



دانشگاه صنعتی امیرکبیر
(پل تکنیک تهران)

پاسخ تمرین اول بینایی کامپیوتر
استاد درس جناب آقای دکتر صفایخش
نیمسال اول سال تحصیلی ۱۴۰۲-۱۴۰۱

محسن عبادپور

شماره دانشجویی:

۴۰۰۱۳۱۰۸۰

ایمیل: m.ebadpour@aut.ac.ir

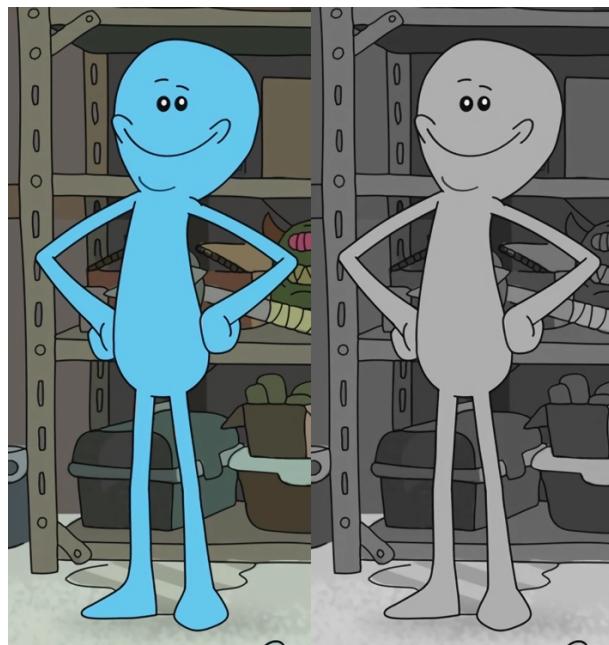
فهرست پاسخ ها

۲ مسئله اول
۳ مسئله دوم
۴ مسئله سوم
۸ مسئله چهارم
۱۱ مسئله پنجم

مسئله اول

مسئله اول

برای پاسخ گویی به این مسئله ابتدا تصویر خواسته شده را با تابع `imread` بارگذاری کرده و سپس توسط تابع `cvtColor` با پارامتر `COLOR_BGR2GRAY` آن را به سطح خاکستری تبدیل می‌کنیم تا نتیجه حاصل شود. برای ادغام خروجی با حالت رنگی نیز از تابع `hconcat` استفاده می‌کنیم تا بصورت افقی تصاویر را کنار هم قرار دهیم:

