاط همان عدد سطر و ستون پیکسل ها است.در این روش شما دو ماتریس تعیین می کنید(ماتریس x و y) که در یکی حاوی عدد سطر نقاط است(x) و ماتریس دوم حاوی عدد ستون این نقاط (y) می باشد.با این کار ناحیه تعیین می شود و برای اجرای مراحل فیلتر گذاری مانند قبل ادامه می دهید.

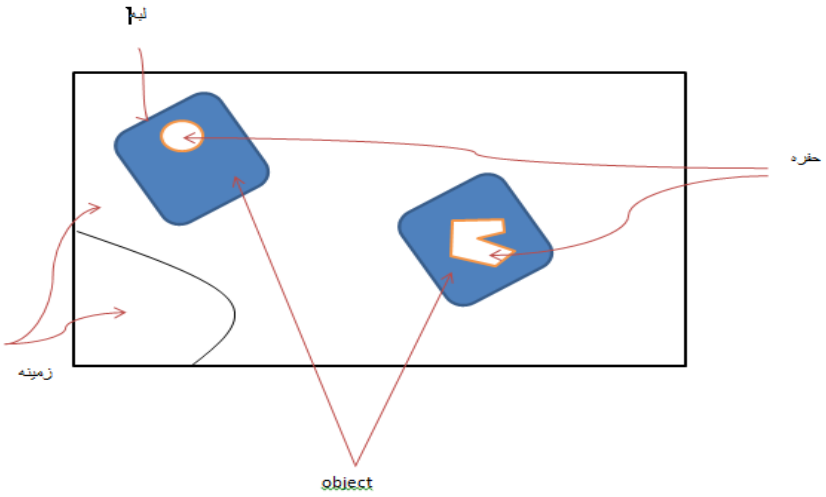
```
>> p=imread('D:\pictures\wallpaper(2)\Wallpaper (25).jpg');
>> x=[1222 1270 1300 1200 1800];
>> y=[200 220 100 125 100];
>> n=roipoly(p,y,x);
>> p=rgb2gray(p);
>> f=fspecial('motion',50,45);
>> pic=roifilt2(f,p,n);
>> subplot(1,2,1),imshow(p);
>> subplot(1,2,2),imshow(pic);
```



۲-۴-۴) لبه در تصویر:

لبه در تصویر چیست؟لبه ها به نقاطی از تصویر گفته می شود که در ان نقاط دو پیکسل در کنار یکدیگر ،دو مقدار متفاوت داشته باشند و یا دو مقدار فاصله زیادی از نظر ارزش عددی داشته باشند.لبه ها در حقیقت مرز بین زمینه و ابجکت را در یک تصویر مشخص می کنند.در شکل زیر تفاوت بین لبه و زمینه و حفره مشاهده می شود.

نکته: فضایی حتی به اندازه یک پیکسل که در یک ابجکت محصور شده است را حفره گویند.



۲-۴-۵) شناسایی لبه:

edge

این تابع با فرم ها و الگوریتم های مختلف ، عملیات شناسایی لبه را پی ریزی می کند. با این روش شما می توانید ابجکت ها را هم در ادامه شناسایی نمایید. در کل این تابع یکی از مهم ترین توابه در علم پردازش تصویر می باشد. الگوریتم های فراوانی برای شناسایی لبه در این تابع وجود دارد و. این نکته بسیار مهم است که بدانید از کدام الگوریتم در کدام موقعیت باید استفاده نمود.برای رسیدن به این اگاهی نیاز به خلاقیت و تمرین بسیار می باشد.

نکته : این تابع برای تصاویر از نوع شدت نور و باینری قابل استفاده است.البته با روش تک رنگ می توان تصاویر سه رنگ را هم لبه برداری نمود.

در ادامه انواع متد ها و الگوریتم های استفاده از این تابع بیان شده است.

متد sobel

متد prewitt

متد Roberts

متدlog

متد zerocross

متد canny

نکته: تصویر خروجی لبه برداری شده تمامی متد ها از نوع باینری خواهد بود

۲-۴-۵-۱) متد *sobel:*

این متد از روش تقریب مشتق تصویر برای شناسایی لبه استفاده می کند و هر جای تصویر که شیبی در اعداد پیکسل ها وجود داشته باشد، همان جا را لبه معرفی می کند.در این متد،تابع به سه فرم مختلف استفاده می شود.

حالت اول: در این فرم تابع به صورت اتوماتیک از تصویر pic، لبه برداری می کند و در متغییر Xقرار می دهد.

```
>> X=edge(pic,'sobel');
>> X=edge(pic,'sobel',s);
>> X=edge(pic,'sobel',s,dir);
```

حالت دوم: در این حالت و فرم تابع،مقدار s مشخص کننده مقدار حساسیت است که اکثر اوقات مقداری بین ۰ تا ۱ می توان برای ان انتخاب کرد.بیشترین حساسیت را مقدار ۰ دارد. در حالت اول مقدار s به صورت پیش فرض بهینه می باشد و نسبت به تصویر مقداری مناسب انتخاب می کند.

حالت سوم: در این فرم تابع به غیر از مقدار s برای میزان حساسیت، گزینه دایرکتوری وجود دارد که می تواند یکی از سه گزینه زیر باشد:

horizontal: این گزینه در زمان لبه برداری ،تنها خطوط عمودی را لبه برداری و نمایش می دهد.

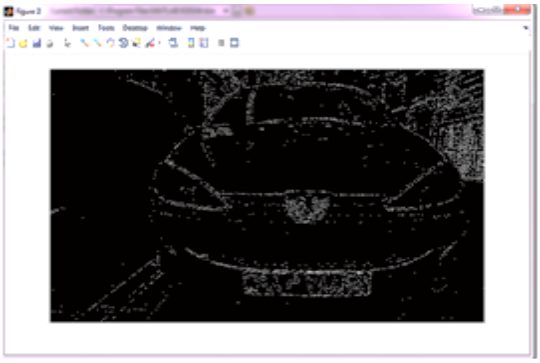
vertical:این گزینه در زمان لبه برداری ،تنها خطوط عمودی را لبه برداری و نمایش می دهد.

both: این گزینه در زمان لبه برداری ، تمامی خطوط افقی و عمودی را لبه برداری و نمایش می دهد. در حالت اول و دوم فرم تابع دایرکتوری گزینه " هر دو" را به صورت پیش فرض دارند.

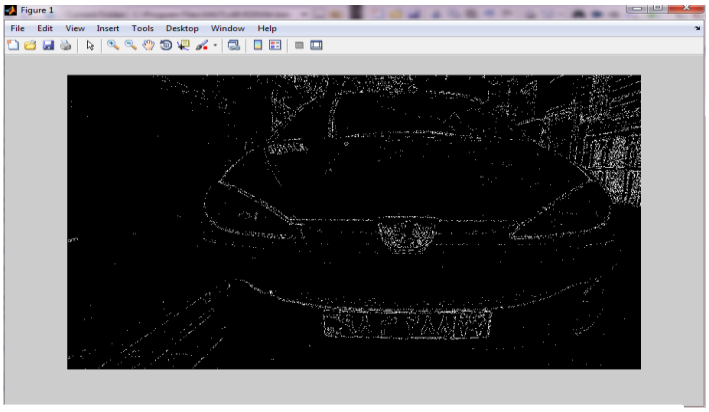
مثال:

```
>> p=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> p=rgb2gray(p);
>> N1=edge(p,'sobel');
>> N2=edge(p,'sobel',0.1,'horizontal');
>> figure,imshow(p);
>> figure,imshow(N1);
>> figure,imshow(N2);
```

تصویر اصلی



شناسایی لبه با متد sobel



شناسایی لبه های افقی با متد sobel

۲-۴-۵-۲) متد *prewitt:*

این متد هم از روش تقریب مشتق تصویر برای شناسایی لبه استفاده می کند و هر جای تصویر که شیبی در اعداد پیکسل ها وجود داشته باشد ،همان جا را لبه معرفی می کند و تنها تفاوت با متد قبل وجود تصریح کننده تصویر می باشد. در این متد، تابه به سه فرم مختلف استفاده می شود.

حالت اول: در این فرم تابع به صورت اتوماتیک از تصویر pic، لبه برداری می کند و در متغییر X قرار

```
>> X=edge(pic,'prewitt');
>> X=edge(pic,'ptewitt',s);
>> X=edge(pic,'prewitt',s,dir);
```

می دهد.

