

به نام خدا

گزارش تمرین 7

نیما کمبرانی ۹۸۵۲۱۴۲۳

مبانی بینایی کامپیوتر

استاد: دکتر محمدی

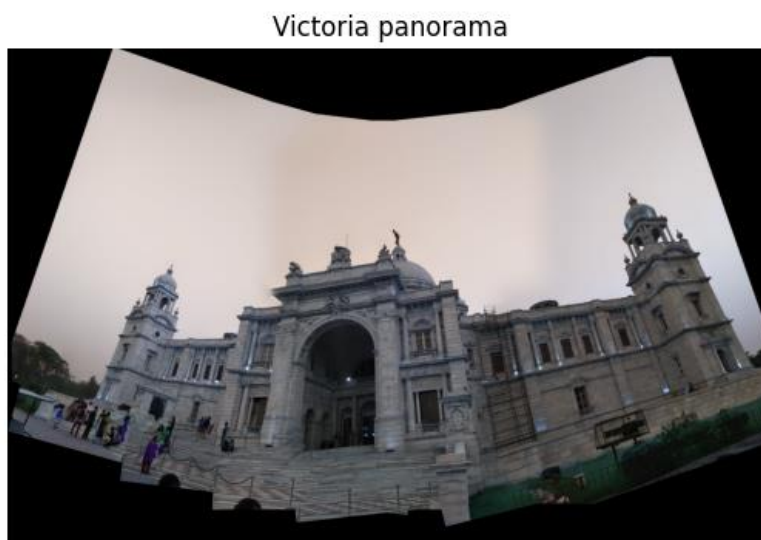
سوال ۱)

در این سوال ۷ تصویر از یک ساختمان داده شده است و می‌خواهیم با اتصال این تصاویر به یکدیگر به یک تصویر با نمای کامل تر از ساختمان برسیم. تصاویر ابتدایی در شکل ۱ آمده است.



شکل ۱. تصاویر ابتدایی از ساختمان

برای اتصال تصاویر به یکدیگر از تابع **Stitcher** در کتابخانه **opencv** استفاده می‌کنیم. این تابع با یافتن نقاط مهم و یافتن تناظر نقاط در بین تصاویر داده شده رابطه‌ی تصاویر را بدست می‌آورد. سپس با تبدیل **Affine** و یافتن پارامتر های آن برای هر تصویر، نقاط تصویر خروجی را بدست می‌آورد. تصویر خروجی از اتصال تصاویر اولیه در شکل ۲ آمده است.



شکل ۲. تصویر حاصل از اتصال تصاویر

سوال ۲)

با توجه به فرمول چرخش برای نقاط داریم:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\text{Cost} = \sum ((\cos \theta + \sin \theta)x_1 - x_2)^2 + ((\cos \theta - \sin \theta)y_1 - y_2)^2$$

در نتیجه باید مقدار مینیموم برای تابع هزینه را بدست آوریم، پس:

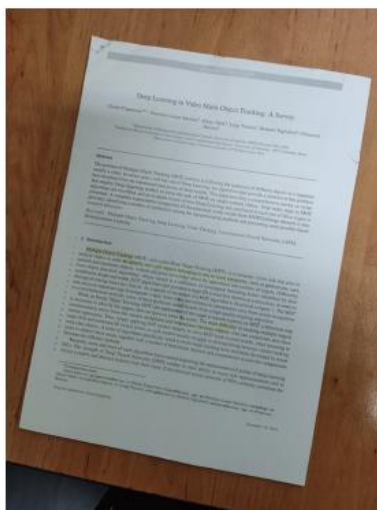
$$d \text{ Cost} / d \theta = \sum 2 * (((\cos \theta - \sin \theta)x_1) * ((\cos \theta + \sin \theta)x_1 - x_2)) +$$

$$2 * (((-\cos \theta - \sin \theta)y_1) * ((\cos \theta - \sin \theta)y_1 - y_2)) = 0$$

$$\sum (\cos (2\theta) x_1^2 - (\cos \theta - \sin \theta)x_2 * x_1) + (-\cos (2\theta)y_1^2 + (\cos \theta + \sin \theta)y_1 * y_2) = 0$$

سوال ۳)

در این سوال در ابتدا یک تصویر از یک کاغذ که بصورت کج قرار گرفته است، داریم. می‌خواهیم با یافتن گوشه های کاغذ آن را از میان پس زمینه برش داده و در نهایت با بهبود کیفیت آن در خروجی نمایش دهیم. این تصویر در شکل ۳ آمده است.



