

# Детекция и классификация военной техники по фотографиям с дрона

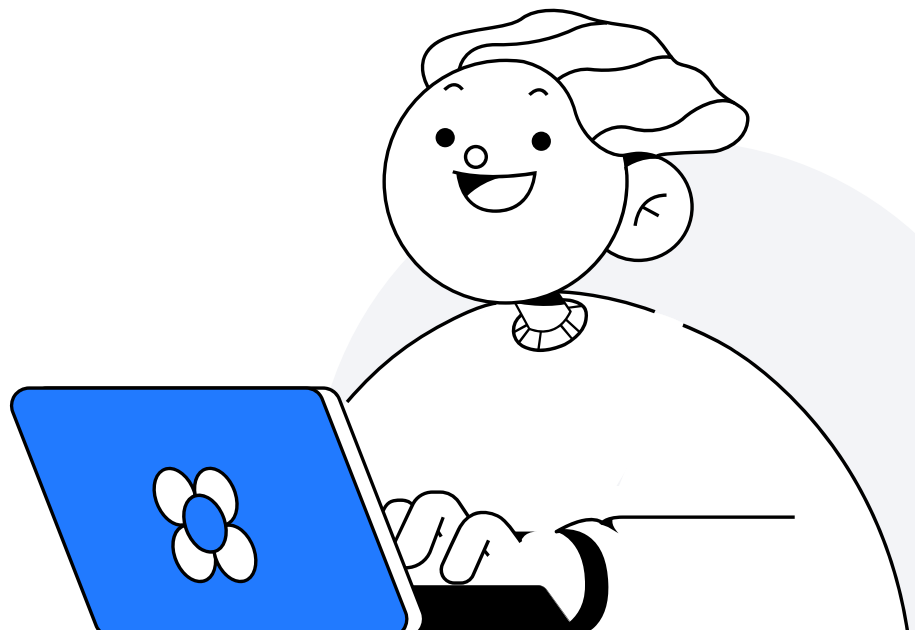
Итоговый проект по курсу «Deep Learning»

Серегин Константин  
Группа DLL-27



# Содержание

- 1 Постановка задачи для глубокого обучения
- 2 Анализ данных
- 3 Методика реализации
- 4 Итоги обучения модели
- 5 Выводы



# Постановка задачи для глубокого обучения



1

# Описание проекта



## Исходная задача:

Обучение моделей Faster R-CNN и SSD для обнаружения и определения типа военных самолетов по фотографиям с беспилотника.

## Актуальность задачи, её место в предметной области:

Решение данной задачи позволит осуществлять поиск военных самолетов с дрона или спутника в автоматическом режиме, возлагая на оператора только обязанности выбора квадрата поиска и управление летательным аппаратом.

## Целевая метрика:

Mean Average Precision (mAP)