حالت دوم: در این حالت و فرم تابع،مقدار s مشخص کننده مقدار حساسیت است که اکثر اوقات مقداری بین ۰ تا ۱ می توان برای ان انتخاب کرد.بیشترین حساسیت را مقدار ۰ دارد. در حالت اول مقدار s به صورت پیش فرض بهینه می باشد و نسبت به تصویر مقداری مناسب انتخاب می کند.

حالت سوم: در این فرم تابع به غیر از مقدار s برای میزان حساسیت، گزینه دایرکتوری وجود دارد که می تواند یکی از سه گزینه زیر باشد:

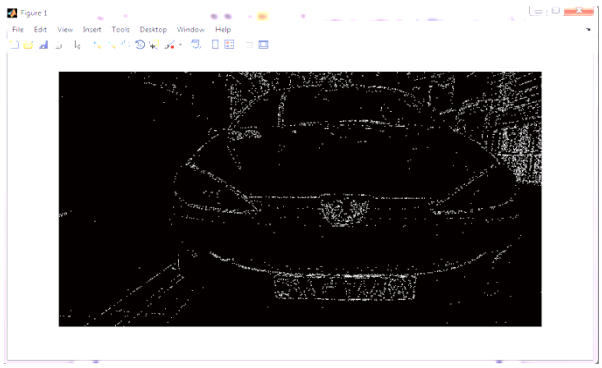
horizontal: این گزینه در زمان لبه برداری ،تنها خطوط عمودی را لبه برداری و نمایش می دهد.

vertical:این گزینه در زمان لبه برداری ،تنها خطوط عمودی را لبه برداری و نمایش می دهد.

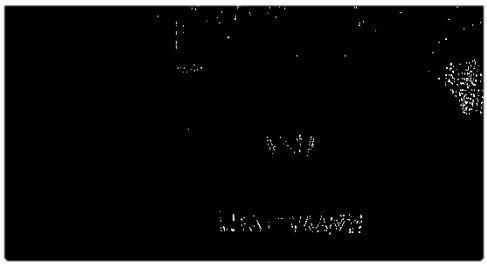
both: این گزینه در زمان لبه برداری ، تمامی خطوط افقی و عمودی را لبه برداری و نمایش می دهد. در حالت اول و دوم فرم تابع دایرکتوری گزینه "هر دو" را به صورت پیش فرض دارند.

مثال:

```
>> p=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> p=rgb2gray(p);
>> N1=edge(p,'prewitt');
>> N2=edge(p,'prewitt',0.1,'vertical');
>> figure,imshow(N1);
>> figure,imshow(N1);
>> figure,imshow(N2);
```



شناسایی لبه با متد prewitt



شناسایی لبه های عمودی با متد prewitt

۲-۴-۵-۲) متد roberts:

این متد هم از روش تقریب مشتق تصویر برای شناسایی لبه استفاده می کند و هر جای تصویر که شیبی در اعداد پیکسل ها وجود داشته باشد ،همان جا را لبه معرفی می کند و تنها تفاوت با متد قبل وجود افزایش دهنده حجم لبه می باشد. در این متد، تابه به سه فرم مختلف استفاده می شود.

```
>> x=edge(pic,'roberts');
>> x=edge(pic,'roberts',s);
>> x=edge(pic,'roberts',s,option);
```

حالت اول: در این فرم تابع به صورت اتوماتیک از تصویر pic، لبه برداری می کند و در متغییر X قرار می دهد.

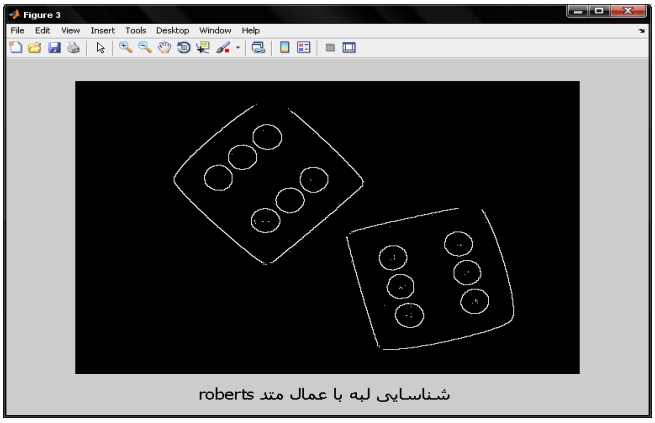
حالت دوم: در این حالت و فرم تابع،مقدار S مشخص کننده مقدار حساسیت است که اکثر اوقات مقداری بین ۰ تا ۱ می توان برای ان انتخاب کرد.بیشترین حساسیت را مقدار ۰ دارد. در حالت اول مقدار S به صورت پیش فرض بهینه می باشد و نسبت به تصویر مقداری مناسب انتخاب می کند.
حالت سوم: در این فرم تابع به غیر از مقدار S برای میزان حساسیت، گزینه" اپشن "وجود دارد که می تواند یکی ازدو گزینه زیر باشد:

thinning: این گزینه در زمان لبه برداری ،خطوط را با حجم کمتری لبه برداری و نمایش می دهد.

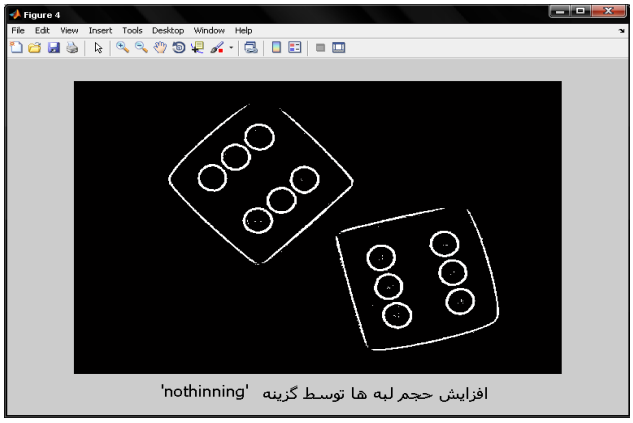
nothinning: این گزینه در زمان لبه برداری ،خطوط را با حجم بیشتری لبه برداری و نمایش می دهد.
مثال :

```
>> p=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> p=rgb2gray(p);
>> N1=edge(p,'roberts');
>> N2=edge(p,'roberts','nothinning');
>> figure,imshow(p);
>> figure,imshow(N1);
>> figure,imshow(N2);
```

تصویر اصلی



شناسایی لبه با عمال متد roberts



افزایش حجم لبه ها توسط گزینه 'nothinning'

۲-۴-۵-۴) متد *log*:

این متد از روش لاپلاس تصویر برای شناسایی لبه استفاده می کند. در این متد ، تابه به سه فرم مختلف استفاده می شود.

حالت اول: در این فرم تابع به صورت اتوماتیک از تصویر pic، لبه برداری می کند و در متغییر X قرار می دهد.

حالت دوم: در این حالت و فرم تابع،مقدار s مشخص کننده مقدار حساسیت است که اکثر اوقات مقداری بین ۰ تا ۱ می توان برای ان انتخاب کرد.بیشترین حساسیت را مقدار ۰ دارد. در حالت اول مقدار s به صورت پیش فرض بهینه می باشد و نسبت به تصویر مقداری مناسب انتخاب می کند.

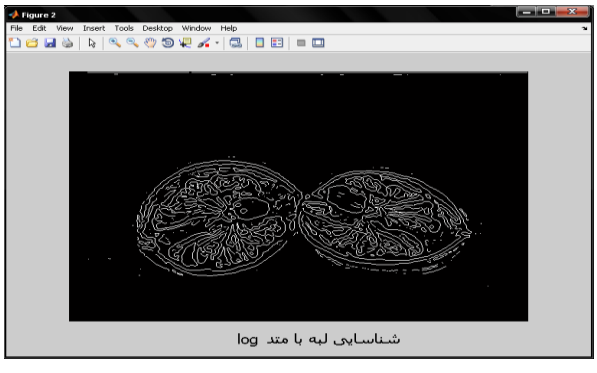
حالت سوم: در این فرم تابع به غیر از مقدار s برای میزان حساسیت، مقداری برای سیگما وجود دارد که در صورت استفاده نکردن از این مقدار (مانند دو حالت قبل) به صورت پیش فرض ۲ را دارد. سیگما در حقیقت یک انحراف معیار برای متد است و ضریبی می باشد که اندازه شبکه لبه برداری را تعیین می کند. این شبکه از رابطه زیر بدست خواهد امد.این شبکه یک ماتریس در اندازه n*n است.

