به نام خدا

در ابتدا کد سی موجود، به عنوان برنامه ی main می باشد . این برنامه با کد اصلی دارای توابعی دیگری هم هست که با رندر کردن یا به عبارتی فراخوانی توابع مربوطه در برنامه main می توان گفت سورس کد کامل است. توابع می توانند R ، B ، CX ، CXX و ... باشند

در ابتدا توضیحی از الگوریتم داده می شود و با توضیح داده شده می توان کد را به راحتی تحلیل نمود.

این پروژه در قطعه بندی تصاویر هوایی جهت مشخص نمودن مکان آبجکت ها در تصویر نوشته شده است.

روش های قطعه بندی تصویر استفاده از الگوریتم RG و ... است اما این الگوریتم ها به این علت که ممکن است نویز تصویر را کاهش ندهند یا نتوانند روی لبه تصویر مانور دهند، داده های دقیقی در اختیارما قرار نمی دهند لازم است روش دیگری امتحان شود. در بیس همه ی این الگوریتم ها در ابتدا باید اطلاعاتی در پیکسل های موجود در عکس به دست آورده شود و سپس روی پیکلس ها کار شود تا بتوان به بهترین قطعه بندی رسید.

مفهوم پیکسل با مفهوم طول و عرض متفاوت است. پیکسل واحد اندازه گیری تصویر است و طول و عرض واحد اندازه گیری بر فضای اشغال شده بر حسب سانتی متر و غیره. تفاوت مفهوم این دو کد مهم است.

توضیح الگوریتم :

الگوریتم به صورت کلی این شکلی است که در ابتدا تمام پیکسل های تصویر دریافتی را میگیرد به عنوان مثال تعداد N پیکسل در تصویر وجود دارد. چون وارد کردن تک تک پیکسل ها در الگوریتم کار درستی نیست باید یک شاخصی تعریف کنیم تا بتوان دیتای ورودی را تا حد امکان کاهش دهیم و سرعت تحلیل بالاتر رود. شاخص اولیه را مفهومی به نام سوپرپیکسل در نظر می گیریم. عکس را به Kقسمت با اندازه های یکسان تقسیم می کنیم و هر قسمت را که اکنون دارای اندازه N/K هست را به عنوان یک سوپر پیسکل در نظر میگیریم. با انجام اینکار میزان دیتای مورد نیاز جهت تحلیل تا حدودی کاهش پیدا می کند. از آنجا که در این حالت ممکن است لبه و edge های اشیا موجود در عکس به خوبی تحت سوپر پیکسل درنیامده باشند. دایره را کمی بزرگتر کرده و به سراغ نقاط شاخص دیگری به نام seed point می رویم. در این حالت محاسباتی seed pointرا در رابطه جذر n/k به دست می آوریم. در واقع این سید پوینت به عنوان یک نقطه ی شاخص مرکزی در همسایگی از پیکسل های همسایه در نظر گرفته می شود و بقیه سوپر پیسکل ها موقعیت و شاخصه های خود را باید در تناسب با این سید پوینت در نظر بگیرند.

پس از این مرحله کار به مرحله ی قطعه بندی تصویر می رسد. برای اینکار، ابتدا با استفاده از الگوریتم بسیار ابتدایی kmeans که بر اساس میزان فاصله کار میکند، یک بخش بندی ساده صورت می گیرد. الگوریتم kmeans بر این اساس کار می کند که میزان اختلاف داده ی موجود را با دیگر دیتا ها می

سنجد سپس آنهایی که میزان اختلاف آنان از یکدیگر در یک رنج و محدوده ی نزدیک است را در یک گروه قرار می دهد و به این صورت طبقه بندی می کند این الگوریتم پرکاربردی در پردازش تصویر است اما دقت کافی ندارد و همواره به عنوان الگوریتم پایه از آن استفاده می شود.

در اینجا، هر پیکسل دارای 5 مولفه است که سه مولفه ی l و a و b به طیف رنگی پیکسل و x و y به مختصات پیکسل مرتبط می شود. در واقع همه این 5 مولفه در یک بردار 5 المانی ظاهر می شوند. همین مولفه ها و الگوریتم kmeans اختلاف هر پیکسل با پیکسل سردار یا همان سید پوینت محاسبه می شود و بدین صورت یک بخش بندی اولیه از نواحی صورت می گیرد.

پس از انجام این بخش بندی اولیه،ممکن است برخی پیکسل ها به هیچ سید پوینتی نزدیک نباشند یا دست نخورده باقی مانده باشند پس باید برای این پیکسل ها هم راه حلی در نظر گرفته شود. برای غلبه بر این مشکل، از الگوریتم merge کردن استفاده می شود. همانطور که از نام این الگوریتم پیداست، این الگوریتم می تواند مواد با خصوصیات یکسان را به هم یکی کند و به یک واحد جداگانه تبدیل کند.

برای مرج کردن نواحی قطعه بندی شده حاصل از الگوریتم kmeans ، از الگوریتم redcap استفاده می کنیم. در این الگوریتم می توان نواحی را به یکدیگر متصل کرد و یک ناحیه واحد را تشکیل داد. در قاعده الگوی الگوریتم ردکپ به این صورت آمده است که ابتدا باید یک شاخص مینیمم درخت از داده های kmeans ساخته شود. سپس این درخت به چند زیر درخت تبدیل شود که در در زیر درخت یک سری داده ی همخوان kmeansشده وجود دارد به این صورت است که اولین مرحله از مرج کردن ناحیه ها صورت میگیرد.

