# Промежуточный отчёт по аугментатору

Ссылка на репозиторий: https://github.com/valkon29/mipt2024f_konovalov_v_r

Автор: Коновалов Валентин

## Input/Output

Вход:

- набор изображений
- файл с разметкой, полученной с помощью VGG Image Annotator. Для каждой картинки она представляет из себя набор точек, которые обозначают полигон вокруг штрих-кода. Помимо этого в атрибутах прусутствует тип кода и метка валидности (за подробностями см. пример в репозитории).

Выход:

- набор изображений, полученных из входных путём некоторых искажений
- разметка в том же формате, что и на входе

## API

Исполняемый файл *main.py*, ему на вход подаются путь к папке с изображениями, путь к файлу и число, задающее кол-во, которые необходимо получить тз исходного посредством аугментации:

```
python3 main.py images via_project_9Nov2024_20h28m_.json 2
```

Сама операция искажения, применяемая к входным изображениям, захардкожена внутри файла *main.py*.

## Искажения

Все искажения изображений осуществляются посредством библиотеки albumentations. Она поддерживает работу с ключевыми точками, масками и bounding_box-ами, поэтому пока что её функционала вполне достаточно. Вот список искажений, взятых из этой библиотеки, которые на данный момент используются:

- **Rotate**: поворот на угол от 0 до 360 градусов;
- **ISONoise**: симуляция шума, возникающего при высоких значениях ISO;
- **Perspective**: углы изображения смещаются на случайные расстояния в заданных пределах, изображение деформируется соотвествующим образом;
- **ChromaticAberration**: хроматические абберации изображения;
- **Defocus**: симуляция расфокусировки камеры.

Кроме того, была создана таблица со всеми искажениями, которые потенциально нам могут пригодиться. В столбцах также присутствуют оценка релевантности отдельных искажений для

нашей задачи, а также местами их краткое описание и допустимые границы параметров. *По многим из них пока есть сомнения, таблицу определённо надо ещё дорабатывать*.

## Примеры

Perspective + ISONoise:

Исходное изображение:


Выходное изображение:

## Некоторые детали

- На данный момент, если при искажении, хоть одна точка разметки кода выходит за границы полученного изображения, код считается не валидным и в разметке это отражается. В дальнейшем можно сделать умнее
- Каждый из типов искажений последовательно применяется с определённой вероятностью, на данный момент все вероятности равны 0.5
- При всех искажениях, которые нарушают "прямоугольность" картинки, всё лишнее обрезалось. Из вышеупомянутых к таковым относятся Rotate и Perspective. Потенциально образовавшиеся области можно, например, заполнять зеркально отражённым содержанием исходной картинки.