

به نام خدا

گزارش پروژه مبانی بینایی کامپیوتر

دانشجویان: دیار حامدی (۹۸۵۲۱۱۳۵) – فاطمه عسکری جیره‌نده (۹۸۴۷۱۴۱۴)

استاد: دکتر محمدرضا محمدی

تابستان ۱۴۰۲

مقدمه

هدف از این پروژه خواندن یک تصویر کارت ملی یا کارت بانکی و استخراج اطلاعات مهم آن است. این اطلاعات شامل موارد زیر می باشد:

- نوع کارت خوانده شده (کارت ملی یا کارت بانکی)
- مکان اعداد مهم در کارت (شماره ملی در کارت ملی و شماره کارت در کارت بانکی)
- زبان ارقام پیدا شده در کارت (انگلیسی یا فارسی)
- مقدار اعداد مهم به صورت رشته ای

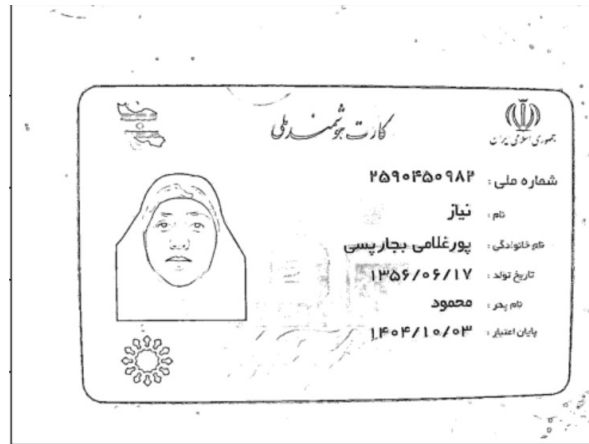
به جهت به دست آوردن این اطلاعات با دقت بالا، مراحل زیر به ترتیب طی میگردند:

۱. یافتن جعبه کارت در تصویر و برش زدن آن به یک مستطیل متعارف
 ۲. تشخیص نوع کارت
 ۳. پیدا کردن مکان اعداد با توجه به نوع کارت
 ۴. تشخیص زبان هرکدام از ارقام پیدا شده در کارت
 ۵. تشخیص مقدار هرکدام از ارقام پیدا شده با توجه به زبان آن رقم
- در ادامه راهکارهای امتحان شده برای هرکدام از این موارد به همراه راهکار انتخابی با جزییات شرح داده شده اند.

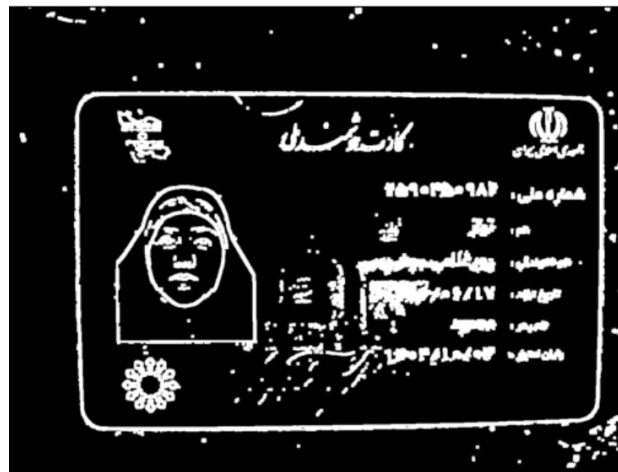
برش کارت:

در ابتدا تصمیم گرفته شد مانند تمرین سوم مبانی بینایی که یک camscanner ساده بود عمل کنیم اما به نتایج دلخواه دست نیافته و در ادامه ایده های مختلف از جمل تبدیل هاف، segmentation، تغییر فضای رنگی نیز امتحان شد که هیچ کدام انعطاف پذیری موردنظر برای شرایط تصویر را نداشتند.

