

Курсовая работа на тему «Детекция и распознавание объектов внутри помещений с применением технологии YoloV8»

Проверил: старший преподаватель Ибатулин М.Ю.
Выполнили студенты группы ИДБ-22-12:
Колдаева У. С.,
Харитонова А. А.

Москва, 2025 у.г.

Creative title here

Команда проекта

W W W . W E B S I T E . C O M



Колдаева Ульяна

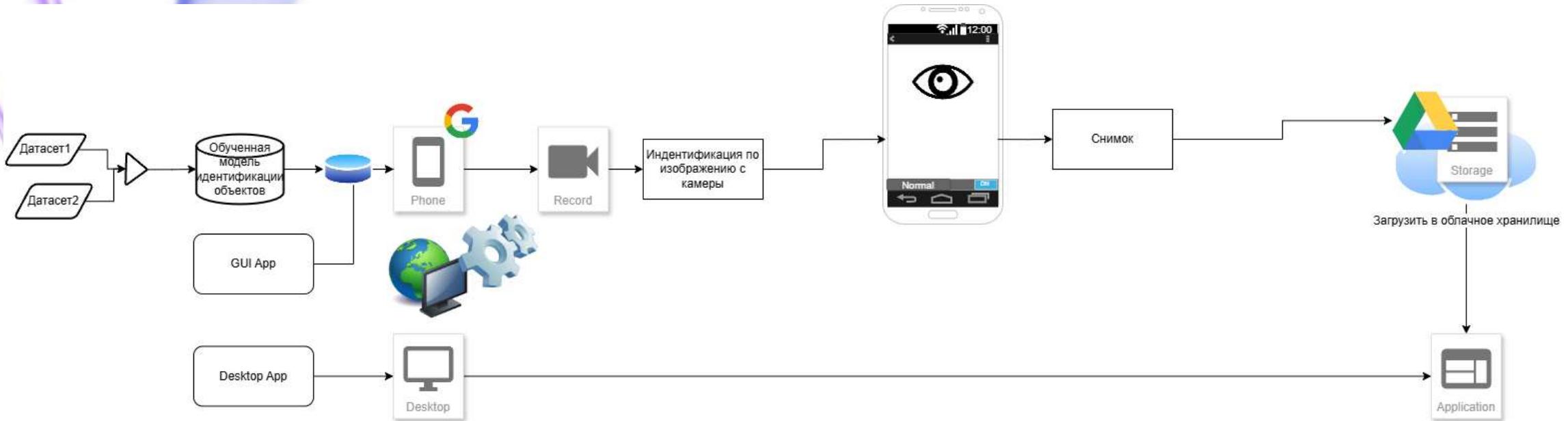
Харитонова Анастасия

Участие в проекте:

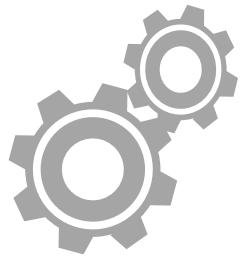
100%

100%

Прототип решения



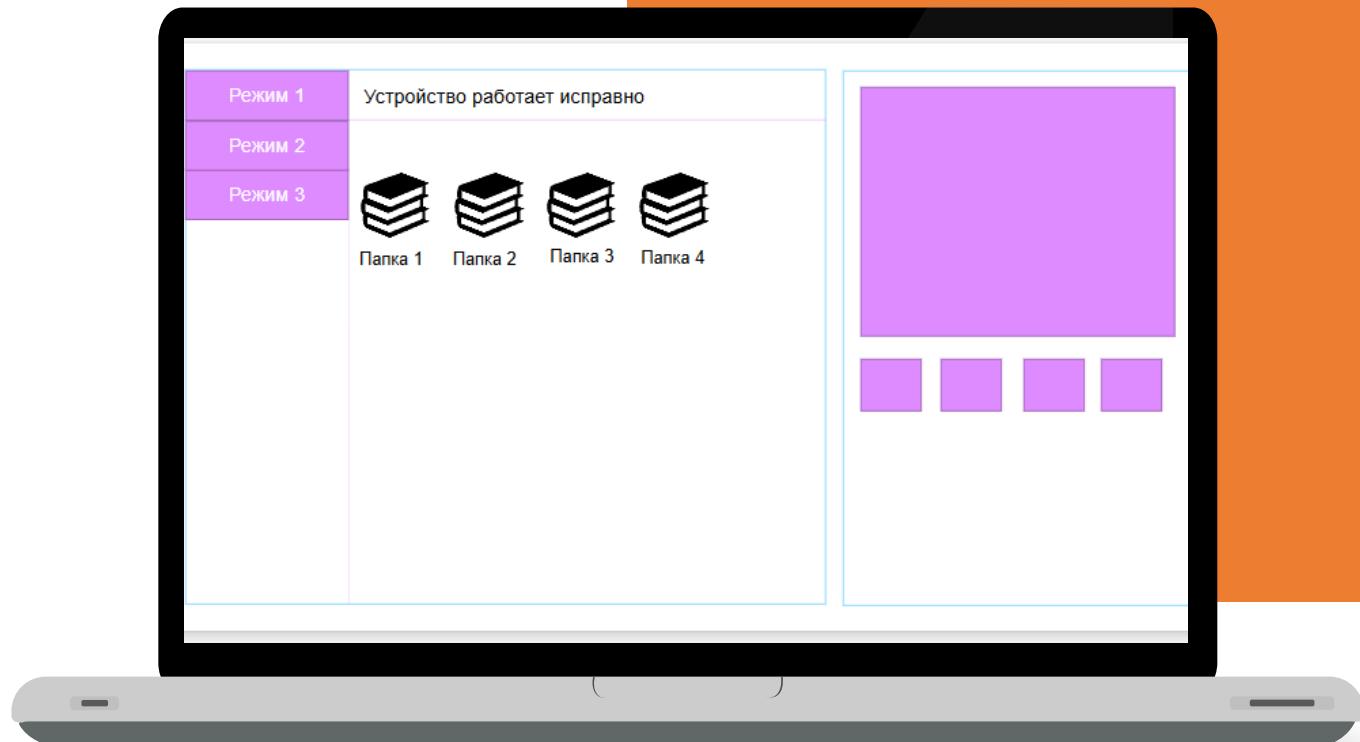
Приложение



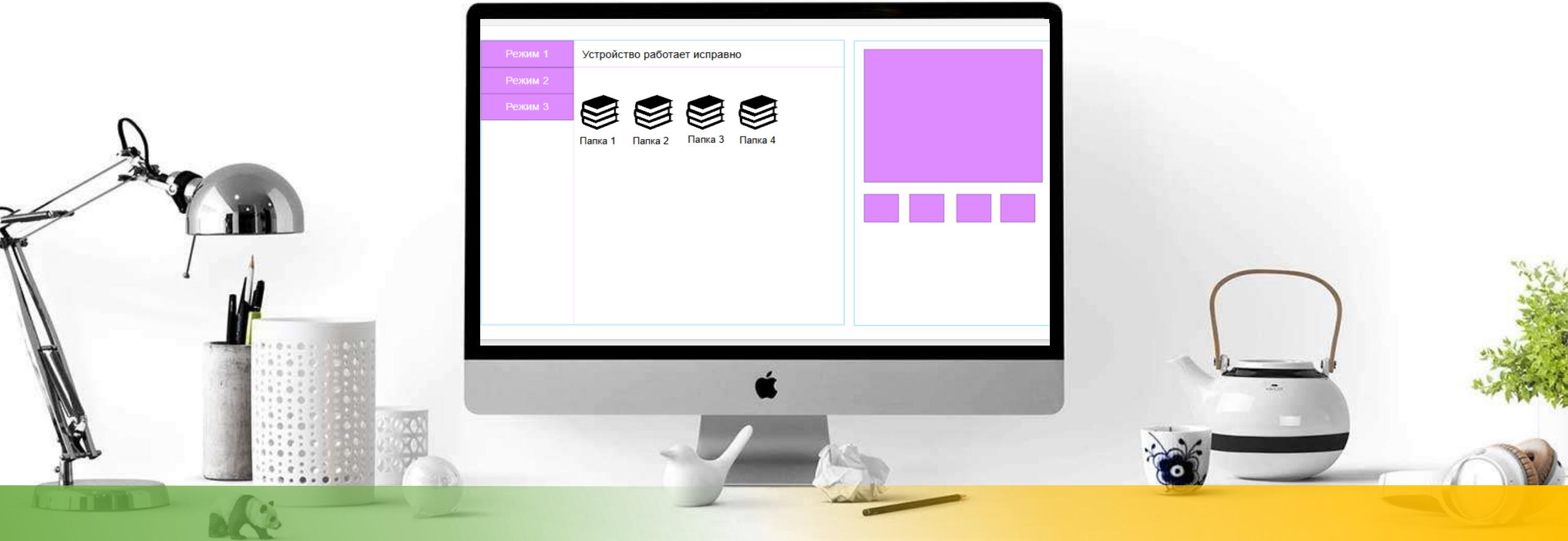
1.Отладка, привыкание - когда пользователь только учится распознаванию объектов.

2.Координация, корректировка - сбор информации и ввод дополнительных настроек, предложенных ИИ на базе собранных изображений.

3.Тестирование и отчёт об ошибках - вся собранная за время работы устройства статистика может быть выгружена для дальнейшей работы над устройством.



Интеграция облачных сервисов



Передача
Данных
С устройства

Доступ
К данным
С рабочей машины

Приложение



План реализации:

- 1) Проектирование и анализ будущей системы
- 2) Разработка модуля на прототипе
- 3) Тестирование модуля
- 4) Обучение модели на датасетах(самостоятельно собранные + интернет kaggle)
- 5) Интеграция в приложение для мобильного телефона

Библиотеки

1
In [1]: !pip install ultralytics # Установка YOLOv8

!pip install roboflow

from roboflow import Roboflow

Подключение Кеггл

2
In [2]: !pip install kagglehub

Requirement already satisfied: kagglehub in /usr/local/lib/python3.11/dist-packages
as /usr/local/lib/python3.11/dist-packages/kagglehub-0.1.1-py3.11.egg

Импорт датасет Кеггл

3
In [3]: import kagglehub

Download latest version
path = kagglehub.dataset_download("thebjordin/indoor-object-detection")

print("Path to dataset files:", path)

Path to dataset files: /kaggle/input/indoor-object-detection

Проверка data.yaml

4
In [4]: !cat /kaggle/input/indoor-object-detection/data.yaml

```
train: /content/data/train/images
val: /content/data/valid/images
test: /content/data/test/images
nc: 10
names:
- door
- cabinetDoor
- refrigeratorDoor
- window
- chair
- table
- cabinet
- couch
- openedDoor
- pole
```

Повторное подключение библиотеки

5
In [5]: !pip install ultralytics

Проверка наличия изображений

6
In [6]:

```
import os  
  
# Проверка существования путей  
paths = [  
    '/kaggle/input/indoor-object-detection/test',  
    '/kaggle/input/indoor-object-detection/train',  
    '/kaggle/input/indoor-object-detection/valid',  
    '/kaggle/input/indoor-object-detection/data.yaml'  
]  
  
for path in paths:  
    exists = os.path.exists(path)  
    print(f"(path): {'✓' if exists else '✗'}")  
  
if exists and path.endswith('images'):  
    print(f"Найдено изображений: {len(os.listdir(path))}")
```

Настройка Yolo

7
In [7]:

/kaggle/input/indoor-object-detection/test: ✓
/kaggle/input/indoor-object-detection/train: ✓
/kaggle/input/indoor-object-detection/valid: ✓
/kaggle/input/indoor-object-detection/data.yaml: ✓

In [8]: # Обновляем настройки Ultralytics

yolo settings datasets_dir=/content
yolo settings runs_dir=/content

Проверяем настройки
yolo settings

Перемещение датасета

8

Перемещаем датасет

Для использования и изменения data.yaml

```
In [1]: import shutil  
import os  
  
# Путь к исходному датасету  
source_path = "/root/.cache/kagglehub/datasets/theprbordin/indoor-object-detection"  
  
# Путь назначения  
destination_path = "/content/data/indoor-object-detection/"  
  
# Создание директории назначения (если не существует)  
os.makedirs(destination_path, exist_ok=True)  
  
# Копирование датасета  
shutil.copytree(source_path, destination_path, dirs_exist_ok=True)  
  
print(f"Датасет скопирован в {destination_path}")  
  
Датасет скопирован в /content/data/indoor-object-detection/  
  
In [1]: from ultralytics import YOLO  
  
# Очистка кэша  
!rm -rf /root/.cache/ultralytics  
  
model = YOLO('yolov8n.pt')  
results = model.train(  
    data='/content/data/indoor-object-detection/data.yaml',  
    epochs=50,  
    imgsz=640,  
    batch=8, # Уменьшили batch для надежности  
    device='0',  
    workers=4, # Уменьшили количество workers  
    name='indoor_outdoor_final_fix',  
    exist_ok=True,
```

Запуск обучения

9

```
Validating /content/detect/indoor_outdoor_final_fix/weights/best.pt...  
Ultralytics 8.3.139 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095M  
IB)  
Model summary (fused): 72 layers, 3,007,598 parameters, 0 gradients, 8.1 GFLOPS  
P50-95%: 100% |██████████| 15/15 [00:03<00:08, 4.83it/s]  
Class Images Instances Box(P) R mAP50 mA
```

