План работ.

1. EDA
2. Строим модель на baseline решении.
3. Используем Sota решения([**FixEfficientNet-B6**](https://paperswithcode.com/paper/fixing-the-train-test-resolution-discrepancy-2))
4. Используем fietuning перенос обучения
5. Используем аргументацию на основе Albumentations, ImageDataGenerator.
6. Используем функцию callback с ранее полученными оптимальными весами слоев сети.
7. Используем разные варианты настройки гиперпараметров: размер изображений, размер batch слоев.
8. Используем разные настройки «головы»(меняем нелинейность модели, используя ‘elu’), включая batch нормализацию
9. Выводим модель в prodaction. (не удалось реализовать)

Результат работы, выводы.

Таблица значений метрики accuracy в зависимости от применяемого метода построения модели.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модель | Аугментация | Batch | FrozenLayer | ImageSize | Batchnorm | ValueMetric |
| BaseLine | ImageDataGenerator | 32 | No | 256 | No | 93 |
| FixEfficientNet-B6 | ImageDataGenerator | 32 | Yes | 128 | No | 94,6 |
| FixEfficientNet-B6 | ImageDataGenerator | 32 | Yes | 256 | No | 96 |
| FixEfficientNet-B6 | Albumentations | 32 | Yes | 256 | Yes | 96,22 |

Так же использовали EfficientNetB6 с 'noisy-student' и ТТА. Оба метода не принесли существеннго прироста метрики. Вероятно улучшить метрику можно за счет увеличения batchsize и размера картинки(мне не хватило квоты ГПУ). Так же можно попробовать поменять параметры Albumentations.