

Пояснительная записка
к итоговому проекту на тему:

**"Детекция и классификация военной техники по
фотографиям с дрона"**

Автор: Серегин Константин
Группа: DLL-27

Оглавление

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

1. Постановка задачи для глубокого обучения

Целью данного проекта является создание модели Глубокого обучения для обнаружения военных самолетов, определение их местоположения и типа на основе снимков с дрона/спутника.

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

2. Анализ данных

Для обучения моделей был использован набор данных «Military Aircraft Recognition dataset» с сайта Kaggle, ссылка:

<https://www.kaggle.com/datasets/khlaifiabile/military-aircraft-recognition-dataset>

Набор данных включает в себя 3842 изображения с 20-ю типами самолетов и 22341-ю помеченными прямоугольными ограничивающими рамками экземплярами.

Изображения представлены в формате jpg, метки (класс и ограничивающая рамка) в формате xml.

Предварительный анализ показал, что данные не имеют пропусков, число файлов jpg соответствует числу файлов xml, названия файлов jpg и xml попарно соответствуют. Изображения в наборе имеют разнообразные расширения.

Техника зафиксирована с различных высот, в различное время суток и в различных погодных условиях. Варьируются степень детализации, расположение и количество техники на снимках.



Рис.1 Пример входных данных – изображения с ограничивающими рамками и метками.

3. Методика реализации

Для реализации проекта использовался фреймворк Pytorch и библиотека OpenCV, среда выполнения Google Colab. В качестве базовых для обучения были взяты модели Faster R-CNN и SSD, хорошо себя зарекомендовавшие в области Object Detection.

Для удобства анализа и корректировки исходные данные, перед подачей в функцию преобразования для подачи в модель, были сведены в DataFrame, где были указаны: название изображения, тип модели самолета, Xmin, Ymin, Xmax, Ymax, ширина изображения, высота изображения, путь до файла jpg, путь до файла xml. Число строк DataFrame = 22341 – число экземпляров самолетов в наборе.

```
42 df.sample(4)
```

	ImageID	LabelName	X_left	Y_down	X_right	Y_up	width	height	way_pic	way_ann
13294	3124.jpg	A1	0.40250	0.54875	0.49250	0.62375	800	800	/content/JPEGImages/3124.jpg	/content/Annotations/Horizontal Bounding Boxes...
253	1046.jpg	A5	0.39375	0.05875	0.51875	0.13875	800	800	/content/JPEGImages/1046.jpg	/content/Annotations/Horizontal Bounding Boxes...
15500	3502.jpg	A20	0.20375	0.42625	0.39875	0.51625	800	800	/content/JPEGImages/3502.jpg	/content/Annotations/Horizontal Bounding Boxes...
390	1071.jpg	A14	0.35875	0.10250	0.56125	0.28125	800	800	/content/JPEGImages/1071.jpg	/content/Annotations/Horizontal Bounding Boxes...

Рис. 2 DataFrame с данными.

В процессе обучения использовались различные способы улучшения модели, такие как аугментация, L1 и L2 – регуляризация, алгоритм оптимизации шага lr_scheduler, изменение числа эпох.

Проблемы возникшие в процессе выполнения проекта:

- во время создания DataFrame появились несколько строчек 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 значения 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. Итоги обучения модели

Процесс обучения завершается выводом метрики mAP (Mean Average Precision).

Для нахождения mAP: в начале находится AP (Average Precision) путем вычисления площади под кривой графика Precision/Recall для каждого класса в отдельности, затем значения площадей каждого класса складываются и делятся на число классов.