Результаты обучения по классам

10

		all	238	1289	0.349	0.215	0.22
0.152	door	63	97	0.0426	0.0206	0.0359	
0.0127	cabinetDoor	99	765	0.594	0.209	0.325	
0.127	refrigeratorDoor	85	192	1	0	0.0377	
0.0179	window	39	91	0.26	0.143	0.104	
0.0521	chair	24	49	0.422	0.342	0.299	
0.18	table	38	40	0.206	0.125	0.113	
0.0542	cabinet	28	32	0.239	0.125	0.15	
0.115	couch	1	1	0.36	1	0.995	
0.895	openedDoor	13	13	0.058	0.0769	0.0228	
0.00787	pole	4	9	0.305	0.111	0.117	
0.0585							

Speed: 0.3ms preprocess, 2.4ms inference, 0.0ms loss, 3.3ms postprocess per image
Results saved to /content/detect/indoor_outdoor_final_fix

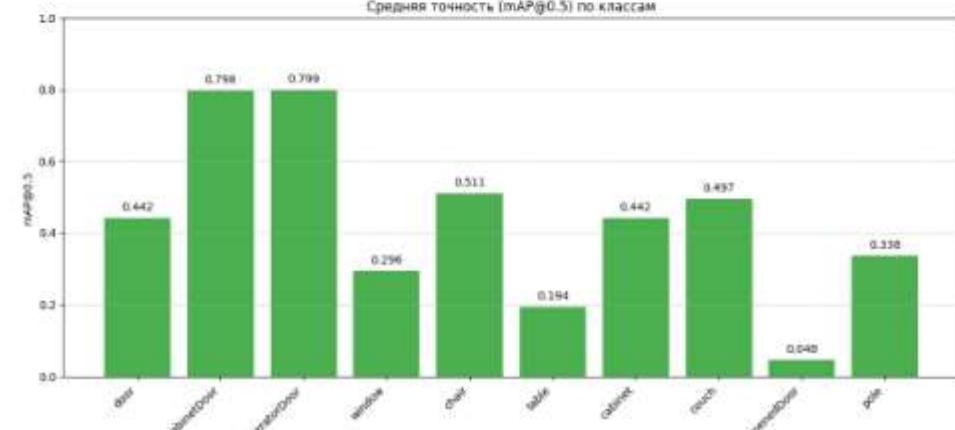
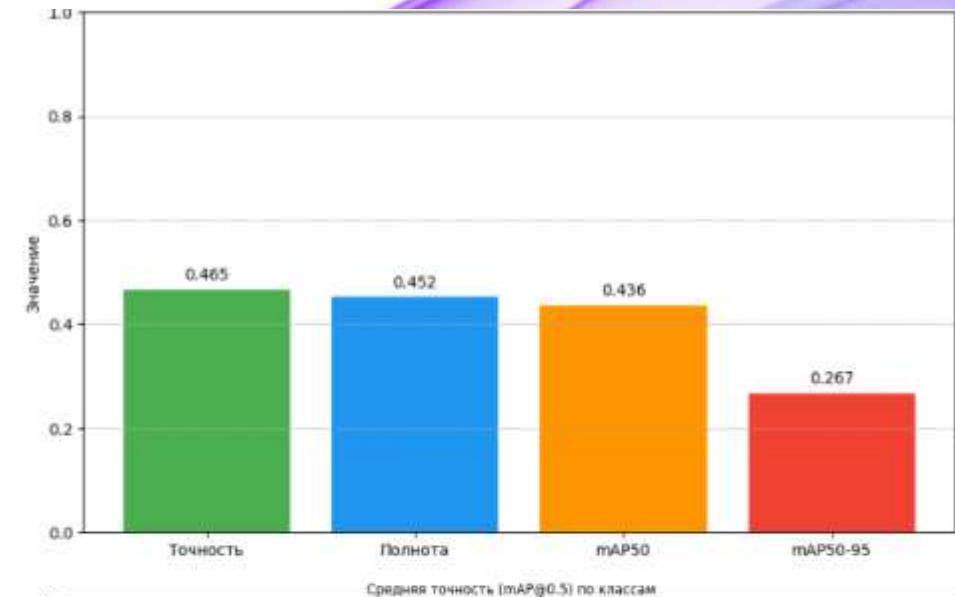
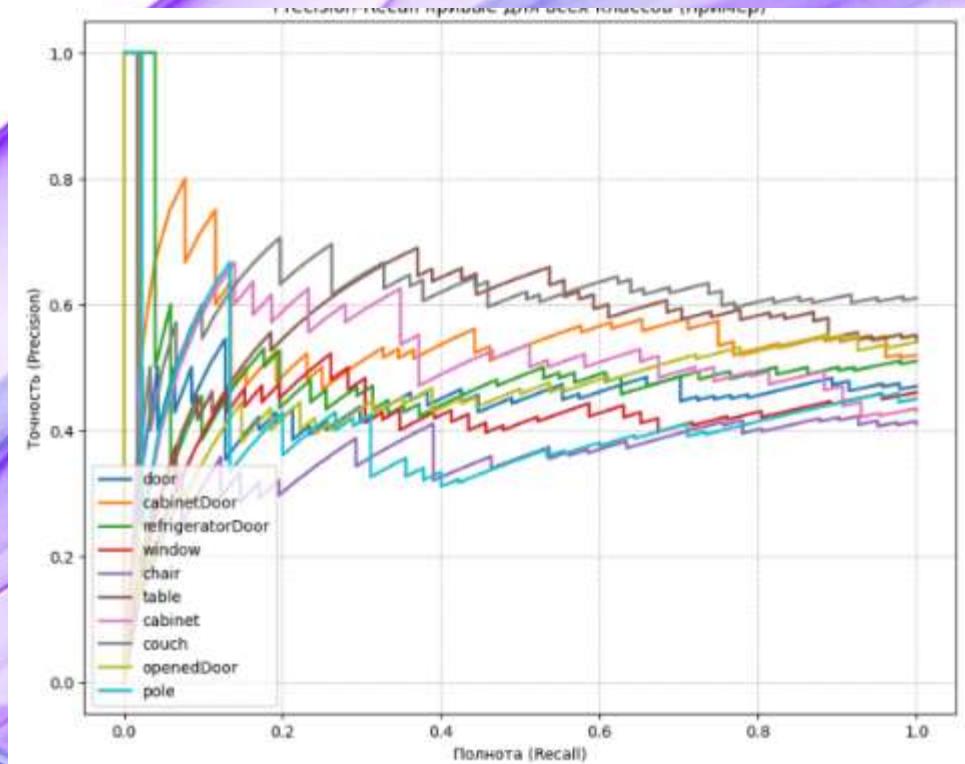
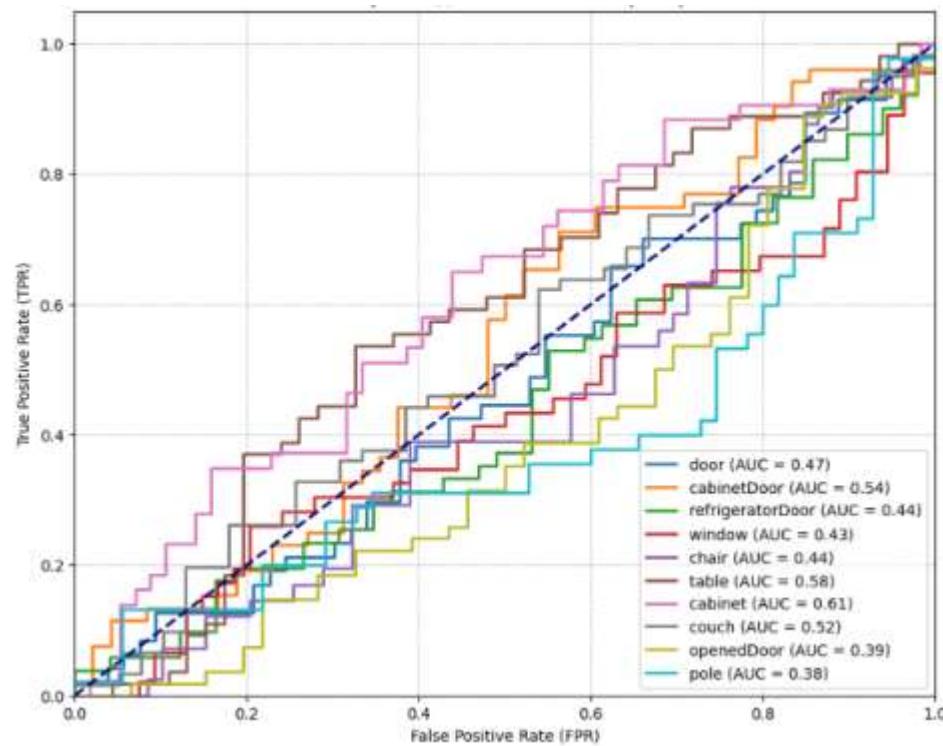