ایده ای که در نهایت به کار برده شد به این صورت است که در ابتدا bilateralFilter اعمال شده تا از نویز های تصویر کاسته شود و سپس از adaptiveThreshold استفاده میشود که نمونه آن در زیر مشاهده میشود:



و بعد از آن تبدیل canny اعمال شد. نکته بسیار مهم این بود که بعد از canny یک dilate زدیم که باعث شد پهنای مستطیل کارت بیشتر بشود که در عملکرد مرحله بعدی تاثیر چشمگیری دارد. تصویر نمونه بعد از این مراحل به صورت زیر در می آید:



پس از آن کانتورهای تصویر به دست آمده و در یک for شکلی که تعداد گوشه های آن 4 تا باشد و بیشتر مساحت را داشته باشد را به دست آوردیم که کد آن مطابق زیر است:

```

perimeter = cv2.arcLength(c, True)
# Approximate the paper's contour with approxPolyDP function
approx = cv2.approxPolyDP(c, 0.02 * perimeter, True)
area = cv2.contourArea(c)
cornerpoints=[]
# If the contour has four sides, show it
if len(approx) ==4 and area>max_area:
    max_area = area
    max_contour = approx
print(len(max_contour))
if len(max_contour)>0:
    cv2.drawContours(img1, [max_contour], -1, (0, 255, 0), 2)
    for point in max_contour:
        x, y = point.ravel()
        cornerpoints.append([x,y])
cornerpoints=np.array(cornerpoints, dtype = 'object')
cornerpoints=np.float32(cornerpoints)

```

و تصویر ما بعد از این مرحله به صورت زیر شد:

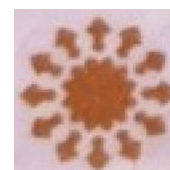


در ادامه، گوشه‌های چهارضلعی تصویر که پیدا کردیم، به عنوان نقاط ورودی برای تبدیل پرسپکتیو یا perspective transform انتخاب می‌شوند. سپس چهار نقطه جدید برای تصویر خروجی تعیین می‌شوند. این نقاط جدید معمولاً با استفاده از اندازه و ابعاد موردنظر برای خروجی تعیین می‌شوند. سپس با استفاده از تابع `getPerspectiveTransform` از OpenCV، ماتریس تبدیل پرسپکتیو برای تبدیل تصویر از یک منظر به دیگری تولید می‌شود. در نهایت، تصویر ورودی با استفاده از تابع `warpPerspective` به تصویر خروجی تبدیل می‌شود. تصویر نهایی به صورت زیر است:



تشخیص نوع کارت:

با توجه به اینکه در تمامی کارت های ملی علامت زیر موجود است از تشخیص کلیشه برای این قسمت استفاده کردیم که کلیشه مورد استفاده تصویر سیاه و سفید علامت زیر است:



الگو بالا با استفاده از تابع `matchTemplate` در تصویر نهایی مطابقت داده می شود. مقدار آستانه (`threshold`) برای نتیجه تعیین شده و مکان قالب در تصویر پیدا می شود. در صورت پیدا شدن قالب، کارت به "National card" تغییر می یابد. در غیر این صورت، نوع کارت به "Bank card" تعیین می شود.



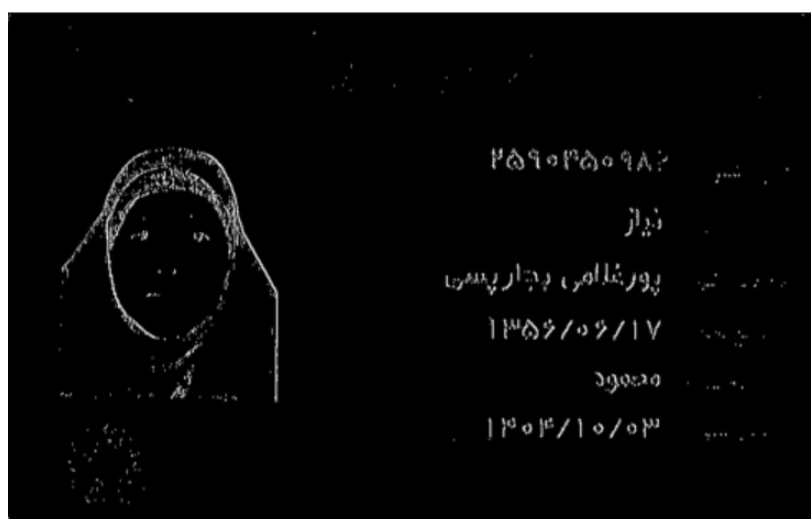
پارت سوم:

در ابتدا تشخیص ناحیه باکس اعداد را برای شماره ملی می گوییم:

ما می دانیم شماره ملی یکسری عدد هستند که پشت سر هم هستند و هم چنین اکثر اعداد دارای لبه های عمودی هستند. در ابتدا کیفیت عکسی که در مرحله اول برش به دست آورده را کمی بالاتر برده تا اعداد در آن واضح تر باشند. بدین منظور، در ابتدا با یک کرنل *sharpe* آن را کانولو می کنیم و در ادامه آن را به فضای رنگی HSV برده و روشنایی پیکسل ها را در آن فضا افزایش میدهیم. بعد از این مرحله عکس به صورت زیر است:

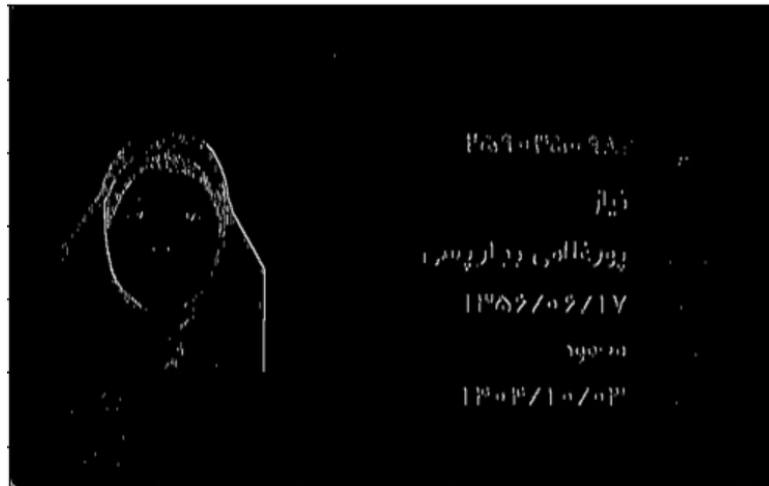


برای پیدا کردن ناحیه شماره ملی اول یک *threshold* و به دنبال آن سوپل افقی اعمال میکنیم. از نتیجه قدرمطلق میگیریم تا هم تغییر از سیاه به سفید و هم از سفید به سیاه را مثبت قلمداد کند. بعد از این مرحله عکس به صورت زیر میشود:

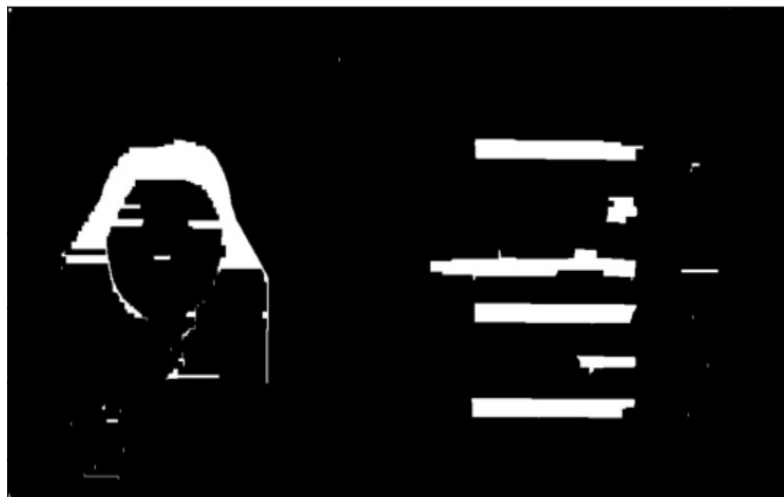


در ادامه عملگر *open* را انجام می دهیم که این نویز های تصویر حذف شوند:

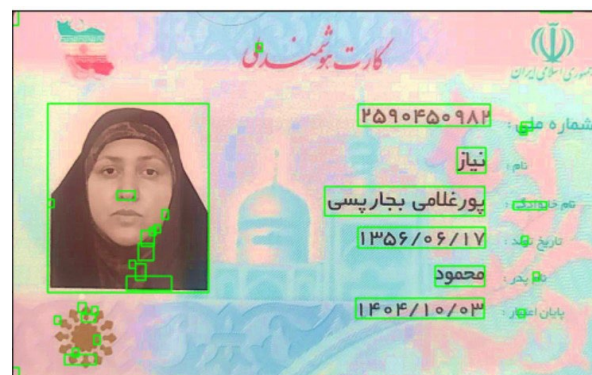
بعد از آن تصویر به صورت زیر میشود:



و در ادامه یک close میزنیم تا این ناحیه های سفید به هم بچسبند:



اگر بر روی تصویر اصلی باکس ها رو بکشیم به صورت زیر میشود:



در این مرحله میتوانیم ازجایی که شماره ملی قرار داده کنیم مثلاً ما می دانیم که شماره ملی در بالا سمت راست قرار دارد می توانیم باکس ها ان ها را در نظر بگیریم که به صورت زیر میشود:

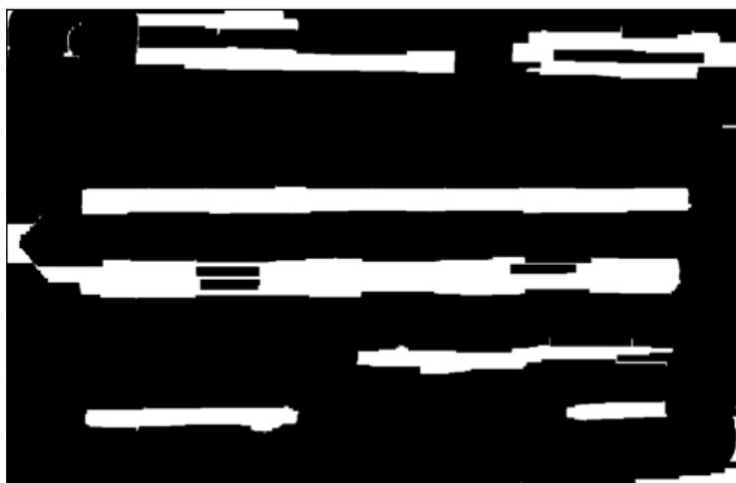


در ادامه مستطیل ها را براساس ۷ مرتب شده و اگر مستطیلی طولش از 0.8 طول بزرگترین مستطیل بیشتر بود ان را به عنوان کاندید شماره کارت ملی انتخاب میکنیم:



در ادامه پیدا کردن شماره کارت برای کارت بانکی را توضیح میدهم:

تقریبا همون مراحل بالا طی میکنیم که عکس ها به صورت زیر است:



که با توجه به پوزیشن که اکثر کارت بانکی ها دارند ناحیه شماره کارت را انتخاب می کنیم:



همانطور که در بالا مشاهده میکنید ناحیه های CVV2 و تاریخ انقضا نیز شناسایی شده اند.

ساخت دیتاست ارقام کارت ملی:

به منظور کسب حداکثر دقت در دو قسمت بعد پروژه (تشخیص زبان ارقام و پیشبینی مقدار آنها) باید از ابزارهای یادگیری عمیق (شبکه های عصبی کانولوشنی) استفاده کنیم. یکی از قسمتهای مهم در این رویکرد جمع آوری داده های مناسب می باشد. برای تشخیص زبان انگلیسی و نوع ارقام انگلیسی نیازمند دیتاست ارقام انگلیسی هستیم که در این خصوص دیتاست تشخیص ارقام MNIST استفاده شده است. برای تشخیص زبان

فارسی و نوع ارقام فارسی ابتدا دیتاست فارسی «هدی» امتحان شد، اما به دلیل نامطلوب بودن نتایج در ادامه تصمیم بر این گرفته شد که یک دیتاست ساختگی از ارقام نوشته شده در کارت ملی تولید و در مراحل بعدی مورد استفاده قرار گیرد.

به منظور تولید داده های هرچه نزدیکتر به داده های واقعی، قسمتی از پس زمینه کارت ملی برش زده شده و یکی از ارقام فارسی با فونت BYekan که فونت اصلی در کارت های ملی است بر روی قسمت برش زده شده نوشته میشود. با اعمال کمی جابجایی و چرخش بر روی این تصویر، کیفیت و تنوع داده ها ارتقا یافته و به عبارت دیگر data augmentation صورت میگیرد. تصویر حاصل شده سپس threshold خورده و در نهایت با احتمال کمی عملیات مورفولوژی erode, opening یا closing بر روی آن اعمال گشته تا نمونه داده به دست آید.

