

ВЫБОР МОДЕЛИ

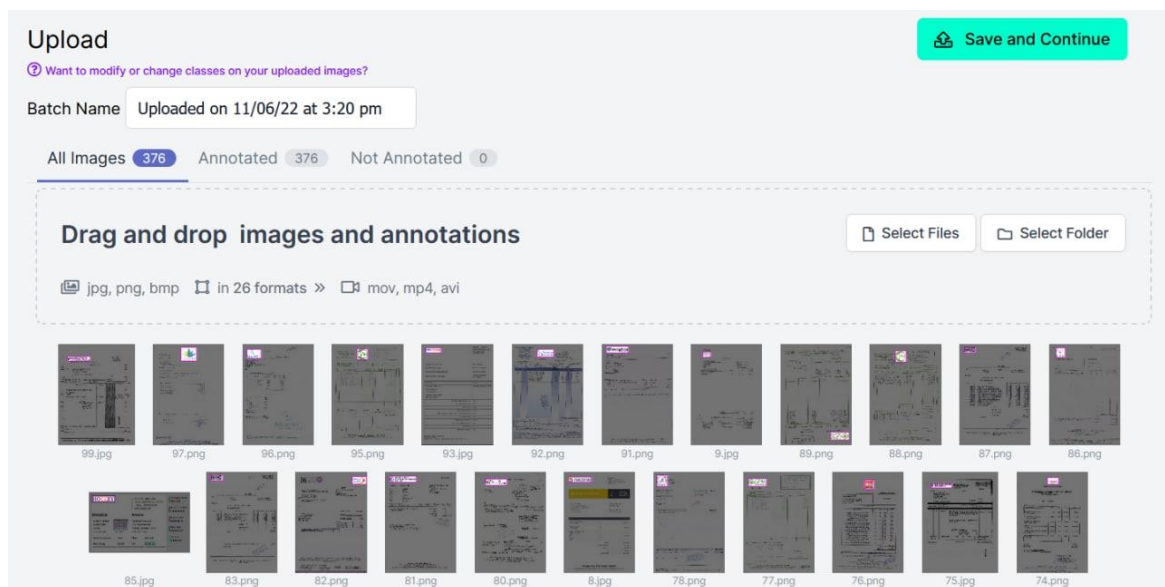
Я изучила некоторые методы, которые сейчас применяются в object detection (R-CNN, Fast R-CNN, YOLO), и решила остановиться на YOLOv5, так как именно для пятой версии нашла больше всего материалов для начинающих и подумала, что это будет хорошей стартовой точкой для экспериментов с object detection.

Модели семейства YOLO предобучены на датасете COCO, что не имело особого смысла для задачи распознавания логотипов. Необходимо было переобучить под распознавание другого класса объектов.

СБОР ДАННЫХ

Я не нашла в открытом доступе наборов изображений invoices (накладных? не уверена насчет самого корректного перевода) именно для распознавания логотипов. Самое близкое, что мне попадалось, — наборы данных для классификации документов на Kaggle. Часть данных я взяла оттуда. ([этот](#) датасет и [этот](#))

Остальные добирались из Гугл-Картинки. Полученные ~350 изображений я разметила с помощью Labelimg, затем выгрузила данные в Roboflow. Часть изображения я оставила неразмеченными для тестирования обученной модели.



В Roboflow для них можно было настроить предобработку и некоторые модификации и таким образом я расширила датасет до порядка 900 изображений.

Generating New Version





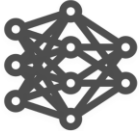
Prepare your images and data for training by compiling them into a version.
Experiment with different configurations to achieve better training results.

✓	Source Images	Images: 374 Classes: 1 Unannotated: 0
✓	Train/Test Split	Training Set: 263 images Validation Set: 75 images Testing Set: 36 images
✓	Preprocessing	Auto-Orient: Applied Resize: Stretch to 500×500
✓	Augmentation	Flip: Horizontal, Vertical 90° Rotate: Clockwise, Counter-Clockwise, Upside Down Rotation: Between -3° and +3° Shear: ±5° Horizontal, ±5° Vertical Blur: Up to 1px Noise: Up to 1% of pixels

НАСТРОЙКА МОДЕЛИ

Чтобы переобучить YOLO на определение одного класс объектов (логотипы), я взяла исходный конфигурационный файл `yolov5s.yaml` и в его копии заменила количество классов на 1, затем обучила и получила набор лучших весов модели `best.pt`.

