Chi Ho Chan
September 15, 2025

Evaluation Report

1. Dataset and Splits

A 3-class subset of COCO 2017 (`person`, `car`, `bus`) was constructed by removing grayscale images and updating annotations. The original COCO17 *val* split was adopted as the *test* set. From the original *train* split, development splits were derived: approximately ∼500 *images per class* for training and ∼50 *images per class* for validation, with no image overlap between splits. The subset is highly imbalanced (`person` ≫ `car` ≫ `bus`).

2. Training Budget and Limitations

Due to time constraints, training data were capped at approximately ∼500 *images per class* for the training split (noting possible cross-class overlap at the image level). The training schedule was limited to **30 epochs** for **RetinaNet** and **20 epochs** for **DETR**. As a result, the models should be considered *not fully converged*. Evidence for undertraining (e.g., continuing loss/metric improvements near the final epochs) can be inspected in the TensorBoard records. Consequently, the results reported here are best interpreted as a **preliminary experiment** that establishes trends and provides a baseline rather than final, fully optimised performance.

3. Models Compared

RetinaNet and **DETR** were evaluated, each using a **ResNet-50** backbone. Other training settings were kept as similar as practicable to ensure a fair comparison.

4. Metrics Used

The following metrics are reported:

- **Per-class PR curves at IoU = 0.50**, used for qualitative diagnosis (Fig. 1 and Fig. 2).
- **Per-class AP** and **Macro-mAP**. Following the COCO evaluation protocol, precision values are computed for each combination of IoU threshold, recall level, object category, object size range, and maximum-detection setting. Class AP is obtained by averaging valid precision values across recall and the ten standard IoU thresholds (0.50:0.95); Macro-mAP is then the unweighted mean across classes (Tables 1 and **??**).

Formally, COCO mAP is

$$\mathrm{mAP}_{\mathrm{COCO}} = \frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{T}\sum_{t=1}^{T} AP_c^{\mathrm{IoU}_t}\right),$$

with T=10 IoU thresholds (0.50–0.95). Macro-mAP is computed with the *same* per-class averaging over these IoU thresholds:

$$\text{Macro-mAP} = \frac{1}{C}\sum_{c=1}^{C} AP_c, \quad \text{with} \quad AP_c = \frac{1}{T}\sum_{t=1}^{T} AP_c^{\mathrm{IoU}_t}.$$

In practice, the standard COCO procedure is followed: for each category, all valid precision values across the ten IoU thresholds and the full range of recall levels are aggregated and averaged to obtain the category AP; a simple mean of the per-category APs then yields Macro-mAP.

5. Model Comparison

5.1. **Qualitative (PR Curves at IoU = 0.50).** Figures 1 and 2 summarise class-wise precision–recall behaviour for RetinaNet and DETR. In Fig. 1 (b), horizontal-flip test-time augmentation (TTA) produces a small but consistent uplift: the `person` and `car` curves stay higher in the low-to-mid recall range, while `bus` remains strong to similar recall. The close similarity between Fig. 1 (a) (validation) and Fig. 1 (c) (test) indicates good generalisation with only modest degradation. For DETR, Fig. 2 (a) (validation) and Fig. 2 (b) (test) exhibit very similar curve shapes and class ordering, also suggesting limited overfitting and stable generalisation.

5.1.1. *Overall Observations.* Across both detectors, `car` is consistently the hardest class (early precision drop), while `bus` maintains higher precision until mid recall. RetinaNet tends to extend the recall tail for `person` (consistent with dense anchor coverage), whereas DETR yields cleaner, high-precision behaviour for larger rigid objects. These trends align with the per-class AP and Macro-mAP in the tables, and the RetinaNet TTA gain is reflected quantitatively as well.

5.2. **Quantitative (AP / Macro-mAP).** Tables 1 and **??** report Macro-mAP (equal class weight) and per-class AP. Macro-mAP prevents the abundant `person` class from dominating the aggregate, making performance on `car` and `bus` visible.

5.2.1. *Key Findings.*

- **Best model.** *RetinaNet_eva_TTA (hflip)* achieves the highest Macro-mAP (bold in Table 1).
- **Generalisation.** Validation (*eva*) scores are very close to test scores for both detectors; the ranking is preserved, indicating good generalisation.
- **Export fidelity.** ONNX evaluation results are within small deltas of the corresponding PyTorch (*eva*) results, indicating faithful conversion (minor differences arise from score rounding and NMS ordering).
- **Metric consistency.** Changes in Macro-mAP align with per-class AP (Table **??**) and the PR curves (Figs. 1, 2), supporting Macro-mAP as a reliable single-number criterion on this imbalanced 3-class subset.

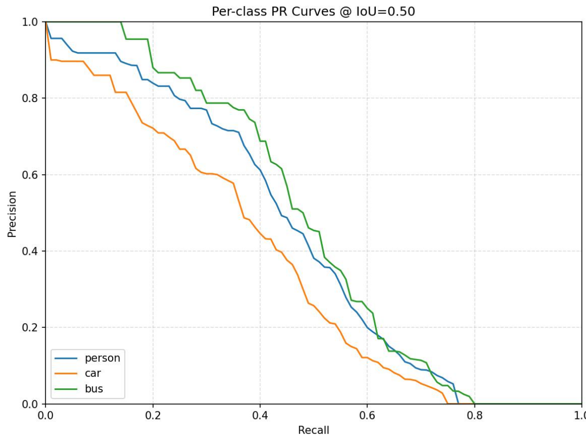
5.3. **Qualitative Examples.** Figure 3 contrasts RetinaNet (top row) with DETR (bottom row). Ground truth boxes are shown in green; PyTorch outputs in orange; corresponding ONNX exports in blue. The exports closely track their PyTorch counterparts (differences mainly from score rounding and NMS ordering). RetinaNet produces tighter, denser hypotheses in crowded scenes (e.g., `person`), occasionally yielding overlaps/duplicates in the `car` scene. DETR provides cleaner one-per-object predictions on larger rigid objects (`car`, `bus`), maintaining higher precision but appearing more conservative on small persons. These visual trends are consistent with the PR curves and Macro-mAP.

6. Justification for the Chosen Model

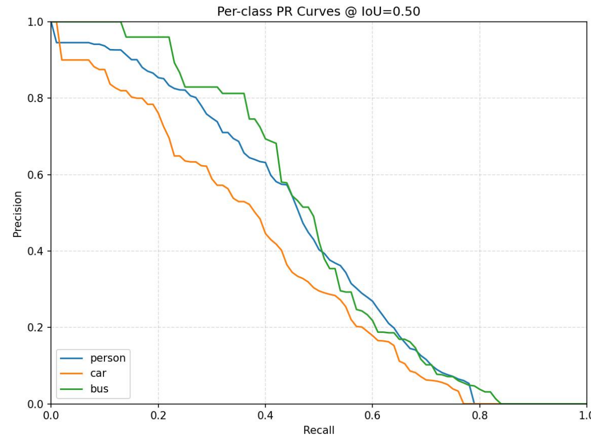
Model selection balances aggregate performance and class-specific behaviour observed in the PR curves:

1) **Priority on large rigid objects (e.g., `bus`) with precise localisation at mid recall:** DETR exhibits more stable precision.
2) **Priority on high recall for numerous small/medium `person` instances:** RetinaNet provides longer recall tails.
3) **Under class imbalance:** Macro-mAP is used as the decision metric because it gives each class equal influence while still enforcing IoU matching at 0.50–0.95, avoiding over-optimisation for `person`.

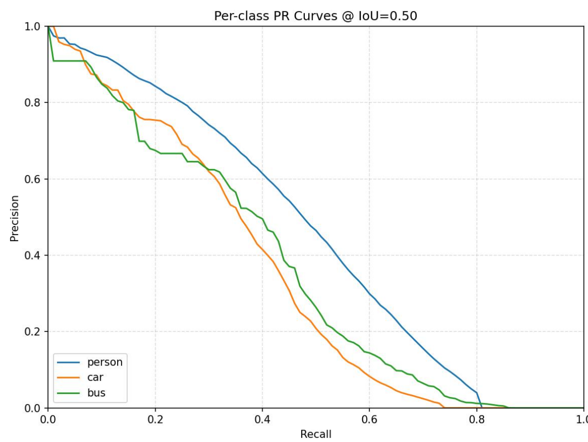
The preferred model is the one that maximises Macro-mAP on the held-out test set, with ties broken in favour of the model that better serves the application's target class. When `bus` performance is prioritised, DETR's stronger mid-recall precision and competitive Macro-mAP make it preferable; when `person` coverage is paramount, RetinaNet may be favoured.



(A) Evaluation

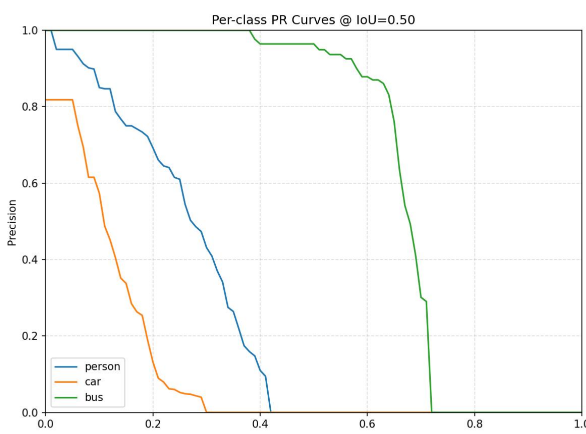


(B) Evaluation_tta

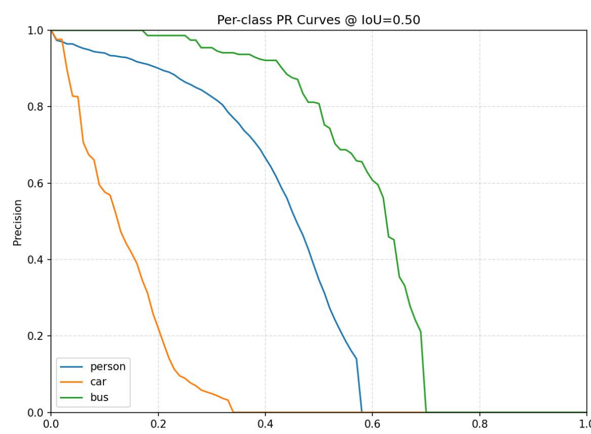


(C) Test

FIGURE 1. Per Class PR Curves of RetinaNet at IOU =0.5



(A) Evaluation



(B) Evaluation_tta

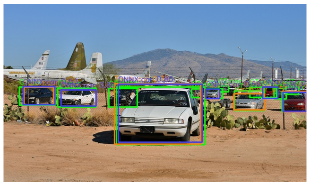
FIGURE 2. Per Class PR Curves of DeTr at IOU =0.5

Table 1. macro_mAP of Models over the validation set of COCO 17 datasets.

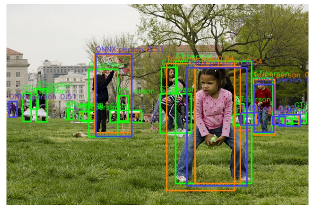
	macro_mAP
RetinaNet_eva	0.1858
RetinaNet_onnx_eva	0.1828
RetinaNet_eva_TTA(hflip)	**0.1983**
RetinaNet_Test	0.1648
DeTr_onnx_eva	0.1637
DeTr_onnx_Test	0.1819

Table 2. Model performances of each class over the validation set of COCO 17 datasets.

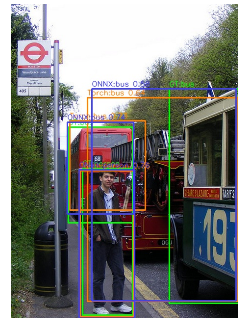
	PerClass	AP	AP50	APS	APM	APL
3*RetinaNet_eva	person	0.1715	0.433	0.1444	**0.2475**	0.2317
	car	0.153	0.3531	0.13	0.2295	0.1943
	bus	0.2328	0.4652	**0.0338**	0.1077	0.3712
3*RetinaNet_onnx_eva	person	0.172	0.4387	0.1429	0.2434	0.2373
	car	0.1542	0.3548	0.1293	0.232	0.2243
	bus	0.2222	0.4613	0.0781	0.0949	0.3577
3*RetinaNet_eva_TTA(hflip)	person	**0.1764**	**0.4454**	**0.1547**	0.2462	**0.2363**
	car	**0.1673**	**0.368**	**0.1447**	**0.2569**	0.1854
	bus	**0.2512**	**0.475**	0.0306	**0.1211**	**0.4004**
3*RetinaNet_Test	person	0.1645	0.4563	0.1435	0.1971	0.1974
	car	0.1528	0.346	0.1261	0.2306	0.1735
	bus	0.1772	0.3609	0.0525	0.096	0.2516
3*DeTr_onnx_eva	person	0.0876	0.2585	0.0099	0.1228	0.3683
	car	0.0376	0.1172	0.0018	0.0657	0.3144
	bus	0.366	0.66	0.0195	0.147	0.5999
3*DeTr_onnx_Test	person	0.1722	0.4199	0.0138	0.1863	0.3954
	car	0.0466	0.1357	0.0072	0.0504	0.3334
	bus	0.3267	0.5841	0.0067	0.0755	0.514



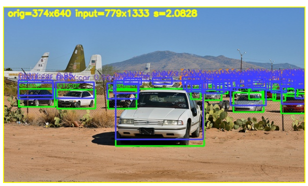
(A)
car$_{R}etinaNet$



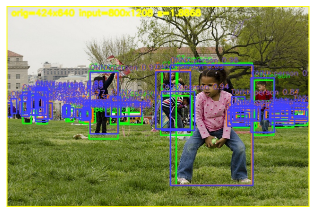
(B)
person$_{R}etinaNet$



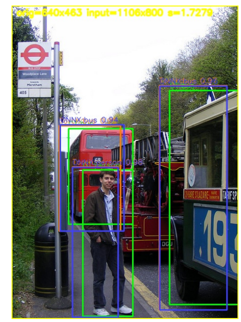
(C)
bus$_{R}etinaNet$



(D)
car$_{D}eTr$



(E)
person$_{D}eTr$



(F)
bus$_{D}eTr$

FIGURE 3. Examples of bounding boxes from groundtruth (Green), pytorch(Orange), Onnx(Blue)