عنوان پروژه: استفاده از مدل YoloV7 برای تشخیص پلاک خودرو ایرانی

- آشنایی با مدل YoloV7
- استفاده از مدل pre-train شده ی YoloV7 و train کردن آن روی پلاک های ایرانی
- استفاده از مدل YoloV7 ترین شده برای تشخیص پلاک ایرانی و متن آن در عکس، ویدئو و دوربین وبکم (به صورت زنده)

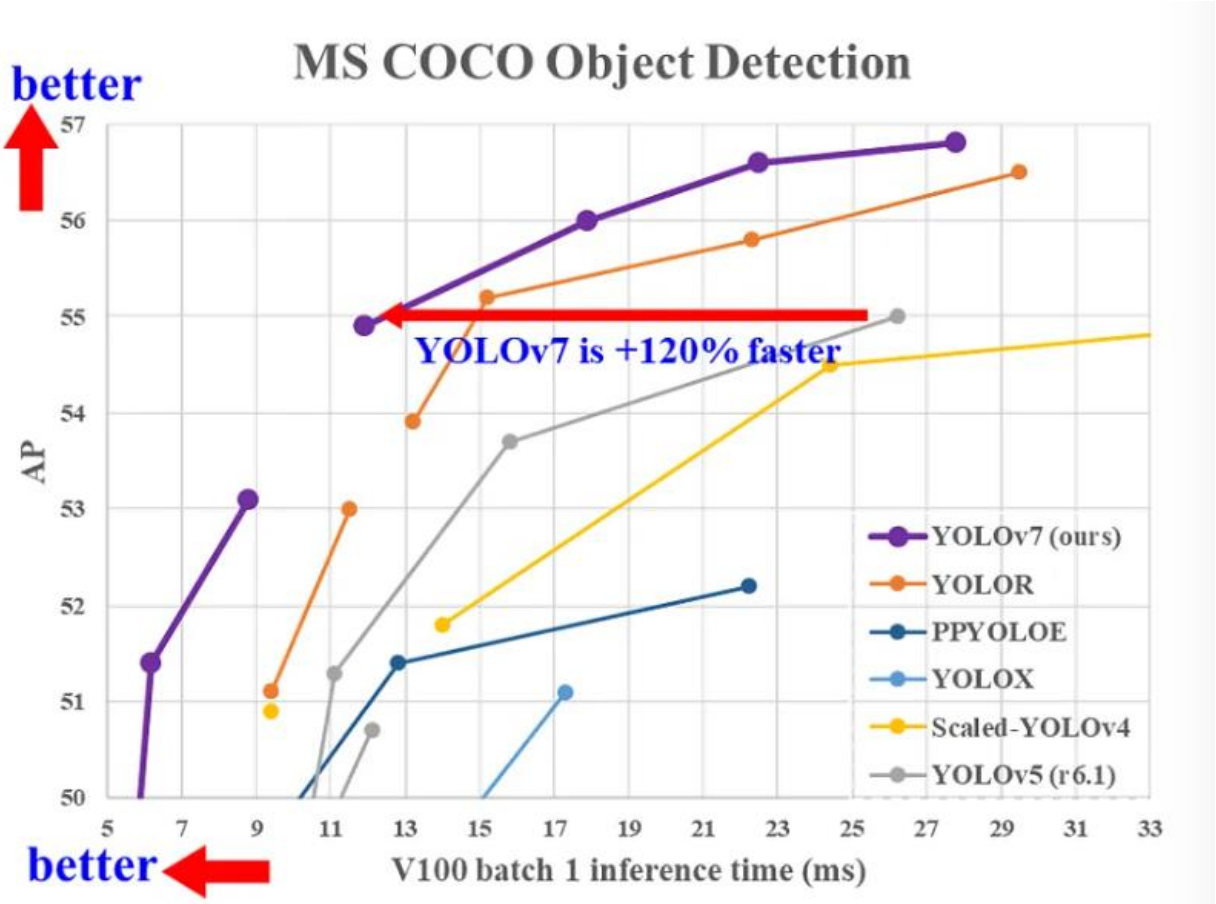
دو بخش اصلی پروژه:

۱- تشخیص پلاک (Object Detection)

۲- تبدیل عکس پلاک به متن برای نمایش شماره ی پلاک (OCR)

آشنایی با مدل YoloV7

یکی از جدیدترین و بهترین مدل ها چه از لحاظ سرعت و چه از لحاظ دقت (YoloV8 به تازگی منتشر شده است! و عملکرد بهتری از YoloV7 دارد) برای object detection است.



