پروژه تشخیص خودرو های ایرانی

مریم واقعی[[1]](#footnote-1)

مریم صابری

|  |  |  |
| --- | --- | --- |
| اطلاعات گزارش |  | چکیده |
| **تاریخ:** |  | در اين گزارش‏‌ قرار است پروژه ی خود در بحث تشخیص خودرو های ایرانی به همراه مدل های آنها را توضیح دهیم. ما در این گزارش روند کار خود و همچنین نتایج خود را شرح داده ایم. |
| **واژگان كليدي:** |  |

فهرست مطالب

[مقدمه 2](#_Toc105617906)

[بخش اول: جمع آوری مجموعه داده 3](#_Toc105617907)

[بخش دوم: پیش پردازش داده ها 3](#_Toc105617908)

[حذف داده های پرت 3](#_Toc105617909)

[برچسب زدن عکس ها 4](#_Toc105617910)

# مقدمه

اکنون با پیشرفت فناوری و ورود به عصر تکنولوژی و هوش مصنوعی، بشر در مسیر خودکارسازی بسیاری از کار ها من جمله ربات های صنعتی، خودرو های خودران، سیستم های تشخیص چهره و... قرار گرفته است. بشر سعی میکند تا با استفاده از هوش مصنوعی و یادگیری عمیق سیستم هایی با دقت بالا و ایمن برای انسان ها طراحی کنند تا زندگی برای انسان ها راحتتر و ایمن تر شود.

ما نیز در این پروژه سعی کردیم با استفاده از مدل های یادگیری عمیق، مدلی برای تشخیص خودروهای ایرانی آموزش دهیم. در این گزارش ما روند کار خود از ابتدایی ترین قدم یعنی جمع آوری داده ها تا قدم نهایی یعنی ارزیابی مدل را توضیح می دهیم.

# بخش اول: جمع آوری مجموعه داده

# بخش دوم: پیش پردازش داده ها

## حذف داده های پرت

دیتاست ما از سایت دیوار جمع آوری شده بنابراین ممکن است برخی عکس ها مناسب مدل ما نباشد. هدف ما تشخیص خودرو های ایرانی است و از آنجایی که برخی عکس های خودرو ها در سایت دیوار از داخل ماشین ها یا از چرخ و ... است و این داده ها برای مدل ما مناسب نیستند لذا باید این عکس ها را از دیتاست خود حذف کنیم تا دیتاست با کیفیت تری داشته باشیم. برای اینکار لازم است تا مدلی داشته باشیم که عکس های خودرو را از عکس های دیگر (مثل عکس های چرخ خودرو، داخل خودرو و...) را تفکیک کند و تشخیص دهد. برای اینکار ما از داده هایی که داشتیم به صورت رندوم دیتاستی ایجاد کردیم که یک دسته شامل عکس های خودرو ها از زوایای مختلف و دسته دیگر شامل عکس هایی از خودرو ها مثل داخل خودرو یا چرخ و ... است.



تصویر 1- نمونه ای از تصاویر دسته خودرو تصویر 2 - نمونه ای از تصاویر دسته غیر خودرو

ما 3 مدل را برای اینکار در نظر گرفتیم و آموزش دادیم[[2]](#footnote-2) که دقت و مقدار خطای مدل ها به شرح زیر است:

|  |  |  |
| --- | --- | --- |
| نام مدل | دقت مدل | مقدار Loss |
| Inception\_resnet\_v2 | 0.985 | 0.044 |
| Inception\_v3 | 0.980 | 0.053 |
| mobilenet | 0.995 | 0.036 |

جدول 1 - مقدار دقت و خطای 3 مدل برای تشخیص داده های پرت از داده های مناسب

همانطور که میبینید این مدل ها دقت مناسبی دارند اما خب هر کدام از این مدل ها در تعداد بسیار کمی از عکس ها ممکن است به اشتباه کلاس داده را تشخیص دهند ما برای آنکه این خطا را به حداقل برسانیم 3 مدل را با هم ترکیب کردیم و یک مدل به نام مدل ensemble ساختیم که لایه نهایی آن یک لایه Average است که از خروجی 3 مدل میانگین میگیرد. در نهایت دقت مدل ensemble برابر با 0.995 و مقدار خطای آن به 0.028 رسید و از آنجایی که ما میخواهیم داده های پرت را حذف کنیم میخواهیم از یک مدل با حداکثر دقت و حداقل خطا استفاده کنیم که مدل ensemble برای اینکار گزینه مناسبی است.

