بسم الله الرحمن الرحيم

دانشکده فنی و مهندسی
بخش مهندسی برق

رشته مهندسی برق گرایش الکترونیک دیجیتال

عنوان گزارش: لبه یابی با استفاده از روش‌های بهینه سازی

مولف:
نگین فلاحت

شماره دانشجویی:۴۰۱۵۵۱۰۰۷

استاد:
جناب آقای دکتر حسین نظام آبادی پور

تیر ۱۴۰۲

فهرست مطالب

عنوان **صفحه**

فصل اول: مقدمه و کلیات تحقیق ۱

۱-۱ مقدمه..... ۲

۱-۲ چرایی تحقیق..... ۵

۱-۳ ساختار گزارش..... ۵

فصل دوم: مروری بر تحقیقات انجام شده..... ٦

۲-۱ معرفی راه‌حل‌های مسئله..... ۷

۱ -۲-۱ روش‌های مبتنی بر جستجو..... ۷

۱-۱-۱-۲ الگوریتم کلونی مورچگان..... ۷

۲-۱-۱-۲ الگوریتم کلونی زنبورعسل..... ۱۷

۲-۱-۲ روش‌های مبتنی بر تکامل..... ۲۰

۲-۱-۲-۱ الگوریتم ژنتیک..... ۲۰

فصل سوم: مواد و روش‌ها ۲۹

۳-۱روش ارزیابی..... ۳۰

فصل چهارم: نتیجه‌گیری ۳۱

۴-۱ نتایج..... ۳۲

منابع و مآخذ ۳۳

الف) منابع انگلیسی..... ۳۴

فصل اول:
مقدمه و کلیات تحقیق

۱-۱ مقدمه

در دنیای امروز، تصاویر، بسیاری از محتوای دیدنی در محیط‌های مجازی و نرم‌افزارهای مختلف را تشکیل می‌دهند. لبه‌یابی در پردازش تصویر، به تشخیص لبه‌های تصویر کمک می‌کند که به نوبه خود ارائه‌دهنده‌ی اطلاعات مهمی در مورد محتوای تصاویر است. در برنامه‌های کاربردی مختلف مانند برنامه‌های تشخیص چهره، تشخیص پلاک خودرو، تشخیص شیء و...، لبه‌یابی برای تشخیص ویژگی‌های مهم تصویر مورد استفاده قرار می‌گیرد. در حوزه‌ی پزشکی، لبه‌یابی برای تشخیص و تحلیل تصاویر پزشکی بسیار حائز اهمیت است. این کاربرد در تشخیص و تصویرسازی اعضای مختلف بدن انسان مانند عروق خونی، بافت‌ها و... استفاده می‌شود. در صنعت، لبه‌یابی برای کنترل کیفیت و تشخیص خطا در محصولات صنعتی مانند قطعات خودرو، الکترونیکی، لوازم خانگی و... بسیار حائز اهمیت است.

لبه‌یابی (Edge detection) در پردازش تصویر، یک فرایند است که در آن لبه‌های تصویر به عنوان نقاطی که تغییر شدت رنگ دارند، شناسایی می‌شوند. لبه‌ها عموماً نشان‌دهنده مرزهای مختلفی هستند که در تصویر وجود دارند و اغلب برای استخراج ویژگی‌های تصویر مهم هستند. الگوریتم‌های لبه‌یابی معمولاً از فیلترینگ تصویر با استفاده از کرنل یا ماسک‌هایی که به عنوان یک رفتار معین در تصویر عمل می کنند، بهره می‌برند اما با افزایش اندازه و پیچیدگی تصویر، دستیابی به اپراتور تشخیص لبه کاری دشوار است.

الگوریتم سوبل(sobel) یکی از روش‌های معروف برای تشخیص لبه در تصاویر است که بر اساس کانولوشن فیلتر کوچک، جداپذیر و با مقادیر صحیح در جهت‌های افقی و عمودی بر تصویر، اعمال می‌شود. این فیلترها به شکل یک ترکیب خطی از مقادیر پیکسل‌های اطراف هر نقطه از تصویر، اعمال می‌شوند و سپس گرادیان و جهت لبه برای هر نقطه محاسبه می‌شود.

این الگوریتم نسبتاً محاسبات ساده‌ای دارد اما همانند هر الگوریتم دیگر، دارای محدودیت‌ها و عیوبی است که در زیر به برخی از آن‌ها اشاره خواهیم کرد:

- نیاز به تعیین پارامترهای آستانه‌سازی: برای استفاده از الگوریتم سوبل برای تشخیص لبه، نیاز است تا پارامترهای آستانه‌سازی مناسب برای تصویر تعیین شود. این پارامترها باید به گونه‌ای تعیین شوند که بتوانند بهترین نقاط لبه را در تصویر تشخیص دهند.
- عدم توانایی در تشخیص لبه‌های بسیار نازک: الگوریتم سوبل عموماً به دنبال تشخیص لبه‌های با ضخامت بزرگ است و نمی‌تواند لبه‌های بسیار نازک را تشخیص دهد

یکی دیگر از روش‌های پرکاربرد برای لبه‌یابی، الگوریتم کنی (Canny) است. این الگوریتم از یک فیلتر گاوسی برای کاهش نویز در تصویر استفاده می‌کند و سپس از دو فیلتر سوبل (یکی برای شناسایی لبه‌های عمودی و دیگری برای شناسایی لبه‌های افقی) برای شناسایی لبه‌های تصویر استفاده می‌کند. در نهایت، یک الگوریتم ترکیبی برای جمع‌آوری لبه‌هایی که به نظر می‌آیند واقعی هستند و رد کردن لبه‌هایی که به نظر می‌آیند نامعتبر است، استفاده می‌شود.

روش Canny برای لبه‌یابی تصویر، یکی از روش‌های پرکاربرد و موثر است ولی همانطور که هر الگوریتمی، دارای نقاط ضعفی است که باید در نظر گرفته شود، برخی از نقاط ضعف روش Canny عبارتند از:

- پارامترهای الگوریتم: الگوریتم Canny دارای پارامترهایی مانند اندازه فیلتر گاوسی، حداقل و حداکثر مقدار پارامتر تشدید لبه و غیره است. این پارامترها باید به طور دقیق تنظیم شوند تا بهترین نتیجه را به دست آورند، در غیر این صورت الگوریتم ممکن است لبه‌های اضافی یا کمبود لبه‌های مهم را شناسایی کند.
- تشخیص لبه در نویز بالا: الگوریتم Canny به دلیل استفاده از فیلتر گاوسی به منظور کاهش نویز، در تشخیص لبه‌ها در محیط‌های با نویز بالا مشکل دارد. در این حالت، لبه‌هایی که به دلیل نویز وجود دارند، ممکن است به عنوان لبه‌های واقعی تشخیص داده شوند.
- کارایی الگوریتم: الگوریتم Canny برای تصاویر با اندازه بزرگ، زمان پردازش بسیار طولانی دارد. این مشکل می‌تواند به خصوص در برنامه‌های پردازش تصویر که نیاز به پردازش سریع دارند، مشکل ساز باشد.

درنتیجه، روش Canny یکی از روش‌های پرکاربرد و موثر برای لبه‌یابی تصویر است، اما با توجه به نوع تصویر و مسئله مورد نظر، ممکن است با مشکلاتی برخورد کند.

بررسی فرایندهایی که در طبیعت با رفتار هوشمندانه رخ می‌دهند، منجر به توسعه روش‌های بهینه‌سازی شده است. یافتن الگوهای هوشمندانه در طبیعت می‌تواند به توسعه روش‌های بهینه‌سازی در مسائل مختلف کمک کند. روش‌های لبه‌یابی با استفاده از بهینه‌سازی، روش‌هایی هستند که برای تشخیص لبه در تصاویر از الگوریتم‌های بهینه‌سازی استفاده می‌کنند. در این روش‌ها، یک تابع هدف بهینه‌سازی مشخص می‌شود و برای پیدا کردن راه‌حل بهینه برای تشخیص لبه در تصویر، از الگوریتم‌های بهینه‌سازی مانند الگوریتم مورچه‌ای، الگوریتم لانه زنبوری، الگوریتم ژنتیک و غیره استفاده می‌شود.

تابع هدف یکی از مهمترین مؤلفه‌های الگوریتم بهینه‌سازی است که در آن برای پیدا کردن بهترین راه‌حل برای یک مسئله، یک تابع بهینه‌سازی تعریف می‌شود. تابع هدف، به عنوان معیاری برای ارزیابی راه‌حل‌های مختلف به کار می‌رود و با بررسی مقدار تابع هدف برای هر راه‌حل، بهترین راه‌حل برای مسئله به دست می‌آید. در مسائل تشخیص لبه در تصاویر نیز، تابع هدف می‌تواند معیاری باشد که برای ارزیابی راه‌حل‌های مختلف در تشخیص لبه به کار برده می‌شود. برای مثال، تابع هدف می‌تواند معیاری مبتنی بر تفاوت شدت پیکسل‌های مجاور باشد که برای تشخیص لبه در تصویر مورد استفاده قرار می‌گیرد.

