

## به نام خدا

توضیح روند انجام پروژه تشخیص پلاک های ایرانی:

**مرحله الف:** تشخیص کادر پلاک ها و برش آن ها

این مرحله در گوگل کولب و توسط gpu انجام شده است. مراحل انجام کار به شکل زیر است.

شرح کد yolov7DetectPlate.ipynb :

- 1- فراخوانی گوگل درایو و دانلود مدل yolov7 و کتابخانه های مورد نیاز آن در این محیط.
- 2- ادغام 2 مجموعه داده ی آماده از پلاک ماشین های ایرانی و خارجی و فایل های ضمیمه ی آن ها در Roboflow و همچنین انجام data augmentation و تقسیم به سه گروه train، validation و test در این سایت.
- 3- فراخوانی مجموعه ی عکس های ساخته شده در مرحله دوم و دانلود آن در گوگل کولب.
- 4- دانلود وزن اولیه برای مدل yolov7.
- 5- آموزش مدل yolov7 با وزن های مرحله 4 و داده های فولدر train مرحله 3 با تعداد اپیک و batch-size موردنظر.
- 6- فراخوانی تابع detect\_and\_acrop و اعمال این تابع به کمک وزن های به دست آمده از مرحله 5 و مجموعه عکس های تست به دست آمده از مرحله 3.
- 7- دانلود عکس های کراپ شده ی پلاک ماشین ها و استفاده در مراحل بعدی.

**مرحله ب:** آموزش مدل برای تشخیص اعداد و حروف

جهت تشخیص اعداد و حروف پلاک های تشخیص داده شده در قسمت قبل، 2 مدل مجزا آموزش داده شد: یک مدل برای تشخیص حروف فارسی و یک مدل برای تشخیص اعداد فارسی.

کلیات 2 مدل کاملاً یکسان هستند و فقط در تعداد کلاس ها ( برای حروف 32 کلاس و برای اعداد 10 کلاس ) و مجموعه ی داده ورودی متفاوت هستند. در ادامه کلیات کد نوشته شده آورده شده است.

برای مجموعه ی داده از داده های زیر استفاده شده است. ابتدا آن ها را در گوگل درایو سیو کرده و سپس از آنجا دانلود شده است. و در پایان اعداد موجود در هر فولدر از حروف جدا شده اند و در فولدر جداگانه ای ریخته شده اند.

```
from roboflow import Roboflow
rf = Roboflow(api_key="VdwBRxsJsFmCMW94xJjd")
project = rf.workspace("object-detection-yolov5").project("plate_ocr_ir")
dataset = project.version(2).download("folder")
```

شرح کدهای detect\_persian\_numbers.py و detect\_persian\_characters.py :

- 1- کتابخانه های لازم فراخوانی شده اند.
- 2- برای کنترل نتایج در هر بار run کردن مدل، برای توابع استفاده شده مقدار seed مشخص شده است.
- 3- متغیرهای تعداد batch تعریف شده اند.
- 4- از gpu سیستم برای یادگیری مدل و گرفتن نتایج نهایی استفاده شده است.
- 5- در هنگام فراخوانی عکس ها سایز آن ها به 90 در 90 تغییر می کند.
- 6- نکته: این عدد پس از امتحان کردن چند سایز مختلف به عنوان بهترین سایز ممکن انتخاب شده است.
- 7- با کمک چرخش و بلور کردن تصاویر، مجموعه ی کل داد ها 3 برابر شدند. (data augmentation)
- از مدل densenet121(pretrained=True) برای آموزش استفاده شده است.
- نکته: تعداد کلاس های لایه ی classifier طبق توضیحات بالا تغییر کرد.

8- از بهینه یاب Adam برای آموزش و تابع CrossEntropyLoss برای پیدا کردن اختلاف لیبیل ها و خروجی مدل استفاده شده است.

9- مدل در هر ایپاک آموزش میبیند و نتیجه ی این یادگیری روی داده های validation اعمال می شود. اگر دقت خروجی داده های validation نسبت به ایپاک های قبلی بهتر باشد وزن مدل آموزش دیده در این ایپاک سیو می شود.

