

Детекция на спутниковых изображениях





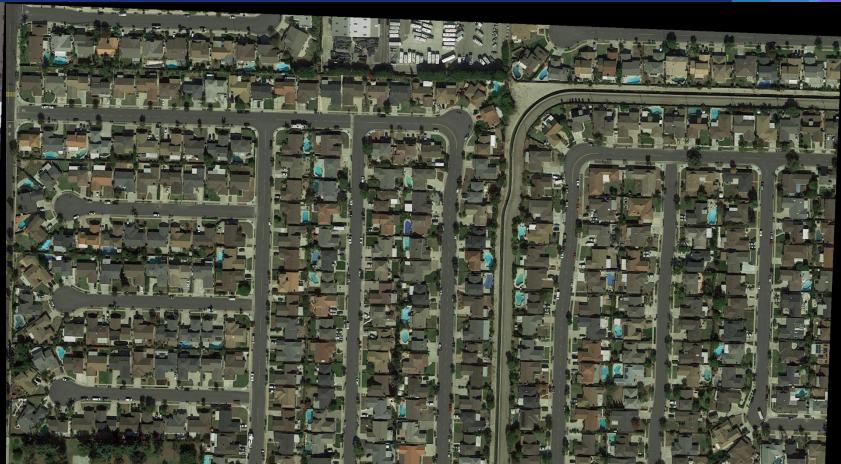
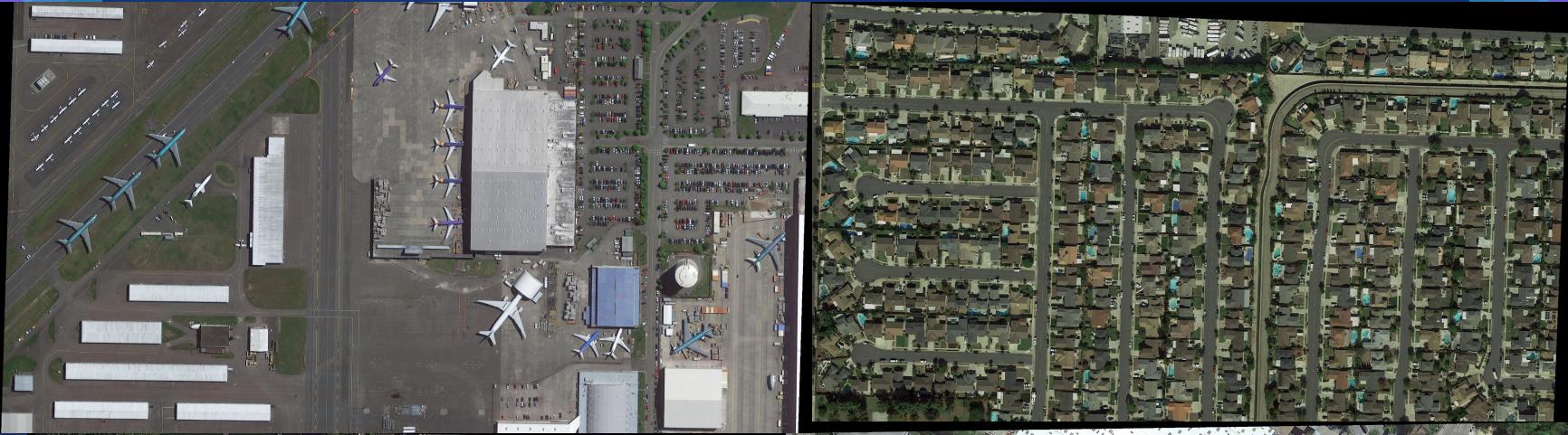
Состав команды



