

Задача 1. Реализовать модуль генерации данных.

Задача решена средствами библиотек `skimage` (`random_shapes`) и `PIL` (`ImageDraw`). Тестовый набор в 100 записей был создан примерно за 2 секунды (CPU Core I9 11900K). Для работы модели генерировался дополнительно `Coco Annotation Dataset`. Отдельно для задачи 2 создан пример динамического датасета с созданием фигур на лету.

Задача 2. Детекция примитивных геометрических фигур на изображении.

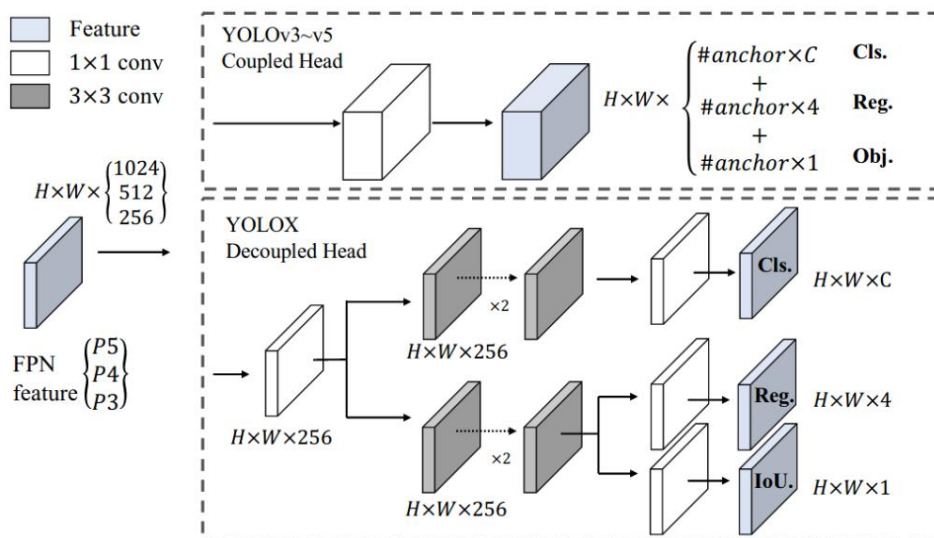
2.1. Описание модели.

В качестве платформы для построения модели была выбрана открытая библиотека `MMDetection 3.0`. `MMDetection` – это набор инструментов, который содержит богатый выбор методов обнаружения объектов, сегментации экземпляров и паноптической сегментации. Огромным плюсом платформы является возможность быстро пробовать десятки моделей, каждая из которых является state-of-the-art в своей отрасли.

В качестве модели предлагается `Yolox-S`. Более старшая модель семейства `Yolox-L` заняла 1-е место в `Streaming Perception Challenge` (конкурс по автономному вождению на CVPR 2021) при использовании единственной (!) модели `Yolox-L`. В 2022 году также первое место в `ECCV 2022 Multiple People Tracking in Group Dance Challenge`. `Yolox-S` для детекции геометрических фигур более, чем достаточна, и гораздо экономичнее в ресурсах по сравнению с `Yolox-L`.

`Yolox` — это одноэтапный детектор объектов, который вносит несколько модификаций в `YOLOv3` с магистралью `DarkNet53`. `Yolox` — это относительно новый высокопроизводительный алгоритм обнаружения объектов `Yolo`. В отличие от предыдущих алгоритмов `YOLO`, он включает в себя три фундаментальных нововведения: `Decoupled head`, `Anchor Free`, `Advanced labels`

Благодаря этим нововведениям `Yolox` обеспечивает лучшие показатели по скорости и производительности, чем его аналоги.



С точки зрения параметра box AP модель Yoloх-L сохраняет одну из лидирующих позиций и сегодня. Именно по этому параметру были сравнены Yoloх, HTC и Mask2Former. Также модель Yoloх предлагает отличные методы аугментации (шире возможностей HTC, например, такие как Mosaic и MixUp), которые важны именно для детекции простых форм. Особенно важной оказалась роль аугментации изображения для учета краевого эффекта и детекции класса объекта. Так, при отключении ряда аугментаций почти не детектировались фигуры у краев (с минимальным зазором от края) и плохо различались маленькие фигуры.

