

بسمه تعالی



دانشکده مهندسی کامپیوتر

یادگیری عمیق

نام استاد: دکتر محمدی

پروژه نهایی

آرمان حیدری – سید سینا ضیایی

شماره دانشجویی: ۹۷۵۲۱۲۵۲ –

بهمن ۱۴۰۰

## چکیده

در این پروژه می خواهیم با استفاده از تکنیک های OCR، که وظیفه تشخیص خودکار متون را دارد اعداد ۱۰ رقمی فارسی موجود روی کارت های ملی (قدیمی یا جدید) و همچنین اعداد انگلیسی ۱۶ رقمی روی کارت بانکی های مختلف را از تصاویر استخراج کنیم.

وظیفه تهیه دیتاست با خودمان بود و سعی کردیم دیتاستی تا حد امکان واقعی را تهیه کنیم و یک شبکه عصبی RCNN با تابع ضرر CTC را با آن آموزش دادیم تا این تسک را انجام دهد. طبیعتاً داده آموزشی و ارزیابی جدا از هم به این منظور در نظر گرفته ایم.

## روند کلی کار

چون تجربه کار زمینه OCR را نداشتیم، به جستجو در اینترنت پرداختیم و با استفاده از این [لینک](#) مسئله را بهتر متوجه شدیم. و توضیحات و کد این [لینک](#) را هم بررسی کردیم. سپس برای تصمیم گیری بهتر در زمینه مدل انتخابی در چنین مسائلی این [لینک](#) که کد خوبی برای تشخیص reCAPTCHA با استفاده از کراس بود را استفاده کردیم.

سپس وارد کد شدیم و در بستر گوگل کولب در section های مختلفی (برای واضح بودن مراحل) این مسئله را پیاده سازی کردیم. به طور کلی اول دیتاستی را ساختیم که برای ذخیره آن، گوگل درایو را mount کردیم تا هر بار مجبور به ایجاد دوباره دیتاست نشویم. در ساختن این دیتاست هم از این [منبع](#) استفاده کردیم.

بعد از preprocessing روی دیتاها، مدلی را ساختیم و آن را روی دیتای آموزشی fit کردیم. در تمام مراحل دیتا های مربوط به فارسی و انگلیسی shuffle بودند و یک مدل آن ها را یاد میگیرد. البته تفاوت طول خروجی را با قرار دادن ۶ کاراکتر space در انتهای کدهای ملی حل کردیم. و در آخر دیتای ارزیابی را بررسی کردیم و دقت مدل و روند کاهش loss آن را بررسی کردیم. این مراحل به تفصیل در ادامه توضیح داده شده اند.

سپس یک پیاده سازی از این مدل با استفاده از فریم ورک فلاتر داشتیم. و اپلیکیشن اندرویدی ساده شده ای از آن را ایجاد کردیم.

```

print("current_path:", current_path)

font_arial_url = 'https://github.com/sinaziaee/optical_digit_recognizer/raw/main/fonts/arial.ttf'
font_ocr_url = 'https://github.com/sinaziaee/optical_digit_recognizer/raw/main/fonts/ocr_a.ttf'
font_yekan_url = 'https://github.com/sinaziaee/optical_digit_recognizer/raw/main/fonts/yekan.ttf'
font_nazanin_url = 'https://github.com/sinaziaee/optical_digit_recognizer/raw/main/fonts/nazanin.ttf'
bank_base_url = 'https://github.com/sinaziaee/optical_digit_recognizer/raw/main/bank_background_images'
id_base_url = 'https://github.com/sinaziaee/optical_digit_recognizer/raw/main/id_background_images/'

!wget -P /content/fonts $font_arial_url -N -q
!wget -P /content/fonts $font_ocr_url -N -q
!wget -P /content/fonts $font_yekan_url -N -q
!wget -P /content/fonts $font_nazanin_url -N -q

for i in range(1, 34):
    img_path = f'{bank_base_url}/{i}.jpg'
    !wget -P /content/bank $img_path -N -q

for i in range(1, 3):
    img_path = f'{id_base_url}/id{i}.jpg'
    !wget -P /content/id $img_path -N -q

