بسم الله الرحمن الرحیم

شرکت مهندسی نرم افزاری هلو

گزارش روش­های آموزشی جهت شناسایی میز – گزارش مجموعه­های داده­ای

کاری از امیرعلی نسیمی

**فهرست**

[مقدمه 2](#_Toc145866163)

[مجموعه­های داده­ای 2](#_Toc145866164)

[مقدمه 2](#_Toc145866165)

[Table Image Dataset 3](#_Toc145866166)

[Indoor Objects Detection 3](#_Toc145866167)

[Zenodo Dataset 3](#_Toc145866168)

[RESNET-50 INDOOR OBJECT DETECTION 4](#_Toc145866169)

[روش­های حل مسئله 4](#_Toc145866170)

[مقدمه 4](#_Toc145866171)

[Duran 4](#_Toc145866172)

[Vote Net 6](#_Toc145866173)

[Azar Indoor Object Detection 8](#_Toc145866174)

[Yolo 8 9](#_Toc145866175)

[روش اول 9](#_Toc145866176)

[روش دوم 14](#_Toc145866177)

# 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 مجموعه های داده­ای Indoor Object Detection کار کرده­اند مورد نظر می­باشند. این مجموعه­های داده­ای معمولا 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 1000 تصویر از میز می­باشد. متاسفانه هیچگونه BBXای برای این مجموعه تعبیه نشده است لذا نیاز به انجام این مسئله جهت آموزش می­باشد. جهت بررسی بیشتر این مجموعه به این [لینک](https://www.kaggle.com/datasets/dataclusterlabs/table-image-dataset-indoor-object-detection) مراجعه شود.

## Indoor Objects Detection

این مجموعه داده شامل 10 دسته و در حدود 1300 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](https://www.kaggle.com/datasets/thepbordin/indoor-object-detection) 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 6 دسته بصورت Sequential می­باشد. این مجموعه داده دارای BBX نمی­باشد. جهت مشاهده جزئیات این مجموعه داده به این [لینک](https://zenodo.org/record/2654485) مراجعه شود. دسته­ها بصورت زیر می­باشد:

|  |  |  |
| --- | --- | --- |
| تابلو خروج | صندلی | ساعت دیواری |
| سطل آشغال | **دستگاه پرینتر** | **مانیتور** |

## RESNET-50 INDOOR OBJECT DETECTION

این مجموعه داده شامل تنها یک دسته و در حدود 2000 نمونه می­باشد. این مجموعه داده دارای BBX نمی­باشد اما بدین دلیل که تنها شامل میز می­باشد لذا بسیار کاربردی خواهد بود. جهت مشاهده جزئیات این مجموعه داده به این [لینک](https://www.kaggle.com/datasets/putrikhairunnisa/resnet50-indoor-object-detection) مراجعه شود.

# روش­های حل مسئله

## مقدمه

روش­هایی که بر مجموعه های داده­ای Indoor Object Detection کار کرده­اند در ادامه معرفی می­شوند:

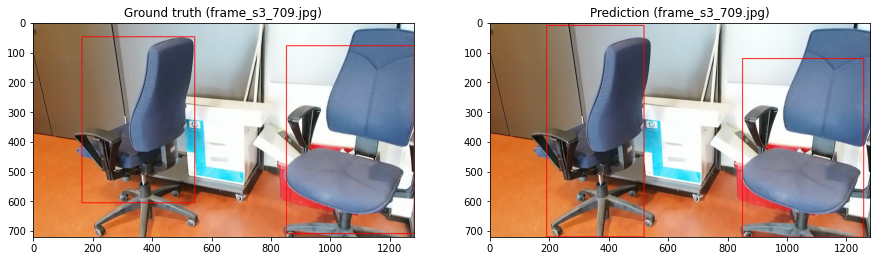
## Duran

این [روش](https://github.com/d-duran/IndoorObjectDetection) قابلیت استفاده از دو شبکه عصبی Dark Net و Yolo را دارد. قبلا 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. مجموعه داده مورد نظر:

|  |  |  |
| --- | --- | --- |
| تابلو خروج | صندلی | ساعت دیواری |
| سطل آشغال | **دستگاه پرینتر** | **مانیتور** |

نتایج مربوط به این مدل بصورت زیر قابل مشاهده می­باشد:

