

بسمه تعالی



دانشکده مهندسی کامپیوتر

بینایی کامپیوتر

نام استاد: دکتر محمدی

تمرین هشتم

نام دانشجو: آرمان حیدری

شماره دانشجویی: ۹۷۵۲۱۲۵۲

دی ۱۴۰۱

فهرست

۳ پاسخ سوال اول
۳ الف)
۴ ب)
۵ پاسخ سوال دوم
۶ پاسخ سوال سوم
۶ الف)
۶ ب)
۶ منابع
۷ پاسخ سوال چهارم

پاسخ سوال اول

(الف)

Homogenous: ناحیه ای که تفاوت مینیمم و ماکسیمم آن از حد آستانه یعنی ۳ کمتر یا مساوی باشد.

ابتدا کل ناحیه را به ۴ بخش تقسیم میکنیم، اگر هر کدام از این ناحیه ها homogenous باشد، از آن فعال رد میشویم، اگر نه آن را به 4 ناحیه کوچک تر میشکنیم و این کار را آنقدر تکرار میکنیم تا تمام نواحی homogenous شوند. مانند شکل زیر:

۶	۴	۶	۶	۷	۷	۶	۶
۶	۷	۶	۷	۴	۴	۵	۷
۶	۶	۵	۵	۳	۲	۴	۶
۴	۵	۴	۵	۲	۳	۵	۶
۰	۳	۲	۳	۳	۲	۵	۷
۰	۰	۰	۰	۲	۲	۴	۶
۱	۱	۰	۱	۰	۳	۵	۵
۱	۰	۱	۰	۲	۳	۴	۵

حال شروع به merge کردن میکنیم، یعنی دو ناحیه مجاور را که در صورت merge شدن باز هم homogenous باقی میمانند را در یک گروه قرار میدهیم. اولویت با آنهایی است که parent یکسان داشتند. این کار را آن قدر تکرار میکنیم تا دیگر این شرایط را نداشته باشیم. به این صورت نواحی که تیک زده ام باهم و نواحی ضربدر خورده باهم merge می شوند:

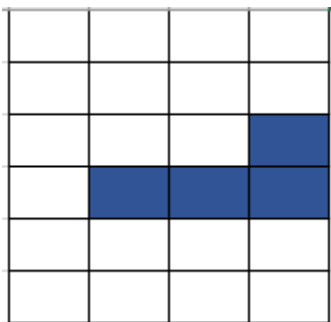
۶	۴	۶	۶	۷	۷	۶	۶
۶	۷	۶	۷	۴	۴	۵	۷
۶	۶	۵	۵	۳	۲	۴	۶
۴	۵	۴	۵	✓۲	۳	۵	۶
۰	۳	۲	۳	۳	۲	۵	۷
۰	۰	✓	۰	✓۲	۲	۴	۶
۱	✓	۰	۱	۰	۳	۵	۵
۱	۰	۱	۰	✓۲	۳	۴	۵

ب)

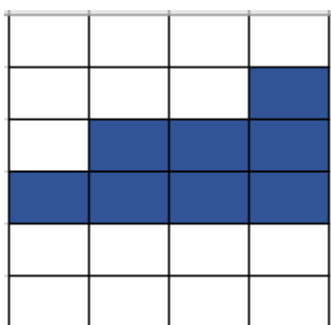
Region splitting and merging ابتدا ناحیه ها را تشکیل می دهد و بعد آن ها را اصلاح میکند تا با توجه به threshold که داریم، homogenous شوند. اما region growing یک نقطه seed در نظر میگیرد و با توجه به threshold، ناحیه ها را گسترش می دهد تا جایی که شرط اضافه شدن به ناحیه را نداشته باشند. و به نوعی برعکس است.

پاسخ سوال دوم

پس از بار اول اجرا نتایج را میبینیم: (فرض کرده ام که zero padding داریم.)



شکل ۱- پس از اعمال erosion اول



شکل ۲- پس از اعمال dilate روی خروجی قبلی

بعد از بار اول هر چند بار که عملگر opening را اجرا کنیم دیگر تفاوتی ایجاد نمیشود. و همواره خروجی همان شکل ۲ است.

پاسخ سوال سوم

(الف)

Otsu: سرعت بیشتر چون یکجا تمام تصویر را بررسی میکند و بین ۲۵۶ مقدار انتخاب می کند – عملکرد ضعیف تر
Adaptive threshold: سرعت کمتر چون به صورت محلی تمام محاسبات را انجام میدهد و مراحل بیشتری هم دارد – عملکرد بهتری دارد چون با شرایط نوری نامتوازن تصویر خود را وفق می دهد

(ب)

در آستانه سازی ساده، مقدار آستانه global است، یعنی برای همه پیکسل های تصویر یکسان است. آستانه تطبیقی روشی است که در آن مقدار آستانه برای مناطق کوچکتر محاسبه می شود و بنابراین برای مناطق مختلف مقادیر آستانه متفاوتی وجود خواهد داشت. در واقع به این صورت عمل میکند که با توجه به همسایگی هر پیکسل یک آستانه پیدا میکند و به هر پیکسل مقدار مناسب میدهد.

ورودی ها به ترتیب:

- عکس ورودی grayscale
- Maxvalue که مقادیر بیشتر از threshold به آن تبدیل میشوند و معمولا ۲۵۵ می گذاریم.
- الگوریتم مورد استفاده، مثلا گاوسی
- نحوه thrsholding، مثلا binary
- اندازه پنجره همسایگی یک پیکسل
- پارامتر ثابتی که در آخر محاسبات هر پیکسل از میانگین کم خواهد شد.

منابع

- [https://www.tutorialspoint.com/opencv/opencv_adaptive_threshold.htm#:~:text=Adaptive%20thresholding%20is%20the%20method,\(\)%20of%20the%20Imgproc%20class.](https://www.tutorialspoint.com/opencv/opencv_adaptive_threshold.htm#:~:text=Adaptive%20thresholding%20is%20the%20method,()%20of%20the%20Imgproc%20class.)

پاسخ سوال چهارم

الف)

به ترتیب سه خروجی otsu و adaptive threshold و بدون پیش پردازش را در سه سلول بخش الف نوتبوک میتوانید ببینید. خروجی adaptive threshold بهتر بوده است چون تصویر نورهای متفاوتی دارد و otsu مناسب آن نیست، بهتر است هزینه runtime زیاد الگوریتم adaptive را برای خروجی مناسب بپذیریم.

ب)

عکس mypic.png پیوست شده است و خروجی به دست آمده هم در نوتبوک قسمت ب آمده است.