

OCR

Ali soltani

Saba razi

Cam scanner



در ابتدا با استفاده از روشی که در تمرین پیاده سازی کردیم:

❖ تبدیل گausین زدیم.

❖ canny میزنیم تا خطوط بدست بیاید.

❖ کانتور هارا بدست می آوریم.

❖ ماکسیمم کانتور را در نظر گرفته و با مقداری درصد خطا کانتور هایی که نزدیک به این ماکسیمم اند را به عنوان گوشه در نظر میگیریم.

❖ نقاط گوشه ی عکسی که محاسبه کرده ایم را به نقاط یک صفحه عادی مپ می کنیم.

خروجی برای بعضی از تصاویر اشتباه بود :

Cam scanner



مشکل کد این بود که برای بعضی حالات چون کانتورهای بدست آمده در تبدیل affine به ترتیب ساعتگرد نبود، تصویر اشتباهی بدست می آمد. بنابراین برای درست کردن جهت این نقاط از تابع reorder استفاده کردیم. این کار با پیدا کردن بالاترین نقطه ی بدست آمده از تصویر انجام میدهم و هر دو حالتی که تصویر ساعتگرد یا پادساعتگرد باشد را با حالت متفاوتی تبدیل میزنیم. برای حالت هایی که کانتورهای بدست آمده از ۴ بیشتر باشند هم یک تصویر محذب از آن بدست می آوریم و آن ها را با ترتیب موردنظر بدست می آوریم.



نکته

- ❖ برای شناخت اعداد و نوع کارت، تصمیم گرفتیم که ابتدا اعداد را روی کارت پیدا کنیم چون با بدست آمدن محدوده ی اعداد می توان از `template matching` استفاده کرد و اعداد را تشخیص داد. البته میتوان `template matching` را بدون بدست آوردن `proposal` انجام داد ولی دقت و سرعت آن کاهش میابد.
- ❖ برای تشخیص کارت ملی بودن یا کارت بانکی بودن هم از فارسی یا انگلیسی بودن اعداد روی کارت استفاده میتوان کرد.

مرحله دوم

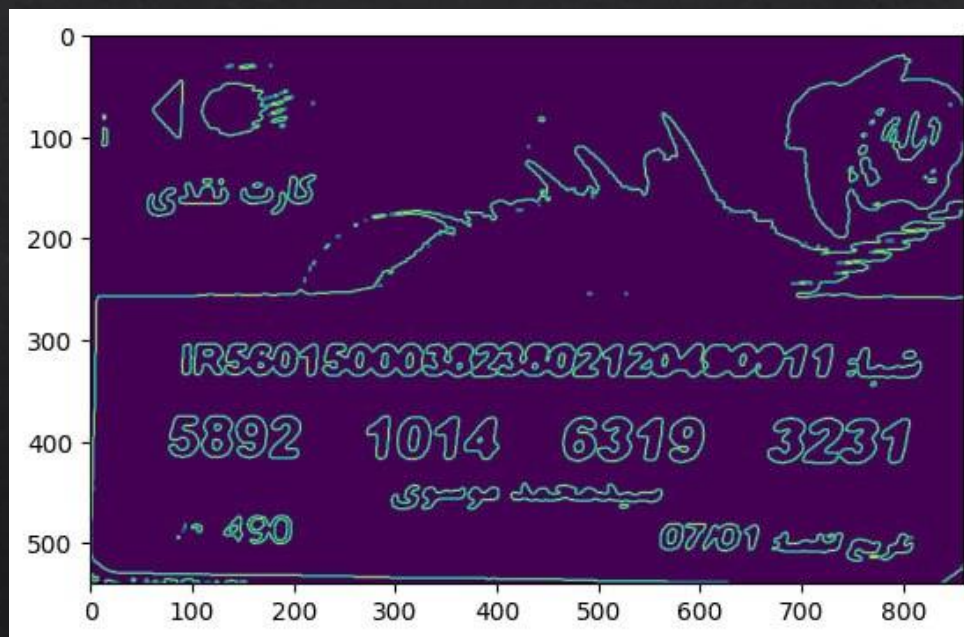
حال برای بدست آوردن proposal ها از روشی که در کلاس برای استخراج proposal های پلاک مطرح شد، استفاده میکنیم:

◊ در این روش ابتدا عکس را با استفاده از otsu باینری میکنیم.