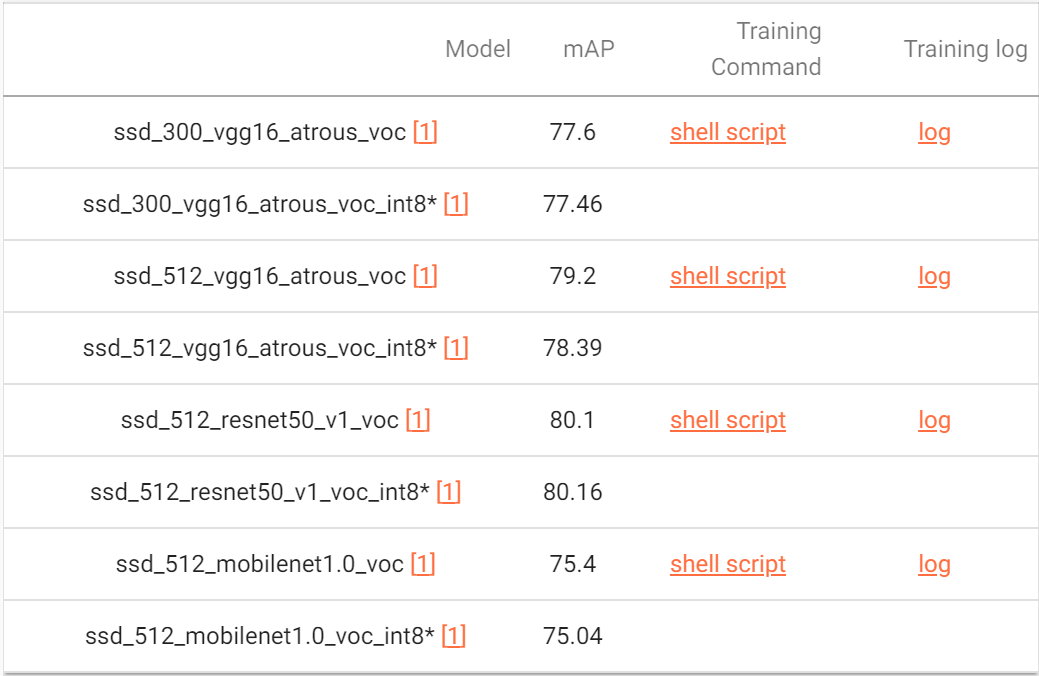
نکته ای که لازم است به آن اشاره کنیم این است که چون ما برای این بخش دیتاست کوچکی در نظر گرفتیم که تنها شامل 500 داده آموزش و 100 داده validation و 200 داده آزمون است بنابراین به روش transfer learning ابتدا وزن های imagenet برای این مدل ها را به عنوان وزن اولیه در نظر گرفتیم و سپس مدل را روی دیتاست خود آموزش دادیم.

## برچسب زدن عکس ها

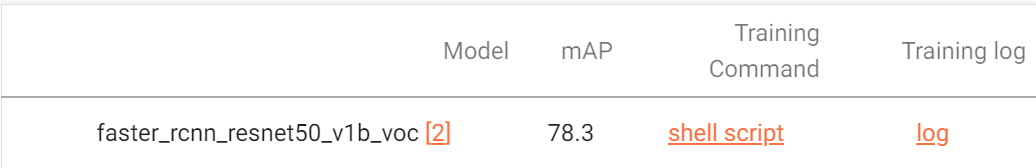
همانطور که میدانید ما حدود 100 هزار عکس از انواع ماشین ها را از سایت دیوار دریافت کرده ایم و حالا نیاز داریم تا لیبل آنها را مشخص کنیم. اگر پروژه تنها شامل طبقه بندی بود، لیبل گذاری بسیار راحت بود و تنها با گذاشتن عکس هر نوع ماشین در پوشه مربوطه، لیبل آن عکس مشخص میشد اما ما در این پروژه علاوه بر طبقه بندی، تشخیص مکان ماشین را نیز داریم پس لازم است تا مختصات ماشین که در اینجا شامل دو نقطه‌ی گوشه بالا سمت چپ و گوشه پایین سمت راست است نیز در برچسب عکس ها مشخص شده باشد. ما اگر بخواهیم برای تک تک عکس ها برچسب گذاری را به صورت دستی انجام دهیم، از آنجایی که تعدادعکس ها بسیار زیاد است در نتیجه برچسب گذاری آنها به شدت زمان خواهد برد. بنابراین باید به دنبال روش های برچسب گذاری اتوماتیک یا شبه اتوماتیک باشیم.