تشخیص زبان و مقدار ارقام:

در مرحله آموزش برای این قسمت تصاویر دیتاست MNIST و دیتاست ساختگی ارقام کارت ملی با همدیگر در یک دیتاست جدید ادغام شده که برچسپ های آن 0 (زبان انگلیسی) و 1 (زبان فارسی) می باشند. جهت هماهنگی ابعاد تصاویر دو دیتاست مورد استفاده، تصاویر دیتاست MNIST به اندازه تصاویر ارقام فارسی تغییر ابعاد میدهند.

پس از ساخت دیتاست تشخیص زبان یک مدل LeNet را با اندازه ورودی تصاویر این دیتاست و خروجی ۲ کلاس تعریف کرده، سپس آن را کامپایل کرده و در نهایت بر روی دیتاست تشخیص زبان آموزش دادیم. همچنین برای تشخیص مقدار ارقام انگلیسی و فارسی دو مدل LeNet دیگر را به ترتیب بر روی دیتاست MNIST و دیتاست ساختگی آموزش دادیم که نتایج آن در زیر مشاهده میشود:

```

=====] - 3s 4ms/step - loss: 0.0049 - accuracy: 0.9985 - val_loss: 0.0522 - val_accuracy: 0.9899
=====] - 3s 5ms/step - loss: 0.0019 - accuracy: 0.9994 - val_loss: 0.0611 - val_accuracy: 0.9895
=====] - 3s 5ms/step - loss: 0.0044 - accuracy: 0.9986 - val_loss: 0.0648 - val_accuracy: 0.9872
=====] - 3s 5ms/step - loss: 0.0035 - accuracy: 0.9990 - val_loss: 0.0515 - val_accuracy: 0.9907
=====] - 3s 4ms/step - loss: 0.0021 - accuracy: 0.9993 - val_loss: 0.0628 - val_accuracy: 0.9891
=====] - 3s 4ms/step - loss: 0.0056 - accuracy: 0.9983 - val_loss: 0.0527 - val_accuracy: 0.9903
=====] - 3s 6ms/step - loss: 0.0022 - accuracy: 0.9994 - val_loss: 0.0507 - val_accuracy: 0.9908
=====] - 3s 5ms/step - loss: 0.0022 - accuracy: 0.9991 - val_loss: 0.0833 - val_accuracy: 0.9892
=====] - 3s 5ms/step - loss: 0.0051 - accuracy: 0.9987 - val_loss: 0.0516 - val_accuracy: 0.9915
=====] - 3s 4ms/step - loss: 6.6736e-04 - accuracy: 0.9998 - val_loss: 0.0580 - val_accuracy: 0.9905

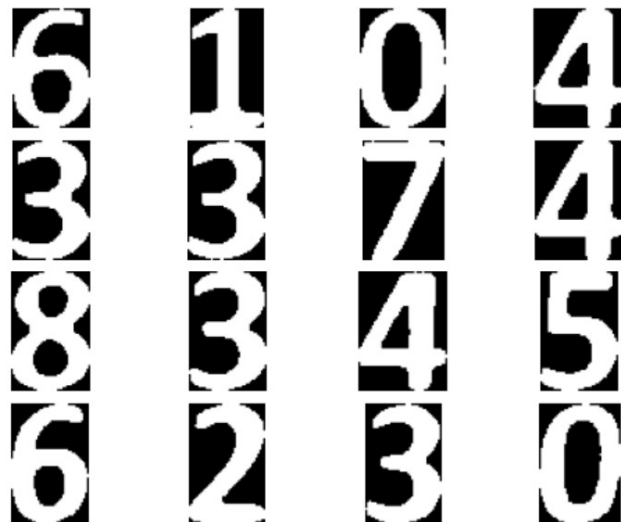
```