iSAID: A Large-scale Dataset for Instance Segmentation in Aerial Images

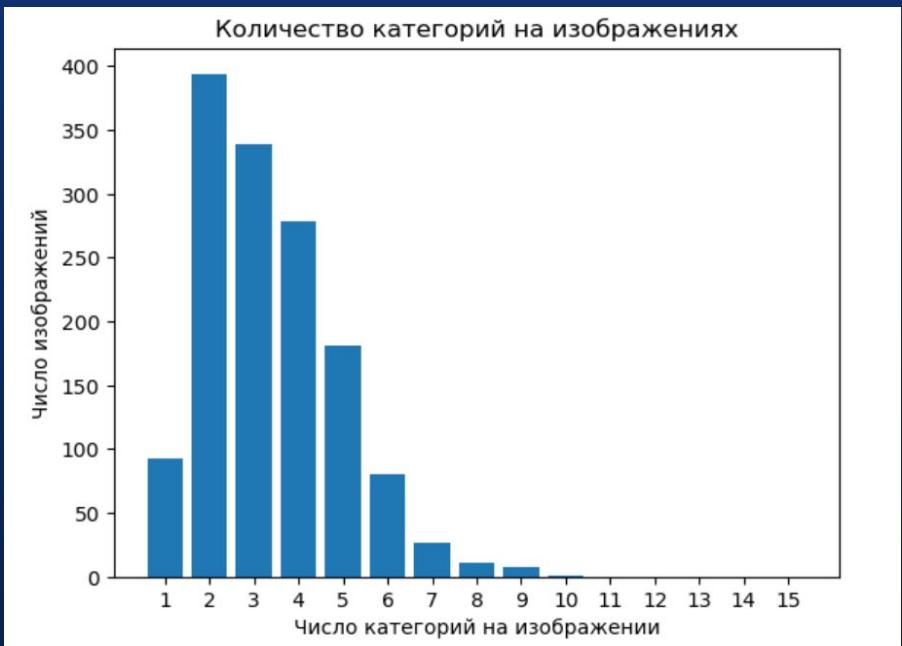
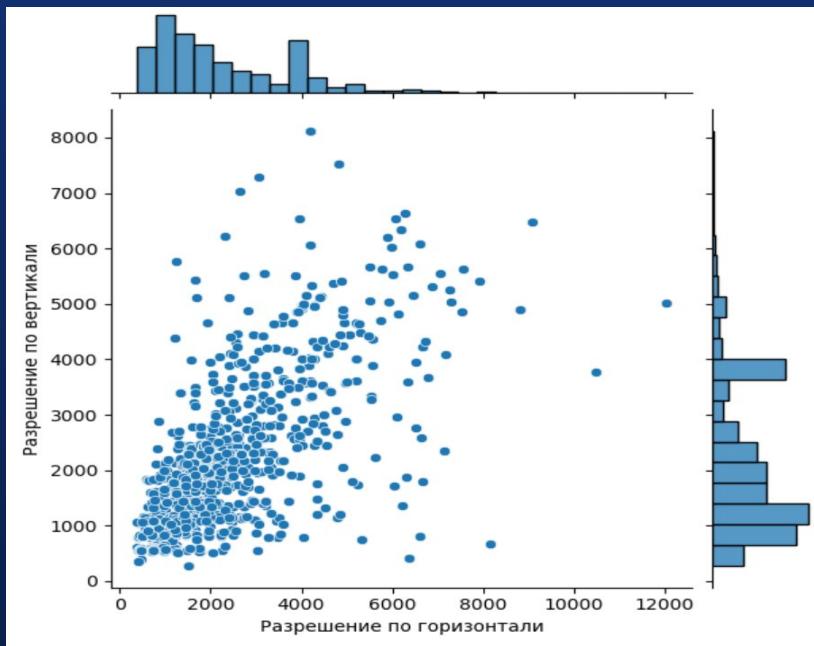
- 2806 изображений высокого разрешения;
- 471760 размеченных объектов;
- 15 различных классов;
- Яндекс S3 – место хранения изображений.

Название класса	Количество изображений	Количество объектов
small_vehicle	1156	253077
large_vehicle	841	39874
tennis_court	537	11632
ground_track_field	536	84456
ship	480	35755
harbor	418	6492
storage_tank	395	18549
swimming_pool	334	2605
plane	292	8781
bridge	289	2272
roundabout	254	3053
baseball_diamond	223	1207
soccer_ball_field	199	529
basketball_court	191	762
helicopter	149	2716