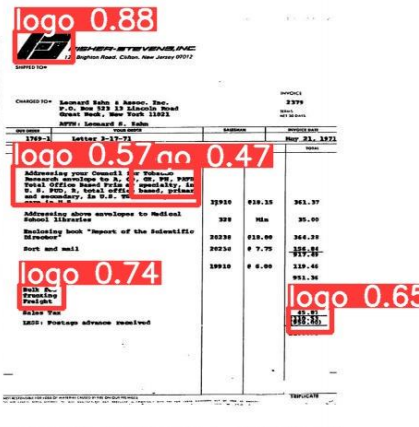
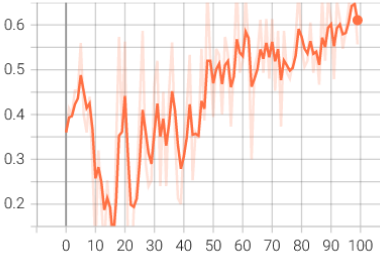
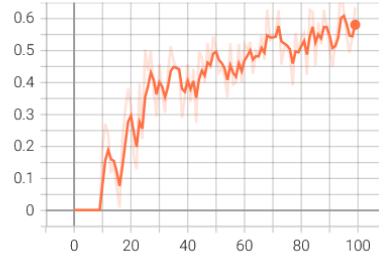
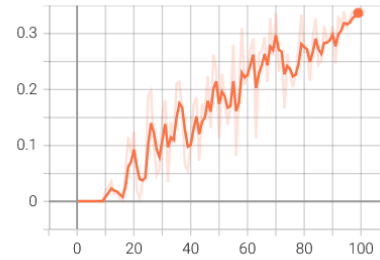
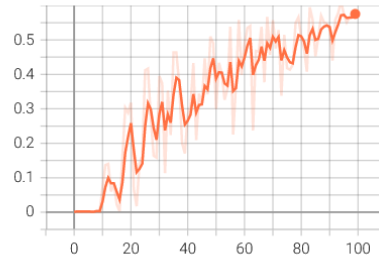
Я выбрала архитектуру `s`, так как она показалась мне хорошим компромиссом между количеством данных и классов объектов (набор данных небольшой, и только 1 класс) и характеристиками архитектур (`y m, l, x` лучше метрики, но и ресурсов они потребляют больше). Если обучение `s` не будет очень успешно, можно будет попробовать `m`.

				
Nano YOLOv5n	Small YOLOv5s	Medium YOLOv5m	Large YOLOv5l	XLarge YOLOv5x
4 MB _{FP16} 6.3 ms _{V100} 28.4 mAP _{COCO}	14 MB _{FP16} 6.4 ms _{V100} 37.2 mAP _{COCO}	41 MB _{FP16} 8.2 ms _{V100} 45.2 mAP _{COCO}	89 MB _{FP16} 10.1 ms _{V100} 48.8 mAP _{COCO}	166 MB _{FP16} 12.1 ms _{V100} 50.7 mAP _{COCO}

В первый раз я обучала с `--img 416 --batch 16 --epochs 100` и получила финальные метрики:

precision = 0.634 recall = 0.556

mAP = 0.591 mAP50-95 = 0.345



Самыми очевидными мерами по улучшению качества были 1) увеличить набор данных и 2) обучать подольше (100 эпох для такой архитектуры не так уж и много).

Изображения добывала из недр Каггла (нашла еще [это](#), [это](#) и [это](#)) и с помощью гугл-запросов типа “tax invoice” и “artist invoice templates”. Расширила до 500 уникальных изображений, на Roboflow различными модификациями сделала из них 1200.

На новом наборе, не меняя гиперпараметров обучения получила:

```
precision = 0.669 recall = 0.767
```

```
mAP = 0.742 mAP50-95 = 0.413
```

Метрики стали значительно выше, но при этом из 51 тестового изображения на 17 не были распознаны логотипы. Скорее всего, дело в высоком confidence threshold при предсказании (0.4).

