

Чекпоинт 5

Тема 27

Детекция объектов методами глубинного обучения



Состав команды:
Зверев Дмитрий

Куратор:
Кирилл Козлов

Введение

В качестве базовой модели (baseline) была использована **YOLOv5m**.

Она была обучена на 28 эпохах, но полученные метрики оказались не самыми хорошими.

- mAP@0.5: 34.9%
- mAP@0.5:0.95: 25.7%
- Precision: 40.8%
- Recall: 36.4%

Чтобы улучшить качество предсказаний, было принято решение первым этапом обработать датасет.

Обработка датасета

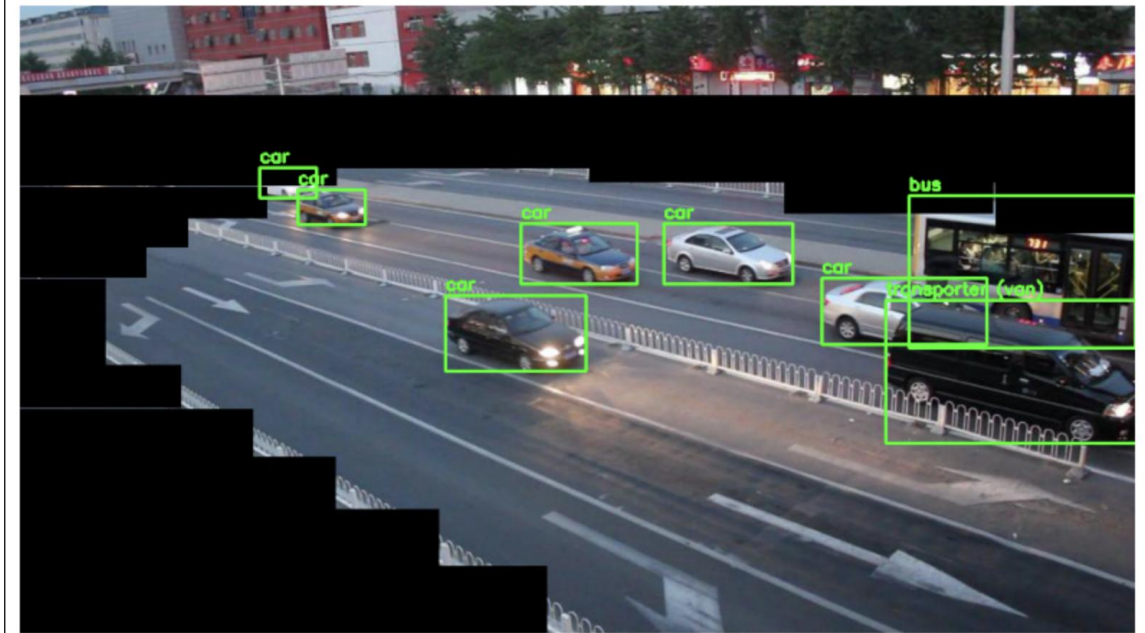
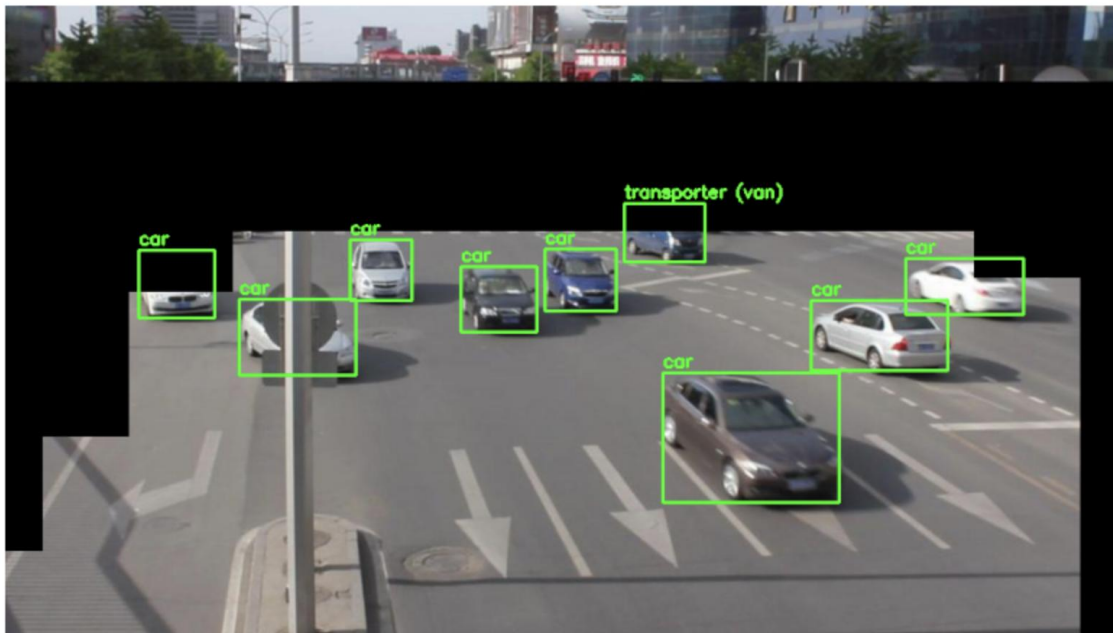
Основные классы в исходном датасете которые были размечены