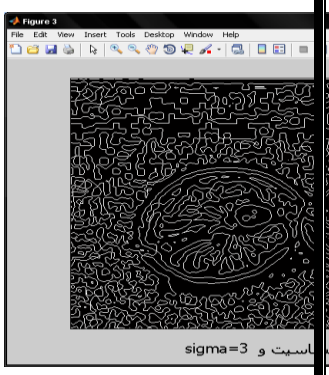
$$n=ceil(sigma*3)*2+1$$

```
>> X=edge(pic,'log');
>> X=edge(pic,'log',s);
>> X=edge(pic,'log',s,sigma);
```

مثا ل :

```
>> p=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> p=rgb2gray(p);
>> N1=edge(p,'log');
>> N2=edge(p,'log',0,3);
>> figure,imshow(p);
>> figure,imshow(N1);
>> figure,imshow(N2);
```





```
>> X=edge(pic,'canny');
>> X=edge(pic,'canny',s);
>> X=edge(pic,'canny',s,sigma);
```

۲-۴-۵-۵) متد *zerocross:*

یکی از بهترین متد های لبه برداری متد" گذر از صفر"می باشد. میزان حساسیت این متد بسیار بالا می باشد. در این متد، تابع به سه فرم مختلف استفاده می شود.

حالت اول: در این فرم تابع به صورت اتوماتیک از تصویر pic، لبه برداری می کند و در متغییر X قرار می دهد.
حالت دوم: در این حالت و فرم تابع،مقدار s مشخص کننده مقدار حساسیت است که اکثر اوقات مقداری بین ۰ تا ۱ می توان برای ان انتخاب کرد.این حساسیت بسیار بالاست به طوری که تغییرات S به میزان

```
>> X=edge(pic,'zerocross');
>> X=edge(pic,'zerocross',s);
>> X=edge(pic,'zerocross',s,filter name);
```

۴ رقم اعشار می تواند در تصویر تغییر ایجاد نمایید.بیشترین حساسیت را مقدار ۰ دارد. در حالت اول مقدار s به صورت پیش فرض بهینه می باشد و نسبت به تصویر مقداری مناسب انتخاب می کند.
حالت سوم: در این فرم تابع به غیر از مقدار s برای میزان حساسیت، می توان از یک فیلتر برای تصویر استفاده نمود.که روش طراحی فیلتر ها قبلا توضیح داده شده است.

۲-۴-۵-۶) متد *canny:*

یکی از بهترین متد های لبه برداری ، متد "هوشمند "در زمینه تشخیص چهره و یا اثر انگشت می باشد. به غیر از میزان حساسیت بسیار بالای این متد، کیفیت لبه برداری تصویر بسیار عالی و مطلوب می باشد. در ایم متد ، تابع به سه فرم مختلف استفاده می شود.

حالت اول: در این فرم تابع به صورت اتوماتیک از تصویر pic، لبه برداری می کند و در متغییر X قرار می دهد.
حالت دوم: در این حالت و فرم تابع،مقدار s مشخص کننده مقدار حساسیت است که اکثر اوقات مقداری بین ۰ تا ۱ می توان برای ان انتخاب کرد.این حساسیت بسیار بالاست به طوری که تغییرات S به میزان ۴ رقم اعشار می تواند در تصویر تغییر ایجاد نمایید.بیشترین حساسیت را مقدار ۰ دارد. در حالت اول مقدار s به صورت پیش فرض بهینه می باشد و نسبت به تصویر مقداری مناسب انتخاب می کند.
حالت سوم: در این فرم تابع به غیر از مقدار s برای میزان حساسیت، مقداری برای سیگما وجود دارد که در صورت استفاده نکردن از این مقدار (مانند دو حالت قبل) به صورت پیش فرض ۱ را دارد. سیگما در حقیقت یک انحراف معیار برای متد است و ضریبی می باشد که اندازه شبکه لبه برداری را تعیین می کند. این شبکه از رابطه زیر بدست خواهد امد.این شبکه یک ماتریس در اندازه n*n است.

n=ceil(sigma*3)*2+1

۲-۴-۶) کار بر روی تصاویر مورد نظر

در این مرحله بیشتر بر روی تصاویر از نوع باینری کار می شود تا به یک شکل کلی و مطلوب از ابجکت مورد نظر برسیم. اما اول باید با مفهوم سازه اشنا شوید.

سازه چیست؟

سازه از یک یا چند پیکسل تشکیل شده که با قرار گرفتن این پیکسل ها در کنا یکدیگر، تشکیل یک شکل خاص می دهند که به انها سازه گویند.

۲-۴-۷) روش طراحی عنصر سازه:

هر شکلی می تواند به شکلی خاص باشد.مثلا مربعی یا دایره، اما دلیل ساخت این این نوع سازه ها متفاوت است. یکی از کاربرد های ساخت این نوع سازه ها فیلتر کردن اشیا و ابجکت های نا خالص است یا به عبارتی ابجکت هایی که ما نیازی به انها نداریم.برای ساخت این سازه ها از انواع اتصالات پیکسل ها استفاده می شود. در ادامه با تابعی اشنا می شوید که با ان می توان انواع سازه های مجازی را ساخت.

۲-۴-۷-۱) *طراحی ساختمان یک سازه:*

strel

سازه از شبکه ای ماتریسی در ابعاد دلخواه تشکیل شده است که تنها دارای مقادیر ۰ و ۱ می باشد. این تابع می تواند انواع مختلفی از فرم پیکر های یک سازه را طراحی نماید. در جدول زیر انواع مدهای استفاده از این تابع معرفی شده است.

مد تابع	تشریح عملکرد
Arbitrary	تبدیل ماتریس دلخواه به سازه
Diamond	سازه لوزی شکل
Disk	سازه ای گرد
Line	سازه ای خطی
Octagon	سازه ای ۸ گوشه
Pair	سازه ای دو عنصری
periodicline	سازه ای خطی با فواصل تعیین شونده
Rectangle	سازه ۴ گوشه
Square	سازه ای مربعی

۲-۴-۷-۲) *ساخت سازه دلخواه :*

(X=strel('arbitrary',mat

با استفاده از این فرم دستور می توانید ماتریس mat دلخواه خود را به یک سازه تبدیل نمایید.با این روش شما می توانید هر شکل همایسگی که مورد نظرتان می باشد را به صورت سازه تبدیل کرده و بکار بگیرید. برای درک بیشتر به مثال توجه نمایید.

با این عملیات شما سازه ای به شکل زیر طراحی کرده اید که دارای ۶ همسایه است. در شکل رنگ خاکستری نشان دهنده سازه است(مقادیر ۱) و این تعداد ۱ ها نشان دهنده تعداد همسایگی در سازه است.

هر سازه دارای یک مرکز ثقل می باشد. در این حالت استفاده از این تابع باید نرم افزار از ماتریس تعریف شده شما یک مرکز ثقل برای خود در نظر بگیرید.به کمک فرمول زیر مرکز ثقل سازه تعریف می شود.

O=floor(size(mat)+1)/2

در رابطه بالا که تمام دستورات درون ان در فصول قبل توضیح داده شده است،متغییر O مکان مرکز ثقل سازه را به صورت عدد سطر و ستون نشان می دهد. برای نمونه در سازه مثال قبل،مرکز ثقل سازه ارایه سطر ۲ و ستون ۲ می باشد.

۲-۴-۷-۳) ساخت سازه لوزی شکل:

در این نوع سازه مرکز ثقل در وسط سازه و مشخص است. مقدار X مشخص کننده فاصله مرکز ثقل سازه تا نوکهای لوزی است(شعاع لوزی).به مثال زیر توجه نمایید.

