

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده برق و رباتیک

گروه برق - الکترونیک

عنوان:

تعیین پارامترهای مناسب در روش گراف کات برای ناحیه بندی تعاملی تصویر

دانشجو:

سمانه ابارشی

استاد راهنما:

دکتر علیرضا احمدی فرد

استاد مشاور:

دکتر علی سلیمانی

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

اردیبهشت ۱۳۸۹

تعهد نامه

اینجانب، دانشجو، دوره کارشناسی ارشد / دکتری رشته دانشگاه صنعتی شاهرود نویسنده پایان نامه / رساله تحت راهنمایی متعهد می شوم.

- تحقیقات در این پایان نامه / رساله توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطابق مدرج در این نامه / رساله تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جایزه نشده است.
- کسب حقوق معنوی من اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و « Shahrood University of Technology » به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه / رساله تأثیرگذار بوده اند در مقالات مستخرج از این نامه / رساله رعایت می گردد.
- کسب هر منافع انجام این پایان نامه / رساله ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در نامه مرجع این پایان نامه / رساله ، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده کرده است، این را رعایت نموده و ضوابط و اصول اخلاقی رعایت شده است.

تاریخ: ۸۸/۱/۱۸

امضای دانشجو

مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه های رایانه ای، نرم افزارها و غیره) محفوظ شده است و متعلق به دانشگاه صنعتی شاهرود می باشد. این مطلب باید به نحو مقتضی در پایان نامه درج و درج شود.
- استفاده از اطلاعات و نتایج موجود در پایان نامه / رساله بدون ذکر مرجع مجاز نمی باشد.

* من این صفحه نیز باید در ابتدا ی نسخه های تکثیر شده پایان نامه / رساله وجود داشته باشد.



مدیریت تحصیلات تکمیلی

فرم شماره ۶

بسمه تعالی

فرم صورتجلسه دفاع پایان نامه تحصیلی دوره کارشناسی ارشد

اینست حدوث متعل و به استعدت از حضرت وی عصر (عج) جلسه دفاع از پایان نامه کارشناسی

رشته: حلیه: سمانه ابارشی رشته: برق گریس: الکترونیک

بحث عنوان: تعیین پارامترهای مناسب در روش گراف کات برای ناحیه بندی تعاملی تصویر

که در تاریخ ۱۳۸۹/۰۲/۰۸ در حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح

زیر است:

بجمله در مقصد و مقادیر پنج صدم

قبول / درجه: <u>بسیار خوب</u> امتیاز: <u>۱۸/۷۵</u>	☐ مردود	☐ دفاع مجدد
--	--------------------------------	------------------------------------

۲- بسیار خوب (۱۸-۱۸/۹۹)

۱- عالی (۱۹-۲۰)

۴- قابل قبول (۱۵/۹۹-۱۶)

۳- خوب (۱۶-۱۶/۹۹)

عضو هیأت داوران	نام و نام خانوادگی	مرتبه علمی	امضاء
۱- استاد راهنما	عبدالله احمدی فرد	استادیار	
۲- استاد مشاور	—	—	—
۳- نماینده شورای تحصیلات تکمیلی	محمد حداد طرفی	استادیار	
۴- استاد منتظر	عباس رضوی	استاد	
۵- استاد منتظر	ایرج میردکی	استادیار	

تأیید رئیس دانشکده:

تقدیم با تمام وجود

به پدر عزیزم:

او که بی دریغ کوشید تا دانش بجویم

به مادر مهربانم:

که فروغ محبتش شور انگیز و امیدبخش است

به همسر عزیزم:

که با صبر و سکینایی مشکلات تحصیل را تحمل نمود

و به دو خواهر و برادرم

که باد لکرمی هایشان مرایاری کردند.

سپاس بی کران خداوند متعال که مرا فرصت اندیشیدن داد.

بر خود لازم می دانم از استاد کراتقدم

جناب آقای دکتر علیرضا احمدی فرد

,

تمامی استیادگرامی که مرا راهنمایی کردند کمال تشکر و

قدردانی را داشته باشم.

چکیده

ناحیه بندی تصاویر اولین گام و یکی از مهمترین بخش های یک سیستم بینایی ماشین می باشد. ناحیه بندی تصاویر را می توان در دو حوزه ناحیه بندی اتوماتیک و ناحیه بندی تعاملی بررسی نمود. در این پایان نامه ناحیه بندی تعاملی مبتنی بر گراف کات را مطالعه می نمائیم. موفقیت ناحیه بندی مبتنی بر گراف کات در گرو انتخاب پارامترهای مناسب برای این الگوریتم است. برای هر تصویر مجموعه بهینه ای از پارامترها وجود دارند که به ازای این پارامترها بهترین جواب های ناحیه بندی به دست می آیند. هدف ما در این پایان نامه ارزیابی خودکار نتیجه ناحیه بندی می باشد. برای ارزیابی نتایج ناحیه بندی گراف کات، دو سیستم به کمک کلاسه بندهای Adaboost و شبکه عصبی پیشنهاد می کنیم.

در هر کدام از سیستم های پیشنهادی، کلاسه بند توسط مجموعه تصاویر ناحیه بندی شده آموزشی، آموزش داده می شود. کلاسه بند آموزش داده شده قادر به ارزیابی یک تصویر ناحیه بندی شده آزمایشی می باشد. ویژگی های استخراج شده از هر تصویر ناحیه بندی شده چه در فاز آموزش و چه در فاز آزمایش، شش ویژگی می باشد. این ویژگی ها خواص همگنی نواحی و تمایز مرزها را از لحاظ روشنایی و بافت نشان می دهند. خطای کلاسه بند Adaboost در ارزیابی جواب های ناحیه بندی % ۱/۷ و مقدار این خطا برای کلاسه بند شبکه عصبی % ۰/۶۵ می باشد.

کلید واژه : ناحیه بندی تعاملی تصاویر، الگوریتم گراف کات، کلاسه بند Adaboost، کلاسه بند شبکه عصبی.

فهرست مطالب

صفحه	عنوان
۱	فصل اول : مقدمه
۲	۱-۱ مقدمه
۳	۲-۱ ناحیه بندی تصاویر
۴	۳-۱ تفکیک الگوریتم های ناحیه بندی
۷	فصل دوم : مروری بر کارهای گذشته و معرفی روش ناحیه بندی گراف کات
۸	۱-۲ مقدمه
۸	۲-۲ آستانه گذاری
۹	۳-۲ رشد ناحیه
۱۰	۴-۲ مدل مار
۱۱	۵-۲ برش های هوشمند
۱۲	۶-۲ مروری بر الگوریتم گراف کات
۱۳	۱-۶-۲ تئوری گراف کات
۱۶	۲-۶-۲ گراف کات ها و بینایی ماشین
۱۹	۳-۶-۲ انتخاب پارامترها
۲۲	۴-۶-۲ مینیمم کات روی گراف
۲۴	۵-۶-۲ تئوری Max-Flow/Min-Cut

۲۵	۶-۶-۲ مسائل ماکزیمم فلو یا مینیمم کات
۲۵	۶-۶-۲-۱ گراف جهتدار
۲۸	۶-۶-۲-۲ ماکزیمم فلو
۳۶	۶-۶-۲ الگوریتم های استاندارد در بهینه سازی ترکیبی
۳۷	۶-۶-۲ الگوریتم جدید ماکزیمم فلو یا مینیمم کات
۳۸	۶-۶-۲-۱ مراحل اجرای الگوریتم
۴۰	۶-۶-۲-۱-۱ مرحله رشد
۴۱	۶-۶-۲-۱-۲ مرحله تقویت
۴۲	۶-۶-۲-۱-۳ مرحله پذیرش
۴۴	۶-۲ مقایسه و تحلیل نتایج
۴۸	۶-۲ نتیجه گیری
۴۹	فصل سوم : ارزیابی خودکار ناحیه بندی توسط الگوریتم Adaboost
۵۰	۳-۱ مقدمه
۵۱	۳-۲ ویژگی های ناحیه بندی مناسب
۵۲	۳-۲-۱ ویژگی های مبتنی بر شدت روشنایی در ناحیه بندی
۵۳	۳-۲-۲ ویژگی زاویه گرادیان در مرز نواحی
۵۴	۳-۲-۳ ویژگی گوشه های مرز
۵۵	۳-۲-۴ ویژگی بافت

۵۷	۳-۳ الگوریتم Adaboost
۵۸	۱-۳-۳ کلاسه بندهای ساده
۵۹	۲-۳-۳ آموزش Adaboost
۶۲	۴-۳ ارزیابی روش پیشنهادی با نتایج ناحیه بندی
۶۲	۱-۴-۳ نتایج آموزش کلاسه بند
۶۶	۲-۴-۳ نتایج ناحیه بندی
۷۱	۵-۳ نتیجه گیری
۷۲	فصل چهارم : کلاسه بندی خودکار تصاویر ناحیه بندی شده توسط شبکه عصبی
۷۳	۱-۴ مقدمه
۷۴	۲-۴ ساختار نورون
۷۵	۱-۲-۴ توابع فعال ساز
۷۶	۳-۴ ساختار شبکه عصبی
۷۸	۴-۴ توپولوژی شبکه عصبی
۷۹	۵-۴ آموزش شبکه عصبی
۸۰	۱-۵-۴ آموزش شبکه عصبی با الگوریتم انتشار به عقب
۸۳	۶-۴ کلاسه بند شبکه عصبی برای ارزیابی خودکار تصاویر ناحیه بندی شده
۸۴	۱-۶-۴ آموزش کلاسه بند
۸۵	۲-۶-۴ نتایج ناحیه بندی

۸۶	۳-۶-۴ مقایسه و تحلیل نتایج
۹۰	۷-۴ مقایسه ارزیابی های کلاسه بندهای Adaboost و شبکه عصبی
۹۰	۸-۴ نتیجه گیری
۹۱	فصل پنجم : نتیجه گیری و پیشنهادات
۹۲	۱-۵ نتیجه گیری
۹۳	۲-۵ پیشنهادی برای ادامه تحقیق
۹۴	مراجع

فهرست شکل ها

صفحه	عنوان
	فصل دوم : مروری بر کارهای گذشته و معرفی روش ناحیه بندی گراف کات
۱۸	شکل (۱-۲) : مثالی ساده برای ناحیه بندی تصویر دو بعدی 3×3
۲۰	شکل (۲-۲) : ناحیه بندی یک تصویر به ازای λ های مختلف
۲۱	شکل (۳-۲) : ناحیه بندی یک تصویر به ازای σ های مختلف
۲۵	شکل (۴-۲) : (a) تصویر اصلی ، (b) یک نمونه برچسب گذاری
۲۶	شکل (۵-۲) : نمونه ای از یک گراف جهتدار
۲۷	شکل (۶-۲) : نمونه ای از یک کات روی گراف جهتدار
۳۰	شکل (۷-۲) : یافتن ماکزیمم فلو از مبدا A به مقصد G
۳۱	شکل (۸-۲) : یافتن ماکزیمم فلو در طول مسیر A-D-E-G
۳۲	شکل (۹-۲) : یافتن ماکزیمم فلو در طول مسیر A-B-E-G
۳۲	شکل (۱۰-۲) : یافتن ماکزیمم فلو در طول مسیر A-C-F-G
۳۲	شکل (۱۱-۲) : یافتن ماکزیمم فلو در طول مسیر A-D-F-G
۳۳	شکل (۱۲-۲) : یافتن ماکزیمم فلو در طول مسیر A-D-F-E-G
۳۴	شکل (۱۳-۲) : فلوهای به دست آمده در هر مسیر
۳۴	شکل (۱۴-۲) : چند نمونه از کات های ممکن روی گراف
۳۵	شکل (۱۵-۲) : مینیمم کات روی گراف

شکل (۲-۱۶) : نمونه ای از درختان جستجو $\mathcal{S}$ و $\mathcal{T}$ را در انتهای مرحله رشد نشان می دهد و مسیر یافت شده از ترمینال S به t مشخص شده است.

۳۸

شکل (۲-۱۷) : ساختار کلی الگوریتم جدید

۳۹

شکل (۲-۱۸) : الگوریتم مرحله رشد

۴۰

شکل (۲-۱۹) : الگوریتم مرحله تقویت

۴۱

شکل (۲-۲۰) : الگوریتم مرحله پذیرش

۴۲

شکل (۲-۲۱) : (a) تصویر اصلی به همراه بذرها ، (b) تصویر ناحیه بندی شده به ازای $\lambda = 0.001$ و $\sigma = 10$ ، (c) تصویر ناحیه بندی شده به ازای $\lambda = 0.101$ و $\sigma = 9$ ، (d) تصویر ناحیه بندی شده به ازای $\lambda = 0.801$ و $\sigma = 7$

۴۵

شکل (۲-۲۲) : (a) تصویر اصلی به همراه بذرها ، (b) تصویر ناحیه بندی شده به ازای $\lambda = 0.001$ و $\sigma = 10$ ، (c) تصویر ناحیه بندی شده به ازای $\lambda = 0.101$ و $\sigma = 9$ ، (d) تصویر ناحیه بندی شده به ازای $\lambda = 0.801$ و $\sigma = 7$

۴۶

شکل (۲-۲۳) : (a) تصویر اصلی به همراه بذرها ، (b) ناحیه بندی نامطلوب تصویر (a) به ازای $\lambda = 0.001$ و $\sigma = 10$ ، (c) تصویر اصلی به همراه بذرها، جدید ، (d) ناحیه بندی مطلوب تصویر (c) به ازای $\lambda = 0.001$ و $\sigma = 10$

۴۷

فصل سوم : ارزیابی خودکار ناحیه بندی توسط الگوریتم Adaboost

شکل (۳-۱) : مراحل کلاسه بندی

۵۱

شکل (۳-۲) : (a) تصویر اصلی (b) ناحیه بندی تصویر اصلی به ازای $\lambda = 0.5$ و $\sigma = 2$ با مرزهای پیچیده ، (c) ناحیه بندی تصویر اصلی به ازای $\lambda = 0.001$ و $\sigma = 3$ با مرزهای ساده

۵۴

شکل (۳-۳) : ساختار کلی Adaboost

۵۷

شکل (۳-۴) : آستانه مناسب جایی است که هستوگرام های تجمعی دو کلاس بیشترین اختلاف را با یکدیگر دارند

۵۹

شکل (۳-۵) : الگوریتم Adaboost

۶۰

شکل (۳-۶) : (a-f) به ترتیب هستوگرام های تجمعی و آستانه های متناظر با هر یک از ویژگی های شدت روشنایی ۱، شدت روشنایی ۲، جهت گرادیان ۱، جهت گرادیان ۲، تعداد گوشه ها و بافت را نشان می دهد

۶۴

شکل (۳-۷) : خطای آموزش برای کلاسه بند Adaboost

۶۶

شکل (۳-۸) : (a) تصویر اصلی به همراه بذرها، (b) ground truth هر تصویر، (c) ناحیه بندی بهینه هر تصویر

۶۸

شکل (۳-۹) : (a) تصویر اصلی به همراه بذرها، (b) ground truth هر تصویر، (c) ناحیه بندی بهینه هر تصویر

۶۹

شکل (۳-۱۰) : (a) تصویر اصلی به همراه بذرها، (b) ground truth هر تصویر، (c) ناحیه بندی بهینه هر تصویر

۷۰

فصل چهارم: کلاسه بندی خودکار تصاویر ناحیه بندی شده توسط شبکه عصبی

شکل (۴-۱) : مدل مصنوعی یک نورون

۷۴

شکل (۴-۲) : شبکه عصبی ساده با سه لایه

۷۷

شکل (۴-۳) : خطای مراحل آموزش، اعتبارسنجی و آزمون در ۱۶ تکرار نشان داده شده است

۸۵

شکل (۴-۴) : (a) تصویر اصلی به همراه بذرها، (b) ground truth هر تصویر، (c) ناحیه بندی بهینه هر تصویر

۸۷

شکل (۴-۵) : (a) تصویر اصلی به همراه بذرها، (b) ground truth هر تصویر، (c) ناحیه بندی بهینه هر تصویر

۸۸

شکل (۴-۶) : (a) تصویر اصلی به همراه بذرها، (b) ground truth هر تصویر، (c) ناحیه بندی بهینه هر تصویر

۸۹

فهرست جداول

صفحه

عنوان

فصل دوم : مروری بر کارهای گذشته و معرفی روش ناحیه بندی گراف کات

۱۷

جدول (۱-۲) : وزن های لبه ها

فصل اول

مقدمه

۱-۱ مقدمه

تاریخچه ناحیه بندی تصاویر دیجیتال توسط رایانه به بیش از چهل سال پیش برمی گردد. از آن زمان تاکنون پیشرفت های زیادی در این زمینه صورت گرفته است [۱]. اما به دلیل اهمیت و کاربردهای خاص این موضوع، هنوز هم یکی از زمینه های تحقیقاتی مهم به شمار می آید که نظر بسیاری از محققین را به خود جلب کرده است. علاوه بر این با پیشرفت سریع تکنولوژی در عرصه های مختلف همه روزه کاربردهای جدیدی برای این موضوع یافت می شود.

ناحیه بندی تصویر یکی از مهمترین اجزاء یک سیستم مبتنی بر بینایی ماشین و پردازش تصویر می باشد. در این مرحله تصویر به نواحی تشکیل دهنده آن تفکیک می شود [۲]. در سال ۱۹۶۵ عملگر روبرت^۱ که یک اپراتور لبه یاب است برای پیدا کردن لبه بین دو ناحیه مختلف یک تصویر طراحی شد. از آن زمان تاکنون الگوریتم های بسیاری برای حل این موضوع ارائه شده است. با این وجود هنوز روش مناسبی برای ناحیه بندی ایده آل تصاویر ارائه نشده است [۱].

ناحیه بندی تصاویر را می توان در دو حوزه ناحیه بندی خودکار^۲ و ناحیه بندی تعاملی^۳ بررسی نمود. در روش ناحیه بندی خودکار کاربر هیچ تعاملی در پیشبرد ناحیه بندی نداشته و تصاویر به صورت خودکار طی الگوریتم هایی به نواحی تشکیل دهنده تفکیک می شوند. در کاربردهایی نظیر بازیابی^۴ و شاخص گذاری چند رسانه ای^۵ که سرعت انجام عملیات حائز اهمیت است می توان از این روش استفاده نمود. اما در مواردی که می خواهیم شی^۶ را از پس زمینه^۷ جدا نماییم دقت بیشتری نیاز است لذا این روش مناسب نمی باشد. به همین دلیل این روش هنوز هم به عنوان یکی از زمینه های

^۱ Robert

^۲ Automatic segmentation

^۳ Interactive segmentation

^۴ retrieval

^۵ multimedia indexing

^۶ object

^۷ background

تحقیقاتی مهم به شمار می آید. اما در روش ناحیه بندی تعاملی، تعاملات کاربر تاثیر زیادی در نتیجه ناحیه بندی دارد بدین ترتیب که کاربر نقاطی از تصویر را به عنوان شی و پس زمینه انتخاب کرده و با انتخاب این نقاط ناحیه بندی شروع می شود [۲]. پس از انجام ناحیه بندی، تصویر به نواحی تشکیل دهنده تقسیم می شود. هر یک از این نواحی تا حد امکان باید از نظر ویژگی هایی نظیر روشنایی^۱، رنگ، بافت^۲ و ... همگن باشند.

روش ناحیه بندی تعاملی موضوع اصلی مورد بحث در این تحقیق می باشد.

۱-۲ ناحیه بندی تصاویر

ناحیه بندی تصویر به معنای تقسیم کردن یک تصویر به نواحی تشکیل دهنده آن می باشد به طوری که این نواحی هیچ اشتراکی با یکدیگر نداشته باشند و مجموع آن ها تصویر اولیه را تشکیل دهد. ویژگی های یک ناحیه بندی خوب را می توان به صورت زیر بیان کرد:

۱. هر ناحیه باید از نظر ویژگی هایی نظیر بافت، رنگ و روشنایی همگن باشد.
۲. هر ناحیه نسبت به نواحی همسایه باید بیشترین اختلاف را در ویژگی همگنی داشته باشد.

با توجه به ویژگی های فوق یک تعریف عام برای ناحیه بندی تصویر I که $R_1, R_2, \dots, R_n$ بیانگر هر یک از نواحی تشکیل دهنده آن می باشد، به صورت زیر مطرح می شود:

$$1. \quad \bigcup_{i=1}^n R_i = I$$

$$2. \quad \text{به ازای هر } i \text{ و } j \text{ که } i \neq j \text{ داریم: } R_i \cap R_j = \emptyset$$

$$3. \quad \text{برای } i = 1, 2, \dots, n \text{ داریم: } P(R_i) = 1$$

^۱ intensity
^۲ texture

۴. برای $j \neq i$ داریم $P(R_i \cup R_j) = 0$.

که $P(R_i)$ یکنواختی تمام المان هایی است که به R_i نسبت داده می شود.

۵. برای همه i ها $i = 1, 2, \dots, n$ یک جزء به هم پیوسته است.

برای توضیح بیشتر شروط بالا می توان گفت که شرط ۱ تاکید می کند که مجموع نواحی مختلف در تصویر ناحیه بندی شده باید تصویر اصلی را پوشش دهد، شرط ۲ به عدم تداخل دو ناحیه مختلف در یک تصویر ناحیه بندی شده اشاره می کند. شرط ۳ چنین بیان می کند که پیکسل های مربوط به یک ناحیه باید در بعضی از ویژگی ها مشابه باشند که در نتیجه آن یکنواختی تصویر در یک ناحیه برابر با ۱ می شود. شرط ۴ بیان می کند که هر دو ناحیه متفاوت از نظر ویژگی ها با یکدیگر اختلاف دارند و از کنار هم گذاشتن آن ها ناحیه یکپارچه ای ایجاد نمی گردد. نهایتاً شرط ۵ نیز بیان می کند که پیکسل های مربوط به یک ناحیه باید به هم پیوسته باشند [۳][۴][۵].

۳-۱ تفکیک الگوریتم های ناحیه بندی

الگوریتم های ناحیه بندی را می توان به ۶ دسته کلی تقسیم بندی نمود. این تقسیم بندی در

مرجع [۲] به صورت زیر بیان شده است:

۱. آستانه گذاری^۱

۲. گسترش ناحیه^۲

۳. کلاسه بندی

۴. گراف کات

^۱ Thresholding
^۲ Region growing

۵. ناحیه بندی تعاملی با استفاده از درختان تقسیم باینری

۶. کانتورهای فعال^۱

آنچه باعث تفاوت عملکرد این الگوریتم ها می باشد ویژگی های مورد استفاده در هر الگوریتم و استراتژی الگوریتم برای کلاسه بندی پیکسل ها بر اساس ویژگی های استخراج شده می باشد. از میان روش های فوق روش مبتنی بر گراف کات که اخیراً مورد توجه قرار گرفته است را در این تحقیق مورد مطالعه قرار می دهیم.

هدف اصلی این تحقیق ارائه روشی است که با استفاده از آن بتوان نتایج ناحیه بندی را به صورت خودکار ارزیابی کرد. در این صورت چنانچه نتیجه ناحیه بندی مطلوب نباشد به صورت خودکار تشخیص داده و با تغییر پارامترهای ناحیه بند اقدام به ناحیه بندی مجدد می کنیم. بدین منظور سعی شده با کلاسه بندی نتایج ناحیه بندی که توسط الگوریتم گراف کات و به ازای پارامترهای مختلف آن به دست آمده اند بهترین ناحیه بندی ها را برای هر تصویر به دست آوریم. در روش های پیشنهادی برای کلاسه بندی تصاویر ناحیه بندی شده از کلاسه بندهای Adaboost و شبکه عصبی استفاده می کنیم.

در فصل دوم به مرور چند روش مهم در ناحیه بندی می پردازیم ، مزایا و معایب هر روش را بررسی می کنیم. سپس روش ناحیه بندی تعاملی مبتنی بر گراف کات را معرفی می کنیم. در این روش با تغییر پارامترها (σ و λ) نتایج ناحیه بندی تغییر می کند.

در فصل سوم کلاسه بند Adaboost را معرفی می کنیم. در این فصل روش جدیدی جهت ارزیابی خودکار نتایج ناحیه بندی شده با استفاده از این کلاسه بند ارائه می دهیم. برای این منظور ویژگی هایی مبتنی بر روشنایی و بافت از تصاویر ناحیه بندی شده استخراج کرده ، به کمک این ویژگی ها و اعمال آن به کلاسه بند Adaboost آن را آموزش می دهیم. پس از آموزش، کلاس بند

^۱ Active contour

قادر است نتایج ناحیه بندی را به دو کلاس خوب و بد تفکیک کند. برای ارزیابی عملکرد کلاس بندی مقدار خطا را محاسبه می کنیم که این خطا نشان دهنده تعداد تصاویری است که توسط کلاس بندی به اشتباه کلاس بندی شده است.

در فصل چهارم به معرفی کلاس بندی شبکه عصبی می پردازیم. در این فصل روشی کارآمد جهت کلاس بندی تصاویر ناحیه بندی شده با استفاده از شبکه عصبی ارائه می دهیم. برای این منظور با اعمال ویژگی های استخراج شده از نتایج ناحیه بندی و به وسیله الگوریتم انتشار، شبکه عصبی را آموزش می دهیم. پس از آموزش، این کلاس بندی قادر است نتایج ناحیه بندی را به دو کلاس خوب و بد تفکیک کند. مشابه با کلاس بندی Adaboost میزان خطا را برای این کلاس بندی نیز محاسبه می نماییم. با مقایسه مقدار خطای به دست آمده توسط هر دو کلاس بندی بدیهی است که شبکه عصبی در مقایسه با Adaboost کلاس بندی موفق تری می باشد.