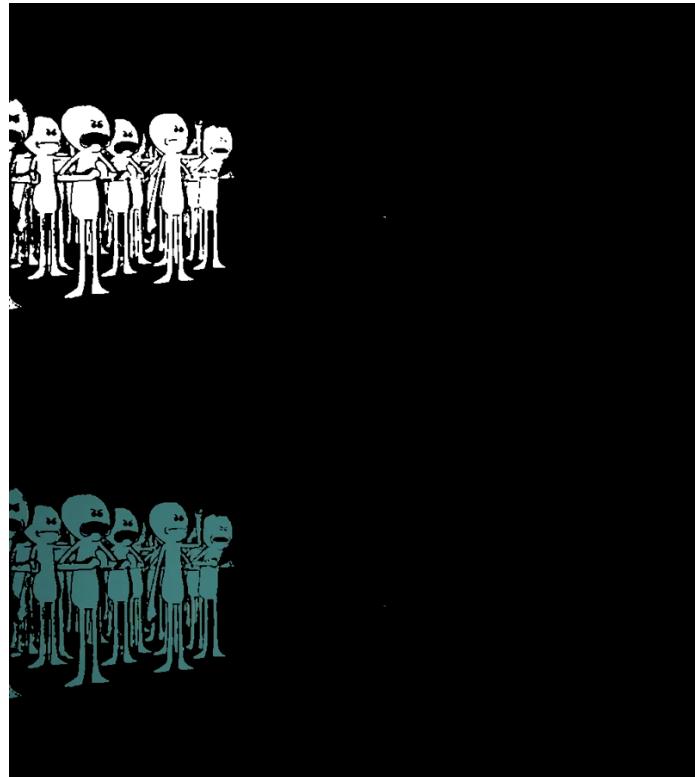


ضمنا با توجه به اینکه تصویر رنگی سه کanalه می باشد اما تصویر خاکستری تک کanalه می باشد، تصویر خاکستری را در سه کanal تکرار می کنیم تا به جهت پیاده سازی امکان کنار هم قرار دادن تصاویر فراهم شود.

مسئله دوم

مسئله دوم

برای حل این قسمت بایستی سطح رنگی مربوط به آدمک ها را استخراج کرده و سپس با تعریف یک حد آستانه‌ی پایین و بالا اقدام به شناسایی آدمک ها نمود بدین نحوه که پیکسل های تصویر را پیمایش کرده و در صورت ارضای شروط آستانگی در هر سه کanal، آن پیکسل را به عنوان پیکسل آدمک در نظر گرفت؛ لذا بنای تشخیص آدمک ها رنگ آنها می باشد. برای این منظور بندۀ تصویر را ابتدا در فتوشاپ بارگذاری کردم و سپس مقدار پیکسل مربوط به بدن یکی از آنها را بدست آورده و سپس در کد پیاده سازی شده با سعی و خطا سعی کردم برای آن یک حد پایین و یک حد بالا محاسبه کنم تا بدین صورت آدمک ها از ماقبی اجرا با خطای کمتری شناسایی شوند که نتیجه آن بصورت زیر می باشد:



در پیاده سازی خود بندۀ از تکنیک masking استفاده کرده و در حین پیمایش پیکسل های تصویر، mask مربوط به آن را بدست آورده و سپس با اعمال AND اقدام به استخراج آدمک ها از تصویر کردم. لازم به ذکر است در سعی و خطا های خود اگر میخواستم دهان آدمکها یا چشم های آنها را هم شناسایی کنم خطای False Positive افزایش چشمگیری می یافت چرا که آن سطح رنگی با شنل قرمز رنگ و قسمت تاریک بالای تصویر همپوشانی بالایی داشت.

مسئله سوم

مسئله سوم

برای قسمت اول، تابعی با امضای SplitCropper پیاده سازی شد که تصویر ورودی را دریافت کرده و با حذف به ترتیب ۱۲ و ۲۵ پیکسل از بالا/پایین و چپ/راست اقدام به حذف حاشیه مشکی می کنیم و سپس با تقسیم عرض تصویر به سه اقدام به جداسازی کanal های رنگی کرده و کanal های استخراجی را در قالب لیست برمیگردانیم.(ضموناً برابری عرض هر کدام از کanal ها بررسی شده است)

برای قسمت دوم سوال، تابعی با عنوان ColorizeRGB پیاده سازی شده است که کanal های سه گانه را دریافت کرده و سپس به تمامی حالات ممکن کanal ها را روی هم قرار می دهد که نتایج آن بصورت زیر حاصل می شود؛ همانطور که قابل ملاحظه است در همهٔ تصاویر ورودی، ترتیب صحیح قرار گیری کanal های رنگی ردیف سوم و ستون سمت راست می باشد که همان تصویر به عنوان خروجی این تابع برمیگرداند می شود:





برای قسمت سوم سوال و حذف تاری تصاویر رنگی تولید شده، از تکنیک منطبق شدن لبه های استخراجی از هر یک از کanal ها استفاده شده است؛ برای این منظور با استفاده از تابع Laplacian از کتابخانه Open-CV اقدام به استخراج لبه به ازای هر یک از کanal های رنگی می کنیم. در گام بعدی، کanal اول(قرمز) را ثابت در نظر گرفته و سپس در یک بازه محدود(برای مثال ۴۰ الی ۸۰) دو کanal دیگر را در راستا های عمودی و افقی لغزش می دهیم و مجموع مربعات اختلاف(sum of squared difference) را در حالات گوناگون حساب می کنیم(معیار شباهت گفته شده در صورت سوال). در موقعیتی که این معیار کمینه باشد یعنی لبه های کanal های مختلف در موقعیت های متضاد روی هم منطبق شده است. از ان موقعیت ها استفاده کرده و کanal های اصلی رنگی را با آن offset روی هم قرار می دهیم تا تاری از بین برود. برای این قسمت تابعی با امضای DeBlur پیاده سازی شده است که تصویر نهایی را باز میگردد.

بایستی به این نکته توجه داشت که استفاده از کanal های رنگی به جای کanal های لبه و محاسبه شباهت بین انها ممکن است خطاهای حاصل از تقاؤت رنگی را لحاظ کرده و بهبود تاری به بهترین نحوه انجام نشود(تقاؤت پیکسل های تمام سبز یا قرمز؛ لذا استفاده از کanal های رنگی انتخاب بهتری است. نتایج و خروجی ها در صفحه بعد آورده شده است.

از جایی که ما در حال لغزش تصویر در راستای عمود و افق هستیم، با پیچیدگی زمانی $O(N^2)$ مواجه می باشیم که در آن N از مرتبه طول/عرض تصویر می باشد چرا که به ازای تغییر و لغش حتی یک پیکسلی در راستای عمود، تمامی حالات ممکن در راستای افق بایستی ارزیابی شده و معیار شباهت سنجیده شود که این مرتبه تقریباً بالایی محسوب می شود. برای اولین گام سریع شدن

پردازش، همانطوری که منطقی و مشهود است، ارزیابی تمامی تغییرات و حالات ممکن در راستای عمود و افق بصورت توأم با یکدیگر مورد نیاز ما نبوده و قابل انتظار است که کانال ها حداکثر با ۲۰۰-۱۰۰ پیکسل لغزش نسبت به حالت اولیه میتوانند روی هم منطبق شوند؛ لذا به جای بررسی و ارزیابی تمامی حالات ممکن(طول تصویر*عرض تصویر)، صرفا با ۷۵ پیکسل لغزش در هر راستا توانستیم تاری تصویر را از بین ببریم.

دومین روش پیشنهادی برای سرعت بخشیدن به الگوریتم رفع تاری و منطبق نمودن کانال های رنگی، استفاده از هرم ویژگی (Pyramid) میباشد؛ در توضیحات بالاتر ملاحظه کردیم که مرتبه اجرایی از مقدار طول و عرض تصویر پیروی می کند و اگر ما میتوانیم این عملیات را با تصویر کوچکتری انجام دهیم، زمان اجرایی کاهش یافته و به الگوریتم تطبیق کانال ها سرعت بخشیده خواهد شد. برای این منظور میتوانیم تصویر را به ابعاد پایین تر تبدیل کرده و سپس در ان مقیاس اقدام به لبه یابی و رفع تاری کنیم؛ با استفاده مقدار offset بدست آمده در مقیاس کوچکتر و محاسبه offset در مقیاس اصلی، میتوانیم کانال های اولیه با ابعاد اصلی را منطبق کنیم.





مسئله چهارم

مسئله چهارم

برای پاسخ به این قسمت از تابع Canny با بهره از پارامتر های درج شده در مرجع اصلی opencv استفاده می کنیم که نتیجه آن بصورت زیر حاصل شده است:



همانگونه که در تصویر قابل ملاحظه است، گندم زار ها لبه های کوچک و غیر مطلوب را فراهم کرده اند که بایستی به نحوی آنها را نادیده گرفت و لبه های ریز را از تصویر اصلی از بین برد تا الگوریتم Canny دچار خطای بالایی در استخراج لبه نشود؛ در کلاس درس بینایی و پردازش تصویر ملاحظه کردیم که در صورت اعمال فیلتر های هموارساز، لبه های ریز تضعیف و حتی از بین می‌روند؛ لذا برای این سوال نیز ما بایستی ابتدا در تصویر خود از فیلتر های هموارساز استفاده کرده و سپس اقدام به اجرای الگوریتم Canny کنیم.