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



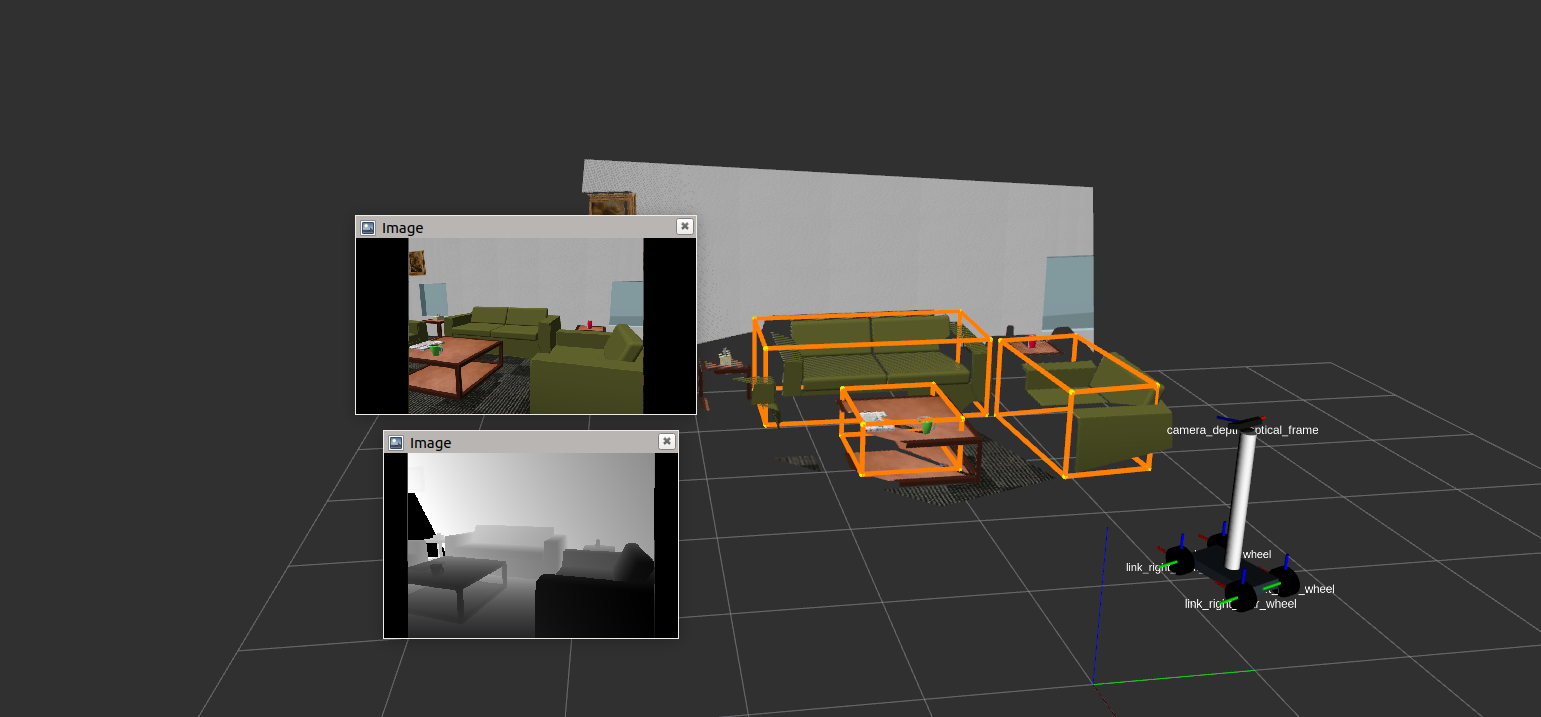
## Vote Net

این [روش](https://github.com/ShivamThukral/Indoor-Object-Detection) با استفاده از شبکه Vote Net اقدام به شناسایی می­کند. بطور خاص این روش جهت شناسایی اشیاء بصورت سه بعدی عمل می­کند. VoteNet

یک چارچوب تشخیص اشیاء سه بعدی بر اساس شبکه‌های عصبی است که برای پردازش داده‌های ابرنقطه[[1]](#footnote-1) طراحی شده است. این ابزار اصولاً برای وظایفی مانند شناسایی و موقعیت‌یابی اشیاء در فضای سه‌بعدی از داده‌های بدست آمده از حسگر عمق استفاده می‌شود که در برنامه‌هایی مانند رانندگی خودکار، رباتیک و واقعیت افزوده معمولاً به‌کار می‌رود. VoteNet بر اساس معماری PointNet ساخته شده است که یک شبکه عصبی برای پردازش داده‌های ابرنقطه است. نحوه کار بدین صورت می­باشد که:

1. Point Cloud Representation: VoteNet یک ابرنقطه را به عنوان ورودی می‌پذیرد، که اصولاً مجموعه‌ای از نقاط سه‌بعدی در فضا است. هر نقطه به طور معمول اطلاعاتی در مورد مختصات سه‌بعدی آن (x، y، z) و ویژگی‌های اضافی مانند رنگ یا شدت را دارد.
2. Voting Process: نام "VoteNet" از مرحله اول عملکرد آن به‌وجود آمده است. در این حالت مراکزی 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[[2]](#footnote-2) نقاط انجام شده و سعی در پیش‌بینی این دارد که هر نقطه قسمتی از مرکز یک شی هست یا نه. این پیش‌بینی معمولاً از طریق یک شبکه عصبی کانولوشنی (CNN) یا معمار مشابه انجام می‌شود.
3. Grouping Votes: بعد از تولید رای‌ها، شبکه آن‌ها را گروه‌بندی می‌کند تا مکان‌های ممکن مراکز اشیاء را تشخیص دهد. این فرآیند، گروه‌بندی رای‌هایی را که در نقاط نزدیک به یکدیگر در فضای سه‌بعدی هستند، به عنوان نماینده‌های ممکن مربوط به مرکز یک شی در نظر می‌گیرد.
4. 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.
5. Post-Processing: بعد از به‌دست 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.
6. Object Classification: این مرحله اختیاری است. می‌توان طبقه‌بندی اشیاء را اضافه کرد تا نوع یا دسته هر شی تشخیص داده شده تعیین شود.
7. خروجی نهایی: خروجی نهایی VoteNet یک لیست از اشیاء تشخیص داده‌شده است، هر یک با مختصات سه‌بعدی، اندازه، جهت و به‌طور 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](https://github.com/Amin-Azar/Indoor-Object-Detection) قابلیت استفاده از دو شبکه عصبی Dense Net و Yolo را دارد. قبلا 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.