فصل دوم

مروری بر کارهای گذشته و

معرفی روش ناحیه بندی گراف کات

۱-۲ مقدمه

معمولا ناحیه بندی یک تصویر مساله مبهم و پیچیده ای است. با توجه به تنوع زیادی که در منابع، محتوا و پیچیدگی تصاویر وجود دارد اکثر اوقات ناحیه بندی کاملا اتوماتیک امکان پذیر نبوده و از این رو روش های تعاملی زیادی برای حل مسائل ناحیه بندی ارائه شده است [۶]. در کاربردهای زیادی، مانند ناحیه بندی بافت در تصاویر پزشکی، تعاملات کاربر کمک زیادی به رفع این ابهامات می کند. تا جایی که گاهی اوقات حتی با راهنمایی های کمی از طرف کاربر کیفیت ناحیه بندی به صورت فوق العاده بالا می رود و از اینروست که در دهه های اخیر تکنیک های ناحیه بندی تعاملی محبوبیت بیشتری پیدا کرده است. در روش ناحیه بندی تعاملی کاربر با انتخاب پیکسل هایی از تصویر به عنوان بذرها کمک زیادی به پیشبرد ناحیه بندی می کند. در این فصل چند روش مهم و کارآمد ناحیه بندی تصاویر مرور شده و مزایا و معایب هر یک را بیان می کنیم. سپس روش ناحیه بندی تعاملی مبتنی بر گراف کات را به طور مفصل معرفی می کنیم.

۲-۲ آستانه گذاری

تاکنون روش های مختلفی برای ناحیه بندی تصاویر ارائه شده است که از آن میان ناحیه بندی تصاویر بر مبنای آستانه گذاری به دلیل سادگی و حجم محاسباتی کم یکی از کاربردی ترین روش های ناحیه بندی تصاویر به شمار می آید. برای ناحیه بندی تصاویر به وسیله آستانه گذاری می توان از هیستوگرام روشنایی پیکسل های تصویر استفاده کرد. هیستوگرام روشنایی تصویر، منحنی فراوانی پیکسل ها بر حسب سطح روشنایی پیکسل های تصویر می باشد. مینیمم یا دره هیستوگرام روشنایی پیکسل ها، آستانه مناسبی برای ناحیه بندی تصویر به دو ناحیه شی و پس زمینه می باشد [۷]. پس از به دست آوردن مقدار آستانه، روشنایی هر یک از پیکسل های تصویر با مقدار آستانه مقایسه می

^۱ seed

شود. در صورتی که روشنایی آن پیکسل از مقدار آستانه بیشتر باشد پیکسل مذکور به عنوان شی و در غیر این صورت به عنوان پس زمینه در نظر گرفته می شود [۸]. اما عواملی نظیر نویز تصویر، وسعت دره ها در منحنی هیستوگرام روشنایی و تعدد مینیمم های محلی باعث دشوار بودن تعیین آستانه مناسب می باشند [۷].

۲-۳ رشد ناحیه^۱

روش رشد ناحیه که توسط Adams و Bischof [۹] ارائه شد روشی ساده و آسان برای ناحیه بندی تعاملی تصاویر محسوب می شود. اگرچه در این روش از هیچ تابع هزینه ای استفاده نمی شود اما محبوبیت این روش به خاطر سرعت و سادگی محاسباتی اش می باشد [۲].

در این روش، کاربر باید تعدادی بذر را در تصویر انتخاب کرده و با رشد بذر های تعیین شده آن ها را به n مجموعه مجزا $S = \{S_j : 0 \leq j \leq n\}$ تقسیم بندی کند که n تعداد نواحی مطلوب در ناحیه بندی می باشد. در مواردی که هدف، جداسازی شی از پس زمینه می باشد $n = 2$ در نظر گرفته می شود، در این صورت به دو مجموعه بذر نیاز داریم: مجموعه S_1 که شامل بذرهای معرف شی و مجموعه S_2 که شامل بذرهای معرف پس زمینه می باشد. رشد نواحی، یک الگوریتم تکرار می باشد. در هر مرحله از الگوریتم پیکسل های همسایگی نواحی در حال رشد تصویر بررسی می شود چنانچه خواص مشابهی نظیر رنگ، بافت و روشنایی با ناحیه در حال رشد دارند به آن متصل می شوند. رشد یک ناحیه وقتی متوقف می شود که هیچ پیکسل دیگری در همسایگی یک ناحیه، شرایط اتصال به آن را نداشته باشد. این الگوریتم به صورت تکراری ادامه یافته تا زمانی که تمام پیکسل های تصویر در یکی از دو مجموعه بذر قرار بگیرند [۲].

^۱Region growing

۴-۲ مدل مار^۱

اولین مدل کانتور فعال که در سال ۱۹۸۸ توسط Kass et al [۱۰] پیشنهاد شد مدل مار نام گرفت. در این مدل تابع انرژی روی کانتور تعریف می شود که از حاصل جمع توابع انرژی داخلی و خارجی کانتور به دست می آید. تابع انرژی داخلی میزان هموار بودن شکل کانتور را کنترل می کند در حالی که تابع انرژی خارجی تغییر شکل کانتور را بسته به تصویر اصلی تعیین می کند. در این مدل کانتوری که تابع انرژی را به حداقل برساند ناحیه مورد نظر را مشخص می کند. در شروع الگوریتم می بایست مجموعه نقاطی در حدود تقریبی مرزهای ناحیه مورد استخراج توسط کاربر مشخص شوند. سپس محاسبه تابع انرژی روی کانتور اولیه موقعیت بعدی این کانتور را با توجه به مینیمم کردن تابع انرژی تعیین می کند. با توجه به حرکات این کانتور که شبیه مار است این روش را ناحیه بندی به کمک مدل مار می نامند. لازم به ذکر است که نقاط اولیه کانتور تقریبی، ناحیه ای می باشد که در ابتدا در فاصله ای دورتر از مرز ناحیه قرار دارد. سپس هر نقطه به سمت لبه هایی حرکت می کند که تابع انرژی در این نقاط مینیمم می شود. سرانجام کانتور روی مرز ناحیه مورد نظر متوقف می شود. [۱۱].

مزیت این روش در مقایسه با آشکارسازهای لبه مثل اپراتور Canny [۱۲] این است که مرزهای پیوسته ای را ایجاد می کند. از معایب این روش این است که به آسانی برای کاربردهای چند بعدی قابل استفاده نمی باشد [۱۱].

^۱ Snake

۲-۵ برش های هوشمند^۱

Barrett و Mortensen در سال ۱۹۹۸ روش برش های هوشمند را برای ناحیه بندی تعاملی تصاویر پیشنهاد دادند [۱۳]. در این روش می توان با استفاده از حرکات ساده توسط ماوس کامپیوتر اشیا را در تصاویر دیجیتال با سرعت و دقت استخراج کرد. هنگامی که اشاره گر ماوس در مجاورت یک لبه از شی قرار بگیرد یک مرز مشخص می شود [۶].

در روش برش های هوشمند، تصویر به صورت گراف وزن داری در نظر گرفته می شود که پیکسل های تصویر گره های گراف را تشکیل می دهند و هر گره توسط لبه های جهتداری به هشت گره همسایه اش متصل می شود [۶]. سپس مرز ناحیه بندی به صورت مسیر بهینه ای روی این گراف محاسبه می شود [۱۴]. مزیت اصلی این روش که آن را از روش های قبلی مبتنی بر مرز متمایز می کند استفاده از ابزار “live wire” است که در سال ۱۹۹۲ ایجاد شده است. ابزار live wire برای شناسایی مرزها در تصاویر به کار می رود. با استفاده از این ابزار کاربر می تواند به صورت تعاملی مرز بهینه ای را انتخاب کند. به این صورت که ابتدا بذری توسط کاربر انتخاب می شود. سپس ابزار live wire مسیر بهینه ای را از مکان جاری مکان نما تا بذر تعریف شده تعیین می کند. در صورتی که مسیر تعریف شده روی مرز مطلوب قرار گرفته باشد، کاربر با کلیک کردن روی مکان جاری مکان نما این پیکسل را به عنوان بذر در نظر گرفته و به این ترتیب کاربر به صورت تعاملی می تواند مسیر بهینه را انتخاب کند. اما چنانچه مسیر تعیین شده توسط ابزار live wire روی مرز مطلوب قرار نگرفته باشد کاربر می تواند با تغییر مکان ماوس مسیر بهینه ای را مرز مطلوب به دست آورد. [۶].

تفاوت این روش نسبت به روش مارها این است که در روش مارها ابتدا یک کانتور مرزی تقریبی ناحیه مورد نظر را تعیین می کند، سپس این کانتور طوری تغییر می کند که تابع انرژی تعریف شده روی کانتور را به حداقل برساند. اما در روش برش های هوشمند ابتدا یک بذر توسط کاربر انتخاب شده

^۱ Intelligent scissors

سپس ابزار live wire تمام مسیرهای بهینه ممکن را از آن بذر به تمام نقاط دیگر در تصویر تولید می کند. حال کاربر می تواند به صورت تعاملی ناحیه مرزی مطلوب را انتخاب کند. روش برش های هوشمند در زمانی کوچکتر و با تلاشی کمتر می تواند یک تصویر را ناحیه بندی کند [۱۵].

لازم به ذکر است که در روش برش های هوشمند کاربر باید بذرهایی را دقیقاً روی مرز ناحیه بندی انتخاب کند سپس مرز ناحیه بندی به صورت کوتاهترین مسیر بین پیکسل های انتخاب شده محاسبه می شود. اشکال این روش با انتخاب چنین بذرهایی این است که ورودی های کاربر باید دقیقاً روی مرز ناحیه بندی قرار بگیرند پس این روش به آسانی نمی تواند برای کاربردهای بیشتر از دو بعد تعمیم داده شود [۱۳].

۲-۶ مروری بر الگوریتم گراف کات

در سال های اخیر ناحیه بندی تعاملی بر اساس یک گراف کات [۱۳][۱۶] محبوبیت زیادی یافته است. به عنوان اولین تحقیقات می توان به مقاله Boykov و Jolly [۱۶] اشاره نمود [۱۷].

هدف این فصل، مروری بر روش ناحیه بندی تعاملی بر اساس گراف کات است که نتیجه آن ناحیه بندی تصویر به دو بخش شی و پس زمینه می باشد. در این روش کاربر با برچسب زنی به پیکسل های خاصی که بذرها نامیده می شوند محدودیت های سختی را برای ناحیه بندی اعمال می کند. پیکسل های انتخاب شده باید متعلق به بخشی از شی و یا پس زمینه باشند. اعمال این محدودیت های سخت سرنخ هایی را برای ناحیه بند ایجاد می کند. مابقی تصویر با بهینه سازی یک تابع معیار به صورت اتوماتیک ناحیه بندی می شود. تابع معیار مورد نظر بر حسب یکپارچگی ناحیه ای و خصوصیات مرزی مطلوب در یک ناحیه بندی درست تعریف می شود که این خواص به عنوان محدودیت های نرم ناحیه بندی در نظر گرفته می شود [۱۶].

بزرگ ترین مزیت گراف کات، جواب بهینه و مطلوبی است که برای ناحیه بندی آرایه های N بعدی در نظر می گیرد [۱۶]. دومین مزیت این روش این است که از هر دو خواص ناحیه ای و مرزی برای ناحیه بندی استفاده می کند. علاوه بر این، تعامل کاربر خیلی ساده و راحت با انتخاب بذرها، مربوط به شی و پس زمینه انجام می گیرد. این بذرها باید داخل نواحی شی و پس زمینه قرار بگیرند و لذا انتخاب آنها مانند سایر روشها همچون live wire [۱۸] که می بایست در ناحیه مرزی انتخاب شوند دشوار نیست. در روش گراف کات با تعاملات مستقیم کاربر، یعنی اضافه یا کم کردن بذرها، (محدودیت های سخت) به سرعت ناحیه بندی مطلوب به دست می آید. کیفیت ناحیه بندی به طور مستقیم وابسته به تعریف تابع هزینه و نحوه مینیمم کردن آن دارد. لازم به ذکر است که تابع هزینه ای که به عنوان محدودیت نرم ناحیه بندی در نظر گرفته می شود نسبتا کلی و عمومی است و باید شامل خواص ناحیه ای و مرزی نواحی نیز باشد [۱۷].

۲-۶-۱ تئوری گراف کات

برای ناحیه بندی یک تصویر به روش گراف کات ابتدا گراف بدون جهت $g = \langle v, \varepsilon \rangle$ که v مجموعه ای از گره ها و ε مجموعه ای از لبه های غیر جهتدار می باشد را ایجاد می کنیم. در ناحیه بندی مبتنی بر گراف کات، گره های گراف نمایشگر پیکسل های تصویر ($p \in P$) و لبه ها نیز نمایانگر رابطه همسایگی بین پیکسل های مجاور می باشند. به مجموعه گره های ناشی از تصویر، دو گره به نام ترمینال شی^۱ (s) و ترمینال پس زمینه^۲ (t) نیز اضافه می شود. مجموعه P کل پیکسل های تصویر را نشان می دهد [۱۶].

بردار باینری $A = (A_1, \dots, A_p, \dots, A_{|P|})$ یک ناحیه بندی را تعریف می کند که مولفه A_p برچسب پیکسل p در مجموعه P می باشد که یکی از حالت های “obj” یا “bkg” (مخفف شی و پس

^۱ source
^۲ sink

زمینه) را می تواند بپذیرد. سپس محدودیات نرمی که روی خواص ناحیه ای و مرزی A اعمال می کنیم توسط تابع هزینه $E(A)$ تعریف می شود [۱۶]:

$$E(A) = \lambda \cdot R(A) + B(A) \quad (۱-۲)$$

$$R(A) = \sum_{p \in P} R_p(A_p) \quad (۲-۲)$$

$$B(A) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(A_p, A_q) \quad (۳-۲)$$

$$\delta(A_p, A_q) = \begin{cases} 1 & \text{if } A_p \neq A_q \\ 0 & \text{otherwise} \end{cases} \quad (۴-۲)$$

$R(A)$ هزینه ناحیه بندی A را با توجه به خواص همگنی دو ناحیه شی و پس زمینه اندازه می گیرد در حالی که $B(A)$ این هزینه را بر اساس خواص مرزی مورد انتظار محاسبه می نماید. در معادله (۱-۲) ضریب $\lambda \geq 0$ اهمیت نسبی هزینه ناحیه ای $R(A)$ به هزینه مرزی $B(A)$ را تنظیم می کند [۱۶]. ضریب $B_{\{p,q\}}$ در رابطه (۳-۲) با استفاده از تابع زیر محاسبه می گردد [۱۷]:

$$B_{\{p,q\}} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \quad (۵-۲)$$

در رابطه فوق I_p و I_q به ترتیب شدت روشنایی پیکسل های p و q می باشند. از آن جایی که تابع فوق متناظر با توزیع نويز در پیکسل های همسایگی تصویر می باشد، بنابراین پارامتر σ می تواند به عنوان نويز دوربين تخمین زده شود [۱۳].

با توجه به رابطه (۵-۲) بدیهی است که ضریب $B_{\{p,q\}}$ تنها مقادیر مثبت را اختیار می کند. در صورتی که شدت روشنایی پیکسل های p و q اختلاف کمی با یکدیگر داشته باشند مقدار این ضریب افزایش یافته و زمانی که اختلاف شدت روشنایی این پیکسل ها بزرگتر باشد مقدار این ضریب به صفر خواهد رسید. هزینه مرزی $B(A)$ ممکن است بر اساس گرادیان روشنایی محلی، لاپلاسیان، جهت گرادیان و یا معیارهای دیگر انتخاب شود [۱۶].

الگوریتم معرفی شده در مرجع [۱۹]، با مینیمم کردن تابع هزینه (۱-۲) بدون اعمال هیچ محدودیت سختی توسط کاربر، به ناحیه بندی تصویر می پردازد. اما هدف ما محاسبه مینیمم مطلق رابطه (۱-۲) با در نظر گرفتن محدودیت های سختی است که توسط کاربر اعمال می شود. با اعمال محدودیت های سخت که همان انتخاب بذره های شی و پس زمینه در تصویر می باشند ناحیه بندی موفق تری حاصل می شود [۱۶].

محدودیت های سخت انواع مختلفی دارند که در گذشته نیز برای روش های ناحیه بندی تعاملی مورد استفاده قرار می گرفتند. به عنوان مثال، در روش های ناحیه بندی تعاملی از جمله intelligent scissors [۶] و live wire [۱۸]، کاربر می بایست پیکسل های روی مرز شی و پس زمینه را به عنوان محدودیت سخت انتخاب کند. لذا این روش ها با انتخاب چنین محدودیت های سختی به آسانی برای کاربردهای سه بعدی قابل استفاده نمی باشند [۱۶]. اما در ناحیه بندی تعاملی بر اساس گراف کات، بذرهایی توسط کاربر داخل نواحی شی و پس زمینه به عنوان محدودیت های سخت در نظر گرفته می شوند و لزومی ندارد که بذرها از پیکسل های مرزی انتخاب شوند. به همین دلیل این روش به آسانی برای کاربردهای N بعدی نیز قابل استفاده می باشد [۱۷].

بدیهی است محدودیت های سخت به تنهایی برای داشتن ناحیه بندی خوب کافی نیستند. در حقیقت محدودیت های سخت را با تابع هزینه $E(A)$ ترکیب می کنیم تا خواص ناحیه ای و مرزی نواحی را شرکت دهیم. لازم به ذکر است که ناحیه بندی اولیه ممکن است مطلوب نباشد و بعد از بازنگری توسط کاربر و اعمال بذره های جدید، الگوریتم به روش مناسبی ناحیه بندی را تغییر می دهد بدون اینکه حل مساله را از ابتدا شروع کند [۱۶].

۲-۶-۲ گراف کات ها و بینایی ماشین

همان طور که در بخش قبل اشاره شد برای ناحیه بندی یک تصویر به روش گراف کات، ابتدا گراف دو ترمینالی $g = \langle v, \mathcal{E} \rangle$ متناظر با تصویر ایجاد می شود که در این گراف ترمینال ها s و t می باشند. سپس به هر لبه $(e \in \mathcal{E})$ وزن غیر منفی W_e اختصاص داده می شود. در گراف g مجموعه $\mathcal{E}$ شامل دو نوع لبه غیر جهتدار به نام های اتصال های ترمینال^۱ و اتصال های همسایگی^۲ می باشد. هر پیکسل p ($p \in P$) توسط دو اتصال ترمینال $\{p, t\}$ و $\{p, s\}$ به ترمینال های متناظر t و s متصل می شود. در حالی که پیکسل های همسایه $\{p, q\}$ در مجموعه N توسط یک اتصال همسایگی با هم ارتباط دارند. N یک سیستم همسایگی است که شامل تمام جفت پیکسل های همسایه (p, q) می باشد. در کاربردهای دو بعدی سیستم های همسایگی به صورت ۴ تایی و یا ۸ تایی در نظر گرفته می شوند [۱۶].

فرض کنید بذرهایی که توسط کاربر برای نواحی شی و پس زمینه انتخاب می شوند به ترتیب با مجموعه های O و B نمایش داده شوند. این مجموعه ها در حالت کلی ($O \subset P$ و $B \subset P$) نباید هیچ اشتراکی با یکدیگر داشته باشند ($O \cap B = \emptyset$). لازم به ذکر است که هدف، محاسبه مینیمم مطلق تابع هزینه $E(A)$ می باشد به شرط این که محدودیت های سخت زیر توسط کاربر اعمال شوند [۱۶].

$$\forall p \in O, A_p = "obj" \quad (۶-۲)$$

$$\forall p \in B, A_p = "bkg" \quad (۷-۲)$$

روابط بالا نشان می دهد در صورتی که پیکسل p به عنوان بذر شی انتخاب شود در بردار باینری A برچسب "obj" و در صورتی که بذر پس زمینه انتخاب گردد برچسب "bkg" را خواهد گرفت. جدول (۱-۲) وزن های لبه ها را در مجموعه $\mathcal{E}$ نشان می دهد [۱۶].

^۱ t-link
^۲ n-link

جدول (۱-۲): وزن های لبه ها [۱۶]

edge	Weight(cost)	For
$\{p, q\}$	$B_{\{p, q\}}$	$\{p, q\} \in N$
$\{p, s\}$	$\lambda \cdot R_p("bkg")$	$p \in P, p \notin OUB$
	K	$p \in O$
	0	$p \in B$
$\{p, t\}$	$\lambda \cdot R_p("obj")$	$p \in P, p \notin OUB$
	0	$p \in O$
	K	$p \in B$

$$K = 1 + \max_{p \in P} \sum_{q: \{p, q\} \in N} B_{\{p, q\}} \quad (۱-۲)$$

همان طور که در جدول فوق ملاحظه می کنید در صورتی که گره p جزء بذره های شی باشد اتصال بین p و ترمینال s برابر با ضریب K و اتصال بین p و ترمینال t برابر با صفر در نظر گرفته می شود. در صورتی که این گره جزء بذره های پس زمینه باشد اتصال آن با ترمینال s برابر با صفر و اتصال آن با ترمینال t برابر با ضریب K می باشد. اما اگر گره p به عنوان بذر انتخاب نشده باشد ضرایب اتصال آن با ترمینال های s و t مطابق با جدول فوق تعیین می گردد. گره های همسایه p و q نیز توسط اتصال همسایگی با وزن $B_{\{p, q\}}$ به یکدیگر متصل می شوند که مقدار این وزن طبق رابطه (۲-۲) تعیین می شود.

برای تعیین هزینه ناحیه ای $R(A)$ تعریف شده در رابطه (۲-۲) از پیکسل های انتخابی به عنوان بذرها استفاده کرده و هیستوگرام هایی برای شدت روشنایی شی و پس زمینه متناظر با $\Pr(I, B)$ و $\Pr(I, O)$ استخراج می گردد. سپس با استفاده از این هیستوگرام ها تابع هزینه $R_p(.)$ به صورت زیر برآورد می شود [۱۶]:

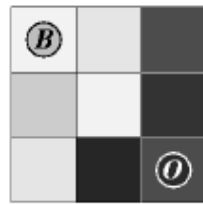
$$R_p("obj") = -\ln \Pr(I_p | O) \quad (۹-۲)$$

$$R_p("bkg") = -\ln \Pr(I_p | B) \quad (10-2)$$

زیر مجموعه ای از لبه ها در گراف g که ترمینال های t و s را از یکدیگر مجزا می کند یک کات نامیده می شود. ارزش کات C ($C \subset E$) بر اساس وزن های لبه های قطع شده توسط آن به صورت زیر تعریف می گردد:

$$|C| = \sum_{e \in C} W_e \quad (11-2)$$

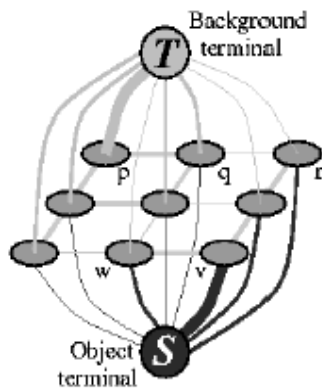
در شکل (۱-۲) مثالی برای ناحیه بندی تصویر 3×3 ملاحظه می کنید [۱۶]. تصویر اصلی به همراه



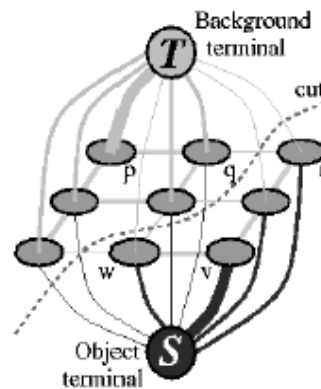
(a) تصویر اصلی به همراه بذرها



(d) نتایج ناحیه بندی



(b) یک گراف



(c) یک کات

شکل (۱-۲): مثالی ساده برای ناحیه بندی تصویر دو بعدی 3×3 . بذرها ی شی و پس زمینه به ترتیب عبارتند از $B=\{p\}$ و $O=\{v\}$. ضخامت هر لبه نشان دهنده ارزش آن لبه می باشد. هزینه مرزی تعریف شده در رابطه (۳-۲) ارزش های اتصالات همسایگی و هزینه ناحیه ای تعریف شده در رابطه (۲-۲) ارزش های اتصالات ترمینال را تعیین می کند [۱۶].

بذرهای انتخاب شده برای شی و پس زمینه که به ترتیب با حروف O و B مشخص شده اند در شکل (۱-۲a) نمایش داده شده است. گراف متناظر با این تصویر که شامل ۹ گره و دو ترمینال s و t می باشد در شکل (۱-۲ b) نشان داده شده است. هدف، یافتن کات با ارزش حداقل می باشد که این کات بهترین ناحیه بندی را در تصویر ایجاد می کند. شکل های (۱-۲ c) و (۱-۲ d) به ترتیب مینیمم کات و ناحیه بندی متناظر با آن را نشان می دهد.

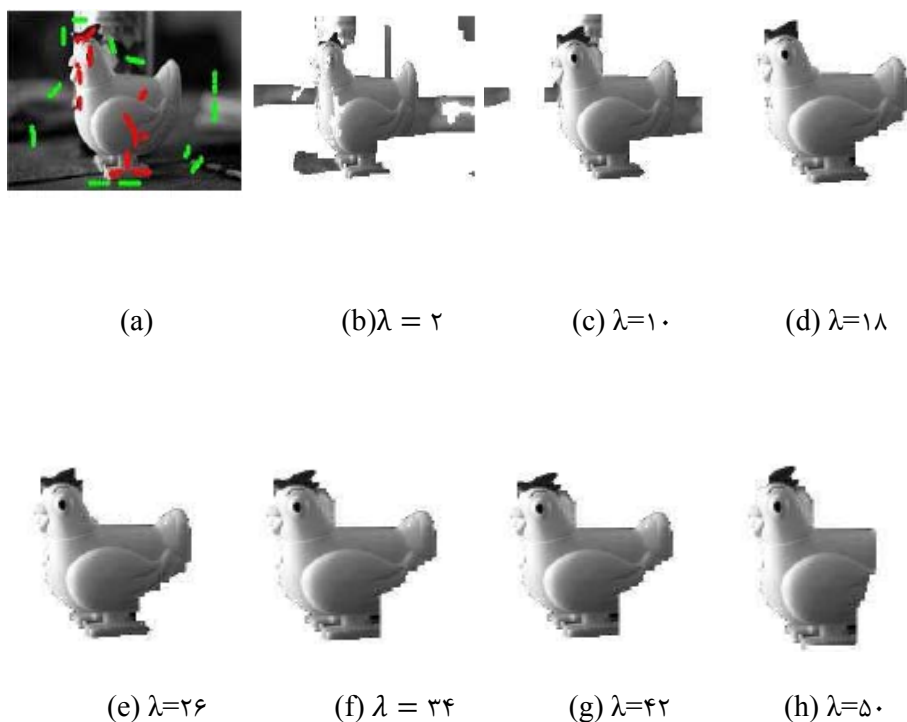
۲-۶-۳ انتخاب پارامترها

یکی از مهمترین و بزرگترین مسائل حل نشده در ناحیه بندی مبتنی بر گراف کات انتخاب پارامترهای مناسب برای الگوریتم می باشد. تاکنون روش مناسبی جهت یافتن پارامترهای یک ناحیه بندی اپتیمم برای یک تصویر وجود نداشته است. از این رو برای به دست آوردن نتیجه مناسب باید تصویر را به ازای پارامترهای مختلف ناحیه بندی کرده تا بهترین ناحیه بندی انتخاب گردد. گاهی برای به دست آوردن جواب مطلوب می بایست با تغییر بذرهای شی و پس زمینه نتایج ناحیه بندی را بهبود داد [۱۷].