В качестве набора аугментаций использовались следующие методы:

- Mosaic - ресайз и последовательная комбинация изображений в одно
- RandomAffine - перемещение, вращение или сдвиг.
- MixUp - наложение изображений.
- YOLOXHSVRandomAug - аугментация по тону, насыщенности, яркости
- RandomFlip - поворот.
- Resize - ресайз.
- Pad - добавление отступов.

2.2 Результаты

Модель успешно выполнила задачу детектирования объектов. Расчет каждой эпохи занимал около 10 минут (GPU RTX 3060). Минимально требуется 3 эпохи для обучения.

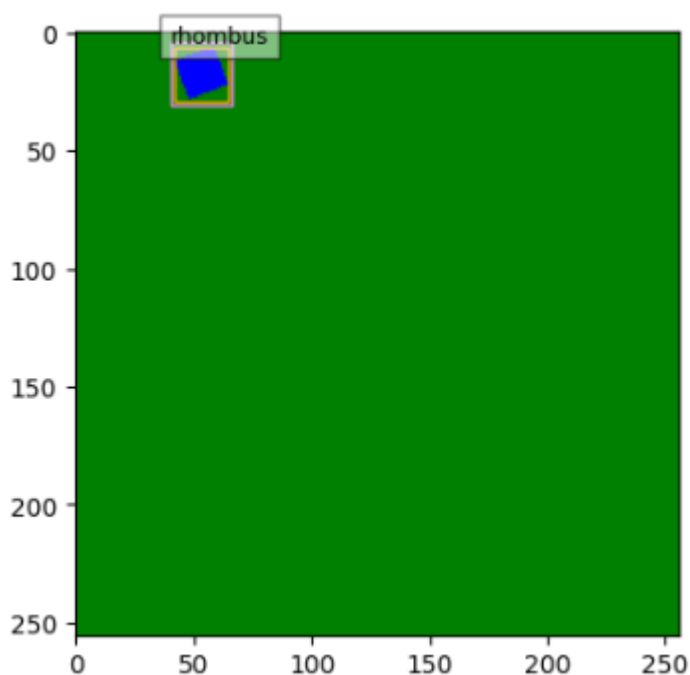
Average Precision (AP) @[IoU>0.50 | area= all | maxDets=1000] = **0.997**

Average Recall (AR) @[IoU>0.50 | area= all | maxDets=1000] = **1.00**

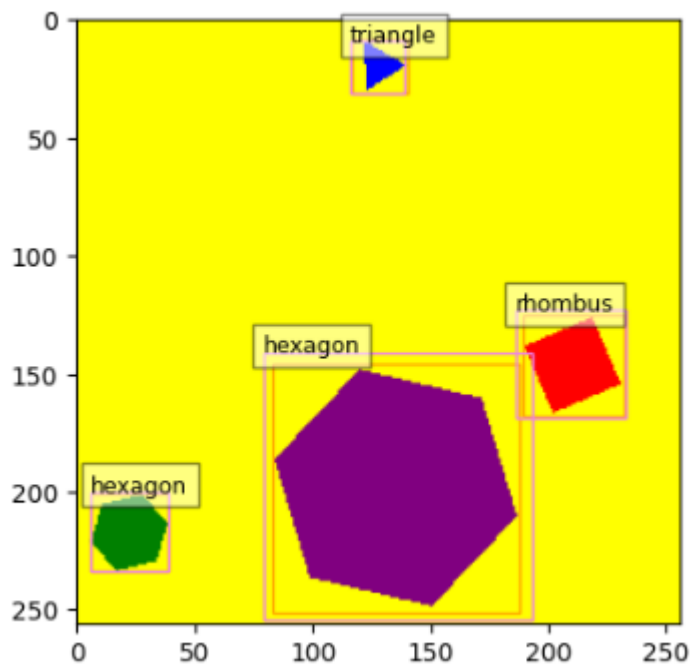
Total shapes (Ground Truth) {'rhombus': **740**, 'triangle': **729**, 'circle': **766**, 'hexagon': **757**}

Total shapes (Detected) {'rhombus': **740**, 'triangle': **729**, 'circle': **767**, 'hexagon': **757**}