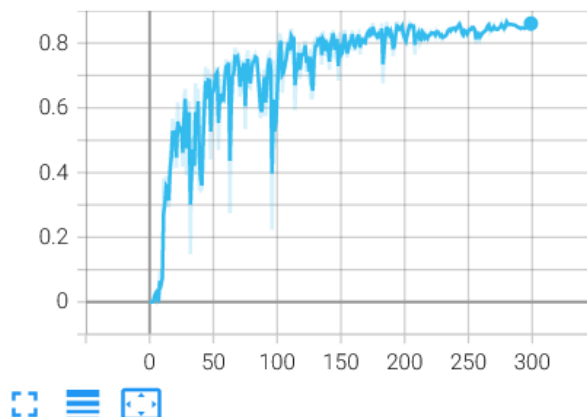
Затем я увеличила количество эпох до 300 и размер батча до 32:

```
precision = 0.857 recall = 0.777
```

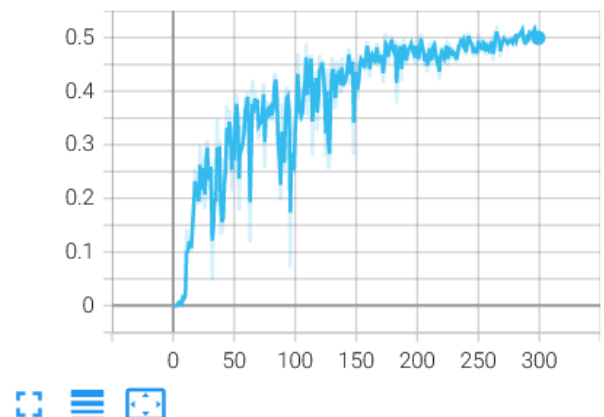
```
mAP = 0.858 mAP50-95 = 0.517
```

Значительно возросли precision и mAP.

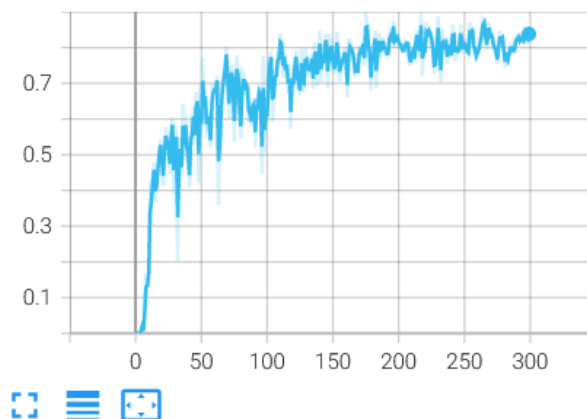
metrics/mAP_0.5
tag: metrics/mAP_0.5



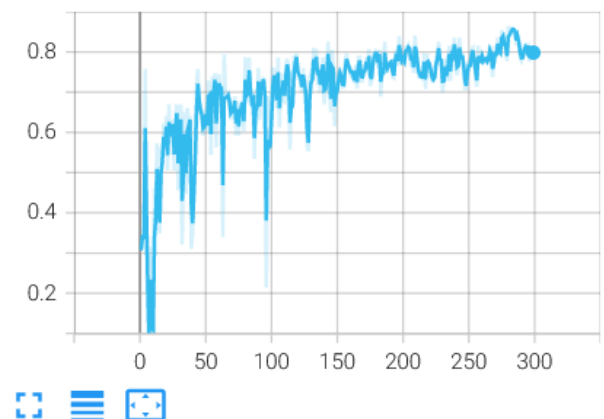
metrics/mAP_0.5:0.95
tag: metrics/mAP_0.5:0.95



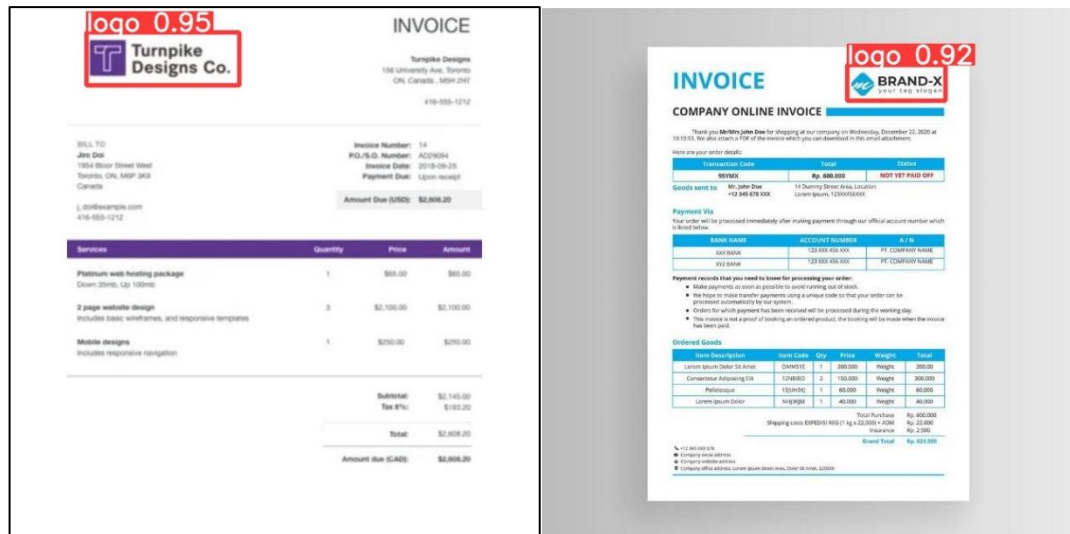
metrics/precision
tag: metrics/precision



