

Детекция масок на лицах людей

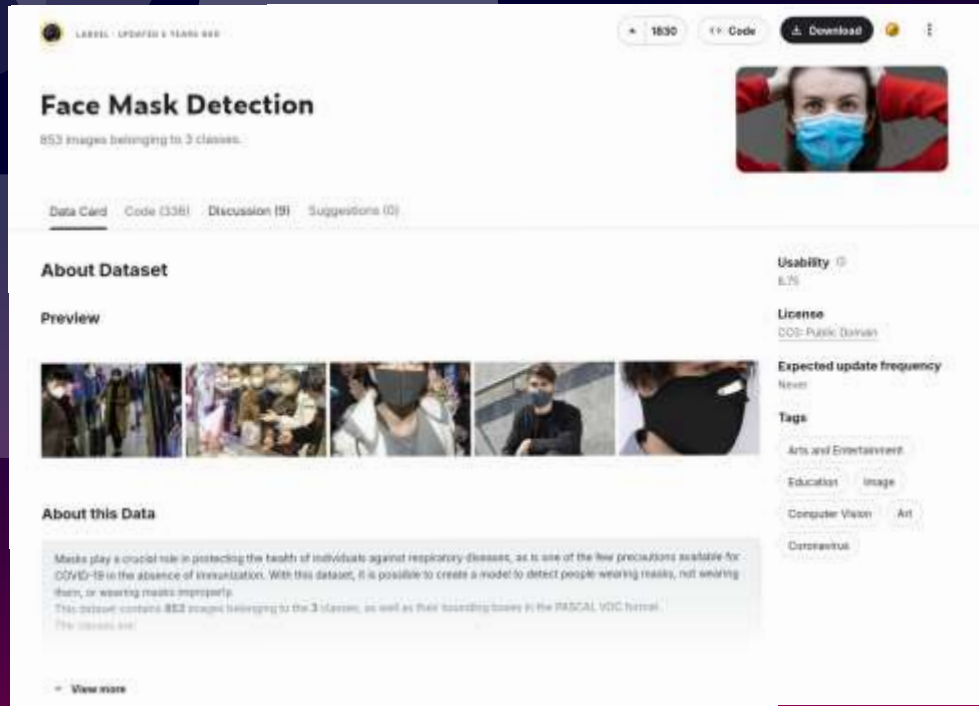
Мельников Андрей 22ПИ-1
Лебедев Тимур 22ПИ-3

Постановка задачи

Цель проекта — обучить модель для детекции медицинских масок на лицах людей.



Датасет



В качестве датасета был выбран датасет Face Mask Detection на kaggle. Данный датасет включает в себя 853 изображения с размеченными bbox-ами для отсутствия маски, наличия маски, и неправильно надетой маски на лице человека.

Выбор моделей

Для решения выбранной задачи детекции были использованы модели YOLO и FasterRCNN. Модель YOLO была обучена в 4 вариантах: yolov11n, yolov11s, yolov8n, yolov8s.



Как работают модели(YOLO)

YOLO — это алгоритм детекции объектов, который обрабатывает изображение за один проход через нейронную сеть. В отличие от методов, которые сначала предлагают регионы интереса (Region Proposals), а затем классифицируют их, YOLO предсказывает bounding box (ограничивающие рамки) и классы объектов одновременно.