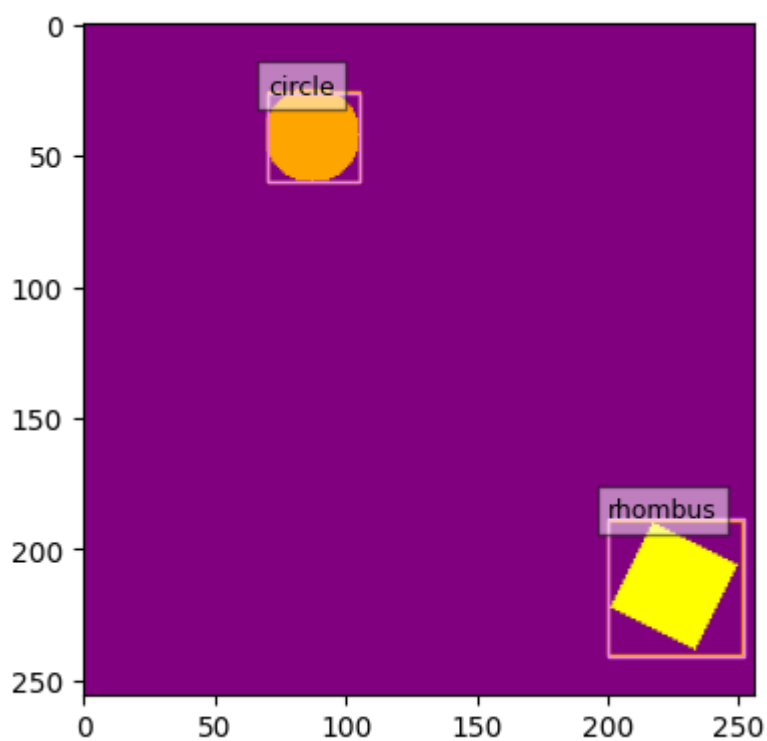
Пример iou min = 0.78



Пример iou mean = 0.971



Пример $\text{iou max} = 1.0$



Промежуточные результаты (после первой эпохи)

Average Precision (AP) @[IoU>0.50 | area= all | maxDets=1000] = **0.960**

Average Recall (AR) @[IoU>0.50 | area= all | maxDets=300] = **0.999**

Total shapes (Ground Truth) {'rhombus': **740**, 'triangle': **729**, 'circle': **766**, 'hexagon': **757**}

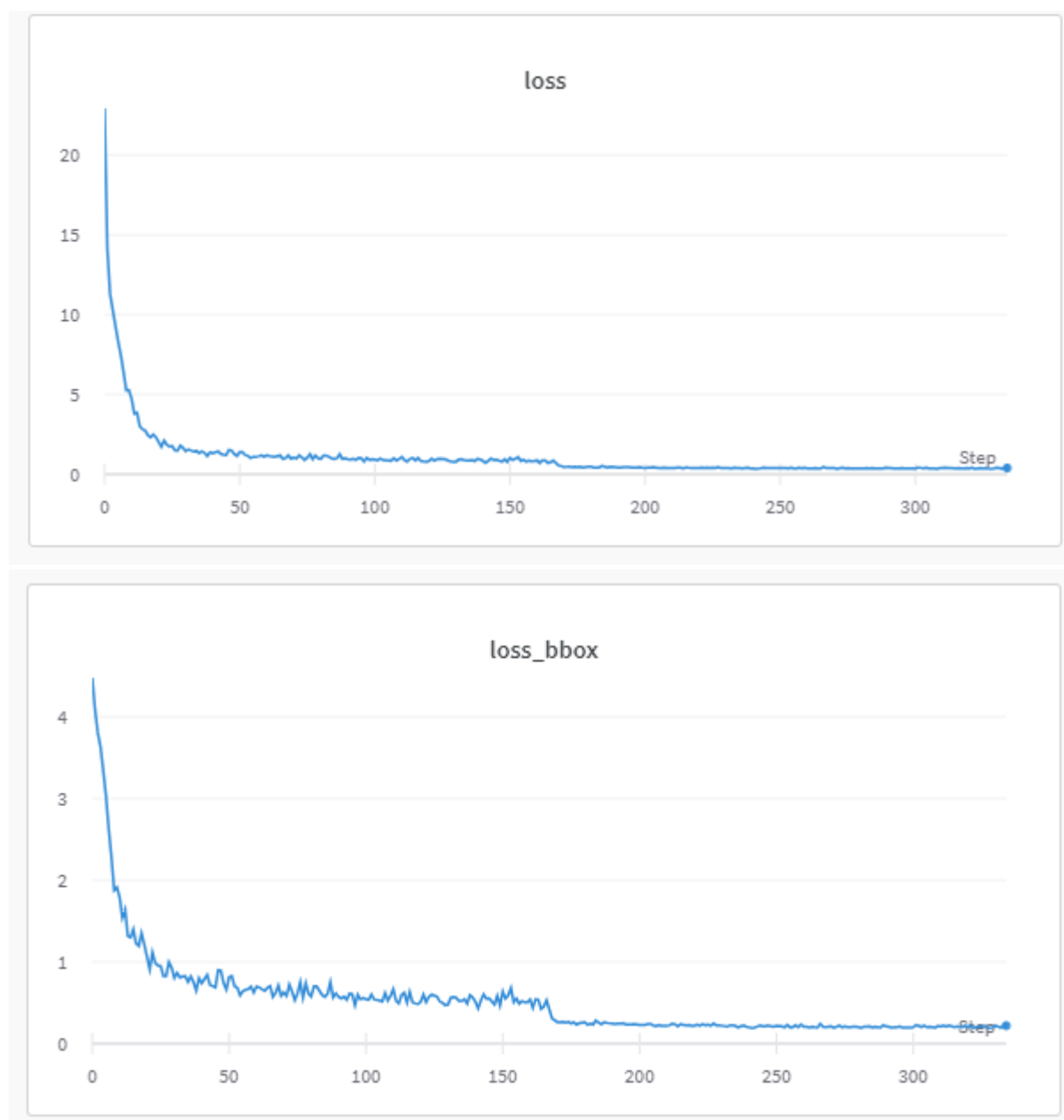
Total shapes (Detected) {'rhombus': **640**, 'triangle': **714**, 'circle': **885**, 'hexagon': **712**}