نکته ی جالب این هست که با وجود دقت بهتر، تعداد پارامترهای YoloV7 از برخی نسخه های قبلی کمتر می باشد.

Model	#Param.	FLOPs	Size	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}
YOLOv4 [3]	64.4M	142.8G	640	49.7%	68.2%	54.3%	32.9%	54.8%	63.7%
YOLOR-u5 (r6.1) [81]	46.5M	109.1G	640	50.2%	68.7%	54.6%	33.2%	55.5%	63.7%
YOLOv4-CSP [79]	52.9M	120.4G	640	50.3%	68.6%	54.9%	34.2%	55.6%	65.1%
YOLOR-CSP [81]	52.9M	120.4G	640	50.8%	69.5%	55.3%	33.7%	56.0%	65.4%
YOLOv7	36.9M	104.7G	640	**51.2%**	**69.7%**	**55.5%**	**35.2%**	**56.0%**	**66.7%**
improvement	-43%	-15%	-	+0.4	+0.2	+0.2	+1.5	=	+1.3
YOLOR-CSP-X [81]	96.9M	226.8G	640	52.7%	**71.3%**	57.4%	36.3%	57.5%	68.3%
YOLOv7-X	71.3M	189.9G	640	**52.9%**	71.1%	**57.5%**	**36.9%**	**57.7%**	**68.6%**
improvement	-36%	-19%	-	+0.2	-0.2	+0.1	+0.6	+0.2	+0.3
YOLOv4-tiny [79]	6.1	6.9	416	24.9%	42.1%	25.7%	8.7%	28.4%	39.2%
YOLOv7-tiny	6.2	5.8	416	**35.2%**	**52.8%**	**37.3%**	**15.7%**	**38.0%**	**53.4%**
improvement	+2%	-19%	-	+10.3	+10.7	+11.6	+7.0	+9.6	+14.2
YOLOv4-tiny-3l [79]	8.7	5.2	320	30.8%	47.3%	32.2%	**10.9%**	31.9%	51.5%
YOLOv7-tiny	6.2	3.5	320	**30.8%**	**47.3%**	**32.2%**	10.0%	**31.9%**	**52.2%**
improvement	-39%	-49%	-	=	=	=	-0.9	=	+0.7
YOLOR-E6 [81]	115.8M	683.2G	1280	55.7%	73.2%	60.7%	40.1%	**60.4%**	69.2%
YOLOv7-E6	97.2M	515.2G	1280	**55.9%**	**73.5%**	**61.1%**	**40.6%**	60.3%	**70.0%**
improvement	-19%	-33%	-	+0.2	+0.3	+0.4	+0.5	-0.1	+0.8
YOLOR-D6 [81]	151.7M	935.6G	1280	56.1%	73.9%	61.2%	**42.4%**	60.5%	69.9%
YOLOv7-D6	154.7M	806.8G	1280	56.3%	73.8%	61.4%	41.3%	60.6%	70.1%
YOLOv7-E6E	151.7M	843.2G	1280	**56.8%**	**74.4%**	**62.1%**	40.8%	**62.1%**	**70.6%**
improvement	=	-11%	-	+0.7	+0.5	+0.9	-1.6	+1.6	+0.7

مقایسه ی YoloV7, YoloV4

در مقایسه با YoloV4، YoloV7 ۷۵ درصد تعداد پارامترهای کمتر و ۳۶ درصد نیاز محاسباتی کمتری دارد و ۱.۵ درصد میانگین دقت بالاتری دارد. در مدل های بسیار بهینه شده این دو مدل YoloV7-tiny ۳۹ درصد تعداد پارامترهای کمتر و ۴۹ درصد نیاز محساباتی کمتری دارد در حالی که به یک میانگین دقت با YoloV4-tiny میرسد.

مقایسه ی YoloV7, YoloV6

در مقایسه با YoloV6، YoloV7 ۱۳.۷ درصد میانگین دقت بالاتری روی دیتاست COCO دارد. در مدل های بسیار بهینه شده این دو مدل روی سخت افزار و شرایط مشابه (V100 GPU, batch=32) YoloV7-tiny ۲۵ درصد سریع تر است و ۰.۲ درصد میانگین دقت بالاتری نسبت به YoloV6-n روی دیتاست COCO دارد.

استفاده از مدل pre-train شده ی YoloV7 و train کردن آن روی پلاک های ایرانی

برای این کار از مدل pretrain شده ی مدل YoloV7 استفاده کرده و با استفاده از Transfer Learning مدل را روی پلاک های ایرانی ترین میکنیم.

یک دیتاست از عکس ماشین ها به همراه پلاک آن ها و لیبل آن ها نیاز داریم تا مدل pre-train شده ی YoloV7 را روی آن ترین کنیم.

