



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

درس روش پژوهش
گزارش نوشتاری

بررسی الگوریتم‌های بینایی ماشین برای آشکارسازی لبه در تصاویر خاکستری

نگارش

امیرپارسا سلمان خواه

استاد راهنما

دکتر رضا صفابخش

خرداد ۱۴۰۱

پاس‌گزاری

از استاد گرامی جناب آقای دکتر رضا صفا بخش که درانتخاب و پیشبرد این پژوهش به عنوان استاد راهنما، کمکهای فراوانی به این جانب داشتند، کمال تشکر را دارم.

امیرارسلان خواه
خرداد ۱۴۰۱

چکیده

یکی از بنیادی ترین عملیات‌ها در الگوریتم‌های پردازش تصویر، آشکارسازی لبه است. با وجود اینکه آشکارسازی لبه و جداسازی مرز بین اجسام به کمک چشم انسان به راحتی صورت می‌گیرد، پیاده‌سازی این فرایند در کامپیوتر به این آسانی نیست و نیازمند الگوریتم‌هایی است که بتوانند در شرایط مختلف، بهترین عملکرد را از خود نشان دهند. برای تحقق این امر در کامپیوتر، الگوریتم‌های متعددی ابداع شده اند که از رویکردهای مختلفی برای انجام این کار استفاده می‌کنند؛ از جمله این رویکردها می‌توان به رویکردهای کلاسیک بینایی ماشین، یادگیری ماشین با نظارت، یادگیری تکاملی، منطق فازی و ... اشاره کرد. هدف این پژوهش، بررسی الگوریتم‌های مبتنی بر بینایی ماشین در تصاویر خاکستری است. در این راستا، در ابتدا به تعریف لبه و آشکارسازی لبه پرداخته و بعد از آن با برخی مفاهیم اولیه در این زمینه آشنا می‌شویم. سپس به بررسی شش الگوریتم آشکارسازی لبه پرداخته و نحوه عملکرد هر یک را بررسی می‌کنیم. در نهایت، این الگوریتم‌ها را با سه معیار ارزیابی دیداری، پیوستگی و زمان و با استفاده از سه تصویر معیار گوناگون مورد ارزیابی قرار می‌دهیم و بهترین الگوریتم را انتخاب می‌کنیم. در میان الگوریتم‌هایی که در این پژوهش برای بررسی انتخاب شدند، الگوریتم کنی^۱ توانست بهترین عملکرد را در بیشتر شرایط از خود نشان دهد.

واژه‌های کلیدی:

آشکارسازی لبه، بینایی ماشین، پردازش تصویر

^۱Canny

فهرست مطالب

صفحه

عنوان

۱	مقدمه	۱
۴	مفاهیم اولیه	۲
۵	۱-۲ نحوه عملکرد	۲
۵	۱-۱-۲ الگوریتم‌های مبتنی بر مشتق اول	۵
۵	۲-۱-۲ الگوریتم‌های مبتنی بر مشتق دوم	۵
۶	۳-۱-۲ سایر الگوریتم‌ها	۶
۶	۲-۲ حد آستانه	۲
۷	۱-۲-۲ حد آستانه محلی	۷
۷	۲-۲-۲ حد آستانه مطلق	۷
۷	۳-۲-۲ حد آستانه‌های محلی و مطلق	۷
۸	۳-۲ خلاصه	۸
۹	۳ الگوریتم‌های مورد بررسی	۹
۱۰	۱-۳ الگوریتم سوبل	۱۰
۱۱	۲-۳ الگوریتم پرویت	۱۱
۱۱	۳-۳ الگوریتم رابرت کراس	۱۱
۱۲	۴-۳ الگوریتم لاپلاس گاوسی	۱۲
۱۳	۱-۴-۳ بدون ترکیب کرنل‌ها	۱۳
۱۴	۲-۴-۳ با ترکیب کرنل‌ها	۱۴
۱۶	۵-۳ الگوریتم کنی	۱۶
۱۷	۶-۳ الگوریتم تابع دودویی	۱۷
۱۹	۷-۳ خلاصه	۱۹
۲۰	۴ ارزیابی الگوریتم‌ها	۲۰
۲۱	۱-۴ انواع معیارهای ارزیابی	۲۱
۲۱	۱-۱-۴ معیارهای دیداری	۲۱
۲۱	۲-۱-۴ معیارهای بدون مرجع	۲۱
۲۱	۱-۲-۱-۴ پیوستگی لبه‌ها	۲۱
۲۲	۲-۲-۱-۴ پیچیدگی زمانی و حافظه‌ای	۲۲
۲۲	۳-۱-۴ معیارهای با مرجع	۲۲
۲۳	۲-۴ معیارهای انتخاب شده و ارزیابی الگوریتم‌ها	۲۳
۲۴	۱-۲-۴ ارزیابی دیداری	۲۴

۲۶ ۲-۲-۴ ارزیابی پیوستگی و ضخامت لبه‌ها
۲۷ ۳-۲-۴ ارزیابی زمان اجرا
۲۸ ۳-۴ خلاصه
۳۰ ۵ نتیجه‌گیری و پیشنهادات
۳۱ ۱-۵ نتیجه‌گیری
۳۲ ۲-۵ پیشنهادات
۳۳ منابع و مراجع

شکل	فهرست اشکال	صفحه
۱-۲	مشتق اول تابع	۵
۲-۲	مشتق دوم تابع	۶
۱-۳	کرنل‌های الگوریتم سوپل	۱۰
۲-۳	کرنل‌های الگوریتم پرویت	۱۱
۳-۳	کرنل‌های الگوریتم رابرت کراس	۱۲
۴-۳	کرنل‌های تقریب زننده عملگر لاپلاس	۱۳
۵-۳	گام‌های کلی الگوریتم لاپلاس گاوسی بدون ترکیب کرنل‌ها	۱۳
۶-۳	نمود هندسی تابع لاپلاس گاوسی	۱۴
۷-۳	کرنل تقریب زننده لاپلاس گاوسی با اندازه 9×9 و با انحراف معیار ۱.۴	۱۵
۸-۳	گام‌های کلی الگوریتم لاپلاس گاوسی با ترکیب کرنل‌ها	۱۵
۹-۳	گام‌های کلی الگوریتم کنی	۱۷
۱۰-۳	ماسک‌های مورد استفاده در الگوریتم تابع دودویی	۱۸
۱۱-۳	گام‌های الگوریتم تابع دودویی به صورت کلی	۱۹
۱-۴	تصاویر معیار برای بررسی الگوریتم‌ها	۲۳
۲-۴	نتایج حاصل برای تصویر معیار اول	۲۴
۳-۴	نتایج حاصل برای تصویر معیار دوم	۲۵
۴-۴	نتایج حاصل برای تصویر معیار سوم	۲۶

صفحه	فهرست جداول	جدول
۲۷	نتایج به دست آمده از معیار پیوستگی	۱-۴
۲۸	زمان اجرای الگوریتمها	۲-۴

فصل اول

مقدمه

آشکارسازی لبه به فرایندی گفته می‌شود که در آن ناپیوستگی‌های ناگهانی در یک تصویر تشخیص داده شده و مکان‌یابی می‌شوند. این ناپیوستگی‌ها که به آن‌ها لبه گفته می‌شود، تغییراتی ناگهانی در شدت رنگ تصویر در یک سلول آن هستند که مرزهای اجسام درون تصویر را شکل می‌دهند [۱]. از کاربرد های آشکارسازی لبه می‌توان به کمک در تشخیص اثر انگشت، تشخیص تومور ها در تصاویر پزشکی، تشخیص وسایل نقلیه در دوربین های راهنمایی و رانندگی، استخراج ویژگی های مفید تصاویر در شبکه‌های عصبی پیچشی^۱ و موارد متعدد دیگری اشاره کرد.

مشکلی که در الگوریتم‌های تشخیص لبه کلاسیک در بینایی ماشین وجود دارد این است که در شرایط مختلف به شکل متفاوت عمل می‌کنند و می‌توانند عملکرد بدی در بسیاری از تصاویر داشته باشند. کیفیت عملکرد این الگوریتم‌ها به عواملی مانند نورپردازی تصویر، وجود اجسام هم‌رنگ در تصویر، وجود نویز^۲ در تصویر و ... بستگی داشته باشد. با وجود اینکه برای برطرف کردن این مشکلات برای هر الگوریتم پارامتر های قابل تنظیمی وجود دارد که به کمک آن‌ها می‌توان تمام یا بخشی از این مشکلات را برطرف کرد، تاکنون هیچ روش قابل اعتمادی برای تنظیم خودکار این پارامترها ابداع نشده است [۲].

متغیرهایی که می‌توانند در انتخاب الگوریتم مناسب برای یک تصویر تاثیر گذار باشند عبارت اند از:

۱- جهت لبه: الگوریتم‌های آشکارسازی لبه ممکن است به جهت خاصی از لبه بیشتر حساس باشند. لذا در صورت وجود لبه‌هایی با زوایای خاص در یک تصویر، می‌توان از الگوریتمی استفاده کرد که به آن زاویه بیشتر حساس است.

۲- وجود نویز در تصویر: آشکارسازی لبه در تصاویر نویزدار بسیار دشوار است زیرا لبه‌ها و نویز ها هر دو یک تغییر ناگهانی در تصویر هستند. از طرفی از بین بردن نویز ها و بلور کردن تصویر می‌تواند به لبه‌ها آسیب بزند و باعث نادقیق شدن مکان تشخیص لبه‌ها شود.

۳- ساختار لبه: همه لبه‌ها یک تغییر ناگهانی در تصویر نیستند. عواملی مانند بازتاب نور و تمرکز^۳ نامناسب تصویر می‌توانند باعث ایجاد عناصری شوند که مرز آن‌ها به شکل یک تغییر تدریجی ظاهر می‌شود. لذا در این شرایط، الگوریتم انتخابی باید به این نوع لبه‌ها نیز حساسیت داشته باشد [۱].

بنابراین، برای انجام عملیات آشکارسازی لبه بر روی یک تصویر باید ویژگی‌های آن تصویر را شناسایی کرده و با توجه به آن‌ها الگوریتمی را انتخاب کنیم که بهترین عملکرد را در آن شرایط داشته باشد. لازمه این کار این است که الگوریتم‌های مختلف بینایی ماشین را در حوزه آشکارسازی لبه شناسایی کنیم و با جزئیات، نحوه پیاده‌سازی و نقاط ضعف و قوت هر یک آشنا شویم. لذا در این پژوهش با این هدف به بررسی تعدادی از الگوریتم‌های آشکارسازی لبه در بینایی ماشین می‌پردازیم.