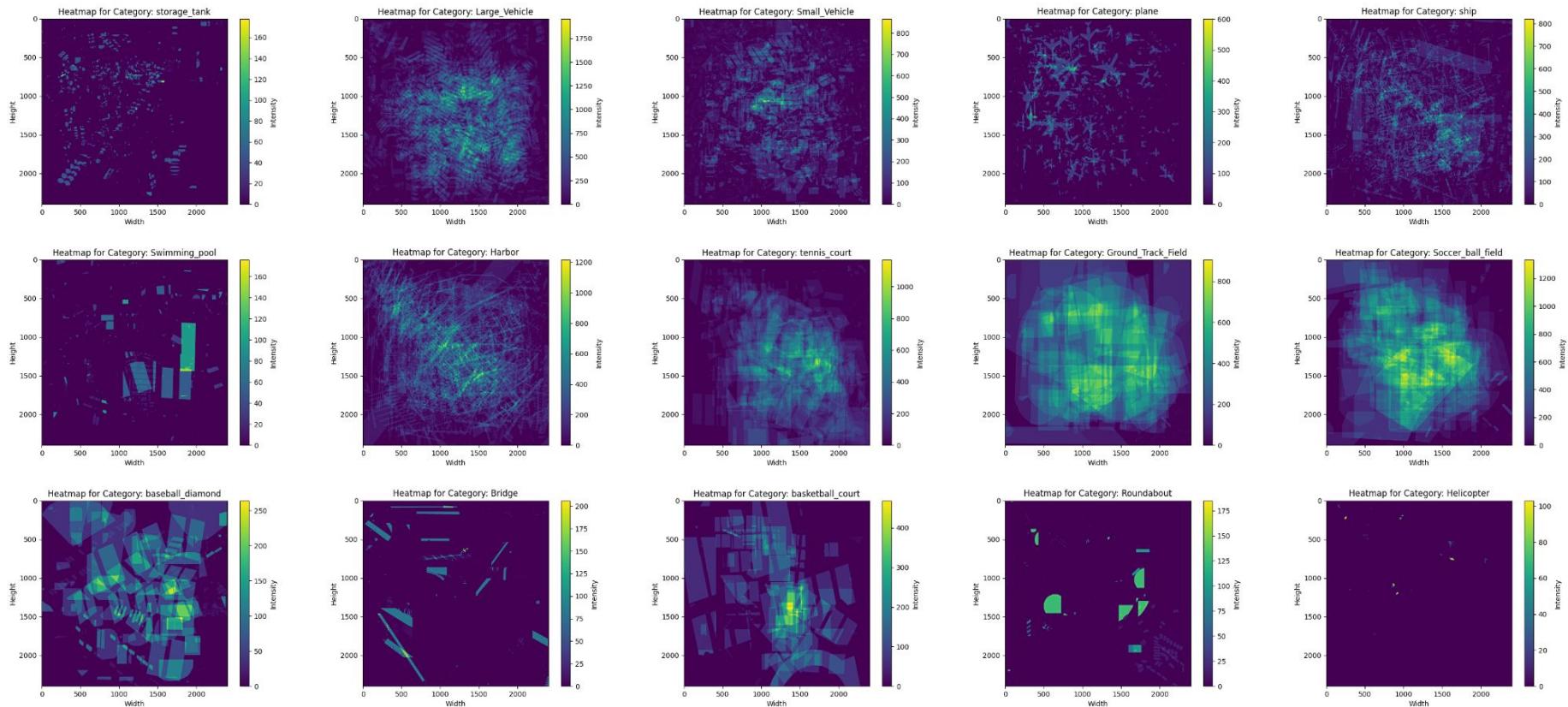




Результаты EDA



Результаты EDA



Выбор модели

- YOLO (You Only Look Once) – это семейство архитектур для детекции объектов в реальном времени.
Мы использовали Ultralitics YOLO v5s.
- YOLO использует подход SSD (Single Shot Detection), что означает, что объекты обнаруживаются в один проход сети.
Это делает его быстрым и эффективным для задач детекции в реальном времени.
