

توضیحات تمرین سری اول

سید امیر محمد سادات شکوهی

شماره دانشجویی: 98100175

1. در این سوال به خاطر نویز نقطه‌ای در تصویر اول یک فیلترمدین با اندازه 15 در 15 اعمال شده است. سپس یک تبدیل لگاریتمی با $\alpha = 0.1$ (در فرمول نوشته شده در اسلاید های درس) اعمال شده است. اندازه فیلتر و α با آزمایش کردن بدست آمده است.

2. در این سوال یک کد برای هر دو حالت تغییر رنگ نوشته شده است. اما این کد برا اجرا شدن نیاز به یک فایل متنی برای گرفتن اطلاعات مربوط به رنگ ها نیاز دارد. فرمت این فایل ها به گونه‌ای است که در خط اول اسم عکس ورودی و خروجی آمده است. حال عکس مورد نظر را لود کرده و به فضای HSV می بریم و یک کپی از ماتریس عکس لود شده در حافظه نگه می‌داریم. حال ادامه فایل متنی را می‌خوانیم. در هر خط ادامه فایل متنی 7 عدد نوشته شده است. که دو عدد اول ابتدا و انتهای یک بازه برای مقدار H در فضای رنگی HSV است. همینطور اعداد سوم و چهارم نشانگر یک بازه برای مقدار S و اعداد پنجم و ششم نشانگر یک بازه در برای مقدار V است. حال تمام پیکسل هایی که مقدار اینتنسیتی آنها در همهی این بازه ها قرار دارد را پیدا می‌کنیم. این کار را در خطوط 18 الی 22 انجام می‌دهیم. حال عدد هفتم را به مقدار H همه این پیکسل های پیدا شده در ماتریس کپی شده اضافه می‌کنیم. این کار را برای همه خطوط درون فایل متنی انجام می‌دهیم. با این روش می توان بازه های متنوعی را انتخاب کرد. مثلا در بعضی مقادیر مربوط به H نیاز است که مقدار S یا V بالا باشد تا رنگ مورد نظر انتخاب شود. در نهایت عکس کپی شده را به فضای RGB برمی‌گردانیم و ذخیره می‌کنیم. مقادیر بازه های مختلف با سمپل گیری از پیکسل های عکس و سعی و خطا پیدا شده است.

3. در این تمرین اول یک مشتق در راستا x از عکس گرفته شده و یک threshold برای مقادیر مثبت زده شده است. این کار برای پیدا کردن تطابق دقیق‌تر است. چون با یک نگاه شهودی می‌بینیم که بهترین حالت تطابق هر کانال رنگی این است که لبه ها رو پیدا کنیم و آنها را بر هم تطابق بدهیم. برای پیدا کردن لبه ها از یکی از ساده و سریع ترین روش ها استفاده می‌کنیم که مشتق گرفتن در یک راستا است. حال سراغ پیدا کردن بهترین تطابق می‌رویم. این کار را به صورت بازگشتی انجام می‌دهیم. تا زمانی که ابعاد عکس از 500 بیشتر است ابعاد آن را نصف می‌کنیم. این اتفاق در تابع `find_translate` اتفاق افتاده است. هنگامی که عکس به ابعاد به اندازه کافی کوچک برسد شروع به جابجا کردن یک چنل در حدود یک باز حول صفر به طول 30 می‌کنیم. از -30 تا 30 در هر دو راستا x و y جابجا میکنیم (این فرایند در تابع `check` پیاده سازی شده‌است). در هر مرحله از جابجا از جابجا کردن ناحیه وسط و مشترک بین همه حالت ها را در نظر می‌گیریم. حال مجموع مربعات اختلاف اینتنسیتی های این ناحیه را محاسبه میکنیم (این فرایند در تابع `get_diff` پیاده سازی شده‌است) و کمترین حالت را پیدا می‌کنیم. و مقدار جابجا شده را به تابع بازگشتی برمی‌گردانیم. حال در مرحله بالاتر و برای عکس با ابعاد دو برابر مقادیر پیدا شده را دو برابر می‌کنیم و در بازه‌ای به طول 5 حول اعداد پیدا شده دوباره به دنبال تطابق دقیق می‌گردیم و مقدار پیدا شده را برمی‌گردانیم.

پس از اتمام الگوریتم بازگشتی حال اعداد به دست آماده را بر روی عکس اصلی اعمال می‌کنیم و کانال های رنگی را جابجا می‌کنیم و نتیجه حاصل می‌شود. کانال سبز به اندازه 10 پیکسل به راست و 88 پیکسل به پایین جابجا شده و کانال قرمز به اندازه 13 پیکسل به راست و 177 پیکسل به پایین جابجا شده است.

4. در این سوال پس از لود کردن تصاویر در هر کانال رنگی اول هیستوگرام را با استفاده از توابع آماده محاسبه می‌کنیم. سپس توزیع جمعی هر هیستوگرام را نیاز بدست می‌آوریم و آنها را نرمالایز می‌کنیم. حال برای هر مقدار در بازه 0 تا 255 حساب می‌کنیم که هر مقدار به چه مقداری باید مپ شود و این مپ کردن را در آرایه t نگه می‌داریم. برای حساب کردن هر مقدار از اینتنسیتی در عکس تاریک تعداد پیکسل های با اینتنسیتی کمتر یا مساوی این اینتنسیتی را با استفاده از توزیع جمعی بدست می‌آوریم. حال در توزیع جمعی هیستوگرام عکس روشن‌تر اندیس اولین جایی که از این تعداد بدست آماده بیشتر باشد رو پیدا می‌کنیم و مقدار اینتنسیتی در عکس تاریک را به این اندیس مپ می‌کنیم. حال پس از پرشدن آرایه t حال مقادیر عکس تاریک را مطابق این آرایه عوض می‌کنیم. ما با این کار هیستوگرام عکس حاصل را دقیقاً مانند هیستوگرام هدف نمی‌کنیم ولی توزیع جمعی به نوبه یکسانی را بدست می‌آوریم.