در فصل دوم این پژوهش به بررسی برخی مفاهیم اولیه که می‌توانند در یادگیری نحوه عملکرد الگوریتم‌ها

¹CNN

²Noise

³Focus

موثر باشند پرداخته و در ادامه، در فصل سوم به معرفی شش الگوریتم سوبل^۴، پرویت^۵، رابرت کراس^۶، لاپلاس گاوسی^۷، تابع دودویی^۸ و کنی می پردازیم. به دنبال آن، در فصل چهارم به بررسی انواع معیارهای ارزیابی استفاده شده در پژوهش های قبلی این حوزه پرداخته و الگوریتم ها را با سه معیار مد نظر خود یعنی معیار دیداری، معیار پیوستگی و معیار زمان مورد بررسی قرار داده و عملکرد هر یک را نسبت به این معیارها می سنجیم. در فصل پنجم نیز، نتیجه گیری نهایی از این پژوهش را می آوریم و پیشنهاداتی را برای بهبود فرایند آشکارسازی لبه ارائه می دهیم.

⁴Sobel

⁵Prewit

⁶Robert Cross

⁷Laplace of Gaussian

⁸Boolean Function

فصل دوم

مفاهیم اولیه

پیش از بررسی الگوریتم‌های آشکارسازی لبه لازم است تا برخی مفاهیم اولیه را در این زمینه مورد بررسی قرار دهیم.

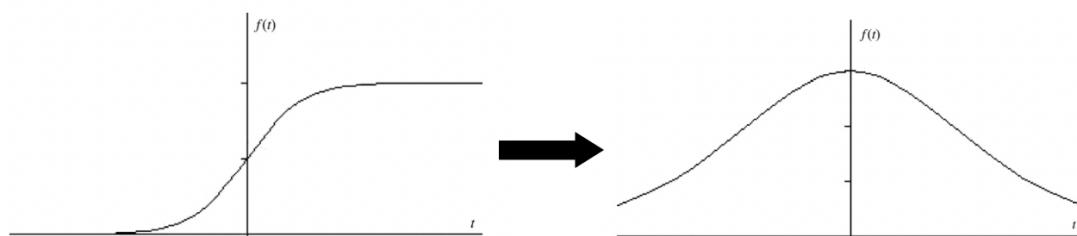
۱-۲ نحوه عملکرد

به طور کلی، الگوریتم‌های آشکارسازی لبه در بینایی ماشین بر اساس نحوه عملکرد به سه دسته تقسیم می‌شوند.

۱-۱-۲ الگوریتم‌های مبتنی بر مشتق اول^۱

در این دسته از الگوریتم‌ها، لبه‌ها با پیدا کردن نقاط بیشینه و کمینه در مشتق اول تابع تصویر تشخیص داده می‌شوند.

در شکل ۱-۲ مشتق اول تابع بر اساس متغیر t رسم شده است. همانطور که از شکل حاصل مشخص است، مقدار بیشینه تابع بعد از اعمال عملیات مشتق‌گیری در نقطه $t = 0$ اتفاق افتاده است. بنابراین از آن‌جا که یک تغییر ناگهانی در مقدار تابع در این نقطه وجود دارد، این نقطه می‌تواند به عنوان یک لبه شناخته شود.



شکل ۱-۲: مشتق اول تابع [۱]

۲-۱-۲ الگوریتم‌های مبتنی بر مشتق دوم^۲

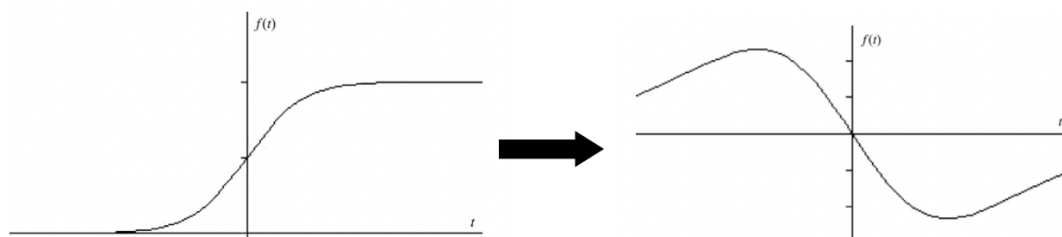
در این دسته از الگوریتم‌ها، لبه‌ها با پیدا کردن محل عبور^۳ مشتق دوم تابع تصویر از محور افقی تشخیص داده می‌شوند.

همانطور که از شکل ۲-۲ مشخص است، با دو بار اعمال عملیات مشتق‌گیری، تابع حاصل شده در نقطه $t = 0$ از محور افقی عبور کرده است. بنابراین در این نقطه یک لبه وجود دارد.

^۱Gradient-based

^۲Laplacian-based

^۳zero-crossing



شکل ۲-۲: مشتق دوم تابع [۱]

۳-۱-۲ سایر الگوریتم‌ها

الگوریتم‌های دیگری وجود دارند که رویکردی متفاوت نسبت به روش‌های قبلی دارند. به عنوان مثال، یکی از این الگوریتم‌ها، الگوریتم تابع دودویی است که با استفاده از تعدادی الگوی از پیش آماده شده، کار آشکارسازی لبه‌ها را انجام می‌دهد. بنابراین، بدون استفاده از مشتق اول و دوم نیز می‌توان به شناسایی لبه‌ها در تصویر پرداخت.

۲-۲ حد آستانه^۴

اکثر الگوریتم‌های آشکارسازی لبه از یک حد آستانه برای تشخیص لبه‌ها در تصویر استفاده می‌کنند. در واقع بعد از اعمال بخش اصلی الگوریتم، مقدار به دست آمده برای شدت رنگ در هر سلول^۵ تصویر با یک حد آستانه مقایسه می‌شود و اگر شدت رنگ آن سلول بیشتر از حد آستانه بود، آن سلول به عنوان مثال به یک سلول سفید و در غیر این صورت به یک سلول سیاه تبدیل می‌شود [۲].

انتخاب یک حد آستانه مناسب برای الگوریتم یکی از مهم‌ترین و سخت‌ترین فرایندها در الگوریتم‌های آشکارسازی لبه است. این کار در اکثر الگوریتم‌ها می‌بایست به شکل دستی و توسط برنامه‌نویس و با توجه به خروجی‌های مختلف الگوریتم نسبت به حد آستانه‌های متفاوت انجام شود. البته در بعضی روش‌ها این کار به صورت خودکار و توسط الگوریتم انجام می‌شود.

الگوریتم‌های آشکارسازی لبه می‌توانند به کمک یک حد آستانه محلی یا مطلق یا با استفاده از هر دو کار خود را انجام دهند.

⁴Threshold

⁵pixel

۱-۲-۲ حد آستانه محلی^۶

دسته‌ای از الگوریتم‌ها با تعریف یک حد آستانه برای هر سلول تصویر مشخص می‌کنند که آن سلول یک لبه است یا خیر. پیدا کردن این حد آستانه با توجه به ویژگی‌های سلول‌های اطراف آن سلول انجام می‌شود و به همین دلیل به آن حد آستانه محلی گفته می‌شود. حد آستانه محلی می‌تواند به اشکال مختلفی مانند میانگین یا میانه شدت رنگ‌های سلول‌های اطراف، میانگین شدت رنگ سلول‌ها با شدت رنگ بیشینه و کمینه، نصف طول بازه شدت رنگ سلول‌های اطراف و ... تعریف شود [۲].

۲-۲-۲ حد آستانه مطلق^۷

انتخاب حد آستانه در بعضی از الگوریتم‌ها با توجه ویژگی‌های کلی تصویر مانند میزان نویز موجود در تصویر انجام می‌پذیرد [۲]. از آنجا که این حد آستانه به شکل مطلق و برای کل تصویر انتخاب می‌شود، به آن حد آستانه مطلق گفته می‌شود. بعضی از الگوریتم‌ها تنها از یک حد آستانه مطلق ساده برای انجام کار خود استفاده می‌کنند. این در حالی است که برخی دیگر از الگوریتم‌ها با استفاده از روش پسماند^۸ در مورد لبه بودن یا نبودن یک سلول تصمیم‌گیری می‌کنند. در روش پسماند دو حد آستانه مطلق مشخص می‌شوند و با استفاده از قوانین زیر در مورد لبه بودن یا نبودن هر یک از سلول‌ها تصمیم‌گیری می‌شود.

- ۱- اگر مقدار سلول از حد آستانه بزرگ‌تر بیش‌تر باشد به عنوان یک لبه شناخته می‌شود.
- ۲- اگر مقدار سلول از حد آستانه کوچک‌تر کم‌تر باشد به عنوان لبه شناخته نمی‌شود.
- ۳- اگر مقدار سلول بین دو حد آستانه باشد و در اطراف آن یک سلول با مقدار بیش‌تر از حد آستانه بزرگ‌تر وجود داشته باشد به عنوان یک لبه شناخته می‌شود.
- ۴- اگر مقدار سلول بین دو حد آستانه باشد و در اطراف هیچ سلولی با مقدار بیش‌تر از حد آستانه بزرگ‌تر وجود نداشته باشد به عنوان یک لبه شناخته نمی‌شود [۳].

۳-۲-۲ حد آستانه‌های محلی و مطلق

دسته دیگری از الگوریتم‌ها از هر دو نوع حد آستانه برای انجام کار خود استفاده می‌کنند. حد آستانه‌های محلی و مطلق مکمل یکدیگر هستند. استفاده از حد آستانه مطلق به تنهایی منجر به

^۶Local^۷Global^۸Hysteresis

نتایج خوبی نخواهد شد. به منظور جلوگیری از عملکرد بد حد آستانه مطلق، می‌توان آن را به صورت ترکیبی با یک حد آستانه محلی استفاده کرد. همچنین، استفاده از حد آستانه محلی به تنهایی نیز می‌تواند به دلیل وجود نویز در عکس موجب تشخیص اشتباه لبه‌ها شود. در این شرایط استفاده از یک حد آستانه مطلق می‌تواند مشکل حساسیت به نویز را برطرف سازد [۲].