راه‌حل‌های مبتنی بر بهینه‌سازی برای لبه‌یابی، خصوصیات مختلفی دارند که می‌توان به برخی از آن‌ها اشاره کرد:

- دقت بالا: روش‌های بهینه‌سازی برای لبه‌یابی، معمولاً دارای دقت بالایی هستند و توانایی تشخیص لبه در تصاویر با شرایط مختلف را دارند.
- امکانات گوناگون: انواع مختلفی از الگوریتم‌های بهینه‌سازی و معیارهای مختلفی برای تشخیص لبه در تصاویر مورد استفاده قرار می‌گیرند. بنابراین، این روش‌ها قابلیت تنوع و گستردگی را دارند و می‌توانند بهترین راه‌حل را برای هر مسئله مورد نظر پیدا کنند.
- سرعت بالا: در بسیاری از موارد، روش‌های بهینه‌سازی برای لبه‌یابی دارای سرعت بالایی هستند و می‌توانند با سرعت بالا به تشخیص لبه در تصاویر بپردازند.
- کاهش نویز: یکی از مشکلاتی که در تصاویر ممکن است پیش آید، وجود نویز است که می‌تواند باعث اشتباه در تشخیص لبه‌ها شود. با استفاده از روش‌های بهینه‌سازی، می‌توان نویز را کاهش داده و دقت تشخیص لبه‌ها را افزایش داد.
- تشخیص لبه‌های پیچیده: با استفاده از روش‌های بهینه‌سازی، می‌توان لبه‌های پیچیده‌تری را نیز تشخیص داد که با استفاده از روش‌های ساده‌تر ممکن است تشخیص دادن آن‌ها مشکل باشد.
- به کارگیری روش‌های آستانه‌سازی پیشرفته: الگوریتم‌های بهینه‌سازی معمولاً از روش‌های آستانه‌سازی پیشرفته‌تری استفاده می‌کنند که به دقت بیشتری در تشخیص لبه کمک می‌کنند.
- ترکیب چندین روش: برخی الگوریتم‌های بهینه‌سازی برای تشخیص لبه، از ترکیب چندین روش مختلف مانند فیلترهای کانولوشنی، فیلترهای لاپلاسین و غیره استفاده می‌کنند تا دقت بیشتری در تشخیص لبه داشته باشند.

روش‌های لبه‌یابی با استفاده از بهینه‌سازی، به طور کلی برای تصاویر با پیچیدگی متوسط تا بالا مناسب

هستند. با این حال، برای تصاویر با پیچیدگی بسیار بالا، ممکن است این روش‌ها با مشکل مواجه شوند و دقت کمی در تشخیص لبه داشته باشند. در تصاویر با پیچیدگی بسیار بالا، تعداد لبه‌ها و تغییرات شدید در رنگ و نور ممکن است باعث شود که الگوریتم بهینه‌سازی به دنبال یافتن جزئیات بیش از حد، به دام بیفتد و در نتیجه دقت کاهش یابد. بنابراین، در این موارد، روش‌های دیگری نظیر شبکه‌های عصبی عمیق که برای تشخیص لبه در تصاویر با پیچیدگی بالا آموزش داده شده‌اند، ممکن است بهترین گزینه باشند.

در کل، در هر مسئله می‌توان از ترکیب چندین روش برای تشخیص لبه استفاده کرد تا دقت بیشتری حاصل شود. همچنین، برای تصاویر با پیچیدگی بالا، ممکن است نیاز به پیش‌پردازش تصویر با استفاده از فیلترهایی نظیر فیلتر گاوسی باشد که باعث کاهش نویز و تقویت لبه‌ها می‌شوند.

۱-۲ چرایی تحقیق

لبه یابی یکی از مراحل مهم در پردازش تصویر است و در بسیاری از کاربردهای پردازش تصویر مانند استخراج ویژگی، شناسایی الگو، تقسیم‌بندی تصویر و مورفولوژی تصویر، مورد استفاده قرار می گیرد. اگر لبه‌ها به درستی شناسایی شوند، مکان تمام اشیاء در تصویر به دقت مشخص می‌شود و برخی ویژگی‌های اساسی مانند سطح و هندسه اشیاء به راحتی اندازه‌گیری می‌شود.

با توجه به اینکه بسیاری از برنامه‌های مختلف نیاز به دقت بالای لبه یابی دارند، جستجو روش‌های جدید برای لبه یابی بسیار مهم هستند. الگوریتم‌های بهینه‌سازی جدید می‌توانند با دقت بیشتری به نقاط لبه در تصویر دسترسی پیدا کنند و جزئیات بیشتری از تصویر را در نظر بگیرند. این روش‌ها می‌توانند در کاربردهایی که دقت بالای لبه یابی از اهمیت بسیاری برخوردار است، نقش مهمی ایفا کنند.

همچنین، با توسعه فناوری و افزایش استفاده از تصاویر دیجیتال، حجم بزرگی از تصاویر در دسترس قرار می‌گیرد. روش‌های جدید برای لبه یابی می‌توانند با پردازش سریع‌تر و دقیق‌تر تصاویر، کارایی و کاربرد پردازش تصویر را بهبود بخشند.

۱-۳ ساختار گزارش

در فصل دوم روش‌های مختلفی که برای تشخیص لبه با استفاده از بهینه‌سازی وجود دارند، دسته بندی و معرفی می‌شوند. همچنین نقاط ضعف و قوت هر دسته بیان می‌شوند. در فصل سوم روش‌های ارزیابی بیان می‌شوند و در نهایت، جمع بندی و نتیجه‌گیری این تحقیق در فصل آخر گردآوری شده است.

فصل دوم:
مروری بر ادبیات تحقیق

۲-۱ معرفی راه‌حل‌های مسئله

روش‌های لبه‌یابی بهینه‌سازی را می‌توان به دو دسته‌ی کلی زیر تقسیم کرد:

- روش‌های مبتنی بر جستجو
- روش‌های مبتنی بر تکامل

۲-۱-۱ روش‌های مبتنی بر جستجو

روش‌های مبتنی بر جستجو، به دسته‌ای از الگوریتم‌های بهینه‌سازی گفته می‌شود که در آن‌ها به دنبال جستجوی یک جواب بهینه در فضای جستجو مشخص هستیم. در این روش‌ها، یک وضعیت فعلی و یک یا چند معیار مشخص برای ارزیابی این وضعیت تعریف می‌شود و سپس با استفاده از یک روش خاص، وضعیت جدیدی برای جستجوی جواب بهینه تولید می‌شود. برخی از مشهورترین روش‌های مبتنی بر جستجو شامل الگوریتم کلونی مورچگان والگوریتم کلونی زنبورعسل هستند.

۲-۱-۱-۱ الگوریتم کلونی مورچگان[۱] (Ant colony optimization)

الگوریتم کلونی مورچه الهام گرفته شده از مطالعات و مشاهدات روی کلونی مورچه‌ها است. این مطالعات نشان داده که مورچه‌ها حشراتی اجتماعی هستند که در کلونی‌ها زندگی می‌کنند و رفتار آن‌ها بیشتر در جهت بقاء کلونی است تا در جهت بقاء یک جزء از آن. یکی از مهم‌ترین و جالب‌ترین رفتار مورچه‌ها، رفتار آن‌ها برای یافتن غذا است؛ به ویژه چگونگی پیدا کردن کوتاه‌ترین مسیر میان منابع غذایی و آشیانه. این نوع رفتار مورچه‌ها دارای نوعی هوشمندی توده‌ای است که مورد توجه دانشمندان قرار گرفته‌است. در دنیای واقعی مورچه‌ها ابتدا به‌طور تصادفی به این سو و آن سو می‌روند تا غذا بیابند. سپس به آشیانه بر می‌گردند و ردّی از فرومون (Pheromone) برجای می‌گذارند. چنین ردهایی پس از باران به رنگ سفید در می‌آیند و قابل رویت اند. مورچه‌های دیگر وقتی این مسیر را می‌یابند، گاه پرسه زدن را رها کرده و آن را دنبال می‌کنند. سپس اگر به غذا برسند، به آشیانه بر می‌گردند و رد دیگری از خود در کنار رد قبل می‌گذارند و به عبارتی مسیر قبل را تقویت می‌کنند.

فرومون به مرور تبخیر می‌شود که از دو جهت مفید است:

۱- باعث می‌شود مسیر، جذابیت کمتری برای مورچه‌های بعدی داشته باشد. از آنجا که یک مورچه در یک زمان بلند مدت، راه‌های کوتاه‌تر را بیش‌تر می‌پیماید و تقویت می‌کند، هر راهی بین خانه و غذا که کوتاه‌تر (بهتر) باشد، بیشتر تقویت می‌شود و آنکه دورتر است، کمتر. در نتیجه، مسیرهای بلندتر که توسط مورچگان پیمایش شوند، جذابیت کمتری دارند و در اولویت پایین‌تری برای جستجوی جواب بهینه قرار می‌گیرند.

۲- اگر فرومون اصلاً تبخیر نمی‌شد، مسیرهایی که چند بار طی می‌شدند، چنان بیش از حد جذّاب می‌شدند که جستجوی تصادفی برای غذا را بسیار محدود می‌کردند. لذا وقتی یک مورچه مسیر کوتاهی (خوبی) را از آشیانه تا غذا بیابد، بقیه مورچه‌ها به احتمال زیادی همان مسیر را دنبال می‌کنند و با تقویت مداوم آن مسیر و تبخیر ردهای دیگر، به مرور همه مورچه‌ها، هم مسیر می‌شوند.

هدف الگوریتم مورچه‌ها تقلید این رفتار توسط مورچه‌های مصنوعی است که روی نمودار در حال حرکت اند. مسئله یافتن کوتاه‌ترین مسیر است و حل‌کننده این مسئله، مورچه‌های مصنوعی اند.

الگوریتم کلونی مورچه، یکی از الگوریتم‌های محاسباتی بهینه‌سازی است که با مدل‌سازی رفتار مورچه در جستجوی غذا، برای حل مسائل بهینه‌سازی مورد استفاده قرار می‌گیرد. این الگوریتم ابتدا یک گراف را به عنوان نمایش مسئله در نظر می‌گیرد، سپس با استفاده از مورچه‌ها، به دنبال مسیرهای بهینه در گراف می‌گردد.

در مسئله تشخیص لبه، گراف مسئله به صورت تصویر تعبیه شده است و هر پیکسل تصویر به عنوان یک گره در گراف در نظر گرفته می‌شود. هدف از اجرای الگوریتم کلونی مورچه در این مسئله، پیدا کردن مسیرهایی است که بین پیکسل‌هایی که لبه دارند، قرار دارند. برای این منظور، ابتدا یک تعدادی مورچه به صورت تصادفی در تصویر قرار داده می‌شوند. سپس هر مورچه با شروع از یک پیکسل، به صورت تصادفی به پیکسل‌های همسایه خود حرکت می‌کند و مسیر حرکت خود را با استفاده از یک فرمول ریاضی به روزرسانی می‌کند.