از مهمترین پارامترهای موثر در الگوریتم گراف کات می توان به دو پارامتر λ و σ در روابط (۱-۲) و (۵-۲) اشاره کرد. در مرجع [۱۷] محققین اثر تغییرات پارامتر λ را در ناحیه بندی مورد بررسی قرار داده اند. در این تحقیق ما سعی داریم تا تغییرات دو پارامتر λ و σ را در ناحیه بندی بررسی کنیم. در این رابطه می خواهیم با بررسی ویژگی های نتیجه ناحیه بندی به ازای λ و σ های مختلف مقادیر بهینه λ و σ را در ناحیه بندی یک تصویر به صورت خودکار به دست آوریم.

نتایج ناحیه بندی یک تصویر به ازای مقادیر مختلف λ در شکل (۲-۲) نشان داده شده است [۱۷]. همان طور که در این شکل ها ملاحظه می کنید نتایج ناحیه بندی به شدت به مقدار λ وابسته

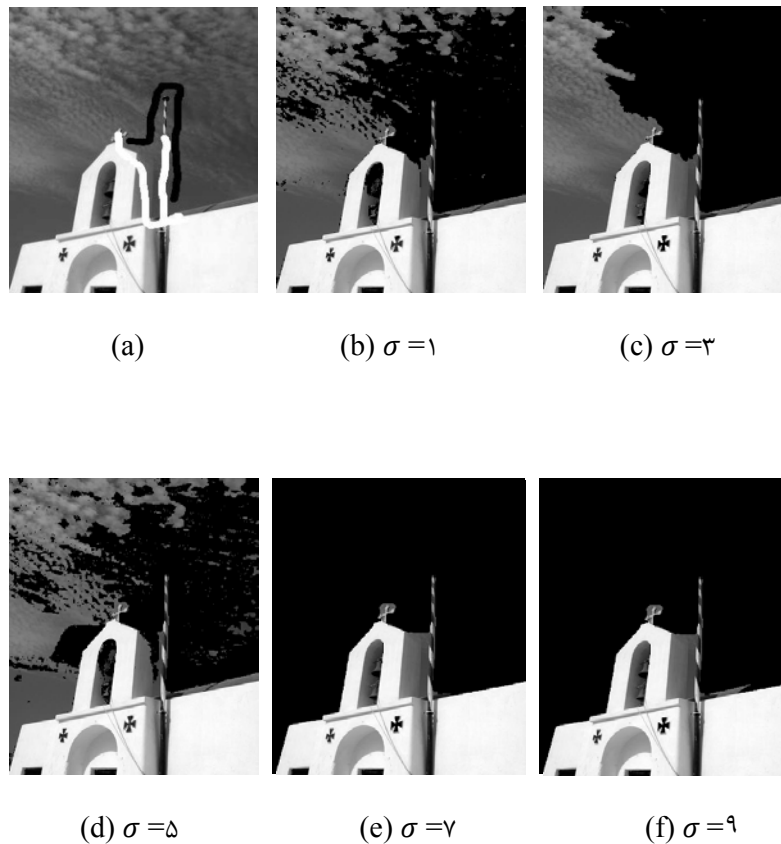
است و در واقع انتخاب نادرست پارامتر λ موجب رخ داد فوق^۱ یا زیر^۲ ناحیه بندی می شود. در پدیده فوق ناحیه بندی شی و پس زمینه مرزهای طولانی و نامنظمی دارند و ناحیه مربوط به شی شامل نواحی زیادی از پس زمینه نیز می باشد. شکل های (b,c)(۲-۲) پدیده فوق ناحیه بندی را نشان می دهند. در پدیده زیر ناحیه بندی نواحی شی و پس زمینه مرز کوتاهی با یکدیگر دارند و قسمتی از شی به اشتباه جزء پس زمینه ناحیه بندی می شود. شکل های (f-h)(۲-۲) پدیده زیر ناحیه بندی را نشان می دهند. در صورتی که λ در محدوده مناسبی انتخاب شود ناحیه بندی مناسب ایجاد می گردد (شکل های (d,e)).



شکل (۲-۲): (a) تصویر اصلی به همراه بذرها (بذرهای شی و پس زمینه به ترتیب با رنگ های قرمز و سبز مشخص شده است); (b-h) نشان دهنده ناحیه شی برای λ های مختلف; (b,c) نشان دهنده فوق ناحیه بندی; (d,e) نشان دهنده ناحیه بندی خوب; (f-h) نشان دهنده زیر ناحیه بندی [۱۷].

^۱ oversegmentation
^۲ undersegmentation

نتایج ناحیه بندی یک تصویر به ازای مقادیر مختلف σ نیز در شکل (۳-۲) نشان داده شده است. همان طور که در این شکل ها ملاحظه می کنید نتایج ناحیه بندی به شدت به مقدار σ وابسته است. در شکل های (b-d) (۳-۲) پدیده فوق ناحیه بندی رخ داده است. در حالی که در شکل های (e,f) (۳-۲) یک ناحیه بندی خوب مشاهده می شود.



شکل (۳-۲) : (a) تصویر اصلی به همراه بذرها (بذرهای شی و پس زمینه به ترتیب با رنگ های سفید و مشکی نشان داده شده است)؛ (b-f) ناحیه مربوط به شی به ازای σ های مختلف و ناحیه پس زمینه با رنگ مشکی مشخص شده است؛ (b-d) نشان دهنده فوق ناحیه بندی ؛ (e,f) نشان دهنده ناحیه بندی خوب.

۲-۶-۴ مینیمم کات روی گراف

در بخش های قبل نحوه به دست آوردن گراف g از یک تصویر توضیح داده شد. همان طور که اشاره شد ناحیه بندی مناسب با یافتن مینیمم کات روی گراف تصویر حاصل می گردد [۱۶].

فرض کنید $\mathcal{F}$ مجموعه ای از تمام کات های ممکن روی گراف g را تشکیل دهد به طوری که هر یک از این کات ها مثل کات C دارای شرایط زیر باشد:

- کات C دقیقا یکی از اتصالات ترمینال را در هر گره p قطع می کند.
- اگر p و q به ترمینال های مختلف متصل شده باشند اتصال همسایگی آن ها باید توسط کات C قطع شود یعنی $\{p, q\} \in C$ می باشد.
- اگر p به ناحیه شی متعلق باشد اتصال p با ترمینال t جزء کات C می باشد.

$$\text{if } p \in O \Rightarrow \{p, t\} \in C$$

- اگر p به ناحیه پس زمینه متعلق باشد اتصال p با ترمینال s جزء کات C می باشد.

$$\text{if } p \in B \Rightarrow \{p, s\} \in C$$

در ادامه به اثبات این که مینیمم کات $\hat{C}$ ناحیه بندی بهینه $\hat{A}$ را برای تصویر ایجاد می کند می پردازیم [۱۶].

اصل ۱: مینیمم کات $\hat{C}$ یکی از کات های ممکن در گراف است ($\hat{C} \in \mathcal{F}$).

اثبات: از آن جایی که هر کات باید ترمینال های s و t را از یکدیگر جدا نماید بنابراین کات $\hat{C}$ باید فقط یکی از اتصالات ترمینال را در هر گره p قطع کند. همچنین در صورتی که گره های همسایه p و q به ترمینال های مختلفی متصل شده باشند مینیمم کات می بایست اتصال همسایگی $\{p, q\}$ را نیز قطع نماید. این بدین خاطر است که هر کات باید ترمینال ها را از یکدیگر مجزا کند. به عکس اگر گره های همسایه p و q به ترمینال مشابهی (s یا t) متصل باشند در این صورت مینیمم کات $\hat{C}$ نباید

اتصال همسایگی $\{p,q\}$ را قطع کند. دو خاصیت اخیر در مورد $\hat{C}$ صحت دارد به خاطر این که در هر گره مثل p ضرب K به دست آمده از رابطه (۲-۸) از مجموع تمام وزن های اتصال های همسایه اش بزرگتر می باشد [۱۶].

حال برای هر کات ممکن $C \in \mathcal{F}$ می توان ناحیه بندی متناظر یکتا $A(C)$ را بنحوی تعریف کرد که :

$$A_p(C) = \begin{cases} "obj" & \text{if } \{p, t\} \in C \\ "bkg" & \text{if } \{p, s\} \in C \end{cases} \quad (۲-۱۲)$$

طبق رابطه (۲-۱۲) زمانی که اتصال p با ترمینال t جزء کات C باشد در بردار باینری A برچسب obj و در صورتی که اتصال p با ترمینال s جزء این کات باشد در بردار باینری A برچسب bkg خواهد گرفت.

صحت رابطه فوق واضح است به خاطر این که هر کات دقیقا یکی از دو اتصال ترمینال را در هر گره قطع می نماید. طبق اصل ۱ مینیمم کات $\hat{C}$ روی گراف ممکن و شدنی است. از اینرو می توان ناحیه بندی متناظر $\hat{A} = A(\hat{C})$ را تعریف کرد [۱۶].

قضیه ۱ : مینیمم کات $\hat{C}$ ناحیه بندی را تعریف می کند که تابع هزینه $E(A)$ در رابطه (۲-۱) حداقل می شود.

اثبات: با استفاده از جدول (۲-۱) که وزن های لبه ها را در گراف g مشخص می کند و تعریف کات های ممکن $\mathcal{F}$ و معادله (۲-۱۲) می توان مقدار هر کات را به صورت زیر در نظر گرفت:

$$\begin{aligned} |C| &= \sum_{p \notin O \cup B} \lambda \cdot R_p(A_p(C)) + \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(A_p(C), A_q(C)) \\ &= E(A(C)) - \sum_{p \in O} \lambda \cdot R_p("obj") - \sum_{p \in B} \lambda \cdot R_p("bkg") \end{aligned} \quad (۲-۱۳)$$

بنابراین $|C| = E(A(C)) - const(C)$

در حقیقت رابطه (۲-۱۲) تناظر یک به یکی بین مجموعه ای از کات های ممکن در $\mathcal{F}$ و ناحیه بندی های A در مجموعه $\mathcal{H}$ را نشان می دهد. پس

$$\begin{aligned} E(\hat{A}) &= |\hat{C}| + \text{const} = \min_{C \in \mathcal{F}} |C| + \text{const} \\ &= \min_{C \in \mathcal{F}} E(A(C)) = \min_{A \in \mathcal{H}} E(A) \end{aligned} \quad (۲-۱۴)$$

و قضیه اثبات می شود [۱۶].

۲-۶-۵ تئوری Max-Flow/Min-Cut

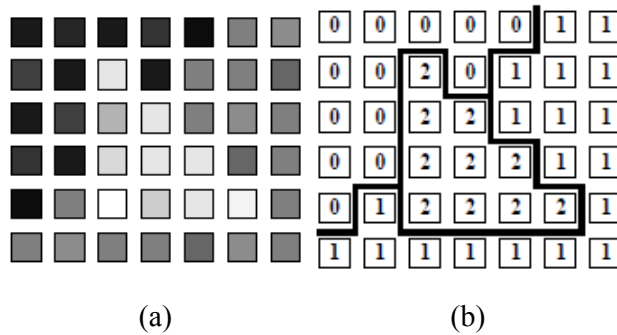
Greig et al [۲۰] نشان داد الگوریتم های ماکزیمم فلو و مینیمم کات از الگوریتم های بهینه سازی ترکیبی برای حداقل کردن توابع انرژی خاصی قابل استفاده هستند. توابع انرژی های مطرح شده به صورت زیر در نظر گرفته می شود [۲۱]:

$$E(L) = \sum_{p \in P} D_p(L_p) + \sum_{\{p,q\} \in N} V_{p,q}(L_p, L_q) \quad (۲-۱۵)$$

این رابطه مشابه با تابع هزینه $E(A)$ تعریف شده در بخش قبل می باشد. در رابطه فوق $L = \{L_p | p \in P\}$ برچسب^۱ تصویر P و $D_p(\cdot)$ و $V_{p,q}$ هزینه های ناحیه ای و مرزی را تعیین می کنند. پس می توان ناحیه بندی A را به صورت برچسب گذاری پیکسل های تصویر در نظر گرفت. نمونه ای از برچسب گذاری یک تصویر در شکل (۲-۴) نشان داده شده است. Greig et al [۲۰] نشان داد مینیمم کات روی گراف دو ترمینالی، برچسب بهینه L را ایجاد می کند [۲۱].

از روش های پیشین برای مینیمم کردن توابع انرژی مورد مطالعه می توان به الگوریتم simulated annealing اشاره نمود. در مرجع [۲۰] نشان داده شده است که حتی در مسائل ساده بازنشانی تصویر نتایج simulated annealing [۲۲] خیلی بدتر از جواب بهینه است [۲۱].

^۱ label



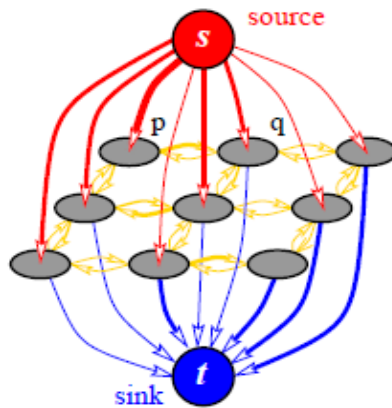
شکل (۲-۴): (a) تصویر اصلی، (b) یک نمونه برجسب گذاری [۲۱].

۲-۶-۶ مسائل ماکزیمم فلو یا مینیمم کات

در این بخش روش بدست آوردن مینیمم کات را روی یک گراف توضیح می دهیم [۲۳]. همان طور که قبلا نیز اشاره شد مینیمم کات مجموعه ای از لبه های گراف است که تابع انرژی تعریف شده در رابطه (۲-۱۵) برای آن حداقل می باشد. برای محاسبه مینیمم کات روی گراف ها الگوریتم های فراوانی ارائه شده اند که این الگوریتم ها از گراف های جهتدار استفاده می کنند [۲۱]. بنابراین در این بخش ابتدا در مورد گراف های جهتدار توضیح مختصری می دهیم و سپس به بیان این الگوریتم ها می پردازیم [۱۳].

۲-۶-۶-۱ گراف جهتدار

نمونه ای از یک گراف جهتدار در شکل (۲-۵) نشان داده شده است. همان طور که در این شکل ملاحظه می کنید ساختار این گراف مشابه با گراف بدون جهت G می باشد با این تفاوت که در گراف جهتدار لبه ها به صورت جهتدار در نظر گرفته می شوند. در یک گراف جهتدار گره های همسایه p و q توسط دو لبه (p,q) و (q,p) با وزن های متفاوت $W_{(p,q)}$ و $W_{(q,p)}$ به یکدیگر متصل می شوند. وزن های لبه ها در یک گراف جهتدار به صورت زیر در نظر گرفته می شود [۱۳]:



شکل (۵-۲) : نمونه ای از یک گراف جهتدار [۲۱].

$$W_{(p,q)} = \begin{cases} 1 & \text{if } I_p \leq I_q \\ \exp\left(\frac{-(I_p - I_q)^2}{2\sigma^2}\right) & \text{if } I_p > I_q \end{cases} \quad (۱۶-۲)$$

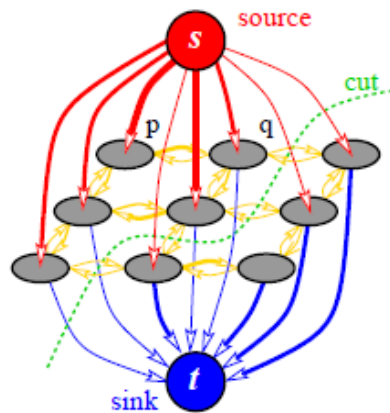
چنانچه بخواهیم $W_{(q,p)}$ را محاسبه کنیم در رابطه فوق جای p و q را با یکدیگر عوض می کنیم. با در نظر گرفتن لبه ها به صورت جهتدار هزینه مرزی بیان شده در رابطه (۳-۲) به صورت زیر تغییر می کند [۱۳]:

$$B(A) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(A_p = \text{"obj"}, A_q = \text{"bkg"}) \quad (۱۷-۲)$$

در این رابطه N یک سیستم همسایگی است که شامل تمام پیکسل های همسایه (p,q) می باشد.

در رابطه فوق چنانچه p به عنوان شی ("obj") و q به عنوان پس زمینه ("bkg") در نظر گرفته شود از تابع $B_{\{p,q\}}$ استفاده می شود و در غیر این صورت یعنی مواقعی که p به عنوان پس زمینه و q به عنوان شی محسوب شود تابع $B_{\{q,p\}}$ مورد استفاده قرار می گیرد [۱۳]. حال به منظور مینیمم کردن تابع هزینه تعریف شده در رابطه (۱-۲) می توان تابع $B_{\{p,q\}}$ را برابر با اتصال همسایگی $W_{(p,q)}$ در نظر گرفت $(W_{(p,q)} = B_{\{p,q\}})$ [۱۳].

در یک گراف جهتدار دو ترمینالی گره ها توسط کات C به زیرمجموعه های مجزای S و T تقسیم می شوند. گره های متصل به ترمینال s به همراه این ترمینال در مجموعه S و گره های متصل به ترمینال t به همراه این ترمینال در مجموعه T قرار می گیرند. بدین ترتیب گره های متعلق به زیر مجموعه S به عنوان شی و سایر گره ها که متعلق به زیرمجموعه T می باشند به عنوان پس زمینه برچسب می خورند. اگر گره p با ترمینال s و گره q با ترمینال t اتصال داشته باشد کات C باید اتصال همسایگی آنها را قطع کند. ارزش کات C در یک گراف جهتدار نیز طبق رابطه (۲-۱۱) بر اساس وزن های لبه های قطع شده توسط کات تعریف می گردد. اما این سوال پیش می آید که برای محاسبه ارزش کات کدام یک از وزن های لبه های جهتدار را باید در نظر گرفت؟ با توجه به اینکه اتصالی از ترمینال s به تمامی گره ها و از تمامی گره ها به ترمینال t وجود دارد، فقط $W_{(p,q)}$ برای محاسبه ارزش کات در نظر گرفته شده و $W_{(q,p)}$ نادیده گرفته می شود. اما در صورتی که گره p به ترمینال t و گره q به ترمینال s اتصال داشته باشد فقط $W_{(q,p)}$ برای محاسبه ارزش کات در نظر گرفته می شود. در شکل (۲-۶) یک کات روی گراف جهتدار نشان داده شده است [۲۱].



شکل (۲-۶) : نمونه ای از یک کات روی گراف جهتدار [۲۱].

یکی از روش های یافتن مینیمم کات روی یک گراف، الگوریتم مبتنی بر ماکزیمم فلو می باشد. Ford و Fulkerson [۲۴] نشان دادند که ماکزیمم فلو از ترمینال s به t ، مجموعه ای از لبه ها را روی گراف دو ترمینالی اشباع می کند. در حقیقت مقدار فلو ماکزیمم با ارزش مینیمم کات برابر است. پس ماکزیمم فلو و مینیمم کات معادل یکدیگرند [۲۱]. در ادامه ابتدا فلو را تعریف کرده و سپس نحوه ماکزیمم کردن آن را بیان می کنیم.

۲-۶-۶-۲ ماکزیمم فلو

فلو در اصطلاح به معنای شارش و جریان می باشد. در مبحث گراف ها مقدار فلو گذر کننده از ترمینال s به t را با تابع f نمایش می دهیم که این تابع باید شرایط زیر را داشته باشد [۲۵]:

- مقدار فلو گذر کننده از هر لبه کوچکتر یا مساوی وزن متناظر با آن لبه می باشد:

$$f(p, q) \leq w(p, q)$$
- اصل تقارن: برای تمامی گره های موجود در گراف قرارداد می کنیم که $f(p, q) = -f(q, p)$.
- اصل پایستگی: برای تمامی گره های موجود در گراف به جز ترمینال های s و t ، مجموع فلوهای وارد شده به هر گره برابر با مجموع فلوهای خارج شده از آن می باشد.

بنابراین مقدار فلو گذر کننده از گراف را به صورت زیر تعریف می کنیم:

$$|f| = \sum_{q \in V} f(s, q) \quad (۱۸-۲)$$

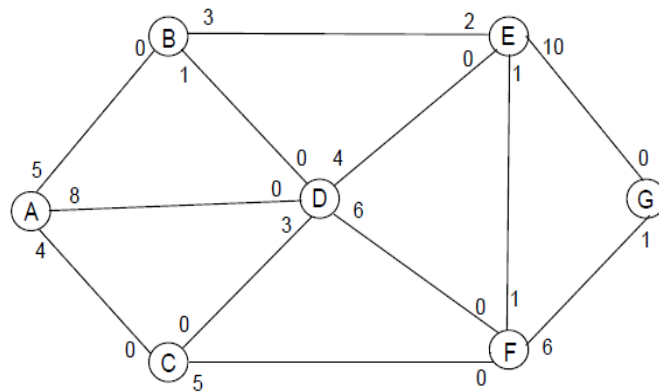
در رابطه فوق V تعداد کل گره های موجود در گراف می باشد. بر اساس این رابطه مقدار فلو گذر کننده از گراف برابر با مجموع فلوهای خارج شده از ترمینال s می باشد [۲۵]. برای روشن تر شدن این مبحث مثال زیر را مطرح می کنیم [۲۳].

خودرویی را در نظر بگیرید که از مبدا A شروع به حرکت کرده و می تواند از مسیرهای مختلفی به مقصد G برسد. همان طور که در شکل (۷-۲) ملاحظه می کنید در نزدیکی هر گره بر روی هر مسیر اعدادی وجود دارند که این اعداد نشان دهنده ظرفیت آن مسیر می باشد. منظور از ظرفیت مسیر تعداد خودروهایی است که می توانند از مسیر مورد نظر عبور کنند. به عنوان نمونه مسیر A تا B را در نظر بگیرید. در این مسیر ظرفیت ۵ در نزدیکی گره A نشان دهنده تعداد خودروهایی است که می توانند در جهت رفت از A به B بروند. اما روی این مسیر در نزدیکی گره B ظرفیت صفر نوشته شده است به این معنا که هیچ خودرویی نمی تواند در جهت برگشت یعنی از B به A حرکت کند [۲۳].

برای حرکت این خودرو از مبدا به مقصد مسیرهای متفاوتی وجود دارد که در ادامه به تعدادی از این مسیرها اشاره می کنیم:

۱. در مسیر A-D-E-G فقط ۴ خودرو اجازه حرکت از مبدا A به مقصد G دارند. به این صورت که ابتدا ۸ خودرو از مسیر A تا D اجازه ورود داشته اما فقط ۴ خودرو می توانند از مسیر D به E عبور کنند. با وجود این که ظرفیت مسیر E به G ۱۰ خودرو می باشد اما همین ۴ خودرو می توانند از این مسیر عبور کنند. بنابراین مقدار ماکزیمم فلو از مبدا به مقصد در طول این مسیر ۴ خودرو بوده که این مقدار فلو کوچکترین ظرفیت در جهت رفت در طول این مسیر می باشد [۲۳].

۲. در مسیر A-B-E-G نیز فقط ۳ خودرو می توانند با حرکت از مبدا A به مقصد G برسند. به این دلیل که کوچکترین ظرفیت در جهت رفت در طول این مسیر مربوط به B-E می باشد. با در نظر گرفتن مقادیر فلو روی مسیرهای A-D-E-G و A-B-E-G مجموع فلوئی که تاکنون روی مسیر E-G به دست آمده برابر با ۷ خودرو می باشد [۲۳].



شکل (۷-۲) : یافتن ماکزیمم فلو از مبدا A به مقصد G [۲۳]

۳. ۴ خودروی دیگر نیز می توانند در طول مسیر A-C-F-G حرکت کرده و به مقصد G برسند. کوچکترین ظرفیت در این مسیر مربوط به A-C می باشد [۲۳].

بنابراین مجموع فلو در طول این سه مسیر $4+3+4=11$ خودرو می باشد که می توانند با حرکت از مبدا A به مقصد G برسند. اما این تعداد حداکثر تعداد خودروهایی نیستند که می توانند از مبدا به مقصد حرکت کنند. به خاطر این که مسیرهای دیگری نیز وجود دارند (به عنوان مثال مسیر A-D-F-E-G) که می توان از آن ها برای رسیدن به مقصد استفاده کرد [۲۳]. برای تعیین فلو گذر کننده از مبدا به مقصد روی مسیرهای مختلف شکل (۷-۲) از الگوریتم ارائه شده توسط Ford و Fulkerson [۲۳] استفاده می کنیم. مراحل اجرای این الگوریتم به صورت زیر می باشد [۲۳]:

گام ۱ : یک مسیر از مبدا به مقصد انتخاب می کنیم که ظرفیت باقیمانده لبه های موجود در این مسیر مقادیر مثبتی داشته باشند. در صورت یافتن چنین مسیری گام های زیر را طی کرده و در غیر این صورت الگوریتم خاتمه می یابد. لازم به ذکر است که در شروع الگوریتم ظرفیت باقیمانده لبه ها برابر با ظرفیت اولیه آن ها می باشند.