برای همین ما تحقیق کردیم تا ببینیم از بین مدل های از پیش آموزش داده شده، دقت کدام مدل بیشتر است.

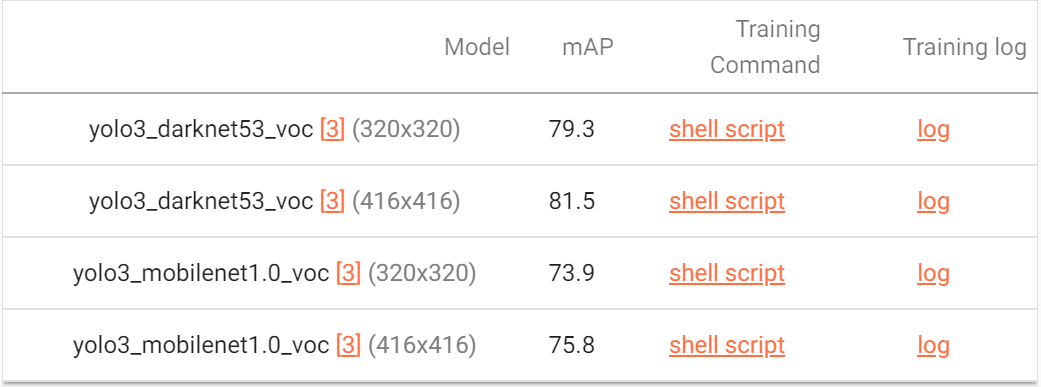
درنهایت پس از بررسی های مختلف به کتابخانه model\_zoo از ماژول gluoncv رسیدیم که این کتابخانه در بحث تشخیص اشیا شامل مدل های مختلف از پیش آموزش دیده روی دیتاست های COCO و Pascal VOC می باشد که در ادامه برخی از آنها را مشاهده میکنید:



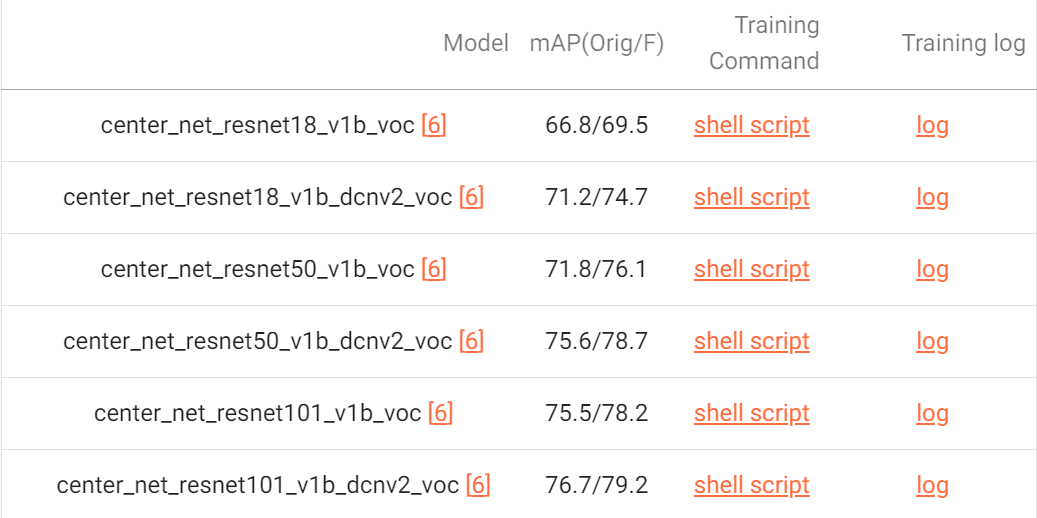
تصویر 3 – مدل های مختلف SSD از پیش آموزش دیده با دقت آن ها روی دیتاست Pascal VOC



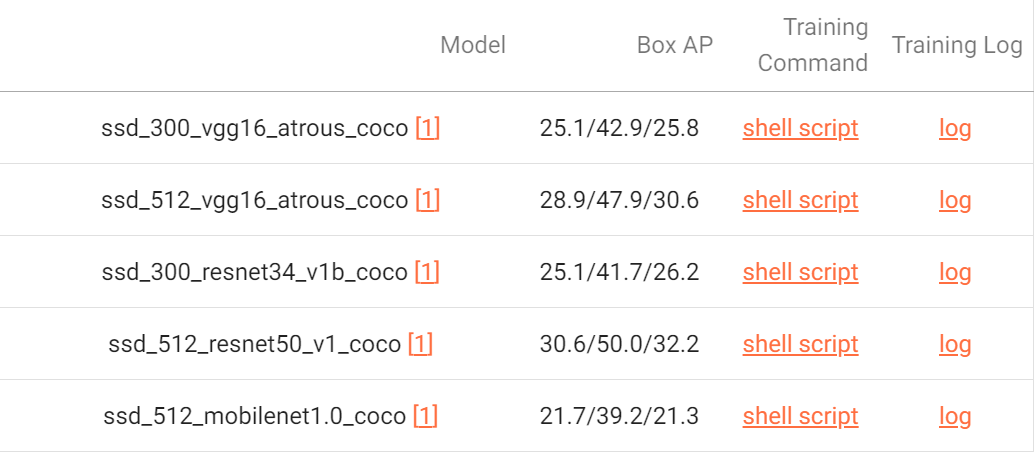
تصویر 4 - مدل های مختلف Faster RCNN از پیش آموزش دیده با دقت آن ها روی دیتاست Pascal VOC



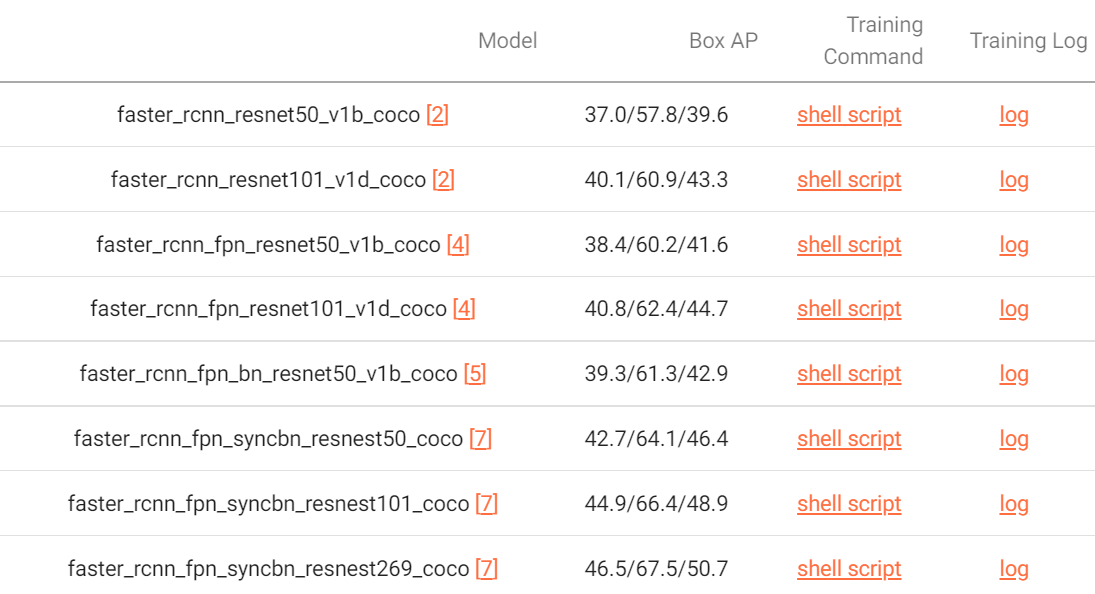
تصویر 5 - مدل های مختلف YOLO V3 از پیش آموزش دیده با دقت آن ها روی دیتاست Pascal VOC