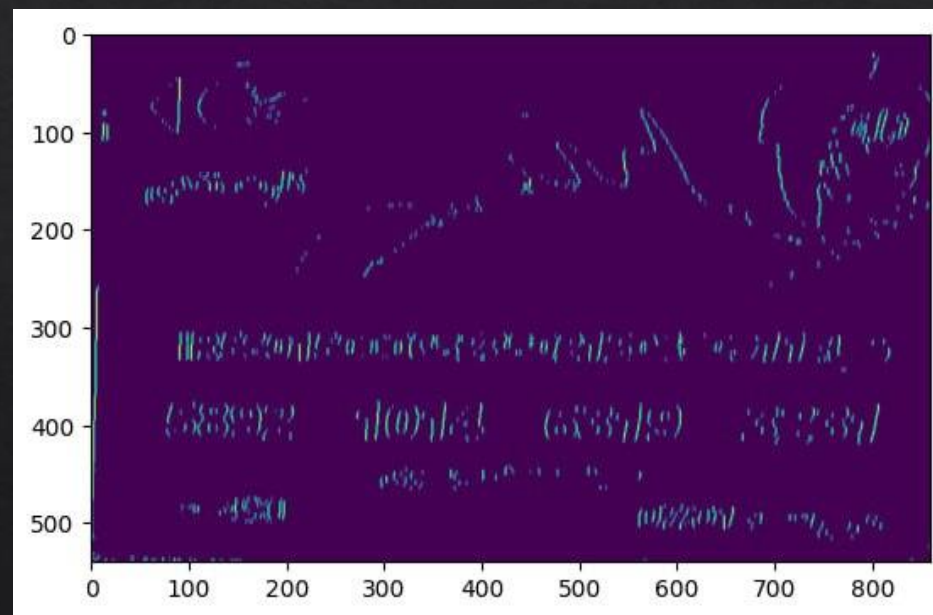
مرحله دوم

- ◆ سپس با استفاده از sobel افقی و عمودی، مقدار magnitude را بدست آورده.
- ◆ روی آن treshhold میزنیم تا دقیقا قسمت هایی که لبه هستند را بدست بیاوریم و بقیه جاها سیاه باشند.



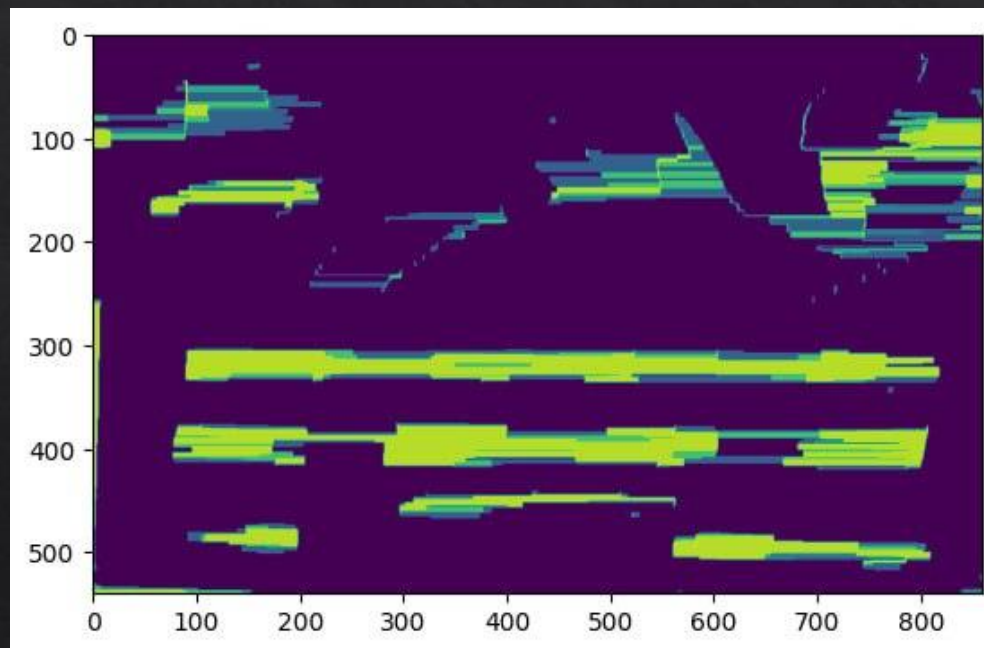
مرحله دوم

◈ در ادامه یک عملیات **open** میزنیم تا نویزها از بین بروند و سطح هموارتر شود و نقاط سفیدی که اطراف آن ها سیاه است، حذف شوند.



