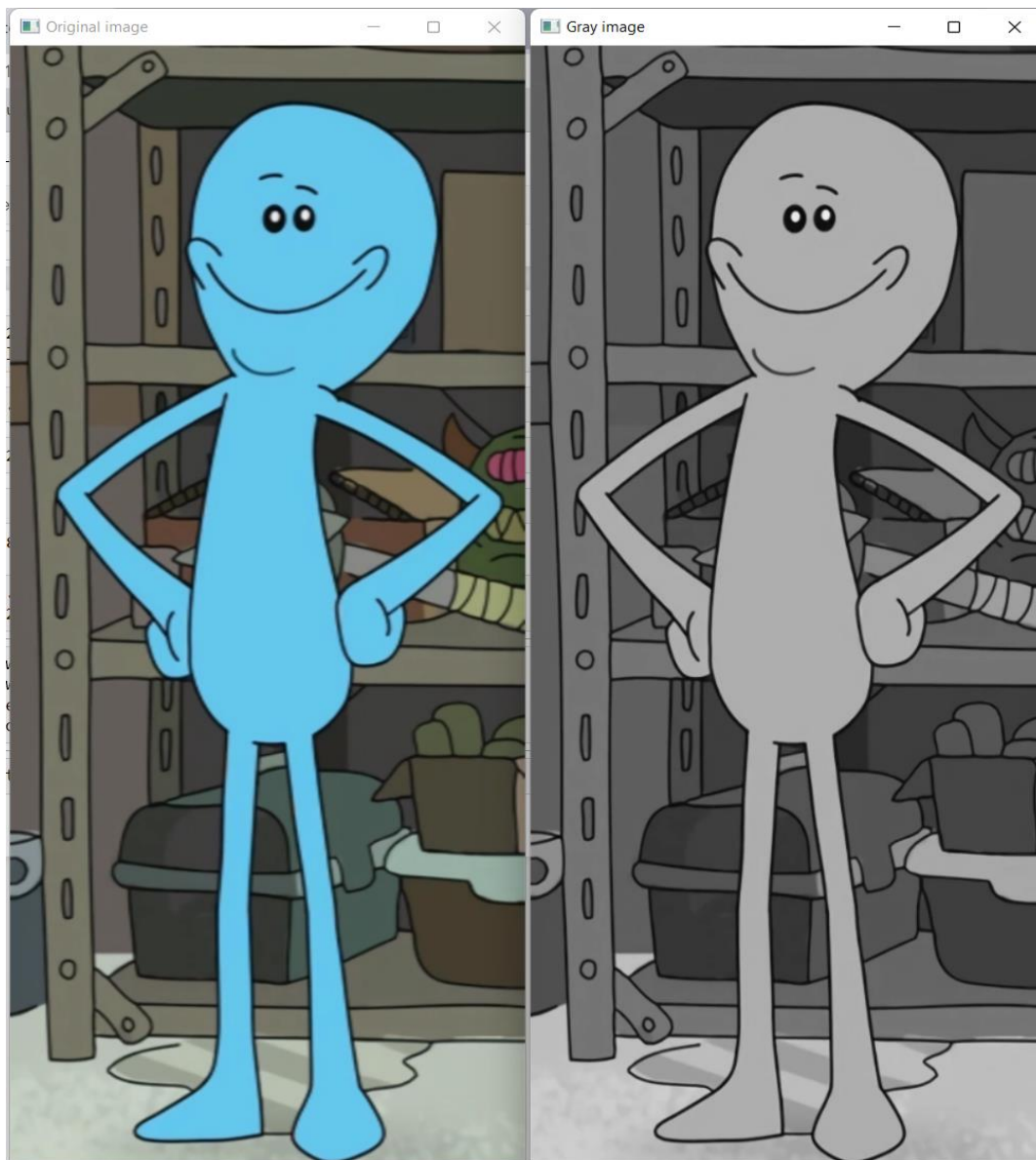


درس : بینایی کامپیوتر

نام و نام خانوادگی: فاطمه توکلی

تمرین شماره ۰۱

۱. در ابتدا تصویر را با استفاده از دستور `cv2.imread("img")` خوانده نمایش آن در شکل ۱ آورده شده است. سپس با استفاده از `cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)` آن را به تصویر سطح خاکستری تبدیل کرده (شکل ۲).