گام ۲ : کوچکترین ظرفیت موجود در مسیر یافت شده به عنوان ماکزیمم فلو f گذر کننده از آن مسیر در نظر گرفته می شود.

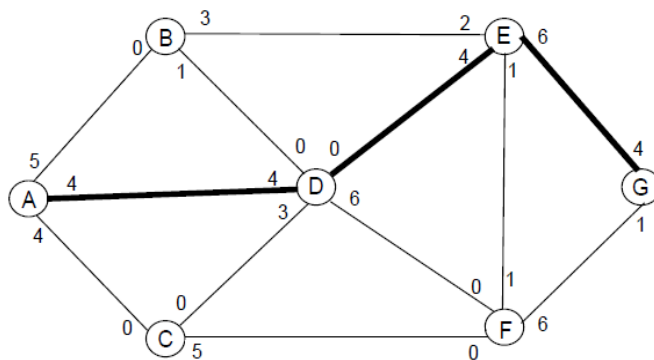
گام ۳ : در طول این مسیر ظرفیت باقیمانده تمام لبه های مسیر رفت را از f کم کرده در حالی که ظرفیت باقیمانده لبه های مسیر برگشت را با مقدار f جمع می کنیم.

گام ۴ : الگوریتم مجدداً از ابتدا تکرار می شود.

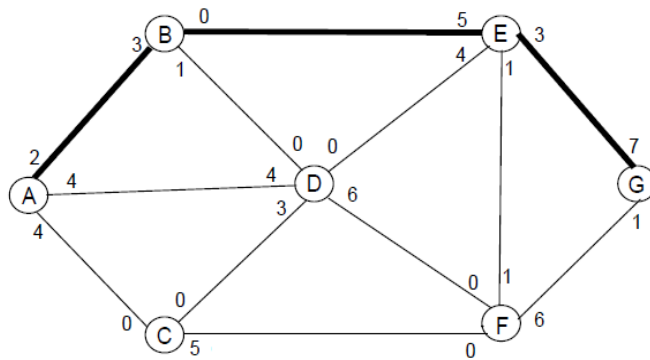
همان طور که در الگوریتم فوق ملاحظه می کنید در هر تکرار ظرفیت لبه ها تغییر می کند که به ظرفیت های جدید به دست آمده برای هر لبه، ظرفیت باقیمانده متناظر با آن لبه گفته می شود.

پس از یافتن تمام مسیرهای ممکن از مبدا به مقصد الگوریتم خاتمه می یابد. با خاتمه الگوریتم می توان ماکزیمم فلو را با مجموع فلوهای به دست آمده برای هر مسیر در گام ۱ محاسبه کرد.

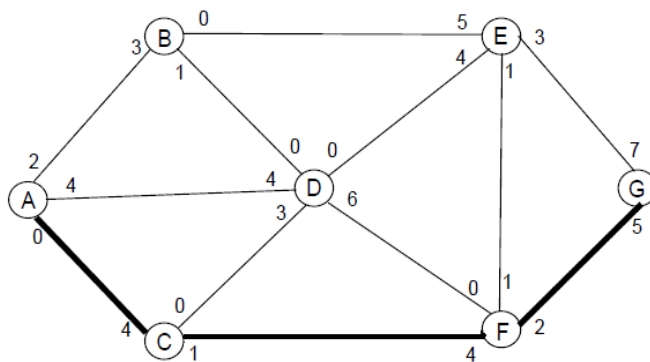
چنانچه الگوریتم فوق را بر روی شکل (۷-۲) پیاده کنیم نتایج به صورت زیر خواهد بود [۲۳].



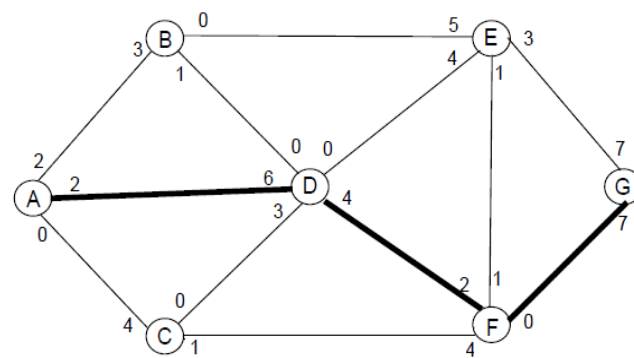
شکل (۷-۲) : در طول مسیر A-D-E-G فقط ۴ خودرو می توانند حرکت کنند [۲۳].



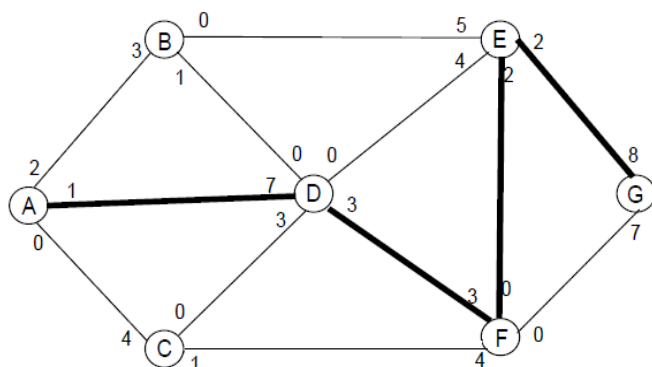
شکل (۹-۲) : در طول مسیر A-B-E-G فقط ۳ خودرو می توانند حرکت کنند [۲۳].



شکل (۱۰-۲) : در طول مسیر A-C-F-G فقط ۴ خودرو می توانند حرکت کنند [۲۳].



شکل (۱۱-۲) : در طول مسیر A-D-F-G فقط ۲ خودرو می توانند حرکت کنند [۲۳].



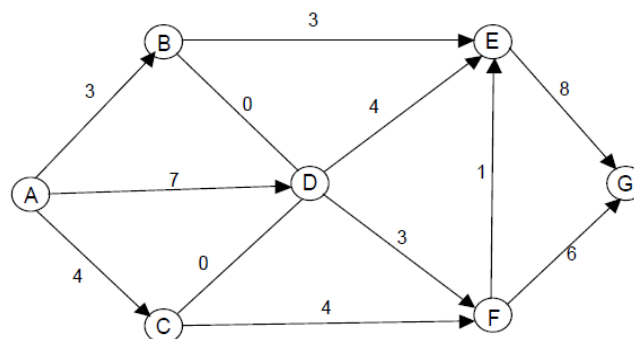
شکل (۲-۱۲) : در طول مسیر A-D-F-E-G فقط ۱ خودرو می تواند حرکت کند [۲۳].

با توجه به شکل (۲-۱۲) با خاتمه الگوریتم دیگر نمی توان مسیری با ظرفیت های باقیمانده مثبت از مبدا به مقصد پیدا کرد. به این دلیل که تمام مسیرهای موجود از مبدا به مقصد از مجموعه لبه های D-E، B-E، F-E، و F-G می گذرند که مقدار ظرفیت تمامی این لبه ها در مسیر رفت صفر شده است. به مجموعه لبه های D-E، B-E، F-E، و F-G که مقدار ظرفیت آن ها در مسیر رفت برابر با صفر شده لبه های اشباع گفته می شود [۲۳].

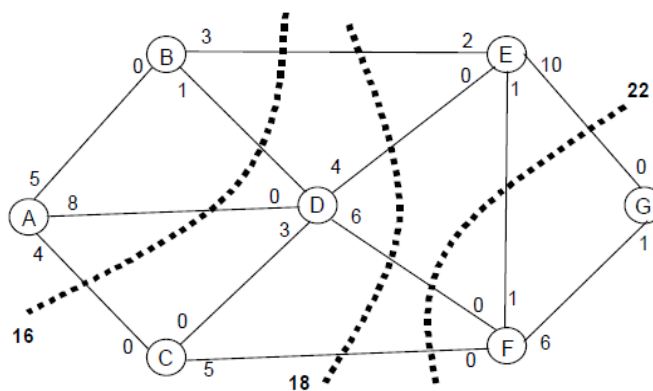
برای محاسبه ماکزیمم فلو از مبدا A به مقصد G می توان فلوهای به دست آمده در ۵ مسیر را با یکدیگر جمع کرد که مقدار ماکزیمم فلو برابر با $1+2+4+3+4=14$ خودرو می باشد. پس ۱۴ خودرو می توانند در طی مسیرهای مختلف به صورت همزمان از مبدا به مقصد بروند [۲۳].

حال می خواهیم مقدار فلو و جهت متناظر با آن را برای تمامی لبه ها تعیین کنیم. برای این منظور اختلاف ظرفیت هر لبه را در مسیر رفت قبل از شروع الگوریتم و پس از اتمام آن محاسبه می کنیم، چنانچه این اختلاف مقدار مثبتی باشد فلویی در جهت رفت خواهیم داشت و در غیر این صورت فلویی در جهت برگشت روی این لبه وجود خواهد داشت [۲۳].

شکل (۱۳-۲) مقادیر فلو را در هر یک از لبه ها نشان می دهد. بدیهی است که اصل پایستگی فلو در تمامی گره ها به جز گره ابتدایی و انتهایی برقرار است ، یعنی مجموع فلوهای وارد شده به هر گره برابر با مجموع فلوهای خارج شده از آن گره می باشد. پس از محاسبه ماکزیمم فلو گذر کننده از مبدا به مقصد روی یک گراف می خواهیم مینیمم کات را روی آن به دست آوریم. لازم به ذکر است که یک کات به صورت مجموعه ای از لبه های جهتدار تعریف می شود که حداقل یکی از لبه ها را در هر مسیر از مبدا به مقصد قطع می کند. کات مینیمم نیز کاتی با مینیمم ارزش بین تمام کات های موجود در گراف می باشد [۲۳]. شکل (۱۴-۲) چند نمونه از کات های ممکن را روی گراف نشان می دهد که ارزش هر کات با برچسبی نشان داده شده است.

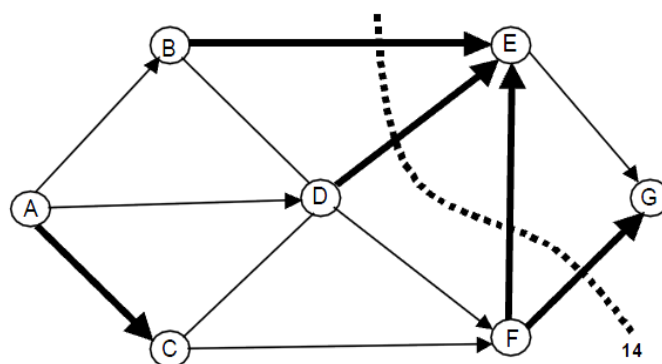


شکل (۱۳-۲) : فلوهای به دست آمده در هر مسیر [۲۳].



شکل (۱۴-۲): چند نمونه از کات های ممکن [۲۳].

نظریه Ford و Fulkerson [۲۴] چنین بیان می کند که مقدار ماکزیمم فلو گذر کننده از مبدا به مقصد روی یک گراف برابر با ارزش مینیمم کات می باشد. پس در این گراف ارزش مینیمم کات برابر با ۱۴ می باشد اما این کات مینیمم از کدام لبه ها می گذرد. در مرجع [۲۳] نشان داده شده که مینیمم کات از لبه های اشباع شده ای که ماکزیمم فلو با استفاده از آن ها به دست می آید، می گذرد. البته ممکن است که تمام لبه های اشباع شده در مینیمم کات وجود نداشته باشند. در شکل (۱۵-۲) مینیمم کات روی این گراف نشان داده شده است..



شکل (۱۵-۲) : مینیمم کات روی گراف [۲۳].

۷-۶-۲ الگوریتم های استاندارد در بهینه سازی ترکیبی

برای یافتن مینیمم کات (ماکزیمم فلو) روی گراف های جهتدار دو ترمینالی، الگوریتم های گوناگونی وجود دارد. این الگوریتم ها در یکی از دو گروه زیر دسته بندی می شوند [۲۱]:

- الگوریتم "مسیر تقویت کننده" ^۱ Ford-Fulkerson [۲۴].
- الگوریتم "push-relabel" به سبک Goldberg-Tarjan [۲۶].

در الگوریتم های مبتنی بر مسیر تقویت کننده، نظیر الگوریتم Dinic [۲۷] از روش جستجو breadth-first برای یافتن کوتاهترین مسیر از ترمینال s به t روی گراف باقیمانده g_f استفاده می شود. بعد از اشباع مسیرهایی با طول k الگوریتم در جستجوی مسیرهایی با طول $k + 1$ می باشد. طی این الگوریتم مقدار فلو f توسط گراف باقیمانده g_f ذخیره می شود. پس از اینکه تمام مسیرهایی موجود از ترمینال s به t حداقل از یک لبه اشباع عبور کنند الگوریتم خاتمه می یابد. لازم به ذکر است که استفاده از کوتاهترین مسیرهها پیچیدگی زمانی را در اجرای الگوریتم های مسیر تقویت کننده بهبود می بخشد. پیچیدگی زمانی الگوریتم های Dinic در بدترین حالت $O(mn^2)$ می باشد که n تعداد گره ها و m تعداد لبه ها در گراف می باشند [۲۱].

الگوریتم های push-relabel [۲۶] از روشی کاملاً متفاوت استفاده می کنند. در این الگوریتم ها بر خلاف الگوریتم های مسیر تقویت کننده، مقدار فلو ذخیره نمی شود اما گره های فعالی به کار می روند که اضافه فلو^۲ مثبتی دارند. در این روش برجسبی از گره ها که کران پایین فاصله تا ترمینال t را در طول لبه های غیر اشباع برآورد می کنند ذخیره می شود. سپس الگوریتم، فلوهای اضافه را به سمت گره هایی با فاصله تخمینی کوچکتر تا ترمینال t می فرستد. در نتیجه فاصله ها

^۱ augmenting path
^۲ Flow excess

(برچسب ها) افزایش پیدا می کنند به طوری که لبه ها اشباع می شوند و در نهایت فلوهای غیرقابل تحویل به ترمینال s برگردانده می شود.

۲-۶-۸ الگوریتم جدید ماکزیمم فلو یا مینیمم کات

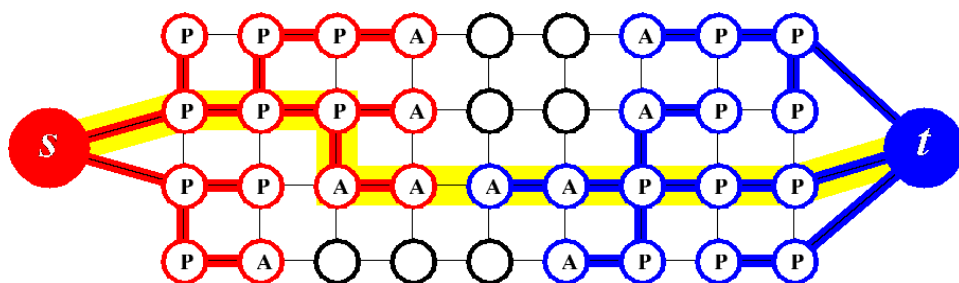
در این بخش به معرفی یکی از روش های نسبتا جدید یافتن مینیمم کات که براساس الگوریتم مسیر تقویت کننده کار می کند می پردازیم [۲۱]. مشابه با الگوریتم Dinic [۲۷] نویسندگان [۲۴] برای شناسایی مسیرهای تقویت کننده، دو درخت جستجو یکی از ترمینال s و دیگری از ترمینال t تشکیل می دهند. اما نقص این روش آن است که مسیرهای تقویت کننده یافت شده لزوما کوتاهترین مسیرها نمی باشند. از این رو پیچیدگی اجرایی الگوریتم در بدترین حالت $O(mn^2|c|)$ خواهد شد که $|C|$ ارزش مینیمم کات می باشد. لازم به ذکر است که پیچیدگی این روش از الگوریتم های استاندارد که در بخش قبل ملاحظه کردید بدتر می باشد. اما مقایسه تجربی نشان می دهد که در بعضی موارد این روش از روش های استاندارد بهتر عمل می کند [۲۱].

دو درخت جستجوی δ و T با منشاهایی در ترمینالهای s و t که هیچ اشتراکی با یکدیگر ندارند ایجاد می گردد. در شکل (۲-۱۶) نمونه ای از درختان جستجو را مشاهده می کنید. گره های گراف g هر یک سه برچسب فعال، غیر فعال و یا آزاد^۱ را می توانند داشته باشند. گره های آزاد متعلق به هیچ یک از دو درخت δ و T نمی باشند. گره های فعال در هر درخت نمایشگر مرز خارجی درخت با سایر گره های گراف g می باشند در حالی که گره های غیر فعال در داخل درخت قرار دارند. گره های فعال یک درخت، گره های آزاد همسایه را که دارای شرایط مناسبی باشند به درخت ملحق می کنند که به این ترتیب به درختان اجازه گسترش می دهند [۲۱].

^۱ free

برای ایجاد یک مسیر تقویت کننده در گراف لازم است که گره فعالی از یک درخت با گره فعالی در درخت همسایه روبرو شود. این الگوریتم به صورت تکراری سه مرحله زیر را انجام می دهد [۲۱]:

۱. مرحله رشد^۱: درختان جستجو $\mathcal{S}$ و $\mathcal{T}$ رشد می کنند تا مسیری از $s \rightarrow t$ یافت شود.
۲. مرحله تقویت^۲: مسیر یافت شده تقویت می شود و درختان جستجو به اجزای کوچکتری تبدیل می شوند.
۳. مرحله پذیرش^۳: درخت های $\mathcal{S}$ و $\mathcal{T}$ به حالت اولیه باز گردانده می شوند. در ادامه جزئیات مراحل رشد، تقویت و پذیرش را شرح می دهیم .



شکل (۲-۱۶) : نمونه ای از درختان جستجو $\mathcal{S}$ (گره های قرمز) و $\mathcal{T}$ (گره های آبی) را در انتهای مرحله رشد نشان می دهد و مسیر یافت شده از ترمینال s به t با خط زرد مشخص شده است. P و A به ترتیب نمایشگر گره های فعال و غیر فعال می باشند و گره های آزاد نیز با رنگ مشکی نمایش داده شده اند [۲۱].

۲-۶-۸-۱ مراحل اجرای الگوریتم

در این الگوریتم گره های فعال (A) و گره های والد از دست داده^۱ (O) ذخیره می شوند در حالی که در الگوریتم مبتنی بر مسیر تقویت کننده [۲۴]، مقدار فلو f و گراف g_f ذخیره می شد. ساختار کلی الگوریتم جدید به صورت شکل (۲-۱۷) می باشد [۲۱]:

^۱ growth stage
^۲ augmentation stage
^۳ adoption stage

```

initialize :       $S = \{s\}$  ,  $T = \{t\}$  ,  $A = \{s,t\}$  ,  $O = \emptyset$ 
while true
    grow  $S$  or  $T$  to find an augmenting path  $P$  from  $s$  to  $t$ 
    if  $P = \emptyset$  terminate
    augment on  $P$ 
    adopt orphans
end while

```

شکل (۲-۱۷) : ساختار کلی الگوریتم جدید [۲۱].

وابستگی هر گره p به درختان جستجو $\mathcal{S}$ و $\mathcal{T}$ توسط تابع $TREE(p)$ نمایش داده می شود به طوری که

$$TREE(p) = \begin{cases} \mathcal{S} & \text{if } p \in \mathcal{S} \\ \mathcal{T} & \text{if } p \in \mathcal{T} \\ \emptyset & \text{if } p \text{ is free} \end{cases} \quad (۲-۱۹)$$

و در صورتی که این گره جزء یکی از درختان جستجو باشد اطلاعات مربوط به والدین آن به صورت تابع $PARENT(p)$ در نظر گرفته می شود. برای ترمینال های s و t (ریشه های درختان جستجو)، گره های والد از دست داده (O) و همچنین گره های آزاد که هیچ یک والدینی ندارند مقدار تابع $PARENT(p) = \cdot$ در نظر گرفته می شود.

برای توصیف وزن باقیمانده لبه ها از نماد $tree-cap(p \rightarrow q)$ استفاده می کنیم بدین معنی که لبه (p,q) ای موجود است که $TREE(p) = \mathcal{S}$ و یا لبه (q,p) ای وجود دارد که $TREE(p) = \mathcal{T}$ می باشد. در صورتی که چنین لبه ای، غیر اشباع باشد گره p برای گره q در همان درخت، به عنوان والدین با ارزشی محسوب می شود [۲۱].

^۱ orphan

۲-۶-۸-۱-۱ مرحله رشد

در این مرحله گره های فعال از مجموعه گره های آزاد مجاورشان که توسط لبه های غیر اشباعی به یکدیگر متصل شده اند فرزندان جدیدی به دست می آورند. سپس این فرزندان جدید اعضای درخت متناظر شده و به این ترتیب درختان جستجو گسترش می یابند. چنانچه گره فعالی با گره فعال مجاورش در درخت دیگر روبرو شود مرحله رشد خاتمه یافته و مسیری از ترمینال s به t به دست می آید [۲۱]. چنین مسیری در شکل (۲-۱۶) نشان داده شده است. الگوریتم این مرحله را در شکل (۲-۱۸) ملاحظه می کنید.

```

while  $A \neq \emptyset$ 
    pick an active node  $p \in A$ 
    for every neighbor  $q$  such that  $tree\_cap(p \rightarrow q) > 0$ 
        if  $TREE(q) = \emptyset$  then add  $q$  to search tree as an active node:
             $TREE(q) := TREE(p)$  ,  $PARENT(q) := p$  ,  $A := A \cup \{q\}$ 
        If  $TREE(q) \neq \emptyset$  and  $TREE(q) \neq TREE(p)$  return  $P = PATH_{s \rightarrow t}$ 
    end for
    remove  $p$  from  $A$ 
end while
return  $P = \emptyset$ 

```

شکل (۲-۱۸) : الگوریتم مرحله رشد [۲۱].

۲-۶-۸-۱-۲ مرحله تقویت

طی این مرحله مسیر P که در مرحله رشد پیدا شد، تقویت می شود. به این معنی که الگوریتم با فرستادن ماکزیمم فلو (Δ) طوری آن را تقویت می کند که وزن های لبه های موجود در مسیر رفت به مقدار Δ کاهش یافته در حالی که وزن لبه های مسیر برگشت به اندازه Δ افزایش می یابد. پس بعضی لبه ها در این مسیر اشباع می شوند (ارزش لبه صفر می شود). بنابراین گره هایی که در درختان جستجو از طریق چنین لبه های اشباع شده ای به والدین شان متصل شده اند، به عنوان گره های والد از دست داده در نظر گرفته می شوند. بنابراین مجموعه O که در ابتدای این مرحله خالی از عضو بود اعضای پیدا می کند. در این مرحله درختان جستجوی S و T به درختان کوچکتری تبدیل می شوند. ترمینال های s و t همچنان ریشه های دو درخت باقی می مانند در حالی که گره های والد از دست داده ریشه های درختان دیگر را تشکیل می دهند [۲۱]. الگوریتم این مرحله در شکل (۲-۱۹) نشان داده شده است.

find the bottleneck capacity Δ on P

update the residual graph by pushing flow Δ through P

for each edge (p,q) in P that becomes saturated

if $TREE(p) = TREE(q) = S$ then set $PARENT(q) := \emptyset$ and $O := O \cup \{q\}$

if $TREE(p) = TREE(q) = T$ then set $PARENT(p) := \emptyset$ and $O := O \cup \{p\}$

end for

شکل (۲-۱۹) : الگوریتم مرحله تقویت [۲۱].

۲-۶-۸-۱-۳ مرحله پذیرش

در این مرحله گره های والد از دست داده که در مجموعه O قرار دارند مورد بررسی قرار گرفته تا جایی که مجموعه O خالی از عضو شود. طی این الگوریتم برای هر گره والد از دست داده، والد جدیدی در نظر گرفته می شود به طوری که والد جدید باید از طریق لبه غیر اشیاعی به گره مورد بررسی در مجموعه O متصل شده باشد. در صورتی که والدینی برای گره والد از دست داده یافت نشود، به عنوان یک گره آزاد در نظر گرفته شده و تمام فرزندان به مجموعه O می پیوندند. این مرحله زمانی خاتمه می یابد که هیچ گره والد از دست داده ای باقی نماند. از این رو ساختار درختان جستجو T و δ به حالت اولیه برگردانده می شود [۲۱]. الگوریتم این مرحله در شکل (۲-۲۰) نشان داده شده است.

While $O \neq \emptyset$

Pick an orphan node $p \in O$ and remove it from O

Process p

end while

شکل (۲-۲۰): الگوریتم مرحله پذیرش [۲۱].

طی مرحله پذیرش، الگوریتم تلاش می کند از میان همسایه های گره p ، والدین با ارزش جدیدی مثل q بیابد که این والد جدید باید شرایط زیر را داشته باشد [۲۱]:

• $TREE(p) = TREE(q)$ به این معنا که گره والد از دست داده p و والدین جدیدش (q)

باید در یک درخت جستجو قرار داشته باشند.

- $tree-cap(q \rightarrow p) > 0$ یعنی باید لبه غیر اشباع (q, p) بین گره p و والد q وجود داشته باشد.
- منشا q باید از ترمینال های s یا t باشد.

توجه داشته باشید که آخرین شرط لازم و ضروری است به این دلیل که طی مرحله پذیرش بعضی گره ها در درختان جستجو $\mathcal{S}$ و $\mathcal{T}$ از گره های والدین از دست داده ناشی می شوند.

