**Тестовое задание**

***1. Реализовать модуль генерации данных***

Реализовать на языке Python класс, один из методов которого будет создавать изображение и описание фигур, находящихся на этом изображении. Сгенерированное изображение должно быть размером 256х256 пикселей и представлять собой однотонный фон случайного цвета, на котором размещены примитивные геометрические фигуры случайного размера, формы, положения и поворота (без выхода за границы изображения и наложения друг на друга) и случайного цвета, отличного от цвета фона. На изображении могут находиться одновременно от 1 до 5 примитивных фигур. В качестве примитивных фигур выступают следующие фигуры: ромб, треугольник, круг, гексагон (правильный шестиугольник). Описанный прямоугольник вокруг сгенерированной фигуры должен иметь размеры по наименьшей стороне не менее 25 пикселей, а по наибольшей не более 150 пикселей.

Выход метода класса:

1. сгенерированное изображение (см. пример на рисунке 1);
2. список параметров *описывающих* прямоугольников сгенерированных примитивных фигур (уникальный идентификатор (id), название фигуры (name), координаты *x*, *y* верхнего левого угла, ширина (*w*) и высота (*h*));

В результате привести листинг кода и папку, содержащую 100 примеров сгенерированных изображений (формат png без сжатия) каждому из которых в соответствие сгенерирован json файл описания (например, изображению 001.png соответствует файл описание 001.json). Примеры сгенерированного изображения и пример файла описания находятся в приложении к этому заданию.

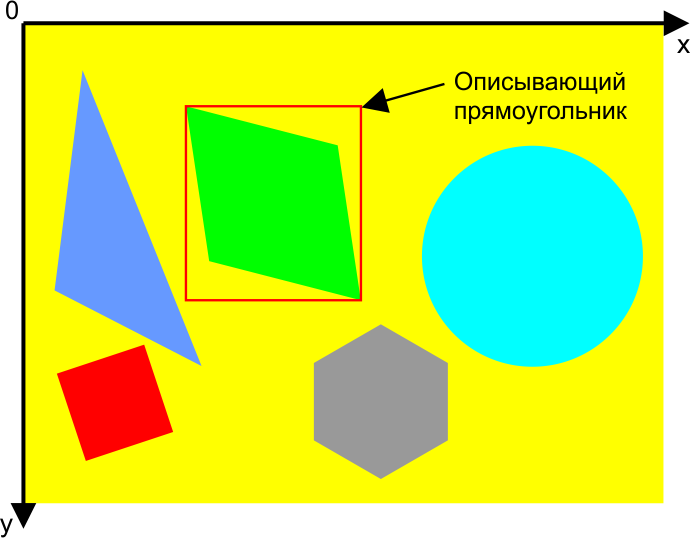


Рисунок 1 – Пример сгенерированного изображения с указанием направления осей и иллюстрацией, что такое описывающий прямоугольник

***2. Детекция примитивных геометрических фигур на изображении***

Реализовать нейросетевую модель (нейронную сеть), которая осуществляет детекцию примитивных геометрических фигур на изображении. В качестве обучающих и тестовых данных использовать изображения, сгенерированные при помощи вашего класс из п.1. Выходом обученной нейронной сети являются параметры *описывающего* прямоугольника. Провести обучение предложенной нейросетевой модели. Привести график функции потерь (loss function) в ходе обучения. В процессе обучения фиксировать параметры обучающей выборки: общее количество изображений и количество изображений на каждый тип фигуры. При обучении использовать аугментацию. Для тестовой выборки из 1000 примеров для 2 разных checkpoint-ов обучения (промежуточного и финального) привести максимальную, минимальную (с приложением примеров изображений и результатов детеции для этих двух случаев) и среднюю оценку IoU, а также посчитать показатели precision и recall для IoU > 0.5. Результаты выполнения задания оформить в виде отчета с описанием архитектуры выбранной сети, обоснованием этого выбора, иллюстрациями результатов, описанием применяемых аугментаций, графиками обучения, комментариями, результатами тестирования и выводами (шрифт Times New Roman, размер шрифта 14, полуторный интервал).

P.S.: реализация в качестве обучающей выборки изображений, сгенерированных на лету, путем вызова соответствующей функции класса, а не сгенерированных предварительно не является обязательным требованием, но дает большой плюс. В этом случае под эпохой обучения принять реализацию обучения на 1 тысячи изображений. В случае использования обучающей выборки из предварительно сгенерированных изображений объем обучающей выборки должен составлять не менее 5 тысяч изображений.

***3. Исследования по дообучению нейронной сети***

Подготовить сбалансированную (примерно одинаковое количество изображений на каждый тип фигуры) обучающую (12000 изображений) и тестовую №1 (3000 изображений) выборку, используя результаты п.1 *без фигуры гексагон*! Заново обучить предложенную в п.2 нейросетевую модель (количество выходов сразу должно быть 5) на сгенерированной обучающей выборке (перед обучением задать весовые коэффициенты сети случайными значениями) и зафиксировать этот checkpoint (назовем его start\_learning). Подготовить сбалансированную тестовую выборку №2 из 3000 изображений, но *содержащую фигуру гексагон*! После этого совершить следующий порядок действий 20 раз:

1. Провести отдельно тестирование нейросетевой модели на тестовой выборке №1 и №2. Для каждой выборки зафиксировать значения precision и recall для IoU > 0.5, а также количество детектированных фигур и общее количество для каждого типа фигуры.
2. Сгенерировать выборку из 400 изображений, содержащую обязательно гексагон (но не только), и добавить ее в обучающую выборку (на первой итерации обучающая выборка станет 12400 изображений, на второй 12800 и т.д.).
3. Дообучить нейросетевую модель c позиции start\_learning.

Результаты исследования привести в форме отчета с таблицами, графиками, гипотезами, своими мыслями и выводами.