Таблица метрик для всех классов

Класс	Точность	Полнота	mAP@0.5	mAP@0.5:0.95
0 all	0.465000	0.452000	0.436000	0.267000
1 door	0.472000	0.464000	0.442000	0.288000
2 cabinetDoor	0.758000	0.769000	0.798000	0.450000
3 refrigeratorDoor	0.791000	0.711000	0.799000	0.530000
4 window	0.466000	0.275000	0.296000	0.194000
5 chair	0.484000	0.510000	0.511000	0.260000
6 table	0.341000	0.250000	0.194000	0.103000
7 cabinet	0.409000	0.531000	0.442000	0.313000
8 couch	0.452000	0.903000	0.497000	0.348000
9 openedDoor	0.000000	0.000000	0.048000	0.023000
10 pole	0.471000	0.111000	0.338000	0.159000



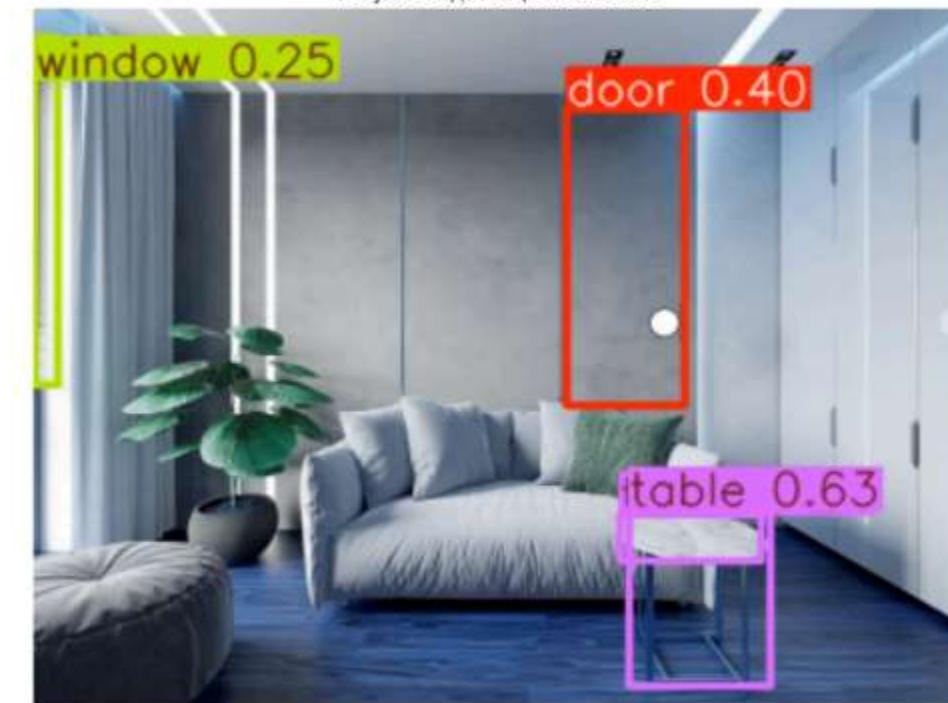
Исходное изображение



Пример детекции после 1 обучения

(1, 3, 480, 640)