- 1: bicycle
- 2: motorcycle
- 3: car
- 4: transporter (van)
- 5: bus
- 6: truck (others)
- 7: trailer (отсутствует в датасете)
- 8: unknown
- 9: mask

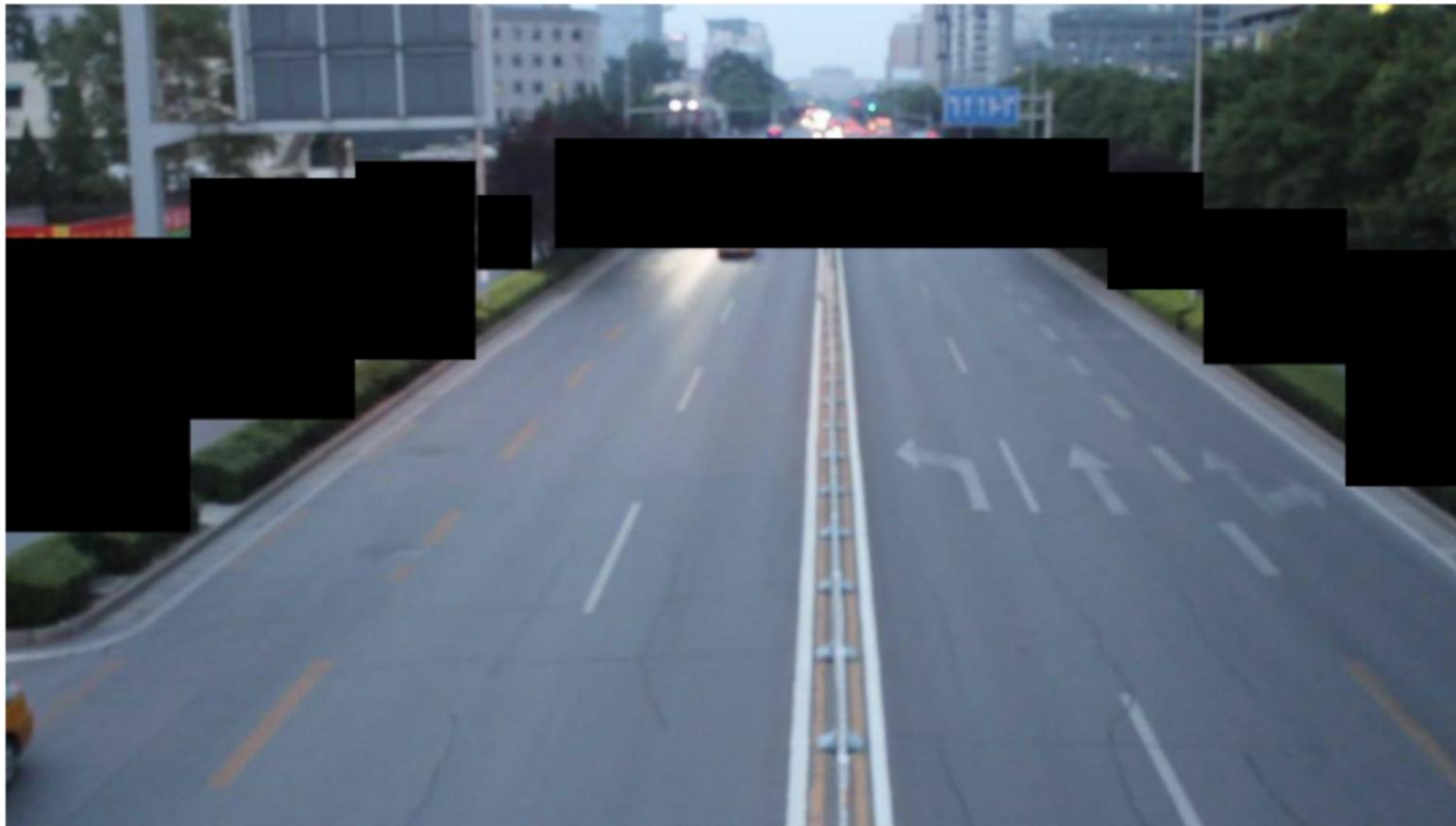
При анализе аннотаций я заметил, что классы `mask` (9) и `unknown` (8) могут негативно влиять на обучение. В описании датасета указано, что эти классы были размечены неполноценно или временно, что могло привести к проблемам в детекции.