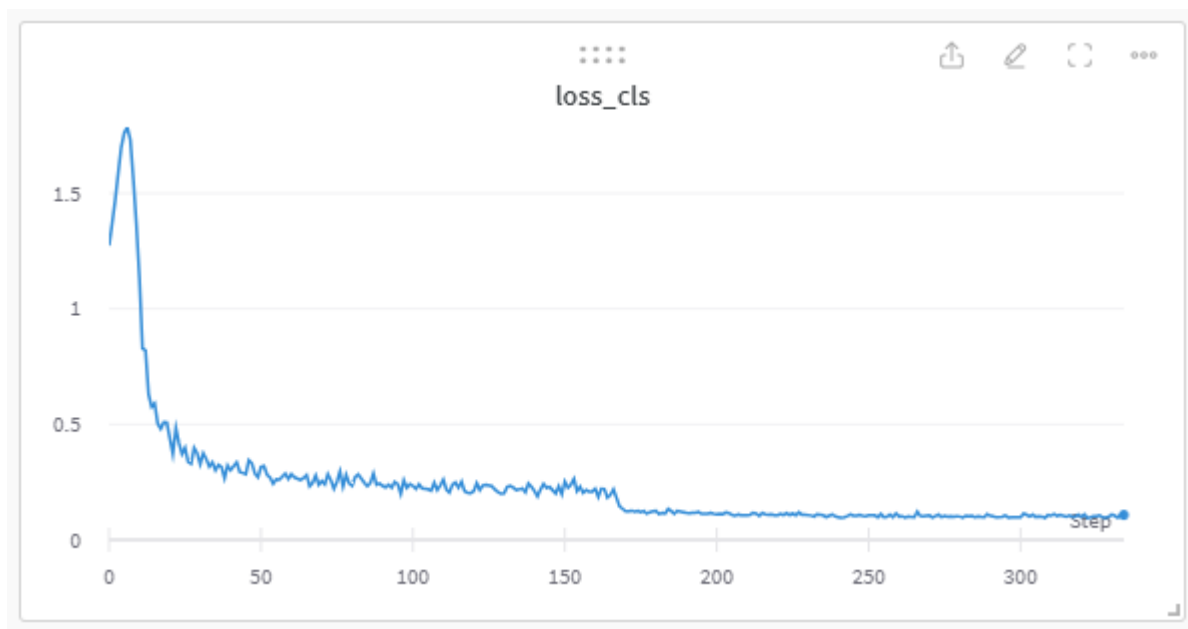
Кривые обучения.

Полностью графики параметров доступны на сайте wandb по ссылке.

<https://wandb.ai/gerasdmi/uncategorized/runs/uj27tdya?workspace=user-gerasdmi>

Ниже приведены только требуемые в задаче графики.