- YOLO v5 имеет модульную архитектуру с общим "backbone" (сверточная нейронная сеть) и "head" (прогнозирование рамки и класса) для обнаружения объектов. Это позволяет легко настраивать и адаптировать модель под конкретные требования.

Метрики модели

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\text{Intersection}}{\text{Union}}$$

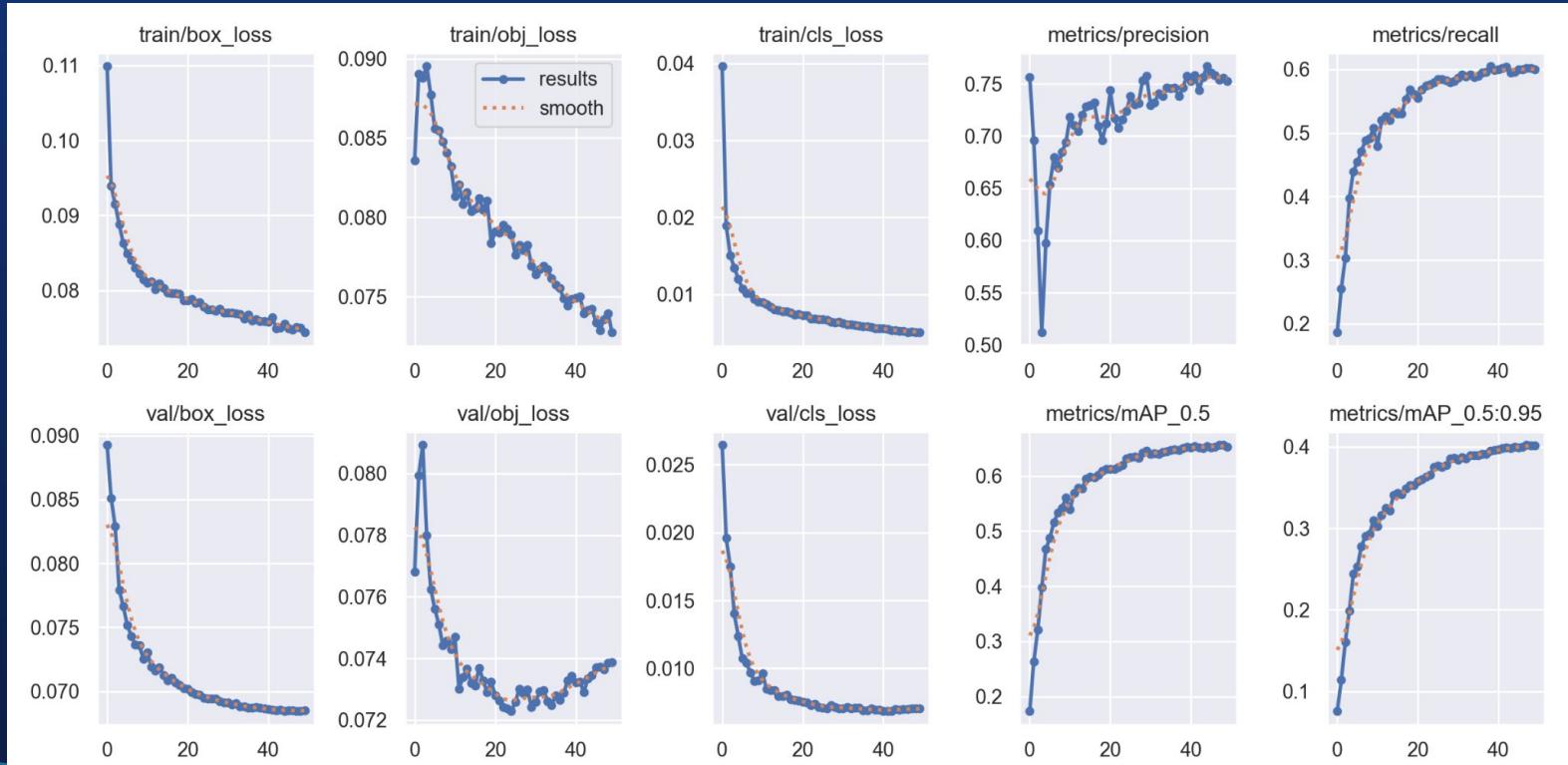
- С помощью IoU (Intersection over union) можно построить PR-кривую
- AP (Average precision) - площадь под PR-кривой
- mAP (mean Average precision) - среднее по классам значение AP