Полученные результаты:

- Faster R-CNN: mAP (train) = **0.937**
 mAP (val) = **0.948**

- SSD: mAP (train) = **0.922**
 mAP (val) = **0.813**

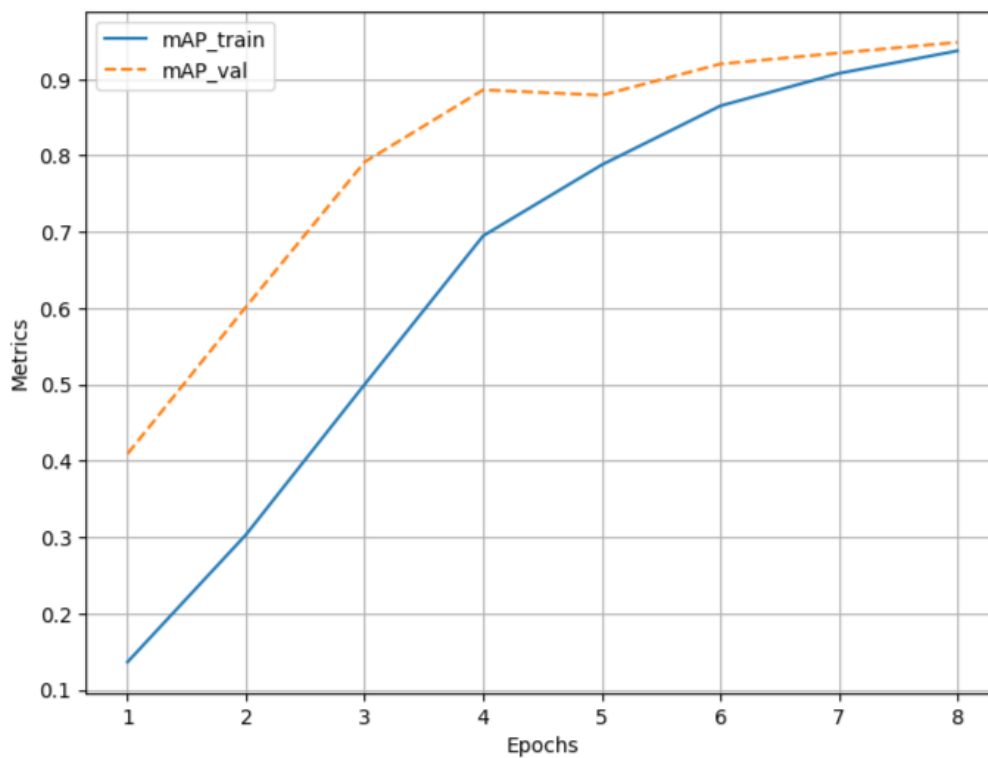


Рис. 3 График изменения значений mAP в процессе обучения модели Faster R-CNN для тренировочного и валидационного наборов.