2.3 Выводы.

Модель слишком мощная для предложенной задачи, можно было использовать модель проще. Это не сказывается на качестве, но тренинг мог быть быстрее.

В датасет при создании внесен небольшой искусственный шум, связанный с особенностью генерации. В результате ground truth bbox оказывается несколько шире положенного для ряда объектов при их ротации. Это усложняет работу модели и не дает ей выйти на абсолютно 100% результат. Интересно то, что, возможно, модель частично научилась предсказывать этот шум также, что требует доп. исследования.

Аугментация решает проблемы границ изображения, маленьких объектов и правильной классификации объектов.

Скорость обучения слишком высокая - виден провал в loss при снижении learning rate. Также есть небольшой резерв повышения точности.

Кол-во эпох слишком велико и будет снижено в задаче 3. Столь длительный тренинг объясняется наблюдением за loss_cls.

Наблюдаемый низкий iou у маленького и единичного объекта связан с тем, что ошибка стандартного смещения детектируемой границы для него относительно велика.

Наблюдаемый высокий iou у объектов обусловлен минимальным шумом при генерации датасета.

2.4 Динамическая генерация датасета.

Платформа `mmdetection` хорошо приспособлена к кастомизации. При необходимости она допускает полностью ручной цикл обучения при сохранении функциональности модулей. В данном случае целесообразнее подключить модуль уровня `pipeline` для динамического формирования файлов и аннотаций. Фактически мы подключаем кастомный модуль аугментации, который полностью определяет содержимое тренингового (а при необходимости и тестового) датасета.

Файл `my_pipeline` содержит код исходного класса генерации объектов из задачи 1 с форматированием во внутренний стандарт платформы `MMDetection`.

К недостаткам данного метода следует отнести необходимость иметь на диске как минимум один файл с любыми аннотациями и один файл с любыми изображением. Данные будут полностью динамическими, файлы выше нужны только для стартовых проверок платформы.

Задача 3. Исследования по дообучению нейронной сети.

3.1 Подготовка.

Проведено создание следующих датасетов (включая задачу 1 и 2).

Dataset Name	Dataset Description
set_1 (100)	Задача 1, демонстрационный датасет
set_2 (10000)	Задача 2, все фигуры, тренинговый датасет
set_3 (1000)	Задача 2, все фигуры, тестовый датасет

set_4 (12000)	Задача 3, кроме гексагона, тренировочный датасет (нулевой для задачи 3)
set_5 (3000)	Задача 3, кроме гексагона, тестовый датасет
set_6 (3000)	Задача 3, обязательный гексагон, тестовый датасет
set_11 (12400)	Задача 3, 12000 без гексагона и 400 с гексагоном, тренировочный датасет
set_12 (12800)	Задача 3, 12000 без гексагона и 800с гексагоном, тренировочный датасет
set_13 (13200)	Задача 3, 12000 без гексагона и 1200 с гексагоном, тренировочный датасет
set_14 (13600)	Задача 3, 12000 без гексагона и 1600 с гексагоном, тренировочный датасет
set_15 (14000)	Задача 3, 12000 без гексагона и 2000 с гексагоном, тренировочный датасет
set_16 (14400)	Задача 3, 12000 без гексагона и 2400 с гексагоном, тренировочный датасет
set_17 (14800)	Задача 3, 12000 без гексагона и 2800 с гексагоном, тренировочный датасет
set_18 (15200)	Задача 3, 12000 без гексагона и 3200 с гексагоном, тренировочный датасет
set_19 (15600)	Задача 3, 12000 без гексагона и 3600 с гексагоном, тренировочный датасет
set_20 (16000)	Задача 3, 12000 без гексагона и 4000 с гексагоном, тренировочный датасет
set_21 (16400)	Задача 3, 12000 без гексагона и 4400 с гексагоном, тренировочный датасет
set_22 (16800)	Задача 3, 12000 без гексагона и 4800 с гексагоном, тренировочный датасет
set_23 (17200)	Задача 3, 12000 без гексагона и 5200 с гексагоном, тренировочный датасет
set_24 (17600)	Задача 3, 12000 без гексагона и 5600 с гексагоном, тренировочный датасет
set_25 (18000)	Задача 3, 12000 без гексагона и 6000 с гексагоном, тренировочный датасет
set_26 (18400)	Задача 3, 12000 без гексагона и 6400 с гексагоном, тренировочный датасет
set_27 (18800)	Задача 3, 12000 без гексагона и 6800 с гексагоном, тренировочный датасет
set_28 (19200)	Задача 3, 12000 без гексагона и 7200 с гексагоном, тренировочный датасет
set_29 (19600)	Задача 3, 12000 без гексагона и 7600 с гексагоном, тренировочный датасет