شکل ۱ - نتیجه آموزش برای تشخیص مقدار ارقام انگلیسی

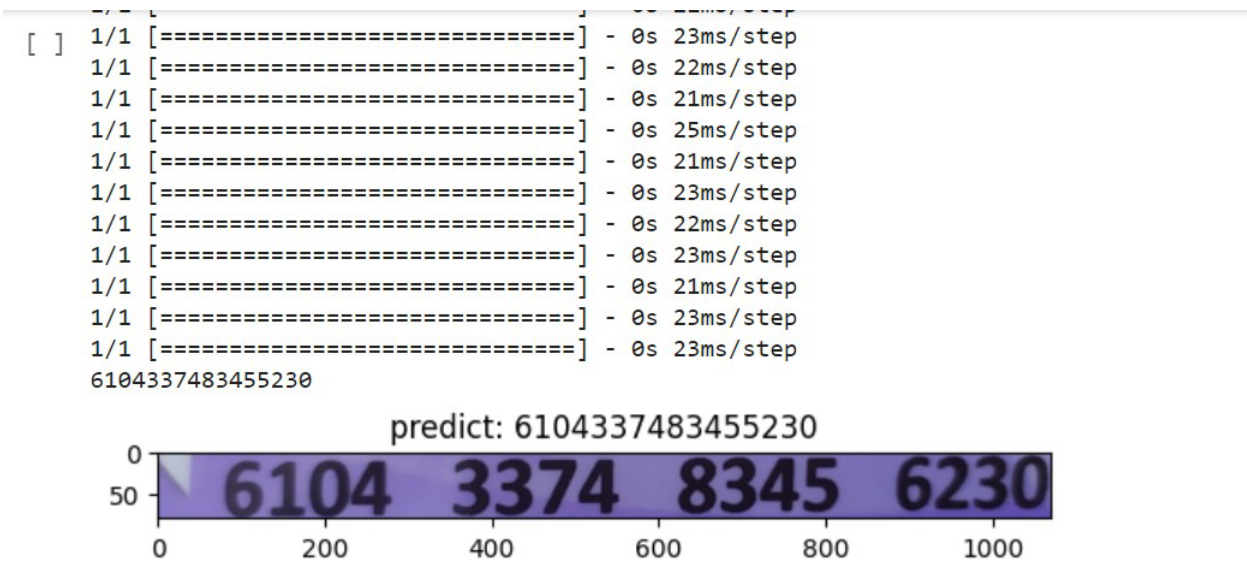
مشاهده میشود که در تمامی موارد، به دقت بسیار بالایی دست یافته ایم و مدل LeNet برای هر سه وظیفه مناسب بوده است.

اکنون که مدل تشخیص زبان آماده است، باید هر کدام از ارقام را از تصویر استخراج کرده و به صورت جداگانه به مدل ورودی دهیم. بدین منظور ابتدا یک **threshold** بر روی جعبه ارقام اعمال میکنیم، سپس جعبه ناحیه های مشکی متصل به همدیگر که همان ارقام هستند را با الگوریتم یک جز در هر زمان استخراج میکنیم و در آخر نواحی ارقام به دست آمده را براساس مختصات X مرتب سازی میکنیم:

6104 3374 8345 6230

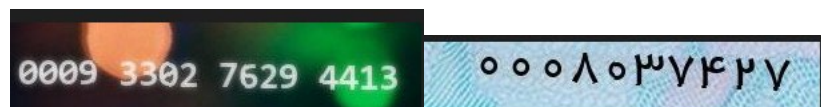


برای پیش بینی زبان و مقدار رقم، ابتدا تصویر را به اندازه ورودی مدل پیش بینی زبان تغییر اندازه داده، بر روی آن **threshold** اعمال کرده و به مدل ورودی می‌دهیم. براساس پیش‌بینی مدل، اندازه تصویر و مدل پیش‌بینی مقدار را مشخص کرده و تصویر را به مدل پیش بینی داده تا مقدار پیش بینی شده برای آن رقم به دست آید. با کنار هم قرار دادن تمامی ارقام پیش بینی شده، پیش بینی کلی ما از شماره موجود در تصویر حاصل می‌گردد. در این قسمت نتیجه پیش‌بینی مدل تشخیص مقدار ارقام انگلیسی را بر روی ارقام استخراج شده نشان می‌دهیم. نکته مهم اینکه به اعداد جدا شده باید **border** اضافه کنیم تا نتیجه خوبی بگیریم:

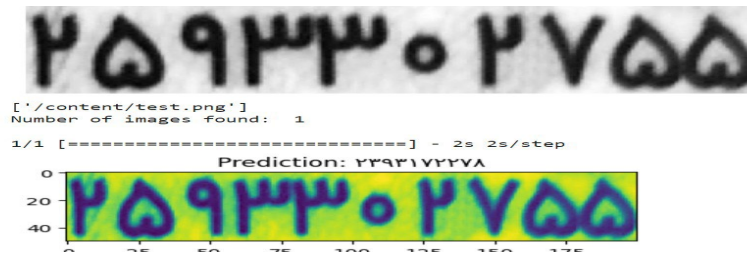


مشاهده می کنید که فقط یک رقم اشتباه گفته.