Нарезка исходных изображений

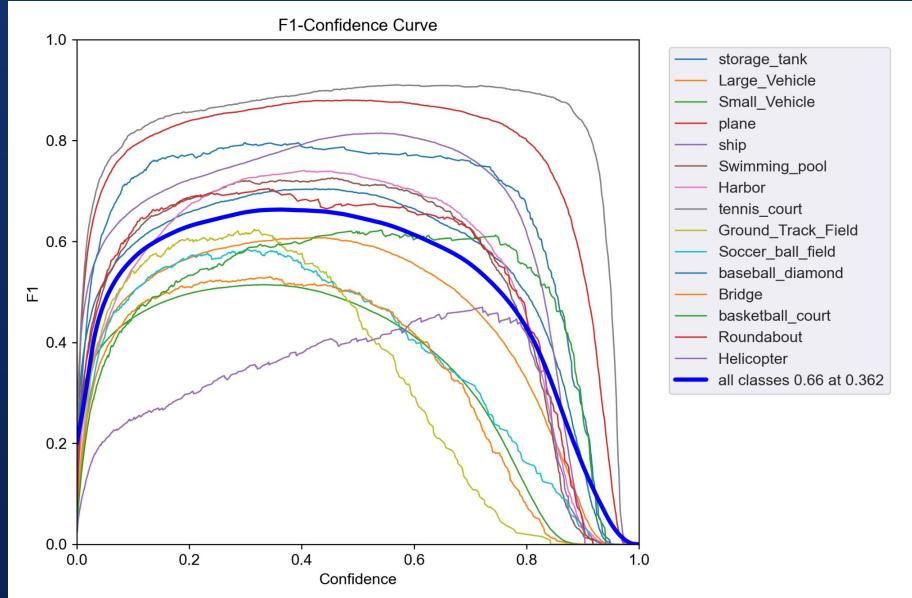
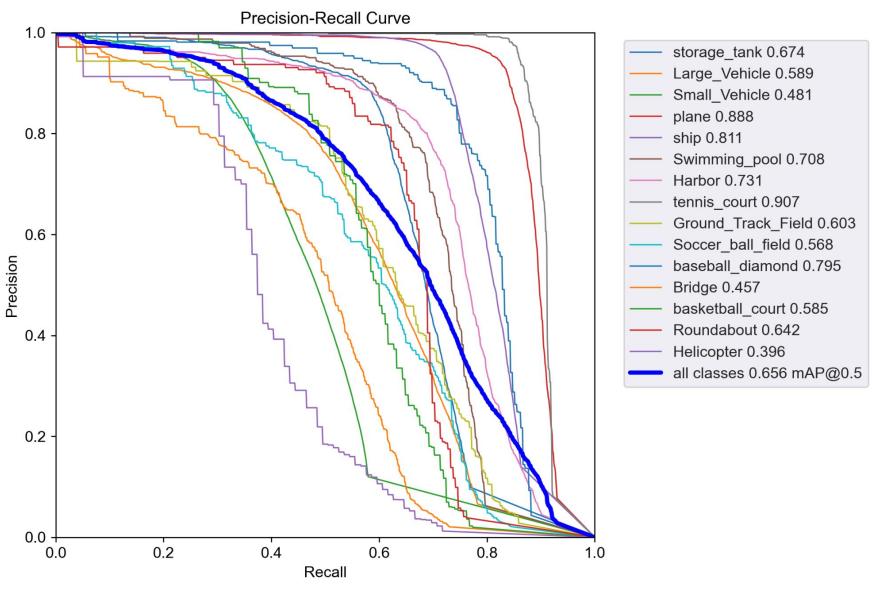
Для улучшения обучения модели при работе с маленькими объектами и большими изображениями мы разделили их без пересечений на более мелкие блоки размером порядка 600 на 600 пикселей, повысив точность предсказаний и оптимизировав вычислительные ресурсы.