Каждый тест разделен на две группы “А” и “Б”. Часть “А” означает тестирование модели на данных без гексагона - датасет set_5 (3000). Часть “Б” означает тестирование модели на данных с обязательным гексагоном в каждой картинке - датасет set_6 (3000).

3.2 Результаты.

Группа моделей А, таблица 1.

Case Name	Train Dataset	Test Dataset	Precision (IoU 0.5)	Recall (IoU 0.5)	Rhombus GT	Rhombus DT	R%
3a	set_4 (12000)	set_5 (3000, no hex)	997	1000	3009	3009	1,00
3a11	set_11 (12400)	set_5 (3000, no hex)	997	1000	3009	3009	1,00
3a21	set_21 (16400)	set_5 (3000, no hex)	997	1000	3009	3009	1,00
3a29	set_29 (19600)	set_5 (3000, no hex)	997	1000	3009	3009	1,00

таблица 1, продолжение

Case Name	Triangle GT	Triangle DT	T%	Circle GT	Circle DT	C%	Hexagon GT	Hexagon DT	H%
3a	3019	3020	1,00	2991	3000	1,00	0	0	0,00
3a11	3019	3020	1,00	2991	2991	1,00	0	0	0,00
3a21	3019	3019	1,00	2991	2991	1,00	0	0	0,00
3a29	3019	3019	1,00	2991	2991	1,00	0	0	0,00

Группа моделей “Б”, таблица 2.

Case Name	Train Dataset	Test Dataset	Precision (IoU 0.5)	Recall (IoU 0.5)	Rhombus GT	Rhombus DT	R%
3b	set_4 (12000)	set_6 (3000, w hex)	741	750	2202	2226	1,01
3b11	set_11 (12400)	set_6 (3000, w hex)	973	976	2202	2286	1,04
3b12	set_12 (12800)	set_6 (3000, w hex)	992	996	2202	2212	1,00
3b14	set_14_(13600)	set_6 (3000, w hex)	995	1000	2202	2202	1,00
3b21	set_21 (16400)	set_6 (3000, w hex)	994	1000	2202	2202	1,00
3b29	set_29 (19600)	set_6 (3000, w hex)	997	1000	2202	2202	1,00

таблица 2, продолжение.

Case Name	Triangle GT	Triangle DT	T%	Circle GT	Circle DT	C%	Hexagon GT	Hexagon DT	H%
3b	2262	2264	1,00	2268	4517	1,99	2260	0	0,00
3b11	2262	2262	1,00	2268	2789	1,23	2260	1754	0,78
3b12	2262	2262	1,00	2268	2642	1,16	2260	2142	0,95

3b14	2262	2262	1,00	2268	2516	1,11	2260	2232	0,99
3b21	2262	2262	1,00	2268	2322	1,02	2260	2260	1,00
3b29	2262	2262	1,00	2268	2283	1,01	2260	2257	1,00

Проведено 4 теста для группы “А” для нулевого, первого, одиннадцатого и девятнадцатого датасета - результаты в части наблюдаемых метрик одинаковы.

Проведено 7 тестов для группы “Б” для нулевого, первого, второго, четвертого, одиннадцатого и девятнадцатого датасета - результаты отличны друг от друга - см. раздел 3.2 Выводы.

3.2 Выводы.

Группа моделей “А” показала полную стабильность при добавлении в обучение гексагонов. Поскольку в тестовых данных гексагонов не было, то добавление гексагонов в тренинговые данные никак не изменило поведение модели в части детекции остальных объектов по параметрам Precision, Recall, R%,T%,C%,H%).

Группа моделей “Б” при нулевой итерации (при полном отсутствии знания о гексагонах) почти полностью отнесла гексагоны к кругам. Однако уже при добавлении всего 400 гексагонов к обучающим данным уже 78% гексагонов было детектировано успешно. При добавлении 800 гексагонов уже 95%. Возможно, применение инструментов типа RandomOverSampler могло бы улучшить детекцию уже с первой итерации.