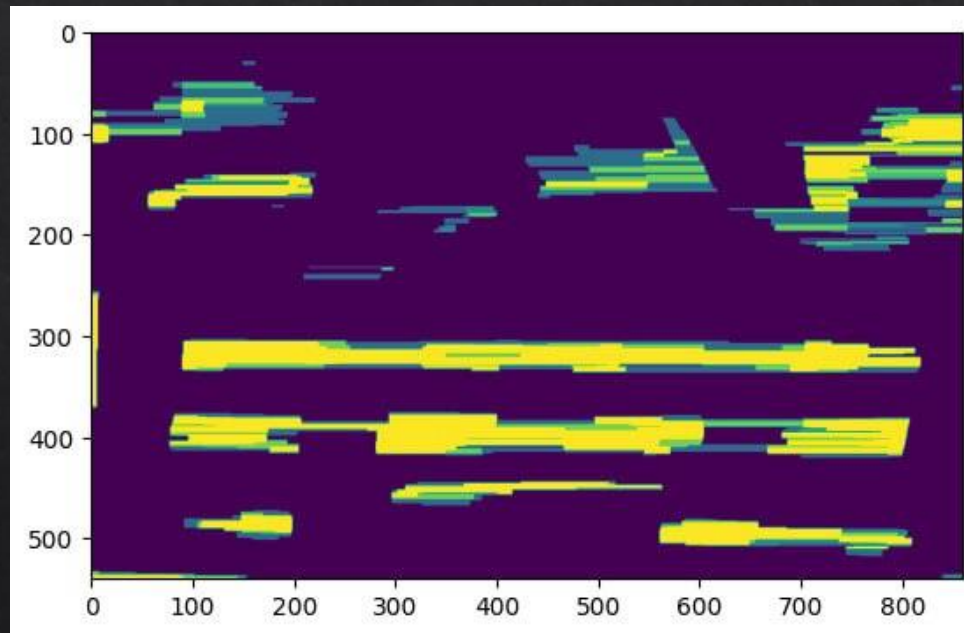
مرحله دوم

◈ بعد یک عملیات close میزنیم تا نقاط سیاهی که بین نقاط سفید هستند از بین بروند و تصویر هموارتر شود و اعداد ای باهم دارند به همدیگر وصل شوند



مرحله دوم

◈ دوباره open میزنیم تا خطوط عمودی از بین برود.



مرحله سوم

- ◆ بعد از انجام این موارد connectedcomponent ها را بدست می آوریم (با opencv).
- ◆ چک میکنیم اگر که اندازه ی طول و عرض آنها از مقداری که میخواهیم بیشتر بود یا طول آن ۸ برابر عرض بود، این connectedcomponent را یک proposal در نظر میگیریم.



پیدا کردن اعداد روی کارت

❖ بعد از پیدا کردن `connectedcomponent` ها، با استفاده از تابع `crop_and_replace_boxes` روی مستطیل هایی که بدست آوریم `otsu treshhold` میزنیم تا اعداد بدست آمده واضح تر شوند و `adaptive` نزدیک تا عکس نويز نگیرد.

❖ در ادامه تابع `make_templates` را داریم. در این تابع از یک عکس تعدادی کاندید برای ساختن تمپلیت میسازد. در این روش ابتدا عکس ها را `resize` میکنیم و به اندازه ی ۵ درجه `rotate` میکنیم و عکس های بدست آمده را ذخیره میکنیم. در ادامه روی عکس اولیه `erode` میزنیم تا ضخامت عدد بدست آمده متفاوت شود و بعد دوباره مانند قسمت قبل `rotate` زده و عکس های حاصل را ذخیره میکنیم. این کار باعث میشود تا از یک عدد در زوایا، اندازه و ضخامت های مختلف دیتا داشته باشیم تا در `template matching` دچار مشکل نشویم.

❖ در تابع `generate_templates` ما تک تک عکس هایی که از ۰ تا ۹ داشتیم (که فونت های متفاوتی دارند) را با استفاده از تابع `make_templates` میچرخانیم و عکس های تولید شده ی جدید را بدست می آوریم.

پیدا کردن اعداد روی کارت

◆ تابع `extract_segments` ابتدا تصویر را باینری میکند و حروف را پیدا میکند. سپس کانتور هارا بدست آورده و مستطیل های موجود را نگه میدارد. ممکن است در یک نقطه چند مستطیل داشته باشیم و برای حل این مشکل مستطیل هایی که درون یک دیگر هستند را بررسی میکنیم و فقط یکی از آن ها را نگه میداریم. ممکن است مستطیل هایی که بدست آوردیم یا خیلی بزرگ و یا خیلی کوچک باشند، در این دو حالت مستطیل هارا حذف میکنیم. حالت بعدی مستطیل های عمودی را حذف میکنیم چون در اعداد انگلیسی هیچ حالتی نیست که مستطیل اطراف عدد عمودی باشد. این موضوع در اعداد فارسی نیست برای مثال عدد ۷ میتواند مستطیل عمودی داشته باشد. در حالت بعد ممکن است مستطیل ها در یک سطر باشند و فاصله ی بین آنها از ۳۰ پیکسل کمتر باشد. در این حالت این دو مستطیل را در یک دسته (cluster) قرار میدهیم.

◆ تابع `template_matching` : در این تابع به ازای هر `template` و هر مستطیلی که در مراحل قبل بدست آوردیم، امتیاز هارا نگه میداریم. برای بدست آوردن امتیاز تک تک `template` هارا `template matching` میزنیم و عدد `score` بدست آمده را سیو میکنیم. بعد از امتیاز های بدست آمده، ماکس میگیریم و عدد بدست آمده را عدد روی عکس در نظر میگیریم.

مشخص کردن کارت ملی یا کارت بانکی

- ◈ عکس خورشید در کارت ها را روی قسمت پایین چپ عکس template matching میزنیم و اگر امتیاز از ۰.۲ بیشتر باشد پس کارت ملی هست.
- ◈ بعد از بدست آمدن اینکه کارت ملی است یا بانکی، دیتاست را فارسی یا انگلیسی انتخاب میکنیم.
- ◈ اگر تعداد اعداد ۱۶ تا بود میفهمیم که شماره کارت است.
- ◈ اگر / داشت پس تاریخ انقضا است.
- ◈ اگر تعداد اعداد ۱۰ تا بود میفهمیم که شماره ی ملی است.