۳-۲ خلاصه

در این بخش با برخی مفاهیم اولیه که لازمه درک الگوریتم‌های آشکارسازی لبه هستند آشنا شدیم. در ابتدا متوجه شدیم که الگوریتم‌های آشکارسازی لبه می‌توانند با رویکردهایی مختلفی نظیر مشتق اول، مشتق دوم و ... به دنبال پیدا کردن لبه‌ها در تصویر باشند و سپس به بررسی نحوه استفاده الگوریتم‌ها از حد آستانه پرداختیم و با حد آستانه‌های محلی، مطلق و ترکیبی از این دو آشنایی پیدا کردیم.

فصل سوم

الگوریتم‌های مورد بررسی

در این بخش به بررسی الگوریتم‌های مختلف آشکارسازی لبه در تصاویر سیاه و سفید می‌پردازیم.

۱-۳ الگوریتم سوبل

الگوریتم سوبل به عنوان یکی از معروف‌ترین الگوریتم‌های آشکارسازی لبه در بینایی ماشین شمرده می‌شود.

در این الگوریتم از دو کرنل ۳ در ۳ برای تشخیص لبه‌ها استفاده می‌شود. این کرنل‌ها در شکل ۱-۳ مشخص هستند. کرنل سمت چپ مشتق اول تابع تصویر را نسبت به محور افقی تخمین می‌زند و به این ترتیب بیشترین حساسیت را نسبت به لبه‌های عمودی دارد. به همین ترتیب، کرنل سمت راست مشتق اول تابع تصویر را نسبت به محور عمودی تقریب می‌زند و بیشترین حساسیت را نسبت به لبه‌های افقی دارد [۱].

-1	0	+1
-2	0	+2
-1	0	+1

G_x

+1	+2	+1
0	0	0
-1	-2	-1

G_y

شکل ۱-۳: کرنل‌های الگوریتم سوبل [۱]

هم‌گشت این کرنل‌ها و تابع تصویر به شکل جداگانه محاسبه می‌شود و مقادیر به دست آمده از آن‌ها با یکدیگر ترکیب می‌شود تا اندازه مشتق در هر سلول به دست آید. این اندازه با استفاده از معادله ۱-۳ به دست می‌آید [۱].

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (۱-۳)$$

همچنین برای کاهش هزینه محاسباتی می‌توان از ۲-۳ برای محاسبه اندازه مشتق استفاده کرد [۱].

$$|G| = |G_x| + |G_y| \quad (۲-۳)$$

بعد از محاسبه اندازه لبه می‌توان از یک حد آستانه مطلق برای به دست آوردن تصویر نهایی استفاده

کرد. مقدار این حد آستانه را می‌توان با استفاده از آزمون و خطا و با توجه به کیفیت تصویر خروجی به دست آورد.

برای محاسبه جهت لبه نیز می‌توان از معادله ۳-۳ استفاده کرد [۱].

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (3-3)$$

۲-۳ الگوریتم پرویت^۱

همانند الگوریتم سوبل، این الگوریتم نیز از دو کرنل با جهت های افقی و عمودی برای تشخیص لبه‌ها استفاده می‌کند با این تفاوت که در این کرنل‌ها تاثیر همه سلول‌های اطراف به یک اندازه است و از تاثیر سلول‌های نزدیک‌تر به سلول مرکز کاسته شده است. این کرنل‌ها در شکل ۲-۳ آورده شده‌اند.

-1	0	+1
-1	0	+1
-1	0	+1

G_x

+1	+1	+1
0	0	0
-1	-1	-1

G_y

شکل ۲-۳: کرنل‌های الگوریتم پرویت [۱]

در این الگوریتم نیز همانند الگوریتم سوبل می‌توان از معادلات ۱-۳ و ۲-۳ برای تعیین اندازه لبه و از معادله ۳-۳ برای تشخیص جهت لبه استفاده کرد و با اعمال یک حد آستانه مطلق تصویر نهایی را به دست آورد.

در نهایت، همانند الگوریتم سوبل با استفاده از یک حد آستانه مطلق می‌توان تصویر نهایی را به دست آورد.

۳-۳ الگوریتم رابرت کراس

روش کار این الگوریتم شباهت زیادی با دو الگوریتم قبل دارد با این تفاوت که کرنل‌های مورد استفاده در این الگوریتم بیشترین حساسیت را در زوایای ۴۵ درجه نسبت به محور افقی و عمودی از خود نشان می‌دهند. این کرنل‌ها در شکل ۳-۳ مشخص هستند.

^۱Prewit

+1	0
0	-1

G_x

0	+1
-1	0

G_y

شکل ۳-۳: کرنل‌های الگوریتم رابرت کراس [۱]

در این الگوریتم نیز همانند دو الگوریتم قبل می‌توان از معادلات ۱-۳ و ۲-۳ برای تعیین اندازه لبه استفاده کرد. از آن جا که کرنل‌های این الگوریتم مشتق را در جهت نیمسازهای دستگاه مختصات محاسبه می‌کنند، برای محاسبه جهت لبه می‌توان از معادله ۳-۴ استفاده کرد [۱].

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) - \frac{3\pi}{4} \quad (۴-۳)$$

۴-۳ الگوریتم لاپلاس گاوسی

عملگر لاپلاس یک عملگر همسانگرد^۲ دو بعدی برای محاسبه مشتق دوم تابع تصویر است. لاپلاس یک تصویر نقطه‌ای که در آن تغییر ناگهانی در تابع تصویر اتفاق می‌افتد را مشخص می‌کند و از آن در بسیاری از کاربردها مانند آشکارسازی لبه استفاده می‌شود. اگر تابع تصویر به شکل $I(x, y)$ باشد، مقدار لاپلاس آن در هر نقطه با استفاده از معادله ۳-۵ محاسبه می‌شود.

$$\theta = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (۵-۳)$$

الگوریتم لاپلاس گاوسی را به دو طریق می‌توان اعمال کرد.

^۲isotropic

۳-۴-۱ بدون ترکیب کرنل‌ها

در این روش در ابتدا فیلتر بلور گاوسی^۳ به شکل جداگانه بر روی تابع تصویر اعمال می‌شود و سپس با استفاده از یک کرنل تقریب زننده، عملگر لاپلاس اعمال می‌شود. از آنجا که تصویر ورودی با استفاده از مجموعه‌ای از سلول‌های گسسته بازنمایی می‌شود باید به دنبال یک کرنل گسسته برای تقریب زدن مشتق دوم در تعریف عملگر لاپلاس باشیم. سه کرنل که به طور معمول برای این کار استفاده می‌شوند در شکل ۳-۴ آورده شده‌اند.

1	1	1
1	-8	1
1	1	1

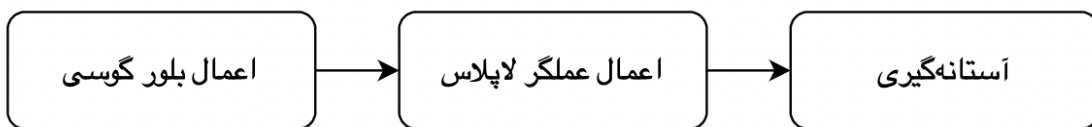
0	1	0
1	-4	1
0	1	0

-1	2	-1
2	-4	2
-1	2	-1

شکل ۳-۴: کرنل‌های تقریب‌زننده عملگر لاپلاس [۱]

از آنجا که این کرنل‌ها تقریبی از مشتق دوم را محاسبه می‌کنند بسیار به نویز حساس هستند. برای مقابله با این مشکل قبل از اعمال این کرنل‌ها یک فیلتر بلور گاوسی بر روی تصویر اعمال می‌شود. این گام پیش پردازشی نقاط دارای نویز با فرکانس بالا را کاهش می‌دهد و موجب صاف‌تر شدن عکس می‌شود. [۱]

بعد از اعمال دو مرحله قبل لازم است تا با پیمایش سلول‌های تصویر، محل‌هایی که در آن‌ها تغییر علامت اتفاق افتاده است را پیدا کرد و در صورتی که شدت این تغییر از یک حد آستانه‌ای بیشتر بود آن سلول را به عنوان یک سلول لبه مشخص کرد. همچنین می‌توان از روش پسماند که در بخش ۲-۲-۲ به آن اشاره شد برای تعیین لبه‌ها استفاده کرد. گام‌های کلی الگوریتم لاپلاس گاوسی بدون ترکیب کرنل‌ها در شکل ۳-۵ آورده شده‌اند.



شکل ۳-۵: گام‌های کلی الگوریتم لاپلاس گاوسی بدون ترکیب کرنل‌ها

^۳Gaussian Blur

۳-۴-۲ با ترکیب کرنل‌ها

از آن‌جا که عملیات هم‌گشت دارای خاصیت شرکت‌پذیری است، با ترکیب کرنل بلور گاوسی با کرنل تقریب‌زننده عملیات لاپلاس می‌توانیم یک کرنل ترکیبی جدید به دست آوریم و با استفاده از این کرنل جدید عملیات هم‌گشت را بر روی تصویر انجام دهیم. این روش دو مزیت دارد:

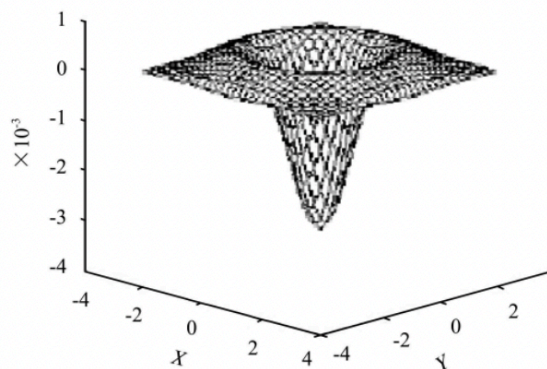
۱- به دلیل این که کرنل حاصل بسیار بزرگ‌تر از دو کرنل قبلی است و تنها لازم است یک عملیات هم‌گشت انجام شود، هزینه محاسباتی این روش کم‌تر است.

۲- کرنل ترکیبی می‌تواند در یک گام پیش‌پردازشی محاسبه شود و تنها عملیات هم‌گشت در زمان اجرای برنامه محاسبه شود.