با این تابع سازه ای لوزی شکل با شعاع برابر ۳ ساخته می شود

```
>> X=strel('diamond',3)
```

```
>> X=strel('disk',r)
```

0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0
0	0	0	1	0	0	0

مرکز ثقل سازه

```
>> X=strel('disk',4)
```

```
>> M=[0 0 1 0;0 1 0 0;1 1 1 0;0 0 1 0];
>> I
```

0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0

0	0	1	0
0	1	0	0
1	1	1	0
0	0	1	0

مرکز ثقل سازه

۲-۴-۷-۴) ساخت سازه دایره ای شکل :

در این سازه که به شکل دایره است،مرکز ثقل کاملا در وسط سازه و مشخص است.مقدار r مشخص

```
X=strel('diamond',x)
```

کننده شعاع دایره می باشد.البته هر چه شعاع یزرگتر باشد، سازه شباهت بیشتری به دایره خواهد داشت.

به مثال زیر توجه نمایید.

با این تابع سازه ای لوزی شکل با شعاع برابر با ۳=۱-۴ ساخته می شود که به صورت شکل زیر می باشد.

نکته : در صورت استفاده از مقدار ۳ برای شعاع این سازه به شکل مستطیلی شکل خواهد شد،چون

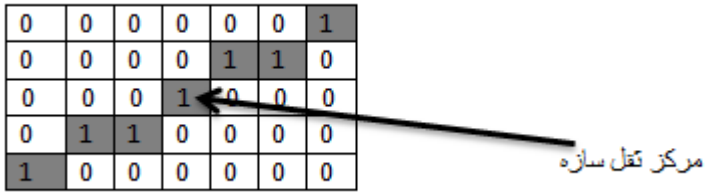
```
>> X=strel('line',L,d)
```

کمترین مقدار لبه های این نوع سازه عدد ۳ می باشد و حتی در صورتی که از اعدد کوچکتر از ۳ برای شعاع سازه استفاده نمایید،سازه لوزی شکل می شود.

```
>> X=strel('line',9,30)
```

۲-۴-۷-۵) ساخت سازه خطی شکل :

این نوع سازه به شکل یک خط است.طول خط را مقدار L مشخص می کند و زاویه ای را که این خط با



سطح افقی تشکیل می دهد، توسط مقدار d مشخص می شود. زاویه بر حسب درجه است. این سازه هم مانند سازه قبل دارای مرکز ثقلی در وسط سازه می باشد. اما برای مشخص کردن طول سازه حتما از اعداد فرد استفاده نمایید تا مرکز ثقل همیشه در وسط سازه قرار بگیرد. به مثال توجه نمایید.

با این فرم شما سازه ای خطی به طول ۷پیکسل و با زاویه ۳۰ درجه نسبت به سطح افقی مانند شکل بالا

```
>> X=strel('octagon',r)
```

طراحی می کنید.

۲-۴-۷-۶) ساخت سازه ۸ گوشه:

برای ساخت یک سازه ۸ گوشه از این فرم تابع استفاده نمایید.مقدار r حتما باید مضربی از عدد ۳ باشد تا تابع عمل کند و سازه ۸ گوشه تشکیل شود.به مثال زیر توجه کنید.

با این فرم سازه ای دارای ۸ گوشه با شعاع ۳ تشکیل می شود. البته با شعاع ۳، سازه شبیه به سازه دایره ای می باشد، اما با افزایش مقدار r شکل ۸ گوشه کامل تر نمایان می شود.

```
>> X=strel('octagon',3)
```

0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0

مرکز ثقل سازه

۲-۴-۷-۷) ساخت سازه ۲ عنصری یا جفت:

```
>> X=strel('pair',n)
```

این نوع فرم در وسط یک عنصر قرار می دهد و با فاصله ای به اندازه سطر و ستونی به سمت راست و پایین، یک عنصر دیگر را قرار می دهد. این فاصله سطر و ستون در ماتریس n قرار گرفته باشد. برای درک بیشتر به مثال زیر توجه نمایید.

```
>> n=[2 3];
>> X=strel('periodicline',x,n)
```

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	1

مرکز ثقل سازه

مشاهده می کنید که عنصر دوم با فاصله ۲ سطر پایین تر و ۳ ستون به سمت راست تشکیل شده است.

۲-۴-۷-۸) ساخت سازه خطی با فواصل **تعیین** شونده:

این نوع سازه همان سازه خطی می باشد،با این تفاوت که فاصله عناصر از یکدیگر قابل تغییر است. و زاویه را همین فواصل بین عناصر تعیین کننده است. اگر وسط این خط را در نظر بگیرید،مقدار x تعیین کننده تعداد عناصر در هر طرف این خط است و مقدار n به صورت یک ماتریس تعیین کننده فاصله هر عنصر با عنصر دیگر است.برای اینکه بیشتر با مفهوم این سازه اشنا شوید،به مثال زیر توجه نمایید.

```
>> n=[1 2];
>> X=strel('periodicline',2,n)
   >> X=strel('rectangle',n)
```

1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1

مرکز ثقل سازه

۲-۴-۷-۹) ساخت سازه ۴ گوشه :

این فرم از تابع یک سازه ۴ گوشه را تشکیل می دهد که اندازه طول و عرض ان به صورت یک ماتریس توسط متغییر n مشخص می شود. به مثال زیر توجه نمایید. برای اینکه این سازه دارای یک مرکز ثقل در وسط سازه باشد، بهتر است که مقادیر ماتریس n، فرد باشد.

```
>> n=[3 5];
>> x=strel('rectangel',n)
```

```
>> x=sterl('square',x)
```

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

مرکز ثقل

۲-۴-۷-۱۰) ساخت سازه مربع شکل :

اخرین سازه ای که مورد برسی قرار می گیرد سازه مربعی شکل است که مقدار x مشخص کننده اندازه ضلع مربع است.بهتر است برای اینکه مرکز ثقل سازه در وسط قرار گیرد از مقادیر فرد برای x استفاده شود.

```
>> x=sterl('square',5)
```

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

مرکز ثقل سازه

۲-۴-۸) استفاده از سازه ها در توابع:

توابعی که از این سازه ها برای انجام عملیات خود استفاده می کنند،کاربرد بسیار زیادی در شناسایی ابجکت ها و تفکیک انها از یکدیگر را دارند. البته این توابه به غیر تصاویر باینری بر روی تصاویر شدت نور هم قابل اجرا هستند. شما باید در استفاده از این توابع خلاقیت زیادی را نشان دهید تا بتوانید بهترین بهره را از توابع ببرید در ادامه با این توابع اشنا می شوید.

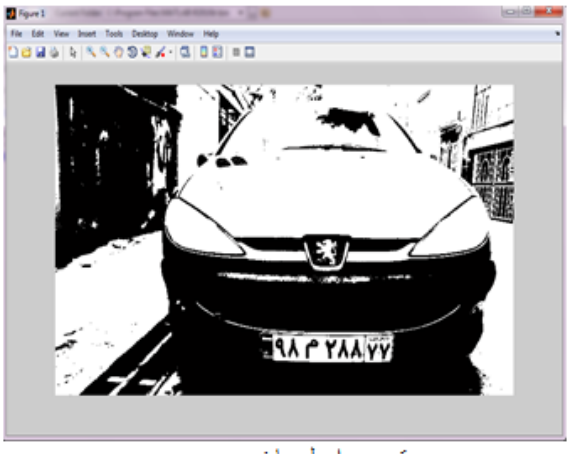
۲-۴-۸-۱) open **image**:

```
imopen(pic,strel name)
```

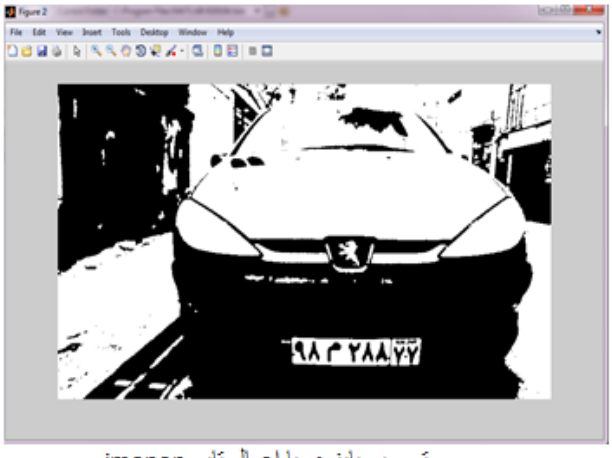
یکی از توابعی که از سازه های ساخته شده استفاده می کند، تابع باز کردن تصویر است.این تابع به صورت یک فیلتر برای حذف ابجکت ها ی هم اندازه و هم شکل با سایز ها (نام سازه) در یک تصویر (pic) عمل می کند. البته در تصاویر شدت نور و سه رنگ هم تغییراتی ایجاد می کند. این تابع در تصاویر شدت نور و سه رنگ هر ابجکتی که هم شکل و هم اندازه در تصویر باشد را به صورت یک رنگ میانگین از همان ابجکت در می اورد.البته این اتفاق در تصاویر باینری هم افتاده است.اما به خاطر دو رنگ بودن این تصاویر ،این اتفاق نمایان نیست. برای درک مطلب فوق به مثال زیر توجه نمایید.