برای پیش بینی اعداد بر روی کارت ملی ابتدا از مدل <https://keras.io/examples/vision/captchaocr> استفاده کردیم که نیاز بود اول یک دیتاست خودمون **generate** کنیم که من 10000 داده از هر دو کارت ملی و کارت بانکی **generate** کردم که دو نمونه آن به صورت زیر است:



و نتیجه به صورت زیر شد:



برای اینکه بعضی از نکات را در درس نخوانده بودیم دو ایده را پیاده سازی کردیم:

1) از دیتاست هدی استفاده کردیم

2) به سراغ درست کردن دیتاست برای اعداد رفتیم چون فونت اعداد byekan هست.

ایده اول:

در ابتدا دیتاست هدی را با شبکه lenet train کردیم که نتایج آن بسیار خوب بود:

=====	- 32s 53ms/step	- loss: 0.0079	- accuracy: 0.9973	- val_loss: 0.0664	- val_accuracy: 0.9841
=====	- 30s 50ms/step	- loss: 0.0077	- accuracy: 0.9971	- val_loss: 0.0676	- val_accuracy: 0.9841
=====	- 30s 50ms/step	- loss: 0.0066	- accuracy: 0.9975	- val_loss: 0.0634	- val_accuracy: 0.9856
=====	- 30s 51ms/step	- loss: 0.0065	- accuracy: 0.9978	- val_loss: 0.0664	- val_accuracy: 0.9839
=====	- 30s 49ms/step	- loss: 0.0084	- accuracy: 0.9973	- val_loss: 0.0611	- val_accuracy: 0.9853
=====	- 30s 50ms/step	- loss: 0.0073	- accuracy: 0.9975	- val_loss: 0.0673	- val_accuracy: 0.9839
=====	- 32s 54ms/step	- loss: 0.0064	- accuracy: 0.9977	- val_loss: 0.0666	- val_accuracy: 0.9852
=====	- 30s 50ms/step	- loss: 0.0052	- accuracy: 0.9981	- val_loss: 0.0680	- val_accuracy: 0.9845
=====	- 30s 50ms/step	- loss: 0.0064	- accuracy: 0.9976	- val_loss: 0.0723	- val_accuracy: 0.9834
=====	- 30s 50ms/step	- loss: 0.0072	- accuracy: 0.9973	- val_loss: 0.0610	- val_accuracy: 0.9861

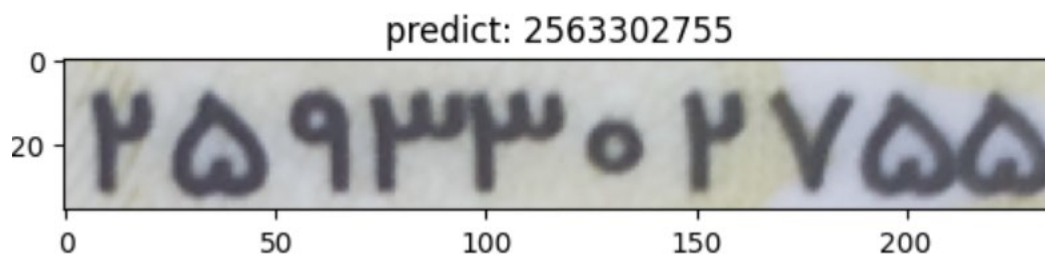
و در ادامه جدا کردن digit ها از آن باکس اعداد بود که از کارت بانکی پرچش تر بود به دلیل رقم ها کوچکتر هستند در ابتدا یک threshold می گذاریم که تصویر به صورت زیر میشود:



مشکلی که وجود دارد این است که خیلی اوقات همانطور که مشاهده میکنید اعداد 3 و 5 که پشت سر هم ظاهر میشوند به هم میچسبند اگر نیز یک erode بذاریم باعث تخریب خود اعداد میشود که در تشخیص آنها به مشکل میخوریم به همین علت ابتدا با استفاده از کانتور ناحیه ها را به دست میاورم وبعد چک میکنم اگر طول ناحیه ای از $\frac{1}{3}$ ناحیه باکس بیشتر بود تقسیم بر دو بکنه ان ناحیه را:



و بر روی مدل predict می کنیم و نتیجه به صورت زیر میشود:



راه دوم که دیتا generate کردیم: