

بسم الله الرحمن الرحيم

شرکت مهندسی نرم افزاری هلو

گزارش روش‌های آموزشی جهت شناسایی میز – گزارش مجموعه‌های داده‌ای

کاری از امیرعلی نسیمی

فهرست

۲ مقدمه
۲ مجموعه‌های داده‌ای
۲ مقدمه
۳ TABLE IMAGE DATASET
۳ INDOOR OBJECTS DETECTION
۳ ZENODO DATASET
۴ RESNET-50 INDOOR OBJECT DETECTION
۴ روش‌های حل مسئله
۴ مقدمه
۴ DURAN
۶ VOTE NET
۸ AZAR INDOOR OBJECT DETECTION
۹ YOLO 8
۹ روش اول
۱۴ روش دوم

مقدمه

در خصوص مسئله میز خالی، شناسایی اولیه‌ی میزها بسیار با اهمیت می‌باشد. با توجه به این مسئله که میزهای خالی توسط الگوریتم Yolo قابل شناسایی نمی‌باشد – بطور دقیق‌تر میز اداری قابل شناسایی نیست – لذا آموزش مدل‌های هوش مصنوعی جهت رسیدگی به این مسئله حائز اهمیت می‌باشد. در این گزارش به بررسی برخی از روش‌ها به همراه مجموعه‌های داده‌ای مربوط به این موارد مورد بررسی قرار گرفته است.

مجموعه‌های داده‌ای

مقدمه

با توجه به نیاز پروژه مبنی بر شناسایی میزهای اداری، باید با دقت عمل کرد. با توجه به این مسئله روش‌هایی که بر مجموعه‌های داده‌ای Indoor Object Detection کار کرده‌اند مورد نظر می‌باشند. این مجموعه‌های داده‌ای معمولاً دارای دسته‌های محدود بصورت زیر می‌باشند:

یخچال کابینت در

پنجره	میز	صندلی
-------	-----	-------

...

با توجه به این مسئله، آموزش مدل‌هایی که بر مجموعه های داده‌ای Indoor آموزش داده شده‌اند مورد نظر می‌باشد.

Table Image Dataset

این مجموعه داده شامل حدود ۱۰۰۰ تصویر از میز می‌باشد. متأسفانه هیچگونه BBX ای برای این مجموعه تعبیه نشده است لذا نیاز به انجام این مسئله جهت آموزش می‌باشد. جهت بررسی بیشتر این مجموعه به این [لینک](#) مراجعه شود.

Indoor Objects Detection

این مجموعه داده شامل ۱۰ دسته و در حدود ۱۳۰۰ نمونه می‌باشد. این مجموعه داده دارای BBX بوده و قابل استفاده در شبکه‌های هوش مصنوعی – مانند Yolo – می‌باشد. جهت مشاهده جزئیات این مجموعه داده به این [لینک](#) مراجعه شود. دسته‌ها بصورت زیر می‌باشد:

درب کابینت	درب یخچال	درب	صندلی
پنجره	درب	میز	مبل
کابینت	میله	درب باز	

Zenodo Dataset

این مجموعه داده شامل ۶ دسته بصورت Sequential می‌باشد. این مجموعه داده دارای BBX نمی‌باشد. جهت مشاهده جزئیات این مجموعه داده به این [لینک](#) مراجعه شود. دسته‌ها بصورت زیر می‌باشد:

تابلو خروج	صندلی	ساعت دیواری
------------	-------	-------------

RESNET-50 INDOOR OBJECT DETECTION

این مجموعه داده شامل تنها یک دسته و در حدود ۲۰۰۰ نمونه می باشد. این مجموعه داده دارای BBX نمی باشد اما بدین دلیل که تنها شامل میز می باشد لذا بسیار کاربردی خواهد بود. جهت مشاهده جزئیات این مجموعه داده به این [لینک](#) مراجعه شود.

روش های حل مسئله

مقدمه

روش هایی که بر مجموعه های داده ای Indoor Object Detection کار کرده اند در ادامه معرفی می شوند:

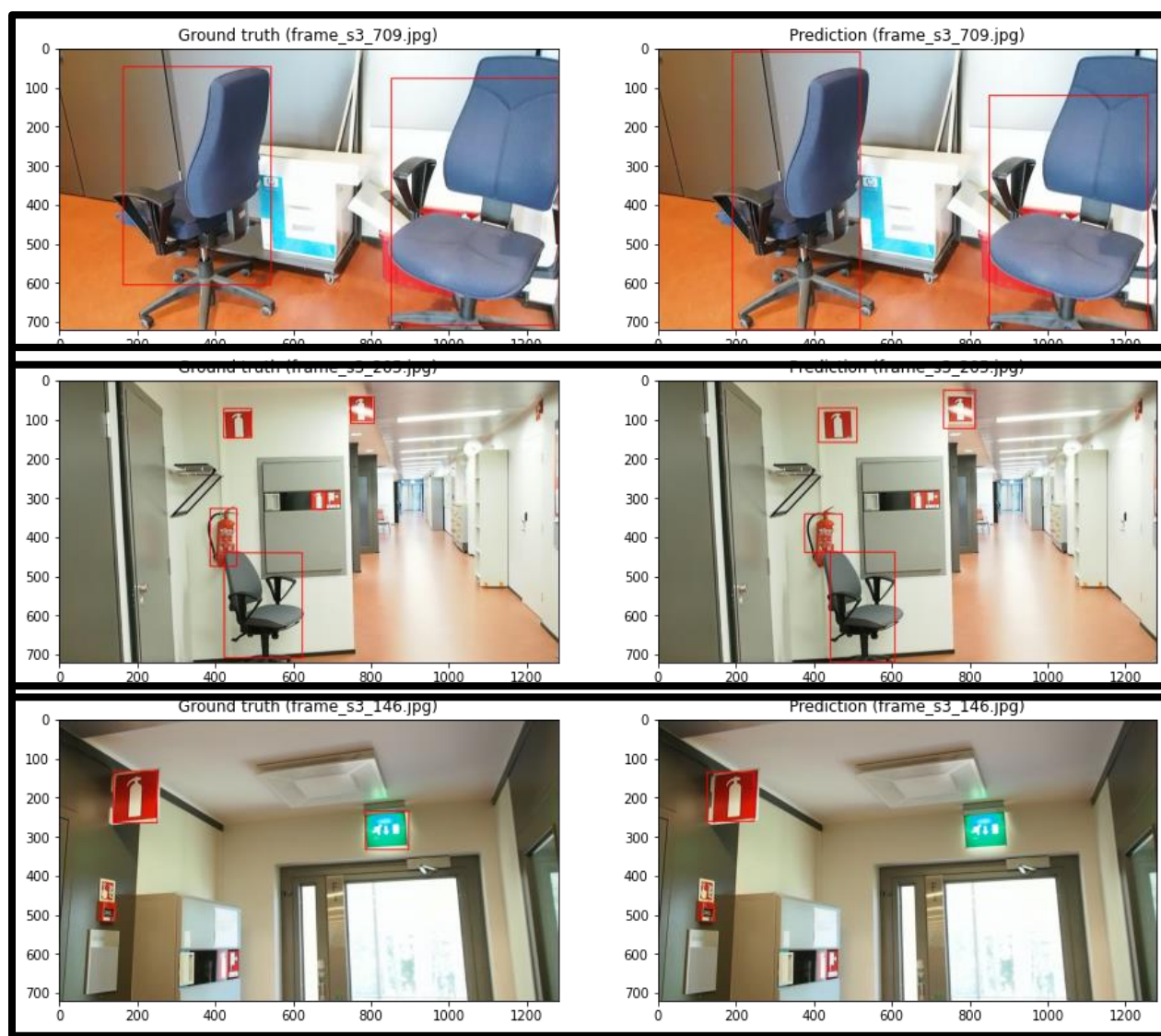
Duran

این [روش](#) قابلیت استفاده از دو شبکه عصبی Dark Net و Yolo را دارد. قبلا آموزش داده شده است و وزن های آن قابل استفاده می باشد — هر چند که مجموعه داده ای که آموزش بر آن انجام داده شده است فاقد میز می باشد. مجموعه داده مورد نظر:

ساعت دیواری	صندلی	تابلو خروج
مانیتور	دستگاه پرینتر	سطل آشغال

نتایج مربوط به این مدل بصورت زیر قابل مشاهده می باشد:

نتیجه: هرچند که روش معرفی شده در مواقعی تاثیر گذار بوده اما برای روش مورد نظر پروژه، کارایی لازم را ندارد. علت این موضوع، **نبود میز در مجموعه داده** مورد نظر می باشد.



Vote Net

این روش با استفاده از شبکه Vote Net اقدام به شناسایی می کند. بطور خاص این روش جهت شناسایی اشیاء بصورت سه بعدی عمل می کند. VoteNet.

یک چارچوب تشخیص اشیاء سه بعدی بر اساس شبکه های عصبی است که برای پردازش داده های ابرنقطه^۱ طراحی شده است. این ابزار اصولاً برای وظایفی مانند شناسایی و موقعیت یابی اشیاء در فضای سه بعدی از داده های بدست آمده از حسگر عمق استفاده می شود که در برنامه هایی مانند رانندگی خودکار، رباتیک و واقعیت افزوده معمولاً به کار می رود. VoteNet بر اساس معماری PointNet ساخته شده است که یک شبکه عصبی برای پردازش داده های ابرنقطه است. نحوه کار بدین صورت می باشد که:

۱. VoteNet: Point Cloud Representation یک ابرنقطه را به عنوان ورودی می پذیرد، که اصولاً مجموعه ای از نقاط سه بعدی در فضا است. هر نقطه به طور معمول اطلاعاتی در مورد مختصات سه بعدی آن (x, y, z) و ویژگی های اضافی مانند رنگ یا شدت را دارد.

۲. Voting Process: نام "VoteNet" از مرحله اول عملکرد آن به وجود آمده است. در این حالت مراکز برای یکسری از اشیاء تعریف شده است. وظیفه این است که به هر مرکز، یک امتیاز - رای - داده شود. این کار با در نظر گرفتن همسایگی محلی^۲ نقاط انجام شده و سعی در پیش بینی این دارد که هر نقطه قسمتی از مرکز یک شی هست یا نه. این پیش بینی معمولاً از طریق یک شبکه عصبی کانولوشنی (CNN) یا معمار مشابه انجام می شود.

۳. Grouping Votes: بعد از تولید رای ها، شبکه آن ها را گروه بندی می کند تا مکان های ممکن مراکز اشیاء را تشخیص دهد. این فرآیند، گروه بندی رای هایی را که در نقاط نزدیک به یکدیگر در فضای سه بعدی هستند، به عنوان نماینده های ممکن مربوط به مرکز یک شی در نظر می گیرد.

¹ point cloud data

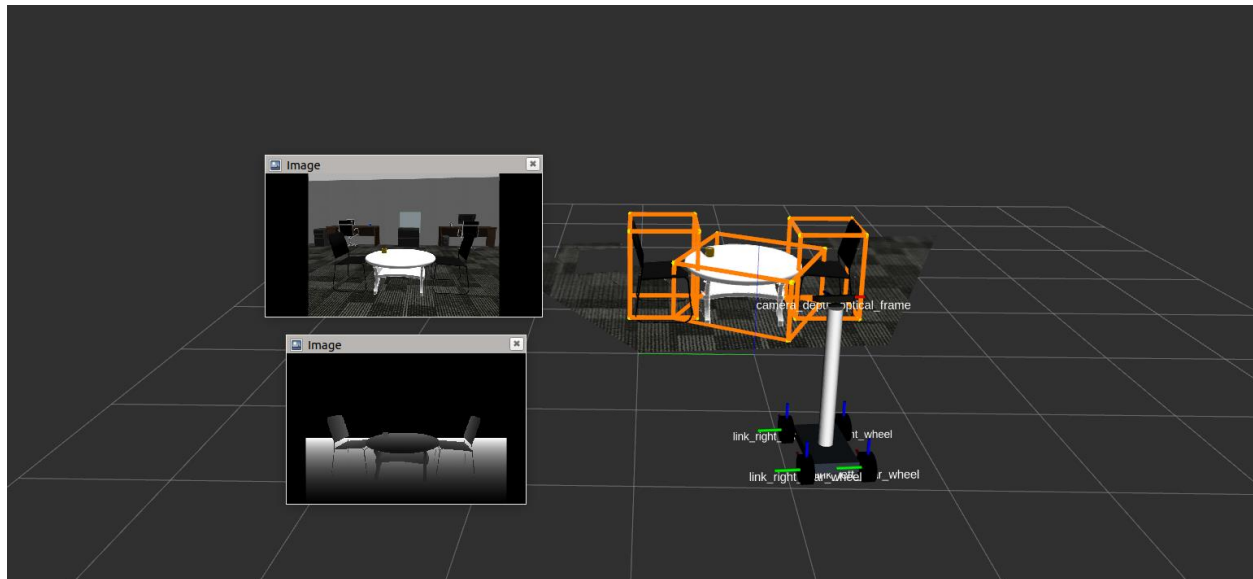
² local neighborhood

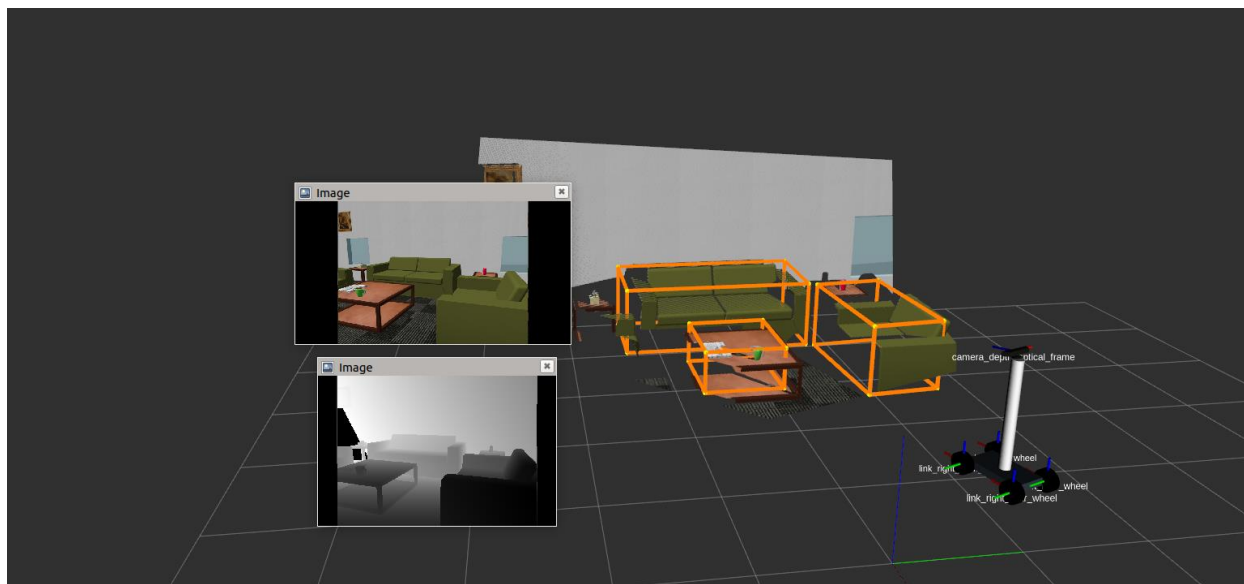
۴. **Object Detection**: هر گروه از رای‌ها به عنوان یک نماینده ممکن برای هر شی در نظر گرفته می‌شود. برای هر نماینده شبکه‌ی عصبی، مکان، اندازه و جهت مراکز اشیاء را در فضای سه‌بعدی در نظر می‌گیرد. این مرحله به طور معمول شامل لایه‌های اضافی شبکه عصبی برای تخمین دقیق این پارامترها است.

۵. **Post-Processing**: بعد از به دست آوردن نمایندگان مراکز اشیاء، **VoteNet** ممکن است مراحل پس‌پردازش را اجرا کند تا تشخیص‌های تکراری یا غیرضروری را حذف کند و نتایج نهایی تشخیص اشیاء را بهبود دهد.

۶. **Object Classification**: این مرحله اختیاری است. می‌توان طبقه‌بندی اشیاء را اضافه کرد تا نوع یا دسته هر شی تشخیص داده شده تعیین شود.

۷. خروجی نهایی: خروجی نهایی **VoteNet** یک لیست از اشیاء تشخیص داده شده است، هر یک با مختصات سه‌بعدی، اندازه، جهت و به طور اختیاری برچسب دسته آن نماینده شده‌اند.





نتیجه: VoteNet از جمله تکنیک‌های مربوط به یادگیری عمیق بوده که توانایی پردازش **داده‌های سه‌بعدی** را دارد. با توجه به این مسئله، استفاده از این شبکه مورد نیاز نمی‌باشد.

Azar Indoor Object Detection

این **روش** قابلیت استفاده از دو شبکه عصبی Dense Net و Yolo را دارد. قبلاً آموزش داده شده است و وزن‌های آن قابل استفاده می‌باشد – هرچند که مجموعه داده‌ای که آموزش بر آن انجام داده شده است فاقد میز می‌باشد.

Yolo 8

نسخه ۸ الگوریتم Yolo – نسبت به نسخه‌های قدیمی تر – از سرعت و دقت بیشتری برخوردار می‌باشد. جهت آموزش از چند مجموعه داده که در ابتدای این گزارش معرفی شده است استفاده گردید. با توجه به وجود نسخه‌های متعدد از این الگوریتم و با توجه به کم بودن تعداد مجموعه داده، موارد زیر مورد نظر می‌باشد:

- از نسخه‌های Nano و Small و Medium استفاده شد.
- برای مجموعه داده از عملیات Augmentation مربوط به خود الگوریتم Yolo استفاده شد. موارد استفاده شده بصورت random crops, flipping, rotation, and distortion می‌باشد. دقت شود که انجام Augment در تعداد نمونه‌ها تغییر ایجاد نمی‌کند.
- آموزش بر دو مجموعه داده ذکر شده انجام شد. با توجه به اینکه یکی از مجموعه‌های داده‌ای – علی رغم تعداد بسیار زیاد و تنوع مناسب – فاقد BBX است لذا در حین آموزش استفاده نشد.

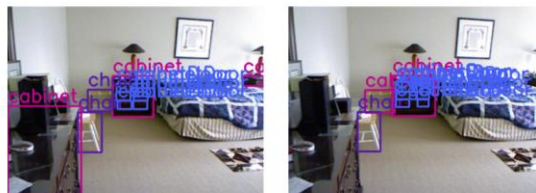
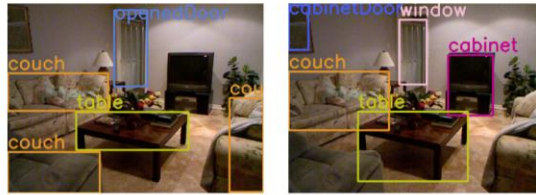
روش اول

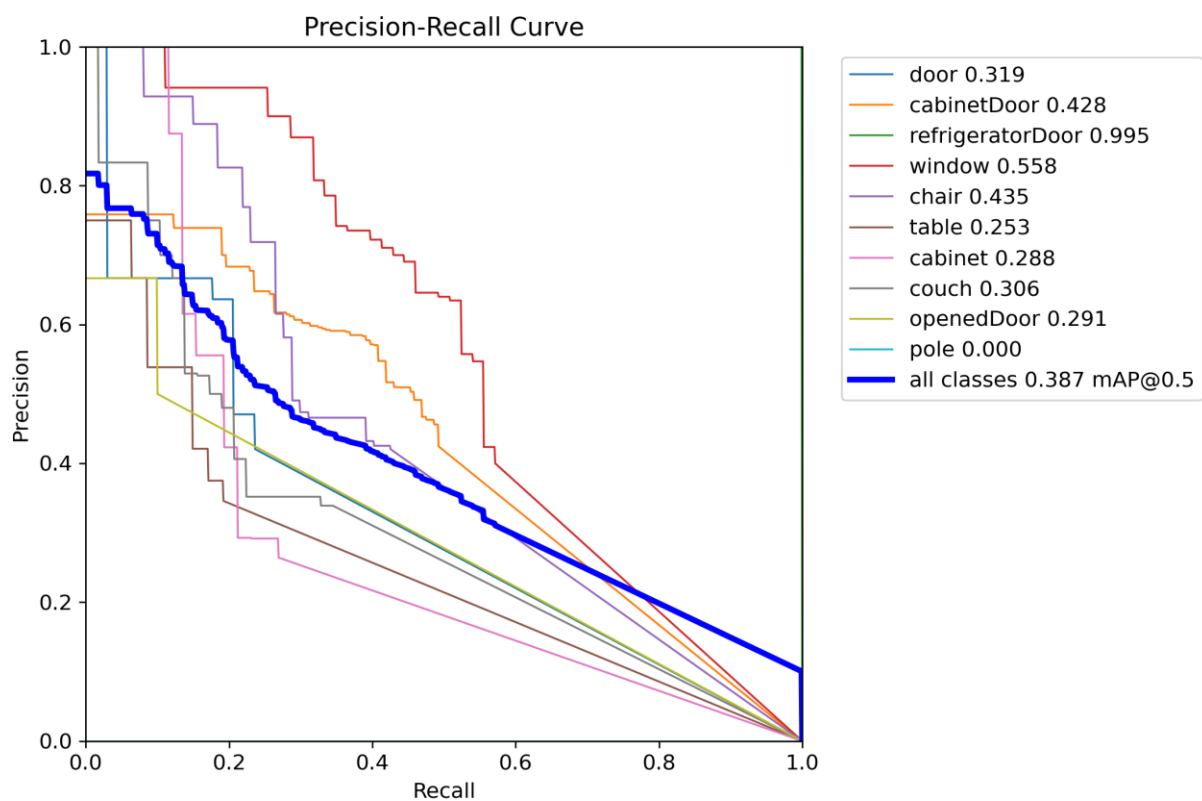
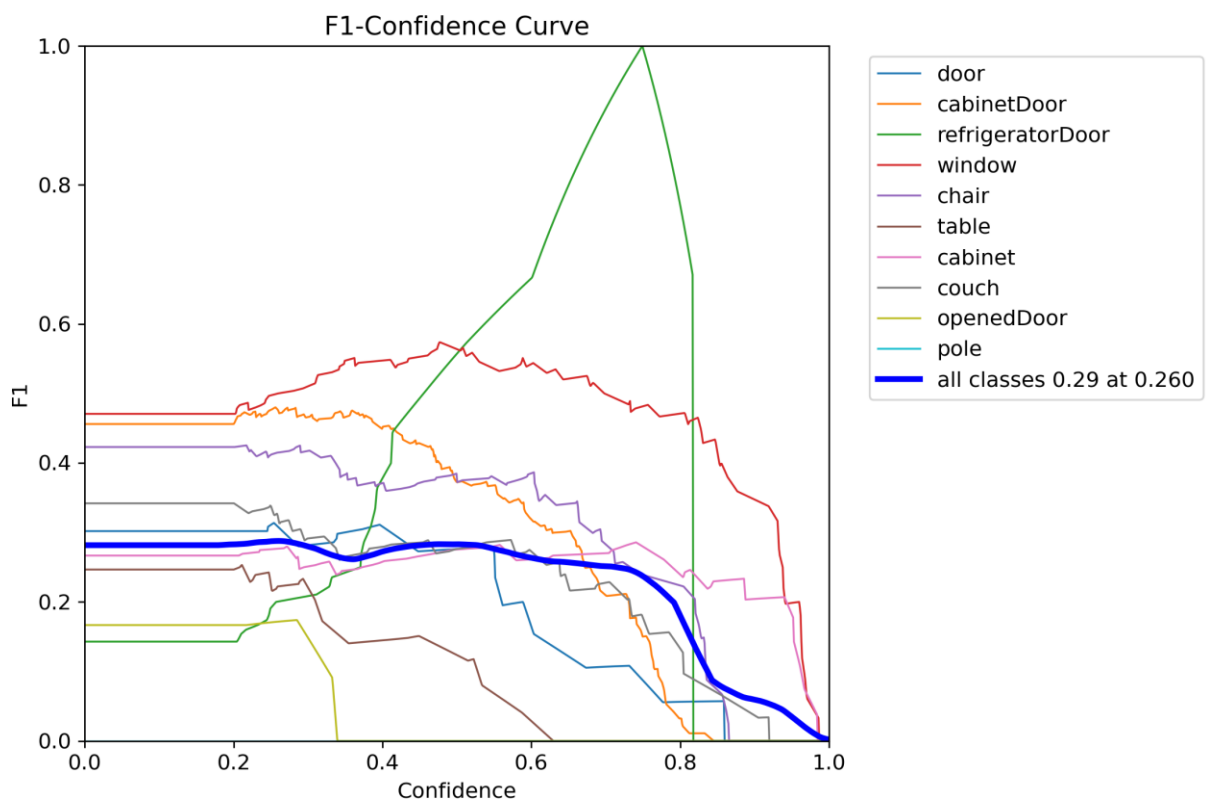
در این روش از کلیه داده‌های مربوط به مجموعه داده RESNET-50 INDOOR OBJECT DETECTION استفاده شده است. بالاترین دقت از میان کلیه آموزش‌های انجام شده مربوط به زیر می‌باشد:

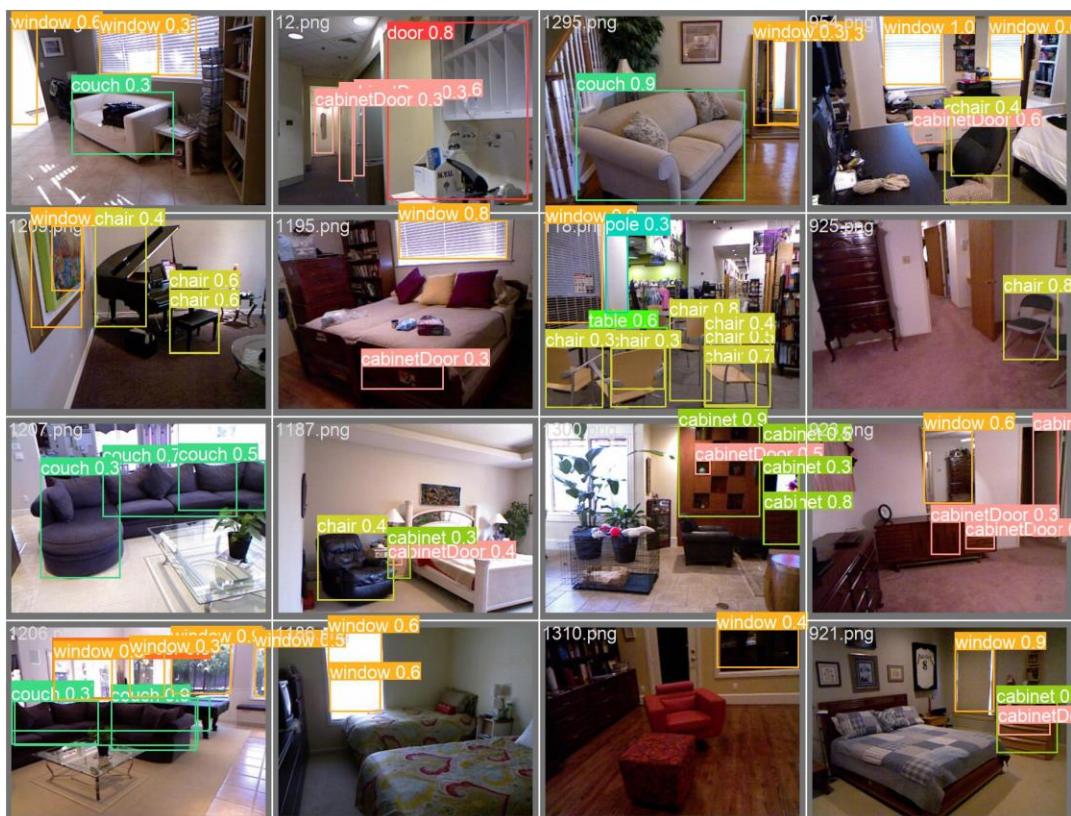
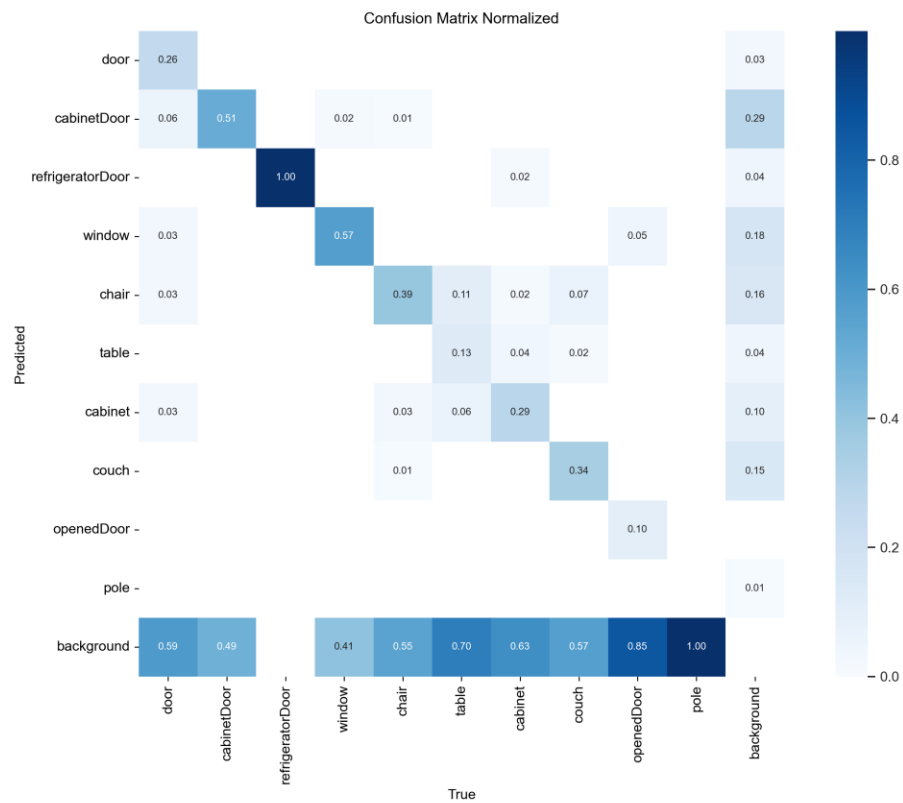
mean_Avg_Precision	
door	0.181935
cabinetDoor	0.182304
refrigeratorDoor	0.796000
window	0.377131
chair	0.239105
table	0.101698
cabinet	0.198527
couch	0.172206
openedDoor	0.156833
pole	0.000000

Mean Average Precision @.5:.95 : 0.24057387569545127
Mean Average Precision @ .50 : 0.38722721629488843
Mean Average Precision @ .70 : 0.2561661336033683









روش دوم

در این روش تنها میز جدا شده و آموزش داده شده است. با توجه به این مسئله، بهترین نتیجه به صورت زیر می باشد:

```
Mean Average Precision @.5:.95 : 0.17311621586112275
Mean Average Precision @ .50    : 0.3226054504926471
Mean Average Precision @ .70    : 0.19961315349205805
```

