به نام خداوند جان و خرد احسان قاسمی - 98102108

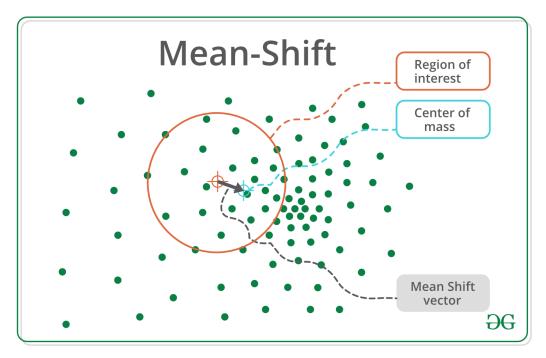
تمرین سوم درس پردازش تصاویر دیجیتال



استاد درس: د كتر داوود پوره جمعه 4/4/2/4/

این فایل برای توضیح کدهای اجرایی و تمامی مواردی است که در طول تمرین به کار برده شده است. خروجی ها با نام های گفته شده ذخیره شده اند و نیز فایل .ipynb نیز که حاوی تمامی خروجی ها و نیز کدهای زده شده است در پیوست ارسال شده است.

روش الگوریتم من برای این سوال به این صورت است:



در نظر بگیرید تعدادی دیتا داریم. این روش برای قطعه بندی به این صورت عمل میکند که روی تمامی نقاط دیتاهای مان پیمایش میکند. برای هر پیمایش هر کدام از نقاط را با توجه به میزان شباهتی که به سایر نقاط دارد به روز رسانی میکند. میزان شباهت را با استفاده از الگورتیم همتنیم. الگورتیم گفته شده را در پایتون شبیه سازی کرده ام. این الگوریتم یک الگورتیم کانیم. الگورتیم گفته شده را در پایتون شبیه سازی کرده ام. این الگوریتم یک الگورتیم المورتیم المورتیم گفته شده را در پایتون شبیه سازی کرده ام. این طول خواهد کشید. ولی من دو ایده اساسی زده ام که زمان اجرای برنامه ام را خیلی خیلی کاهش میدهد. البته هنوزم در حدود یک تا دو دقیقه برای اجرا باید زمان صرف کنید. ایده اولیه من این بود که برای به روز رسانی هر نقطه تمامی iter ماهی گفته شده را طی نخواهیم کرد. به این صورت عمل میکنیم که یک آستانه پایین در نظر میگیریم که در صورتی که اختلاف دو نقطه متوالی در دو گام متوالی از یک حدی کمتر شود میزان پیشروی را برای آن نقطه متوقف میکنیم. این ایده زمان اجرا را خیلی خیلی کاهش میدهد. ایده دوم هم نحوه ورودی به تابع مان است. در نظر بگیرید که تمامی پیکسل های ورودی را اگر به تابع بدهیم تعداد خیلی زیادی پیکسل تکراری را اجرا کرده ایم تمامی پیکسل های ورودی را اگر به تابع بدهیم تعداد خیلی زیادی پیکسل تکراری را اجرا کرده ایم تمامی پیکسل های ورودی با تابع بدهیم تعداد خیلی زیادی بیکسل تکراری را اجرا کرده ایم چرا که در تصاویر پیکسل های وارده به تابع نمیتوانند هر مقداری داشته باشند و همسایه های

موجود در تصویر دارای مقادیر یکسانی هستند. این ایده را من به عنوان یک پرچم در نظر گرفته ام و در صورتی که پرچم فعال باشد آن را اجرا خواهیم کرد.

A. Gaussian mean-shift (MS) algorithm

```
 \begin{aligned} & \underbrace{\frac{\mathbf{for}}{\mathbf{x} \leftarrow \mathbf{x}_n}} & \underbrace{\frac{\mathbf{repeat}}{\mathbf{repeat}}} \\ & \forall n \colon p(n|\mathbf{x}) \leftarrow \frac{\exp\left(-\frac{1}{2}\|(\mathbf{x} - \mathbf{x}_n)/\sigma\|^2\right)}{\sum_{n'=1}^{N} \exp\left(-\frac{1}{2}\|(\mathbf{x} - \mathbf{x}_{n'})/\sigma\|^2\right)} \\ & \mathbf{x} \leftarrow \sum_{n=1}^{N} p(n|\mathbf{x})\mathbf{x}_n \\ & \underbrace{\mathbf{until}}_{\mathbf{z}_n} & \mathrm{stop} \\ & \mathbf{z}_n \leftarrow \mathbf{x} \\ & \underbrace{\mathbf{end}}_{\mathbf{connected-components}}(\{\mathbf{z}_n\}_{n=1}^{N}, \epsilon) \end{aligned}
```