اکنون نواحی مرج شده اند ، باید یک شاخصه ای را تعریف کنیم که بر اساس آن بتوانیم تعیین کنیم چه تعداد ناحیه ی بخش بندی شده برای مثال اگر این تصویر را به 20 ناحیه تقسیم کنیم اطلاعات استخراجی دقیق تر خواهند بود یا 50 قطعه هر چر تعداد بالاتر باشد نتایج بهتر است اما باید در نظر داشت که تعداد بالا هم محدودیت دارد. در غیر اینصورت، نشان دهنده ی ضعف در به کارگیری الگوریتم است و دوباره به نقطه اول بر خواهیم گشت که میلیاردها اطلاع از تمام پیکسل ها در اختیار داریم و تحلیل آنها دشوار است.

پس باید پس از تقسیم بندی نواحی، بهترین و کافی ترین تعداد بخش بندی را پیدا کنیم. برای اینکار از فرمول های واریانس محلی و میانگین واریانس محلی برای هر ناحیه ی مرج شده استفاده میکنیم. این روش طبق بررسی های صورت گرفته ، بهترین روش است و بهترین اطلاعات را در اختیار قرار می دهد. پس از اینکه مقدار واریانس محلی و میانگین واریانس محلی برای هر یک از نواحی را حساب کردیم ، دیاگرام آنها را رسم میکنیم تا متوجه شویم چه تعداد بخش بندی، برای دریافت اطلاعات مفید از عکس مناسب تر است. در جایی که نمودار هموار تر می شود، یعنی مرز بندی پیکسل ها بهتر تعیین شده و الگوریتم kmeans و توابع الگوریتم redcap در این تعداد به بعد، خروجی های بهتری در اختیار قرار داده اند. در مقادیر 80 قطعه به بعد قطعه بندی بهتری از تصویر ارائه داده است با یافتن نقطه ی متناظر در نمودار میانگین واریانس محلی ، به بهترین تعداد یعنی 128 ناحیه بخش بندی رسیده ایم.

مریم ابراهیمی نیا

آنچه که گفتم ، روش الگوریتم و پروسه است و کد متناظر با الگوریتم مذکور نوشته شده هیچ کدام از فرمول های واریانس محلی، میانگین آن ، الگوریتم فاصله یابی kmeans ، الگوریتم redcap برای مرج کردن در کد متلب وجود ندارد .

تحلیل برنامه

در این برنامه پردازش تصویر، همواره عکس ورودی به صورت یک عکس رنگی rgb است. برای تبدیل این عکس به تصویری قابل تحلیل باید یک سری اعمال یا به عبارتی نویز گیری روی عکس انجام شود. یعنی ابتدا باید عکس رنگی به یک عکس خاکستری (به عنوان مثال، فرمت های دیگری هم وجود دارد.) تبدیل شود ، سپس این عکس خاکستری تهیه شده ، آستانه گذاری شود یعنی میزان شدت روشنایی پیسکل ها تعیین شود. برای مثال اگر یک عکسی داشته باشیم که رنگ های روشن بیشتر در آن به کار رفته باشد، به هنگام ترسیم نمودار آستانه گذاری رنج پیکسل های 150 تا 250 دارای تعداد بیشتری خواهند بود.

پس از تعیین آستانه گذاری، باید عکس تبدیل به یک عکس باینری شود. یعنی حالا که شدت روشنایی عکس استخراج شد، باید کاری کنیم که پیکسل ها از لحاظ منطقی فقط دارای 0 یا 1 باشند. مثلا پیکسل هایی که شدت درجه روشنایی آنها بیشتر از 150 باشد، مقدار 1 و آنهایی که کمتر از این مقدار را دارند، مقدار 0 را داشته باشند. به این صورت تصویر را باینری می کنند.

پس از تبدیل تصویر، به حالت باینری، عملیاتی روی عکس انجام می شود. این عملیات ریخته گری نام دارد. یعنی باید shape و قالب اصلی اشیا موجود در تصویر به گونه ای استخراج شود. مورفولوژی یا ریخته گری دارای چهار حالت، چاق کردن، لاغر کردن، باز کردن و سپس بستن و بستن و سپس باز کردن است که بسته به کاربرد و نیاز از هر یک از این چهار حالت استفاده می شود. خوبی ریخته گری این است که می تواند نویز را هم از بین ببرد. در عملیات لاغر کردن، برخی اشیا از آنچه که هستند لاغر تر می شوند، در عملیات چاق کردن بالعکس و در بازکردن و سپس بستن ، تصویر ابتدا چاق می شود و سپس لاغر و برای حالت چهارم هم عکس حالت سوم صورت می گیرد.

با اینکار، وقتی شی ای را لاغر می کنیم، بسته به تعداد دفعات ، در واقع آن را از تصویر مورد تحلیل، حذف می کنیم. یا در چاق کردن، در واقع وجود ان را چند برابر می کنیم. این شی اگر نویز هم باشد که بسته به انتخاب هر یک از این حالات، همین اعمال روی آن انجام میگیرد. پس باید در تعیین تعداد دفعات چاق و لاغر کردن دقت داشت.

پس از انجام عملیات مورفولوژی و رسیدن به اسکلت اصلی تصویر مورد پردازش، پردازش های ریاضیاتی شروع می شوند . الگوریتم های موردنیاز از اینجا به بعد وارد کار می شوند. با استفاده از حلقه های موردنیاز در برنامه نویسی و غیره الگوریتم ها نوشته می شوند و خروجی این الگوریتم ها به عنوان دیتاهای خروجی هستند که می توانند به عنوان تصویر خروجی نمایش داده شوند.

در پایان امیدوارم مورد مقبول استاد عزیزم قرار بگیرد.

مریم ابراهیمی نیا