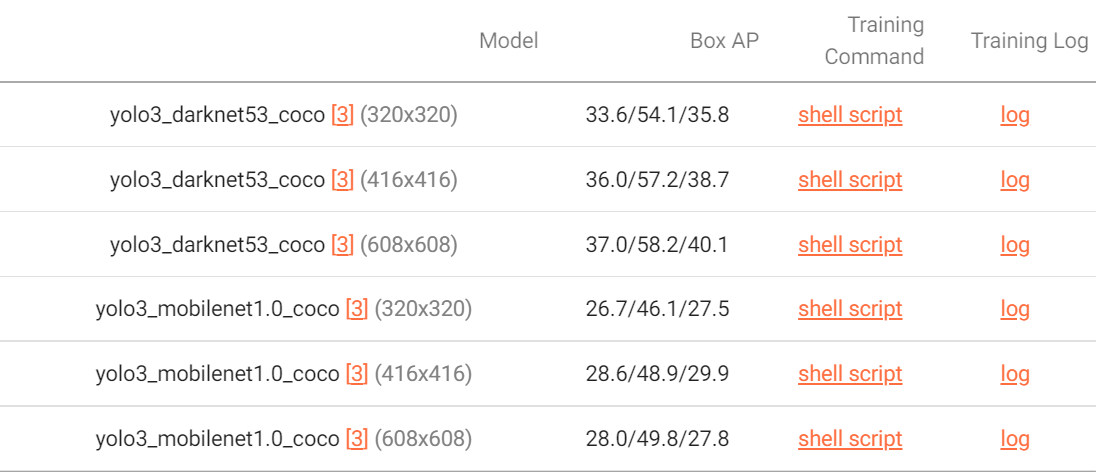
تصویر 6 - مدل های مختلف CenterNet از پیش آموزش دیده با دقت آن ها روی دیتاست Pascal VOC



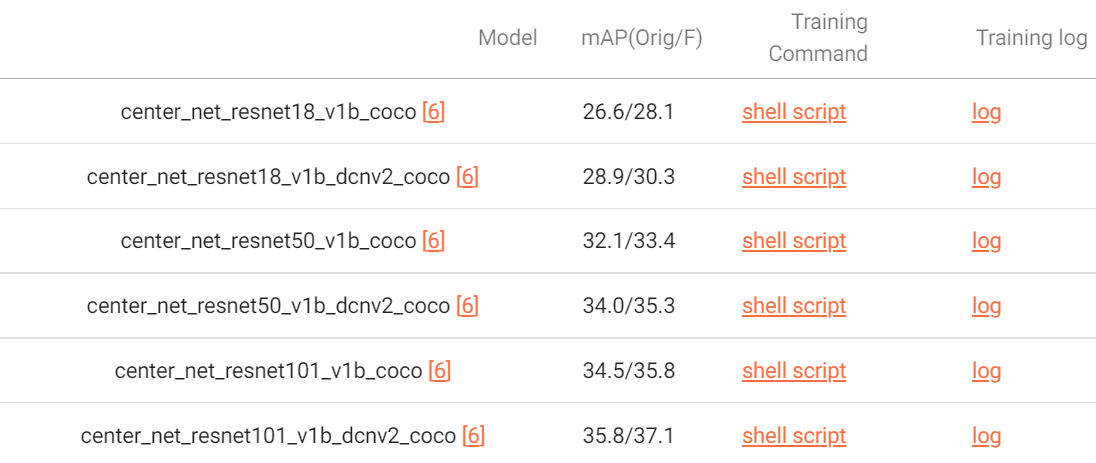
تصویر 7 - مدل های مختلف SSD از پیش آموزش دیده با دقت آن ها روی دیتاست Pascal VOC



تصویر 8 - مدل های مختلف Faster RCNN از پیش آموزش دیده با دقت آن ها روی دیتاست Pascal VOC



تصویر 9 - مدل های مختلف YOLO V3 از پیش آموزش دیده با دقت آن ها روی دیتاست Pascal VOC



تصویر 10 - مدل های مختلف CenterNet از پیش آموزش دیده با دقت آن ها روی دیتاست Pascal VOC

ما تمام مدل های از پیش آموزش دیده‌ی این کتابخانه روی دیتاست Coco و Pascal VOC را تست کردیم و در نهایت 3 تا از مدل ها دقت مناسبی روی ماشین ها و تشخیص مختصات آنها داشتند که به شرح زیر می باشند:

1. faster\_rcnn\_resnet50\_v1b\_voc با دقت 78.3 mAP
2. ssd\_512\_resnet50\_v1\_voc با دقت 80.1 mAP
3. yolo3\_darknet53\_voc با دقت 81.5 mAP

این مدل ها از لحاظ تشخیص مختصات ماشین ها تقریبا مشابه هم بودند و از بین تمام مدل ها دقت بیشتری داشتند.