10- در انتها بهترین وزن به مدل داده می شود و مدل حاصل بر مجموعه ی تست اعمال می شود.

نکته: برای هر سه نوع داده و برای تمام ایپاک ها، confusion matrix رسم می شود و دقت و مقدار loss هم ذکر می شود.

نکته: هر مدل به همراه بهترین وزن ها در فایل های Augdensenet-numbers90.pth و

Augdensenet-characters90.pth سیو شده است.

نکته: در مدل مربوط به آموزش حروف دقت روی مجموعه ی تست و بهترین دقت مجموعه ی validation به ترتیب برابر با 99.12 و 98.97 می باشد و کل فرآیند آموزش و اعمال روی مجموعه ی تست برای 20 ایپاک در مدت 42 دقیقه انجام می شود.

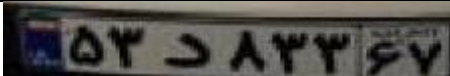

نکته: در مدل مربوط به آموزش اعداد دقت روی مجموعه ی تست و بهترین دقت مجموعه ی validation به ترتیب برابر با 100 و 100 می باشد و کل فرآیند آموزش و اعمال روی مجموعه ی تست برای 20 ایپاک در مدت 14 دقیقه انجام می شود.

**مرحله ج:** تشخیص پلاک های به دست آمده از مرحله اول به کمک مدل های به دست آمده از مرحله ی دوم

شرح کد detect\_persian\_plates.py:

- 1- 3 تابع برای پیدا کردن بلندترین خط در عکس بریده شده ی (کراپ شده) پلاک ماشین، پیدا کردن زاویه ای که این بلندترین خط با سطح افق می سازد و چرخاندن عکس به اندازه ی این زاویه تعریف می کنیم.
- 2- مدل سیو شده در مرحله ی "ب" فراخوانی می شوند و در حالت model.eval() قرار می گیرند.
- 3- عکس مورد نظر فراخوانی می شود، سائز آن به سائز مناسب تغییر داده می شود و به کمک 3 تابع معرفی شده در مرحله 1 چرخش انجام می شود.
- 4- لیستی از 8 نقطه مورد نظر برای کراپ عکس و جداشدن کاراکترها از هم معرفی می شود.
- 5- در ادامه هر یک از کاراکترها به سائز مناسب مدل برگردانده می شوند و بعد و type آن ها مناسب مدل های معرفی شده در مرحله ی "ب" می شوند.
- 6- در صورتی که کاراکتر جدا شده ی سوم پلاک حرف باشد به مدل تشخیص حرف داده می شود در غیر این صورت به مدل تشخیص اسم داده می شود.
- 7- و در انتها پلاک تشخیص داده شده پرینت می شود.

در ادامه چند پلاک تشخیص داده شده توسط مدل مرحله "الف" و نوشته ی آن که توسط کد مرحله "ج" تشخیص داده شده اند قرار داده می شود.

	
53 d 8 3 3 6 7	7 3 ta 1 5 1 8 2

	
6 2 ghaf 1 9 3 7 9	6 6 b 3 1 9 6 2

به طور مشخص یکی از مشکلات فعلی چرخش پلاک کراپ شده است. که امتحان کردن روش های دیگری برای تعیین نقاط کراپ می توان بخش زیادی از مشکل را رفع کرد. همچنین بعد از چرخش، کاراکتر ها مقداری کشیده می شوند که با قرار دادن این حالت ها به عنوان داده ی مرحله ی "ب" نیز می توان نتایج را ارتقا داد.

نکته: ممکن است با تغییر ایپاک ها، batch-size ، loss function ، data- augmentation های دیگر، بهینه یاب و نرخ آن، تغییر مدل و افزایش داده ها در مرحله ی "الف" و "ب" به دقت نهایی بهتری رسید. البته تعدادی از این حالت ها امتحان شده است و نتایج ارائه شده بهترین نتایج بین تمام مدل های امتحان شده می باشد.