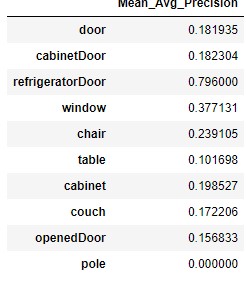
## Yolo 8

نسخه 8 الگوریتم Yolo – نسبت به نسخه­های قدیمی­تر – 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:

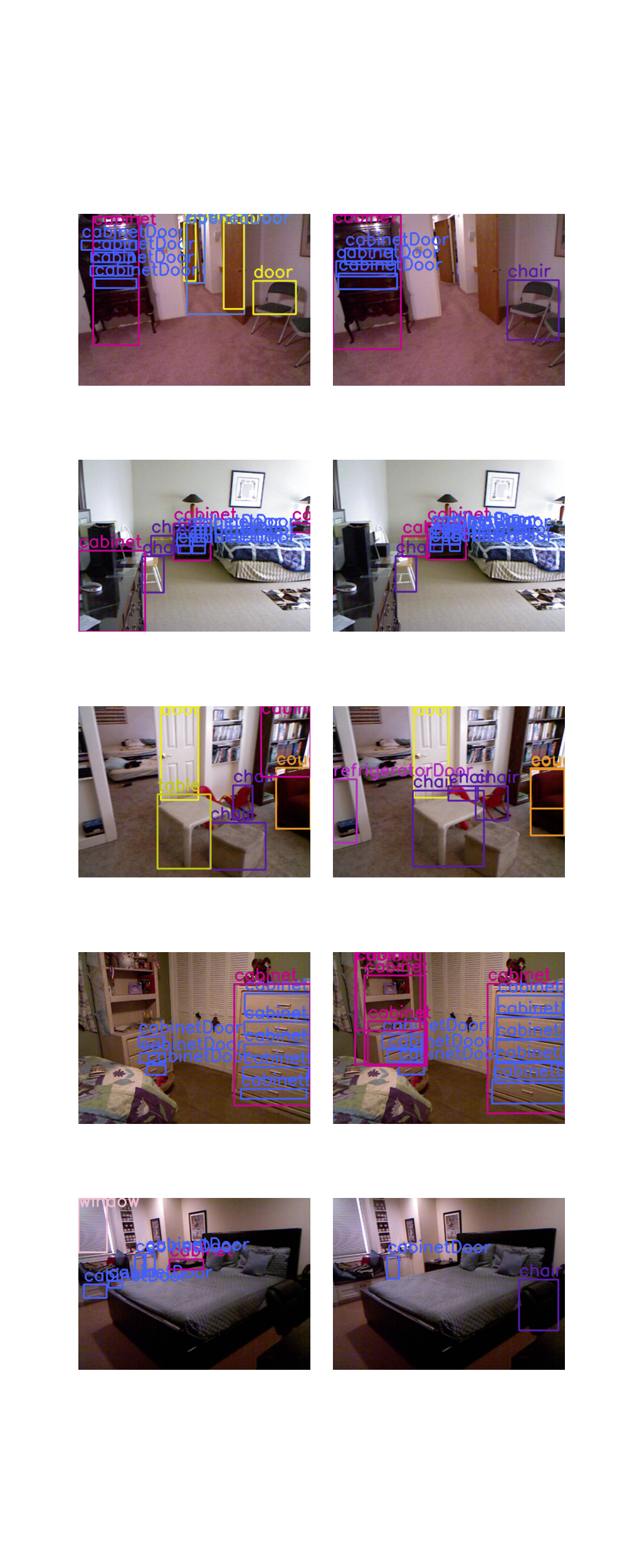
* از نسخه های Nano و Small و Medium استفاده شد.
* برای مجموعه داده از عملیات Augmentation مربوط به خود الگوریتم Yolo استفاده شد. موارد استفاده شده بصورت random crops, flipping, rotation, and distortion می­باشد. دقت شود که انجام Augment در تعداد نمونه­ها تغییر ایجاد نمی­کند.
* آموزش بر دو مجموعه داده 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

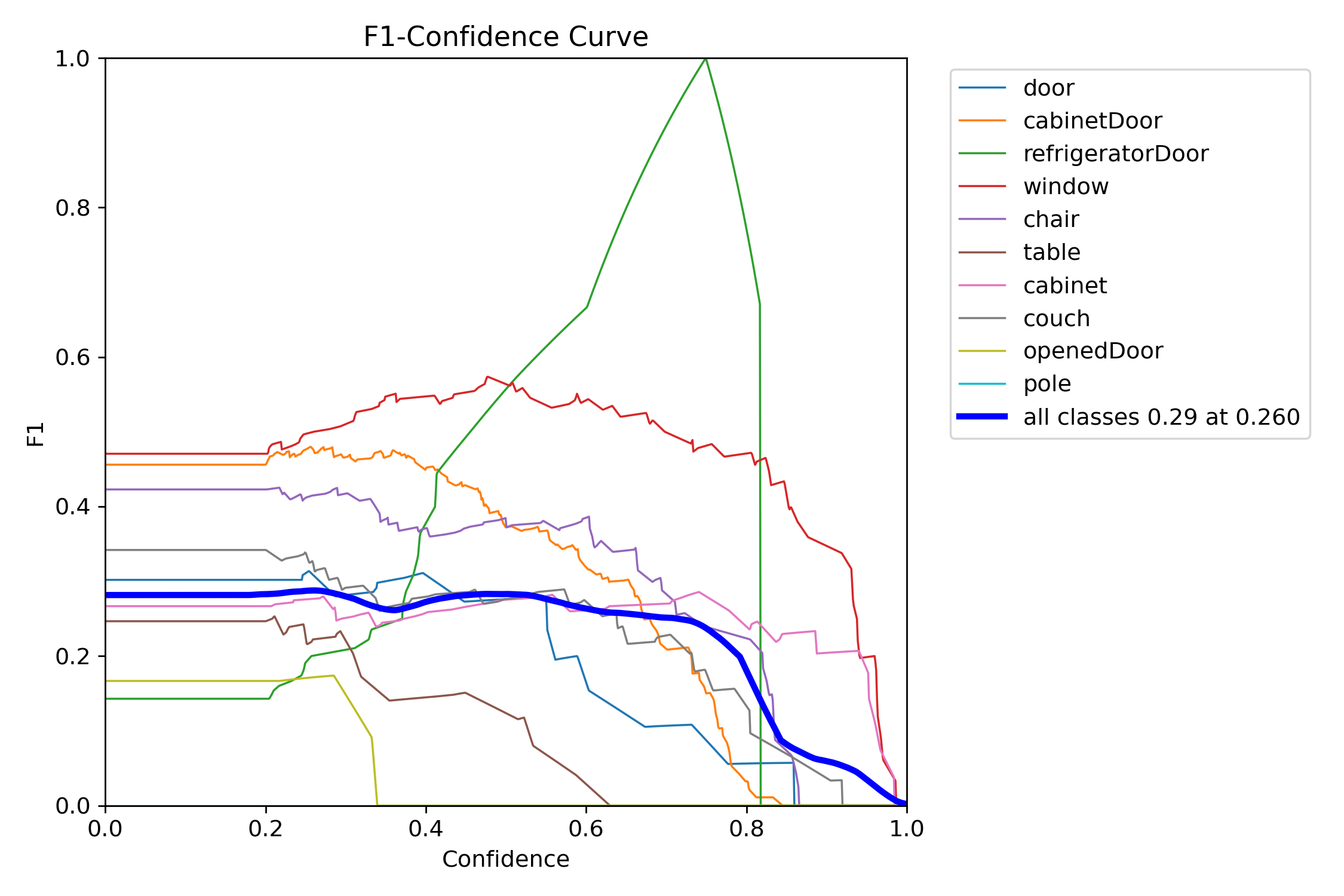
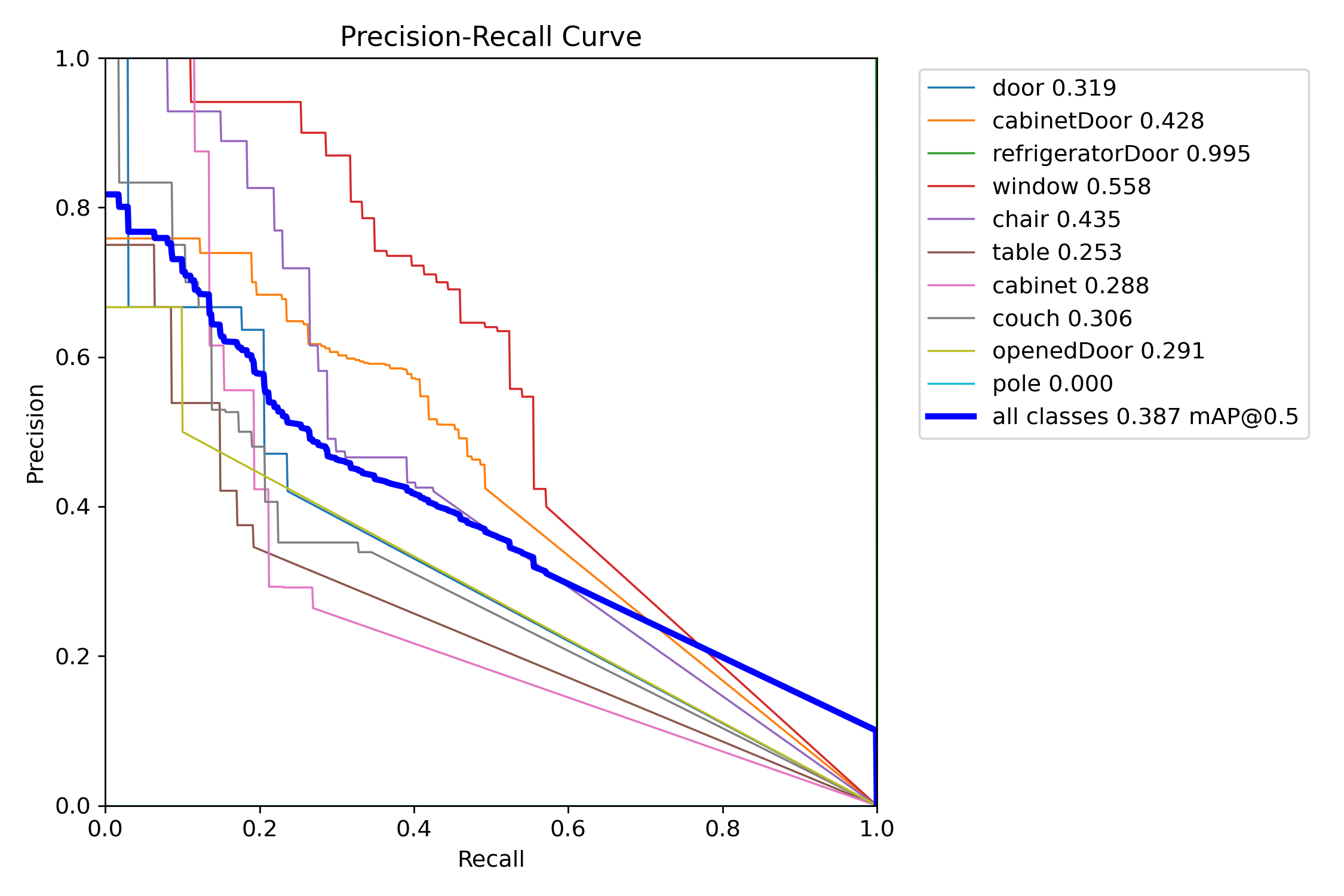
در این روش از کلیه داده­های مربوط به مجموعه داده RESNET-50 INDOOR OBJECT DETECTION استفاده شده است. بالاترین دقت از میان کلیه آموزش­های انجام شده مربوط به زیر می­باشد:

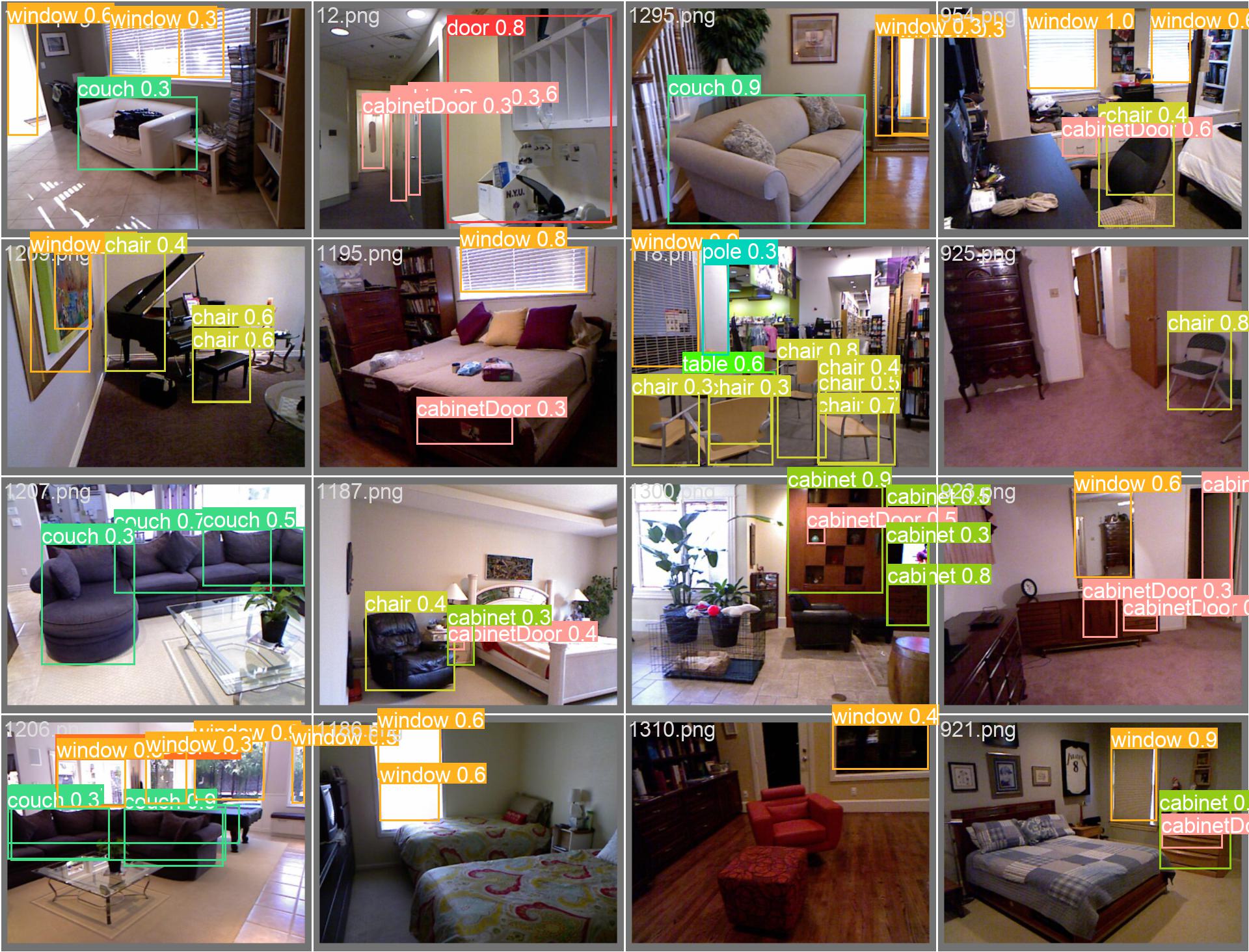
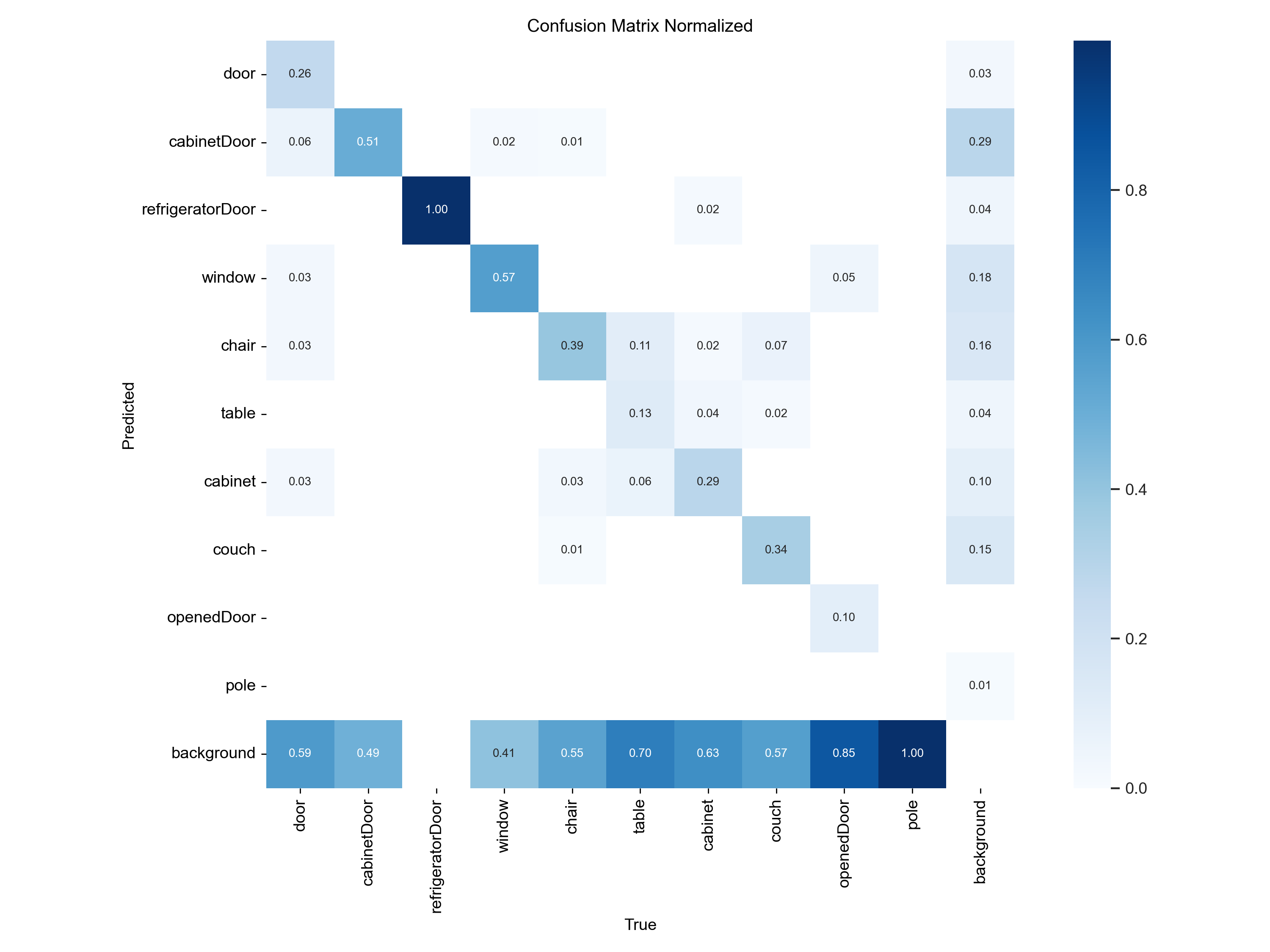








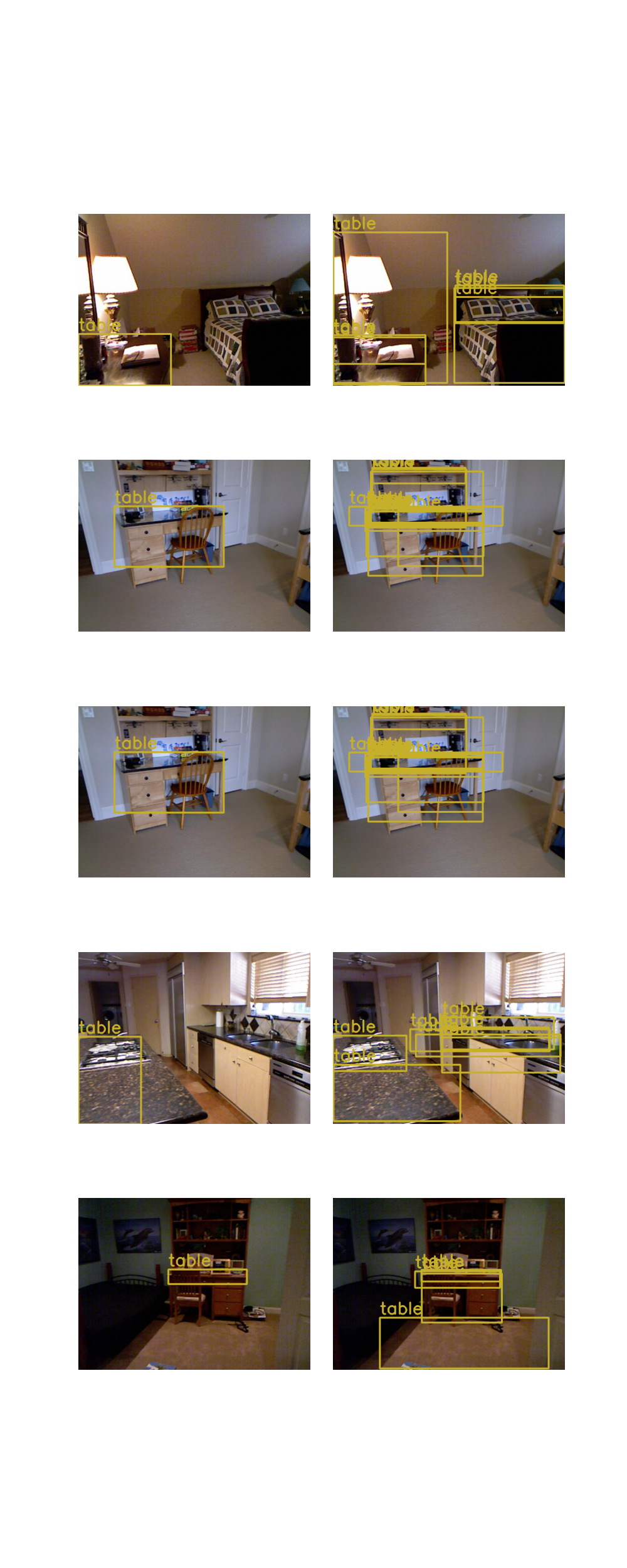


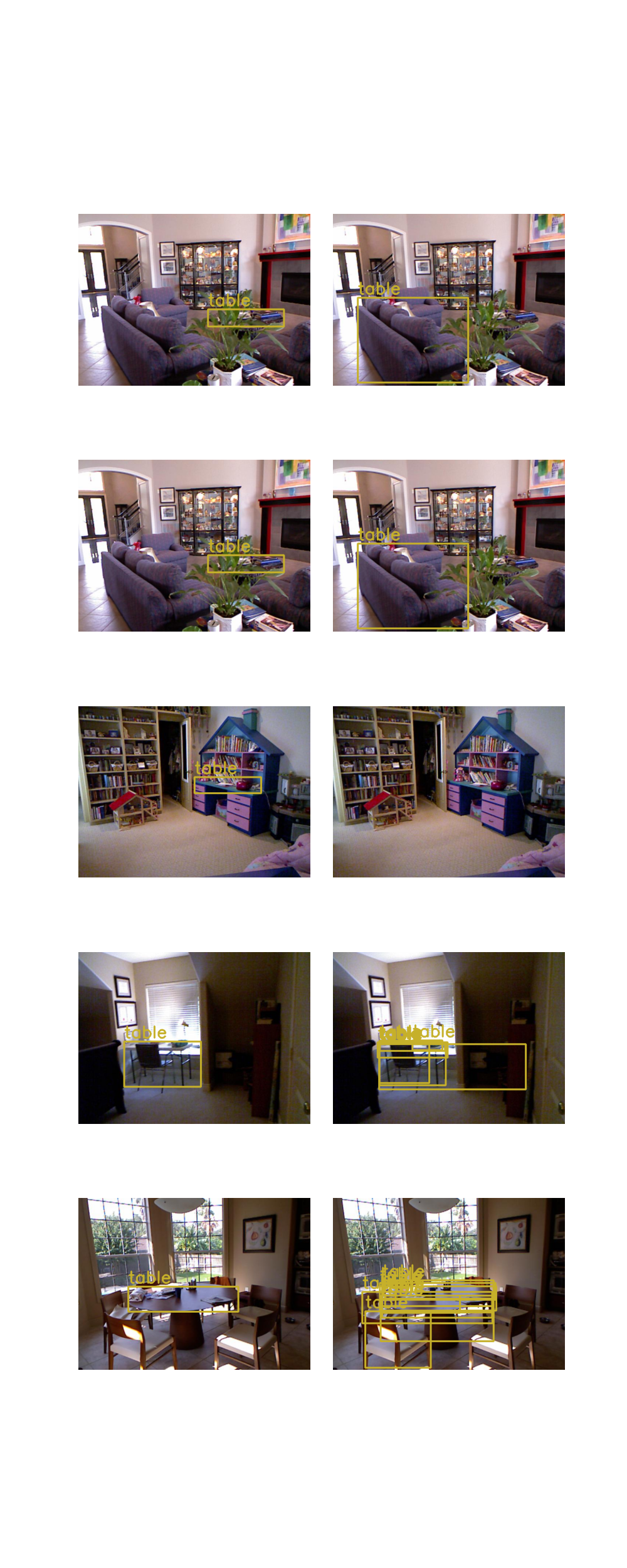


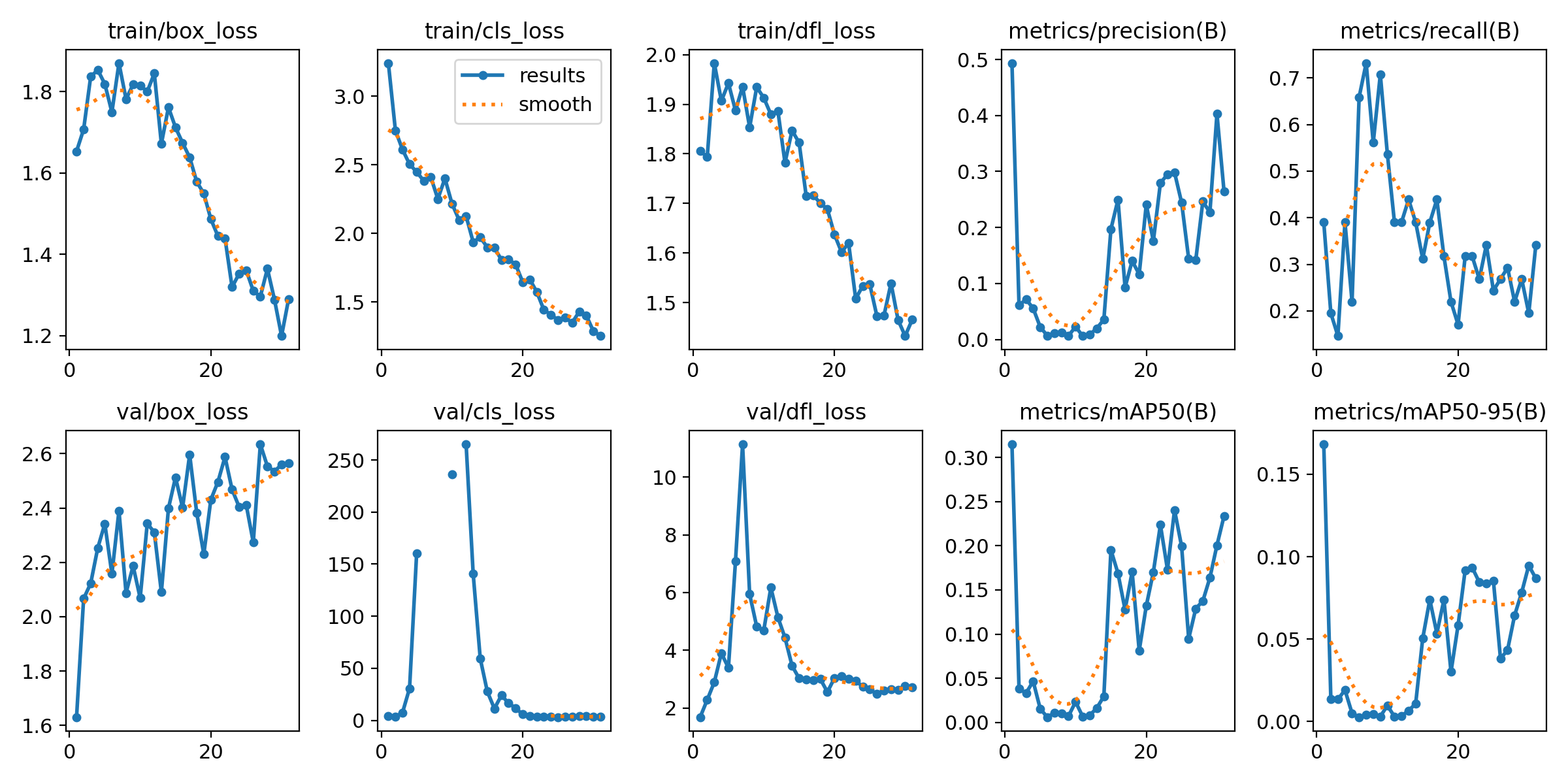
### روش دوم

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









1. point cloud data [↑](#footnote-ref-1)
2. local neighborhoo [↑](#footnote-ref-2)