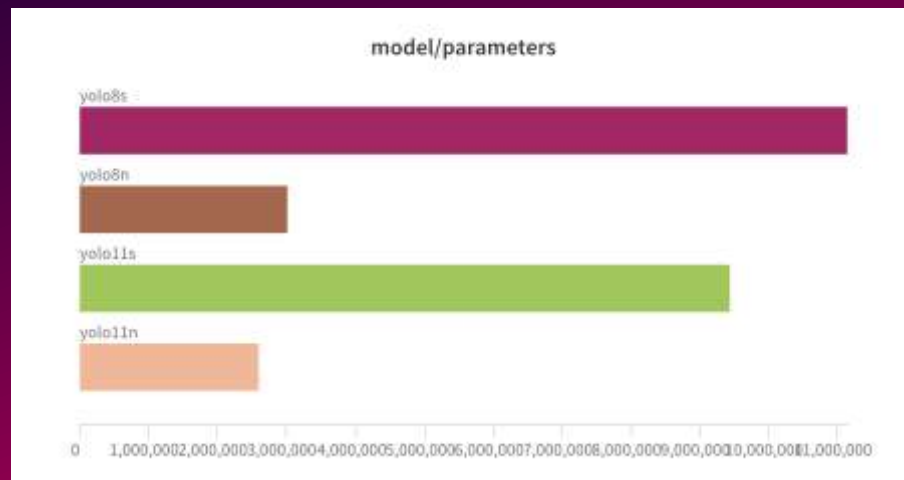
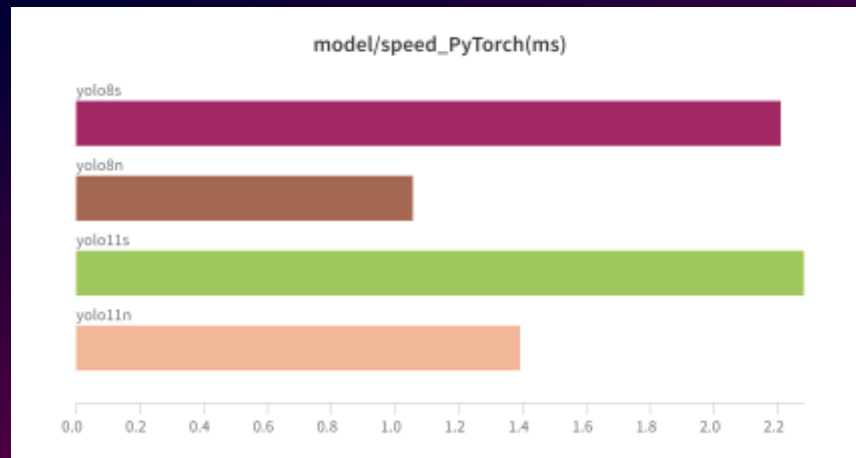
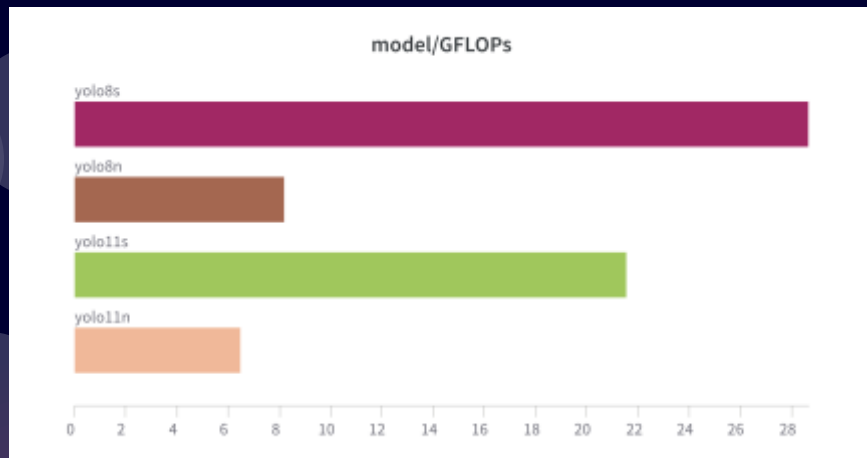
- Преимущества:
- Высокая скорость работы, так как изображение обрабатывается за один проход.
- Хорошо работает в реальном времени.
- Недостатки:
- Может хуже справляться с мелкими объектами или объектами, которые находятся близко друг к другу.

Как работают модели(FasterRCNN)

Faster R-CNN — это двухэтапный алгоритм детекции объектов. Сначала он генерирует регионы интереса (Region Proposals), а затем классифицирует их и уточняет bounding box.