اما ما به دنبال مدلی هستیم که هم از لحاظ دقت و هم از لحاظ سرعت مدل بهتری باشد. پس این سه مدل را از لحاظ سرعت پیش بینی نیز مقایسه کردیم که به ترتیب سرعت به شرح زیر می باشند:

1. ssd\_512\_resnet50\_v1\_voc با دقت 80.1 mAP
2. yolo3\_darknet53\_voc با دقت 81.5 mAP
3. faster\_rcnn\_resnet50\_v1b\_voc با دقت 78.3 mAP

بنابراین چون بیشترین سرعت مربوط به مدل ssd\_512\_resnet50\_v1\_voc بود لذا تصمیم گرفتیم تا از این مدل برای لیبل گذاری داده های خود استفاده کنیم و اشیائی که امتیاز آنها از یک آستانه (که در اینجا 0.8 است) بیشتر است آن ها را در نظر میگیریم.

در دیتاست ما تعداد کمی از عکس ها شامل ماشین های دیگری در پس زمینه ی ماشین اصلی نیز می باشد و مشخصا مدلی که در نظر گرفتیم، این ماشین های پس زمینه را نیز تشخیص میدهد. اما چالشی که ما به آن برخوردیم و این چالش تنها در تعداد بسیار کمی از عکس ها به چشم میخورد اما با این حال میخواستیم برای آنها نیز تشخیص مناسبی داشته باشیم، این است که در حین تشخیص ماشین های داخل یک عکس، این مدل و خیلی از مدل های دیگری که آنها را تست کردیم ضریب اطمینان مربوط به برخی ماشین های پس زمینه را بیشتر از ماشین اصلی میدادند، لذا ما اگر بر اساس امتیاز (ضریب اطمینان) بخواهیم ماشین ها را انتخاب کنیم ممکن است ماشینی در پس زمینه انتخاب شود که مدل آن الزاما با مدل ماشین مدنظر ما یکی نباشد. مثلا فرض کنید یک تصویر داریم که ماشین اصلی در آن سمند است و در پس زمینه چند ماشین از جمله یک پراید می باشد و ما میدانیم در این عکس به دنبال پیدا کردن سمند و مکان آن در تصویر هستیم. حال اگر ضریب اطمینان پراید از سمند بیشتر باشد (مثلا برای پریاد ضریب اطمینان 0.99 و سمند 0.95 باشد) و ما براساس بیشترین ضریب اطمینان شی مدنظر را انتخاب کنیم در نهایت به اشتباه مکان سمند داخل عکس را تشخیص داده ایم.



تصویر 11- تصویری که در آن ضریب اطمینان مربوط به ماشین اصلی کمتر از ماشین پس زمینه می باشد.

مدل ما باکس ها را با دقت خیلی بالایی تشخیص میدهد و به گونه ای میتوانیم بگوییم که میزان درستی باکس ها از ضریب اطمینانی که مدل به ما میدهد بیشتر است. درنتیجه ما برای جلوگیری از این مشکل آمدیم از ایده ی انتخاب شی با بیشترین مساحت bounding box استفاده کردیم. به این صورت که از بین تمام ماشین هایی که مقدار ضریب اطمینان آنها از آستانه بیشتر است، از آنجایی که ماشین اصلی در عکس قطعا بیشترین مساحت از عکس را اشغال میکند، لذا ما باکسی که بیشترین مساحت را دارد انتخاب میکنیم.

1. پست الکترونیکی: mwaghei78@gmail.com [↑](#footnote-ref-1)
2. مدل ها شامل 3 مدل inception\_resnet\_v2 و inception\_v3 و mobilenet می باشد. [↑](#footnote-ref-2)