Обучение модели



Результаты обучения

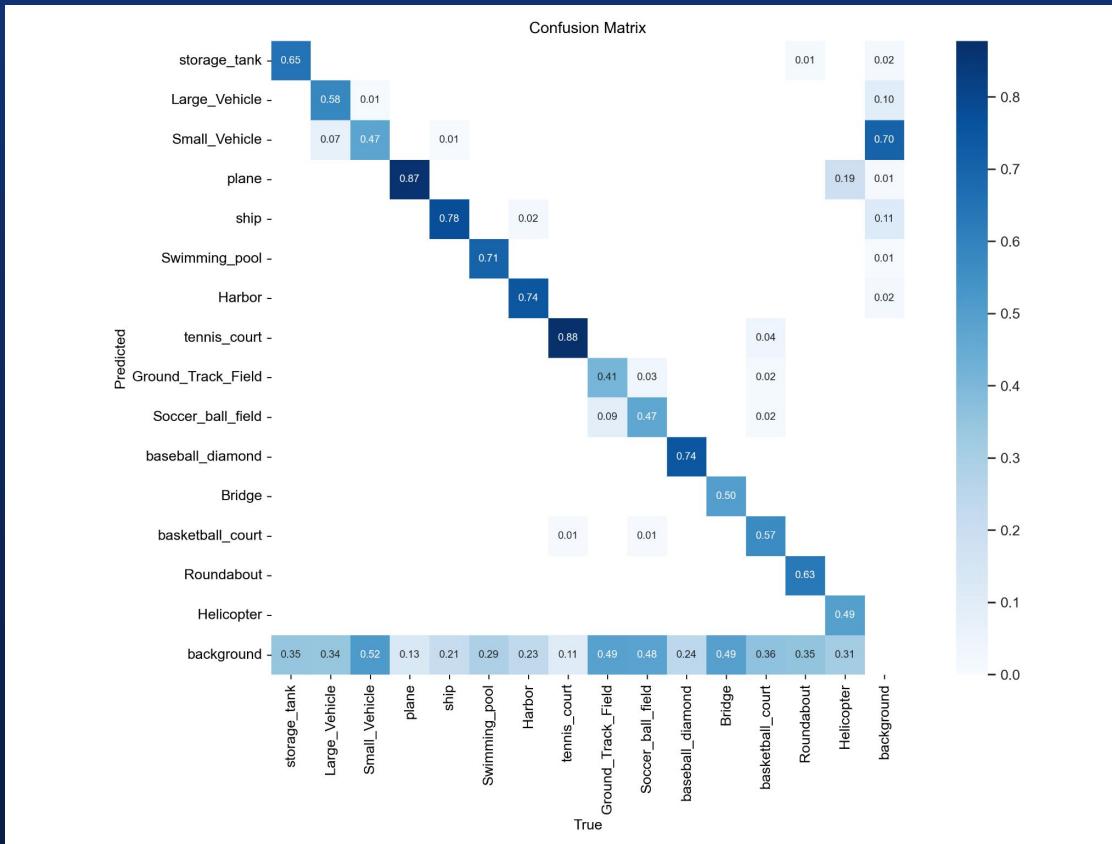


Результаты обучения

Task2 - Horizontal Object Detection																		
Row 100 entries		Search: <input type="text"/>																
Team Name	mAP	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	Date	
DH_RSIA	0.831	0.891	0.874	0.662	0.818	0.8	0.866	0.892	0.939	0.893	0.881	0.735	0.694	0.868	0.843	0.808	2020-06-11 05:44:35.432337	
orange_data	0.82	0.887	0.87	0.658	0.812	0.798	0.866	0.889	0.909	0.893	0.862	0.728	0.682	0.862	0.839	0.753	2020-06-10 02:44:31.810264	
ying_hu	0.819	0.902	0.873	0.631	0.8	0.815	0.832	0.881	0.908	0.88	0.884	0.724	0.714	0.85	0.832	0.75	2020-04-08 10:59:39.409045	
cc_zhang	0.816	0.885	0.857	0.65	0.792	0.717	0.862	0.888	0.909	0.893	0.873	0.733	0.672	0.866	0.834	0.808	2020-06-03 00:46:16.455555	
DaoRirs	0.815	0.883	0.872	0.633	0.812	0.796	0.865	0.888	0.909	0.886	0.876	0.727	0.682	0.865	0.841	0.69	2020-06-05 01:31:11.710588	
dh_data	0.815	0.875	0.87	0.659	0.812	0.796	0.793	0.873	0.939	0.893	0.862	0.721	0.681	0.853	0.839	0.754	2020-06-11 02:14:28.513540	
Sat_IAI	0.814	0.902	0.873	0.631	0.8	0.815	0.772	0.881	0.908	0.88	0.884	0.709	0.714	0.85	0.832	0.75	2020-02-02 07:52:52.316729	
DetectionTeamCSU	0.812	0.901	0.87	0.646	0.748	0.784	0.836	0.877	0.907	0.876	0.876	0.738	0.689	0.82	0.836	0.782	2020-01-03 13:31:27.251569	
pca_lab	0.81	0.904	0.861	0.635	0.822	0.784	0.826	0.863	0.908	0.887	0.872	0.696	0.689	0.855	0.821	0.732	2019-12-18 13:35:13.435914	
dingjian	0.808	0.879	0.851	0.626	0.777	0.758	0.85	0.882	0.909	0.873	0.862	0.715	0.704	0.849	0.841	0.736	2019-10-26 03:37:59.957790	