Результат детекции объектов



Исходное изображение

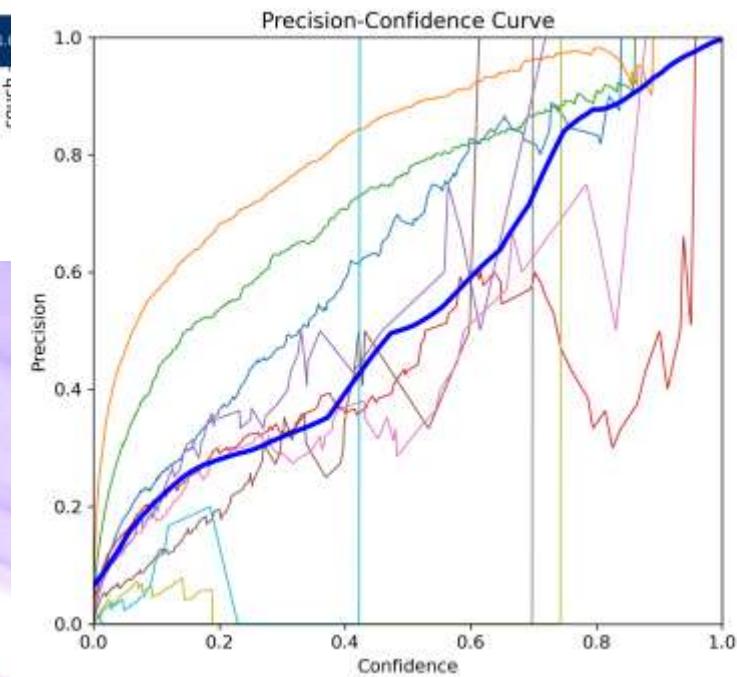
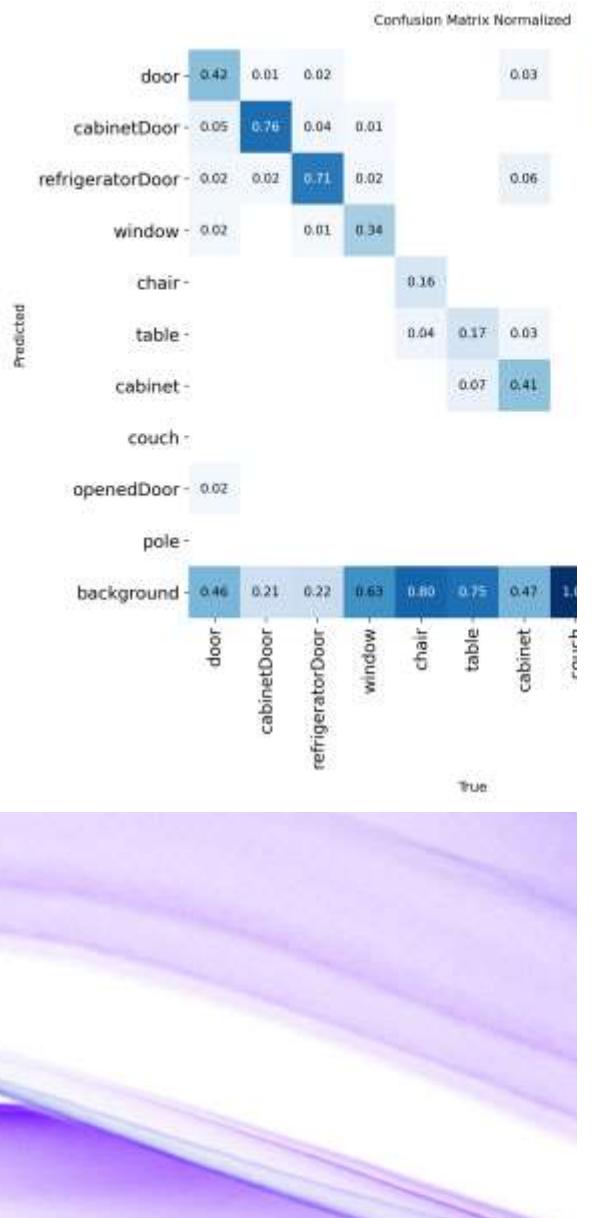


Пример детекции после 1 обучения

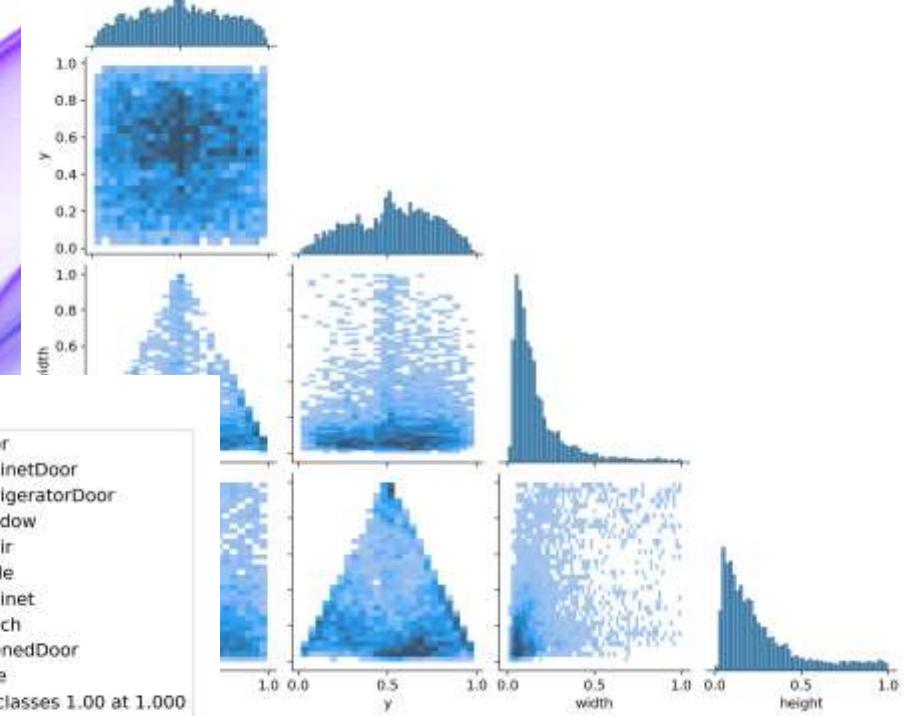
(1, 3, 448, 640)