```

در اینجای کار سرغ منابع اولیه مانند فونت‌های مورد نیاز و عکس‌های بک گراندی که بعداً قرار هست برای تولید عکس به کار روند استفاده میکنیم.

```

# creating folders if they don't exist
if not os.path.isdir(f'/content/drive/MyDrive/{drive_path}/'):
    os.mkdir(f'/content/drive/MyDrive/{drive_path}/')

if not os.path.isdir(f'/content/drive/MyDrive/{drive_path}/bank/'):
    os.mkdir(f'/content/drive/MyDrive/{drive_path}/bank/')

if not os.path.isdir(f'/content/drive/MyDrive/{drive_path}/id/'):
    os.mkdir(f'/content/drive/MyDrive/{drive_path}/id/')

if not os.path.isdir(f'{bank_bg_base_path}/'):
    os.mkdir(f'{bank_bg_base_path}/')

if not os.path.isdir(id_bg_base_path):
    os.mkdir(id_bg_base_path)

if not os.path.isdir(id_base_path):
    os.mkdir(id_base_path)

if not os.path.isdir(bank_base_path):
    os.mkdir(bank_base_path)

```

در اینجای کار ما به دنبال ساخت پوشه هایی در google drive هستیم که بعداً قرار است عکس‌ها در آن‌ها ذخیره شوند.

```
# cropping images with the help of opencv and cutting
for i, file in enumerate(files):
    img = cv.imread(files[i])
    h, w, c = img.shape
    new = cv.cvtColor(img, cv.COLOR_BGR2RGB)
    if w > 200:
        h_sec = int(h/7)
        w_sec = int(w/6)
        new = img[3*h_sec:6*h_sec, w_sec:5*w_sec,]
    else:
        h_sec = int(h/14)
        w_sec = int(w/12)
        new = img[3*h_sec:12*h_sec, w_sec:12*w_sec,]
    try:
        cv.imwrite(f'{bank_bg_base_path}/{i}.jpg', new)
    except Exception as e:
        pass
print(len(files))
```

در این بخش نیز به کمک کتابخانه ی opencv عکس‌ها را لود کردیم و متناسب با اینکه عرض عکس از 200 پیکسل کوچکتر هست یا خیر عکس را کراپ کرده‌ایم تا به بخشی که معمولاً شماره های کارت های بانکی قرار میگیرند برسیم.

```
random_bank_num = 10
random_id_num = 100
```

این دو متغیر رندم نیز تعداد دفعاتی هستند که قرار است در حلقه ی loop مربوط به هر دیتاست عدد random تولید کنیم (التبه تعدادشان در اندازه ی loop های دیگر نیز ضرب می شود).

حال می‌رویم سراغ تولید دیتاست های مورد نیاز. مراحل کد های تولید عکس‌ها و دیتاست در کد کامنت گذاری نیز شده است.

## کارت های بانکی)

از ترکیبی از دو فونت انگلیسی و ۳۳ عکس بک گراند واقعی کارت های بانکی با استفاده از یک عدد رندم استفاده شده. همچنین اعداد روی کارت ها با دو رنگ سفید و سیاه نوشته شده اند که هر دو حالت را ساپورت کنند. پس از این مرحله با یک حلقه ی `for loop` روی داده های هر بخش یک عدد رندم ایجاد بین ۱ تا ۱۰۰ ایجاد میکنیم که اگر از ۴۰ این عدد کمتر باشد عکس را به صورت رندم `rotate` و اگر بین ۲۰ و ۶۰ باشد `blur` میکنیم و این داده های اضافی شده را به دیتاست داده های اولیه میدهیم.

در آخر نیز حدود ۱۹ عکس از کارت های بانکی واقعی کراپ شده را به دیتاست اضافی میکنیم.

## کارت های ملی)

در اینجا نیز از ترکیبی از دو فونت فارسی و ۵ عکس پس زمینه ی واقعی کارت های ملی استفاده میکنیم ( همه در `for loop` ها قرار دارند) و به صورت رندم کارت های ملی را تولید میکنیم. سپس مرحله ی چرخش و `blur` را مانند بالا روی این عکس ها هم به صورت رندم اجرا میکنیم.

## تصاویر واقعی)

تعدادی تصویر واقعی هم در درایو ذخیره کردیم و از آن ها هم برای آموزش مدل استفاده کردیم تا به نتایج بهتری برسیم.