حال اگر q به عنوان والدین p در نظر گرفته شود مقدار تابع $PARENT(p) = q$ خواهد شد. در چنین حالتی گره p در همان درخت جستجو باقی مانده و وضعیت فعال یا غیر فعال بودنش تغییر نمی کند. در غیر این صورت p به یک گره آزاد تبدیل شده و تمام همسایه هایش که از طریق لبه های غیر اشباعی به آن متصل شده اند به صورت گره های فعال در نظر گرفته می شوند. پس از اتمام مرحله پذیرش، الگوریتم به مرحله رشد برگشته و مراحل تکرار می شود [۲۱].

الگوریتم زمانی خاتمه می یابد که درختان جستجو $\mathcal{S}$ و $\mathcal{T}$ قادر به رشد نباشند و در چنین صورتی درختان توسط لبه های اشباع شده از یکدیگر مجزا می شوند و در نهایت ماکزیمم فلو روی گراف به دست می آید [۲۱].

۷-۲ مقایسه و تحلیل نتایج

در این بخش نتایج حاصل از روش ناحیه بندی گراف کات را به ازای پارامترهای λ و σ بررسی کرده و مورد ارزیابی قرار می دهیم. در شکل های (۲-۲۱) و (۲-۲۲) تصویر اصلی و نتایج ناحیه بندی به ازای مقادیر مختلف این پارامترها نشان داده شده است.

لازم به ذکر است که در شکل های زیر در قسمت (a)، تصاویر اصلی و بذرهای مربوط به هر یک از نواحی شی و پس زمینه به ترتیب با رنگ های سفید و مشکی مشخص شده است. در قسمت (b) تصویر ground truth متناظر با هر شکل را ملاحظه می کنید که در این تصویر ناحیه شی به رنگ قرمز مشخص شده است. در قسمت (c)، ناحیه بندی انتخاب شده توسط کلاسه بند را ملاحظه می کنید. در این تصویر ناحیه پس زمینه با رنگ مشکی مشخص شده است.



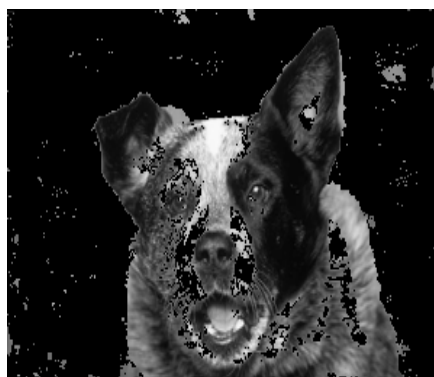
(a)



(b)



(c)



(d)

شکل (۲۱-۲) : (a) تصویر اصلی به همراه بذرها (بذرهای مربوط به شی با رنگ سفید و بذرهای مربوط به پس زمینه با رنگ مشکی نشان داده شده است)، (b) تصویر ناحیه بندی شده به ازای $\lambda = 0.001$ و $\sigma = 10$ ، (c) تصویر ناحیه بندی شده به ازای $\lambda = 0.001$ و $\sigma = 9$ ، (d) تصویر ناحیه بندی شده به ازای $\lambda = 0.001$ و $\sigma = 7$.



(a)



(b)



(c)



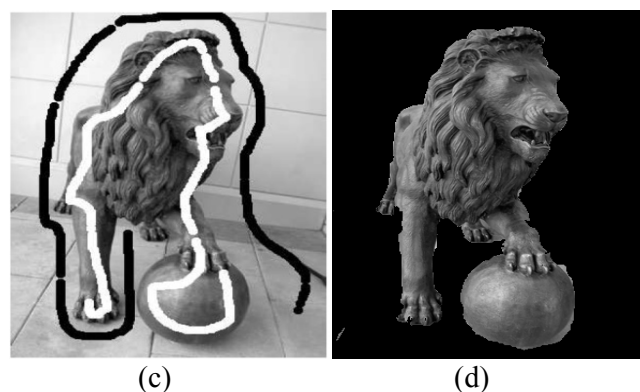
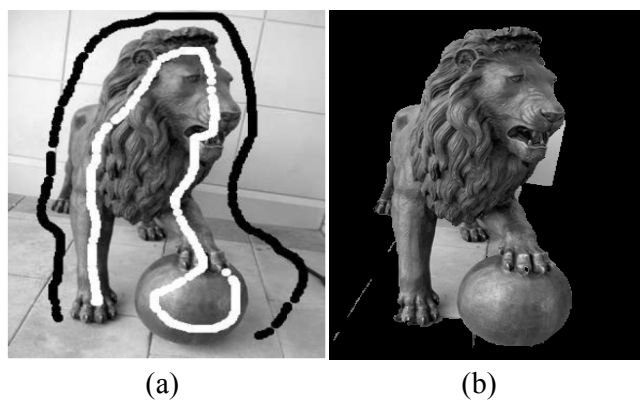
(d)

شکل (۲-۲۲) : (a) تصویر اصلی به همراه بذرها (بذرهای مربوط به شی با رنگ سفید و بذرها مربوط به پس زمینه با رنگ مشکی نشان داده شده است)، (b) تصویر ناحیه بندی شده به ازای $\lambda = 0.001$ و $\sigma = 10$ ، (c) تصویر ناحیه بندی شده به ازای $\lambda = 0.101$ و $\sigma = 9$ ، (d) تصویر ناحیه بندی شده به ازای $\lambda = 0.801$ و $\sigma = 7$.

برای ارزیابی تصاویر ناحیه بندی شده توسط گراف کات به ازای λ و σ های مختلف معیار های کیفی مختلفی وجود دارد. یکی از معیارهای کیفی رایج ، معیار بصری است. بر این اساس هر اندازه نواحی شی و پس زمینه همگن تر باشند تصویر مورد نظر بهتر ناحیه بندی شده است. با توجه به معیار

بصری می توان گفت که در تصاویر فوق به ترتیب از (b) تا (d) از کیفیت تصاویر ناحیه بندی شده کاسته شده است. در تصاویر قسمت های (c,d) در هر دو شکل پدیده فوق ناحیه بندی رخ داده است به دلیل این که ناحیه مربوط به شی شامل تکه های زیادی از پس زمینه می باشد. در هر دو شکل تصاویر ناحیه بندی شده به ازای $\lambda = 0.001$ و $\sigma = 10$ بهترین ناحیه بندی ها را نشان می دهند که در قسمت (b) ارائه شده اند.

لازم به ذکر است که انتخاب بذرها در روش گراف کات تاثیر زیادی در نتیجه ناحیه بندی خواهد داشت. به طوری که اگر تعداد بذرها کافی نباشند نتیجه ناحیه بندی نامطلوب خواهد بود. در حالی که با افزودن تعداد بذرها نتایج ناحیه بندی مطلوب تر خواهد شد که این موضوع نشان دهنده تاثیر تعاملات کاربر در نتیجه ناحیه بندی می باشد. برای این منظور شکل (۲-۲۳) را در نظر بگیرید.



شکل (۲-۲۳): (a) تصویر اصلی به همراه بذرها، (b) ناحیه بندی نامطلوب تصویر (a) به ازای $\lambda = 0.001$ و $\sigma = 10$ ، (c) تصویر اصلی به همراه بذرها، (d) ناحیه بندی مطلوب تصویر (c) به ازای $\lambda = 0.001$ و $\sigma = 10$.

۸-۲ نتیجه گیری

در این فصل روش ناحیه بندی تعاملی بر اساس گراف کات مرور شد. برای ناحیه بندی تصاویر به این روش، ابتدا گراف متناظر با تصویر تشکیل می شود، سپس مینیمم کات روی گراف، ناحیه بندی بهینه ای را ایجاد می کند. برای محاسبه مینیمم کات روی گراف، الگوریتم جدیدی در این فصل مرور شده است. در روش ناحیه بندی گراف کات نتایج ناحیه بندی به شدت به پارامترهای λ و σ وابسته است. نتایج بدست آمده به ازای پارامترهای مختلف، به وسیله معیار بصری قابل مقایسه می باشند. آزمایش ها برای تصاویر مختلف و به ازای مقادیر متفاوتی از پارامترهای λ و σ انجام شدند.

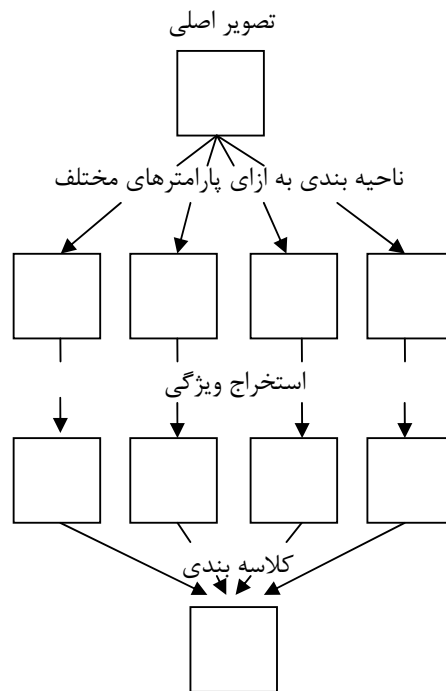
فصل سوم

ارزیابی خودکار ناحیه بندی توسط الگوریتم Adaboost

۱-۳ مقدمه

در این فصل روشی پیشنهاد می کنیم که قادر است نتایج ناحیه بندی تصاویر به ازای پارامترهای مختلف λ و σ در الگوریتم گراف کات را به طور خودکار ارزیابی کند. سپس به کمک روش پیشنهادی بهترین ناحیه بندی را برای یک تصویر به صورت اتوماتیک بدست می آوریم. برای ارزیابی ناحیه بندی یک تصویر، ویژگی هایی را از تصویر نتیجه که مبتنی بر همگن بودن نواحی از لحاظ روشنایی، بافت و همچنین سادگی مرزی نواحی می باشد استخراج می کنیم. سپس به کمک نتیجه ناحیه بندی تصاویر آموزشی یک کلاسه بند Adaboost را آموزش می دهیم تا بر اساس ویژگی های نتیجه ناحیه بندی تشخیص دهد آیا ناحیه بندی قابل قبول است یا خیر.

همان طور که نتایج نشان می دهند پس از آموزش، کلاسه بند قادر است تا از میان ناحیه بندی های به دست آمده از تصویر ورودی که به ازای پارامترهای مختلف حاصل شده اند، ناحیه بندی مطلوب را انتخاب نماید. این مراحل به صورت شماتیک در شکل (۱-۳) نشان داده شده است. در ادامه این فصل ابتدا نحوه استخراج ویژگی از تصویر ناحیه بندی شده و سپس کلاسه بند Adaboost را معرفی می کنیم و پس از آن، نتایج کلاسه بند Adaboost را در ارزیابی ناحیه بندی تصاویر بررسی خواهیم کرد.



شکل (۳-۱): مراحل کلاسه بندی

۳-۲ ویژگی های ناحیه بندی مناسب

در این بخش برای ارزیابی نتایج ناحیه بندی یک تصویر، ویژگی هایی را از نتیجه ناحیه بندی یک تصویر به ازای λ و σ های مختلف استخراج می کنیم. مطابق با مرجع [۱۷] ما از شش ویژگی برای ارزیابی ناحیه بندی استفاده می کنیم. این ویژگی ها عبارتند از: دو ویژگی مربوط به همگن بودن شدت روشنایی در هر ناحیه، زاویه بردار گرادیان در مرز نواحی، همگن بودن بافت در هر ناحیه و پیچیدگی مرز نواحی. در ادامه در روابط به کار رفته از I به عنوان تصویر ورودی و از S به عنوان تصویر ناحیه بندی شده استفاده می کنیم.

۱-۲-۳ ویژگی های مبتنی بر شدت روشنایی در ناحیه بندی

برای ارزیابی همگن بودن روشنایی در ناحیه شی و متمایز بودن متوسط روشنایی شی از پس زمینه از دو ویژگی مطابق با مرجع [۱۷] استفاده می کنیم. معمولاً پیکسل های تصویر در ناحیه شی تغییرات روشنایی کمی دارند، در حالی که روشنایی پیکسل های دو طرف مرز شی و پس زمینه اختلاف نسبتاً زیادی دارند. اولین ویژگی شدت روشنایی، میانگین قدر مطلق اختلاف روشنایی پیکسل ها در ناحیه شی و مرز شی و پس زمینه را اندازه می گیرد و به صورت زیر تعریف می گردد [۱۷]:

$$F_{intensity}(I, S) = \frac{\sum_{(p,q) \in B} |I_p - I_q|}{|B|} - \frac{\sum_{(p,q) \in O} |I_p - I_q|}{|O|} \quad (1-3)$$

در این معیار I_p و I_q به ترتیب شدت روشنایی پیکسل های p و q می باشد و B مجموعه زوج پیکسل های همسایه که روی مرز شی و پس زمینه قرار گرفته اند می باشد:

$$B = \{(p, q) | S_p = 1, S_q = 0 \text{ or } S_p = 0, S_q = 1\} \quad (2-3)$$

در رابطه (۱-۳) O مجموعه زوج پیکسل های همسایه داخل ناحیه شی را نمایش می دهد و اندازه مجموعه های O و B نیز به ترتیب با $|O|$ و $|B|$ مشخص شده اند.

$$O = \{(p, q) | S_p = 1, S_q = 1\} \quad (3-3)$$

لازم به ذکر است که در روابط فوق $S_p = 0$ و $S_p = 1$ به این معنا است که پیکسل p به ترتیب متعلق به شی و پس زمینه می باشد.

در رابطه (۱-۳) اولین جمله، متوسط اختلاف روشنایی پیکسل های مرزی و دومین جمله متوسط اختلاف روشنایی پیکسل های مربوط به ناحیه شی را نشان می دهد. از آن جایی که در یک ناحیه بندی خوب، نواحی شی و پس زمینه همگن هستند اختلافات مرزی خیلی بزرگتر از اختلافات ناحیه شی می باشد. بنابراین برای تصویری که به طور مناسب ناحیه بندی شده است، انتظار داریم ویژگی

$F_{intensity}$ مقدار مثبت و بزرگی داشته باشد. همچنین می توان هیستوگرام قدرمطلق اختلاف روشنایی زوج پیکسل ها را در مجموعه های O و B با یکدیگر مقایسه کرد. برای اندازه گیری عدم تشابه این هیستوگرام ها از رابطه زیر استفاده می کنیم [۱۷]:

$$F_{intensity}(I, S) = \sum_i \frac{(H(i, B) - H'(i))^2}{H'(i)} \quad (۴-۳)$$

$$H'(i) = \frac{[H(i, B) + H(i, O)]}{2} \quad (۵-۳)$$

در روابط فوق $H(i, X)$ مقدار هیستوگرام X در بین i ام و $H'(i)$ مقدار متوسط هیستوگرام های ناحیه شی و مرز شی و پس زمینه برای این بین می باشد. همچنین در این روابط B و O طبق روابط (۲-۳) و (۳-۳) تعریف می شوند.

۳-۲-۲ ویژگی زاویه گرادیان در مرز نواحی

بردار گرادیان در هر پیکسل تصویر ویژگی مهم دیگری است که در ارزیابی ناحیه بندی باید مورد توجه قرار گیرد [۱۷]. برای یک ناحیه بندی خوب انتظار می رود گرادیان در بعضی نقاط روی مرز به صورت یکنواخت تغییر کند. اما از آن جایی که اندازه گرادیان روی مرز شی و پس زمینه مقدار خیلی کوچکی دارد بنابراین از زاویه گرادیان در این ویژگی به صورت زیر استفاده می شود [۱۷]:

$$F_{gradient}(I, S) = \frac{\sum_{(p,q) \in B} |\vec{G}(p) - \vec{G}(q)|}{|B|} \quad (۶-۳)$$

در رابطه (۶-۳) $\vec{G}(p)$ زاویه گرادیان در پیکسل p و B طبق رابطه (۲-۳) مجموعه زوج پیکسل های روی مرز شی و پس زمینه را نشان می دهد. در این رابطه $F_{gradient}$ میانگین اختلاف زاویه گرادیان پیکسل های مرزی را نشان می دهد. در یک تصویر با ناحیه بندی مناسب که گرادیان روی

^۱ bin

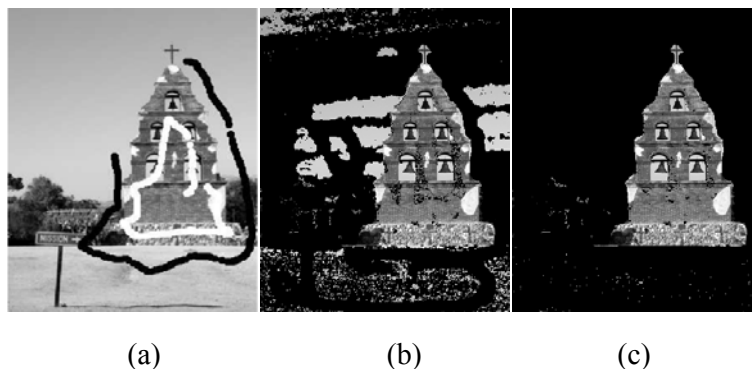
مرز شی و پس زمینه به صورت یکنواخت تغییر می کند اندازه این ویژگی کوچک خواهد بود. ویژگی دیگری که بر اساس زاویه گرادیان در مرجع [۱۷] مطرح شده است به صورت زیر تعریف می گردد:

$$F_{gradient}(I, S) = \frac{\sqrt{\sum_{(p,q) \in B} (|\vec{G}(p) - \vec{G}(q)| - \overline{G})^2}}{|B|} \quad (7-3)$$

در این رابطه $\overline{G}$ میانگین اختلاف زاویه های گرادیان روی مرز می باشد.

۳-۲-۳ ویژگی گوشه های مرز

در یک تصویر که به طور مناسب ناحیه بندی شده است معمولا مرزهای نواحی ساده و تعداد نقاط گوشه^۱ ای در آن کم می باشد. برای استخراج گوشه های یک مرز ابتدا تابع انحنا^۲ را روی نقاط مرز محاسبه نموده و سپس گوشه های مرز نقاطی هستند که مشتق تابع انحنا در آن ها برابر با صفر بوده و اندازه تابع انحنا از یک مقدار آستانه ای بیشتر باشد. تعداد گوشه ها را در تصویر ناحیه بندی شده S با ویژگی $F_{boundary}$ نشان می دهیم. در مورد تصاویری با ناحیه بندی خوب انتظار داریم $F_{boundary}$ مقدار کوچکی داشته باشد که این نشانه ساده بودن مرزهای دو ناحیه است [۱۷].



شکل (۳-۲): (a) تصویر اصلی به همراه بذرها (بذره های مربوط به شی با رنگ سفید و بذره های مربوط به پس زمینه با رنگ مشکی نشان داده شده است)، (b) ناحیه بندی تصویر اصلی به ازای $\lambda = 0.5$ و $\sigma = 2$ با مرزهای پیچیده، (c) ناحیه بندی تصویر اصلی به ازای $\lambda = 0.01$ و $\sigma = 3$ با مرزهای ساده.

^۱ corner
^۲ curvature

۳-۲-۴ ویژگی بافت

توجه به بافت های همگن در ناحیه بندی تصاویر دارای بافت، بسیار مهم می باشد. یک ناحیه با بافت همگن دارای ساختاری است که با الگوی معینی تکرار می شود. می توان به بافت ناحیه برگ ها در یک درخت، بافت چمن، ابرها و ... اشاره کرد. پرپود، جهت و پیچیدگی سه ویژگی مهم بافت ها هستند که باعث می شود انسان بتواند بافت های مختلف را از یکدیگر تفکیک کند. از آن جا که پرپود و جهت بافت را می توان به ترتیب با فرکانس و زاویه چرخش فیلترهای میان گذر توصیف نمود، برای استخراج ویژگی بافت از فیلترهای گبور^۱ استفاده می شود که در آن انتخاب فرکانس و زاویه امکان پذیر است [۲۸][۲۹]. تابع گبور به صورت زیر تعریف می شود [۳۰]:

$$g_{\lambda, \theta, \varphi}(x, y) = e^{-((x')^2 + y'^2)/2\sigma^2)} \cos \left(2\pi \frac{x'}{\lambda} + \varphi \right) \quad (۸-۳)$$

$$x' = x \cos \theta + y \sin \theta, \quad y' = -x \sin \theta + y \cos \theta$$

$$\sigma = 0.56 \lambda, \quad \gamma = 0.5$$

که در رابطه فوق $g(x, y)$ یک تابع گبور دو بعدی و $\frac{1}{\lambda}$ ، θ و φ به ترتیب فرکانس، زاویه چرخش و کرنل تابع گبور می باشند.

تعدادی از محققین در مراجع [۳۱][۳۲][۳۳] برای استخراج ویژگی مبتنی بر بافت از بانکی از فیلترهای گبور استفاده می کنند. کرنل $\varphi = 0$ در مرجع [۳۰] برای بانک فیلتر گبور مناسب بیان شده است. در این مرجع، از بانکی شامل ۲۴ فیلتر استفاده می شود که در نتیجه استفاده از ۳ فرکانس و ۸ زاویه چرخش متفاوت بدست می آید. فرکانس و زوایای چرخش فیلترها در این تحقیق بر اساس پیشنهادات ارائه شده در این مرجع تنظیم شده است. زوایای چرخش لازم برای تشکیل چنین فیلتر بانکی عبارتند از [۳۰]:

^۱ Gabor filter

$$\theta = k \left(\frac{\pi}{\lambda} \right) \quad k = 0, 1, 2, \dots, 7 \quad (9-3)$$

و فرکانس های استفاده شده برای این فیلترها مقادیر ۲۳ و ۳۱ و ۴۷ در نظر گرفته می شود [۳۰].

برای استخراج ویژگی مبتنی بر بافت، حاصل کانولوشن تصویر ورودی $I(x,y)$ را با هر یک از توابع گبور تشکیل دهنده بانک فیلتر محاسبه می کنیم. در رابطه (۳-۱۰) حاصل کانولوشن تصویر ورودی با یکی از توابع گبور نشان داده شده است [۳۰]:

$$r(x,y) = \iint_{\Omega} I(\xi,\eta) g(x-\xi, y-\eta) d\xi d\eta \quad (10-3)$$

در رابطه فوق Ω مجموعه نقاط تصویر، g یک فیلتر گبور موجود در بانک و r تصویر ویژگی متناظر با این فیلتر گبور می باشد. پس خروجی چنین بانک فیلتری ۲۴ تصویر ویژگی است و یا به بیان دیگر می توان گفت که خروجی، یک بردار ویژگی ۲۴ بعدی در هر نقطه از تصویر ورودی می باشد [۳۰].

حال برای مقایسه بافت شی و پس زمینه، می بایست ۲۴ هیستوگرام با توجه به خروجی بانک فیلتر به دست آمده به صورت جداگانه برای نواحی شی و پس زمینه در تصویر ناحیه بندی شده K ترسیم کنیم. سپس $F_{texture}$ را به صورت فاصله بین هیستوگرام ها طبق رابطه (۳-۴) محاسبه می کنیم. در یک تصویر که به نحو مطلوب ناحیه بندی شده اختلاف ویژگی بافت استخراج شده از شی و پس زمینه مقدار بزرگی است این در حالی است که برای ناحیه بندی نامطلوب این مقدار کوچک می باشد [۱۷].

۳-۳ الگوریتم Adaboost

Adaboost یکی از الگوریتم های کلاسه بندی داده ها می باشد که قادر است به کمک مجموعه ای از کلاسه بندهای ساده و نسبتاً ضعیف یک کلاسه بند توانا را ایجاد کند. این ساختار در شکل (۳-۳) نشان داده شده است [۳۴].



شکل (۳-۳) : ساختار کلی Adaboost [۳۴].

با توجه به این ساختار، مجموعه داده های آموزشی و کلاسه بندهای ساده، ورودی های الگوریتم Adaboost می باشند. در مسئله مورد مطالعه مجموعه آموزشی، تصاویر ناحیه بندی شده توسط الگوریتم گراف کات به ازای مقادیر مختلف λ و σ می باشند که به صورت دستی توسط یک ناظر به دو کلاس ناحیه بندی "خوب" و "بد" تفکیک شده اند، بدین ترتیب که تصاویر ناحیه بندی شده مطلوب در کلاس خوب برچسب $\gamma = 1$ و تصاویر ناحیه بندی شده نامطلوب در کلاس بد برچسب $\gamma = 0$ می گیرند. کلاسه بندهای ساده با استفاده از شش ویژگی که از تصاویر ناحیه بندی شده آموزشی استخراج شده ایجاد می گردند. هدف کلی الگوریتم Adaboost این است که با ترکیب خطی کلاسه بندهای ساده مبتنی بر این ویژگی ها کلاسه بندی نهایی را آموزش دهد [۳۴]. در ادامه به توضیح مختصری در مورد نحوه ایجاد کلاسه بندهای ساده می پردازیم.

۳-۳-۱ کلاسه بندهای ساده

در الگوریتم Adaboost کلاسه بندهای ساده بر اساس ویژگی های استخراج شده از پترن های ورودی طراحی می شوند. با توجه به این که از تصاویر ناحیه بندی شده شش ویژگی استخراج نمودیم، شش کلاسه بند ساده در طراحی Adaboost به کار گرفته می شود. اما فقط یکی از این کلاسه بندها در هر تکرار توسط الگوریتم Adaboost انتخاب می شود [۳۴].

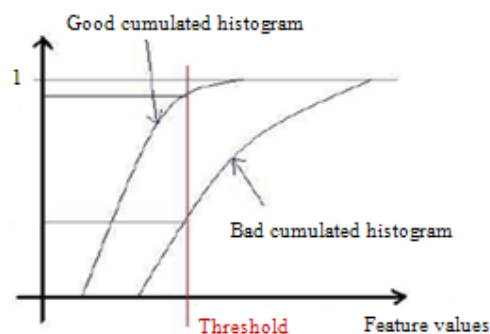
کلاسه بند ساده ویژگی z ام، $h_j(x)$ به صورت زیر تعریف می گردد:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \quad (11-3)$$

در رابطه (۱۱-۳) f_j ویژگی z ام، θ_j آستانه متناظر با این ویژگی و p_j ضریب توازن^۱ است که جهت نامساوی را تغییر می دهد. در این رابطه x یک تصویر ناحیه بندی شده آموزشی وزن دار می باشد که وزن تمام تصاویر آموزشی در هر مرحله از الگوریتم تغییر می یابد. لازم به ذکر است که ضریب توازن p_j توسط ناحیه بند به صورت ۱ و یا -۱ در نظر گرفته می شود.

مطابق رابطه (۱۱-۳) برای ایجاد کلاسه بند ساده h_j به تعیین آستانه متناظر با ویژگی z ام یعنی θ_j نیاز داریم. برای این منظور هیستوگرام های تجمعی^۲ مربوط به ویژگی f_j را برای تصاویر آموزشی ناحیه بندی شده هر دو کلاس خوب و بد را به دست آورده، آستانه مناسب θ_j جایی قرار می گیرد که مقدار هیستوگرام های تجمعی دو کلاس بیشترین اختلاف را داشته باشند [۳۴]. در شکل (۳-۴) هیستوگرام های تجمعی دو کلاس و آستانه مناسب برای آن ها نشان داده شده است. پس از تعیین آستانه مناسب متناظر با هر ویژگی، کلاسه بندهای ساده می توانند هر یک از تصاویر آموزشی را برحسب یکی از ویژگی هایشان طبق رابطه (۱۱-۳) با برچسب های ۱ و ۰ از یکدیگر تفکیک کنند.

^۱ parity
^۲ cumulative histogram



شکل (۳-۴): آستانه مناسب جایی است که هستوگرام های تجمعی دو کلاس بیشترین اختلاف را با یکدیگر دارند [۳۴].

۳-۳-۲ آموزش Adaboost

پس از تعیین ورودی های Adaboost ابتدا باید کلاسه بند را آموزش دهیم تا بتواند کلاس ناحیه بندی تصاویر اعمال شده به ورودی را تشخیص دهد. برای آموزش Adaboost از الگوریتم (۳-۵) استفاده می کنیم [۳۵].

همان طور که در این الگوریتم ملاحظه می گردد برای آموزش کلاسه بند از N تصویر ناحیه بندی شده آموزشی برچسب گذاری شده $(x_1, y_1), \dots, (x_N, y_N)$ استفاده می شود که x_i امین تصویر آموزشی و y_i برچسب متناظر با این تصویر می باشد. همان طور که در بخش قبل نیز اشاره شد تصاویر ناحیه بندی شده آموزشی توسط الگوریتم گراف کات به ازای λ و σ های مختلف حاصل شده اند که توسط کاربر به دو کلاس خوب و بد به ترتیب با برچسب های $y = 1$ و $y = 0$ تفکیک می شوند. با شروع الگوریتم به تمام N تصویر آموزشی وزن یکسان $w_i^1 = 1/N$ اختصاص داده می شود. لازم به ذکر است که w_i^1 نشان دهنده وزن i امین نمونه در اولین تکرار می باشد. سپس الگوریتم گام های ۱ تا ۵ را T بار تکرار می کند. در اولین گام با نرمالیزه کردن وزن ها توزیع احتمالاتی نمونه های ورودی p^t محاسبه می شود که t مقداری در محدوده ۱ تا T را می پذیرد.

Input sequence of N labeled examples $\langle (x_1, y_1), \dots (x_N, y_N) \rangle$

distribution $\mathbf{D}$ of the examples

weak learning algorithm **weakLearn**

integer T specifying number of iterations

Initialize the weight vector : $w_i^1 = D(i)$ for $i=1, \dots, N$

Do for $t=1, \dots, T$

1. Set

$$\mathbf{p}^t = \frac{\mathbf{w}^t}{\sum_{i=1}^N w_i^t}$$

2. Call **WeakLearn**, providing it with the distribution $\mathbf{p}^t$; get back a hypothesis $h_t: X \rightarrow [-1]$.

3. calculate of error $h_t: \epsilon_t = \sum_{i=1}^N p_i^t |h_t(x_i) - y_i|$.

4. Set $\beta_t = \epsilon_t / (1 - \epsilon_t)$.

5. Set the new weight vectors to be

$$w_i^{t+1} = w_i^t \beta_t^{1 - |h_t(x_i) - y_i|}$$

Output the hypothesis

$$h_f(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T (\log 1 / \beta_t) h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \log 1 / \beta_t \\ -1 & \text{otherwise} \end{cases}$$

شكل (٣-٥) : الگوریتم Adaboost [٣٥].

سپس در گام بعدی الگوریتم با استفاده از وزن های نرمالیزه شده و طبق رابطه (۳-۱۱) کلاسه بندهای ساده ای ایجاد می کند. در سومین گام اختلاف برچسب واقعی تصاویر ناحیه بندی شده آموزشی " y_i " با نتایج کلاسه بندهای ساده " $h_t(x_i)$ " محاسبه شده و متوسط خطای وزن دار ϵ_t با کمک توزیع p^t محاسبه می شود. از آن جایی که شش ویژگی و نتیجتاً شش کلاسه بند ساده مورد استفاده قرار می گیرد بنابراین شش خطای ϵ_t متناظر با آنها ایجاد می گردد. اما الگوریتم در هر تکرار فقط کلاسه بندی که کمترین مقدار خطای ϵ_t را ایجاد کرده، انتخاب می کند سپس مقدار ضریب β_t متناظر با این خطا را محاسبه می نماید. نتایج تجربی نشان می دهد که کلاسه بندهای ساده بیشتر از ۵۰٪ نمونه ها را به درستی کلاسه بندی می کنند [۳۴] بنابراین مقدار خطای $\gamma - \epsilon_t \leq 0.5$ می باشد که γ در محدوده بین ۰ تا ۰.۵ تغییر می کند [۳۵]، پس با توجه به گام چهارم الگوریتم، مقدار $1 \leq \beta_t \leq 0$ می باشد.

در آخرین گام الگوریتم وزن تصاویر آموزشی طبق رابطه زیر تغییر می کند:

$$w_i^{t+1} = w_i^t \beta_t^{|h_t(x_i) - y_i|} \quad (۳-۱۲)$$

در رابطه (۳-۱۲) w_i^t و w_i^{t+1} به ترتیب وزن تصویر آموزشی i ام در تکرار t و $t+1$ می باشد. همچنین مقدار $|h_t(x_i) - y_i| - 1$ در این رابطه تابع هزینه نامیده می شود.

طی الگوریتم Adaboost تصاویر آموزشی که توسط کلاسه بندهای ساده به درستی کلاسه بندی شده باشند یعنی $h_t(x_i) = y_i$ ، مقدار تابع هزینه بزرگی خواهند داشت. اما با توجه به مقدار β_t که در محدوده ۰ تا ۱ تغییر می کند، وزن کمتری به این تصاویر در تکرارهای بعدی اختصاص داده می شود. در حالی که تصاویر آموزشی که توسط کلاسه بندهای ساده به اشتباه کلاسه بندی شده اند $h_t(x_i) \neq y_i$ مقدار تابع هزینه کوچکتری خواهند داشت و در تکرارهای بعدی وزن بیشتری می گیرند. الگوریتم زمانی خاتمه می یابد که مقدار خطای $\epsilon_t = 0$ باشد اما اکثر اوقات خطای کلاسه بندهای ساده برابر با صفر نمی شود، بنابراین الگوریتم پس از T تکرار خاتمه می یابد.

خروجی الگوریتم، کلاسه بند نهایی h_f می باشد که می تواند با اعمال یک تصویری ناحیه بندی شده به ورودی، تشخیص دهد ناحیه بندی خوب یا بد بوده است. بدین صورت که چنانچه تصویر x به ورودی این کلاسه بند اعمال شود، ترکیبی از کلاسه بندهای ساده انتخابی در هر مرحله از آموزش که در ضریب $\log(1/\beta_t)$ آن مرحله ضرب شده، توسط کلاسه بند نهایی محاسبه می گردد. در صورتی که نتیجه به دست آمده از مقدار آستانه $1/2 \sum_{t=1}^T \log 1/\beta_t$ بزرگتر باشد کلاسه بند، نمونه وارد شده را در کلاس خوب با برچسب ۱ و در غیر این صورت در کلاس بد با برچسب ۰ قرار می دهد.

۳-۴ ارزیابی روش پیشنهادی با نتایج ناحیه بندی

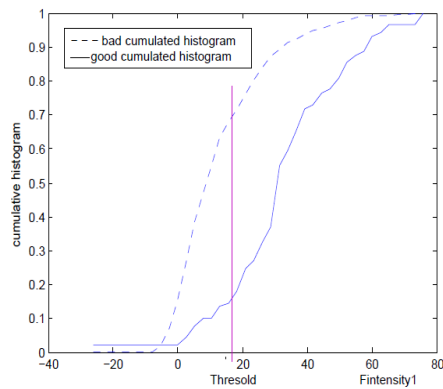
در این بخش می خواهیم تصاویر ناحیه بندی شده به ازای λ و σ های مختلف را توسط کلاسه بند Adaboost ارزیابی کنیم. برای این منظور ابتدا کلاسه بند را آموزش داده سپس برای ارزیابی تصاویر از آن استفاده می کنیم.

۳-۴-۱ نتایج آموزش کلاسه بند

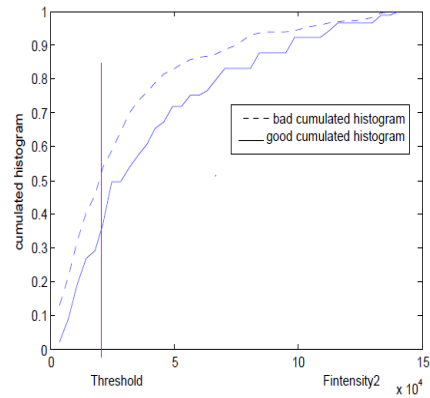
همان طور که در تئوری Adaboost اشاره کردیم ورودی های کلاسه بند، مجموعه آموزشی و کلاسه بندهای ساده می باشند. مجموعه آموزشی که در این تحقیق برای آموزش کلاسه بند استفاده نموده ایم شامل ۵۰ تصویر می باشد. هر یک از این تصاویر را به وسیله الگوریتم گراف کات و به ازای ۲۸۰ مقدار مختلف λ و σ ناحیه بندی می کنیم که λ در محدوده ۰ تا ۱ (با گام ۰/۵) و σ در محدوده ۰ تا ۱۰ (با گام ۰/۷) تغییر می کند. اما آموزش کلاسه بند با $(50 \times 280) \times 14000$ تصویر فرآیند زمان بری می باشد. به همین دلیل برای هر یک از تصاویر آموزشی ۳۰ نمونه ناحیه بندی شده را به گونه ای انتخاب می کنیم که شامل نمونه های ناحیه بندی مثبت و منفی باشد. لازم به ذکر است که نمونه های مثبت تصاویری با ناحیه بندی درست (کلاس خوب) و نمونه های منفی تصاویری با ناحیه بندی نادرست (کلاس بد) می باشند. با توجه به این که نمونه های مثبت بهترین ناحیه بندی ها

را برای هر تصویر نشان می دهند λ و σ های متناظر با آن ها پارامترهای بهینه برای هر تصویر محسوب می شوند. اما برای هر تصویر نمونه های منفی را از میان تصاویری با ناحیه بندی نادرست و با پارامترهایی حول پارامترهای بهینه متناظر با آن تصویر انتخاب می کنیم. بنابراین مجموعه آموزشی ما شامل ۱۵۰۰ (50×30) تصویر می باشد که از این میان ۸۷ تصویر نمونه مثبت و ۱۴۱۳ تصویر نمونه منفی هستند.

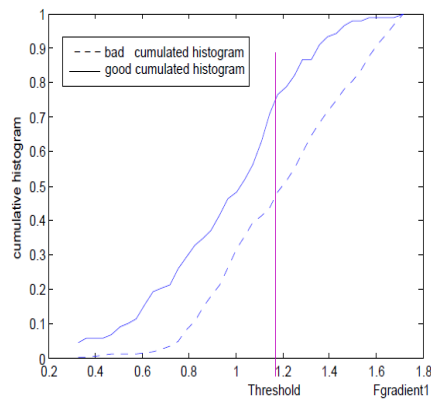
همان طور که در مباحث قبل اشاره شد برای ایجاد کلاسه بندهای ساده از شش ویژگی استفاده می کنیم. هیستوگرام های تجمعی مربوط به هر ویژگی برای نمونه های آموزشی مثبت و منفی و آستانه متناظر با هر ویژگی در اولین تکرار الگوریتم در شکل (۳-۶) نشان داده شده است.



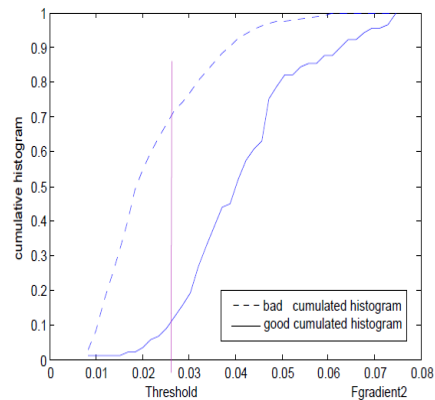
(a)



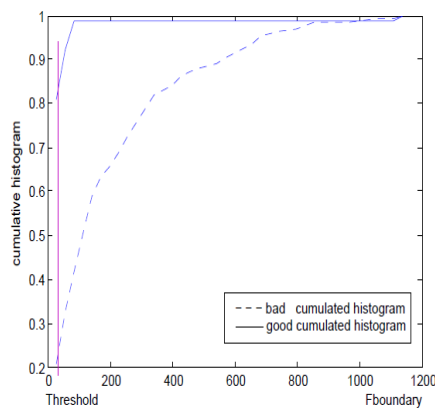
(b)



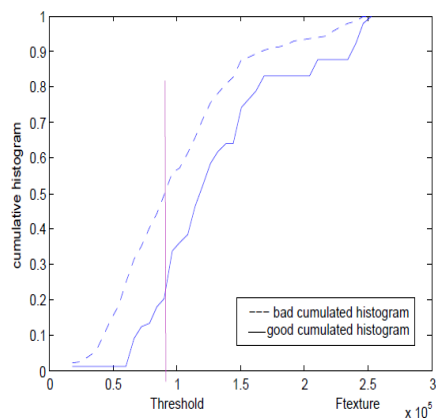
(c)



(d)



(e)



(f)

شکل (۳-۶): (a-f) به ترتیب هیستوگرام های تجمعی و آستانه های متناظر با هر یک از ویژگی های شدت روشنایی ۱،

شدت روشنایی ۲، جهت گرادیان ۱، جهت گرادیان ۲، تعداد گوشه ها و بافت را نشان می دهد.

پس با استفاده از آستانه های تعیین شده در شکل های فوق و طبق رابطه (۳-۱۱) می توان شش کلاسه بند ساده متناظر با ویژگی ها در نظر گرفت. به عنوان مثال می خواهیم کلاسه بند ساده ای برای اولین ویژگی شدت روشنایی ($F_{intensity_1}$) در نظر بگیریم. با توجه به مطالبی که در بخش قبل بیان شد می دانیم که $F_{intensity_1}$ برای تصاویری با ناحیه بندی درست در مقایسه با تصاویری با ناحیه بندی نادرست، مقدار بزرگتری خواهد داشت. بنابراین کلاسه بند ساده ای که برای این ویژگی در نظر می گیریم به صورت زیر تعریف می شود:

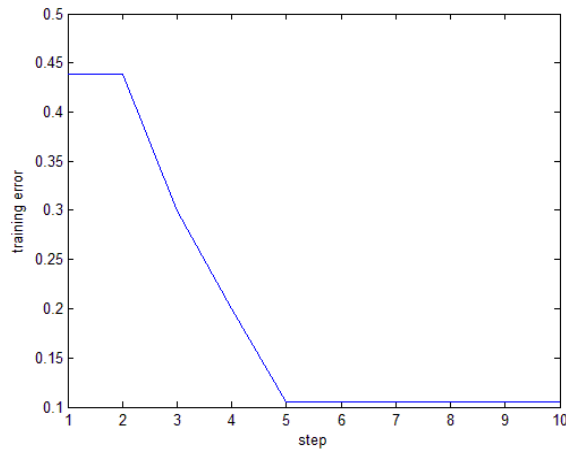
$$h_1(x) = \begin{cases} 1 & \text{if } f_1(x) > \theta_1 \\ 0 & \text{other wise} \end{cases} \quad (3-13)$$

در این رابطه منظور از f_1 همان $F_{intensity_1}$ می باشد و θ_1 آستانه متناظر با این ویژگی بوده که طبق شکل (۳-۶) برابر با ۱۸ در نظر گرفته می شود. کلاسه بند تعریف شده در رابطه (۳-۱۳) بدین صورت عمل می کند که اگر در تصویر ورودی x ، مقدار $F_{intensity_1}$ بزرگتر از آستانه θ_1 باشد تصویر مورد نظر برچسب ۱ (کلاس خوب) و در غیر این صورت برچسب ۰ (کلاس بد) خواهد گرفت.

حال کلاسه بند را توسط الگوریتم بیان شده در شکل (۳-۵) آموزش می دهیم. برای ارزیابی عملکرد کلاسه بند، خطای آموزشی ε_{tr} که توسط مجموعه آموزشی ایجاد می شود به صورت زیر تعریف می کنیم:

$$\varepsilon_{tr} = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} |h_f(x_i^{(tr)}) - y^{(tr)}(x_i^{(tr)})| \quad (3-14)$$

در رابطه فوق h_f کلاسه بند نهایی، $x_i^{(tr)}$ i امین نمونه آموزشی و $y^{(tr)}(x_i^{(tr)})$ برچسب متناظر با این نمونه می باشد. N_{tr} تعداد نمونه های آموزشی مثبت و منفی می باشد که در این تحقیق ۱۵۰۰ نمونه در نظر گرفته شده است. خطای آموزش کلاسه بند برای ده تکرار الگوریتم ($T = 10$) در شکل (۳-۷) نشان داده شده است.



شکل (۷-۳): خطای آموزش برای کلاسه بندی که با ۸۷ تصویر با ناحیه بندی درست و ۱۴۱۳ تصویر با ناحیه بندی نادرست آموزش دیده است.

همان طور که در شکل فوق ملاحظه می کنید. در اولین تکرارهای الگوریتم، خطای مرحله آموزش زیاد بوده و در تکرارهای بعدی خطا کاهش می یابد.

۳-۴-۲ نتایج ناحیه بندی

در این بخش می خواهیم کلاسه بند h_f که با استفاده از ۱۵۰۰ نمونه آموزشی با شش ویژگی و طی ده مرحله آموزش دیده شده را توسط ۴۰ تصویر بیازمائیم. ابتدا هر یک از این تصاویر را مشابه با مرحله آموزش به ازای ۲۰ مقدار λ در بازه ۰ تا ۱ (با گام ۰/۰۵) و ۱۴ مقدار σ در بازه ۰ تا ۱۰ (با گام ۰/۷) توسط الگوریتم گراف کات ناحیه بندی می کنیم. سپس تصاویر ناحیه بندی شده که مجموعاً ۱۱۲۰۰ (۴۰×۲۸۰) تصویر می باشند برای آزمایش به کلاسه بند اعمال می کنیم. برای ارزیابی نتایج حاصله خطای کلاسه بندی ϵ_A را به صورت زیر تعریف می کنیم:

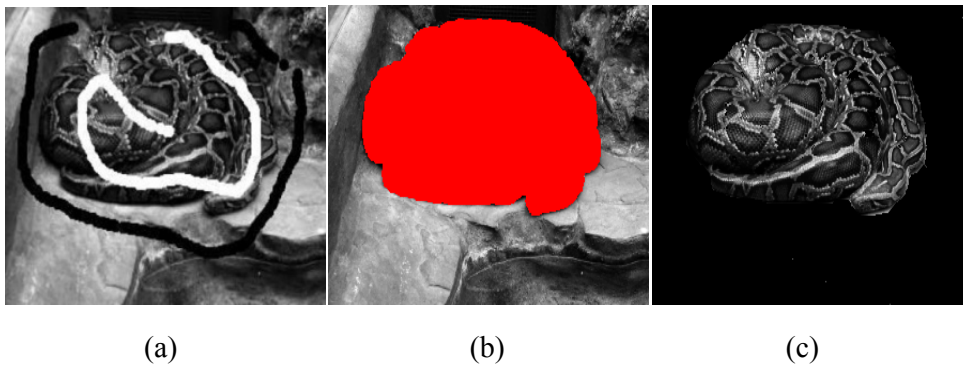
$$\epsilon_A = \frac{1}{N_{te}} \sum_{i=1}^{N_{te}} |h_f(x_i^{(te)}) - y^{(te)}(x_i^{(te)})| \quad (۱۵-۳)$$

در رابطه (۳-۱۵) N_{te} تعداد کل تصاویر آزمایشی، $x_i^{(te)}$ ، امین نمونه آزمایشی و $y^{(te)}(x_i^{(te)})$ برچسب متناظر با این نمونه بوده که توسط کاربر تشخیص داده می شود.

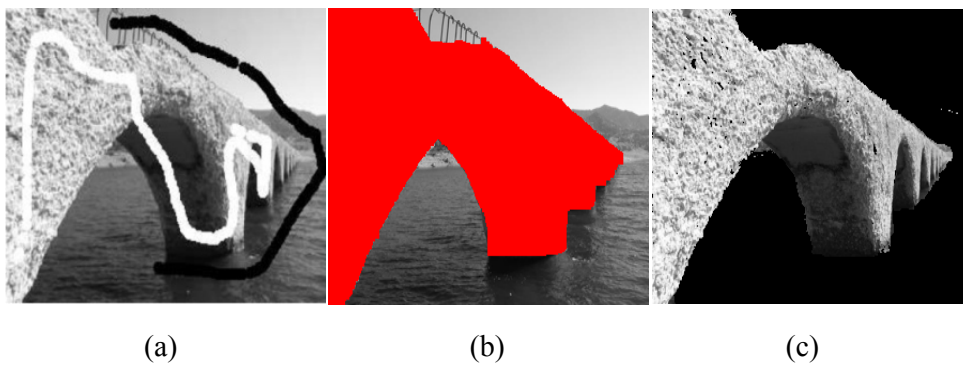
حال خطای هر یک از ۴۰ تصویر را طبق رابطه (۳-۱۵) محاسبه کرده و با میانگین گیری روی این خطاها، خطای کل کلاسه بند Adaboost را محاسبه می کنیم. خطای کل محاسبه شده برای این کلاسه بند با توجه به رابطه فوق $\epsilon_A = 0.17$ می باشد.

برای ارزیابی بهتر روش پیشنهادی چند نمونه از تصاویر ناحیه بندی شده مورد آزمون را در شکل های (۳-۸)، (۳-۹) و (۳-۱۰) به نمایش گذاشته ایم. در قسمت (c) هر یک از این اشکال، تصویر ناحیه بندی شده شی مورد نظر که به صورت اتوماتیک کلاسه بندی شده است نمایش داده می شود. سپس میزان خطای مربوط به ناحیه بندی شی را با استفاده از رابطه (۳-۱۵) محاسبه نموده ایم.

برای محاسبه خطای ناحیه بندی هر تصویر (ϵ_C) تفاضل تصویر ناحیه بندی شده انتخاب شده توسط کلاسه بند و ناحیه بندی دستی (ground truth) را به دست آورده و برای نرمالیزه کردن آن مقدار حاصله را بر تعداد کل پیکسل های تصویر تقسیم می کنیم. در شکل های زیر مقدار خطاهای ϵ_A و ϵ_C برای هر یک از تصاویر نشان داده شده است.

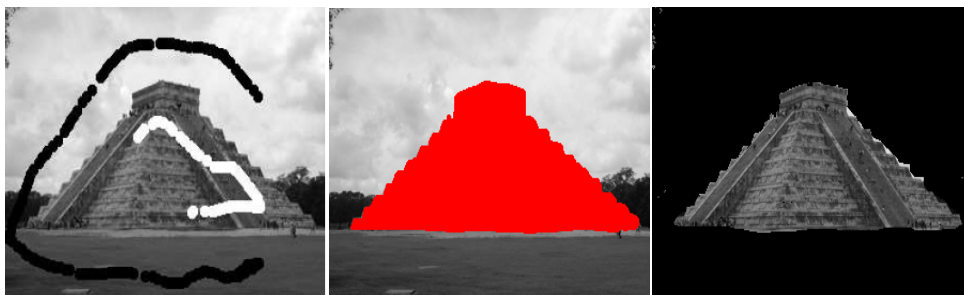


$$\varepsilon_A = 0, \quad \varepsilon_C = 0.243$$



$$\varepsilon_A = 0.036, \quad \varepsilon_C = 0.19$$

شکل (۳-۸): (a) تصویر اصلی به همراه بذرها (بذرهای مربوط به شی با رنگ سفید و بذرهای مربوط به پس زمینه با رنگ مشکی نشان داده شده است)، (b) ground truth هر تصویر، (c) ناحیه بندی بهینه هر تصویر



(a)

(b)

(c)

$$\varepsilon_A = 0.21, \quad \varepsilon_C = 0.11$$



(a)

(b)

(c)

$$\varepsilon_A = 0.07, \quad \varepsilon_C = 0.195$$

شکل (۳-۹): (a) تصویر اصلی به همراه بذرها (بذره‌های مربوط به شی با رنگ سفید و بذره‌های مربوط به پس زمینه با رنگ مشکی نشان داده شده است)، (b) ground truth هر تصویر، (c) ناحیه بندی بهینه هر تصویر



(a)

(b)

(c)

$$\varepsilon_A = 0, \quad \varepsilon_C = 0.127$$



(a)

(b)

(c)

$$\varepsilon_A = 0.082, \quad \varepsilon_C = 0.214$$

شکل (۳-۱۰): (a) تصویر اصلی به همراه بذرها (بذره‌های مربوط به شی با رنگ سفید و بذره‌های مربوط به پس زمینه با رنگ مشکی نشان داده شده است)، (b) ground truth هر تصویر، (c) ناحیه بندی بهینه هر تصویر

۳-۵ نتیجه گیری

در این فصل روشی جهت انتخاب خودکار نتیجه ناحیه بندی مبتنی بر کلاسه بند Adaboost ارائه نمودیم. ورودی های این کلاسه بند مجموعه تصاویر ناحیه بندی شده آموزشی و کلاسه بندهای ساده می باشند. کلاسه بندهای ساده به کمک شش ویژگی یک ناحیه بندی مطلوب که همگن بودن نواحی و خواص مرزی را بررسی می کنند، طراحی شده اند. برای استفاده از کلاسه بند Adaboost، یک مرحله آموزش با تعداد قابل توجهی نمونه آموزشی نیاز است. کلاسه بند را با استفاده از الگوریتم Adaboost و توسط ۱۵۰۰ نمونه آموزش دادیم. پس از آموزش، کلاسه بند قادر است هر تصویر ناحیه بندی شده را در یکی از دو کلاس خوب یا بد قرار دهد. آزمایشات برای تصاویر مختلف که توسط الگوریتم گراف کات به ازای λ و σ های متفاوت ناحیه بندی شده اند انجام شد. کلاسه بند طراحی شده، بهترین جواب های ناحیه بندی را برای هر تصویر نتیجه می دهد. خطای کلاسه بندی و ناحیه بندی قابلیت روش پیشنهادی را نشان می دهند.