تابع دو بعدی لاپلاس گاوسی با مرکزیت صفر و با انحراف معیار σ در معادله ۳-۶ مشخص شده است.

$$LoG(x, y) = \frac{-1}{\pi\sigma^4} \left[1 - \left(\frac{x^2 + y^2}{2\sigma^2} \right) \right] \exp \frac{x^2 + y^2}{2\sigma^2} \quad (۳-۶)$$

نمود هندسی تابع لاپلاس گاوسی در شکل ۳-۶ آورده شده است. همانطور که مشخص است شکل این تابع مانند یک کلاه مکزیکی است؛ به همین دلیل این عملگر را "عملگر کلاه مکزیکی" نیز می‌نامند [۳].



شکل ۳-۶: نمود هندسی تابع لاپلاس گاوسی [۱]

همانطور که در بخش ۳-۴-۱ عنوان شد، تصاویر به وسیله سلول‌های گسسته بازنمایی می‌شوند. لذا لازم است تا این تابع را نیز با استفاده از یک کرنل گسسته تقریب بزنیم. برای به دست آوردن یک کرنل $n \times n$ متناظر با تابع لاپلاس گاوسی می‌توان از معادله ۳-۷ استفاده کرد که در آن L_{ij} درایه‌ای از کرنل

است که در ردیف i ام و ستون j ام قرار دارد.

$$L_{ij} = LoG(i - \lfloor \frac{n-1}{2} \rfloor, j - \lfloor \frac{n-1}{2} \rfloor) \quad (۷-۳)$$

همچنین مقدار مناسب σ برای یک اندازه خاص کرنل را می‌توان از معادله ۸-۳ به دست آورد.

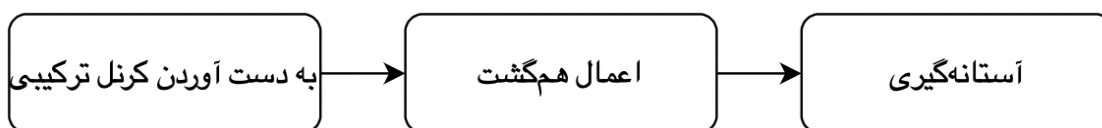
$$\sigma = \frac{n-1}{6} \quad (۸-۳)$$

به عنوان مثال یک کرنل گسسته 9×9 با میزان انحراف معیار ۱.۴ در شکل ۷-۳ آورده شده است. لازم به ذکر است که مقدار سلول وسط این کرنل بر روی عدد ۴۰- مقیاس شده است و سایر سلول‌ها نیز به همان نسبت مقیاس شده‌اند.

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

شکل ۷-۳: کرنل تقریب‌زننده لاپلاس گاوسی با اندازه 9×9 و با انحراف معیار ۱.۴ [۱]

بنابراین با توجه به نکات ذکر شده، با یک بار هم‌گشت این کرنل با تابع تصویر می‌توان نتیجه حاصل از اعمال بلور گاوسی و عملگر لاپلاس را بر روی تصویر به شکل تقریبی به دست آورد. در نهایت، بعد از اعمال این کرنل می‌بایست از روش‌های آستانه‌گیری ذکر شده در بخش ۳-۴-۱ استفاده کرد تا تصویر نهایی را به دست آورد. گام‌های کلی الگوریتم لاپلاس گاوسی با استفاده از ترکیب کرنل‌ها در شکل ۸-۳ آورده شده‌اند.



شکل ۸-۳: گام‌های کلی الگوریتم لاپلاس گاوسی با ترکیب کرنل‌ها

۳-۵ الگوریتم کنی^۴

الگوریتم آشکارسازی لبه کنی به عنوان یک الگوریتم استاندارد در صنعت شناخته می‌شود. این الگوریتم توسط جان کنی^۵ در سال ۱۹۸۳ ابداع شد و همچنان عملکرد بهتری از بسیاری از الگوریتم‌های ارائه شده بعد از آن از خود نشان می‌دهد. کنی مسئله‌ی تشخیص لبه را همانند یک مسئله بهینه‌سازی پردازش سیگنال می‌دید؛ در نتیجه یک تابع هدف برای رسیدن به جواب این مسئله پیدا کرد. جواب کنی به این مسئله یک تابع نمایشی پیچیده بود اما کنی راه‌های زیادی برای تقریب‌زدن و بهینه‌سازی آن برای مسئله آشکارسازی لبه پیدا کرد [۲].

تمرکز کنی در آن زمان بیشتر بر روی بهبود الگوریتم‌هایی بود که تا آن زمان برای آشکارسازی لبه ارائه شده بودند. برای تحقق این امر، کنی سه شرط را ضروری می‌دانست: [۱]

۱- اولین و مهم‌ترین شرط، نرخ خطای پایین بود. برای کنی بسیار مهم بود که لبه‌های موجود در تصویر تا حد امکان از دست نروند و همچنین سلول‌هایی که لبه نیستند به اشتباه به عنوان لبه شناخته نشوند.

۲- محلیت لبه‌ها شرط دیگری بود که برای کنی اهمیت داشت. این شرط به این معنی است که فاصله سلول‌های لبه پیدا شده توسط الگوریتم و لبه‌های واقعی باید کمینه باشد.

۳- شرط پایانی این بود که الگوریتم باید به هر لبه تنها یک بار پاسخ دهد. این شرط به این دلیل اضافه شد که دو شرط قبل به تنهایی برای این امر کافی نبودند و امکان داشت که الگوریتم پاسخ‌های متعددی به یک لبه بدهد.

فرض کنید تابع تصویر به شکل $f(x, y)$ باشد. الگوریتم کنی شامل مراحل زیر است:

۱- یک کرنل بلور گاوسی روی تصویر اعمال می‌شود تا نویزها و جزئیات ناخواسته تصویر از بین بروند [۱].

$$g(m, n) = g_{\sigma}(m, n) * f(m, n) \quad (۹-۳)$$

۲- سپس از یکی از الگوریتم‌های مبتنی بر مشتق اول مانند سوبل و پرویت برای به دست آوردن اندازه و جهت لبه در هر سلول استفاده می‌شود [۱].

$$M(m, n) = \sqrt{g_m^2(m, n) + g_n^2(m, n)} \quad (۱۰-۳)$$

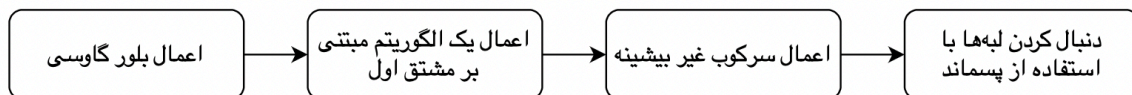
^۴Canny

^۵John Canny

۳- لبه‌ها در سلول‌هایی اتفاق می‌افتند که اندازه مشتق در آن نقطه مقدار بیشینه دارد. بنابراین تمامی سلول‌های غیر بیشینه باید از بین بروند. به این منظور اندازه و جهت مشتق در هر سلول در نظر گرفته می‌شود و مقدار سلول با همسایه‌های آن سلول که در راستای لبه قرار دارند مقایسه می‌شود و اگر مقدار آن از هر دوی آن‌ها بیشتر نبود، مقدار آن صفر می‌شود و به عنوان لبه در نظر گرفته نمی‌شود. در غیر این صورت مقدار سلول حفظ می‌شود [۳]. این فرایند "سرکوب غیر بیشینه" نامیده می‌شود.

۴- با در نظر گرفتن دو حد آستانه مطلق و با استفاده از روش پسماند که در بخش ۲-۲-۲ به آن اشاره شد، تعیین می‌شود که مقادیر بین دو مقدار آستانه به عنوان لبه در نظر گرفته شوند یا خیر [۳].

عملکرد الگوریتم کنی وابستگی زیادی به متغیرهای قابل تنظیم آن دارد. یکی از این متغیرها σ است که میزان انحراف معیار بلور گاوسی را مشخص می‌کند. همچنین این متغیر اندازه کرنل گاوسی را کنترل می‌کند و هرچه بیشتر باشد، اندازه کرنل نیز بزرگتر می‌شود و میزان مات شدن تصویر بیشتر می‌شود. با بیشتر مات شدن تصویر انتظار می‌رود که میزان دقت محلیت لبه‌ها کمتر شود و فاصله لبه‌های تشخیص داده شده از لبه‌های واقعی بیشتر شود. همچنین اگر مقدار این متغیر کم باشد، میزان مات شدن تصویر کم می‌شود و لبه‌ها کمتر آسیب می‌بینند. متغیرهای دیگر دو حد آستانه‌ی مورد استفاده در فرایند پسماند هستند که باید با توجه به ویژگی‌های تصویر توسط برنامه‌نویس انتخاب شوند. گام‌های الگوریتم کنی به طور خلاصه در شکل ۳-۹ مشخص شده‌اند.



شکل ۳-۹: گام‌های کلی الگوریتم کنی

۳-۶ الگوریتم تابع دودویی

این الگوریتم با بسیاری از الگوریتم‌های نوین دیگر که برگرفته از الگوریتم کنی هستند متفاوت است و متکی به مشتق یا بلور گاوسی نیست و از هر دو نوع حد آستانه مطلق و محلی برای پیدا کردن لبه‌ها استفاده می‌کند. برخلاف سایر الگوریتم‌های آشکارسازی لبه، این الگوریتم یک پنجره از سلول‌ها را به یک الگوریتم دودویی بر اساس حد آستانه محلی تبدیل می‌کند و بعد از آن ماسک‌هایی را بر روی آن اعمال می‌کند تا تشخیص دهد که آیا در نقطه مرکزی پنجره لبه‌ای وجود دارد یا خیر. گام‌های این الگوریتم به شرح زیر هستند:

۱- در ابتدا باید یک حد آستانه محلی به یک پنجره 3×3 از تصویر اعمال شود. از آن‌جا که بازه شدت رنگ تصویر در یک پنجره 3×3 بسیار کوچک است و مقادیر شدت رنگ سلول‌های موجود

در پنجره به سلول وسط نزدیک هستند، در نظر گرفتن میانگین شدت رنگ سلول‌های پنجره به تنهایی مناسب نیست. نحوه تعریف این حد آستانه در معادله ۳-۱۱ مشخص شده است که C در آن یک ثابت است که باید توسط برنامه‌نویس تعیین شود و M در آن میانگین شدت رنگ سلول‌های موجود در پنجره است که از معادله ۳-۱۲ به دست می‌آید [۲].

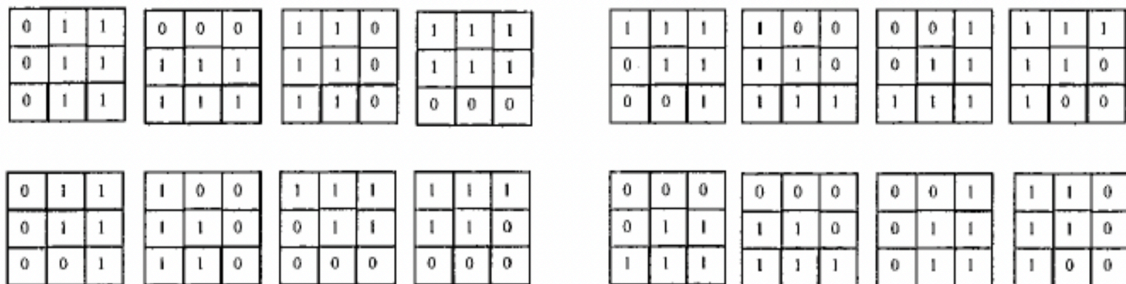
$$T = M - C \quad (۳-۱۱)$$

$$M = \frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j) \quad (۳-۱۲)$$

سپس مقدار شدت رنگ تصویر در هر یک از سلول‌های موجود در پنجره با حد آستانه به دست آمده مقایسه می‌شود. اگر مقدار شدت رنگ سلول از آن حد آستانه بیشتر باشد به آن سلول عدد یک و دیر غیر این صورت عدد صفر مقداردهی می‌شود و یک الگوی دودویی با اندازه 3×3 به دست می‌آید [۲].

۲- الگوی دودویی به دست آمده با شانزده ماسک که از قبل آماده شده‌اند مقایسه می‌شود و اگر هر یک از ماسک‌ها منطبق با الگوی به دست آمده باشد، سلول مرکز پنجره به عنوان لبه شناخته می‌شود.

ماسک‌های مورد استفاده در این الگوریتم در شکل ۳-۱۰ آورده شده‌اند.



شکل ۳-۱۰: ماسک‌های مورد استفاده در الگوریتم تابع دودویی [۲]

۳- مراحل ۱ و ۲ برای تمامی پنجره‌های 3×3 موجود در تصویر انجام می‌شود تا برای هر یک از مراکز پنجره‌ها در مورد لبه بودن یا نبودن تصمیم‌گیری شود. بعد از اتمام این مرحله بعضی از لبه‌ها بر اثر نویز به اشتباه به عنوان لبه شناخته شده‌اند. این مشکل در مرحله بعد برطرف می‌شود [۱].

۴- در این مرحله لبه‌های اشتباه به وسیله یک حد آستانه مطلق از بین می‌روند. برای این کار از یک حد آستانه مطلق با نام T_n استفاده می‌شود که باید بر اساس میزان نویز موجود در تصویر انتخاب شود.

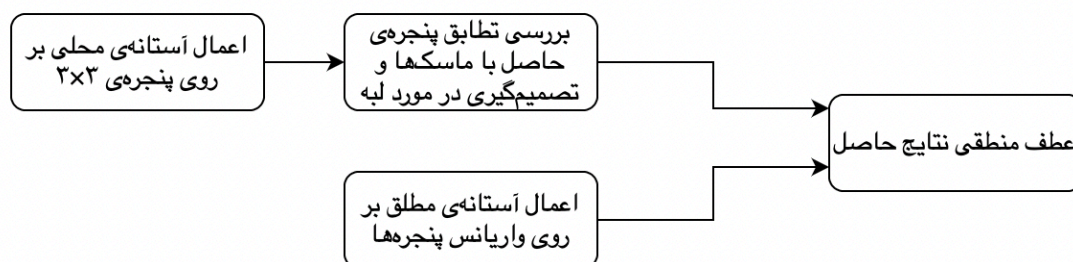
در یک لبه، تابع واریانس مقدار بیشینه خود را اختیار می‌کند. بنابراین، واریانس هر پنجره 3×3 به شکل محلی محاسبه می‌شود و بر اساس یک حد آستانه مطلق برای لبه بودن یا نبودن سلول مرکزی تصمیم‌گیری می‌شود.

واریانس هر پنجره از معادله $3-13$ به دست می‌آید که در آن $I(x, y)$ تابع شدت رنگ تصویر و $mean_{i,j}$ میانگین شدت رنگ سلول‌ها در یک پنجره 3×3 با مرکزیت سلول (i, j) می‌باشد. مقدار به دست آمده از این معادله با استفاده از T_n آستانه‌گیری می‌شود و تصویر نهایی حاصل از این مرحله به دست می‌آید [۲].

$$\sigma_{x,y}^2 = \frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [I(i, j) - mean_{i,j}]^2 \quad (13-3)$$

۵- در نهایت، تصاویر حاصل از مراحل ۳ و ۴ با یکدیگر و با استفاده از عملگر عطف منطقی ترکیب می‌شوند تا تصویر نهایی به دست آید [۲].

گام‌های الگوریتم تابع دودویی به صورت کلی در شکل ۳-۱۱ مشخص هستند.



شکل ۳-۱۱: گام‌های الگوریتم تابع دودویی به صورت کلی

۷-۳ خلاصه

در این بخش با الگوریتم‌های مورد بررسی در این پژوهش آشنا شدیم. در ابتدا چند الگوریتم کلاسیک بینایی ماشین نظیر سوبل، پرویت، رابرت کراس و لاپلاس گاوسی را مورد مطالعه قرار دادیم و در ادامه با الگوریتم‌های پیچیده تری مانند تابع دودویی و کنی آشنا شدیم. هر یک از این الگوریتم‌ها به طریقی کار آشکارسازی لبه را انجام می‌دهند و در محیط‌های مختلف به اشکال مختلف عمل می‌کنند و جزئیات پیاده سازی آن‌ها در زمینه‌های مختلفی نظیر نحوه پیدا کردن لبه‌ها و چگونگی استفاده از حد آستانه متفاوت است. درک دقیق این جزئیات لازمه پیاده سازی صحیح این الگوریتم‌ها به منظور ارزیابی آن‌هاست.

فصل چهارم

ارزیابی الگوریتم‌ها

در این بخش در ابتدا به معرفی معیارهای مختلف ارزیابی الگوریتم‌های آشکارسازی لبه پرداخته و بعد از آن معیارهای مورد نظر خود را در این پژوهش انتخاب می‌کنیم و سپس به مقایسه و ارزیابی الگوریتم‌های بررسی شده می‌پردازیم.

۴-۱ انواع معیارهای ارزیابی

بر اساس یک پژوهش که بر روی ۷۳ مقاله مربوط به حوزه آشکارسازی لبه انجام شده است، هیچ معیار استاندارد برای ارزیابی این الگوریتم‌ها وجود ندارد و پژوهشگران از معیارهای ارزیابی مختلفی در پژوهش خود استفاده می‌کنند. اما به طور کلی این معیارها را می‌توان به ۳ دسته معیارهای دیداری، معیارهای ارزیابی بدون مرجع و معیارهای ارزیابی با مرجع تقسیم کرد [۴].

۴-۱-۱ معیارهای دیداری

در این دسته از معیارها، عملکرد الگوریتم توسط نظر شخصی انسان و از طریق چشم او ارزیابی می‌شود. بنابراین در این معیارها نیازی به تصاویر مرجع و از پیش آماده که در آن‌ها تمامی لبه‌های تصویر مشخص شده اند، وجود ندارد [۴].

۴-۱-۲ معیارهای بدون مرجع

این دسته از معیارها نیز نیازی به تصاویر مرجع و از پیش آماده ندارند بلکه تنها از اطلاعاتی که از نتایج الگوریتم‌ها به دست آمده است، استفاده می‌کنند [۴]. در ادامه به بررسی دو معیار ارزیابی مهم از این دسته می‌پردازیم.

۴-۱-۲-۱ پیوستگی لبه‌ها

یک روش محاسبه میزان پیوستگی لبه‌ها از طریق معادله ۴-۱ است که در آن TEN تعداد عناصر لبه و CEN تعداد قطعه‌های لبه می‌باشد.

$$R = \frac{TEN}{CEN} \quad (۴-۱)$$

هر چه مقدار این کسر بیشتر باشد، نشانگر این است که قطعات لبه طولانی‌تر و در نتیجه پیوسته‌تر هستند.

یک روش دیگر برای محاسبه پیوستگی استفاده از معادلات ۴-۲ و ۴-۳ است که در آن‌ها A برابر با تعداد عناصر لبه، B برابر با تعداد قطعات لبه با تحلیل مولفه‌های ۴تایی به هم متصل و C برابر با تعداد قطعات

لبه با تحلیل مولفه‌های ۸ تایی به هم متصل است. هر چه این کسر ها کوچکتر باشند، لبه‌ها به ترتیب پیوسته‌تر و نازک‌تر خواهند بود [۴].

$$M = \frac{C}{A} \quad (۲-۴)$$

$$N = \frac{C}{B} \quad (۳-۴)$$

۲-۲-۱-۴ پیچیدگی زمانی و حافظه‌ای

زمان پردازش یک معیار برای اندازه‌گیری زمانی است که یک الگوریتم آشکارسازی لبه نیاز دارد تا فرایند آشکارسازی لبه را به طور کامل انجام دهد. این زمان به طور معمول با واحد های ثانیه یا میلی ثانیه گزارش می‌شود. این زمان پردازش بهتر است تا با انواع مختلف ورودی گزارش شود تا عملکرد الگوریتم در حالت های مختلف سنجیده شود [۴].

در کنار زمان پردازش، میزان استفاده الگوریتم از حافظه نیز معیار مهم دیگری است که معمولاً واحد مگابایت گزارش می‌شود. هر چه این میزان بیشتر باشد به این معنی است که الگوریتم پیچیده‌تر است و فضای بیشتری از حافظه را اشغال می‌کند [۴].