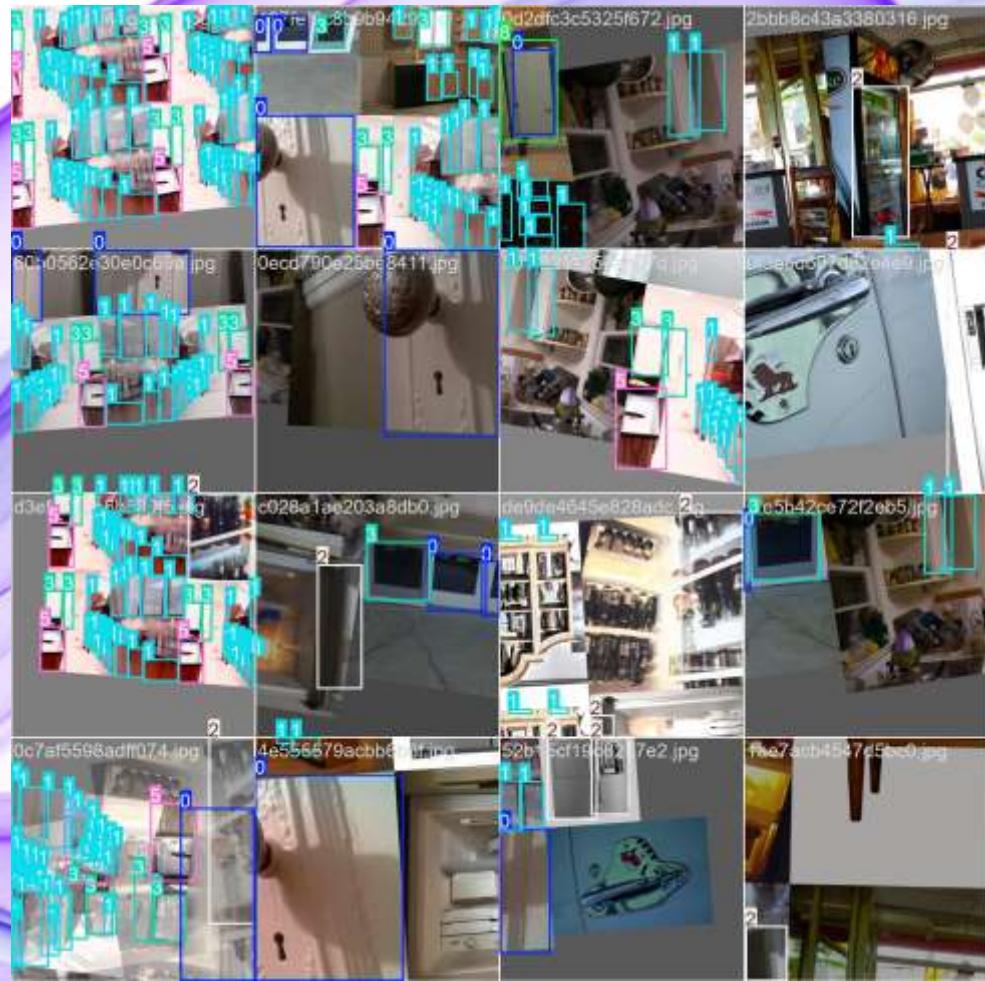
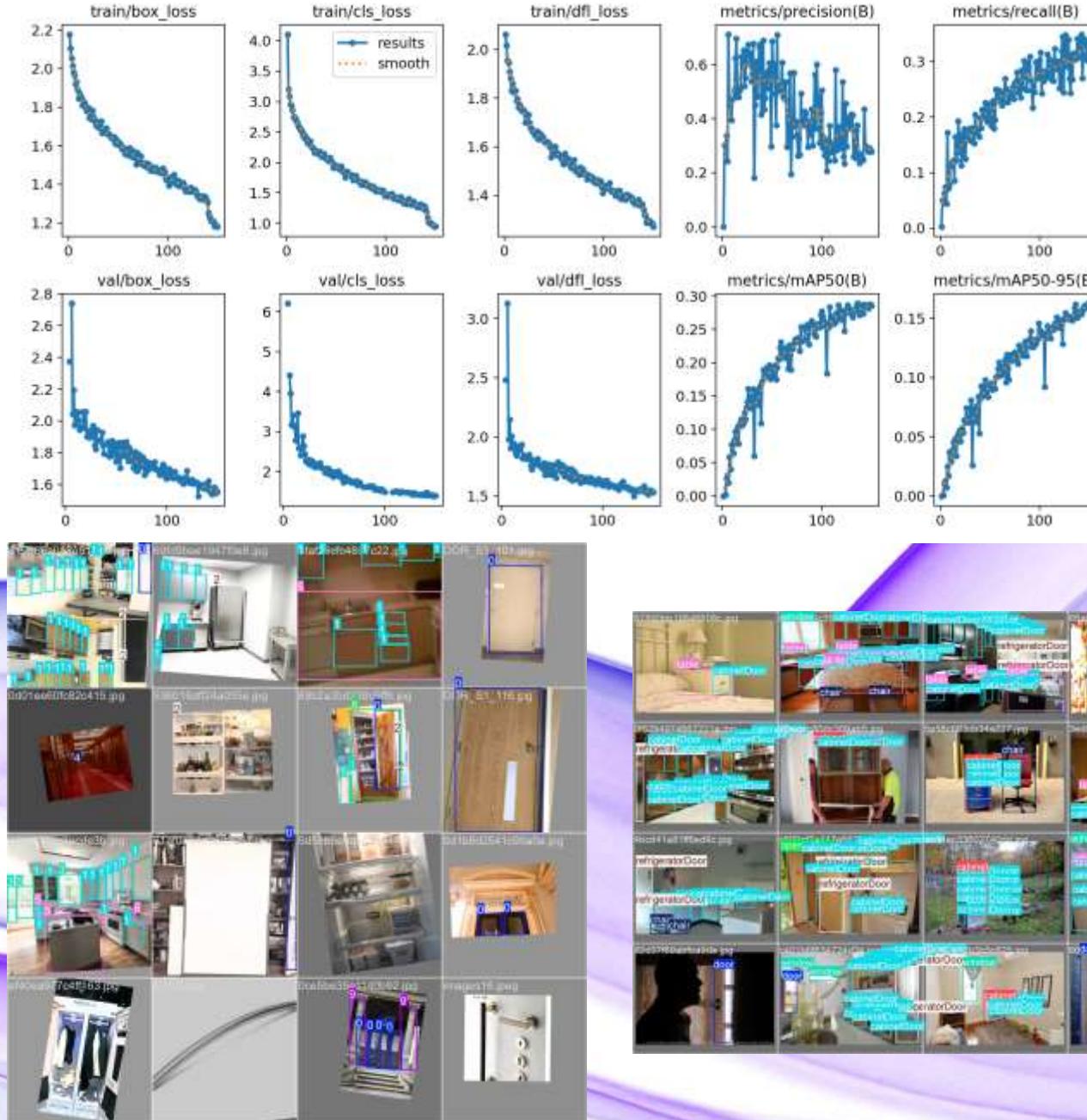
Дообучаем, увеличим количество эпох до 100