یکی از فرمول‌های مورد استفاده در الگوریتم مورچه برای تشخیص لبه، فرمول تعیین مقدار فرومون است. در این فرمول، هرمورچه پس از حرکت به یک پیکسل، مقدار فرومون را بین آن پیکسل و پیکسل قبلی خود به روزرسانی می‌کند. سپس در هر مرحله، احتمال حرکت هر مورچه به پیکسل‌های همسایه خود، بر اساس مقدار فرومونی که بین آن پیکسل و پیکسل مورد نظر وجود دارد، محاسبه می‌شود. در نهایت، مسیرهایی که بیشترین مقدار فرومون را دارند، به عنوان مسیرهای لبه شناسایی می‌شوند

استفاده از الگوریتم مورچه‌ای برای تشخیص لبه در تصاویر، نیازمند تنظیم پارامترهای مختلف الگوریتم است. برای مثال، تعداد مورچه ها، مقدار فرومون اولیه، پارامترهای فرمول به روزرسانی فرومون و ... از جمله پارامترهای قابل تنظیم هستند برای بهبود عملکرد الگوریتم.

در نهایت، باید گفت که استفاده از الگوریتم مورچه‌ای برای تشخیص لبه در تصاویر، یکی از روش‌های پیشرفته و موثر در این زمینه است. با این حال، این الگوریتم، همانند سایر الگوریتم‌های پردازش تصویر، با مشکلاتی مانند تاثیر نویز و ... روبرو می‌شود. بنابراین، برای استفاده موفق از این الگوریتم، نیاز به تنظیم پارامترهای مناسب و پیش‌پردازش تصویر قبل از اجرای الگوریتم وجود دارد.

مراحل الگوریتم:

در طول حرکت مورچه ها، انتخاب موقعیت بعدی با تابع احتمال انتقال تعیین می شود. احتمال اینکه یک مورچه از پیکسل (i, j) به پیکسل(m, n) حرکت کند، طبق رابطه زیر نشان داده می‌شود:

$$P^{k}_{(i,j)\to(m,n)} = \frac{\left(\tau^{k-1}_{(m,n)}\right)^{\alpha}\left(\eta_{(m,n)}\right)^{\beta}}{\sum_{(m,n)\in\Omega(i,j)}\left(\tau^{k-1}_{(m,n)}\right)^{\alpha}\left(\eta_{(m,n)}\right)^{\beta}} \qquad (۱)$$

در اینجا $\tau^{k-1}_{(m.n)}$ مقدار فرومون است. $\eta_{(m.n)}$ قابلیت دید پیسکل(m,n) است که با استفاده از حداکثرتغییرات سطح خاکستری تصویر، تعیین می‌شود.همچنینα و β پارامترهای کنترلی می‌باشند.

با حرکت مورچه در امتداد لبه، فرمون غلظت باقیمانده هر پیکسل در مسیر تغییر خواهد کرد. بر این اساس، دو به روزرسانی از فرمون در فرآیند تکرار الگوریتم وجود دارد:

اولین عملیات، به روز رسانی محلی، همانطور که در معادله نشان داده شده است، پس از تکمیل انتقال هر مورچه انجام می شود:

$$\tau^{k-1}_{(m,n)} = \begin{cases} (1-\rho)\,\tau^{k-1}_{(m,n)} + \rho\Delta\tau^{p}_{(m,n)}, \\ \qquad \text{the ant p passes through the pixel} \\ \tau^{k-1}_{(m,n)}, \text{ other} \end{cases} \qquad (۲)$$

بعد از اینکه همه k مورچه تکرار را کامل کردند، دومین عملیات به روز رسانی فرامحلی انجام خواهد شد:

(۳)

$$\tau_{(m,n)}^{k} = (1-\rho)\tau_{(m,n)}^{k-1} + \Delta\tau_{(m,n)}^{k-1}$$

با انجام تمام حرکات مورد نیاز توسط مورچه‌ها، یک آستانه مناسب برای تشخیص غلظت فرومون باقی‌مانده در مسیر مورچه‌ها انتخاب می‌شود. سپس پیکسل‌هایی که غلظت فرومون آن‌ها بیشتر از آستانه تعیین شده است، به عنوان پیکسل‌های لبه مشخص می‌شوند و تمام پیکسل‌های لبه در تصویر استخراج می‌شوند تا لبه کامل تصویر به دست آید.

در ادامه به بررسی نحوه بهبود روش لبه یابی Canny به کمک الگوریتم کلونی مورچه می‌پردازیم.[۲]

به منظور حل مشکل ناپیوستگی لبه در روش‌های قبلی لبه یابی، روش جدید تشخیص لبه پیشنهاد شده است که عملگر Canny را با الگوریتم بهینه سازی کلونی مورچه‌ها ترکیب کرده و بهبود یافته است.

از این روش می توان در سیستم های کمک راننده، جهت کاهش تصادفات و افزایش ایمنی و در نتیجه نجات جان افراد بهره برد. یکی از مسائل پیچیده و چالش برانگیز برای این سیستم ها، تشخیص خطوط جاده است. تشخیص خطوط جاده به دلیل وجود شرایط متفاوت در جاده های مختلف، یک مسئله پیچیده است. در این پژوهش، یک رویکرد ترکیبی با استفاده از بهینه‌سازی مورچه‌ها برای تشخیص خطوط جاده ارائه شده است. در این روش، ابتدا با استفاده از Canny، لبه‌های تصویر استخراج می‌شوند، سپس با استفاده از چندین فرایند، خطوط جاده استخراج می‌شوند. این روش می‌تواند برای جاده‌های مستقیم استفاده شود.

برای حل مسئله تشخیص خطوط، دو نوع روش موجود است که شامل روش مبتنی بر ویژگی ها و روش مبتنی بر مدل هستند. روش مبتنی بر مدل از چندین پارامتر برای نمایش تصویر خط استفاده می کند وموجب محاسبات پیچیده میشود. در حالی که روش مبتنی بر ویژگی، خطوط در تصاویر با تقسیم بندی ساده تصویر شناسایی می شوند.

ویژگی‌های اصلی که روش‌های تشخیص خطوط (یا مرزها) باید دارا باشند در زیر بیان شده‌اند:

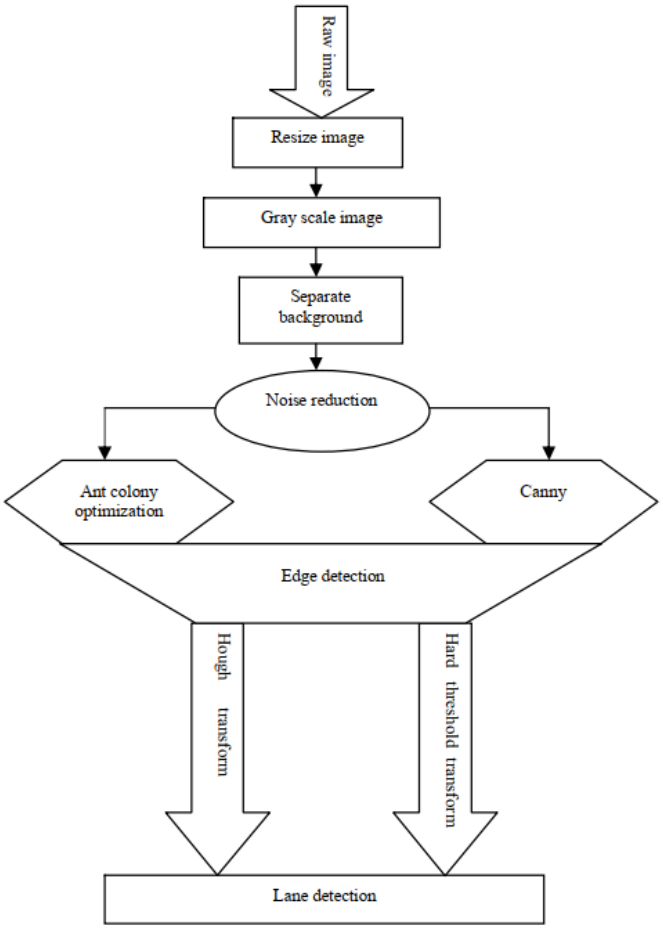
• باید بتوانند با جاده‌های منحنی نیز سر و کار داشته باشند و فرض کردن جاده‌های راست برای آنها قابل قبول نیست.

• باید قابلیت پردازش جاده‌های رنگی و بدون رنگ را داشته باشند.

سه مشکل اصلی ممکن است تشخیص خط را دچار مشکل کنند:

- تغییر نور
- سایه ها
- مسدود شدن خطوط با وسایل ن

بررسی اجمالی الگوریتم:



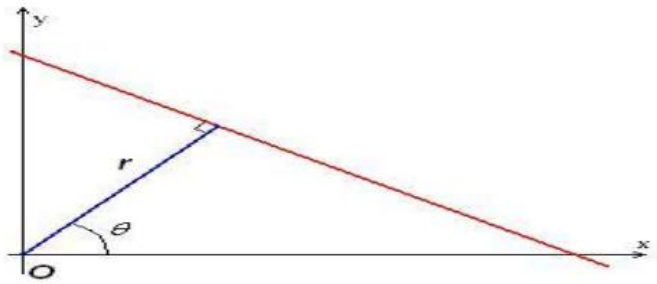
شکل ۲-۱ الگوریتم تشخیص خط جاده با لبه های Canny بهبود یافته با استفاده از بهینه سازی کلونی مورچه ها

در سیستم تشخیص خط برای خودرو، ابتدا یک تصویر رنگی توسط دوربین رنگی گرفته می‌شود و سپس در حافظه‌ی کامپیوتر ذخیره می‌شود. سیستم تشخیص خط، تصویر را از حافظه‌ی کامپیوتر می‌خواند و با پردازش تصویر شروع به تشخیص خطوط می‌کند. برای به دست آوردن تخمین خوبی از خطوط و بهبود سرعت الگوریتم، اندازه تصویر اصلی به ۲۵۵x۲۵۵ پیکسل کوچک شده است. استفاده از اطلاعات رنگی برای تفکیک جاده از خطوط مرزی با مشکلاتی در تشخیص لبه‌ها روبه‌رو می‌شود. در عمل، سطح جاده می‌تواند به دلیل سایه‌ها، سبک یا سن مختلف کف سطح جاده، از رنگ‌های مختلفی تشکیل شود که باعث تغییر رنگ سطح جاده و نشانه‌های خط و خطوط از یک ناحیه تصویر به ناحیه دیگری می‌شود. به همین دلیل، تصاویر رنگی به تصاویر سطح خاکستری تبدیل می‌شوند. با این کار، پردازش تصاویر سطح خاکستری نسبت به تصاویر رنگی، کمترین هزینه را دارد. . مرزهای خطوط توسط تفاوت شدید بین سطح جاده و خطوط رنگی یا سطح غیر خاکی تعریف می‌شوند. این تفاوت شدید باعث ایجاد لبه‌ها در تصویر می‌شود. در واقع، این لبه‌ها مرزهای خطوط را تشکیل می‌دهند که در ادامه برای تشخیص خطوط مورد استفاده قرار می‌گیرند.