# Анализ данных



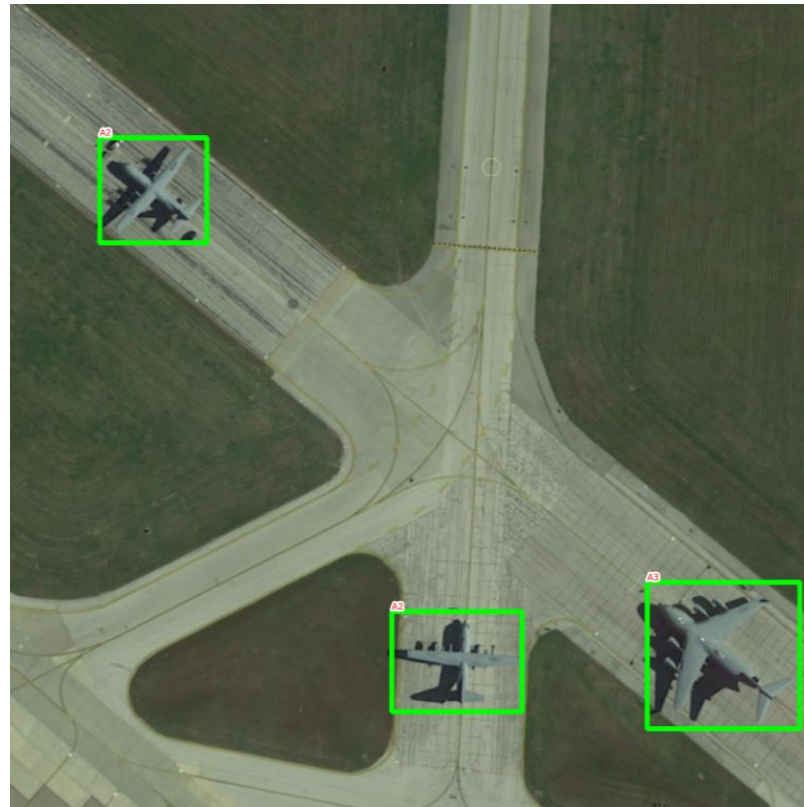
2

# Анализ данных

1. Источник данных:  
<https://www.kaggle.com/datasets/khlaifiabile/military-aircraft-recognition-dataset>
2. Тип данных:  
данные представлены в виде 2-х папок, одна со снимками с дрона в формате jpg, вторая с метками (класс и ограничивающая рамка самолета) в формате xml.
3. Размер датасета:  
датасет включает в себя 3842 изображения с 20-ю типами самолетов и 22341-ю экземплярами, помеченными прямоугольными ограничивающими рамками.
4. Результаты разведочного анализа данных:  
разведочный анализ показал, что данные не имеют пропусков, все файлы jpg соответствуют файлам xml (нет ошибок в названиях).
5. Основные шаги предобработки данных, их результаты:  
данные были сведены в таблицу DataFrame, где указаны: название изображения, название самолета, Xmin, Ymin, Xmax, Ymax, ширина изображения, высота изображения, путь до файла jpg, путь до файла xml. Были закодированы классы, DataFrame был разбит на тренировочный и тестовый наборы.



6. Примеры изображений с метками и рамками:



# Методика реализации



3



# Каким образом решали задачу?

1. Проведенные эксперименты:  
Были обучены 2 модели Faster R-CNN и SSD. В процессе обучения менялись размер батча, использовался алгоритм оптимизации шага lr\_scheduler, аугментация и L1, L2 регуляризации.
2. Используемые источники для каждого эксперимента (ссылки на статьи и т.д.):
  - Modern Computer Vision with PyTorch. V Kishore Ayyadevara. Packt Publishing Limited, 2020
  - PyTorch Computer Vision Cookbook. Michael Avendi. Packt Publishing Limited, 2020
  - Deep Learning with PyTorch. Стивенс Эли, Антига Лука, Виман Томас. Издательство Питер
  - Глубокое обучение: лёгкая разработка проектов на Python. Вейдман Сет. Издательство Питер



### 3. Проблемы возникшие в процессе выполнения проекта:

- во время создания DataFrame появились несколько строк со значением inf, которые вызывали ошибку во время обучения, проблема решена удалением строк;
- в процессе преобразования ограничивающих рамок для подачи в модель из-за того, что изображения на фото имеют маленький размер, некоторые значения ширины и высоты обращались в "0", либо в значение близкое к "0", проблема была решена введением блока, который отсеивал все значения  $< 0.005$ ;
- при расчете mAP выпадала ошибка деления на "0", проблема решена введением коэффициента.

### 4. Полученные результаты:

- Faster R-CNN: mAP (train) = 0.937  
mAP (тест) = 0.948
- SSD: mAP (train) = 0.922  
mAP (тест) = 0.813



Результат работы Faster R-CNN



Результат работы Faster SSD



# Итог



4

# Итог

Обе модели показали высокое качество обнаружения самолетов и определения их типа, интересным явлением оказалось то, что модель Faster R-CNN находит объекты лучше, чем SSD, но степень уверенности в том, что она нашла то, что нужно - ниже. Не смотря на хорошие результаты данные модели имеют высокий потенциал для развития:

- улучшение самой модели путем регулировки параметров обучения;
- изменение входных данных: увеличения/уменьшения изображений самолетов для возможности определения разномасштабных объектов, изменения погодных условий, использование аугментации;
- расширение области применения модели путем введения новых классов (людей, танков, вертолетов).



# Выводы



5

1

Лучшая модель: Faster R\_CNN

2

Удачные эксперименты: обе модели выдали отличные результаты

3

Неудачные эксперименты: -

4

Дальнейшие пути развития и улучшения решения:

- в Faster R-CNN не наблюдается переобучения, для начала, можно идти в сторону увеличения числа эпох
- модель SSD начинает незначительно переобучаться в районе 3-ей итерации, в продолжение развития модели может помочь введение L1 или L2 регуляризации, которые будут штрафовать аномально высокие или низкие веса (сопутствующие переобучению).



**Спасибо за внимание!**