مثال برای تصاویر باینری:

```
>> p=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> p=im2bw(p);
>> x=strel('diamond',10);
>> t=imopen(p,x);
>> figure(1),imshow(p);
>> figure(2),imshow(t);
```



تصویر اصلی باینری



تصویر باینری با اعمال تابع imopen

همانطور که مشاهده می کنید، نقاط کوچک که هم اندازه و شکل سازه لوزی شکل است، حذف شده است.

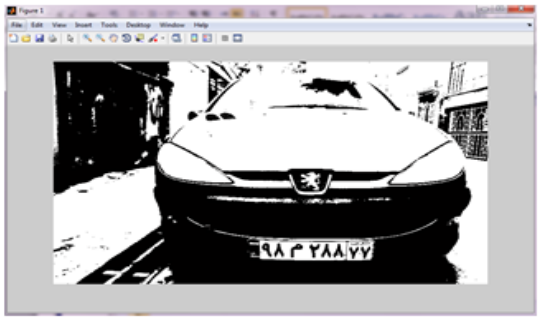
۲-۴-۸) close *image* : :

```
imclose(pic,strel name)
```

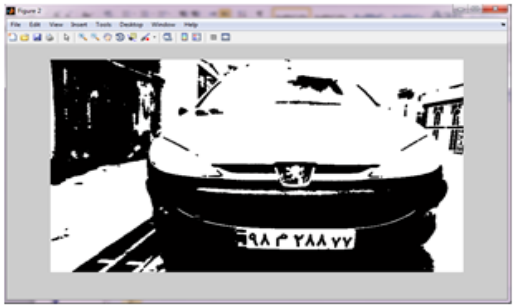
این تابع دقیقا عکس تابع باز کردن عکس عمل می کند و افزایش دهنده نقاطی است که به اندازه و شکل سازه مورد نظر می باشد.به مثال زیر توجه نمایید.

مثالی برای تصویر باینری:

```
>> p=imread('C:\Users\parviz\Desktop\pelak\2(13).jpg');
>> p=im2bw(p);
>> x=strel('diamond',10);
>> t=imclose(p,x);
>> figure(1),imshow(p);
>> figure(2),imshow(t);
```



تصویر اصلی باینری



تصویر باینری با اعمال تابع imclose

در این مثال نقاط کوچک که هم اندازه و شکل سازه لوزی شکل است، دارای حجم بیشتری شده است.

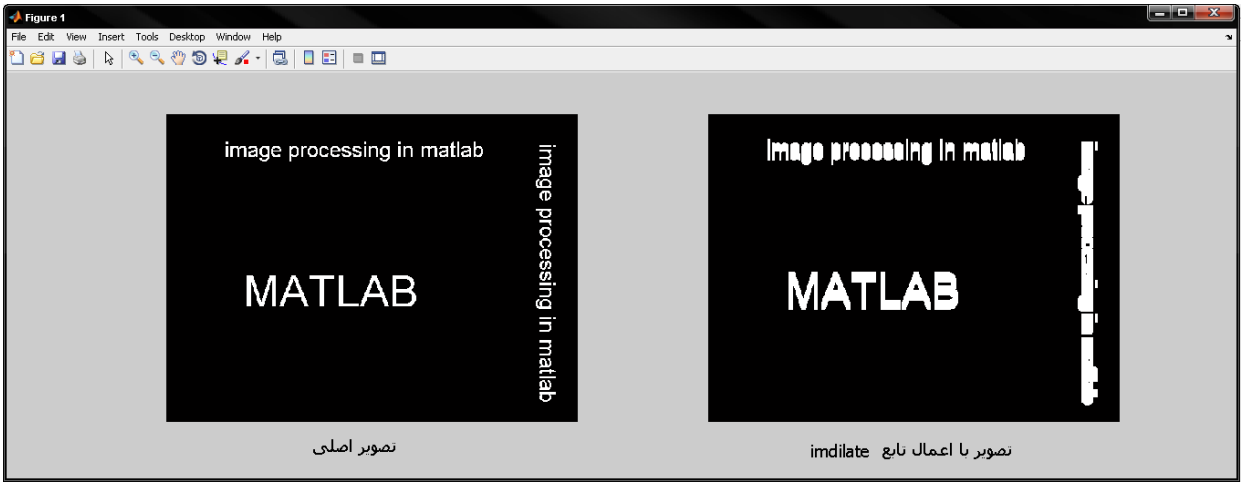
۲-۴-۹) **اضافه کننده و افزایش دهنده همسایگی** :

این تابع پیکسل هایی که دارای همسایگی می باشند را در نظر می گیرد و به انها یک سازه که مد نظر

```
imdilate(pic,strel name)
```

کاربر می باشد را اضافه می کند. برای درک بیشتر به مثال زیر توجه کنید.

```
>> p=imread('C:\Users\parviz\Desktop\New folder\126(01).png');
>> x=strel('line',10,90);
>> t=imdilate(p,x);
>> subplot(1,2,1),imshow(p);
>> subplot(1,2,2),imshow(t);
```