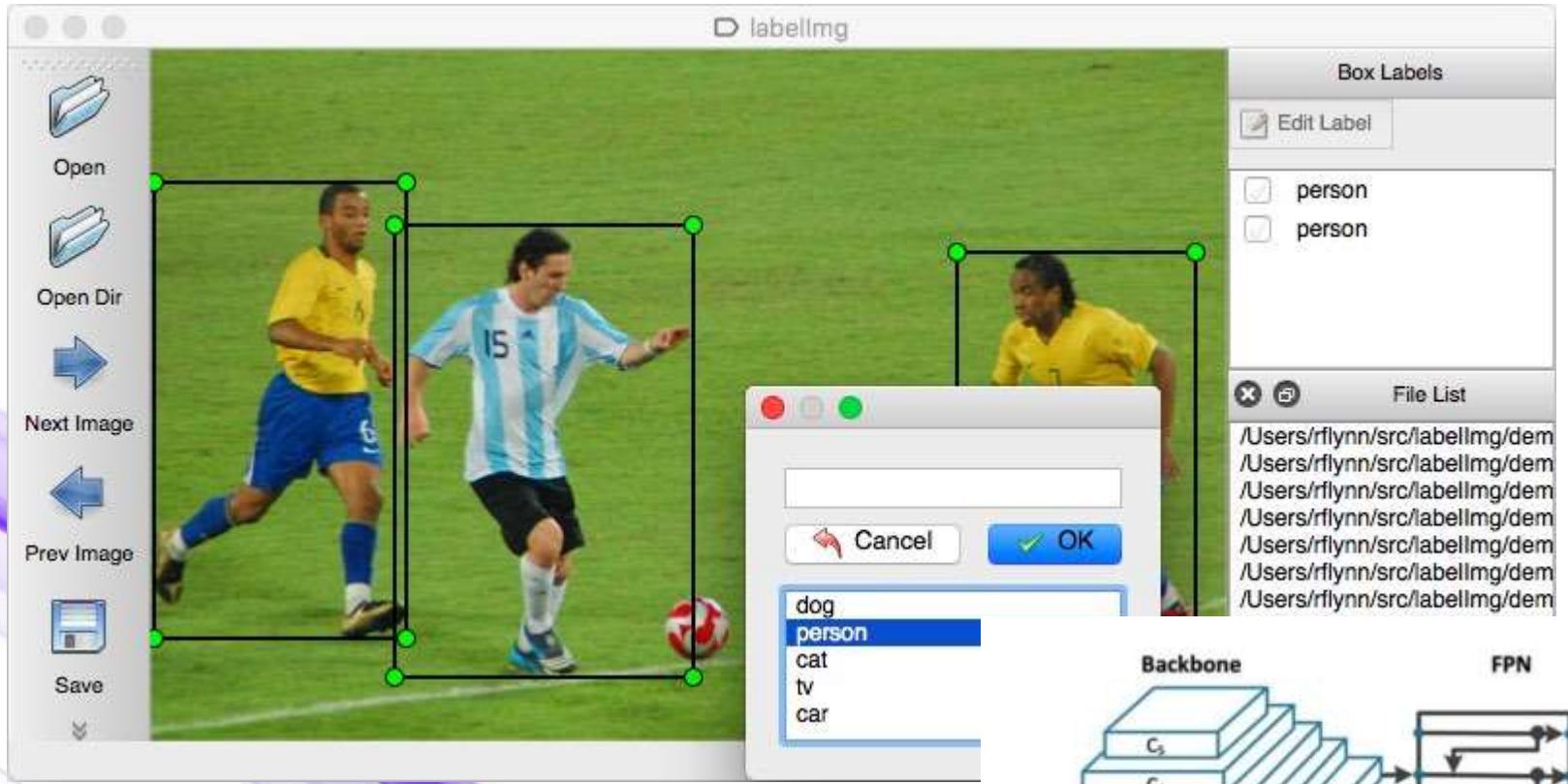


door
cabinetDoor
refrigeratorDoor
window
chair
table
cabinet
couch
openedDoor
pole
all classes 1.00 at 1.000