 ${f C}.$ Gaussian MS algorithm in matrix form

```
\mathbf{Z} = \mathbf{X}
\frac{\mathbf{repeat}}{\mathbf{W} = \left(\exp\left(-\frac{1}{2} \|(\mathbf{z}_m - \mathbf{x}_n)/\sigma\|^2\right)\right)_{nm}}
\mathbf{D} = \operatorname{diag}\left(\sum_{n=1}^{N} w_{nm}\right)
\mathbf{Q} = \mathbf{W}\mathbf{D}^{-1}
\mathbf{Z} = \mathbf{X}\mathbf{Q}
\frac{\mathbf{until}}{\mathbf{stop}}
connected-components(\{\mathbf{z}_n\}_{n=1}^{N}, \epsilon)
```

B. Gaussian blurring mean-shift (BMS) algorithm

```
\frac{\mathbf{repeat}}{\mathbf{for}} m \in \{1, \dots, N\}
\forall n: \ p(n|\mathbf{x}) \leftarrow \frac{\exp\left(-\frac{1}{2}\|(\mathbf{x}_m - \mathbf{x}_n)/\sigma\|^2\right)}{\sum_{n'=1}^{N} \exp\left(-\frac{1}{2}\|(\mathbf{x}_m - \mathbf{x}_{n'})/\sigma\|^2\right)}
\mathbf{y}_m \leftarrow \sum_{n=1}^{N} p(n|\mathbf{x}_m)\mathbf{x}_n
\mathbf{end}
\forall m: \ \mathbf{x}_m \leftarrow \mathbf{y}_m
\mathbf{until} \ \text{stop}
\text{connected-components}(\{\mathbf{x}_n\}_{n=1}^{N}, \epsilon)
```

D. Gaussian BMS algorithm in matrix form

```
\frac{\text{repeat}}{\mathbf{W} = \left(\exp\left(-\frac{1}{2}\left\|(\mathbf{x}_m - \mathbf{x}_n)/\sigma\right\|^2\right)\right)_{nm}}
\mathbf{D} = \operatorname{diag}\left(\sum_{n=1}^{N} w_{nm}\right)
\mathbf{P} = \mathbf{W}\mathbf{D}^{-1}
\mathbf{X} = \mathbf{X}\mathbf{P}
\underline{\mathbf{until}} \text{ stop}
\text{connected-components}(\{\mathbf{x}_n\}_{n=1}^{N}, \epsilon)
```

2. این روش از قطعه بندی در همان دستور کار تمرین توضیح داده شده است. من هم همان روش را پیاده کرده ام. به این صورت که در ابتدا تعداد منظمی مرکز قرار میدهیم. در هر پیمایش از حلقه مان فاصله هر پیکسل را با دو مرکز اطراف خودش مقایسه میکنیم و نزدیک ترین مرکز را به آن پیکسل مرتبط میکنیم. به این صورت در هر پیمایش تمامی پیکسل ها به یکی از مرکزها مرتبط میشود و نیز در پایان هر پیمایش هم مرکزها را به این صورت به روز رسانی میکنیم که همه نقاط مرتبط شده به آن مرکز را میانگین میگیریم و مرکز جدید را تعیین میکنیم. زمان اجرای این بسیار طولانی است. چرا که روی همه پیکسل ها به اندازه mm_iter ها و نیز به ازای هر پیکسل بر روی تمامی مرکزهای تعیین شده پیمایش میکنیم. ولی باز من در این جا ایده زدم و تمامی مرکز ها بررسی نمیشن و فقط مرکز های اطراف هر پیکسل تا حد مورد نظری بررسی میشوند اما با این حال هم همچنان زمان اجرای برنامه در کمترین حالت در حدود 10 تا 20 دقیقه زمان طول خواهد کشید. خروجی های این قسمت هم همان طور که خواسته شده نام گذاری شده اند و نیز کد و فایل jupyter notebook

مورد آخر که در متن سوال خواسته شده است درباره ضریب alpha است. با کمی جست و جو متوجه شدیم که این ضریب وابسته به مقدار عکس s است. یعنی با فاصله هر دو مرکز متوالی رابطه عکس دارد. با این حال این مقدار وابستگی مقدار شدیدی نیست و من برای تصاویر و خروجی های ایجاد شده با دو بار آزمایش و خط مقدار 15 را انتخاب کرده ام.