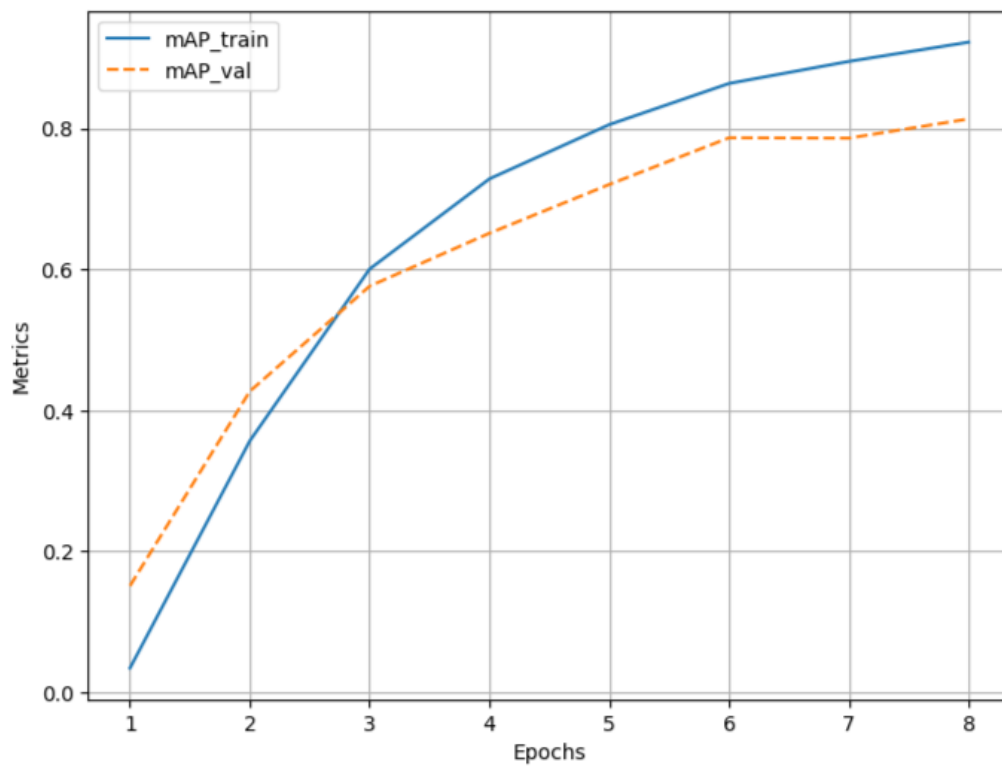


Рис. 4 График изменения значений mAP в процессе обучения модели SSD для тренировочного и валидационного наборов.

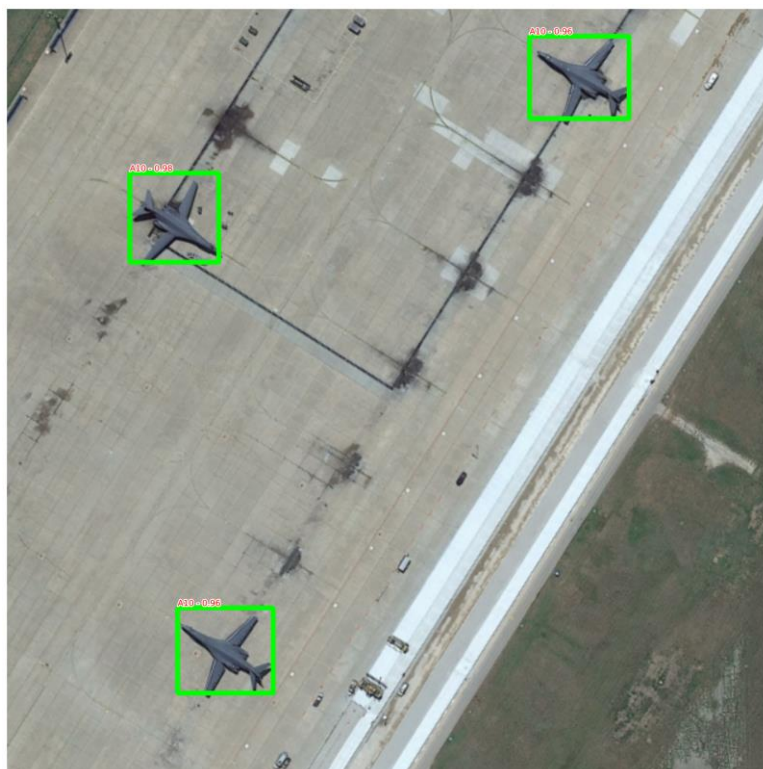


Рис. 5 Результат работы Faster R-CNN



Рис. 6 Результат работы SSD

5. Выводы и заключение

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:

1. Улучшение самой модели:

- в модели 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 можно попробовать эксперименты с L1 или L2 регуляризации, которые будут штрафовать аномально высокие или низкие веса (сопутствующие переобучению), после подавления переобучения попробовать увеличить число эпох;

2. Изменение входных данных: увеличения/уменьшения изображений самолетов для возможности определения разномасштабных объектов, изменения погодных условий, использование аугментации;

3. Расширение области применения модели путем введения новых классов (людей, танков, вертолетов).