- Преимущества:
- Высокая точность детекции, особенно для сложных сцен с множеством объектов.
- Хорошо справляется с объектами разных размеров.
- Недостатки:
- Медленнее, чем YOLO, из-за двухэтапного процесса.

Сравнение разных версий YOLO



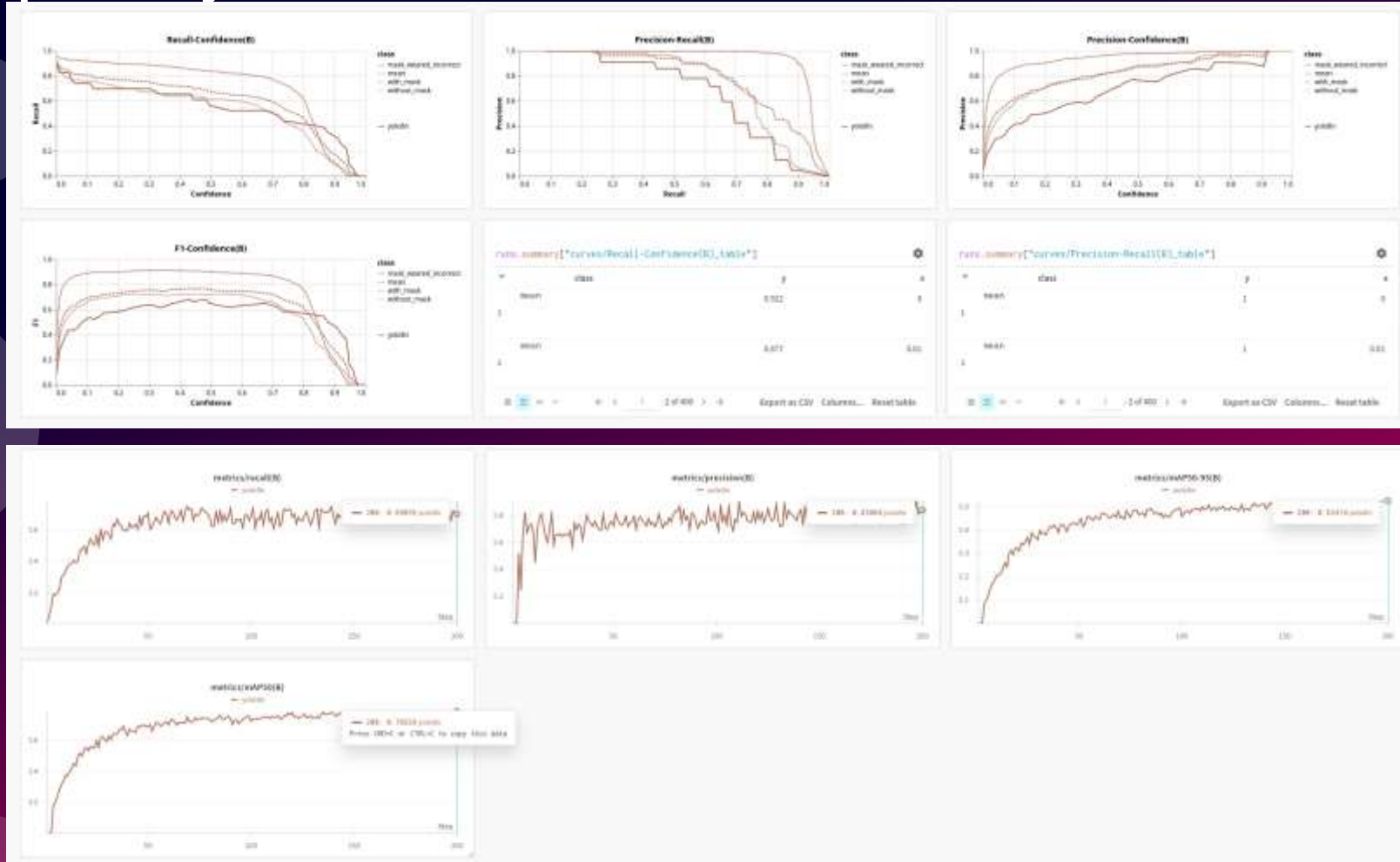
Метрики

Для оценки качества моделей были подсчитаны следующие метрики:

- Precision (P): $TP / (TP + FP)$
- Recall (R): $TP / (TP + FN)$
- mAP@0.5: Mean Average Precision при IoU 0.5.
- mAP@0.5:0.95: Average mAP при IoU от 0.5 до 0.95 (с шагом 0.05).
- F1-Score: Гармоническое среднее между Precision и Recall: $2 * (P * R) / (P + R)$



Метрики yolov8n



Результаты yolov8n



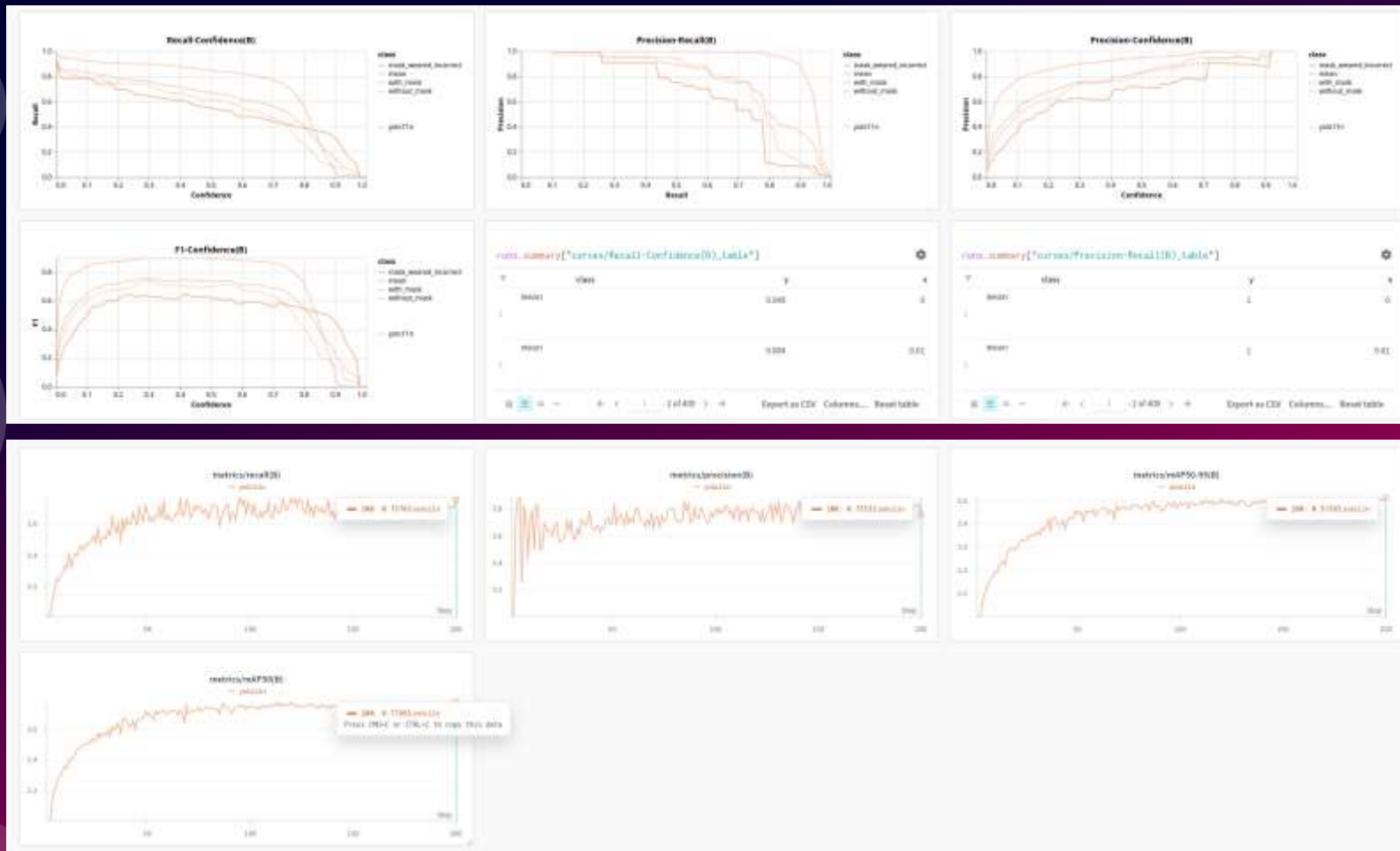
Метрики yolov8s



Результаты yolov8s



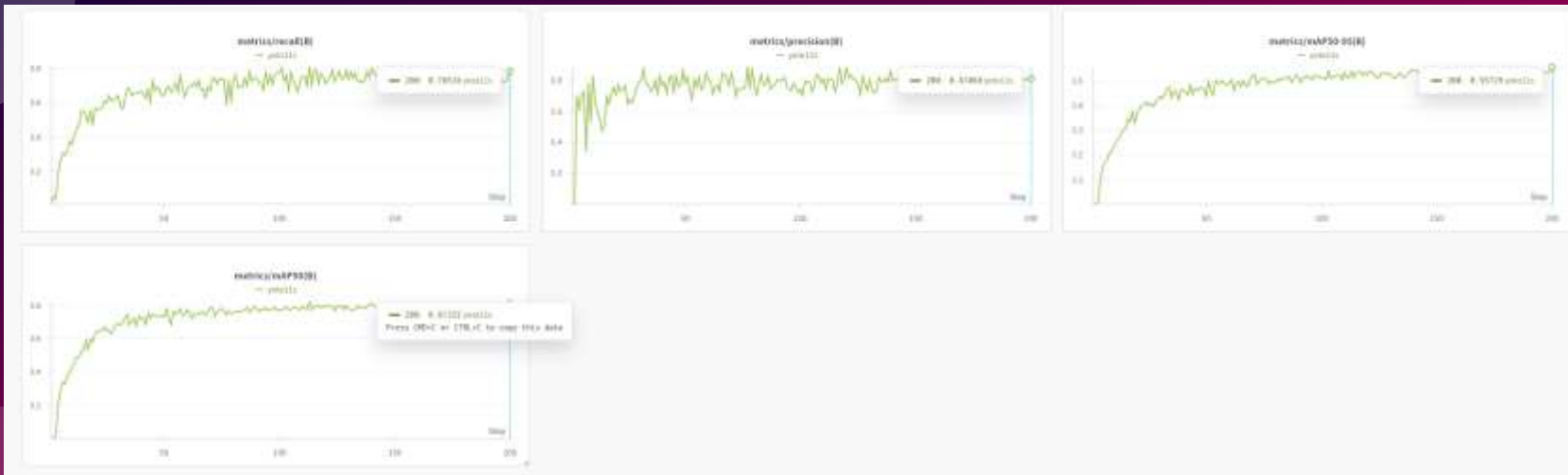
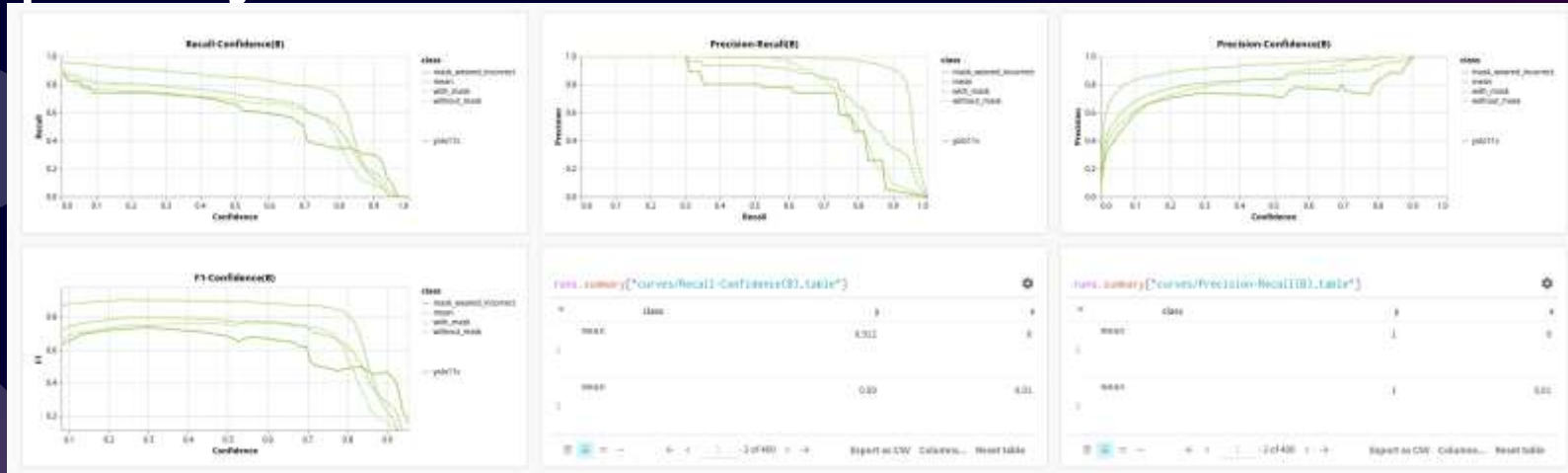
Метрики yolov11n