فصل چهارم

کلاس بندی خودکار تصاویر ناحیه بندی شده

توسط شبکه عصبی

۴-۱ مقدمه

در این فصل یک کلاسه بند شبکه عصبی را جهت ارزیابی خودکار تصاویر ناحیه بندی شده پیشنهاد می کنیم. سپس بهترین ناحیه بندی را برای یک تصویر به صورت اتوماتیک به دست می آوریم. ویژگی های استخراج شده از تصاویر ناحیه بندی همان شش ویژگی مطرح شده در فصل سه می باشند.

همان طور که نتایج نشان می دهد پس از آموزش، کلاسه بند قادر است تا از میان ناحیه بندی های به دست آمده از تصویر ورودی که به ازای پارامترهای مختلف حاصل شده اند، ناحیه بندی مطلوب را انتخاب نماید. در ادامه ابتدا کلاسه بند شبکه عصبی معرفی می گردد، سپس به روش پیشنهادی به ارزیابی ناحیه بندی تصاویر خواهیم پرداخت.

یک شبکه عصبی ساختگی^۱ (ANN) که معمولاً به عنوان شبکه عصبی از آن یاد می شود تقلیدی از یک شبکه زیست شناختی مثل مغز انسان می باشد. یک شبکه عصبی از مجموعه ای از نورون های مصنوعی متصل به هم تشکیل شده است که قدرت یادگیری دارند. یک شبکه پس از یادگیری قادر به پردازش اطلاعات ورودی می باشد. از کاربردهای مهم شبکه های عصبی می توان به دو کاربرد کلاسه بندی الگوها و پیاده سازی فیلترهای تطبیقی غیر خطی اشاره کرد. از مزایای شبکه عصبی در مقایسه با سایر کلاسه بندها قدرت آن در ایجاد مرزهای غیر خطی و پیچیده بودن کلاس ها می باشد. البته برای ایجاد یک کلاسه بند موفق، شبکه عصبی نیاز به یک مرحله آموزش با تعداد قابل توجهی نمونه های آموزشی دارد [۳۶].

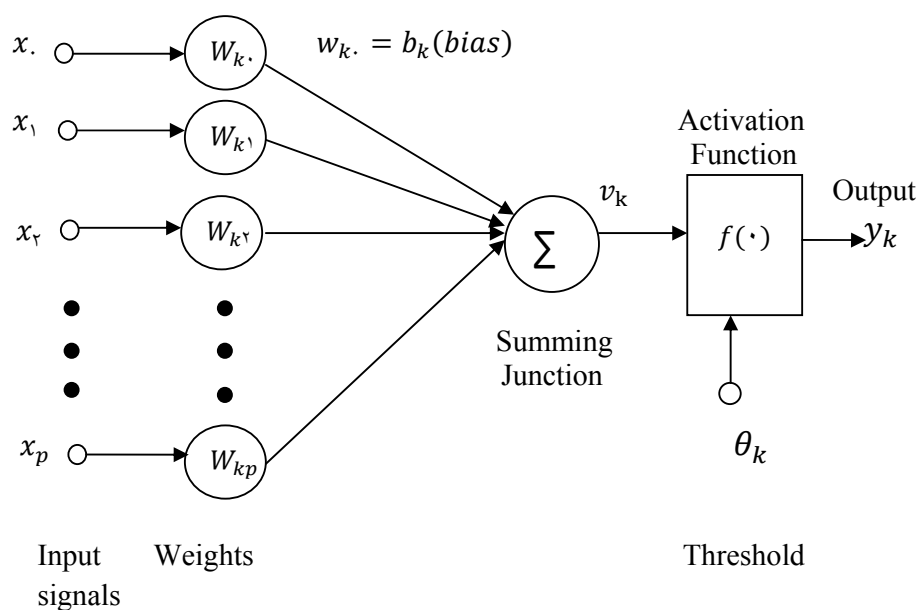
^۱ Artificial Neural Network

۲-۴ ساختار نورون

اولین نورون مصنوعی در سال ۱۹۴۳ توسط Warren McCulloch و Walter Pits ایجاد شد. بعد از معرفی نورون ها، شبکه های عصبی پدیدار شدند. این نورون ها به صورت مدل هایی از نورون های زیستی معرفی شده و اجزاء ادراکی برای سیستم های محاسباتی محسوب می شوند [۳۶][۳۷].

یک نورون در شبکه عصبی مصنوعی دارای ساختاری مشابه شکل (۱-۴) می باشد و توسط سه عنصر اساسی شکل می گیرد. اولین عنصر وزن های اتصال دهنده نورون به ورودی ها می باشند. دو عنصر دیگر تشکیل دهنده یک نورون، جمع کننده و تابع فعال ساز می باشند. ابتدا ورودی های نورون در وزن های متناظر هر ورودی ضرب شده سپس توسط واحد جمع کننده جمع می شوند. در نهایت خروجی نورون پس از عبور سیگنال مجموع از تابع فعال ساز حاصل می گردد. توابع فعال ساز، خروجی هایی در محدوده ۰ تا ۱ و یا -۱ تا ۱ ایجاد می کنند [۳۶].

Fixed input $x_i = \pm 1$



شکل (۱-۴): مدل مصنوعی یک نورون [۳۶].

با توجه به مدل نورون می توان فعالیت داخلی آن را به صورت زیر در نظر گرفت:

$$v_k = \sum_{j=1}^p W_{jk} X_j \quad (1-4)$$

در این رابطه تعداد ورودی های نورون $p + 1$ می باشند و هر ورودی توسط وزن خاصی به نورون متصل می شود. با توجه به شکل (1-4) خروجی نورون y_k می باشد که با استفاده از تابع فعال ساز تعیین می گردد.

۴-۲-۱ توابع فعال ساز

توابع فعال ساز $f(.)$ متفاوتی توسط محققین پیشنهاد گردیده است که سه نوع مهم آن عبارتند از :

- تابع آستانه گذار^۱
- تابع تکه خطی^۲
- تابع سیگموئید^۳

تابع فعال ساز آستانه گذار، ورودی داده شده را با سطح آستانه مشخصی مقایسه نموده و یکی از مقادیر ۰ یا ۱ را در خروجی ظاهر می کند [۳۶].

$$f(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \quad (2-4)$$

دومین تابع فعال ساز، تابع تکه خطی است که خروجی تابع در حوالی صفر به صورت خطی و خارج آن مقادیر ۰ یا ۱ را می پذیرد [۳۶].

$$f(v) = \begin{cases} 1 & v \geq 1/2 \\ v & -1/2 > v > 1/2 \\ 0 & v \leq -1/2 \end{cases} \quad (3-4)$$

^۱ Threshold Function

^۲ Piecewise-Linear Function

^۳ sigmoid Function

تابع فعال ساز سیگموئید تابع یکنوای ملایمی است که خروجی این تابع با تغییر ورودی از منفی بی نهایت تا مثبت بی نهایت مقادیر ۰ تا ۱ را می پذیرد. اما گاهی اوقات خروجی این تابع در محدوده ۱- تا ۱ نیز تنظیم می گردد. تابع تانژانت هایپربولیک نمونه ای از یک تابع سیگموئید می باشد که به صورت زیر تعریف می شود [۳۶]:

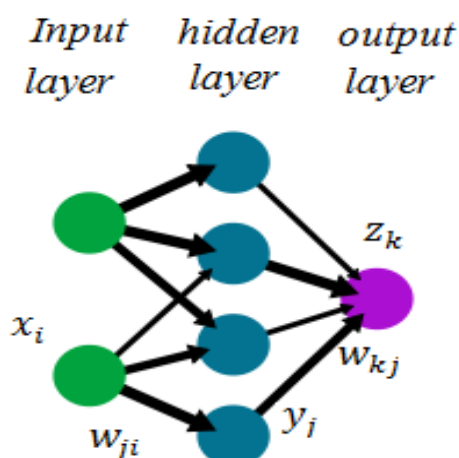
$$\tanh\left(\frac{v}{\tau}\right) = \frac{1 - \exp(-v)}{1 + \exp(-v)} \quad (4-4)$$

ویژگی این تابع فعال ساز که در دو فعال ساز قبلی وجود ندارد مشتق پذیری آن می باشد.

۳-۴ ساختار شبکه عصبی

در ساختار یک شبکه عصبی، سه نوع لایه وجود دارد که به ترتیب عبارتند از لایه ورودی، لایه مخفی و لایه خروجی. هر یک از لایه ها از گره هایی تشکیل شده است. گره های لایه ورودی، ورودی های شبکه را در بر دارند در حالی که گره های لایه های مخفی و لایه خروجی نورون های مصنوعی می باشند. در شکل (۴-۲) یک شبکه عصبی ساده نشان داده شده است. همان طور که ملاحظه می کنید در این شبکه فقط یک لایه مخفی با ۴ گره وجود دارد اما شبکه هایی با چند لایه مخفی نیز موجود هستند که در این شبکه ها نیز اتصالات مشابه با شبکه های ساده می باشد. در بعضی شبکه ها یک گره بایاس نیز در لایه ورودی وجود دارد که به تمام گره ها غیر از گره های ورودی متصل می شود [۳۸].

در یک شبکه عصبی، لایه ها توسط اتصالات وزن داری به یکدیگر متصل می شوند. در کاربرد شبکه عصبی به عنوان کلاسه بند، گره های ورودی مولفه هایی از بردار ویژگی را نشان می دهند. بدین ترتیب که ابتدا بردار ویژگی به لایه ورودی اعمال می شود. سپس هر گره مخفی برای محاسبه ورودی فعال ساز خود، مجموع وزن دار ورودی هایش را محاسبه می کند. ورودی فعال ساز گره مخفی z_{am}



شکل (۲-۴) : شبکه عصبی ساده با سه لایه [۳۹].

را با net_j نمایش داده و به صورت زیر محاسبه می کنیم [۳۸]:

$$net_j = \sum_{i=1}^d x_i w_{ji} \equiv w_j^t x \quad (۵-۴)$$

در این رابطه d تعداد گره های لایه ورودی و زیرنویس های i و j به ترتیب اندیس گره های لایه ورودی و لایه مخفی می باشند. w_{ji} وزن های لایه ورودی به مخفی را در گره مخفی j ام نشان می دهد. از این پس به جای واژه ورودی فعال ساز از لفظ نت استفاده می کنیم. پس نت هر گره مخفی را می توان از ضرب داخلی ورودی ها در وزن های لایه ورودی به مخفی به دست آورد. خروجی هر گره مخفی به صورت تابع غیر خطی از نت مربوطه اش تعریف می شود $f(net)$ ، به طوری که

$$y_j = f(net_j) \quad (۶-۴)$$

در این رابطه y_j خروجی گره مخفی j ام و $f(.)$ یک تابع فعال ساز است که می تواند یکی از توابع فعال سازی که در بخش قبل بیان کردیم انتخاب گردد [۳۸]. هر گره خروجی نیز به طریق مشابهی نت فعال ساز خود را بر اساس وزن های لایه مخفی محاسبه می کند، به طوری که :

$$net_k = \sum_{j=1}^{n_H} y_j w_{kj} \equiv w_k^t y \quad (۷-۴)$$

در این رابطه زیرنویس k اندیس گره ها را در لایه خروجی و n_H تعداد گره های لایه مخفی را نشان می دهد. هر گره خروجی نیز تابع غیرخطی از نت مربوط به خود را به صورت زیر محاسبه می کند.

$$z_k = f(net_k) \quad (۸-۴)$$

در این رابطه z_k خروجی گره k ام و $f(.)$ نیز یک تابع فعال ساز است که می تواند یکی از توابع فعال ساز در نظر گرفته شود [۳۸]. لزومی ندارد توابع فعال ساز گره های مخفی و گره های خروجی از یک نوع باشند.

۴-۴ توپولوژی شبکه عصبی

در این بخش نوع اتصالات بین لایه ها و نحوه پخش داده ها مورد توجه قرار می گیرد. شبکه ها را از نظر اتصالات بین لایه ای می توان به دو دسته ذیل تقسیم بندی کرد [۳۶]:

- **شبکه های عصبی مستقیم^۱**: این شبکه ها اولین و ساده ترین نوع شبکه های عصبی محسوب می شوند. داده ها از گره های ورودی توسط اتصالات مستقیم به گره های مخفی و سپس به گره های خروجی ارسال می شوند. هیچ اتصال فیدبکی بین خروجی گره ها و ورودی آن ها وجود ندارد. Perceptron و Adaline نمونه هایی از شبکه عصبی مستقیم می باشند [۳۶][۴۰].
- **شبکه های عصبی بازگشتی^۲**: در این شبکه ها اتصالات فیدبک بین گره ها وجود دارد. برخلاف شبکه های مستقیم، رفتار دینامیکی شبکه حائز اهمیت است. در بعضی موارد نت گره ها دستخوش تغییراتی می شود که شبکه باید شرایط پایداری را فراهم کند تا نت ها بدون تغییر باقی بمانند. در بعضی کاربردها تغییر نت نوروں های خروجی اهمیت دارد به طوری که

^۱ Feed-forward neural network
^۲ Recurrent neural network

رفتار دینامیکی، خروجی شبکه عصبی را تشکیل می دهد [۳۶]. شبکه های بازگشتی ساده (SRN) و شبکه های Hopfield نمونه هایی از شبکه های عصبی بازگشتی هستند [۳۶] [۴۰].

۴-۵ آموزش شبکه عصبی

همان طور که اشاره شد یک شبکه عصبی از سه لایه مختلف تشکیل شده است. گره های هر لایه توسط اتصالات وزن داری به گره های لایه بعدی متصل می شوند. وزن اتصالات شبکه عصبی توسط یک فرآیند آموزش تعیین می گردد. قواعد یادگیری شبکه های عصبی را می توان به سه دسته تقسیم بندی کرد:

- یادگیری با سرپرست^۱ یا یادگیری شرکت پذیر^۲: در این روش مجموعه ای از نمونه های آموزشی (x, y) که $x \in X$ یکی از نمونه های آموزشی و $y \in Y$ خروجی متناظر با این نمونه می باشد برای آموزش شبکه عصبی مورد استفاده قرار می گیرد. هدف از آموزش، یافتن تابع $f: X \rightarrow Y$ است که با اعمال ورودی های آموزشی، خروجی های متناظر با آن ها را در خروجی شبکه ایجاد کند. برای یافتن این تابع، تابع هزینه ای به صورت میانگین مربع خطا در نظر گرفته می شود. این تابع هزینه، میانگین مربع خطای بین خروجی شبکه، $f(x)$ و خروجی مطلوب y را روی تمام نمونه های آموزشی به حداقل می رساند. برای آموزش شبکه های عصبی به این روش می توان از الگوریتم انتشار به عقب^۳ استفاده کرد که در ادامه به معرفی آن می پردازیم [۴۰].
- یادگیری بدون سرپرست^۴ یا خودسازمان^۵: در این روش داده x به ورودی شبکه عصبی اعمال می شود، اما برخلاف روش یادگیری با سرپرست کلاس داده ورودی مشخص نشده است. هر تابعی بر اساس ورودی x و خروجی شبکه $f(x)$ می تواند به عنوان تابع هزینه در نظر گرفته شود که

^۱ Supervised learning

^۲ Associative learning

^۳ Backpropagation algorithm

^۴ Unsupervised learning

^۵ Self-organisation

شبکه با مینیمم کردن این تابع هزینه وزن های اتصالات را تعیین می کند. از این روش یادگیری می توان در کلاسترینگ، فیلترینگ و تخمین توزیع های استاتیکی استفاده کرد [۴۰].

- یادگیری مجدد^۱: در این روش ماشین یادگیری عملیاتی را روی محیط انجام داده و پاسخ های فیدبکی را دریافت می کند. سپس سیستم یادگیری با توجه به پاسخ های دریافتی از محیط، عملیات را به صورت خوب و بد تقسیم بندی کرده و پارامترهایش را تنظیم می کند [۳۶]. در کاربرد شبکه های عصبی در مسائل کنترل و بازی ها از این روش یادگیری استفاده می شود [۴۰].

۴-۵-۱ آموزش شبکه عصبی با الگوریتم انتشار به عقب [۳۸]:

الگوریتم انتشار یکی از ساده ترین و کلی ترین روش های آموزش با سرپرست برای شبکه های عصبی چند لایه محسوب می شود. برای آموزش شبکه ابتدا نمونه های آموزشی به لایه ورودی اعمال شده سپس مقدار خروجی در لایه خروجی تعیین می گردد. مقادیر خروجی با مقادیر مطلوب شبکه مقایسه شده و خطایی را ایجاد می کند. این خطا (تابع معیار) تابعی از وزن های شبکه می باشد. زمانی که خروجی های واقعی شبکه با خروجی های مطلوب برابر باشند، مقدار خطا به حداقل می رسد، از این رو وزن ها طوری تغییر می کنند که مقدار خطا را کاهش دهند. مقدار خطا برای این الگوریتم به صورت زیر تعریف می شود:

$$J(w) = \frac{1}{2} \sum_{k=1}^c (t_k - z_k)^2 = \frac{1}{2} \|t - z\|^2 \quad (4-9)$$

در رابطه فوق t و z به ترتیب بردارهای خروجی مطلوب و خروجی واقعی شبکه هستند. c تعداد گره های لایه خروجی و w تمام وزن ها را در شبکه نمایش می دهد.

^۱ Reinforcement learning

در شروع الگوریتم وزن ها مقادیر تصادفی انتخاب می شوند، سپس در جهتی تغییر می کنند که مقدار خطا کاهش یابد. در رابطه ذیل تغییرات وزن ها بر اساس کاهش گرادیان تابع خطا ملاحظه می گردد:

$$\Delta w = -\eta \frac{\partial J}{\partial w} \quad (10-4)$$

در این رابطه ضریب η نرخ یادگیری است که میزان تغییر وزن ها را کنترل می کند. رابطه فوق را می توان به فرم کلی زیر نیز بیان کرد:

$$\Delta w_{pq} = -\eta \frac{\partial J}{\partial w_{pq}} \quad (11-4)$$

الگوریتم انتشار در هر تکرار وزن ها را تغییر داده و وزن ها در تکرار $m+1$ به صورت زیر تعریف می شوند:

$$w(m+1) = w(m) + \Delta w(m) \quad (12-4)$$

در یک شبکه عصبی مقدار خطا به طور مستقیم به w_{jk} وابسته نیست، بنابراین از قاعده زنجیری زیر استفاده می کنیم:

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial net_k} \frac{\partial net_k}{\partial w_{kj}} = -\delta_k \frac{\partial net_k}{\partial w_{kj}} \quad (13-4)$$

δ_k در رابطه (13-4) حساسیت گره k ام را نشان داده که به صورت زیر تعریف می شود:

$$\delta_k = -\frac{\partial J}{\partial net_k} \quad (14-4)$$

حال می خواهیم تغییرات خطا را نسبت به نت فعال ساز گره ها به دست آوریم. با فرض این که تابع فعال ساز $f(.)$ مشتق پذیر باشد از معادله (9-4) مشتق گرفته و δ_k را به صورت زیر بیان می کنیم:

$$\delta_k = -\frac{\partial J}{\partial net_k} = -\frac{\partial J}{\partial z_k} \frac{\partial z_k}{\partial net_k} = (t_k - z_k) f'(net_k) \quad (15-4)$$

اگر از رابطه (۷-۴) مشتق بگیریم مقدار $y_j = \frac{\partial net_k}{\partial w_{kj}}$ می شود.

با استفاده از روابط فوق می توان قاعده یادگیری را برای وزن های لایه مخفی به خروجی به صورت زیر به دست آورد:

$$\Delta w_{kj} = \eta \delta_k y_j = \eta (t_k - z_k) f'(net_k) y_j \quad (۱۶-۴)$$

قاعده یادگیری برای وزن های لایه ورودی به مخفی نیز به همین صورت به دست می آید. با استفاده از معادله (۱۱-۴) و قاعده زنجیری می توان تغییرات خطا را نسبت به وزن های ورودی به مخفی به صورت زیر بیان کرد:

$$\frac{\partial J}{\partial w_{ji}} = \frac{\partial J}{\partial y_j} \frac{\partial y_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} \quad (۱۷-۴)$$

در رابطه فوق اولین عبارت سمت راست را به شکل زیر ساده می کنیم:

$$\begin{aligned} \frac{\partial J}{\partial y_j} &= \frac{\partial}{\partial y_j} \left[\sum_{k=1}^c (t_k - z_k)^2 \right] = - \sum_{k=1}^c (t_k - z_k) \frac{\partial z_k}{\partial y_j} \\ &= - \sum_{k=1}^c (t_k - z_k) \frac{\partial z_k}{\partial net_k} \frac{\partial net_k}{\partial y_j} = - \sum_{k=1}^c (t_k - z_k) f'(net_k) w_{kj} \end{aligned} \quad (۱۸-۴)$$

مشابه با رابطه (۱۵-۴) می توان حساسیت هر گره مخفی را به صورت زیر تعریف کرد:

$$\delta_j = f'(net_j) \sum_{k=1}^c w_{kj} \delta_k \quad (۱۹-۴)$$

طبق رابطه فوق برای به دست آوردن حساسیت هر گره مخفی می بایست مجموع حساسیت هر یک از گره های خروجی را که توسط وزن گره های خروجی با گره مخفی متناظر وزن دار شده محاسبه نمود، سپس مقدار حاصله را در مشتق تابع فعال ساز گره مخفی متناظر ضرب کرد. از این رو قاعده یادگیری برای وزن های ورودی به مخفی برابر می شود با:

$$\Delta w_{ji} = \eta x_i \delta_j = \eta \underbrace{\left[\sum_{k=1}^{\ell} w_{kj} \delta_k \right]}_{\delta_j} f'(net_j) x_i \quad (20-4)$$

معادلات (۴-۱۶) و (۴-۲۰) الگوریتم انتشار یا الگوریتم انتشار خطا را توصیف می کنند. این الگوریتم به این دلیل الگوریتم انتشار خطا نامیده می شود که در طی مرحله آموزش خطا باید از لایه خروجی به لایه مخفی منتشر شود تا وزن های لایه ورودی به مخفی آموزش ببینند.

۴-۶ کلاسه بند شبکه عصبی برای ارزیابی خودکار تصاویر ناحیه

بندی شده

در این بخش می خواهیم تصاویر ناحیه بندی شده توسط الگوریتم گراف کات که به ازای λ و σ های مختلف به دست آمده اند را توسط کلاسه بند شبکه عصبی ارزیابی کنیم. برای این منظور ابتدا ساختار شبکه عصبی مورد استفاده را معرفی نموده، سپس کلاسه بند را آموزش داده و به ارزیابی تصاویر ناحیه بندی شده توسط این کلاسه بند می پردازیم.

همان طور که در تئوری شبکه عصبی بیان شد در ساختار این شبکه سه لایه مختلف وجود دارد. از آن جایی که در این تحقیق شش ویژگی از تصاویر ناحیه بندی شده استخراج می کنیم تعداد گره های لایه ورودی را هفت در نظر می گیریم که یک گره به عنوان بایاس افزوده شده است. در شبکه پیشنهادی، لایه مخفی شامل ۱۶ گره می باشد که برای به دست آوردن تعداد گره ها در این لایه از معیار ارائه شده در مرجع [۴۱] استفاده می کنیم :

$$n_H \cdot n_d + n_H \cdot c = \frac{\text{تعداد نمونه های آموزشی}}{10} \quad (21-4)$$

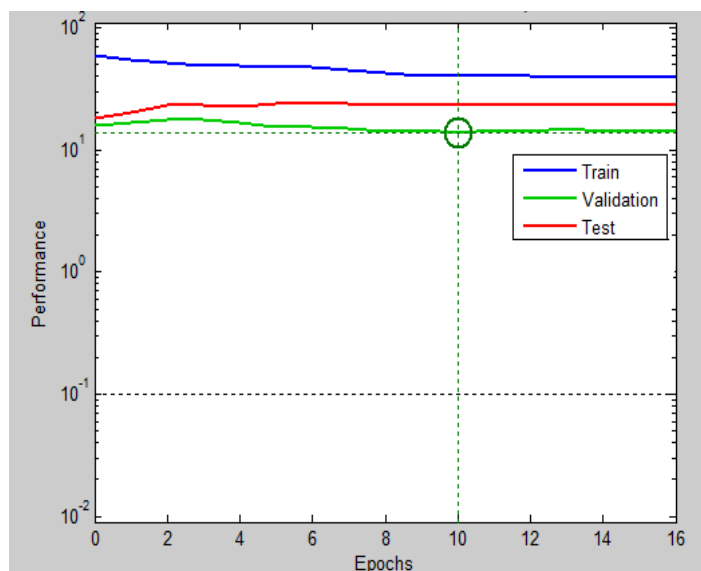
در این رابطه n_H ، n_d و c تعداد گره ها را به ترتیب در لایه ورودی ، لایه مخفی و لایه خروجی

نشان می دهند. از آن جایی که در این شبکه می خواهیم تصاویر را به دو دسته تصویر با ناحیه بندی های درست و نادرست تفکیک کنیم تعداد گره های لایه خروجی را ۲ در نظر می گیریم. برای آموزش این شبکه نیز از همان مجموعه تصاویر آموزشی تعریف شده در فصل سوم استفاده نموده ایم. همان طور که اشاره شد تعداد نمونه های آموزشی ۱۵۰۰ تصویر می باشد که به این ترتیب طبق رابطه فوق تعداد گره ها در لایه مخفی برابر با ۱۶ خواهد شد. در شبکه عصبی پیشنهادی، توابع فعال ساز گره های لایه مخفی را سیگموئید و توابع فعال ساز گره های لایه خروجی را خطی انتخاب کرده ایم.

