

## Лучшая модель:

В качестве итогового решения использовалась модель **You Only Look at One Sequence: Rethinking Transformer in Vision through Object Detection** (<https://arxiv.org/abs/2106.00666>). Была использована реализация tiny модели с huggingface (<https://huggingface.co/hustvl/yolos-tiny>), предобученная на датасетах ImageNet-1k и COCO 2017 object detection. В качестве лосса использовался лосс, встроенный в модель, т.е. комбинация bipartite matching, cross-entropy,  $\ell_1$  и generalized IoU. Параметры трейна: batch\_size=2, оптимизатор - AdamW, lr - 4 эпохи с lr=2.5e-5 и затем 10 эпох с lr=2.5e-6. Логи находятся по ссылке: <https://wandb.ai/azatiussssss/uncategorized/reports/Yolos-tiny--VmldzozMDQ5ODM2?accessToken=050r03r6h1yp10l0ek1fo6z0xy4apkc4vj0vlcyaur80xgklvdk9i2ushytf8hi8>

## Воспроизведение:

Либо запустить ячейки ноутбука my\_train.ipynb в соответствии с гиперпараметрами обучения, указанными выше, за основу взят ноутбук- пример, и для работы с данными и подсчета метрик используется такой же код с точностью до мелких изменений, датасет самописный, он наследуется от датасета из основного репо, поменял только конструктор немного и получение элемента по индексу, чтобы совместить с пайплайном для transformer, само обучение производится при помощи pytorch lightning, поэтому там все довольно лаконично; либо можно загрузить модель из приложенного чекпоинта model\_checkpoint.pth и использовать его.

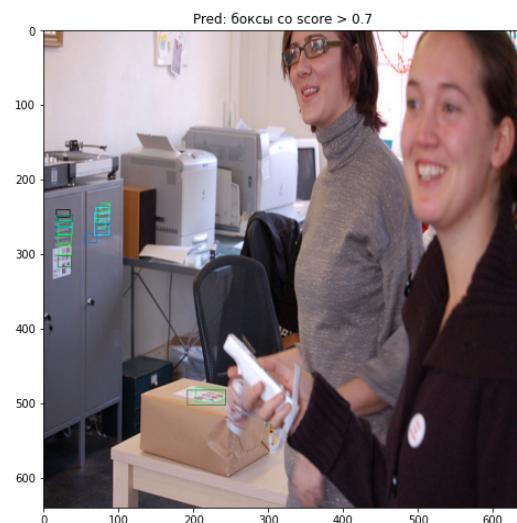
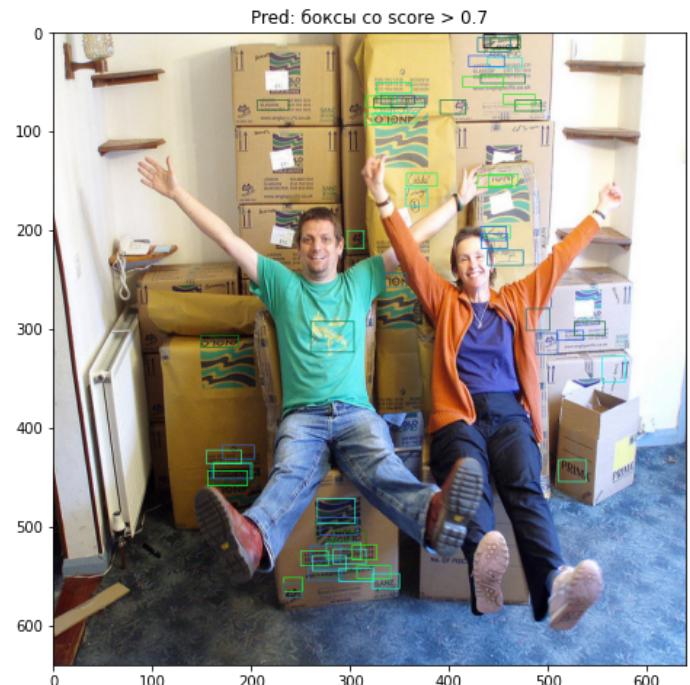
## Эксперименты:

Во-первых, в рамках обучения трансформера, описанного выше, пробовал добавить аугментации в данные (GaussianBlur и ColorJitter) и дообучить модель, но это не сильно помогло, итоговое значение метрики получилось порядка 30.7. Также пробовал уменьшать лосс и прогнать еще несколько эпох, но это опять же ни к чему существенному не привело, значение метрики колебалось в окрестности 30. Аналогично попытка тюнить отдельно классификатор и регрессор боксов не принес положительного прироста метрики. Я думаю, что модель просто уперлась в свой потолок, обучаемых параметров там достаточно мало (~ в 20 раз меньше, чем в yolos-base), надо попробовать обучить полную модель, когда будут ресурсы, думаю, что там будет качество значительно лучше. Также пробовал добавить ауги при обучении бейзлайна, но это только сломало входную модель и значение метрики получилось в районе единицы. Аналогичная ситуация была и с моделью SSDlite и значение метрики было вообще порядка десятых, но там была именно проблема с классификатором, потому что он “хитрил” и возвращал для всех боксов значение гиперпараметра positive\_fraction, судя по всему сильно влиял hard negative sampling, используемый при обучении модели, попытки уменьшать значение гиперпараметра не привели к улучшению итогового результата, при значении 1e-6 классификатор перестал выдавать одинаковый скор всем боксам, но значения были все равно так себе, несбалансированность классов, похоже, все-таки сильно сказывалась. Была

мысль добавить веса классов в кросс-энтропию, но лосс в реализации в torchvision захардкожен и менять source code локально я не решился.

## Ошибки модели:

Ошибка 1? Модель помечает как текст, причем довольно уверенно то, что отдалено визуально как-то похоже на текст, как, например, на изображении 3 или 2. Но, тут важно заметить, что не всегда модель ошибается, иногда есть и проблемы в разметке, например, на 1 изображении не на всех коробках текст помечен как текст, а вот модель его находит, но особенно ярко это видно на последнем изображении, где надпись "Cola" на борте автобуса почему-то не размечена, а вот модель её задетектила





Вторая проблема, которую я наблюдаю, заключается в том, что модель генерит очень много нерелевантных боксов с маленьким скором  $\sim < 0.05$ . Тут, возможно, дело в том, что модель пытается обмануть систему и получить маленький штраф за неправильный класс, и не получить меньший штраф за регрессию на боксах, но я не уверен