از این رو، روش پیشنهادی بدین صورت است که از فیلتر های هموارساز گوناگون استفاده کرده و سپس با اجرای Canny لبه های اصلی را بدست می‌آوریم به گونه ای که هموار سازی باعث شده تشخیص گندم زار به عنوان لبه تضعیف شود. بنده از ۴ فیلتر هموارسازی Median, Averaging, Gaussian Averaging و Bilateral از هر کدام به ازای شش اندازه کرنل آزمایش انجام داده ام که بهترین خروجی هر کدام از فیلتر ها در صفحات بعد آورده شده است (سایر نتایج و به ازای همه آزمایش ها در پوشه پیوستی قرار گرفته شده است).

همانطور که قابل ملاحظه است، طبق سعی و خطا و آزمایش های متعدد که نتایج همگی پیوست شده است و همچنین با توجه به عملکرد بهترین آزمایش مربوط به هر یک از فیلتر ها، فیلتر گوسی با اندازه کرنل ۷ و سیگما ۳ توانسته است تصویر را طوری

هموار سازد که الگوریتم Canny با حذف تمام گندمzar به عنوان لبه و با حذف کمترین جزئیات از شی اصلی(خودرو)، لبه های اساسی و اصلی تصویر را استخراج کند.



خروجی ۱ مربوط به فیلتر Averaging با اندازه کرنل ۵



خروجی ۲ مربوط به فیلتر Bilateral با اندازه کرنل ۹



خروجی ۳ مربوط به فیلتر Gaussian با اندازه کرنل ۷(بهترین نتیجه)