۴-۶-۱ آموزش کلاسه بند

همان طور که در تئوری شبکه عصبی بیان شد برای آموزش این شبکه سه روش مختلف وجود دارد که عبارتند از یادگیری با سرپرست، یادگیری بدون سرپرست و یادگیری مجدد. ما در این تحقیق برای آموزش شبکه عصبی از روش یادگیری با سرپرست به وسیله الگوریتم انتشار به عقب استفاده می کنیم. همان طور که قبلاً نیز اشاره شد برای آموزش شبکه به این روش به یک مجموعه آموزشی نیاز داریم. مجموعه آموزشی که ما برای آموزش شبکه عصبی استفاده می کنیم همان مجموعه ای است که برای آموزش کلاسه بند Adaboost نیز در نظر گرفته شد. جهت یادآوری مجموعه آموزشی مورد استفاده شامل ۱۵۰۰ تصویر می باشد که از این میان ۸۷ تصویر، نمونه مثبت و ۱۴۱۳ تصویر، نمونه منفی هستند. کلاسه بند را با استفاده از این مجموعه آموزشی و طی ۱۶ مرحله تکرار آموزش دادیم. در شکل (۳-۴) خطای مربوط به مراحل آموزش، اعتبارسنجی^۱ و آزمون به ترتیب با رنگ های آبی، سبز و قرمز نشان داده شده است. همان طور که در شکل زیر ملاحظه می کنید در اولین تکرارهای الگوریتم، خطای مرحله آموزش زیاد بوده و در تکرارهای بعدی خطا کاهش می یابد و مرحله آموزش زمانی خاتمه می یابد که خطای مربوط به اعتبارسنجی به حداقل برسد. مطابق شکل (۳-۴) آموزش کلاسه بند را پس از ۱۰ مرحله تکرار خاتمه دادیم.

^۱ validation



شکل (۴-۳): خطای مراحل آموزش، اعتبارسنجی و آزمون در ۱۶ تکرار نشان داده شده است.

۴-۶-۲ نتایج ناحیه بندی

در این بخش می خواهیم کلاسه بند شبکه عصبی که با استفاده از ۱۵۰۰ نمونه آموزشی و طی ۱۰ مرحله آموزش دیده شده را توسط ۴۰ تصویر بیازمائیم. ابتدا هر یک از این تصاویر را مشابه با مرحله آموزش به ازای ۲۰ مقدار λ در بازه ۰ تا ۱ (با گام ۰/۰۵) و ۱۴ مقدار σ در بازه ۰ تا ۱۰ (با گام ۰/۷) توسط الگوریتم گراف کات ناحیه بندی می کنیم. سپس تصاویر ناحیه بندی شده که مجموعاً ۱۱۲۰۰ (۴۰×۲۸۰) تصویر می باشند برای آزمایش به کلاسه بند اعمال می کنیم. برای ارزیابی نتایج حاصله خطای کلاسه بند ε_N را به صورت زیر تعریف می کنیم:

$$\varepsilon_N = \frac{1}{N_{te}} \sum_{i=1}^{N_{te}} |h_f(x_i^{(te)}) - y^{(te)}(x_i^{(te)})| \quad (۴-۲۲)$$

در این رابطه N_{te} تعداد کل تصاویر تست، $i, x_i^{(te)}$ امین نمونه تست و $y^{(te)}(x_i^{(te)})$ برچسب متناظر با این نمونه بوده که توسط ناحیه بند تشخیص داده می شود. خطای هر یک از ۴۰ تصویر را طبق رابطه (۴-۲۲) محاسبه کرده و با میانگین گیری روی این خطاها، خطای کل کلاسه بند شبکه عصبی را به دست آورده ایم. خطای کل محاسبه شده برای این کلاسه بند با توجه به رابطه فوق $\epsilon_N = 0.0065$ می باشد.

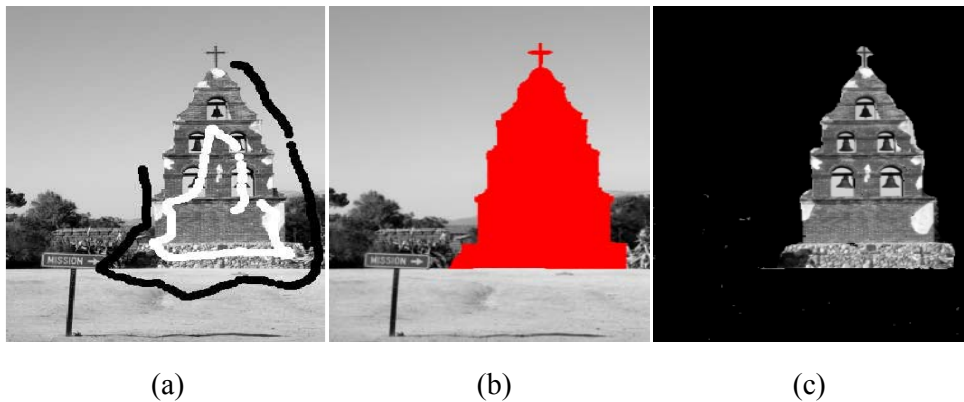
۴-۶-۳ مقایسه و تحلیل نتایج

در این بخش به ارزیابی نتایج کلاسه بندی به کمک شبکه عصبی پیشنهادی می پردازیم. هر یک از تصاویر آزمون به ازای ۲۸۰ مقدار مختلف λ و σ ناحیه بندی شده اند و ویژگی های استخراج شده از هر تصویر جهت کلاسه بندی به شبکه عصبی داده می شود. چند نمونه از تصاویر مورد آزمایش شده را در شکل های (۴-۴)، (۴-۵) و (۴-۶) ملاحظه می کنید. در قسمت (c) هر یک از این اشکال، بهترین ناحیه بندی تشخیص داده شده توسط شبکه عصبی نشان داده شده است. میزان خطای مربوط به هر تصویر را با استفاده از رابطه (۴-۲۲) به ازای $N_{te} = 280$ محاسبه نموده ایم.

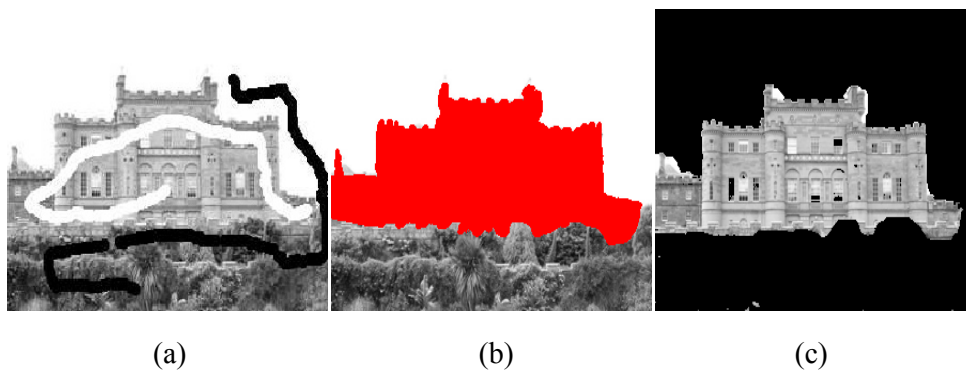
برای ارزیابی الگوریتم گراف کات خطای ناحیه بندی ϵ_C را از تفاضل تصویر ناحیه بندی انتخاب شده توسط کلاسه بند و تصویر ground truth به دست می آوریم. برای نرمالیزه کردن خطا تفاضل حاصله بر تعداد کل پیکسل های تصویر تقسیم می شود. مقدار خطاهای ϵ_C و ϵ_N برای هر یک از تصاویر نمونه نشان داده شده است.

لازم به ذکر است که در شکل های زیر در قسمت (a)، تصاویر اصلی و بذره های مربوط به هر یک از نواحی شی و پس زمینه به ترتیب با رنگ های سفید و مشکی مشخص شده است. در قسمت (b) تصویر ground truth متناظر با هر شکل را ملاحظه می کنید که در این تصویر ناحیه شی به رنگ

قرمز مشخص شده است. در قسمت (c)، ناحیه بندی انتخاب شده توسط کلاسه بند را ملاحظه می کنید. در این تصویر ناحیه پس زمینه با رنگ مشکی مشخص شده است.



$$\varepsilon_N = 0.0036, \varepsilon_C = 0.009$$



$$\varepsilon_N = 0.0107, \varepsilon_C = 0.325$$

شکل (۴-۴): (a) تصویر اصلی به همراه بذر ها (بذر های مربوط به شی با رنگ سفید و بذر های مربوط به پس زمینه با رنگ مشکی نشان داده شده است)، (b) ground truth هر تصویر، (c) ناحیه بندی بهینه هر تصویر

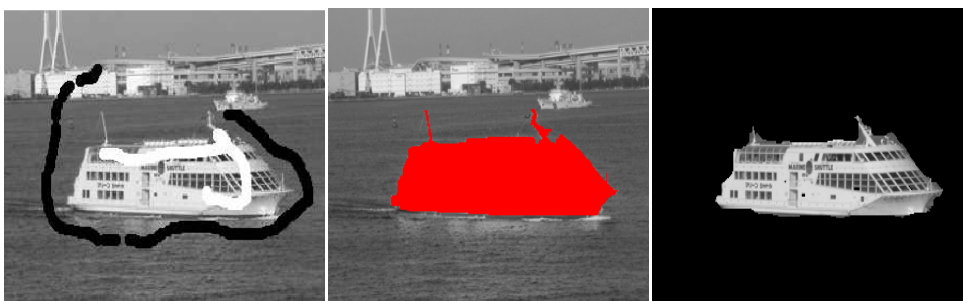


(a)

(b)

(c)

$$\varepsilon_N = 0, \quad \varepsilon_C = 0.250$$



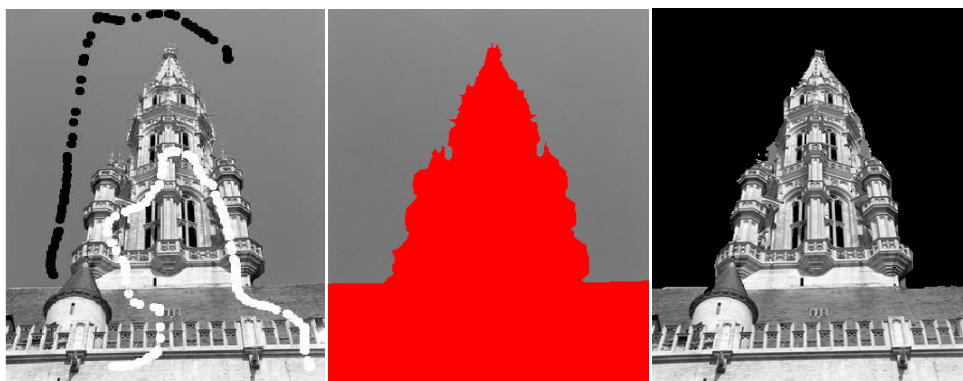
(a)

(b)

(c)

$$\varepsilon_N = 0.143, \quad \varepsilon_C = 0.148$$

شکل (۴-۵): (a) تصویر اصلی به همراه بذرها (بذره‌های مربوط به شی با رنگ سفید و بذره‌های مربوط به پس زمینه با رنگ مشکی نشان داده شده است)، (b) ground truth هر تصویر، (c) ناحیه بندی بهینه هر تصویر



(a)

(b)

(c)

$$\varepsilon_N = 0.0036, \varepsilon_C = 0.0168$$



(a)

(b)

(c)

$$\varepsilon_N = 0.0036, \varepsilon_C = 0.019$$

شکل (۴-۶): (a) تصویر اصلی به همراه بذرها (بذره‌های مربوط به شی با رنگ سفید و بذره‌های مربوط به پس زمینه با رنگ مشکی نشان داده شده است)، (b) ground truth هر تصویر، (c) ناحیه بندی پهنه هر تصویر

۷-۴ مقایسه ارزیابی های کلاسه بندهای Adaboost و شبکه عصبی

در دو فصل اخیر کلاسه بندهای Adaboost و شبکه عصبی مطرح شدند. همان طور که در تئوری این کلاسه بندها بیان شد برای استفاده از آن ها یک مرحله آموزش نیاز است. در این تحقیق ما هر دو کلاسه بند را با یک مجموعه آموزشی آموزش دادیم. سپس نتایج ناحیه بندی توسط الگوریتم گراف کات و به ازای λ و σ های مختلف را با هر دو کلاسه بند ارزیابی کرده و خطای مرحله تست هر کلاسه بند را محاسبه نمودیم. با توجه به آزمایشات انجام شده مقدار خطای متوسط کلاسه بند Adaboost $\epsilon_A = 0.017$ و خطای شبکه عصبی $\epsilon_N = 0.065$ می باشد. با مقایسه خطاهای به دست آمده به نظر می رسد که شبکه عصبی در مقایسه با Adaboost بهتر عمل می کند.

۸-۴ نتیجه گیری

در این فصل جهت ارزیابی نتایج ناحیه بندی کلاسه بند شبکه عصبی را معرفی کردیم. این کلاسه بند را مشابه با Adaboost با ۱۵۰۰ نمونه آموزش دادیم. پس از آموزش، کلاسه بند قادر است تصاویر ناحیه بندی شده که توسط الگوریتم گراف کات و به ازای λ و σ های مختلف به دست آمده را به دو کلاس خوب و بد تفکیک کند. آزمایشات برای تصاویر مختلف انجام شد. برای ارزیابی نتایج به دست آمده توسط این کلاسه بند معیار خطایی را محاسبه نمودیم. با مقایسه خطاهای به دست آمده توسط شبکه عصبی و Adaboost نتیجه گرفتیم که شبکه عصبی در مقایسه با Adaboost موفق تر می باشد.

فصل پنجم

نتیجه گیری و پیشنهادات

۵-۱ نتیجه گیری

در این پایان نامه دو روش کارآمد جهت ارزیابی خودکار نتایج ناحیه بندی مبتنی بر گراف کات ارائه شد. در این روش ها ابتدا نتایج ناحیه بندی را به وسیله الگوریتم گراف کات و به ازای پارامترهای مختلف به دست آورده ، سپس ویژگی هایی را از نتایج ناحیه بندی شده استخراج کرده و با اعمال این ویژگی ها به کلاسه بندها نتایج حاصل از ناحیه بندی را به دو کلاس خوب و بد تفکیک کردیم. برای این منظور ابتدا الگوریتم گراف کات را بررسی کرده و سپس به ارزیابی نتایج حاصله به روش های پیشنهادی پرداختیم.

در فصل سوم کلاسه بند Adaboost مورد بررسی قرار گرفت. در روش پیشنهادی برای ارزیابی نتایج ناحیه بندی توسط این کلاسه بند، دو ورودی در نظر گرفته شد که یکی از ورودی ها مجموعه آموزشی و دیگری کلاسه بندهای ساده می باشند. با توجه به این که شش ویژگی از تصاویر ناحیه بندی شده استخراج نمودیم شش کلاسه بند ساده برای طراحی Adaboost در نظر گرفتیم. با استفاده از کلاسه بندهای ساده و مجموعه آموزشی کلاسه بند را توسط الگوریتم Adaboost آموزش دادیم. پس از آموزش ، نشان دادیم کلاسه بند قادر است از میان تصاویر ناحیه بندی شده بهترین جواب ها را انتخاب کند. نتایج حاصل از این کلاسه بندی را ارزیابی نمودیم و خطای مربوط به کلاسه بندی کارآیی روش پیشنهادی را نشان می دهد.

روش دوم پیشنهادی جهت ارزیابی نتایج ناحیه بندی با استفاده از کلاسه بند شبکه عصبی می باشد. برای این منظور یک شبکه با سه لایه را به وسیله الگوریتم انتشار به عقب ، آموزش دادیم. نشان دادیم که شبکه عصبی قادر است به خوبی نتایج ناحیه بندی را به دو کلاس خوب و بد تفکیک کند. جهت ارزیابی عملکرد این کلاسه بند مشابه با Adaboost مقدار خطا را محاسبه نمودیم. مقایسه خطاهای به دست آمده از کلاسه بند شبکه عصبی در مقایسه با Adaboost نشان داد که کلاسه بند

شبکه عصبی در ارزیابی نتایج ناحیه بندی موفق تر از Adaboost عمل می کند. این می تواند به دلیل انعطاف پذیری شبکه عصبی در ایجاد مرزهای غیرخطی در تصمیم گیری باشد.

۵-۲ پیشنهادی برای ادامه تحقیق

در ادامه می توان پیشنهاداتی در جهت ادامه تحقیق انجام شده ارائه نمود.

۱. استفاده از ویژگی های دیگری در کنار شش ویژگی به کار رفته می تواند ارزیابی بهتری از ناحیه بندی را نتیجه دهد. در این راستا می توان استفاده از اطلاعات رنگ و معیارهای پیشرفته تری از بافت نواحی را پیشنهاد نمود.
۲. با سیستم طراحی شده می توان یک ناحیه بند کاملاً اتوماتیک را پیشنهاد نمود. در این راستا کاربر بذره های شی و پس زمینه را مشخص می کند سپس به کمک روش ارائه شده در این پایان نامه می توان با تغییر پارامترهای ناحیه بندی و ارزیابی اتوماتیک آن یک ناحیه بندی مطلوب را برای تصویر ورودی ایجاد کرد.

مراجع

- [١] ZHANG Y.J. (٢٠٠٦), “**ADVANCES IN Image and Video Segmentation**”, IRM Press, Tsinghua University, Beijing, China, ١.
- [٢] McGuinness K. and oconnorn N., (٢٠٠٨).” **A Comparative Evaluation of Interactive Segmentation Algorithms**”
- [٣] ZHANG Y.J. (٢٠٠٦), “**ADVANCES IN Image and Video Segmentation**”, IRM Press, Tsinghua University, Beijing, China, ٤.
- [٤] Haralick R. M. and Shapiro L.G. (١٩٨٥), “**Image segmentation techniques**”, CVGIP, ٢٩, ١٠٠-١٣٢.
- [٥] Fu K.S. and Mui, J.K. (١٩٨١), “**A survey on image segmentation**”, Pattern recognition, ١٣, ٢-١٦.
- [٦] E. N. Mortensen and W. A. Barrett, (١٩٩٨), “**Interactive segmentation with intelligent scissors**”. Graphical Models and Image Processing, ٦٠:٣٤٩-٣٨٤.
- [٧] E.R.DAVIES, (٢٠٠٦), “**Machine Vision**”, ٣RD edition, MORGAN KAUFMANN PUBLISHERS, ١٠٥.
- [٨] Matthew Marsh, “**A Literature Review of Image Segmentation Techniques and Matting for the Purpose of Implementing Grab-Cut**”.
- [٩] Adams R. and Bischof L. (١٩٩٤), “**Seeded region growing**”, IEEE Trans. on Pattern Anal. and Mach. Intell. ١٦ (٦) ٦٤١-٦٤٧.
- [١٠] M. Kass and A. Witkin, and D. Terzopoulos, (١٩٨٨), “**Snakes, active contour model**”, International Journal of Computer Vision, pp. ٣٢١-٣٣١.
- [١١] Pedro Lee C., (٢٠٠٥), **PhD. Thesis, “Robust Image Segmentation using Active Contours: Level Set Approaches**”, Dept. of Electrical and Computer Engineering North Carolina State University.
- [١٢] J. Canny, (١٩٨٦), “**A computational approach to edge detection**”, IEEE Transactions on Pattern Analysis and Machine Intelligence, no. ٨, p. ٧٦٩.
- [١٣] Y. BOYKOV and G. FUNKA-LEA, (٢٠٠٦), “**Graph Cuts and Efficient N-D Image Segmentation**”, International Journal of Computer Vision ٧٠(٢), ١٠٩-١٣١.
- [١٤] E.W. Dijkstra, (١٩٥٩), “**A Note on Two Problems in Connexion with Graphs**”, Numerische Mathematik, Vol.١, pp. ٢٦٩-٢٧٠.

- [15] E. N. Mortensen and W.A. Barrett , “**Intelligent Scissors for Image Composition**”.
- [16] Yuri Boykov and M. P. Jolly, (2001), “**Interactive graph cuts for optimal boundary and region segmentation**”, In ICCV, volume I, pages 100–112.
- [17] Bo Peng and Olga Veksler , “**Parameter Selection for Graph Cut Based Image Segmentation**”.
- [18] A. X. Falaco, J.K. Udupa, S. Samarasekara, and S. Sharma , (1998) , “**User-steered image segmentation paradigms: Live wire and live lane**”. In Graphical Models and Image Processing, volume 60, pages 233–260.
- [19] D. Greig, B. Porteous, and A. Seheult ,(1989), “**Exact maximum a posteriori estimation for binary images**”. Journal of the Royal Statistical Society, Series B, 51(2):271–279.
- [20] D. Greig, B. Porteous, and A. Seheult , (1989), “**Exact maximum a posteriori estimation for binary images**” , Journal of the Royal Statistical Society, Series B.
- [21] Y. Boykov and V. Kolmogorov ,(2004), “**An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision**”, *TPAMI*, 26(9):1124–1137.
- [22] DAVID G.STORK and RICHARD O. DUDA and PETER E. HART , (2000) , “**Pattern Classification** ” , Second Edition ,301.
- [23] J.W.Chinneck , (2000), “**chapter 9 :Maximum Flow and the MinimumCut**”.
- [24] L. Ford and D. Fulkerson , (1962), “**Flows in Networks**”. Princeton University Press.
- [25] T. Collins, (2004), “**Graph Cut Matching In Computer Vision**”.
- [26] Andrew V. Goldberg and Robert E. Tarjan , (1978), “**A new approach to the maximum-flow problem**” , Journal of the Association for Computing Machinery, 30(4):921–940.
- [27] E. A. Dinic , (1970), “**Algorithm for solution of a problem of maximum flow in networks with power estimation**” , Soviet Math. Dokl., 11:1277–1280.
- [28] Rao A. R. , Lohse G.L., (1993), “**Identifying high level features of texture perception**” , CVGIP : Graphical Models and Image Proc. 50(3) , pp.5218 – 5233.
- [29] Clausi D., Jernigan M. Ed , (2000), “**designing Gabor filters for optimal texture separability**” , pattern recognition , vol.33 , pp.1835-1849.

- [٣٠] S.E. Grigorescu, N. Petkov, and P. Kruizinga , (٢٠٠٢), “**Comparison of texture features based on gabor filters**”, *TIP*, ١١(١٠):١١٦٠–١١٦٧.
- [٣١] M. Clark, A. C. Bovik, and W. S. Geisler, (١٩٨٧), “**Texture segmentation using Gabor modulation/demodulation**”, *Pattern Recognit. Lett.*, vol. ٦, pp. ٢٦١–٢٦٧.
- [٣٢] I. Fogel and D. Sagi, (١٩٨٩) , “**Gabor filters as texture discriminator**” , *Biol. Cybern.*, vol. ٦١, pp. ١٠٣–١١٣.
- [٣٣] A. C. Bovik, (١٩٩١), “**Analysis of multichannel narrowband filters for image texture segmentation**” , *IEEE Trans. Signal Processing*, vol. ٣٩, pp. ٢٠٢٥–٢٠٤٣.
- [٣٤] J. Meynet , (٢٠٠٣) , “**Fast Face Detection Using AdaBoost**”.
- [٣٥] Yoav Freund and Robert E. Schapire , (١٩٩٧), “**A decision-theoretic generalization of on-line learning and an application to boosting**”, *J. Comput. Syst. Sci.*, ٥٥(١):١١٩–١٣٩.
- [٣٦] <http://www.learnartificialneuralnetworks.com/>.
- [٣٧] http://www.doc.ic.ac.uk/~nd/surprise_٩٦/journal/vol٤/cs١١/report.html.
- [٣٨] DAVID G.STORK and RICHARD O. DUDA and PETER E. HART , (٢٠٠٠) , “**Pattern Classification**” , Second Edition , ٢٨٤ , ٢٨٨–٢٩٢.
- [٣٩] http://en.wikipedia.org/wiki/Neural_network.
- [٤٠] http://en.wikipedia.org/wiki/Artificial_neural_network.
- [٤١] DAVID G.STORK and RICHARD O. DUDA and PETER E. HART , (٢٠٠٠) , “**Pattern Classification**” , Second Edition , ٣١٠.

Abstract

Image segmentation is one of the first steps in each machine vision system. Image segmentation problem could be surveyed in two branches: automatic image segmentation and interactive image segmentation. Here we will study graph cut based interactive image segmentation. Success of graph cut based segmentation is dependent suitable parameters selection. Usually each image has its own optimal set of parameters. In this research our goal is automatic evaluation of segmentation results. we will propose two systems using Adaboost and neural network classifiers for evaluating graph cut based segmentation results.

In each of the proposed systems ,classifiers is trained using training set. Then the trained classifiers can evaluate a segmented image. We extract 6 features from segmented images in training and testing stages. This features show homogeneous properties of segments and differences of boundary using intensity and texture. Experimental results shows an error rate of 1.7 % for Adaboost classifier and an error rate of 0.65 % for neural network classifier.

Key words: Interactive image segmentation, Graph cut algorithm, Adaboost classifier, Neural network classifier.