شکل ۳. تصویر اولیه

الف)

با استفاده از تابع `cvtColor` عکس رنگی را به حالت سیاه و سفید در می‌آوریم. تصویر خروجی در شکل ۴ آمده است.

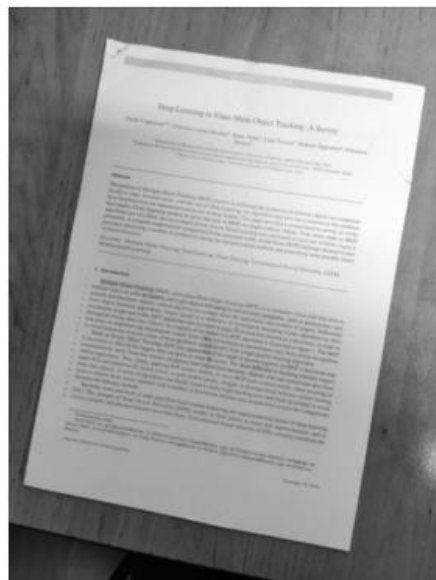


شکل ۴. خروجی تصویر سیاه و سفید از تصویر اولیه

(ب)

برای کاهش نویز و مؤلفه های فرکانس بالای تصویر، مانند نوشته های روی کاغذ، از فیلتر گاوسی استفاده می کنیم. اینکار باعث می شود در مراحل بعد برای یافتن لبه ها و شکل ها بسیاری از لبه های غیر ضروری کاهش یابند و محاسبات با سرعت و دقت بالاتری انجام شود.

تصویر خروجی پس از اعمال هموارساز در شکل ۵ آمده است.



شکل ۵. تصویر خروجی پس از اعمال فیلتر گاوسی با سایز ۷

(پ)

برای یافتن لبه های داخل تصویر از لبه یاب Canny استفاده می کنیم. تصویر خروجی این مرحله در شکل ۶ آمده است که با اعمال فیلتر هموارساز در مرحله قبل بسیاری از لبه های غیر ضروری به طوری تضعیف شده اند که تشخیص داده نمی شوند اما لبه های دور کاغذ که مورد نیاز ما است، به خوبی باقی مانده اند و به راحتی قابل تشخیص هستند.



شکل ۶. تصویر خروجی پس از اعمال لبه یاب Canny

(ت)

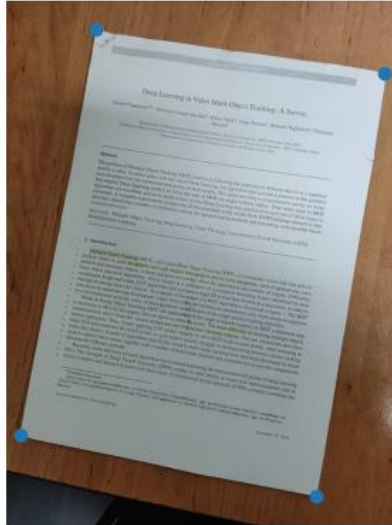
برای تشخیص شکل کاغذ داخل تصویر از contour ها استفاده می کنیم. یک Countour مجموعه ای از منحنی های متصل به هم هستند که محیط یک فضا از پیکسل ها با رنگ یکسان را نشان می دهند. برای کارایی عملکرد آن ها باید ابتدا تصویر را لبه یابی یا آستانه گذاری کنیم. در این سوال در مرحله قبل با استفاده از canny شکل لبه یابی شده است.

با توجه به اینکه findcontour تمامی شکل های داخل تصویر را در خروجی باز می گرداند، نیاز داریم تا شکل کاغذ را از میان آن ها بیابیم. برای اینکار با توجه به اینکه این شکل از بقیه شکل های داخل تصویر بزرگتر است، با ابتدا مساحت هریک از شکل ها را می یابیم سپس شکل با بیشترین مساحت را بعنوان خروجی انتخاب می کنیم.