<https://captain-whu.github.io/DOTA/results.html>

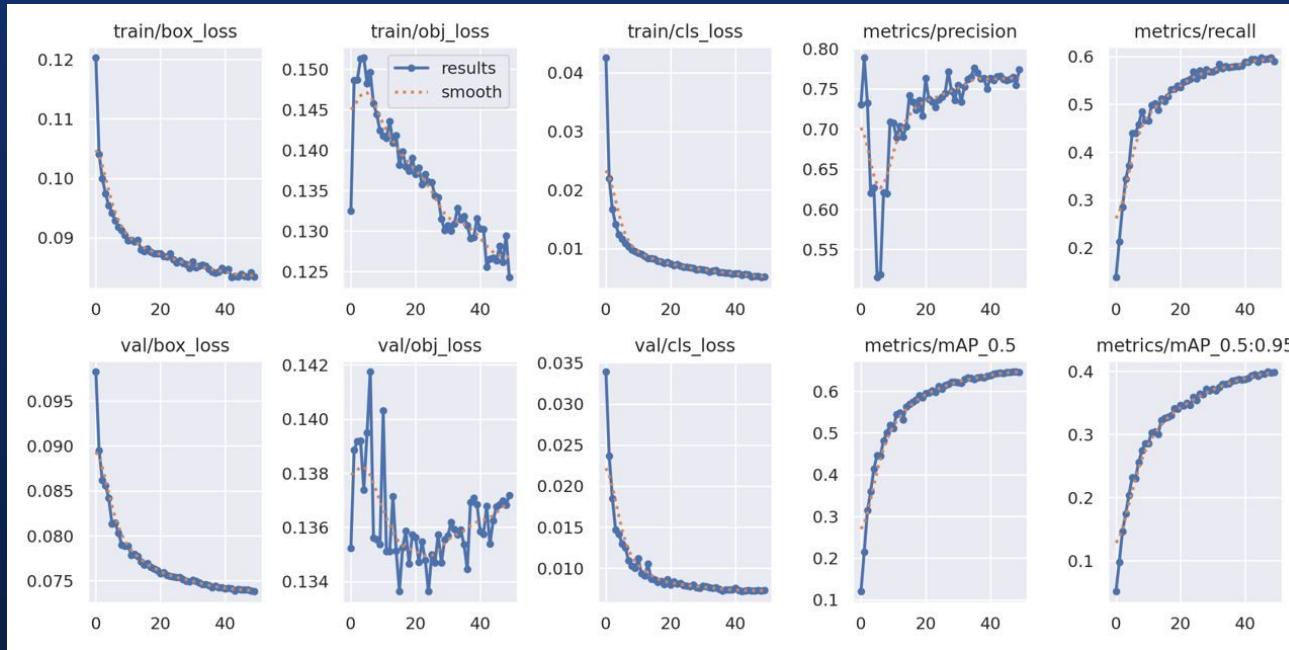
Результаты обучения



UPDATE

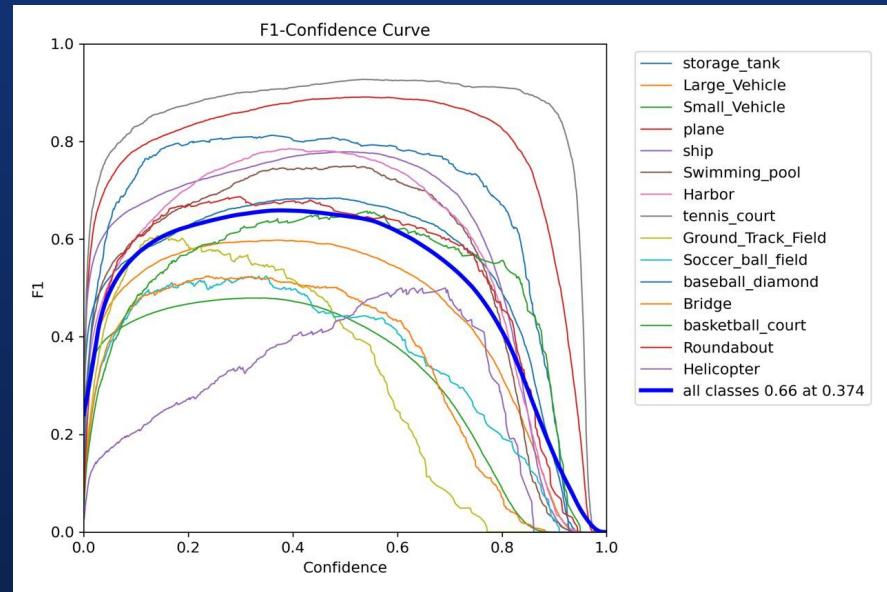
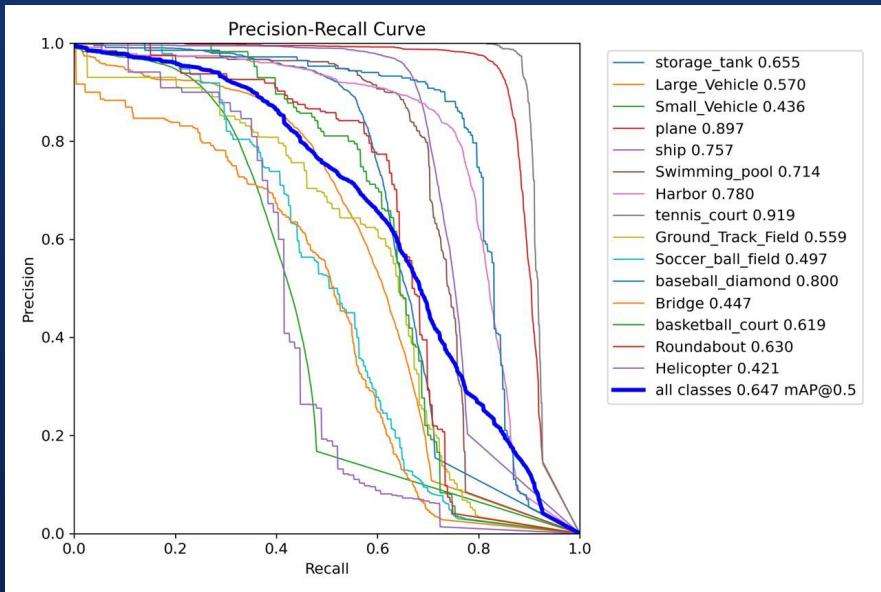
Yolo v5 с размерами изображений 960x960

Мы попробовали обучить Yolo v5 на изображениях размером 960x960 пикселей (до этого размер был 600x600). Процесс обучения представлен ниже:



UPDATE

Результат обучения на изображениях 960x960



В сравнении с предыдущими результатами, AUC-PR немного снизилась

UPDATE

Обучение с помощью HOG+SVC

Мы попробовали классифицировать изображения, используя метод опорных векторов из библиотеки sklearn. Из каждой категории мы взяли по 300 изображений. Размер изображений для обучения модели выбран 80x80 пикселей.

Итоговый accuracy_score SVC модели:

0.62666

	precision	recall	f1-score	support
Bridge	0.54	0.69	0.61	59
Ground_Track_Field	0.41	0.47	0.44	62
Harbor	0.54	0.61	0.57	59
Helicopter	0.69	0.76	0.72	58
Large_Vehicle	0.75	0.88	0.81	49
Roundabout	0.50	0.57	0.53	63
Small_Vehicle	0.75	0.68	0.71	53
Soccer_ball_field	0.41	0.43	0.42	54
Swimming_pool	0.55	0.54	0.54	65
baseball_diamond	0.58	0.40	0.47	65
basketball_court	0.73	0.79	0.76	66
plane	0.91	0.79	0.84	61
ship	0.67	0.48	0.56	64
storage_tank	0.59	0.51	0.55	57
tennis_court	0.90	0.85	0.87	65
accuracy			0.63	900
macro avg	0.63	0.63	0.63	900
weighted avg	0.63	0.63	0.63	900

UPDATE

Классификация по форме

Также мы проверили возможность классификации объектов исключительно на основе их формы (без учета цвета). Основная задача - определить, насколько снижается качество классификации, когда основным признаком является только форма объекта.