تصویر اصلی

تصویر با اعمال تابع imdilate

فصل سوم

پروژه پیشنهادی

۳-۱) برنامه نوشته شده در متلب

```
clf;
clear;
clc;
aks='2 (13).jpg;'
٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪
pic=imread(aks;)
٪ pic= imresize(pic,[1200,1600]);
 pic01 = fspecial('average');
 pic02= imfilter(pic,pic01,'replicate');
 pic1=rgb2gray(pic02);
 pic11=imadjust(pic1]);
 pic12=medfilt2(pic11,[10,10]);
 pic13=wiener2(pic11,[10,10]);
 pic2=edge(pic1,'sobel');
٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪
pic290=bwmorph(pic2,'diag');
pic2911=bwmorph(pic290,'dilate');
pic2912=bwmorph(pic2911,'dilate');
pic2913=bwmorph(pic2912,'dilate');
pic292=imfill(pic2913,'holes');
pic293=imopen(pic292,strel('line',50,0));
pic294=imopen(pic293,strel('line',15,90));
pic295=bwareaopen(pic294,15000);

pic200=imdilate(pic2,strel('line',2,0));
pic201=imdilate(pic200,strel('line',1,90));
pic202=imfill(pic201,'holes');
pic203=imopen(pic202,strel('line',50,0));
pic204=imopen(pic203,strel('line',15,90));
pic205=bwareaopen(pic204,15000);


pic210=imdilate(pic2,strel('line',4,0));
pic211=imdilate(pic210,strel('line',2,90));
```

```
pic212=imfill(pic211,'holes');
pic213=imopen(pic212,strel('line',50,0));
pic214=imopen(pic213,strel('line',15,90));
pic215=bwareaopen(pic214,10000);

pic220=imdilate(pic2,strel('line',6,0));
pic221=imdilate(pic220,strel('line',3,90));
pic222=imfill(pic221,'holes');
pic223=imopen(pic222,strel('line',50,0));
pic224=imopen(pic223,strel('line',15,90));
pic225=bwareaopen(pic224,25000);

pic230=imdilate(pic2,strel('line',8,0));
pic231=imdilate(pic230,strel('line',2,90));
pic232=imfill(pic231,'holes');
pic233=imopen(pic232,strel('line',50,0));
pic234=imopen(pic233,strel('line',15,90));
pic235=bwareaopen(pic234,10000);

pic240=imdilate(pic2,strel('line',8,0));
pic241=imdilate(pic240,strel('line',4,90));
pic242=imfill(pic241,'holes');
pic243=imopen(pic242,strel('line',50,0));
pic244=imopen(pic243,strel('line',15,90));
pic245=bwareaopen(pic244,15000);

pic250=imdilate(pic2,strel('line',12,0));
pic251=imdilate(pic250,strel('line',8,90));
pic252=imfill(pic251,'holes');
pic253=imopen(pic252,strel('line',50,0));
pic254=imopen(pic253,strel('line',15,90));
pic255=bwareaopen(pic254,20000);


pic260=imdilate(pic2,strel('line',18,0));
pic261=imdilate(pic260,strel('line',10,90));
pic262=imfill(pic261,'holes');
pic263=imopen(pic262,strel('line',50,0));
```

```
pic264=imopen(pic263,strel('line',15,90));
pic265=bwareaopen(pic264,15000);

pic270=imdilate(pic2,strel('line',24,0));
pic271=imdilate(pic270,strel('line',12,90));
pic272=imfill(pic271,'holes');
pic273=imopen(pic272,strel('line',50,0));
pic274=imopen(pic273,strel('line',15,90));
pic275=bwareaopen(pic274,20000);

pic280=imdilate(pic2,strel('line',30,0));
pic281=imdilate(pic280,strel('line',15,90));
pic282=imfill(pic281,'holes');
pic283=imopen(pic282,strel('line',50,0));
pic284=imopen(pic283,strel('line',15,90));
pic285=bwareaopen(pic284,15000);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for ii=0:7
mm=['pic225'  'pic295'  'pic285'  'pic205'  'pic255'  'pic265'  'pic245'  'pic275'  'pic235']
mmm=mm((1+(ii*6)):(6*(ii+1)))
mmmm=eval(mmm);
[lab ww]=bwlabel(mmmm);
Dat=regionprops(lab,  'BoundingBox'  ,  'Orientation'  );

ww=0;
for i=1:ww
   b=dat(i);
   b1=b.BoundingBox  ;
   b4=b.Orientation;



   if b1(3)>=b1(4)*3          &          b1(3)<=b1(4)*7
   out=b1;
   zavie=b4;
  www=www+1
```

```
ss=0;
while (ss<20)
 ss=ss+1
 rand=rands(1);

im=imcrop(pic1,[out(1)+(20*rand)   out(2)-10   out(3)+20   out(4)+20]);
im1=imrotate(im,0-zavie);
im2=imadjust(im1,[]);
im3=decorrstretch(im2,'tol',0.01);
im4=im2bw(im3,(rand*0.2)+graythresh(im3));
im5=imdilate(im4, strel('disk',0));
im6=imfill(im5,'holes');
im7 = xor(im5,im6);
im8=bwareaopen(im72,300);
im9=imfill(im8,'holes');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[x,y]=size(im9);
  s=sum(im9);

  for i=1:y
    if s(i)>=0.75*x  ||  s(i)<=((0.02.*rand)+0.02)*x
       s(i)=0;
    end
  end;

 y3=zeros(1,16);
 n=1;
for i=2:y
    end
    if s(i)>=1  &&  s(i-1)==0
       y3(1,n)=i;
       n=n+1;
    end
    if s(i)==0  &&  s(i-1)>=1
       y3(n)=i;
```

```
        n=n+1;
end


[x4,y4]=size(y3);
for i=2:y4-2
   if y3(i)-y3(i-1)<=4
      y3(i-1)=0;
   end
   if y3(i+1)-y3(i)<=4
      y3(i+1)=0;
   end
end


y5=zeros(1,y4);
n=1;
for i=1:y4
   if y3(i)>=1
      y5(n)=y3(i);
      n=n+1;
   end
end

if n==17
   break
end
end



if n==17
   break
end
end
end

if n==17
```

```
   break
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

m11=imcrop(im9 ,[y5(1), 0 , y5(2)-y5(1) , x]);
m21=imcrop(im9,[y5(3), 0 , y5(4)-y5(3) , x]);
m31=imcrop(im9 ,[y5(5), 0 , y5(6)-y5(5) , x]);
m41=imcrop(im9 ,[y5(7), 0 , y5(8)-y5(7) , x]);
m51=imcrop(im9 ,[y5(9),0 ,  y5(10)-y5(9) , x]);
m61=imcrop(im9 ,[y5(11),0 , y5(12)-y5(11) , x]);
m71=imcrop(im9 ,[y5(13),0 , y5(14)-y5(13) , x]);
m81=imcrop(im9 ,[y5(15),0 , y5(16)-y5(15) , x]);

m12=imopen(m11,strel('disk',1));
m22=imopen(m21,strel('disk',1));
m32=imopen(m31,strel('disk',1));
m42=imopen(m41,strel('disk',1));
m52=imopen(m51,strel('disk',1));
m62=imopen(m61,strel('disk',1));
m72=imopen(m71,strel('disk',1));
m82=imopen(m81,strel('disk',1));



m13=bwareaopen(m12,200);
m23=bwareaopen(m22,200);
m33=bwareaopen(m32,200);
m43=bwareaopen(m42,200);
m53=bwareaopen(m52,200);
m63=bwareaopen(m62,200);
m73=bwareaopen(m72,200);
m83=bwareaopen(m82,200);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[x1,y1]=size(m13);
```

```
[x,y]=find(m13==1);
minx=min(x);
maxx=max(x);
 miny=min(y);
 maxy=max(y);
 m1=imcrop(m13 ,[miny, minx , maxy-miny , maxx-minx]);
m1=imresize(m1,[60 30]);

[x1,y1]=size(m23);
 [x,y]=find(m23==1);
 minx=min(x);
 maxx=max(x);
 miny=min(y);
 maxy=max(y);
 m2=imcrop(m23 ,[miny, minx , maxy-miny , maxx-minx]);
 m2 = imresize(m2, [60 30]);

 [x1,y1]=size(m33);
 [x,y]=find(m33==1);
 minx=min(x);
 maxx=max(x);
  miny=min(y);
 maxy=max(y);
 m3=imcrop(m33 ,[miny, minx , maxy-miny , maxx-minx]);
 m3 = imresize(m3, [60 30]);

 [x1,y1]=size(m43);
 [x,y]=find(m43==1);
 minx=min(x);
 maxx=max(x);
  miny=min(y);
 maxy=max(y);
 m4=imcrop(m43 ,[miny, minx , maxy-miny , maxx-minx]);
 m4 = imresize(m4, [60 30]);

 [x1,y1]=size(m53);
 [x,y]=find(m53==1);
 minx=min(x);
```

```
maxx=max(x);
 miny=min(y);
 maxy=max(y);
m5=imcrop(m53 ,[miny, minx , maxy-miny , maxx-minx]);
m5 = imresize(m5, [60 30]);

[x1,y1]=size(m63);
[x,y]=find(m63==1);
minx=min(x);
maxx=max(x);
 miny=min(y);
 maxy=max(y);
m6=imcrop(m63 ,[miny, minx , maxy-miny , maxx-minx]);
m6 = imresize(m6, [60 30]);

[x1,y1]=size(m73);
[x,y]=find(m73==1);
minx=min(x);
maxx=max(x);
 miny=min(y);
 maxy=max(y);
m7=imcrop(m73 ,[miny, minx , maxy-miny , maxx-minx]);
m7 = imresize(m7, [60 30]);