metrics/recall
tag: metrics/recall



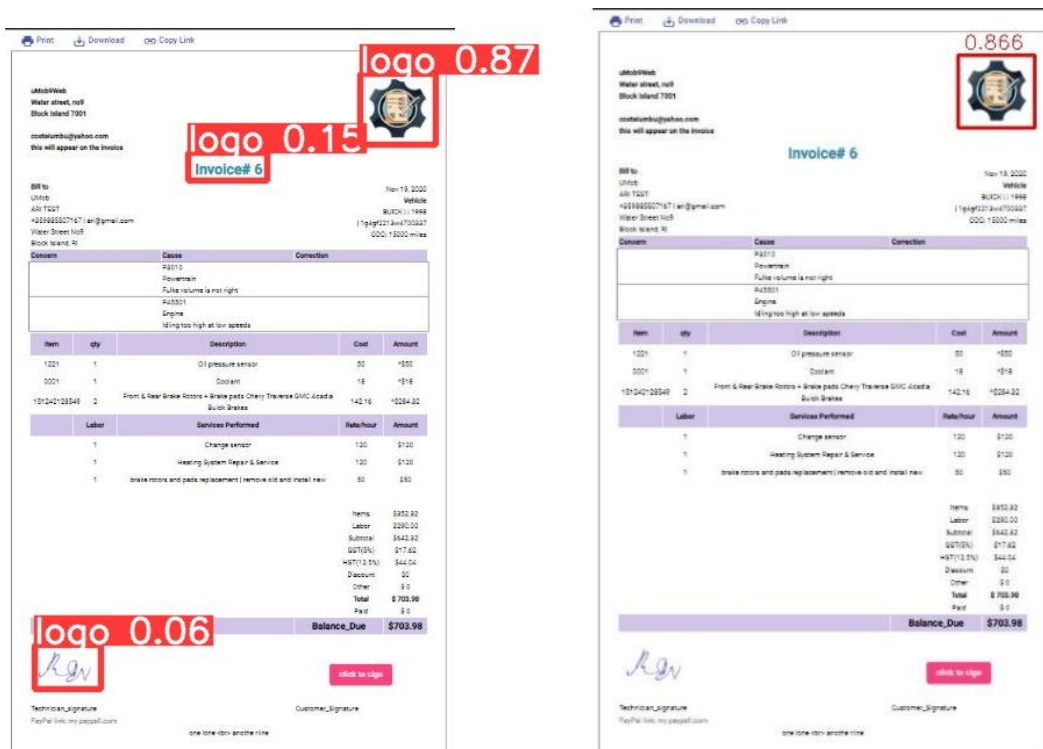
Доля изображений, где не было выявлено ни одного логотипа, уменьшилась. Также повысилась confidence на тестовых экземплярах, где и до этого успешно выявлялись логотипы.



В самом деле, проблема того, что на некоторых изображениях не выявлялось ни одного логотипа, решилась понижением confidence threshold при обучении. Но чтобы на всех тестовых экземплярах хоть что-то выявить, пришлось задать этому порогу практически нулевое значение.

В таком случае на изображениях, с которыми не было проблем, выявляются мнимые объекты, но при презентации результатов в таком случае можно рисовать только bbox с наибольшей confidence.

Для сравнения на 1-ом изображении то, что выдает модель при очень низком confidence threshold, а на 2-ом отрисована bbox только вокруг объекта с наибольшей confidence.



Обучение на 500 эпохах не дало серьезных улучшений, процесс остановился намного раньше:

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
338/499	9.1G	0.02099	0.006372	0	4	640: 100%
Class	Images	Instances	P	R	mAP50	mAP50-95: 100%
all	103	103	0.843	0.832	0.854	0.503

Stopping training early as no improvement observed in last 100 epochs. Best results observed at epoch 238, best model saved as best.pt.

339 epochs completed in 1.480 hours.

//todo попробовать еще больше расширить датасет ИЛИ обучить yolov5m