Используемые шаги в эксперименте:

- Изображения были преобразованы к черно-белому формату и изменены до размера 64 на 64 пикселя.
- После этого, с использованием фильтра, пиксели были отмечены значением 1, если их интенсивность была больше нуля, иначе - ноль.
- В качестве модели использовался метод опорных векторов (SVC).

	precision	recall	f1-score	support
Bridge	0.15	0.31	0.20	478
Ground_Track_Field	0.25	0.37	0.30	158
Harbor	0.68	0.43	0.53	1000
Helicopter	0.04	0.08	0.05	86
Large_Vehicle	0.37	0.23	0.28	1000
Roundabout	0.16	0.36	0.22	188
Small_Vehicle	0.44	0.36	0.40	1000
Soccer_ball_field	0.17	0.37	0.23	188
Swimming_pool	0.14	0.17	0.15	743
baseball_diamond	0.24	0.32	0.27	220
basketball_court	0.11	0.27	0.16	154
plane	0.76	0.42	0.54	1000
ship	0.22	0.18	0.20	1000
storage_tank	0.55	0.51	0.53	1000
tennis_court	0.64	0.64	0.64	788
accuracy			0.36	9003
macro avg	0.33	0.34	0.31	9003
weighted avg	0.43	0.36	0.38	9003

Итоговый accuracy_score SVC модели:

0.3587

Демонстрация работы бота

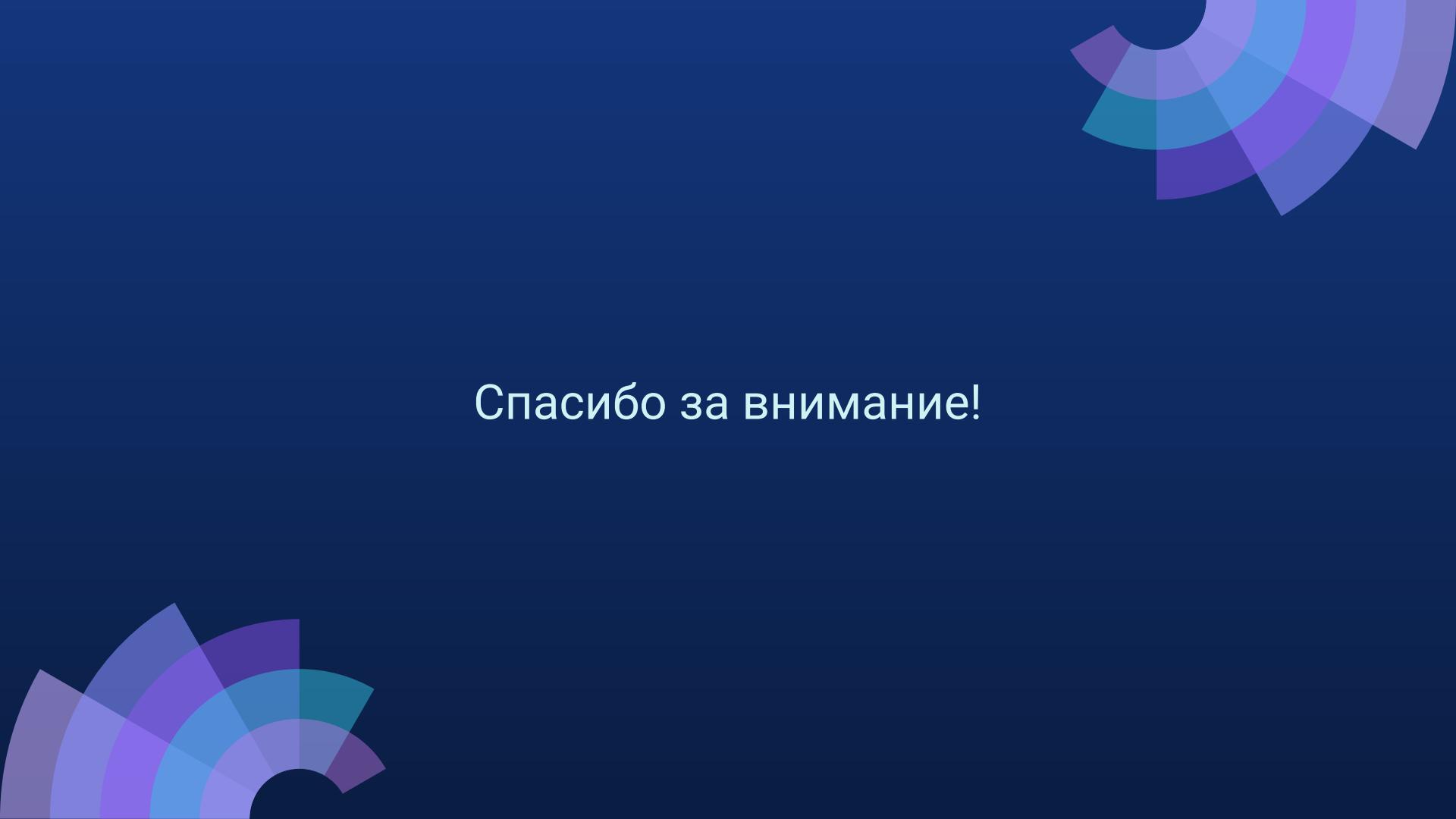


Дальнейшие планы

- Обучим YOLO v8/nas;
- Напишем Streamlit & FastAPI сервисы;
- Реализуем быстрый инференс модели и real-time тесты (Google Earth/Open Street Map).

UPDATE

Так как 21 февраля вышла Yolo v9, скорее всего будем обучать на ней



Спасибо за внимание!