[x1,y1]=size(m83);
[x,y]=find(m83==1);
minx=min(x);
maxx=max(x);
 miny=min(y);
 maxy=max(y);
m8=imcrop(m83 ,[miny, minx , maxy-miny , maxx-minx]);
m8 = imresize(m8, [60 30]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mach1=zeros(1,9);
mach2=zeros(1,9);
```

```
mach3=zeros(1,26);
mach4=zeros(1,9);
mach5=zeros(1,9);
mach6=zeros(1,9);
mach7=zeros(1,9);
mach8=zeros(1,9);
adade1=imread('adade1.bmp');
adade1=~adade1;
adade2=imread('adade2.bmp');
adade2=~adade2;
adade3=imread('adade3.bmp');
adade3=~adade3;
adade4=imread('adade4.bmp');
adade4=~adade4;
adade5=imread('adade5.bmp');
adade5=~adade5;
adade6=imread('adade6.bmp');
adade6=~adade6;
adade7=imread('adade7.bmp');
adade7=~adade7;
adade8=imread('adade8.bmp');
adade8=~adade8;
adade9=imread('adade9.bmp');
adade9=~adade9;
b=imread('b.bmp');
b=~b;
ba=imread('ba.bmp');
ba=~ba;
lam=imread('lam.bmp');
lam=~lam;
jim=imread('jim.bmp');
jim=~jim;
dal=imread('dal.bmp');
dal=~dal;
ta=imread('ta.bmp');
ta=~ta;
sad=imread('sad.bmp');
sad=~sad;
```

```
mem=imread('mem.bmp');
mem=~mem;
he=imread('he.bmp');
he=~he;
se=imread('se.bmp');
se=~se;
ne=imread('ne.bmp');
ne=~ne;
qaf=imread('qaf.bmp');
qaf=~qaf;
qaff=imread('qaff.bmp');
qaff=~qaff;
ye=imread('ye.bmp');
ye=~ye;
vav=imread('vav.bmp');
vav=~vav;
te=imread('te.bmp');
te=~te;
aen=imread('aen.bmp');
aen=~aen;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  n=1;
for x=1:60
   for y=1:30
      if m1(x,y)==adade1(x,y)
         mach1(n)=mach1(n)+1;
      else
         mach1(n)=mach1(n)-1   ;
      end

      if m2(x,y)==adade1(x,y)
         mach2(n)=mach2(n)+1;
      else
         mach2(n)=mach2(n)-1;
      end
```

```
    if m4(x,y)==adade1(x,y)
       mach4(n)=mach4(n)+1;
    else
       mach4(n)=mach4(n)-1;
    end

    if m5(x,y)==adade1(x,y)
       mach5(n)=mach5(n)+1;
    else
       mach5(n)=mach5(n)-1;
    end

    if m6(x,y)==adade1(x,y)
       mach6(n)=mach6(n)+1;
    else
       mach6(n)=mach6(n)-1;
    end

    if m7(x,y)==adade1(x,y)
       mach7(n)=mach7(n)+1;
    else
       mach7(n)=mach7(n)-1;
    end

    if m8(x,y)==adade1(x,y)
       mach8(n)=mach8(n)+1;
     else
       mach8(n)=mach8(n)-1;
    end

  end
end

%%%%%%%%%%%%%%%%%%%%

n=2;
for x=1:60
  for y=1:30
```

```
    if m1(x,y)==adade2(x,y)
       mach1(n)=mach1(n)+1;
       else
       mach1(n)=mach1(n)-1  ;
    end
    if m2(x,y)==adade2(x,y)
       mach2(n)=mach2(n)+1;
       else
       mach2(n)=mach2(n)-1;
    end
    if m4(x,y)==adade2(x,y)
       mach4(n)=mach4(n)+1;
       else
       mach4(n)=mach4(n)-1;
    end
    if m5(x,y)==adade2(x,y)
       mach5(n)=mach5(n)+1;
       else
       mach5(n)=mach5(n)-1;
    end
    if m6(x,y)==adade2(x,y)
       mach6(n)=mach6(n)+1;
       else
       mach6(n)=mach6(n)-1;
    end
    if m7(x,y)==adade2(x,y)
       mach7(n)=mach7(n)+1;
       else
       mach7(n)=mach7(n)-1;
    end
    if m8(x,y)==adade2(x,y)
       mach8(n)=mach8(n)+1;
       else
       mach8(n)=mach8(n)-1;
    end

  end
end
```

```
%%%%%%%%%%%%%%%%%%%%

n=3;
for x=1:60
   for y=1:30
      if m1(x,y)==adade3(x,y)
         mach1(n)=mach1(n)+1;
         else
         mach1(n)=mach1(n)-1  ;
      end
      if m2(x,y)==adade3(x,y)
         mach2(n)=mach2(n)+1;
         else
         mach2(n)=mach2(n)-1;
      end
      if m4(x,y)==adade3(x,y)
         mach4(n)=mach4(n)+1;
         else
         mach4(n)=mach4(n)-1;
      end
      if m5(x,y)==adade3(x,y)
         mach5(n)=mach5(n)+1;
         else
         mach5(n)=mach5(n)-1;
      end
      if m6(x,y)==adade3(x,y)
         mach6(n)=mach6(n)+1;
         else
         mach6(n)=mach6(n)-1;
      end
      if m7(x,y)==adade3(x,y)
         mach7(n)=mach7(n)+1;
         else
         mach7(n)=mach7(n)-1;
      end
      if m8(x,y)==adade3(x,y)
         mach8(n)=mach8(n)+1;
```

```
        else
        mach8(n)=mach8(n)-1;
      end

   end
end

%%%%%%%%%%%%%%%%%%%%

n=4;
for x=1:60
   for y=1:30
      if m1(x,y)==adade4(x,y)
         mach1(n)=mach1(n)+1;
         else
         mach1(n)=mach1(n)-1  ;
      end
      if m2(x,y)==adade4(x,y)
         mach2(n)=mach2(n)+1;
         else
         mach2(n)=mach2(n)-1;
      end
      if m4(x,y)==adade4(x,y)
         mach4(n)=mach4(n)+1;
         else
         mach4(n)=mach4(n)-1;
      end
      if m5(x,y)==adade4(x,y)
         mach5(n)=mach5(n)+1;
         else
         mach5(n)=mach5(n)-1;
      end
      if m6(x,y)==adade4(x,y)
         mach6(n)=mach6(n)+1;
         else
         mach6(n)=mach6(n)-1;
      end
      if m7(x,y)==adade4(x,y)
```

```
        mach7(n)=mach7(n)+1;
        else
        mach7(n)=mach7(n)-1;
      end
      if m8(x,y)==adade4(x,y)
        mach8(n)=mach8(n)+1;
        else
        mach8(n)=mach8(n)-1;
      end

   end
end

%%%%%%%%%%%%%%%%%%%

n=5;
for x=1:60
   for y=1:30
      if m1(x,y)==adade5(x,y)
        mach1(n)=mach1(n)+1;
        else
        mach1(n)=mach1(n)-1  ;
      end
      if m2(x,y)==adade5(x,y)
        mach2(n)=mach2(n)+1;
        else
        mach2(n)=mach2(n)-1;
      end
      if m4(x,y)==adade5(x,y)
        mach4(n)=mach4(n)+1;
        else
        mach4(n)=mach4(n)-1;
      end
      if m5(x,y)==adade5(x,y)
        mach5(n)=mach5(n)+1;
        else
        mach5(n)=mach5(n)-1;
      end
```

```
      if m6(x,y)==adade5(x,y)
         mach6(n)=mach6(n)+1;
         else
         mach6(n)=mach6(n)-1;
      end
      if m7(x,y)==adade5(x,y)
         mach7(n)=mach7(n)+1;
         else
         mach7(n)=mach7(n)-1;
      end
      if m8(x,y)==adade5(x,y)
         mach8(n)=mach8(n)+1;
         else
         mach8(n)=mach8(n)-1;
      end

   end
end



%%%%%%%%%%%%%%%%%%%%

n=6;
for x=1:60
   for y=1:30
      if m1(x,y)==adade6(x,y)
         mach1(n)=mach1(n)+1;
         else
         mach1(n)=mach1(n)-1  ;
      end
      if m2(x,y)==adade6(x,y)
         mach2(n)=mach2(n)+1;
         else
         mach2(n)=mach2(n)-1;
      end
      if m4(x,y)==adade6(x,y)
         mach4(n)=mach4(n)+1;
```

```
        else
        mach4(n)=mach4(n)-1;
    end
    if m5(x,y)==adade6(x,y)
        mach5(n)=mach5(n)+1;
        else
        mach5(n)=mach5(n)-1;
    end
    if m6(x,y)==adade6(x,y)
        mach6(n)=mach6(n)+1;
        else
        mach6(n)=mach6(n)-1;
    end
    if m7(x,y)==adade6(x,y)
        mach7(n)=mach7(n)+1;
        else
        mach7(n)=mach7(n)-1;
    end
    if m8(x,y)==adade6(x,y)
        mach8(n)=mach8(n)+1;
        else
        mach8(n)=mach8(n)-1;
    end

  end
end

%%%%%%%%%%%%%%%%%%%

n=7;
for x=1:60
  for y=1:30
    if m1(x,y)==adade7(x,y)
        mach1(n)=mach1(n)+1;
        else
        mach1(n)=mach1(n)-1  ;
    end
    if m2(x,y)==adade7(x,y)
```

```
        mach2(n)=mach2(n)+1;
        else