ورودی الگوریتم تشخیص خط برای خودرو، خروجی الگوریتم کنی می باشد. این تصویر یک تصویر دودویی با لبه‌های باریک است که هر یک از این لبه‌ها ۱ پیکسل اند.

هدف الگوریتم کلونی مورچه، استخراج اطلاعات لبه‌های اضافی است که در زمان تشخیص لبه، با استفاده از الگوریتم تشخیص لبه کنی در نظر گرفته نمی‌شوند. به همین دلیل، در اینجا الگوریتم بهینه‌سازی مورچه برای حل مسئله تشخیص لبه در تصاویر معرفی شده است. حال با استفاده از چندین فرآیند، خطوط با الگوریتم استخراج شده و با استفاده از تبدیل هاف (Hough Transform) تشخیص داده می‌شوند. این روش برای جاده‌های رنگی و خطوط ساده مورد استفاده قرار می‌گیرد.

نحوه تشخیص خط(Line Detection):

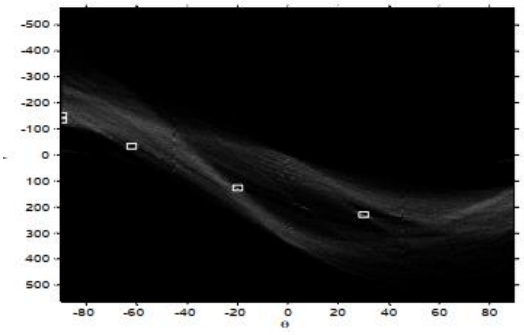


شکل۲-۲

در اینجا، پارامتر r فاصله بین خط و مبدا را نشان می‌دهد، در حالی که پارامتر θ زاویه بردار از مبدا به نزدیک‌ترین نقطه را با استفاده از این پارامترها نشان می‌دهد. به عبارت دیگر، در این روش، خطوط با استفاده از پارامترهای r و θ مشخص می‌شوند.

$$r = x\cos\theta + y\sin\theta \qquad (۴)$$

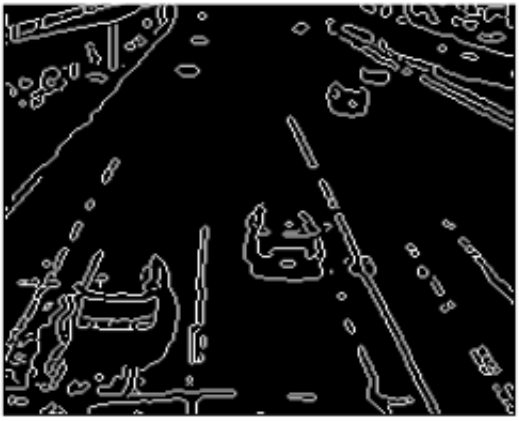
در ادامه از تبدیل هاف برای تشخیص خط برای خودرو یا راننده استفاده می‌شود. در این فضا، قله‌های نشان داده شده (شکل۲-۳) نشان دهنده وجود خطوط در تصویر واقعی می‌باشند. محور افقی θ، موقعیت خط را نسبت به تصویر نشان می‌دهد و محور عمودی ρ، فاصله خط از گوشه صفحه تصویر را نشان می‌دهد. به عبارت دیگر، در این فضا، قله‌ها نشان دهنده‌ی خطوط موجود در تصویر واقعی با توجه به موقعیت و فاصله این خطوط در تصویر می‌باشند.



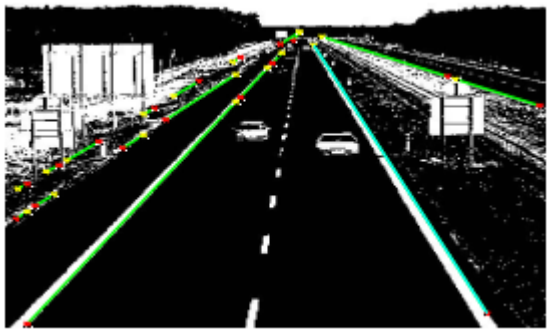
شکل۲-۳

نتایج آزمایشات نشان می‌دهد که روش پیشنهادی قابل قبول و قابل اطمینان است.

شکل۲-۴
تشخیص لبه با الگوریتم بهینه سازی کلونی مورچه‌ها



شکل ۲-۵
تشخیص لبه با الگوریتم Canny



شکل۲-۶
تشخیص خط

مشکلاتی که ممکن است در کاربرد عملی الگوریتم کلونی مورچه برای تشخیص خطوط لبه در تصاویر به‌وجود آید:

- ممکن است موقعیت اولیه مورچه‌ها نادرست باشد که می‌تواند باعث کاهش دقت الگوریتم شود.

- همچنین، مورچه‌ها در حالت جستجوی خود، به نویز حساس هستند و می‌تواند موجب کاهش دقت در تشخیص خطوط لبه شود
- همگرایی زودهنگام

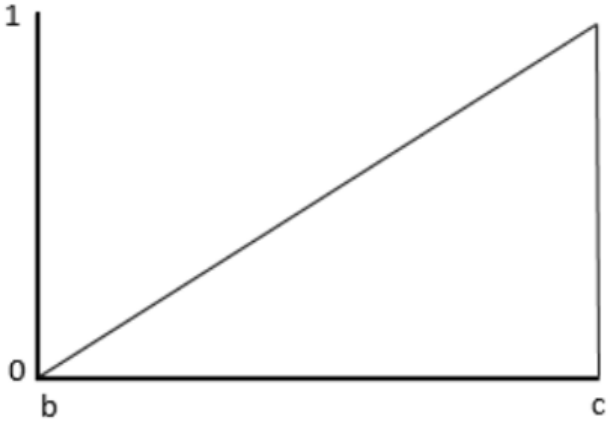
در الگوریتم مورچه‌ها، همگرایی زودهنگام زمانی رخ می‌دهد که مورچه‌ها در حرکت به سمت لبه‌های تصویر، به مینیمم محلی در فضای حل مسئله برخورد می‌کنند. به عبارت دیگر، ممکن است برخی از مورچه‌ها به دلیل جایگاه تصادفی شروع حرکت، در فاصله‌ی بسیاری از لبه‌های تصویر قرار گیرند و به دلیل اینکه به سمت مینیمم محلی در این نواحی حرکت می‌کنند، نتوانند به لبه‌هایی که در فاصله‌ی دورتری قرار دارند، دسترسی پیدا کنند. در اینجا همگرایی زودهنگام باعث تشخیص نادرست لبه‌های تصویر و استخراج ناکامل آن‌ها می‌شود.

روشی که برای بهبود الگوریتم بهینه‌سازی مورچه‌ها ارائه شده است، اضافه کردن تابع عضویت فازی مثلثی(Fuzzy Triangle Membership function) به الگوریتم بهینه‌سازی مورچه‌ها می‌باشد. [۳] در این روش تابع عضویت فازی مثلثی با اطلاعات خاکستری ترکیب شده است تا تابع هیوریستیک الگوریتم بهینه‌سازی مورچه‌ها تعیین شود. در این روش، اطلاعات خاکستری به عنوان یکی از جنبه‌های ورودی الگوریتم بهینه‌سازی مورچه‌ها در نظر گرفته می‌شود. تابع عضویت فازی مثلثی در معادله زیر نشان داده شده است:

$$\eta_{(m,n)} = \frac{\Delta}{(c-b)} \qquad (۵)$$

$$\Delta = f(i,j) - \left(\frac{\sum_{k=-1 to 1} f(i+k, j+k)}{8} \right) \qquad (۶)$$

در اینجا f(i,j) مقدار خاکستری پیکسل در ردیف i و ستون j می‌باشد.همچنین c بزرگترین مقدار خاکستری محلی و b کوچکترین مقدار خاکستری محلی را نشان می‌دهند.



شکل۲-۷ رابطه بین عضویت فازی و ارزش خاکستری محلی

برای تشخیص لبه در تصویر، ابتدا باید بررسی شود که آیا هر پیکسل در تصویر به لبه تعلق دارد یا خیر. برای این منظور، از اختلاف مقدار خاکستری پیکسل مرکزی با میانگین مقادیر خاکستری هشت پیکسل مجاور استفاده می‌شود. در صورتی که این اختلاف بزرگ باشد، به این معنی است که مقدار خاکستری پیکسل مرکزی با مقادیر خاکستری پیکسل‌های مجاورش تفاوت زیادی دارد و بنابراین احتمال بالایی دارد که این پیکسل به لبه تعلق داشته باشد. با استفاده از رابطه‌ی مربوطه، می‌توان مقدار عضویت فازی هر پیکسل را به عنوان ماتریس اطلاعات هیوریستیک الگوریتم بهینه‌سازی مورچه‌ها به دست آورد. به این ترتیب، با استفاده از این ماتریس هیوریستیک، الگوریتم بهینه‌سازی مورچه‌ها می‌تواند بهترین مسیرهایی را برای تشخیص لبه در تصویر پیدا کند.

در این روش، الگوریتم جستجوی مورچه‌ها زمانی متوقف می‌شود که به یکی از سه شرط زیر برسد:

الف) مورچه به نقطه پایانی مسیر رسیده باشد.