شکل ۱: تصویر اصلی

شکل ۲: تصویر سطح خاکستری

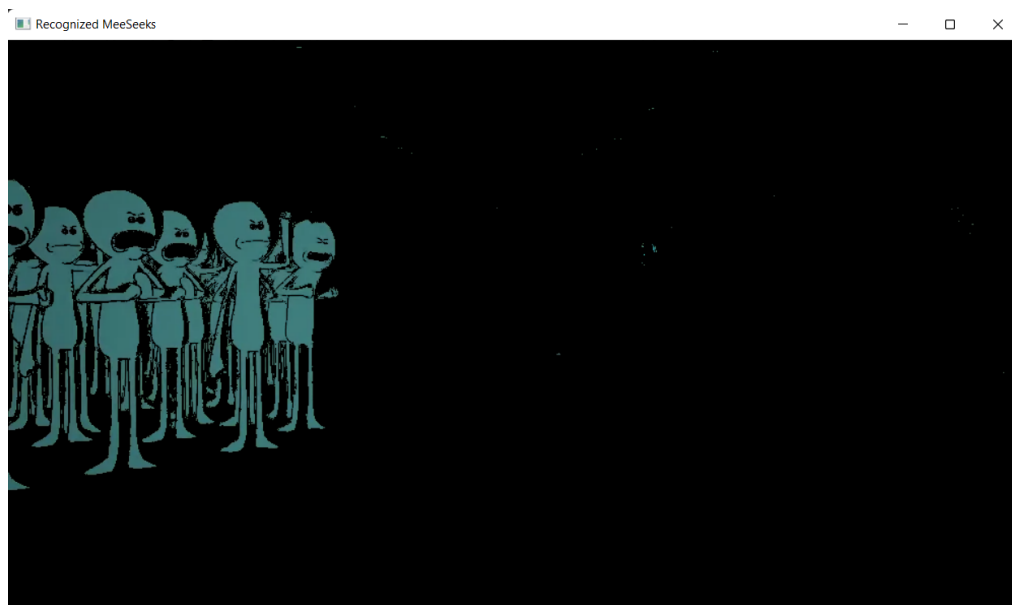
کد این قسمت در فایل `HW01.ipynb` و قسمت Q1 وجود دارد.

۲. در این قسمت با تعیین مقدار آستانه که در شکل ۳ آورده‌ایم برای هر کدام از کانال های تصویر که تقریباً با توجه به رنگ Meeseeks است به نتیجه ایی که در شکل ۴ آمده رسیده‌ایم.

```
(83 < pixel[0] < 255 and 81 < pixel[1] < 255 and 10 < pixel[2] < 80):
```

شکل ۳: مقدار آستانه هر سه کانال رنگی تصویر

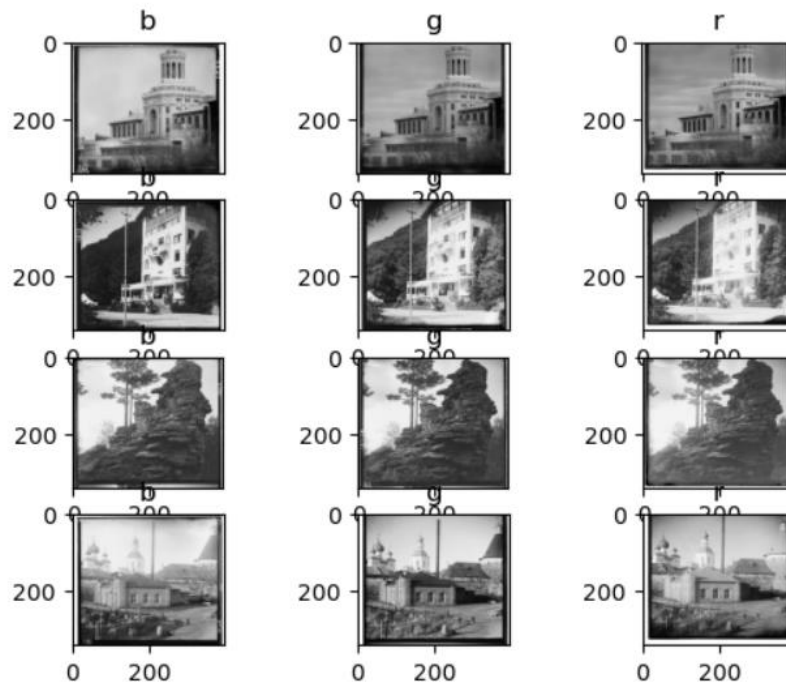
همچنین به علت اینکه به خوبی پس زمینه جدا نشده روش دیگری نیز امتحان شده‌است که با استفاده از مورفولوژی ماسکی برای تشخیص Meeseeks پیدا می‌کند و سپس بر روی عکس و پس زمینه به ترتیب خود ماسک و عکس ماسک را bitwise_and می‌کنیم و سپس دو عکس را با استفاده از bitwise_or ترکیب می‌کنیم نتیجه در شکل ۴ نمایش داده شده‌است:



شکل ۴: Meeseeks جدا شده از پس زمینه

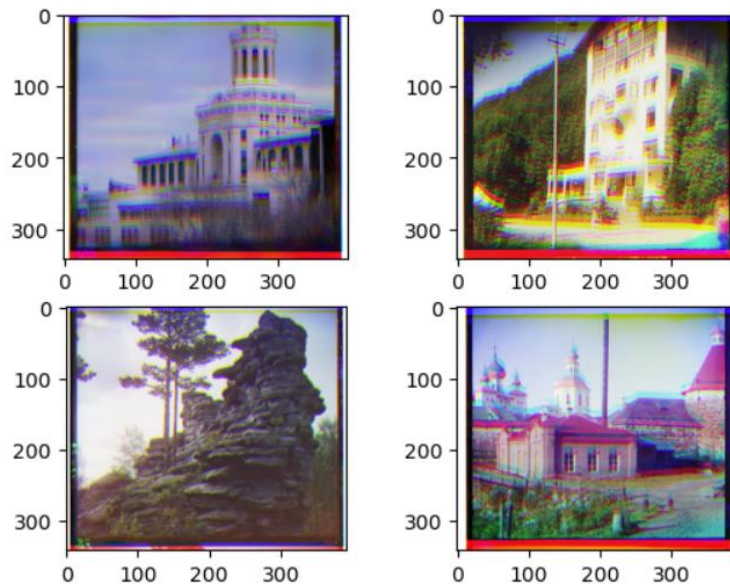
کد این قسمت در فایل HW01.ipynb و قسمت Q2 وجود دارد.

- تابع مورد نظر تصویر ورودی را به سه بخش عمودی مساوی تقسیم و برمیگرداند. رنگ کانال‌ها با توجه به خروجی بخش بعدی با آزمون و خطا پیدا شده‌است. نتایج به صورت زیر می‌باشد:



شکل ۵: هر کانال جداگانه تصاویر داده‌شده

- تصاویر بالا ترکیب شده و نتیجه به صورت شکل ۶ می‌باشد. همانطور که انتظار می‌رفت کانال‌ها به خوبی روی هم تطبیق نیافته‌اند:



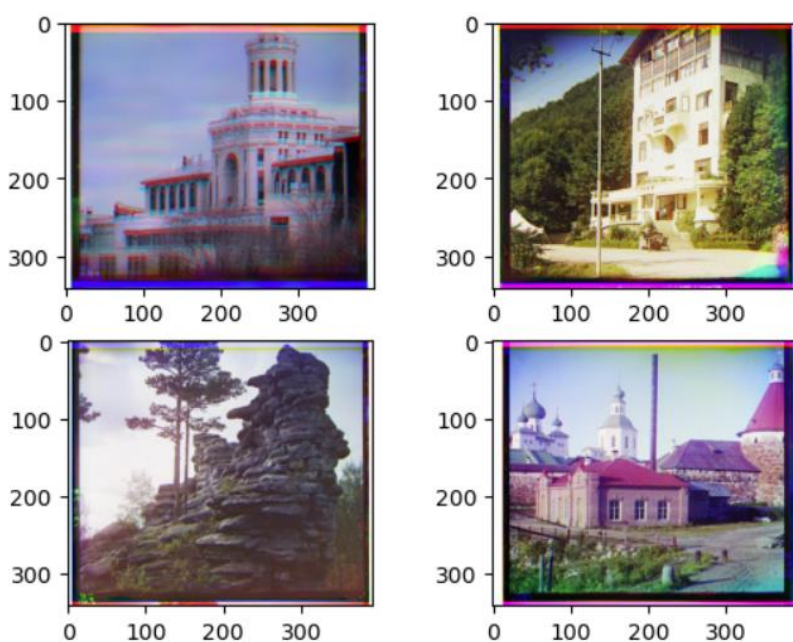
شکل ۶: تصویر ایجاد شده از ترکیب ۳ کانال

- از ایجاد تغییرات روی مجموع ضرایب همبستگی پیرسون به عنوان معیار شباهت دو تصویر استفاده کرده‌ایم. بدین منظور کانال g را ثابت فرض می‌کنیم و دو کانال دیگر را روی آن منطبق می‌کنیم. اگر فرض کنیم حلقه دو بعدی جستجو قسمت اصلی برنامه بوده‌است و n بازه جستجو باشد؛ در این صورت زمان اجرای برنامه قسمت d برابر با:

$$O(0.25n^2)_{\text{reduced_image}} + O(0.25n^2)_{\text{original_imag}}$$

زمان اجرای این قسمت برای ۴ عکس ۷۰ ثانیه بوده‌است که برای هر عکس حدوداً ۱۷ ثانیه می‌باشد.

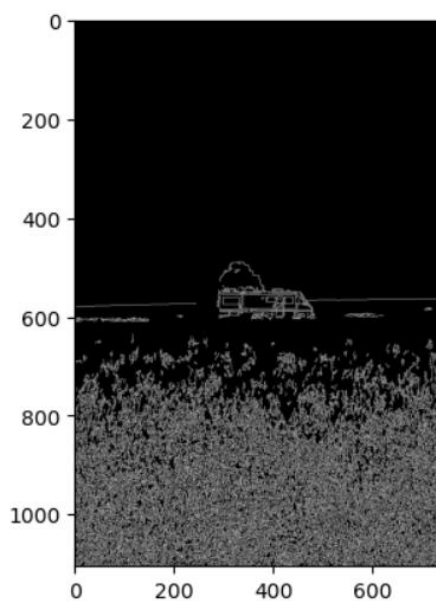
با توجه به مسئله روش دیگری هم وجود دارد که زمان اجرای آن به صورت $O(n^2)_{\text{original_image}}$ بوده ولی زمان اجرا نتیجه به دست آمده تقریباً ۲ برابر بهتر از آن است و نتایج در شکل ۷ به نمایش در آمده‌است.



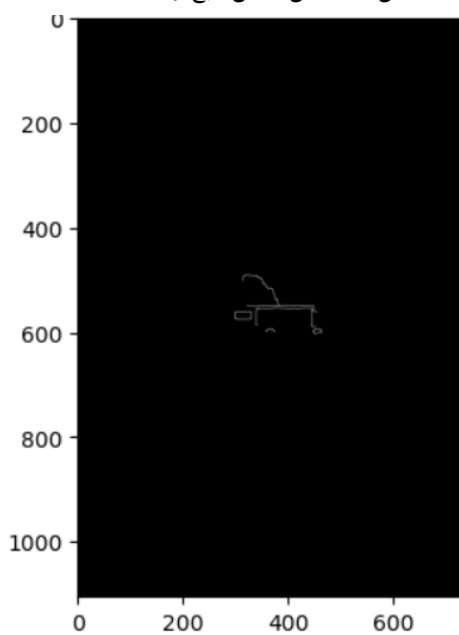
شکل ۷: تصویر بهبود یافته قسمت قبل

کد این قسمت در فایل HW01.ipynb و قسمت Q3_a,b,c وجود دارد.