Чтобы исключить их влияние:

- Объекты классов `mask` и `unknown` были закрашены чёрным на изображениях.
- Эти классы удалены из `.txt` аннотаций.
- Класс `trailer` (7) также удалён за ненадобностью.



Так же были обнаружены и удалены изображения и аннотации, на которых были размечены только классы mask и unknown.



Распределение классов после очистки

После обработки датасета я проанализировал распределение оставшихся классов. Выяснилось, что они распределены неравномерно и это может негативно сказаться на качестве обучения модели.

Чтобы сбалансировать тренировочный набор и быстрее проверить гипотезу, я решил ограничить датасет 20 000 изображениями, отобранными по приоритету классов с наименьшей представленностью.

Для этого использовалась формула с коэффициентом 0.9, что означало:

- 90% изображений содержат классы с наименьшей представленностью
- 10% изображений выбраны случайным образом из оставшегося датасета.

Распределение классов после отбора:

- bicycle : 1430
- motorcycle : 2559
- car : 129784
- transporter (van): 17270
- bus : 12555
- truck : 1185

Логично, что класс “car” по-прежнему преобладает, однако мне удалось выделить больше изображений с “transporter (van)”.

Изначально YOLO не умеет различать этот класс, так как по умолчанию определяет его как “car”. Благодаря новому балансу данных, мы можем эффективно обучить модель правильно детектировать все классы нашего датасета.

Начало обучения

Для обучения была выбрана модель **YOLOv12m**, которая обладает хорошим балансом между скоростью и точностью.

Перед обучением была проведена валидация на необученной модели, и получены следующие метрики:

- mAP@0.5: 15.7%
- mAP@0.5:0.95: 10.8%
- Precision: 30.0%
- Recall: 23.8%

Низкие метрики на старте - это логично. Модель неверное детектирует некоторые классы.

Была запущена тренировка на 30 эпохах с следующими параметрами:

- epochs=30,
- imgsz=640,
- batch=32,
- device=0,
- amp=True

После обучения **YOLOv12m** получила значительное улучшение метрик:

Class	Images	Instances	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
all	3359	33047	0.611	0.51	0.545	0.422
bicycle	294	296	0.0533	0.0439	0.0187	0.00647
motorcycle	549	756	0.229	0.373	0.209	0.114
car	3275	26104	0.925	0.828	0.907	0.701
transporter (van)	1434	2138	0.857	0.684	0.806	0.657
bus	2224	3459	0.934	0.809	0.911	0.712
truck (others)	271	294	0.664	0.322	0.417	0.339

Выводы по результатам обучения

Существенный рост точности и полноты

- Общий mAP@0.5 вырос с 15.7% до 54.5%
- mAP@0.5:0.95 улучшился с 10.8% до 42.2%

Проблемы с “bicycle” и “motorcycle”

- Возможная причина низких метрик → маленький размер объектов, из-за чего YOLO плохо их детектирует на `imgsz=640`.
- Возможное решение → попробовать увеличить разрешение изображений.
- Возможное решение → попробовать подобрать гиперпараметры на предикте (Confidence Threshold, Intersection Over Union Threshold)
- Возможное решение → добавление данных с минорными классами

Успешное обучение класса “transporter (van)”

- После обучения mAP@0.5 для transporter составил 80.6%, а mAP@0.5:0.95 – 65.7%, что очень хорошо для урезанного датасета всеголишь на 30 эпохах

Общие выводы

После обучения YOLOv12m на очищенном и сбалансированном датасете удалось значительно улучшить метрики по сравнению с исходной моделью.

Model	mAP@0.5	mAP@0.5:0.95	Precision	Recall
YOLOv5m	34.9	25.7	40.8	36.4
YOLOv12m (до)	15.7	10.8	32.0	23.6
YOLOv12m (после)	54.5	42.2	61.1	51.0

Следующими этапами будут:

- Оптимизация детекции мелких объектов (“bicycle”, “motorcycle”)
- Увеличение датасета для улучшения качества модели
- Продолжение обучения на большем количестве эпох