توضیح سوال 2

در این سوال از ما خواسته شده است که الگوریتم mean-shift را روی تصویر داده شده پیاده کنیم. در ابتدا یک فیلتر روی تصویر زدم تا نویزهایش گرفته شوند. لازم به ذکر است که من هرگز موفق به اجرای این الگوریتم روی تصویر با سایز اصلی نشدم و انقدر این الگوریتم سنگین بود که روی لپتاپ من هرگز اجرایش تمام نشد. این شد که برای آنکه اجرای آن زمان معقولی طول بکشد، تصویر اولیه را با مقیاس دلخواه تحت معیار scale کوچک کردم. البته لازم به ذکر است که چون تصویر نهایی هم در هر صورت قرار بود کمی شکل کارتونی بگیرد و جزییات آن از دست برود، اطلاعات خاصی را از دست نداده‌ایم. همچنین یک فرض ساده کننده دیگر گذاشته ام که تعداد iterationها حداکثر 5 تا باشد تا خیلی طولانی نشود. به عنوان یک مثال که نسبتا سریع و حدود 1 دقیقه ای اجرا شود، scale را برابر 0.04 گرفتم و این نتیجه حاصل شد:

این الگوریتم به جهت کلاستر کردن نواحی است. در ابتدا باید معیار کلاستر کردنها را مشخص کنیم که این کار در تابع create_feature_vector انجام شده است. برای هر کلاستر 5 معیار در نظر گرفته ایم که سه معیار اول رنگ هستند (بسته به فضایی که در آنها کار میکنیم میتوانند RGB یا HSV یا LUV یا ... باشد). چون در اینجا عمدتا بسته به رنگ میخواهیم کلاستر کنیم و برای هر کلاستر هم رنگ میانگین آن خوشه را برای همه نقاطش میگذاریم، لذا بهتر است به فضای HSV که به رنگ حساستر است برویم. لازم به ذکر است ابتدا مطابق اسلایدها به فضای LUV بردم اما نتیجه زیر حاصل شد که خیلی مطلوب نبود:

همچنین طبق اعدادی که میدیدم و نسبتها، اگر نسبت معیارهای طول و عرض به 3 معیار رنگی را 1 به 4 بگیرم، بردار ویژگی بدست آمده متوازن خواهد بود. برای هر خوشه مقدار intensity ها بهم نزدیک است و در مرکز خوشه بیشینه است.هر چه قدرهم که از مرکز خوشه دورتر شویم، مقدار کاهش پیدا میکند. در کل تصویر نیز در k نقطه (مثلا) که تعداد مرکزهاست مقدار intensity بیشینه است. (برخلاف الگوریتم kmeans که ممکن بود همه کلاسترها یکسان باشند و توزیع یونیفرم هم داشته باشند).

در تابع mean shift هم هر بار یک نقطه را درنظر میگیریم. برای نقاطی که در شعاع همسایگی آن نقطه هستند، میانگین میگیریم بین همه نقاط و نقطه را از آنجا به نقطه جدید انتقال میدهیم. برای نقطه جدید هم همین را تکرار میکنیم تا ... . در نهایت اگر نقطه مرکزی خیلی تغییر نکند الگوریتم همگرا شده است. برای این کار همه اختلافها را سنجیده ایم و اندازه آنها را جمع زده ایم باهم که اگر از یک مقدار آستانه‌ای کمتر بود، دیگر الگوریتم همگرا شده است و آن را تمام میکنیم. پس از اجرای الگوریتم نیز روی آن فیلتر میزنیم تا یکسانتر و یکدستتر شود. علت اصلی آن است که در این روش یک سری کلاستر داخل یک سری کلاستر بزرگتر ایجاد میشود (همانند سوال 3) و باید تاثیر آنها را کمتر کرد. اما چون خیلی تصاویری که امتحان میکردم روشون کوچک بوند، یک فیلتر 3 در 3 برایشان زیاد به حساب میامد و باعث میشد که خیلی تار شوند. لذا این خط را کامنت کردم. در صورتی که روش عکسی با مقیاس بزرگ مثلا در حدود یک چهارم عکس اولیه امتحان میکنید آن را اجرا کنید.

در آخر زمان اجرای الگوریتم نیز چاپ میشود.