Результаты yolov11n



Метрики yolov11s



Метрики yolov11s



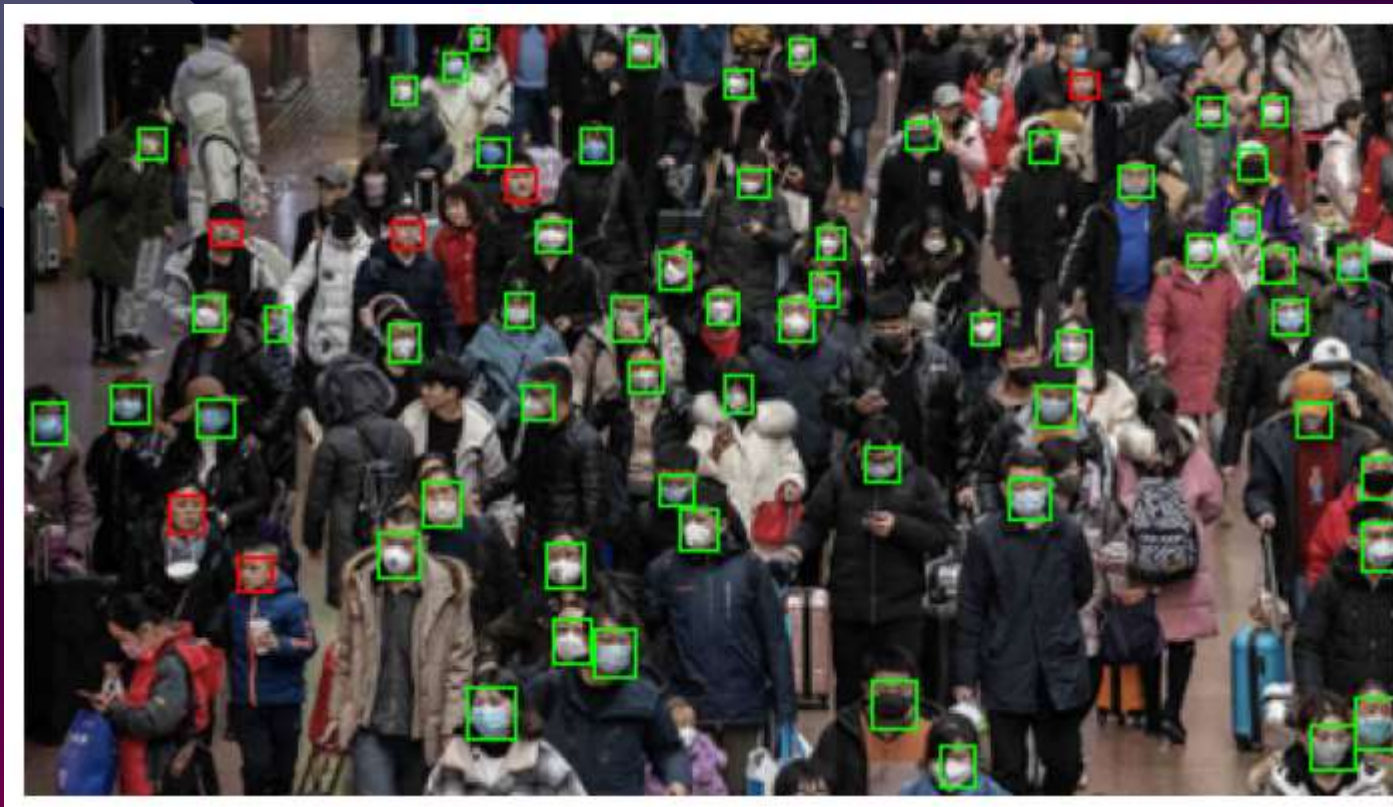
Метрики FasterRCNN

Average Precision	(AP) @[IoU=0.50:0.95	area= all	maxDets=100	= 0.747
Average Precision	(AP) @[IoU=0.50	area= all	maxDets=100	= 0.817
Average Precision	(AP) @[IoU=0.75	area= all	maxDets=100	= 0.812
Average Precision	(AP) @[IoU=0.50:0.95	area= small	maxDets=100	= 0.758
Average Precision	(AP) @[IoU=0.50:0.95	area=medium	maxDets=100	= 0.813
Average Precision	(AP) @[IoU=0.50:0.95	area= large	maxDets=100	= 0.741
Average Recall	(AR) @[IoU=0.50:0.95	area= all	maxDets= 1	= 0.248
Average Recall	(AR) @[IoU=0.50:0.95	area= all	maxDets= 10	= 0.722
Average Recall	(AR) @[IoU=0.50:0.95	area= all	maxDets=100	= 0.784
Average Recall	(AR) @[IoU=0.50:0.95	area= small	maxDets=100	= 0.764
Average Recall	(AR) @[IoU=0.50:0.95	area=medium	maxDets=100	= 0.841
Average Recall	(AR) @[IoU=0.50:0.95	area= large	maxDets=100	= 0.866

```
print(f"mAP@0.5: {coco_eval.stats[1]}")
print(f"mAP@0.5:0.95: {coco_eval.stats[0]}")
print(f"Precision: {coco_eval.stats[5]}")
print(f"Recall: {coco_eval.stats[11]}")
```

```
mAP@0.5: 0.8172210956258421
mAP@0.5:0.95: 0.7472871227642387
Precision: 0.7407044994244809
Recall: 0.865625
```

Результаты FasterRCNN



Выводы

Нами было протестировано 5 моделей для детекции. Мы решали задачу детекции на 3 класса: наличие маски, отсутствие маски, и неправильно надетая маска. Лучший результат показала модель FasterRCNN, обогнав по метрикам все остальные модели. YOLO11s и YOLO8s оказались на втором месте, и хотя восьмая модель показала себя чуть лучше, она тяжелее по числу параметров и дольше работает. Последние места заняли YOLO11n и YOLO8n, хотя они и показали довольно неплохие результаты для их размера и производительности.

ССЫЛКИ

- <https://github.com/aeksin/Mask-detector>
- https://wandb.ai/aerrries-hse/mask_detection?nw=nwuseraerrries