## پیش پردازش داده ها

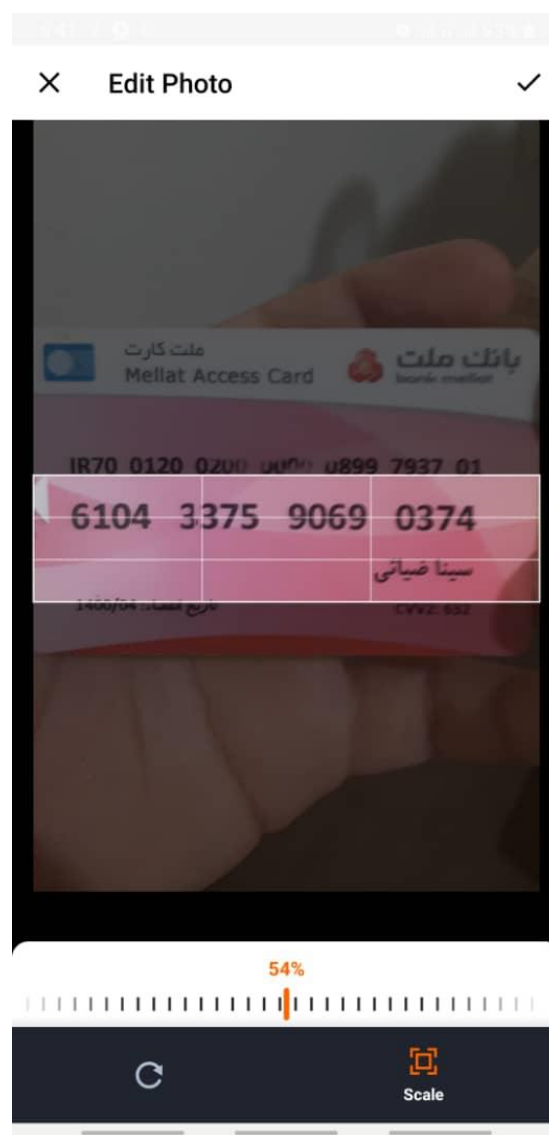
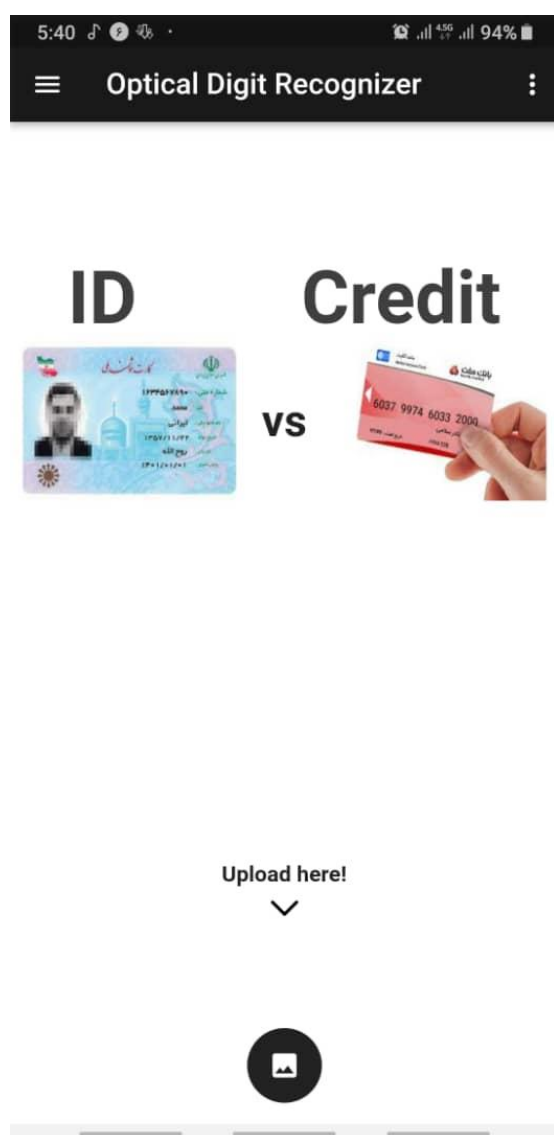
## تعریف مدل و آموزش آن



## ارزیابی مدل و نتایج

## اپلیکشن اندروید

نکته ی قابل توجه ای نیز وجود دارد که بعد از آموزش مدل و خروجی گرفتن از آن کدی مربوط به پیاده سازی مدل در برنامه ی اپلیکشن اندرویدی نیز زده شده که با خروجی عکس گرفتن و یا انتخاب عکس از گالری میتوان به عنوان ورودی مدل داده و پیشبینی کرد که عدد ما کارت بانکی یا کارت شناسایی است و چه عددی داخلش وجود دارد.



که کد مربوط به اپلیکشن اندرویدی در odr\_android ضمیمه می شود.