ب) مورچه به یک نقطه تکراری در مسیر برخورد کند.

ج) مورچه به یک نقطه با شدت بسیار کمی از ماتریس هیوریستیک برخورد کند.

با تعیین این سه شرط برای متوقف کردن جستجوی مورچه‌ها، جستجوی بی‌معنی در لبه‌ها و مناطق غیر لبه در تصویر به طور موثری کاهش می‌یابد. پس از اتمام تکرار تمامی مورچه‌ها، با تعیین آستانه غلظت فرومون، پیکسل‌های با سطح فرومون بالاتر از آستانه به عنوان پیکسل‌های لبه انتخاب می‌شوند. سپس، منطقه میان دو نقطه پایانی لبه در تصویردوباره تشخیص داده می‌شود و اطلاعات لبه با استفاده از این روش بهتر و پیوسته‌تر خواهد بود.

در ادامه یک روش تشخیص خطوط لبه با استفاده از الگوریتم بهبودیافته کلونی مورچه و اپراتور Canny پیشنهاد شده است.

در الگوریتم Canny، برای تصمیم‌گیری در مورد اینکه آیا یک پیکسل ممکن است به خط لبه تعلق داشته باشد یا خیر، از دو آستانه استفاده می‌شود. انتخاب مناسب این دو آستانه در تشخیص خطوط لبه بسیار مهم است. انتخاب نامناسب آستانه‌ها می‌تواند منجر به تشخیص نادرست خطوط لبه، شکست خطوط لبه و از دست دادن اطلاعات خطوط لبه شود. برای بهبود انتخاب دو آستانه در الگوریتم Canny ، می‌توان با استفاده از الگوریتم بهبود یافته کلونی مورچه، به جای انتخاب دستی، آستانه‌های مناسب برای تصمیم‌گیری در مورد پیکسل‌های خطوط لبه تعیین کرد.

در این روش، در ابتدا، لبه تصویر با استفاده از عملگر Canny استخراج می‌شود. سپس انتهای لبه به عنوان موقعیت اولیه مورچه در نظر گرفته می‌شود. سپس برای تولید ماتریس هیوریستیک، از تابع عضویت سه‌گانه فازی به جای اطلاعات گرادیان استفاده می‌شود.

در الگوریتم بهینه‌سازی مورچه‌ها برای تشخیص لبه در تصاویر، ماتریس هیوریستیک، ماتریسی است که برای هر پیکسل تصویر، یک امتیاز به عنوان فاکتور تصمیم‌گیری به الگوریتم می‌دهد. این امتیاز، به صورت عددی بیان می‌شود و نشان می‌دهد که هر پیکسل به چه اندازه می‌تواند لبه داشته باشد.سپس هر مورچه به ازای هر مرحله، یکی از مسیرهای پیشنهادی را با توجه به اطلاعات موجود در ماتریس هیوریستیک به صورت تصادفی انتخاب می‌کند و به سمت آن حرکت می‌کند. هر مورچه در حرکت خود، با توجه به ماتریس هیوریستیک و اطلاعات جمعیتی دیگر مورچه‌ها، اطلاعات جدیدی درباره مسیر به دست می‌آورد و سپس این اطلاعات را در ماتریس هیوریستیک بروزرسانی می‌کند. هر مورچه تا زمانی که به یکی از سه شرط توقف برسد، به طور پیوسته در تصویر حرکت می‌کند و سعی می‌کند مسیرهایی را که احتمالاً شامل لبه هستند، پیدا کند. اما هرگاه یکی از شرایط توقف رخ دهد، مورچه دیگر ادامه جستجوی خود را در این مسیر نمی‌دهد و به جستجوی مسیرهای دیگرمی‌پردازد.

با استفاده از این الگوریتم، نواحی ناپیوسته در لبه‌های تصویر پر شده و اطلاعات لبه با دقت و پیوستگی بیشتری تشخیص داده می‌شود. در نتیجه، تصاویر ورودی بهبود و روشن‌تر می‌شوند و برای استفاده در الگوریتم‌های دیگر، مانند تشخیص الگو و تشخیص شی، مناسب‌تر می‌شوند.

شکل۲-۸
تصویر اصلی

شکل۲-۹
نتایج
تشخیص لبه
الگوریتم
مورچگان
بهبودیافته و
کنی

۲-۱-۱- ۲ الگوریتم کلونی زنبورعسل[۴](Artificial Bee Colony Algorithm)

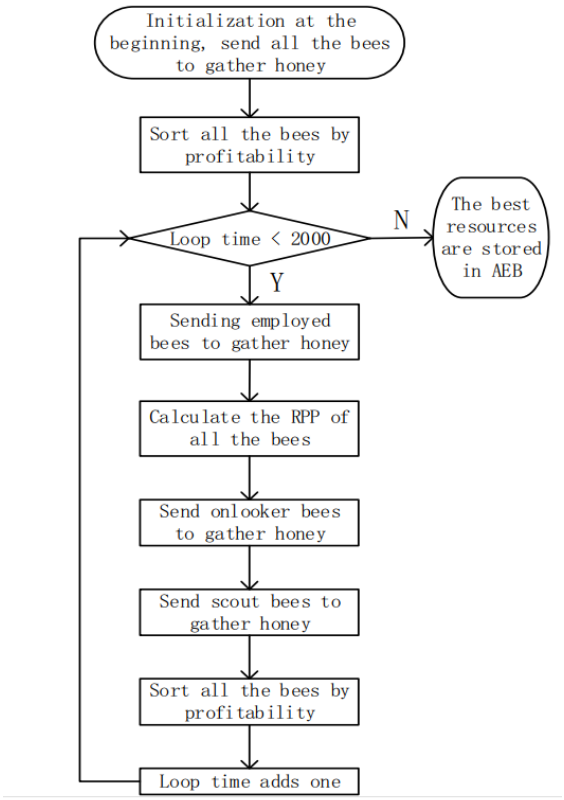
الگوریتم کلونی زنبورعسل (ABC)، که در سال ۲۰۰۵ توسط Dervis Karaboga ارائه شده است، یکی از الگوریتم‌های هوشمند است که از آن برای بهبود بهینه‌سازی مسائل استفاده می‌شود. یکی از مزایای این الگوریتم این است که در هر تکرار، جستجوی محلی و فرامحلی انجام می‌دهد و برای پیاده‌سازی بسیار سریع و آسان است.

برخی تحقیقات نشان داده که الگوریتم کلونی زنبورعسل بهتر از الگوریتم بهینه‌سازی کلونی مورچه در تشخیص لبه تصاویر عمل می‌کند و همچنین زمان کمتری نیز برای اجرای آن نیاز است.

زنبورهای عسل در کلونی‌ها زندگی می‌کنند و ویژگی‌هایی مانند جستجوی غذا، رقص زنبور، تصمیم‌گیری گروهی، انتخاب محل ایجاد کندو و سیستم های مسیریابی در آنها وجود دارد که می‌توانند به عنوان مدل‌هایی برای الگوریتم‌های هوشمند مورد استفاده قرار گیرند. در واقع، بسیاری از پژوهشگران تحت تاثیر رفتارهای زنبورها الهام گرفته‌اند و الگوریتم‌هایی را توسعه داده‌اند.

در این الگوریتم، یک کلونی زنبور مصنوعی شامل زنبوران کارگزار، ناظر و پیشاهنگ است. هر زنبور کارگزار با یک منبع غذایی مطابقت دارد و با استفاده از اطلاعات غذایی که شامل سودآوری و فاصله منبع غذایی است، به دنبال غذا می‌گردد. پس از پایان فرآیند جستجوی زنبوران کارگزار، بالای گلزاری که اندوخته کیفی مطمئنی از گرده دارد، رقص خاصی را اجرا می‌کند. این رقص که به نام رقص چرخشی شناخته می‌شود، اطلاعات مربوط به جهت گلزار (نسبت به کندو)، فاصله تا گلزار و کیفیت گلزار را به زنبورهای دیگر انتقال می‌دهد. بر اساس رقص، هر زنبور ناظر تعدادی منبع غذایی با توجه به مقدار احتمال مرتبط با آنها انتخاب می‌کند. زنبور ناظر با در نظر گرفتن سودآوری منبع غذایی، به عنوان زنبور کارگزار جایگزین می‌شود و به دنبال یافتن غذا می‌گردد. زنبور کارگزار می‌تواند بهترین موقعیت منبع غذایی را به خاطر بسپارد و بر اساس این خاطره، در همسایگی به دنبال غذا بگردد. وظیفه زنبور پیش آهنگ نیز جستجوی موقعیت جدید است و به دنبال اطلاعات منبع غذایی جدید می‌گردد.

بررسی اجمالی الگوریتم:[۵]



شکل۲-۱۰ نمودار جستجوی نقاط بهینه شده فرامحلی در الگوریتم کلونی زنبور مصنوعی

مراحل اصلی را می توان به شرح زیر توصیف کرد:

۱- همه زنبورها مقدار دهی اولیه می‌شوند.

۲- چهار مرحله را به صورت زیر تکرار می‌شود:

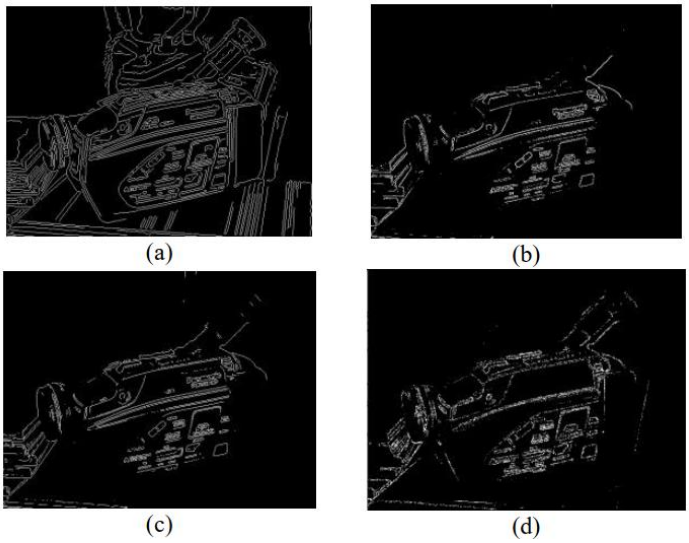
- هنگامی که زنبور کارگزار یک منبع غذایی را پیدا می‌کند، اطلاعات مربوط به آن را به زنبورهای دیگر نیز ارسال می‌کند. این اطلاعات شامل سودآوری و فاصله منبع غذایی است. این کار به دیگر زنبورها کمک می‌کند تا به سرعت به دنبال منابع غذایی جدید بگردند و به حل مسئله بهینه‌سازی برسند. لذا می‌توان گفت که زنبور کارگزار با تایید منابع غذایی دیگر در اطراف منبع غذایی، اطلاعات را با سایر زنبورها به اشتراک می‌گذارد.
- زنبور ناظر با توجه به سودآوری منبع غذایی، به عنوان یک زنبور کارگزار جایگزین می‌شود و به دنبال یافتن منابع غذایی جدید می‌گردد. زنبور ناظر با تایید منابع غذایی با استفاده از اطلاعات خود و اطلاعاتی که از زنبور کارگزارهای دیگر دریافت می‌کند، به دنبال یافتن منابع غذایی جدید می‌گردد و اطلاعات را با سایر زنبورها به اشتراک می‌گذارد.
- زنبور پیشاهنگ معمولاً در صورتی که زنبورهای دیگر نتوانسته‌اند منبع غذایی جدیدی پیدا کنند، به دنبال یافتن موقعیت جدیدی می‌گردد. برای جستجوی موقعیت جدید، زنبور پیشاهنگ به طور تصادفی در فضای حل جستجو می‌کند و به دنبال منابع غذایی جدید می‌گردد. اگر زنبور پیشاهنگ موفق به پیدا کردن منبع غذایی جدید شد، اطلاعات را به سایر زنبورها ارسال می‌کند تا آن‌ها نیز بتوانند به دنبال آن منبع غذایی بگردند.
- در الگوریتم زنبور عسل، اطلاعاتی در مورد منابع غذایی جدید که توسط زنبورهای کارگزار، ناظر و پیشاهنگ جمع‌آوری شده است، در حافظه نگهداری می‌شود. با نگهداری اطلاعات بهترین منبع غذایی در حافظه، الگوریتم قادر خواهد بود تا در تکرارهای بعدی، بهترین منبع غذایی را در نظر گرفته و به سمت آن هدایت شود. این کار به دست آوردن بهینه‌ترین پاسخ الگوریتم کمک می‌کند.

۳- تکرارهای الگوریتم به مدت مشخصی ادامه پیدا می‌کند تا شرایطی که برای پایان الگوریتم تعیین شده‌اند برآورده شوند. این شرایط می‌تواند یک حداکثر تعداد تکرار، یک حداقل مقدار خطا یا یک حداکثر سود باشد. درصورتی که شرایط مورد نظر برآورده شود، تکرارهای الگوریتم متوقف می‌شود و پاسخ الگوریتم به عنوان خروجی ارائه می‌شود.

با استفاده از این الگوریتم، بهترین منابع غذایی در تصاویر با دقت بالا و سرعت بیشتری پیدا خواهند شد.

شکل۲-۱۱ تصویر اصلی

.



شکل۲-۱۲

(a)کنی (b) رابرت(c) سوبل

(d) الگوریتم کلونی زنبور مصنوعی

۲-۱-۲ روش‌های مبتنی بر تکامل

این روش‌ها بر اساس ایده‌ی تکامل زیستی، با تولید جمعیتی از حلقه‌های احتمالی، به سمت جواب بهینه حرکت می‌کنند. یکی از مشهورترین روش‌های مبتنی بر تکامل، الگوریتم ژنتیک می‌باشد.

روش‌های مبتنی بر تکامل در بهینه‌سازی، شامل الگوریتم‌هایی هستند که بر اساس فرایند تکاملی مانند انتخاب طبیعی، تلاش برای تولید نسل‌های جدید و ارزیابی عملکرد هر نسل، بهبود راه‌حل‌ها را به دست می‌آورند. در این روش‌ها، معمولاً یک جمعیت از راه‌حل‌ها با استفاده از تکنیک‌های تولید تصادفی یا مبتنی بر قوانین ژنتیکی تولید می‌شود. سپس با اعمال عملیات تکاملی ، برای تولید نسل جدید و ترکیب نسل‌های مختلف، راه‌حل‌های بهینه‌تری به دست می‌آید.

۲-۱-۲-۱ الگوریتم ژنتیک[۶](Genetic Algorithms)

الگوریتم ژنتیک توسط چارلز داروین بنیان گذاری شده‌است. این الگوریتم‌ها از انتخاب طبیعی افرادی با برتری برای حل مسئله بهینه‌سازی استفاده می‌کنند. قانون انتخاب طبیعی بدین صورت است که تنها گونه‌هایی از یک جمعیت ادامه نسل می‌دهند که بهترین خصوصیات را داشته باشند و آن‌هایی که این خصوصیات را نداشته باشند، به تدریج و در طی زمان از بین می‌روند. در دنیای کامپیوتر، به جای ژنتیک، رشته‌هایی از بیت‌ها و به جای انتخاب طبیعی، تابع برازندگی (fitness function) قرار می‌گیرد.

این الگوریتم دارای پنج مرحله است که به شرح زیر است:

۱. با تولید تصادفی جمعیتی به اندازه N کروموزوم که N اندازه جمعیت و l طول کروموزوم x است، شروع میکنیم.

۲. برای تابع برازندگی، تابع φ(x) را برای هر کروموزوم x محاسبه کنید.

۳. تا زمانی که N فرزند تولید شود، تکرار کنید:

۳-۱. به‌صورت احتمالی یک جفت کروموزوم از جمعیت فعلی با استفاده از تابع برازندگی انتخاب کنید.

۳-۲. با استفاده از عملگرهای تلاقی و جهش، فرزند yi را تولید کنید، جایی که i=1,2,3,…,N

۴. جمعیت فعلی را با جمعیت جدید ایجاد شده جایگزین کنید.

۵. به مرحله ۲ بروید.

در این الگوریتم، با شروع با یک جمعیت تصادفی از کروموزوم‌ها، تابع برازندگی هر کروموزوم محاسبه می‌شود و سپس فرزندان با ترکیب ژن‌های والدین با هم تولید می‌شوند. فرزندان جدید در جایگزینی جمعیت قبلی با جمعیت جدید استفاده می‌شوند. این مراحل تا زمانی که N فرزند تولید شود تکرار می‌شوند. این الگوریتم به عنوان یک الگوریتم بهینه‌سازی در مسائل بهینه‌سازی استفاده می‌شود.

برای تشخیص لبه به کمک الگوریتم ژنتیک، به شیوه زیر عمل می‌کنیم:

۱- در تشخیص لبه، تعداد متغیرها برابر با تعداد ضرایب فیلتر است. برای فیلتر ۳ × ۳، تعداد ضرایب ۹ است. در واقع، فیلتر ۳ × ۳ یک ماتریس ۳ در ۳ است که با آن، عملیات کانولوشن را بر روی تصویر انجام می‌دهیم. این ضرایب از طریق یادگیری ماشین، بهینه‌سازی شده و به منظور تشخیص لبه در تصاویر استفاده می‌شوند.

X_{13}	X_{23}	X_{33}
X_{12}	X_{22}	X_{32}
X_{11}	X_{21}	X_{31}

Mask for Δf_x

X_{11}	X_{12}	X_{13}
X_{21}	X_{22}	X_{23}
X_{31}	X_{32}	X_{33}

Mask for Δf_y

شکل۲-۱۳

۲- در مرحله بعد باید اندازه جمعیت را مشخص کنیم. با افزایش اندازه جمعیت، منطقه جستجو افزایش می‌یابد، اما ممکن است زمان اجرا افزایش یابد. مقداردهی اولیه به $f_{\Delta y}$ اعمال می‌شود در حالی که $f_{\Delta y}$می‌تواند با چرخش $f_{\Delta x}$به اندازه ۹۰ درجه تولید شود.

۳- فیلترهای ماسک بر تصویر در دو جهت اعمال می‌شوند و برای هر پیکسل، قدرت فرکانسی(strength of frequency) محاسبه می‌شود. در واقع، فیلترهای ماسک برای تشخیص لبه و جزئیات تصویر مورد استفاده قرار می‌گیرند و در نهایت، با محاسبه قدرت فرکانسی، می‌توان برای هر پیکسل، شدت سطح خاکستری را تعیین کرد. این قدرت فرکانسی، میزان تغییرات شدت نور (رنگ) در هر نقطه از تصویر را نشان می‌دهد که با استفاده از آن می‌توان لبه‌ها و نقاط قوت تصویر را تشخیص داد.

$$Strengthoffrequency = E(i,j) = \sqrt{[\Delta f_x(i,j)]^2 + [\Delta f_y(i,j)]^2} \qquad (۷)$$

۴- تابع هزینه به عنوان معیاری برای ارزیابی کیفیت جواب‌های به دست آمده در الگوریتم بهینه‌سازی استفاده می‌شود. انتخاب تابع هزینه به عنوان مهم‌ترین مورد در الگوریتم ژنتیک شمرده میشود. انتخاب تابع هزینه بستگی به شرایط مورد نیاز دارد. در تشخیص لبه تصویر، تابع هزینه باید به گونه‌ای انتخاب شود که باعث شود کیفیت تشخیص لبه‌ها بهینه شود. می توان توانایی تشخیص لبه‌ها را با استفاده از انحراف معیار کم بیان کرد. به عبارت دیگر، هدف این است که انحراف معیار تصویر تشخیص داده شده توسط الگوریتم ژنتیک کمترین مقدار را داشته باشد.

$$\cos tfunction = s = \sqrt{\frac{1}{(M*N)-1}\sum_{i=1}^{M}\sum_{i=1}^{N}(X(i,j)-\bar{X})^2} \qquad (۸)$$

$$\bar{X} = \frac{1}{M*N}\sum_{i=1}^{M}\sum_{j=1}^{N}X(i,j) \qquad (۹)$$

در تابع هزینه اولیه که معرفی شد ، هدف کمینه کردن انحراف معیار تصویر است. با این حال، با این تابع هزینه نمی‌توان مقدار دقیقی از انحراف معیار را به دست آورد. در نتیجه، برای به دست آوردن مقدار دقیقی از انحراف معیار، نیاز به تغییر تابع هزینه است. به عبارت دیگر، با تغییر تابع هزینه، هدف این است که انحراف معیار تصویر تشخیص داده شده توسط الگوریتم ژنتیک، به یک مقدار خاص و دلخواه برسد.

(۱۰)

$$\cos tfunction = |s - C|$$

جایی که C مقدار مورد نیاز انحراف معیار است.

۵- در ابتدا، الگوریتم ژنتیک با هدف کمینه کردن تابع هزینه شروع می‌شود. در این مرحله، کمترین هزینه با کروموزوم‌های مرتبط انتخاب شده و سایر کروموزوم‌ها حذف می‌شوند. سپس، عملیات تولید نسل بعدی با استفاده از تولید کروموزوم جدید از روی کروموزوم‌های انتخاب شده و اعمال عملیات تحول (مانند جفت‌گیری و جهش) انجام می‌شود. در نهایت، این نسل جدید به همراه تابع هزینه بررسی می‌شود و همین چرخه به صورت تکراری ادامه می‌یابد تا بهترین پاسخ به دست آید. در واقع، هدف این الگوریتم، بهینه‌سازی تابع هزینه و به دست آوردن بهترین کروموزوم برای رسیدن به حل بهینه مسئله مورد نظر است. تعداد تکرارهای مورد نیاز برای رسیدن به پاسخ بهینه در الگوریتم ژنتیک ممکن است متفاوت باشد و در برخی موارد الگوریتم ممکن است قبل از اتمام ۱۰۰ تکرار، به یک پاسخ بهینه برسد. در واقع، این بستگی به مسئله مورد نظر، اندازه جمعیت، تابع هزینه و سایر پارامترهای مربوط به الگوریتم ژنتیک دارد.

در ادامه تشخیص لبه برای انحراف استاندارد های مختلف نشان داده شده است:

Standard deviation = 0. 2426

Standard deviation = 0.268

Standard deviation = 0. 3458

Standard deviation = 0. 5473

Standard deviation = 0. 6806

Standard deviation = 0. 8200

شکل۲-۱۴
تصویر حاصل
از الگوریتم
ژنتیک

در ادامه یک روش برای تشخیص لبه تصاویر بر اساس اپراتور Sobel بهبود یافته و الگوریتم ژنتیک پیشنهاد شده است.[۷]

همانطور که گفته شده، آستانه‌یابی در الگوریتم‌های تشخیص لبه بسیار اهمیت دارد. آستانه‌یابی به معنی تعیین یک آستانه برای جداسازی پیکسل‌های لبه از پیکسل‌های پس‌زمینه در تصویر است. آستانه می‌تواند به صورت دستی تعیین شود و یا با استفاده از روش‌های خودکار به دست آید. در هر حالت، انتخاب آستانه به شدت تأثیرگذار در کیفیت و دقت تشخیص لبه است. به عبارت دیگر، به دنبال یافتن بهترین آستانه برای تشخیص لبه در تصویر، یکی از چالش‌های اصلی در الگوریتم‌های تشخیص لبه است.

الگوریتم سوبل(Sobel) بر اساس اعمال فیلتر کوچک، جداپذیر و با مقادیر صحیح در جهت‌های افقی و عمودی بر تصویر، اعمال می‌شود. این الگوریتم نسبتاً ارزان در محاسبات است اما تقریب گرادیانی که تولید می‌کند، به خصوص برای تغییرات با فرکانس بالا در تصویر، نسبتاً نامناسب است. الگوریتم سوبل فقط می‌تواند لبه‌های جهت افقی و عمودی را تشخیص دهد و برای تصاویر با بافت پیچیده، عملکرد مناسبی ندارد. برای جبران این محدودیت‌ها، از چهار ماسک با اندازه ۵x۵ استفاده می‌شود. این روش Improved Sobel نام دارد.

$$T_x=\begin{bmatrix}2&3&0&-3&-2\\3&4&0&-4&-3\\6&6&0&-6&-6\\3&4&0&-4&-3\\2&3&0&-3&-2\end{bmatrix}\quad T_y=\begin{bmatrix}2&3&6&3&2\\3&4&6&4&3\\0&0&0&0&0\\-3&-4&-6&-4&-3\\-2&-3&-6&-3&-2\end{bmatrix}$$

$$T_{45}=\begin{bmatrix}0&-2&-3&-2&-6\\2&0&-4&-6&-2\\3&4&0&-4&-3\\2&6&4&0&-2\\6&2&3&2&0\end{bmatrix}\quad T_{135}=\begin{bmatrix}-6&-2&-3&-2&0\\-2&-6&-4&0&2\\-3&-4&0&4&2\\-2&0&4&6&2\\0&2&3&2&6\end{bmatrix}$$

شکل۲-۱۵
ماسک‌های سوبل بهبودیافته

وزن هر مکان در الگوریتم با توجه به فاصله آن از مرکز و جهت مکان تعیین می‌شود. نقاطی که فاصله آن‌ها از مرکز برابر است، وزن یکسانی دارند. با استفاده از این روش، نسبت به الگوریتم Sobel، تشخیص لبه در تصاویر با بافت پیچیده و دارای نویز بهتر می‌شود.

در ادامه قبل از اینکه به روش لبه یابی بپردازیم، ابتدا مروری بر الگوریتم اتسو(OTSU) انجام می‌دهیم.

$$\sigma(t)^2=\omega_0(t)*\varpi_1(t)*(\mu_0(t)-\mu_1(t))^2 \qquad (۱۱)$$

آن t که بزرگترین واریانس را میدهد، بهترین آستانه برای تقسیم بندی است.

برای یافتن بهترین مقدار آستانه، باید تمامی مقادیر خاکستری پیکسل‌ها در دامنه تعیین شده بررسی شوند و واریانس آن‌ها محاسبه شود. در نهایت باید مقدار آستانه‌یی که بهترین واریانس را دارد پیدا شود. محاسبه این واریانس هزینه محاسباتی بسیار بالا و کارآیی پایینی دارد. تعیین مقدار آستانه بهینه در نهایت یک مسئله بهینه‌سازی است. الگوریتم ژنتیک می‌تواند کارایی محاسباتی را بهبود بخشد.

جزئیات الگوریتم به صورت زیر می‌باشد:

۱- با استفاده از چهار ماسکی که در روش Improved Sobel معرفی شدند، تصویر گرادیان را محاسبه کنیم.

۲- بهترین آستانه برای تقسیم‌بندی تصویر گرادیان را به روش Otsu بر اساس الگوریتم ژنتیک بدست آوریم. این الگوریتم شامل شش مرحله به شرح زیر است:

۲-۱ رمزگذاری(encoding) کروموزوم درالگوریتم ژنتیک: برای تعیین مقدار آستانه‌ی مناسب برای تقسیم تصویر، هر کروموزوم با یک رشته‌ی ۸ بیتی از اعداد دودویی رمزگذاری می‌شود که به عنوان مقدار آستانه به کار می‌رود.

۲-۲ در الگوریتم ژنتیک، ایجاد جمعیت اولیه یکی از مراحل مهم است. در این مرحله، جمعیت اولیه شامل تعدادی کروموزوم تصادفی ایجاد می‌شود.

۲-۳ در الگوریتم ژنتیک، تابع برازندگی یا fitness function، به عنوان معیاری برای ارزیابی کروموزوم‌ها و تعیین اینکه کدام کروموزوم‌ها بهترین عملکرد را از خود نشان داده‌اند، استفاده می‌شود.

در اینجا، تابع برازندگی برای تعیین بهترین مقدار آستانه برای تقسیم تصویر به بخش‌های مختلف، براساس واریانس بیشینه بین کلاس‌های مختلف محاسبه می‌شود. به عبارت دیگر، بیشینه کردن واریانس بین کلاس‌های مختلف با استفاده از مقدار آستانه‌ای که کروموزوم فعلی نشان می‌دهد، به عنوان تابع برازندگی استفاده می‌شود.

۲-۴ درالگوریتم ژنتیک، عملگر چندرشته‌ای یا crossover operator برای ترکیب کروموزوم‌های دو نفر در جمعیت استفاده می‌شود. این عملگر به منظور ایجاد نسل جدیدی از کروموزوم‌ها با ترکیب صفات دو کروموزوم مختلف به کار می‌رود.

در اینجا از روش چندرشته‌ای نقطه‌ای یا single-point crossover برای ترکیب کروموزوم‌ها استفاده می‌شود. احتمال ترکیب در این روش برابر با ۰/۹ می‌باشد، به این معنا که در هر دور از الگوریتم ۹۰ درصد از جفت کروموزوم‌های جدید با استفاده از این عملگر تولید می‌شوند.

۲-۵ در الگوریتم ژنتیک، عملگر جهش یا mutation operator برای تغییر تصادفی یک یا چند بیت از کروموزوم استفاده می‌شود. این عمل باعث می‌شود که جستجو در فضای جستجوی الگوریتم بیشتر تنوع پیدا کرده و از گیر افتادن در یک نقطه ثابت جلوگیری شود.

نتایج آزمایش‌ها نشان می‌دهد که پایداری حل‌های الگوریتم ژنتیک زمانی بهتر است که احتمال جهش بسیار کوچک باشد. با این حال، یکی از مشکلات این روش این است که اگر الگوریتم وارد یک نقطه محلی از جستجو شود، خارج شدن از آن بسیار سخت می‌شود و این موجب گرایش زودهنگام الگوریتم به یک نقطه محلی می‌شود. بنابراین، بهتر است احتمال جهش کم باشد. اما اگر احتمال جهش بسیار کم باشد، عملکرد الگوریتم به شدت کاهش می‌یابد و تبدیل به جستجوی تصادفی می‌شود. بنابراین، احتمال جهش به صورت تطبیقی تعیین می‌شود تا عملکرد الگوریتم بهبود یابد.