۳-۱-۴ معیارهای با مرجع

بر خلاف دو دسته قبل، این دسته از معیارها نیازمند به تصاویر مرجع یا مجموعه تصاویر آموزشی و تست هستند. تعدادی از این معیارها عبارتند از [۴]:

۱- تعداد عناصر لبه تشخیص داده شده

۲- معیارهای مبتنی بر ماتریس در هم‌ریختگی^۱

۳- تعداد لبه‌های اشتباه

۴- اشتراک بر روی اجتماع^۲

۵- خطای تشخیص لبه

۶- بی‌نظمی^۳

^۱Confusion Matrix

^۲IoU

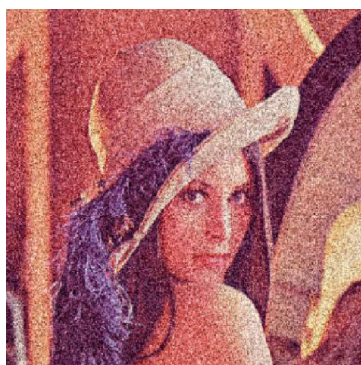
^۳Entropy

معیارهای بامرجع بیشتر در پژوهش‌هایی استفاده شده اند که از شبکه‌های عصبی و روش‌های خودکار دیگر برای انجام عملیات آشکارسازی لبه استفاده شده است زیرا استفاده از آن‌ها در روش‌هایی که پارامترهای الگوریتم به شکل دستی تنظیم می‌شوند بسیار زمان بر است و پژوهشگر باید تمامی تصاویر را به شکل دستی آماده و با نتایج موجود در تصاویر مرجع مقایسه کند [۴].

۲-۴ معیارهای انتخاب شده و ارزیابی الگوریتم‌ها

از آنجا که الگوریتم‌های مطرح شده در این پژوهش نیازمند به تنظیم دستی پارامترها هستند، استفاده از معیارهای با مرجع برای ارزیابی آن‌ها گزینه مناسبی به نظر نمی‌رسد. لذا در این پژوهش برای ارزیابی الگوریتم‌ها از سه روش ارزیابی دیداری، ارزیابی پیوستگی لبه‌ها و ارزیابی زمان اجرای الگوریتم‌ها استفاده می‌کنیم.

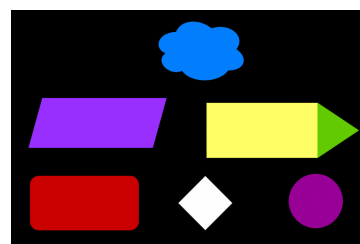
به منظور بررسی قدرت الگوریتم‌ها از سه تصویر آسان، متوسط و سخت استفاده می‌شود. این سه تصویر در شکل ۴-۱ آورده شده‌اند. به کمک تصویر الف می‌توانیم به بررسی عملکرد الگوریتم‌ها در گوشه‌ها و با زوایای مختلف لبه پردازیم. تصویر ب یک معیار استاندارد^۴ برای بررسی عملکرد الگوریتم‌های آشکارسازی لبه است که در بسیاری از پژوهش‌ها از آن استفاده شده است. تصویر ج نیز حالت نویز دار تصویر ب است که با آن عملکرد الگوریتم‌ها در شرایطی که تصویر دارای نویز بسیاری است سنجیده می‌شود.



(ج) تصویر سخت



(ب) تصویر متوسط



(آ) تصویر آسان

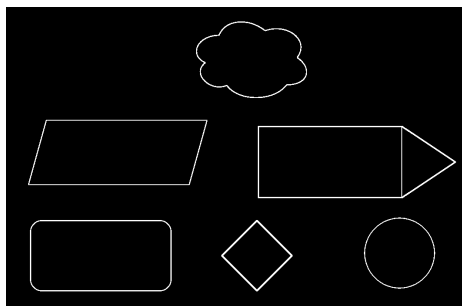
شکل ۴-۱: تصاویر معیار برای بررسی الگوریتم‌ها

نتایج حاصل از اعمال الگوریتم‌ها بر روی معیار اول در شکل ۴-۲، برای معیار دوم در شکل ۴-۳ و برای معیار سوم در شکل ۴-۴ آورده شده‌اند. در هر یک از تصاویر سعی شده است تا بهترین عملکرد الگوریتم‌ها در آن تصویر نشان داده شود. این نتایج با پیاده‌سازی مجدد این الگوریتم‌ها از طریق کتابخانه اوپن سی وی^۵ در زبان پایتون^۶ به دست آمده‌اند.

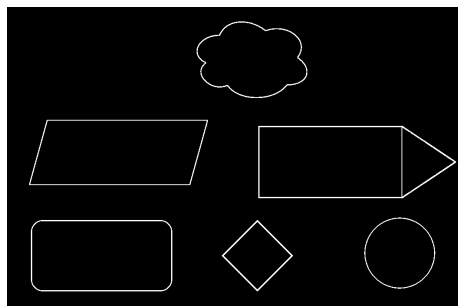
^۴Lena Benchmark

^۵OpenCV

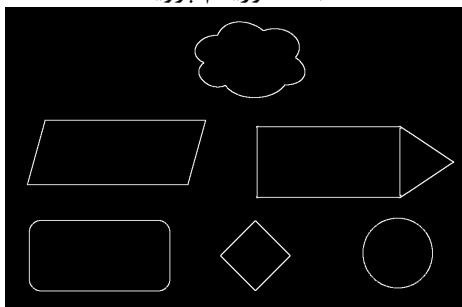
^۶Python



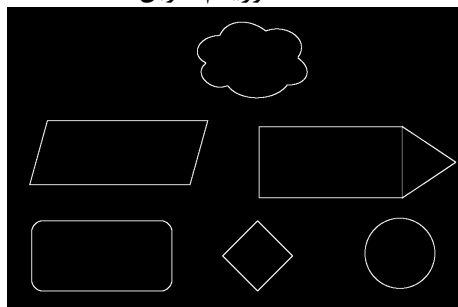
(ب) الگوریتم پرویت



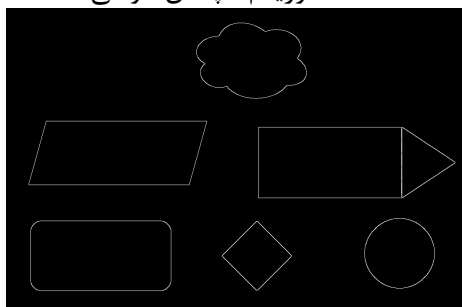
(آ) الگوریتم سوبل



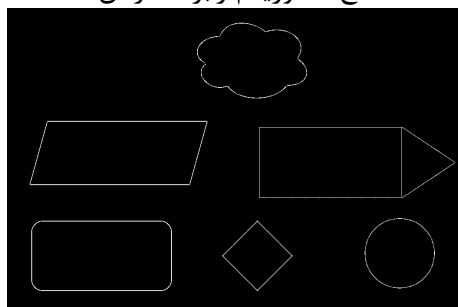
(د) الگوریتم لاپلاس گوسی



(ج) الگوریتم رابرت کراس



(و) الگوریتم کنی



(ه) الگوریتم تابع دودویی

شکل ۴-۲: نتایج حاصل برای تصویر معیار اول

۱-۲-۴ ارزیابی دیداری

با توجه به شکل ۴-۲ دو الگوریتم سوبل و پرویت عملکردی مشابه به هم داشته‌اند و در تشخیص گوشه‌ها و لبه‌ها با زاویه‌های متفاوت عملکرد خوبی داشته‌اند. تنها مشکل این الگوریتم‌ها ایجاد لبه‌های ضخیم‌تر در مقایسه با سایر الگوریتم‌ها است.

الگوریتم رابرت کراس نیز در تشخیص گوشه‌ها و لبه‌ها بسیار خوب عمل کرده است و همچنین لبه‌های نازک تری پیدا کرده است.

از آن‌جا که الگوریتم لاپلاس گاوسی از بلور گاوسی استفاده کرده است، در تشخیص بعضی گوشه‌ها دچار مشکل شده است اما در تشخیص سایر لبه‌ها عملکرد خوبی داشته است.

الگوریتم تابع دودویی در تشخیص همه لبه‌ها با زوایای مختلف موفق نبوده است زیرا بعضی از آن‌ها توسط ۱۶ الگوی ذکر شده برای این الگوریتم شناسایی نمی‌شوند. در مقابل، این الگوریتم از جهت ایجاد لبه‌های نازک نسبت به الگوریتم‌های دیگر عملکرد خوبی داشته است.