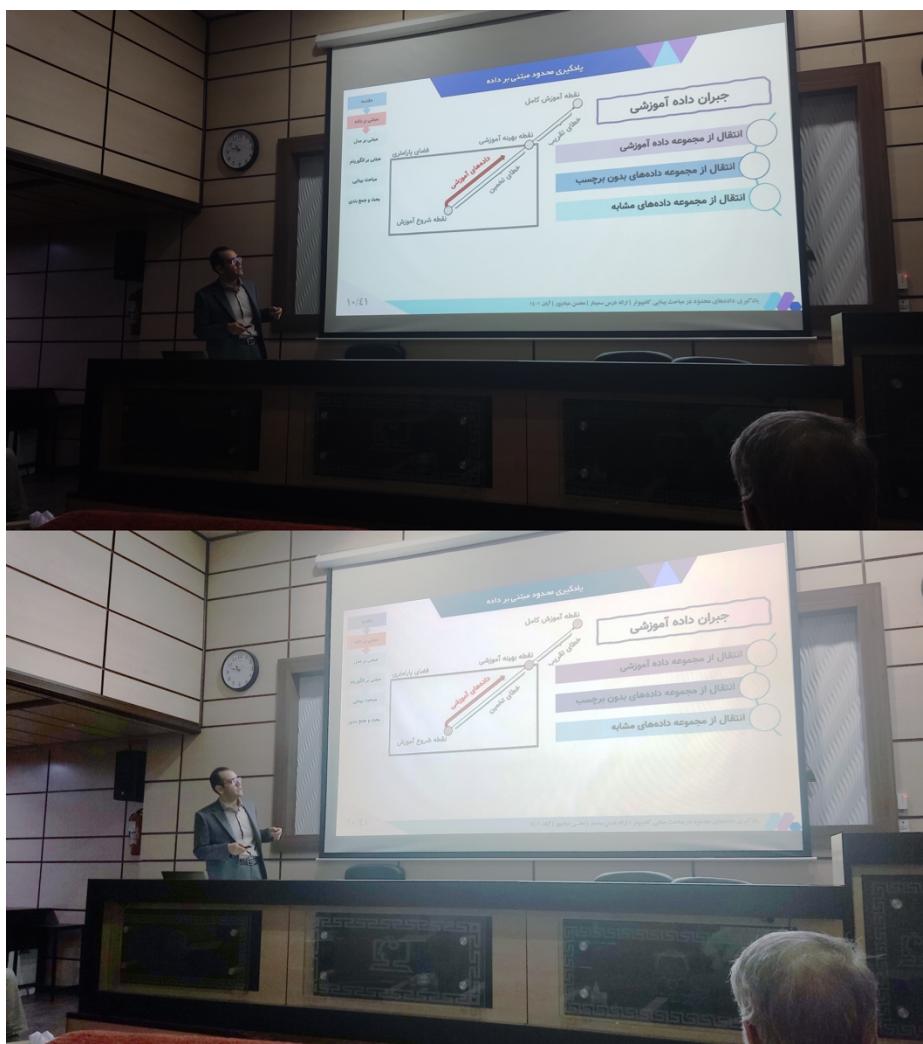


خروجی ۴ مربوط به فیلتر Median با اندازه کرنل ۷

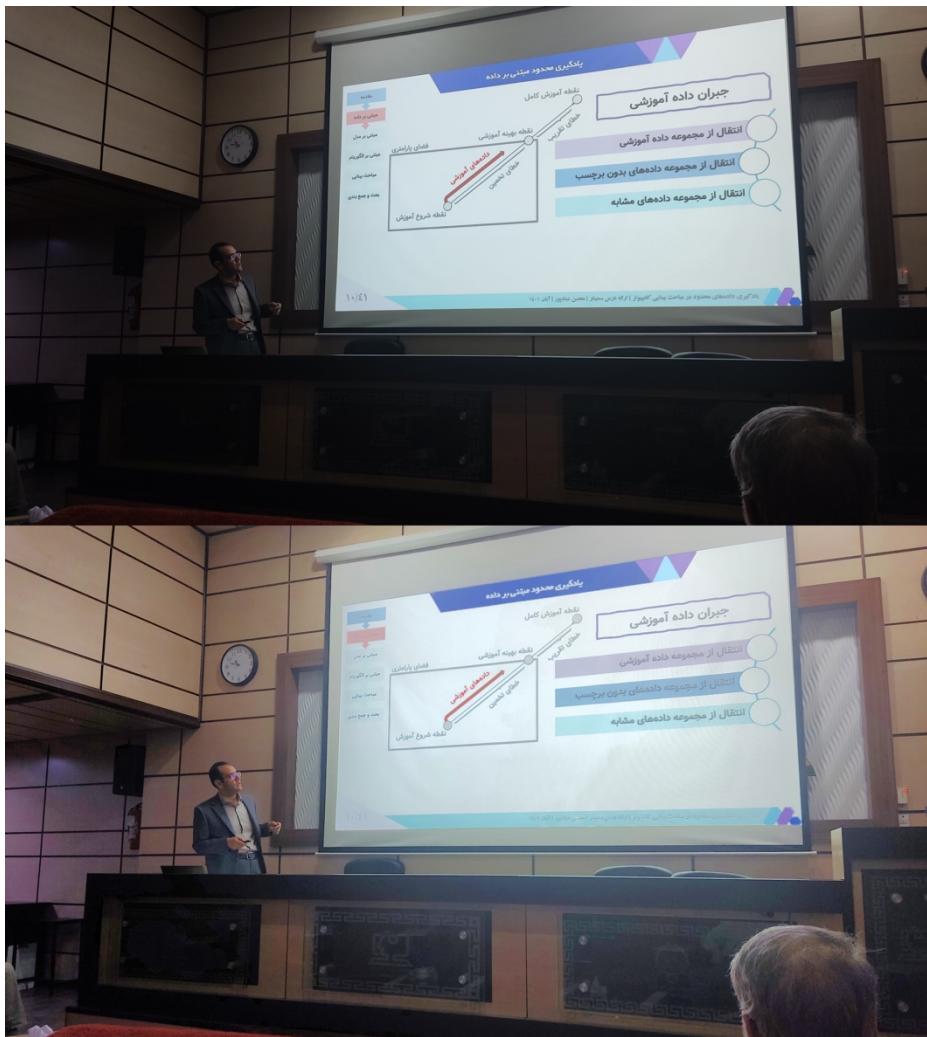
مسئله پنجم

برای هموارسازی هیستوگرام تصاویر رنگی میتوان به چندین طرق مختلف اقدام نمود. ساده لوح ترین روش تبدیل تصویر رنگی به تصویر خاکستری و اعمال هموارسازی هیستوگرام می باشد که در انتهای تصویر رنگی از تصویر خاکستری تخمین زده شود. این روش که در درس پردازش تصویر مورد بررسی قرار گرفته بود نتایج چندان جالبی ارائه نمی کند چرا که تخمین مقدار سه کanal از یک کanal همواره با خطأ مواجه بوده و یک به یک نمی باشد.

دومین روش پیشنهادی اعمال هموار سازی هیستوگرام در کanal های رنگی RGB بصورت مجزا می باشد بدین صورت که هیستوگرام هر کanal رنگی را به تنها یک هموار کرده و نتایج را مجدد کنار هم قرار می دهیم تا تصویر رنگی جدید حاصل شود. بنده این روش را پیاده سازی نموده و آن را روی یکی از عکس های اخیر خود که در حال ارائه درس سمینار با حضور دکتر صفابخش هستم اعمال کرده ام که نتیجه بصورت زیر حاصل شده است (:). تاثیری که این روش در هموارسازی تصویر رنگی دارد این است که اگر تراکم یا نامتعارفی در هیستوگرام هر کدام از کanal های آبی، قرمز یا سبز وجود