(احتمال جهش نشان دهنده‌ی این است که در هر نسل از الگوریتم ژنتیک، چه میزان از کروموزوم‌های جدید تغییر می‌کنند.)

(۱۲)

$$p_m = \begin{cases} 0.1, \dfrac{f_{\max} - f}{f_{\max} - \bar{f}} > 1 \\ 0.002, \dfrac{f_{\max} - f}{f_{\max} - \bar{f}} < 1 \end{cases}$$

با توجه به این فرمول، احتمال جهش هر کروموزوم در نسل جدید، براساس مقدار تابع برازندگی آن کروموزوم و میانگین و بیشینه تابع برازندگی در جمعیت فعلی محاسبه می‌شود. این مقدار جدید احتمال جهش، به منظور افزایش تنوع جستجو و جلوگیری از گیر افتادن در یک نقطه محلی، در نسل بعدی از الگوریتم ژنتیک استفاده می‌شود.

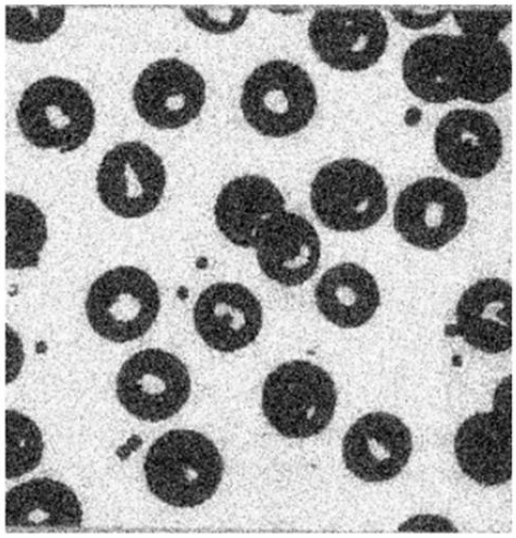
۲-۶ بعد از اینکه الگوریتم ژنتیک به تعداد مشخص شده‌ای از نسل‌ها رسید، باید بالاترین تابع برازندگی یا fitness function کروموزوم به دست آمده، برای به دست آوردن بهترین آستانه‌ای که برای تقسیم تصویر به بخش‌های مختلف به کار می‌رود، استفاده شود.

برای این منظور، ابتدا باید کروموزومی با بالاترین تابع برازندگی را انتخاب کنیم. سپس این کروموزوم را با استفاده از فرایند رمزگشایی، به بهترین آستانه‌ای که برای تقسیم تصویر به بخش‌های مختلف مناسب است، تبدیل می‌کنیم.

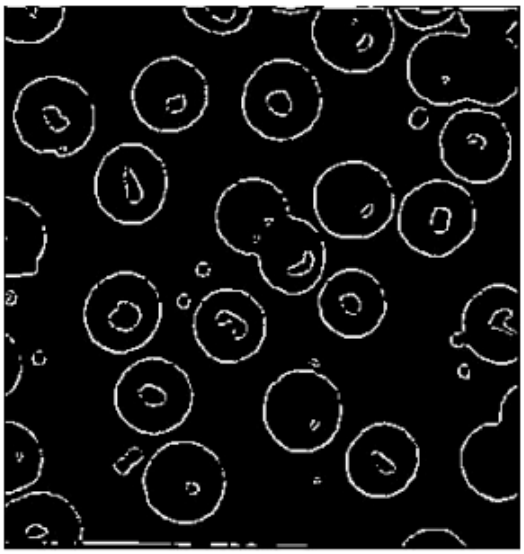
۳- براساس آستانه به دست آمده، اگر مقدار خاکستری تصویر بزرگتر یا مساوی با آستانه باشد، می‌توان آن نقطه را به عنوان نقطه لبه اولیه تعیین کرد. در غیر این صورت، آن نقطه به عنوان نقطه‌ای که لبه ندارد تشخیص داده می‌شود.

به طور کلی، الگوریتم جدید به عنوان یک روش بهبود یافته برای تشخیص لبه در تصاویر است، که علت بهبود آن نسبت به روش‌های قبلی، از جمله الگوریتم Sobel، قابلیت ضد نویز، سرعت محاسبات مناسب و دقت بیشتر در تشخیص لبه‌ها است.

الگوریتم جدید پیشرفتی نسبت به الگوریتم‌های قبلی در تشخیص لبه‌های تصویر است، اما هنوز می‌تواند بهبود یابد و اپراتورهای بهتری برای تشخیص لبه‌ها در آینده توسعه داده خواهند شد.



شکل۲-۱۶ تصویر اصلی



شکل۲-۱۷تصویر حاصل ازآستانه‌گذاری به کمک روش ژنتیک

فصل سوم:
روش‌های ارزیابی

۳-۱ روش ارزیابی

برای تحلیل نتایج از مفهوم فاصله همینگ استفاده شده است.[۴] فاصله همینگ(Hamming Distance) بیت هایی را که در دو تصویر یکسان هستند را اندازه‌گیری می‌کند. در این روش، بیت‌های تصاویر مقایسه می‌شوند و سپس تصمیم می‌گیرد که این تصاویر یکسان هستند یا خیر. در این روش، مقدار خروجی بین ۰ تا ۱ متغیر است. اگر این مقدار به ۰ همگرا شود، درصد شباهت افزایش می یابد. بنابراین، اگر تصاویر یکسان با این روش مقایسه شوند، نتیجه به صفر میل می‌کند. تصاویر ورودی X و Y با استفاده از فاصله همینگ با معادله (۱۳) مقایسه می‌شوند. N تعداد بیت در تصویر است و به عبارت دیگر اندازه بردار ویژگی است. XOR عملگر منطقی است و Xj و Yj تصاویر ورودی هستند که مورد بررسی قرار می گیرند.

$$HD = \frac{1}{N}\sum_{j=1}^{N} X_j (XOR) Y_j \qquad (۱۳)$$

فصل چهارم:
نتیجه‌گیری

٤-١ نتیجه‌گیری

در مسائل پردازش تصویر، لبه‌یابی یکی از مهمترین و پرکاربردترین فعالیت‌ها است. برای بهبود دقت و کارایی لبه‌یابی، از روش‌های بهینه‌سازی مانند الگوریتم مورچگان، الگوریتم زنبور مصنوعی و الگوریتم ژنتیک استفاده می‌شود.

الگوریتم کلونی مورچگان یک الگوریتم بهینه‌سازی است. با اضافه کردن، تابع عضویت مثلثی فازی (fuzzy triangle membership function) به این الگوریتم، می‌توان آن‌را بهبود داد و به دقت و کارایی قابل توجهی در لبه‌یابی رسید.

الگوریتم کلونی زنبور مصنوعی نیز یک الگوریتم بهینه‌سازی است که بر مبنای رفتار زنبورها در طبیعت طراحی شده است. یکی از مزایای این الگوریتم، این است که در هر تکرار، جستجوی محلی و کلی را انجام می‌دهد و بسیار سریع و آسان برای پیاده‌سازی است. برخی تحقیقات نشان داده که این الگوریتم بهتر از الگوریتم بهینه‌سازی کلونی مورچه در تشخیص لبه تصاویر عمل می‌کند و همچنین زمان کمتری نیز برای اجرای آن نیاز است.

الگوریتم ژنتیک نیز یک الگوریتم بهینه‌سازی است که بر مبنای روش کارکرد ژنتیک در طبیعت طراحی شده است. در لبه‌یابی، این الگوریتم می‌تواند به ما در پیدا کردن بهترین آستانه برای تقسیم‌بندی تصویر کمک کند.

بنابراین، استفاده از روش‌های بهینه‌سازی مانند الگوریتم مورچگان، الگوریتم زنبور مصنوعی و الگوریتم ژنتیک می‌تواند بهبود قابل توجهی در دقت و کارایی لبه‌یابی ایجاد کند. البته انتخاب روش بهینه‌سازی مناسب بستگی به مسئله و داده‌های مورد بررسی دارد و نیاز به آزمایش و مقایسه روش‌های مختلف دارد.

اما با توجه به نتایجی که از روش‌های مختلف مشاهده شد، روش تشخیص لبه تصاویر بر اساس اپراتور Sobel بهبود یافته و الگوریتم های ژنتیک، به دلیل ایجاد لبه‌های نازک و منحنی های پیوسته، نتایج خیلی بهتری نسبت به دیگر روش‌ها ایجاد کرد.

منابع

الف)منابع انگلیسی:

١-Hossein Nezamabadi-pour,Saeid Saryazdi, Esmat Rashedi, *Edge detection using ant algorithms, Soft Comput (2006) 10: 623–628*
DOI 10.1007/s00500-005-0511-y

2- P.M.Daigavane, P.R.Bajaj , *Road Lane Detection with Improved Canny Edges Using Ant Colony Optimization* , Third International Conference on Emerging Trends in Engineering and Technology,2010, doi/10.1109/ICETET.2010.128

3- R. Rajeswari and R. Rajesh, "A modified ant colony optimization based approach for image edge detection," 2011 International Conference on Image Information Processing, Shimla, India, 2011, pp. 1-6, doi: 10.1109/ICIIP.2011.6108930.

4-Elif Deniz Yigitbasi , Nurdan Akhan Baykan, *Edge Detection using Artificial Bee Colony Algorithm, International Journal of Information and Electronics Engineering, Vol. 3, No. 6, November 2013*

5- Y. Liu and S. Tang, "An application of artificial bee colony optimization to image edge detection," 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Guilin, China, 2017, pp. 923-929, doi: 10.1109/FSKD.2017.8393400.

٦- Riyadh Mwad Naife , Hawrra Hassan Abass, "*OPTIMAL EDGE DETECTION FILTER USING GENETIC ALGORITHM",*
Journal of Kerbala University , Vol. 13 No.1 Scientific . 2015

7- Zhang Jin-Yu, Chen Yan and Huang Xian-Xiang, "Edge detection of images based on improved Sobel operator and genetic algorithms," 2009 International Conference on Image Analysis and Signal Processing, Linhai, China, 2009, pp. 31-35, doi: 10.1109/IASP.2009.5054605