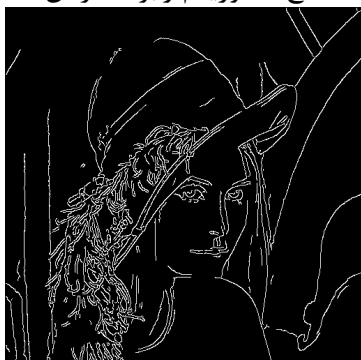
(ج) الگوریتم رابرت کراس



(ب) الگوریتم پرویت



(آ) الگوریتم سوبل



(و) الگوریتم کنی



(ه) الگوریتم تابع دودویی

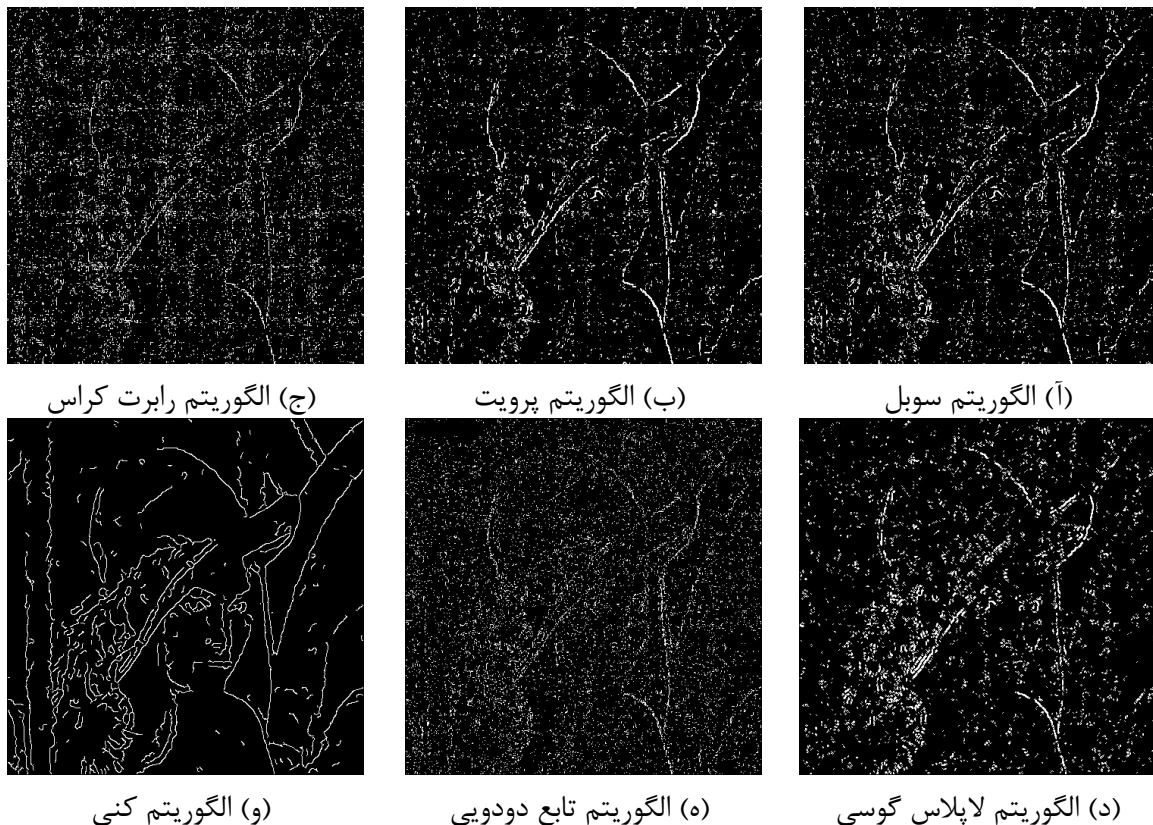


(د) الگوریتم لاپلاس گاوسی

شکل ۳-۴: نتایج حاصل برای تصویر معیار دوم

الگوریتم کنی نیز از آنجا که از الگوریتم سوبل برای انجام کار خود استفاده می‌کند عملکرد خوبی داشته است اما به دلیل اعمال فرایند سرکوب بیشینه، لبه‌های نازک تر و تیزتری را به دست آورده است. بنابراین به طور کلی برای تصویر معیار اول می‌توان گفت که الگوریتم‌های کنی و سه الگوریتم سوبل، پرویت و رابرت کراس عملکرد بهتری را نسبت به دو الگوریتم دیگر داشته‌اند. با توجه به شکل ۳-۴ سه الگوریتم سوبل، پرویت و رابرت کراس همانند قبل، عملکرد مشابهی داشته‌اند اما با توجه به کوچکتر بودن کرنل الگوریتم رابرت کراس، این الگوریتم حساسیت بیشتری به نویز و لبه‌های غیر مفید از خود نشان داده است و در بعضی نقاط تصویر لبه‌های ناپیوسته و کوچکی ایجاد کرده است. اما با این حال، این الگوریتم لبه‌های نازک‌تری نسبت به دو الگوریتم دیگر ایجاد کرده است. الگوریتم لاپلاس گاوسی در این تصویر عملکرد بدی داشته است و لبه‌هایی بسیار ضخیم ایجاد کرده است.

دو الگوریتم تابع دودویی و کنی در این تصویر بهترین عملکرد را داشته‌اند و لبه‌های نازک و مفیدی ایجاد کرده‌اند. الگوریتم کنی با اعمال روش پسماند و دنبال کردن لبه‌ها توانسته است لبه‌هایی پیوسته ایجاد کند در حالی که الگوریتم تابع دودویی از انجام آن ناتوان بوده است. بنابراین می‌توان گفت که عملکرد کنی بهترین عملکرد را در این تصویر داشته است و بعد از آن به ترتیب الگوریتم‌های تابع دودویی، سوبل، پرویت، رابرت کراس و لاپلاس گاوسی قرار دارند. با توجه به شکل ۴-۴ دو الگوریتم رابرت کراس و تابع دودویی بیشترین حساسیت را نسبت به نویز از



شکل ۴-۴: نتایج حاصل برای تصویر معیار سوم

خود نشان داده‌اند و شخص درون تصویر کاملاً در آن‌ها محو شده است. دو الگوریتم سوبل و پرویت همانند دو تصویر قبل عملکرد مشابهی داشته‌اند و هر دو به نویز حساسیت نشان داده‌اند و به نتیجه مناسبی نرسیده‌اند. الگوریتم لاپلاس گاوسی نیز با وجود استفاده از بلور گاوسی عملکرد خوبی نداشته است و علاوه بر عدم برطرف کردن نویز، در بسیاری از نقاط حتی لبه‌های مفید را شناسایی نکرده است. الگوریتم کنی در این تصویر عملکرد بی‌نظیری داشته است و با استفاده از بلور گاوسی و سرکوب غیربیشینه موفق به حذف بخش زیادی از نویزها شده است و با این حال توانسته است شمار زیادی از لبه‌های مفید را نیز تشخیص دهد. بنابراین در این تصویر نیز الگوریتم کنی بهترین نتیجه را برگردانده است و بعد از آن به ترتیب الگوریتم‌های سوبل، پرویت، لاپلاس گوسی، رابرت کراس و تابع دودویی قرار دارند.

۲-۲-۴ ارزیابی پیوستگی و ضخامت لبه‌ها

برای بررسی میزان پیوستگی و ضخامت لبه‌ها، برای هر الگوریتم مقدار کسرهای موجود در معادلات ۲-۴ و ۳-۴ محاسبه شد. نتایج به دست آمده در جدول ۴-۱ موجود است. همانطور که در بخش ۴-۱-۲-۱ به این موضوع اشاره شد، هر چه مقدار این کسر ها کوچک‌تر باشد،

الگوریتم لبه‌های پیوسته‌تر و نازک‌تری را ایجاد می‌کند. بنابراین، با توجه به جدول می‌توان دریافت که دو الگوریتم تابع دودویی و کنی در دو تصویر اول عملکرد بسیار خوبی از خود نشان داده‌اند و اعداد به دست آمده در جدول با ارزیابی دیداری نیز مطابقت دارند. اما، در تصویر معیار دوم، دو الگوریتم کنی و لاپلاس گاوسی مقادیر کوچک‌تری به دست آورده‌اند و براساس این معیار عملکرد بهتری داشته‌اند. کم بودن مقادیر برای الگوریتم لاپلاس گاوسی را می‌توان ناشی از تعداد بالای لبه‌های شناخته شده و ضخیم بودن آن‌ها دانست. در مقابل، الگوریتم کنی توانسته است با تعداد معقولی لبه به اعداد مناسبی نیز برسد و مطابق با ارزیابی دیداری، بهترین عملکرد را در این تصویر داشته است.

جدول ۴-۱: نتایج به دست آمده از معیار پیوستگی

تصویر معیار	الگوریتم	A	B	C	C/A	C/B
تصویر ۴-۱	سوبل	۱۵۵۳۹	۶	۶	۰.۰۰۰۴۴	۱
	پرویت	۱۶۳۹۶	۶	۶	۰.۰۰۰۴۲	۱
	رابرت کراس	۱۱۰۰۴	۶	۶	۰.۰۰۰۶۰	۱
	لاپلاس گاوسی	۱۱۱۸۴	۱۰۵	۶	۰.۰۰۰۵۳	۰.۰۵۷۱۴
	تابع دودویی	۷۳۸۴	۱۰۸۴	۱۰۹	۰.۰۱۴۷۶	۰.۱۰۰۵۵
	کنی	۶۰۸۶	۱۷۵	۶	۰.۰۰۱۰۶	۰.۰۳۴۲۸
تصویر ۴-۱ ب	سوبل	۲۷۳۴۴	۹۹۵	۴۷۰	۰.۰۱۷۱۸	۰.۴۷۲۳۶
	پرویت	۲۷۹۷۱	۸۶۲	۴۴۸	۰.۰۱۶۰۱	۰.۵۱۹۷۲
	رابرت کراس	۲۲۱۳۳	۱۶۵۱	۷۶۹	۰.۰۳۳۷۴	۰.۴۶۵۷۷
	لاپلاس گاوسی	۳۳۷۲۵	۱۵۱۰	۱۴۹	۰.۰۰۴۴۱	۰.۰۹۸۶۷
	تابع دودویی	۲۰۹۰۸	۴۱۰۲	۲۴۴۳	۰.۲۶۸۱۶	۰.۵۹۵۵۶
	کنی	۱۵۵۴۵	۱۵۶۹	۲۵۶	۰.۲۶۸۱۶	۰.۱۶۳۱۶
تصویر ۴-۱ ج	سوبل	۴۳۵۲	۳۶۰	۳۵۹۳	۰.۲۶۴۷۷	۰.۸۲۵۵۹
	پرویت	۱۲۸۴۶	۳۹۰۷	۳۳۹۴	۰.۲۶۴۲۰	۰.۸۶۸۶۹
	رابرت کراس	۱۵۹۶۰	۷۰۶۷	۵۶۷۷	۰.۳۵۵۷۰	۰.۸۰۳۳۱
	لاپلاس گاوسی	۱۴۱۲۷	۵۰۱۳	۲۱۳۵	۰.۱۵۱۱۲	۰.۴۲۵۸۹
	تابع دودویی	۱۶۲۶۶	۹۲۴۰	۷۵۱۹	۰.۴۶۲۲۵	۰.۸۱۳۷۴
	کنی	۱۱۴۲۶	۱۱۹۰	۳۲۸	۰.۰۲۹۸۳	۰.۲۷۵۶۳

۴-۲-۳ ارزیابی زمان اجرا

برای بررسی پیچیدگی زمانی الگوریتم‌ها، زمان اجرای هر الگوریتم برای هر تصویر محاسبه شد. نتایج به دست آمده در جدول ۴-۲ موجود است. لازم به ذکر است که این نتایج برای نسخه غیر موازی الگوریتم‌ها محاسبه شده است و قابلیت موازی‌سازی الگوریتم‌ها در نظر گرفته نشده است.

با توجه به نتایج به دست آمده می‌توان دریافت که الگوریتم‌های سوبل، پرویت و رابرت کراس به دلیل اینکه تنها متشکل از یک عملیات هم‌گشت هستند، در مقایسه با سایر الگوریتم‌ها زمان اجرای بسیار کمتری دارند.

الگوریتم کنی با توجه نتایج بسیار خوبی که بر می‌گرداند زمان اجرای معقولی داشته است و حتی از الگوریتم‌های لاپلاس گاوسی و تابع دودویی نیز سریع‌تر بوده است.