داشته باشد با این روش هموار می شود؛ اما نکته منفی در خصوص این روش آن است که با استی سه بار عملیات و پردازش هموارسازی هیستوگرام صورت گیرد. این ضعف در روش بعدی حل شده است(نقطه ضعف بعدی مستقل فرض کردن می باشد که در واقعیت چنین نیست چرا که اصالت رنگ دچار تغییر می شود):



سومین روش پیشنهادی، اعمال هموارسازی هیستوگرام در فضاهای رنگی دیگری نظیر HSV می‌باشد که در آنها اصالت و روشنایی و نور رنگ‌ها در کanal های مجزا از هم ذخیره و نگهداری می‌شوند. در این صورت اگر هموارسازی هیستوگرام در کanal مقدار(V) اعمال شود، اصالت رنگ حفظ شده و صرفا نور رنگ‌ها هموار می‌شود. این روش هر دو نقطه ضعف روش قبلی را جبران می‌کند: اول اینکه صرفاً یکبار هموارسازی و در یک کanal صورت می‌گیرد؛ دوم اینکه خود رنگ‌ها تغییر نمی‌یابند. این روش نیز پیاده سازی و با تصویر قبلی ارزیابی شده است که بصورت زیر نتیجه حاصل شده است:



حال اگر نتایج هر دو روش در این عکس بخصوص را کنار هم مقایسه کنیم، ملاحظه می‌کنیم که هموارسازی صورت گرفته در فضای HSV با کیفیت تر، شفاف تر و نزدیک به واقعیت است در صورتی که هموارسازی صورت گرفته در فضای RGB دارای روشنایی تصنویعی بوده و نور سفید توجه زیادی را به خود جلب می‌کند.