در این پروژه از ۲ دیتاست یکی عکس ماشین های خارجی به همراه پلاک آن ها (حدودا ۴۳۳ عکس) و دیگری عکس ماشین های ایرانی به همراه پلاک آن ها (حدودا ۴۴۴ عکس) استفاده شده است:

1- https://www.kaggle.com/datasets/andrewmvd/car-plate-detection
2- https://www.kaggle.com/datasets/skhalili/iraniancarnumberplate

حال میخواهیم این دو دیتاست را با هم ترکیب کرده و برای ترین کردن مدل استفاده کنیم.

برای آماده سازی دیتاست از سایت roboflow استفاده میکنیم که یک ابزار مناسب برای annotate کردن و آماده سازی داده ها به فرمت مناسب با YoloV7 در اختیار ما قرار میدهد.

app.roboflow.com

به همراه عکس هر ماشین فایل annotation آن با فرمت .xml نیز قرار دارد که ابزار Robo Flow به طور خودکار آن را تشخیص داده و دیتاست را آماده میکند.

در حقیقت استفاده ما از این ابزار برای این است که فایل annotation موجود را به فرمت YoloV7 درآورده و دو دیتاست را به نحو مناسب با هم ترکیب کند.

۱۰ درصد دیتاست را برای تست، ۲۰ درصد برای ولیدیشن و ۷۰ درصد برای ترین اختصاص میدهیم.

با این ابزار میتوان پیش پردازش های مختلف مانند data augmentation نیز روی تصویر انجام داد.

در این پروژه به منظور data augmentation از روش های Horizontal Flip و Shear استفاده شده است.

به هنگام Export کردن باید Format را روی YoloV7 بگذاریم و گزینه ی show download code را بزنیم تا api_key تولید شود.

حال باید از مدل pre-train شده ی YoloV7 که روی دیتاست COCO از قبل ترین شده است استفاده کنیم با اینکه دیتاست COCO کلاس پلاک ماشین ندارد اما میتوان با استفاده از Transfer Learning مدل را روی دیتاست آماده شده ترین (fine tune) کرد.

پیشنهاد میشود در صورت عدم دسترسی به GPU برای اجرای کد از GoogleColab استفاده شود چراکه GPU خوبی در اختیار قرار میدهد و مدت زمان ترین شدن مدل را کاهش میدهد.

برای این کار ابتدا گیت هاب YoloV7 (https://github.com/augmentedstartups/yolov7.git) را کلون کرده و سپس فولدر آن را به گوگل درایو انتقال میدهیم.

سپس با استفاده از فایل requirements.txt تمام کتابخانه های لازم برای اجرای کد را نصب میکنیم.

برای استفاده از دیتاست تولید شده در roboflow باید کتابخانه ی آن نیز با دستور زیر نصب شود.

```
!pip install roboflow
```

حال از api_key تولید شده استفاده کرده و از آن برای آپلود دیتاست در کولب استفاده میکنیم.
دیتاست ما توسط roboflow در فولدر ANPR_IR-1 در مسیر اصلی پروژه قرار داده شده است.

نکته: اگر میخواهید از دیتاست استفاده شده در این پروژه به صورت مستقیم استفاده کنید باید فولدر ANPR_IR-1 موجود در dataset را پس از clone کردن گیت هاب YoloV7 در مسیر اصلی آن قرار دهید.

حال با استفاده از script موجود در گیت هاب YoloV7 با نام train.py مدل را روی دیتاست جدید ترین میکنیم.
نکته مهم: به هنگام اجرای اسکریپت train.py به خطای Indices should be either on cpu or on the same device as the indexed tensor برخورد کردیم که با اعمال تغییرات زیر خطا رفع شد:

1- You can fix it by changing utils/loss.py line 685 to:
from_which_layer.append((torch.ones(size=(len(b),)) * i).to('cuda'))

2- Also add a line after 756 to put fg_mask_inboxes on your cuda device:
fg_mask_inboxes = fg_mask_inboxes.to(torch.device('cuda'))

به هنگام اجرای اسکریپت ساختار مدل و یکسری هایپرپارامترها نشان داده میشود.
مدت زمان ترین شدن مدل: حدودا ۱.۵ ساعت
آخرین و بهترین وزن های مدل در انتها ذخیره شده است که میتوان از آن برای prediction استفاده کرد.

حال برای پیش بینی مدل روی داده های تست از دیتاست موجود در مسیر dataset/ANPR_IR-1/test استفاده میکنیم.