با توجه به نوع انتخاب شده برای تابع، تعداد نقاط مشخص شده برای یک شکل می تواند کم یا زیاد باشد. در این مسئله با توجه با توجه به اینکه ما به دنبال یک مستطیل در تصویر هستیم، نیاز داریم ۴ گوشه ی آن را بیابیم. تابع

approxPolyDP

با گرفتن نقاط مشخص شده برای یک شکل منحنی و مقدار دقت مورد نیاز، حداقل تعداد نقاطی که منحنی را با دقت خواسته شده مشخص می کنند باز می گرداند. اگر مقدار خطای قابل قبول کم و نزدیک به ۰ باشد نقاط باز گردانده شده، زیاد و برابر با نقاط یافته شده در مرحله قبل خواهند بود. با افزایش خطای قابل قبول، تعداد نقاط کمتر و نقاط مشخص شده با فاصله دقت کمتری شکل اولیه را باز سازی می کنند. در نتیجه با افزایش خطا نقاط روی اضلاع که انحنای آن ها را مشخص می کنند حذف می شوند و تنها ۴ گوشه مستطیل باز گردانده می شوند. نقاط یافته شده برای تصویر کاغذ، در شکل ۷ آمده است.



شکل ۷. گوشه های تشخیص داده شده برای کاغذ

(ت)

برای برش کاغذ از تصویر و تبدیل آن به مستطیل، با توجه به گوشه های پیدا شده در مرحله قبل برای این شکل، هر یک از این نقاط را به یکی از ۴ گوشه صفحه نگاشت می دهیم سپس با یافتن تبدیل مورد نیاز برای اینکار، بقیه نقاط کاغذ را به نقطه متناظر در صفحه نگاشت می کنیم.



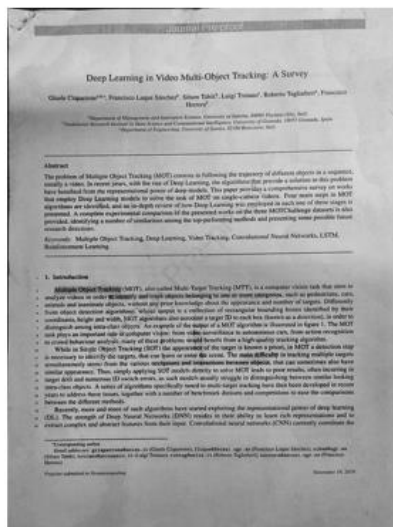
شکل ۸. تصویر بریده شده از کاغذ با استفاده از گوشه های پیدا شده برای آن

(ج)

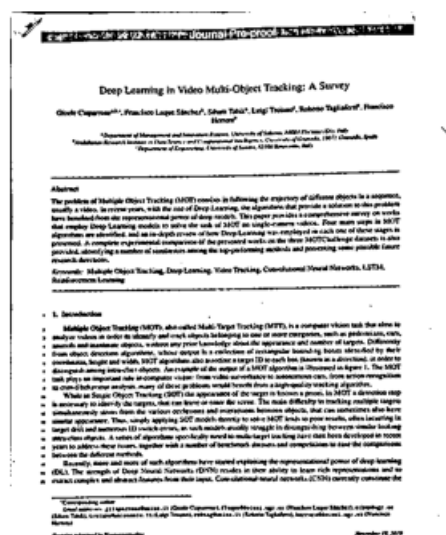
در نهایت برای بهبود کیفیت تصویر، عملیات های زیر به ترتیب بر روی آن انجام می پذیرد:

1. تغییر فضای رنگی از RGB به HSV
2. افزایش مقدار کانال S با ضریب ۲
3. اعمال Clahe بر روی کانال V

4. تبدیل فضای رنگی از HSV به RGB
5. تبدیل تصویر به حالت سیاه سفید
6. اعمال adaptive threshold بر روی تصویر خروجی مرحله ۵



شکل ۹ تصویر سیاه سفید خروجی مرحله ۵



شکل ۱۰ تصویر خروجی نهایی پس از اعمال آستانه گذاری

روش های دیگری مانند بدست آوردن لبه ها با استفاده از sobel و افزودن آن ها به تصویر نیز بررسی شد که نتایج مطلوبی نداشت و ترکیب رنگی کاغذ و لبه ها را تغییر می داد.