        mach2(n)=mach2(n)-1;
      end
      if m4(x,y)==adade7(x,y)
        mach4(n)=mach4(n)+1;
        else
        mach4(n)=mach4(n)-1;
      end
      if m5(x,y)==adade7(x,y)
        mach5(n)=mach5(n)+1;
        else
        mach5(n)=mach5(n)-1;
      end
      if m6(x,y)==adade7(x,y)
        mach6(n)=mach6(n)+1;
        else
        mach6(n)=mach6(n)-1;
      end
      if m7(x,y)==adade7(x,y)
        mach7(n)=mach7(n)+1;
        else
        mach7(n)=mach7(n)-1;
      end
      if m8(x,y)==adade7(x,y)
        mach8(n)=mach8(n)+1;
        else
        mach8(n)=mach8(n)-1;
      end

  end
end

%%%%%%%%%%%%%%%%%%%%

n=8;
for x=1:60
  for y=1:30
```

```
    if m1(x,y)==adade8(x,y)
       mach1(n)=mach1(n)+1;
       else
       mach1(n)=mach1(n)-1  ;
    end
    if m2(x,y)==adade8(x,y)
       mach2(n)=mach2(n)+1;
       else
       mach2(n)=mach2(n)-1;
    end
    if m4(x,y)==adade8(x,y)
       mach4(n)=mach4(n)+1;
       else
       mach4(n)=mach4(n)-1;
    end
    if m5(x,y)==adade8(x,y)
       mach5(n)=mach5(n)+1;
       else
       mach5(n)=mach5(n)-1;
    end
    if m6(x,y)==adade8(x,y)
       mach6(n)=mach6(n)+1;
       else
       mach6(n)=mach6(n)-1;
    end
    if m7(x,y)==adade8(x,y)
       mach7(n)=mach7(n)+1;
       else
       mach7(n)=mach7(n)-1;
    end
    if m8(x,y)==adade8(x,y)
       mach8(n)=mach8(n)+1;
       else
       mach8(n)=mach8(n)-1;
    end
  end
end
```

```
n=9;
for x=1:60
   for y=1:30
      if m1(x,y)==adade9(x,y)
         mach1(n)=mach1(n)+1;
         else
         mach1(n)=mach1(n)-1  ;
      end
      if m2(x,y)==adade9(x,y)
         mach2(n)=mach2(n)+1;
         else
         mach2(n)=mach2(n)-1;
      end
      if m4(x,y)==adade9(x,y)
         mach4(n)=mach4(n)+1;
         else
         mach4(n)=mach4(n)-1;
      end
      if m5(x,y)==adade9(x,y)
         mach5(n)=mach5(n)+1;
         else
         mach5(n)=mach5(n)-1;
      end
      if m6(x,y)==adade9(x,y)
         mach6(n)=mach6(n)+1;
         else
         mach6(n)=mach6(n)-1;
      end
      if m7(x,y)==adade9(x,y)
         mach7(n)=mach7(n)+1;
         else
         mach7(n)=mach7(n)-1;
      end
      if m8(x,y)==adade9(x,y)
         mach8(n)=mach8(n)+1;
         else
         mach8(n)=mach8(n)-1;
      end
```

```
   end
end

%%%%%%%%%%%%%%%%%%%%

n=10;
for x=1:60
   for y=1:30
      if m3(x,y)==jim(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end

n=11;
for x=1:60
   for y=1:30
      if m3(x,y)==dal(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end

n=12;
for x=1:60
   for y=1:30
      if m3(x,y)==sad(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end
```

```
n=13;
for x=1:60
   for y=1:30
      if m3(x,y)==ta(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end

n=14;
for x=1:60
   for y=1:30
      if m3(x,y)==b(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end

n=15;
for x=1:60
   for y=1:30
      if m3(x,y)==lam(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end

n=16;
for x=1:60
   for y=1:30
      if m3(x,y)==mem(x,y)
         mach3(n)=mach3(n)+1;
```

```
        else
        mach3(n)=mach3(n)-1;
      end
   end
end

n=17;
for x=1:60
   for y=1:30
      if m3(x,y)==ba(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end

n=18;
for x=1:60
   for y=1:30
      if m3(x,y)==he(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end

n=19;
for x=1:60
   for y=1:30
      if m3(x,y)==se(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end
```

```
n=20;
for x=1:60
   for y=1:30
      if m3(x,y)==ne(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end

n=21;
for x=1:60
   for y=1:30
      if m3(x,y)==qaf(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end
n=22;
for x=1:60
   for y=1:30
      if m3(x,y)==ye(x,y)
         mach3(n)=mach3(n)+1;
         else
         mach3(n)=mach3(n)-1;
      end
   end
end

n=23;
for x=1:60
   for y=1:30
      if m3(x,y)==vav(x,y)
         mach3(n)=mach3(n)+1;
```

```
        else
        mach3(n)=mach3(n)-1;
      end
   end
end

n=24;
for x=1:60
   for y=1:30
      if m3(x,y)==te(x,y)
        mach3(n)=mach3(n)+1;
        else
        mach3(n)=mach3(n)-1;
      end
   end
end

n=25;
for x=1:60
   for y=1:30
      if m3(x,y)==aen(x,y)
        mach3(n)=mach3(n)+1;
        else
        mach3(n)=mach3(n)-1;
      end
   end
end

n=26;
for x=1:60
   for y=1:30
      if m3(x,y)==qaff(x,y)
        mach3(n)=mach3(n)+1;
        else
        mach3(n)=mach3(n)-1;
      end
   end
end
```

```
end

٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪

a(1)=find(mach1==max(mach1));
a(2)=find(mach2==max(mach2));
a(3)=find(mach3==max(mach3));
a(4)=find(mach4==max(mach4));
a(5)=find(mach5==max(mach5));
a(6)=find(mach6==max(mach6));
a(7)=find(mach7==max(mach7));
a(8)=find(mach8==max(mach8));

٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪٪

for i=1:8
   if a(i)==10
      q(i)='ج';
     elseif a(i)==11
      q(i)='د';
     elseif a(i)==12
      q(i)='ص';
     elseif a(i)==13
      q(i)='ط';
     elseif a(i)==14
      q(i)='ب';
      elseif a(i)==15
      q(i)='ل';
     elseif a(i)==16
      q(i)='م';
      elseif a(i)==17
      q(i)='ب';
      elseif a(i)==18
      q(i)='ه'=
      elseif a(i)==19
```

```
      q(i);'س'=
      elseif a(i)==20
      q(i);'ن'=
      elseif a(i)==21
      q(i);'ق'=
      elseif a(i)==22
      q(i);'ی'=
     elseif a(i)==23
      q(i);'و'=
      elseif a(i)==24
      q(i);'ت'=
      elseif a(i)==25
      q(i);'ع'=
      elseif a(i)==26
      q(i);'ق'=
    else
       q(i)=num2str(a(i));
   end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 imshow(pic);
 imrect(gca,out);

title(['شماره پلاک : ',q(8),'',q(7),'',q(6),'',q(5),'',q(4),'',q(3),'',q(2),'',q(1)]
     ,'fontName','aria' ,'FontSize',19)

  saveas(gcf,aks)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
If   ((q(8)=='1')&(q(7)=='1')&(q(6)=='3')&(q(5)=='6')&(q(4)=='3' )&
     (q(3)=='ب')&(q(2)=='7')&(q(1)=='4'))
'ok'
else
'  error   '
```

```
end
end
```

فصل ششم
نتیجه‌گیری

۴-۱) *نتیجه گیری:*

با توجه به نتایج آزمایشات مشاهده می شود روش ارائه شده نسبت به شرایط نورپردازی، زاویه وجهت تصویر برداری،اندازه پلاک و همچنین شکل پلاک تا حدود زیادی موفق است، اما در مواردی که به دلایل مذکور در قسمت قبل تشخیص اشتباه رخ می دهد و همچنین در مقایسه نتایج دومرحله نیاز به بهینه سازی دارد . بنابراین می توان با استفاده از بازخورد اطلاعات جهت بهبود الگوریتم و رسیدن به نتیجه مطلوب تلاش نمود.

فصل ششم

پیشنهادات

برای بهبود این پروژه میتوان از واحد گرافیکی متلب(gui) استفاده شود یکی دیگر از روش های بهبود پروژه میتوان اضافه کردن برنامه برای پلاک های رنگی (قرمز_سبز) است و همچنین قابلیت جستجو پلاک های ذخیره شده را میتوان استفاده کرد.

مـــراجـــــع

۱. کتاب پردازش تصویر عبدالرحمن

۲. کتاب پردازش تصویر گونزالس