یکی از دلایل کند بودن الگوریتم لاپلاس گاوسی این است که در پیاده‌سازی انجام شده از نسخه بدون ترکیب کرنل‌ها استفاده شده است و در صورت استفاده است نسخه ترکیبی، عملکرد این الگوریتم بهتر خواهد شد. دلیل دیگر این است که عملیات پیدا کردن نقاط تغییر علامت در این الگوریتم زمان‌بر است. الگوریتم تابع دودویی با توجه به تعداد زیاد الگوهایی که باید با پنجره‌های تصویر مقایسه شوند، بسیار کند بوده و در هر سه تصویر بیش از ده ثانیه زمان برای اجرا نیاز داشته است. البته قابلیت موازی‌سازی این الگوریتم نکته‌ای است که باید به آن توجه داشت و می‌توان آن را با موازی‌سازی بسیار سریع‌تر کرد.

جدول ۴-۲: زمان اجرای الگوریتم‌ها

تصویر معیار	الگوریتم	زمان اجرا (ثانیه)
تصویر ۴-۱۱	سوبل	۰.۶۹۵۸۴
	پرویت	۰.۰۴۳۰۳
	رابرت کراس	۰.۰۲۵۳۵
	لاپلاس گاوسی	۳.۲۷۶۶۸
	تابع دودویی	۳۵.۵۷۸۶۰
تصویر ۴-۱۱	کنی	۱.۷۳۳۳۰
	سوبل	۰.۰۱۵۰۶
	پرویت	۰.۰۱۵۶۱
	رابرت کراس	۰.۰۰۹۲۷
	لاپلاس گاوسی	۱.۶۸۸۱۱
تصویر ۴-۱۱	تابع دودویی	۱۲.۰۱۷۶۲
	کنی	۰.۸۰۳۰۸
	سوبل	۰.۰۱۳۸۷
	پرویت	۰.۰۱۳۰۳
	رابرت کراس	۰.۰۰۷۸۵
تصویر ۴-۱۱	لاپلاس گاوسی	۱.۲۱۵۷۹
	تابع دودویی	۱۰.۲۷۵۶۵
	کنی	۰.۶۷۷۵۲

۳-۴ خلاصه

در این فصل به ارزیابی الگوریتم‌های معرفی شده پرداختیم و دریافتیم که معیارهای ارزیابی متعددی برای بررسی عملکرد الگوریتم‌های آشکارسازی لبه وجود دارد. سپس سه معیار ارزیابی دیداری، ارزیابی پیوستگی و ارزیابی زمانی را مد نظر قرار دادیم و نتایج به دست آمده از الگوریتم‌ها را به کمک این معیار ها با هم مقایسه کردیم و متوجه شدیم که هر یک از الگوریتم‌ها مزایا و معایبی نسبت به سایر الگوریتم‌ها دارند. در نهایت، با توجه ارزیابی‌های انجام شده به این نتیجه رسیدیم که الگوریتم کنی از لحاظ پیوستگی و نازک بودن لبه‌ها و نتیجه تولید شده عملکرد بهتری نسبت به سایر الگوریتم‌ها داشته است. الگوریتم تابع دودویی در تصاویر بدون نویز تصاویری شبیه به الگوریتم کنی اما با لبه‌های ناپیوسته تولید می‌کند و نسبت به نویز بسیار حساس است و در حالت غیر موازی زمان اجرای آن طولانی است. الگوریتم‌های

سوبل، پرویت و رابرت کراس از لحاظ زمانی به صرفه هستند و تصاویر خوبی نیز تولید می‌کنند اما بیشتر لبه‌های تولید شده توسط آن‌ها ضخیم هستند. الگوریتم لاپلاس گاوسی نیز در بیشتر موارد لبه‌های ضخیم تولید می‌کند اما با توجه به استفاده از بلور گاوسی حساسیت کمتری به نویز دارد.

فصل پنجم

نتیجه‌گیری و پیشنهادات

در فصل پایانی گزارش به جمع‌بندی و مرور سیر مطالب ارائه شده پرداخته و سپس پیشنهاداتی را به منظور ادامه کار در زمینه الگوریتم‌های آشکارسازی لبه ارائه می‌کنیم.

۵-۱ نتیجه‌گیری

در این پژوهش در ابتدا به تعریف لبه و آشکارسازی لبه پرداختیم و با موانعی که بر سر راه آشکارسازی لبه قرار دارند آشنا شدیم و متوجه شدیم که هر یک از الگوریتم‌های آشکارسازی لبه در بینایی ماشین در شرایط خاصی می‌توانند عملکرد خوبی از خود به نمایش بگذارند. بنابراین، ضروری دانستیم تا با بررسی جزئیات هر یک از الگوریتم‌ها، نقاط ضعف و قوت هر یک را پیدا کنیم تا بدانیم در چه شرایطی باید از هر یک از الگوریتم‌ها استفاده کرد. به منظور بررسی جزئیات الگوریتم‌ها، لازم بود تا با برخی مفاهیم اولیه مانند نحوه کلی عملکرد الگوریتم‌ها و چگونگی استفاده آن‌ها از حد آستانه‌شنایی داشته باشیم. بعد از بررسی این موارد، جزئیات هر یک از الگوریتم‌های سوبل، پرویت، رابرت کراس، لاپلاس گاوسی، تابع دودویی و کنی را مورد مطالعه قرار دادیم و با چگونگی عملکرد هر یک آشنا شدیم و این الگوریتم‌ها را پیاده‌سازی کردیم تا آن‌ها را مورد ارزیابی قرار دهیم. به منظور ارزیابی این الگوریتم‌ها، در ابتدا با انواع معیارهای ارزیابی استفاده شده در پژوهش‌های مربوط به الگوریتم‌های آشکارسازی لبه آشنا شدیم و با توجه به اینکه در این پژوهش از الگوریتم‌هایی استفاده کردیم که به شکل دستی تنظیم می‌شوند، معیارهای با مرجع را کنار گذاشته و از سایر معیارهای ارزیابی یعنی ارزیابی دیداری، ارزیابی پیوستگی و ارزیابی زمان اجرا برای مقایسه الگوریتم‌ها استفاده کردیم. به منظور انجام عمل ارزیابی الگوریتم‌ها، از سه تصویر معیار آسان، متوسط و سخت استفاده کردیم تا بتوانیم هر یک از الگوریتم‌ها را از جهت درستی لبه‌های پیدا شده، پیوستگی و ضخامت لبه‌ها و نحوه عملکرد در محیط‌های گوناگون مورد بررسی و تحقیق قرار دهیم.

در نهایت، با توجه به نتایج به دست آمده متوجه شدیم که الگوریتم کنی می‌تواند در بسیاری از محیط‌ها عملکرد بسیار خوبی از خود نشان دهد و لبه‌ها را به درستی شناسایی کرده و با کمک مکانیزم‌های سرکوب غیربیشینه و آستانه‌گیری پسماند، آن‌ها را نازک‌تر و پیوسته‌تر سازد. الگوریتم تابع دودویی نیز علی‌رغم استفاده از یک رویکرد کاملاً متفاوت می‌تواند در تصاویر بدون نویز نتایجی نزدیک به الگوریتم کنی ولی با لبه‌های ناپیوسته‌تر ایجاد کند اما باید به این توجه داشت که نسخه غیر موازی آن هزینه محاسباتی بالایی دارد. سه الگوریتم سوبل، پرویت و رابرت کراس در بیشتر تصاویر عملکردی مشابه دارند به جز تصاویر نویز دار که الگوریتم رابرت کراس به دلیل حساسیت بالا به نویز عملکرد ضعیف‌تری از خود نشان می‌دهد؛ با این حال اگر تصویر دارای نویز زیادی نباشد، الگوریتم رابرت کراس لبه‌های نازک‌تری نسبت به دو الگوریتم دیگر ایجاد می‌کند و از این جهت برتر است. الگوریتم لاپلاس گاوسی نیز در بیشتر موارد لبه‌هایی ضخیم طول می‌کشد اما با توجه به اعمال بلور گاوسی می‌تواند در تصاویر نویز دار عملکرد بهتری نسبت به الگوریتم‌های حساس به نویز داشته باشد.

۲-۵ پیشنهادات

با توجه به این موضوع که الگوریتم‌ها در شرایط مختلف، عملکرد متفاوتی از خود نشان می‌دهند، می‌توان پیش از اعمال الگوریتم‌ها از یک مکانیزم برای تشخیص الگوریتم مناسب برای هر تصویر استفاده کنیم. برای پیاده‌سازی چنین مکانیزمی می‌توان از رویکردهای مبتنی بر یادگیری ماشین مانند یادگیری با نظارت و یادگیری تکاملی استفاده کرد.

همچنین، از آنجا که تنظیم پارامترهای هر یک از الگوریتم‌ها توسط انسان فرایند زمان‌بری است، می‌توان به این سمت پیش رفت تا الگوریتمی طراحی کرد که بتواند پیش از اعمال الگوریتم، بهترین پارامترها را بر اساس تصویر داده شده برای الگوریتم مورد نظر انتخاب کند.

در نهایت، از آنجا که هوش مصنوعی و الگوریتم‌های یادگیری ماشین و شبکه‌های عصبی نظیر شبکه عصبی پیچشی روز به روز در حال پیشرفت هستند و با توجه به قدرت بالای این الگوریتم‌ها و وجود مجموعه داده‌های متعدد در سطح اینترنت، می‌توان روش‌های کلاسیک بینایی ماشین را کنار گذاشت و صرفاً با استفاده از الگوریتم‌های مبتنی بر یادگیری ماشین، عمل آشکارسازی لبه را انجام داد. البته انجام این عمل نیازمند تعداد زیادی تصویر آموزشی و آزمایشی خواهد بود و در صورت کمبود این تصاویر، این الگوریتم‌ها دقت بسیار پایین‌تری از الگوریتم‌های کلاسیک بینایی ماشین خواهند داشت.

منابع و مراجع

- [1] Maini, Raman and Aggarwal, Himanshu. Study and comparison of various image edge detection techniques. International journal of image processing (IJIP), 3(1):1–11, 2009.
- [2] Ahmad, Muhammad Bilal and Choi, Tae-Sun. Local threshold and boolean function based edge detection. IEEE Transactions on Consumer Electronics, 45(3):674–679, 1999.
- [3] Nadernejad, Ehsan, Sharifzadeh, Sara, and Hassanpour, Hamid. Edge detection techniques: Evaluations and comparisons. Applied Mathematical Sciences, 2(31):1507–1520, 2008.
- [4] Tariq, Nazish, Hamzah, Rostam Affendi, Ng, Theam Foo, Wang, Shir Li, and Ibrahim, Haidi. Quality assessment methods to evaluate the performance of edge detection algorithms for digital image: A systematic literature review. IEEE Access, 2021.