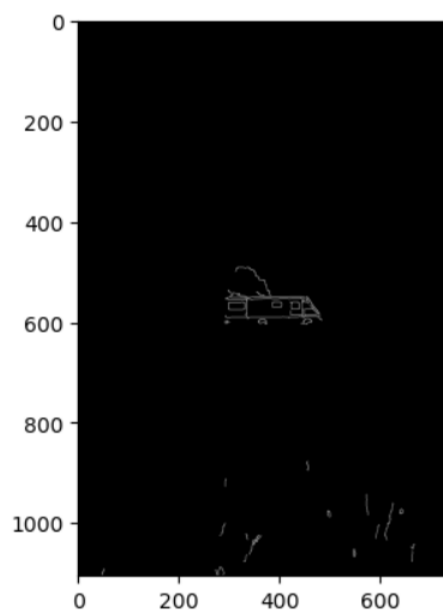
۴. لبه‌های تصویر با استفاده از تابع Canny و مقدار آستانه بالا و پایین به ترتیب ۱۰۰ و ۲۵۰، در شکل ۸ ارائه شده‌است. تعیین مقادیر آستانه می‌تواند تا حدی در برطرف کردن لبه‌های غیرواقعی موثر باشد اما کافی نیست. راه بهتر، اعمال یک فیلتر هموارساز برای حذف جزئیات کوچک و سپس استفاده از تابع Canny است. در شکل‌های ۹ و ۱۰ تصویر لبه‌ها پس از اعمال یک فیلتر گاوسی با واریانس‌های ۲ و ۱.۳۵ و با مقادیر آستانه مشابه ارائه شده‌است.



شکل ۸: حاصل اعمال تابع canny



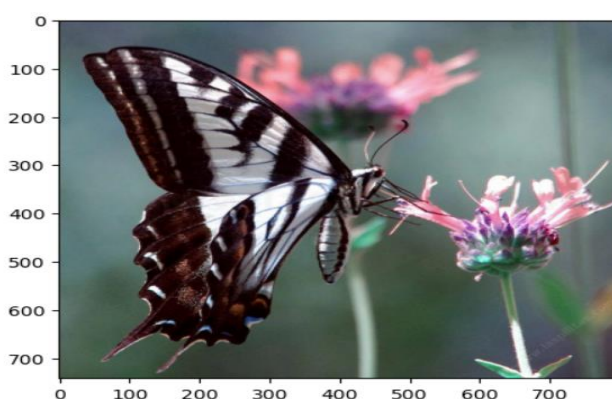
شکل ۹: نتیجه با مقدار آستانه ۲



شکل ۱۰: نتیجه با مقدار آستانه ۱.۳۵

کد این قسمت در فایل HW01.ipynb و قسمت Q4 وجود دارد.

۵. تصویر انتخاب شده و هیستوگرام آن که با استفاده از `cv2.calcHist()` به دست آمده به ترتیب در شکل ۱۱ و شکل ۱۲ ارائه شده است:

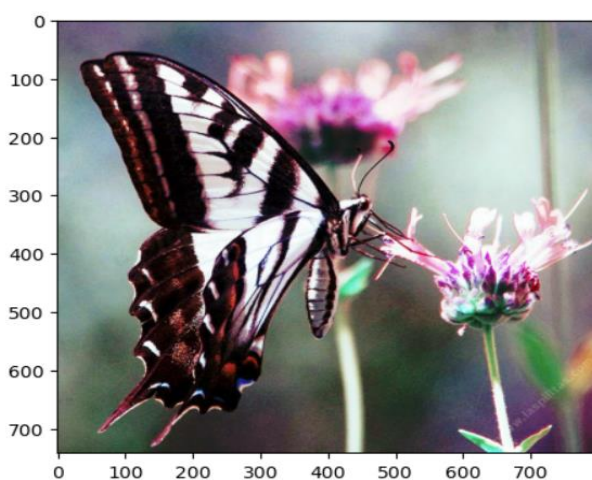


شکل ۱۱: تصویر انتخاب شده

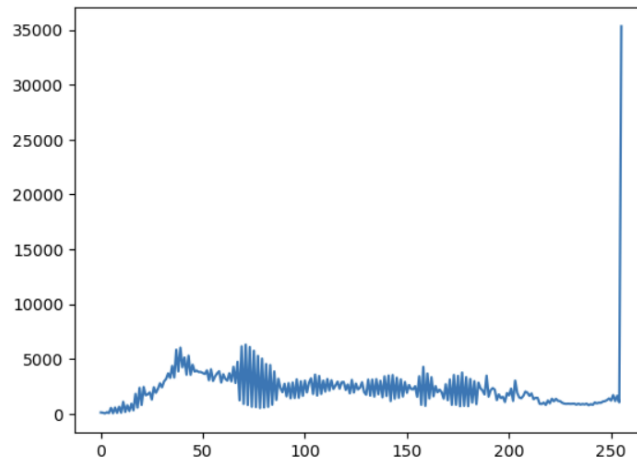


شکل ۱۲: هیستوگرام به دست آمده از تصویر انتخابی

ابتدا تصویر را به سه کانال جدا تقسیم میکنیم سپس تعداد تکرار هر سطح روشنایی در بازه ۰ تا ۲۵۵ به دست می آوریم و cdf حساب میکنم و نگاشت به مقدار جدید را در سه کانال انجام می دهیم در نهایت هر سه کانال را با مقدار جدید ترکیب می کنیم و نتیجه عکس و هیستوگرام آن در شکل ۱۲ و شکل ۱۳ نمایش داده شده است.



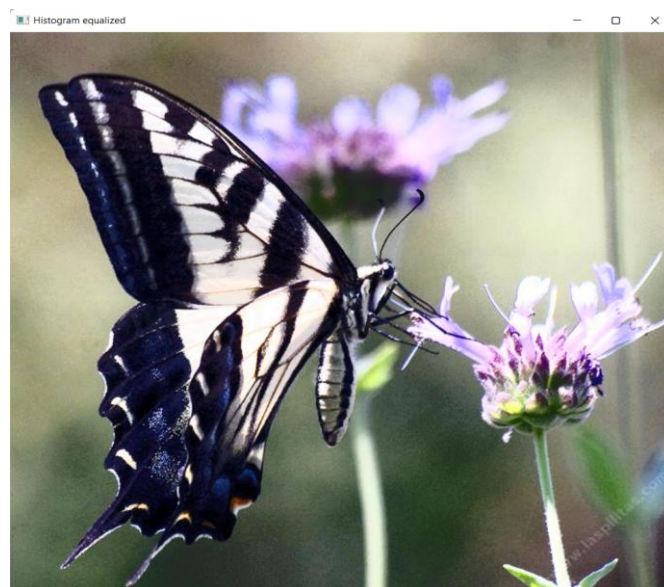
شکل ۱۲: حاصل هموارسازی هیستوگرام



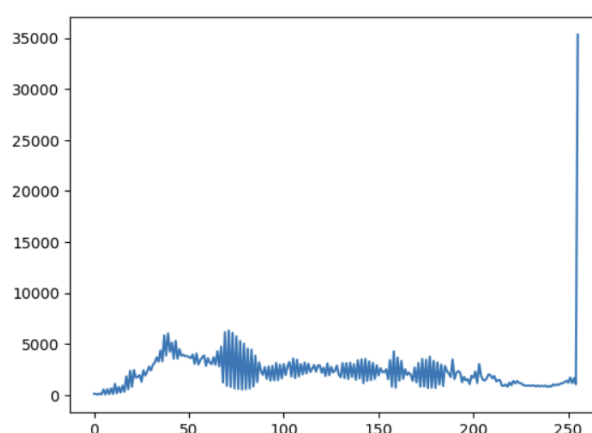
شکل ۱۲: هیستوگرام هموار شده از تصویر انتخابی

همانطور از عکس به دست آمده نیز مشاهده می‌شود اگر سه کانال را مستقل از یکدیگر هموارسازی کنیم رابطه بین آن‌ها بهم به درستی تنظیم می‌شود و رنگ تصویر خراب می‌شود.

همچنین با استفاده از `cv2.equalizeHis` می‌توان کانال رنگی Y را هموار کرد که نتیجه و هیستوگرام آن به ترتیب در شکل ۱۳ و شکل ۱۴ آمده‌است.



شکل ۱۳: حاصل هموارسازی هیستوگرام



شکل ۱۴: هیستوگرام هموار شده از تصویر انتخابی

کد این قسمت در فایل HW01.ipynb و قسمت Q5 وجود دارد.