برای شناسایی از اسکریپت detect.py موجود در گیت هاب YoloV7 استفاده میکنیم به این صورت که بهترین وزن ها را از مسیر runs/train/exp/weights/best.pt (همچنین در گیت هاب پروژه فولدر weights بهترین وزن موجود می باشد.) آپلود کرده و روی تصاویر موجود در فولدر test کار پیش بینی را انجام میدهیم.
نتیجه ی نهایی شناسایی پلاک های دیتاست تست در مسیر runs/detect/exp (همچنین در گیت هاب پروژه نتایج در مسیر results/detect/Licence_Plate_Detected قرار دارد.) قرار گرفته است.

همانطور که مشاهده میشود اکثر پلاک ها را با confidence score بالای ۹۰ درصد به درستی تشخیص داده است.

از مسیر runs/train/exp/weights/best.pt میتوان بهترین وزنهای مدل را دانلود کرده و در ادامه ی مسیر از آن استفاده کرد.

استفاده از مدل YoloV7 ترین شده برای تشخیص پلاک ایرانی و متن آن در عکس، ویدئو و دوربین webcam (به صورت زنده)

برای اجرای plate detection و تشخیص متن آن از local machine استفاده میکنیم چراکه کولب برای نمایش برخی محتواهای بصری از جمله ویدئو مکان مناسبی نیست.

ابتدا با استفاده از conda یک env درست میکنیم سپس گیت هاب YoloV7 را کلون کرده و وارد فولدر آن میشویم.

فونت Yekan (موجو در فولدر font) را در فولدر utils و بهترین وزن را در فولدر weights (خودمان این فولدر را میسازیم) قرار میدهیم.

پس از نصب کتابخانه های موجود در requirements.txt (کتابخانه های مورد نیاز YoloV7) باید کتابخانه های easyocr, deepsort را نیز به منظور object tracking و OCR نصب کنیم.

فایل Yolov7_IRPD_OCR.py را درون فولدر yolov7 (گیت هاب YoloV7) قرار میدهیم.

در این اسکریپت ابتدا باید مسیر عکس ها و فیلم هایی که میخواهیم شناسایی کنیم و مسیری که فایل های دیتکت شده میخواهیم در آن قرار بگیرند را مشخص کنیم.

سپس مدل را load میکنیم و بعد از آن reader را با easyocr تشکیل میدهیم.

تابع detect_plate عکس یا هر فریم از فیلم را میگیرد آن را دیتکت میکند و مختصات پلاک و confidence score آن را برمیگرداند.

تابع ocr_plate برای شناسایی متن هر پلاک، قسمت کراپ شده ی پلاک را گرفته و با استفاده از reader، متن را شناسایی میکند.

۱- عکس:

در تابع get_plates_from_image هدف جدا کردن قسمت مربوط به پلاک می باشد.

برای نشان دادن متن پلاک شناسایی شده باید از plot_one_box_PIL استفاده کنیم تا با متن فارسی همخوانی داشته باشد از طرف دیگر باید مسیر فونت یکان را به این تابع بدهیم برای این کار باید مسیر فونت را در قسمت font از تابع plot_one_box_PIL در فایل utils/plots.py کپی کنیم.

نتیجه شناسایی متن با استفاده از OCR خوب است اما به وضوح easyocr در شناسایی متن از روی عکس در محتوای فارسی ضعیفتر از محتوای انگلیسی عمل میکند.

میبینیم که به درستی محدوده ی پلاک را تشخیص داده است و در تبدیل آن به متن نیز چیزی را غلط تشخیص نداده است اما از ۸ حرف ۳ حرف را جاگذاشته است.

۲- ویدیو:

برای تشخیص پلاک در ویدیو و وبکم علاوه بر object detection باید object tracking (با استفاده از deepsort) نیز استفاده شود.

با دستور cv.VideoCapture فیلم را میخوانیم بعد طول و عرض و همچنین تعداد فریم بر ثانیه را اندازه میگیریم.

با استفاده از deepsort یک آبجکت tracker تعریف میکنیم.

حال تک به تک فریم های ویدیو را میخوانیم. هر فریم را به تابع detect_plate میدهیم تا پلاک را تشخیص داده و هر یک را track میکنیم و به هر پلاک یک track_id اختصاص میدهیم و سپس با ocr_plate متن آن را میخوانیم.

در خروجی track_id, text, confidence score نوشته میشود.

۳- وبکم:

کد مربوط به این بخش نیز کاملا یکسان با کد ویدیو میباشد.

برای افزایش سرعت شناسایی پلاک ها میتوان به جای آنکه پلاک را برای هر فریم شناسایی کند، هر ۱۰ فریم کار شناسایی پلاک توسط مدل انجام شود.