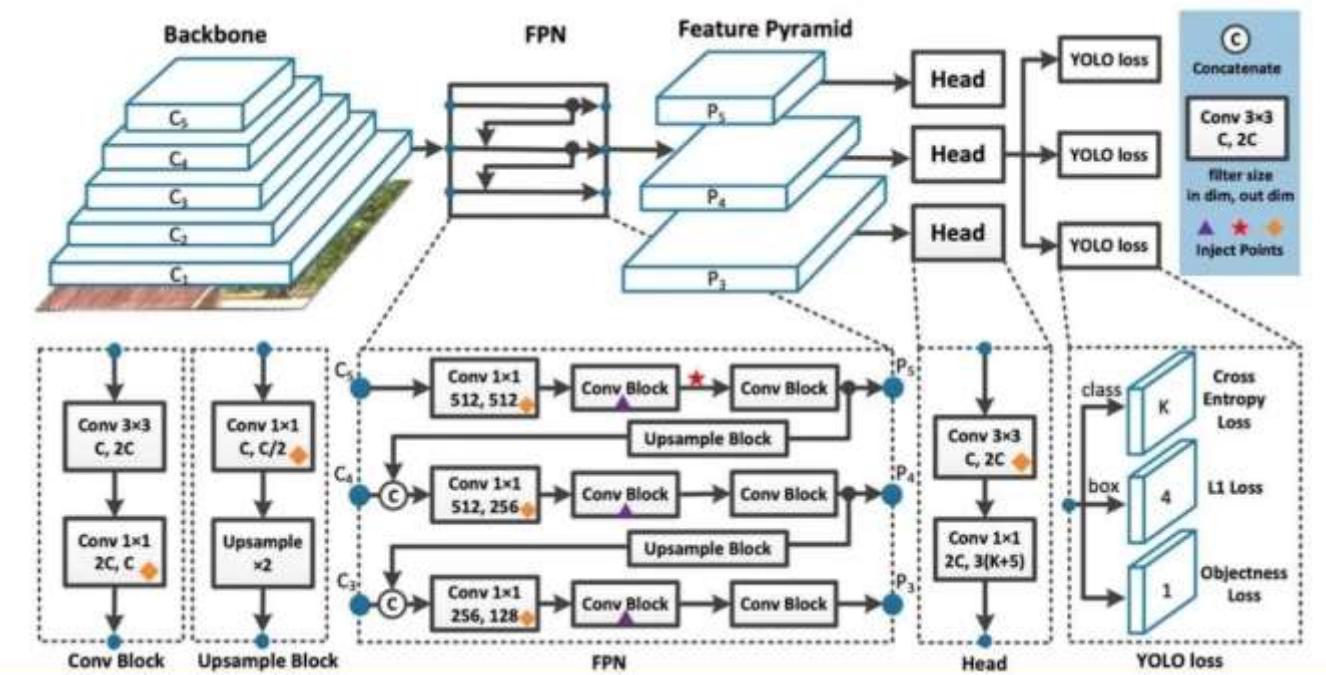


Анализ обучения после дообучения

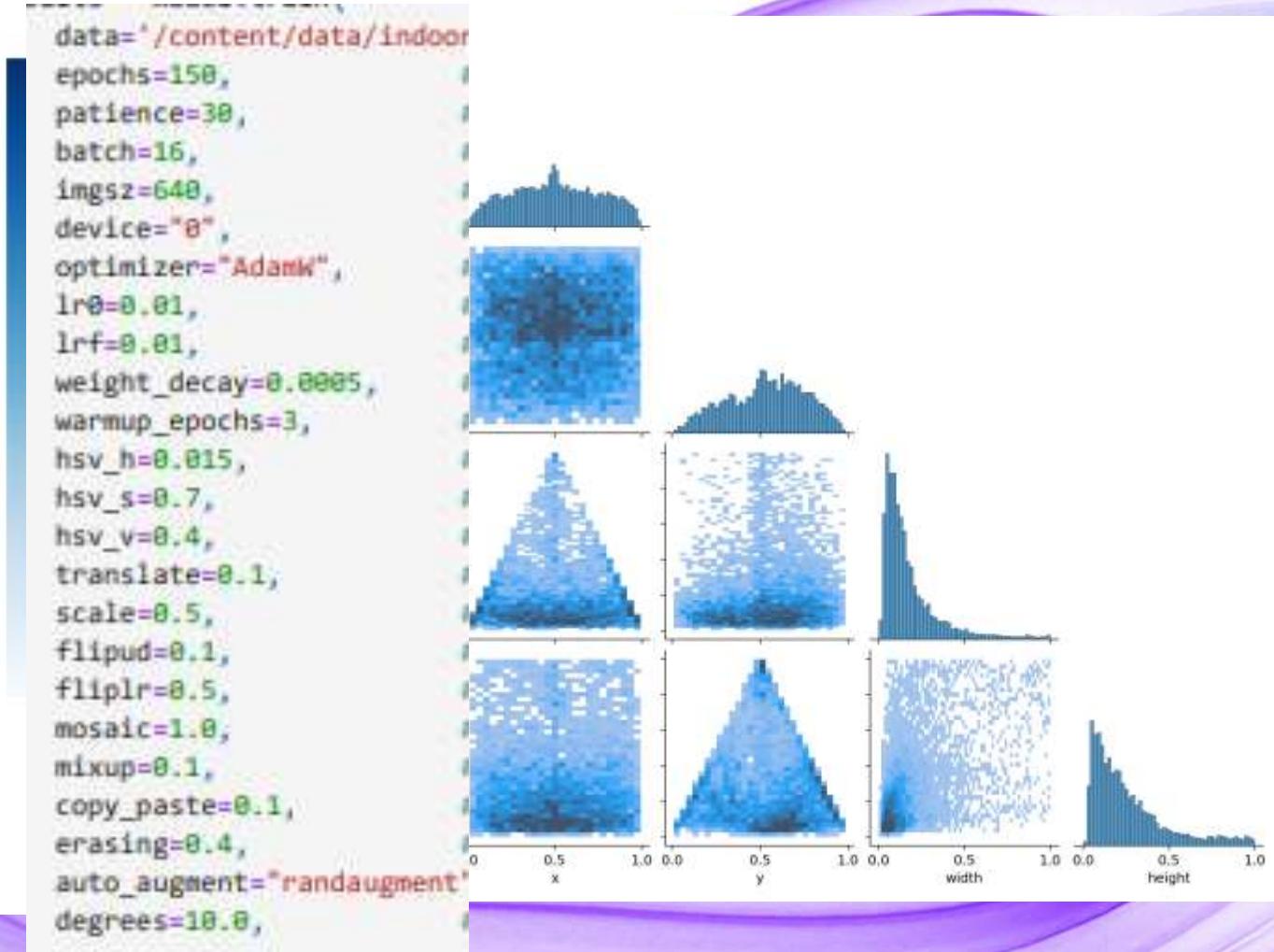
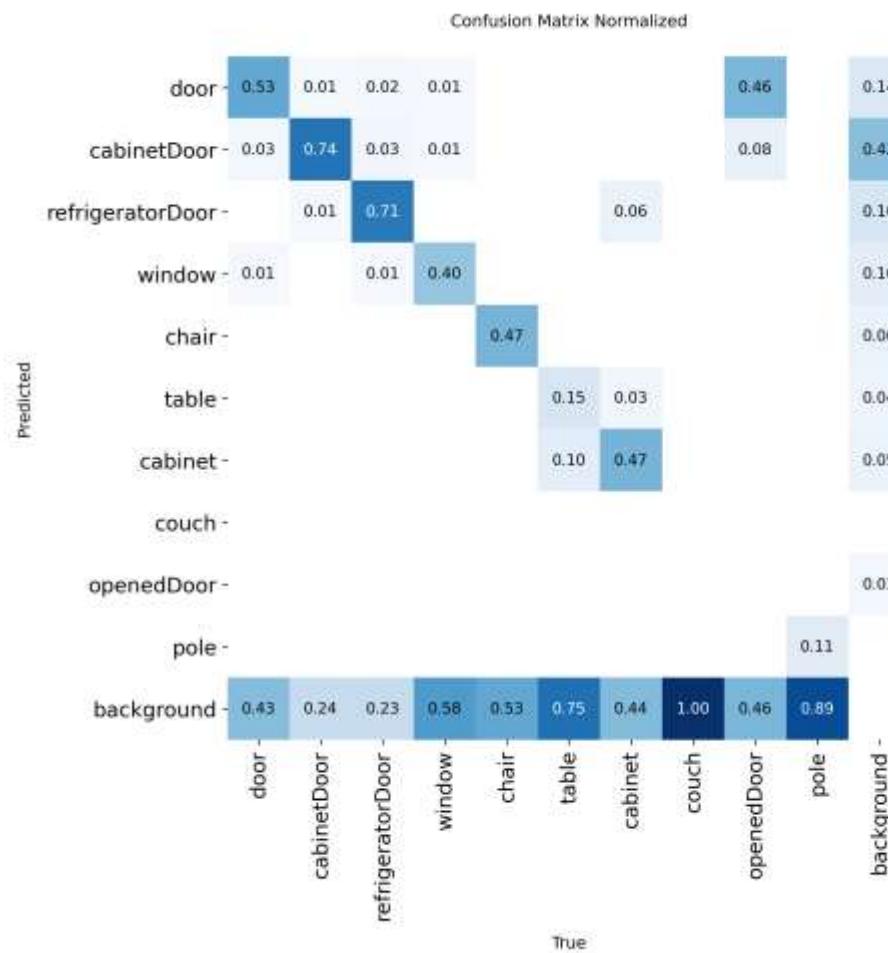




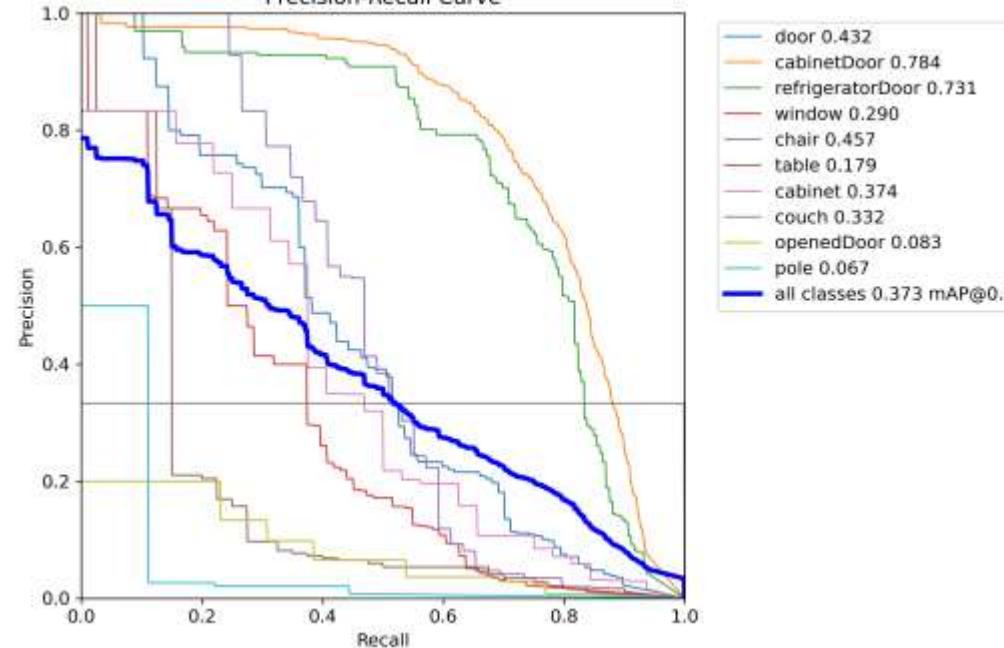
Использование разметки Yolo



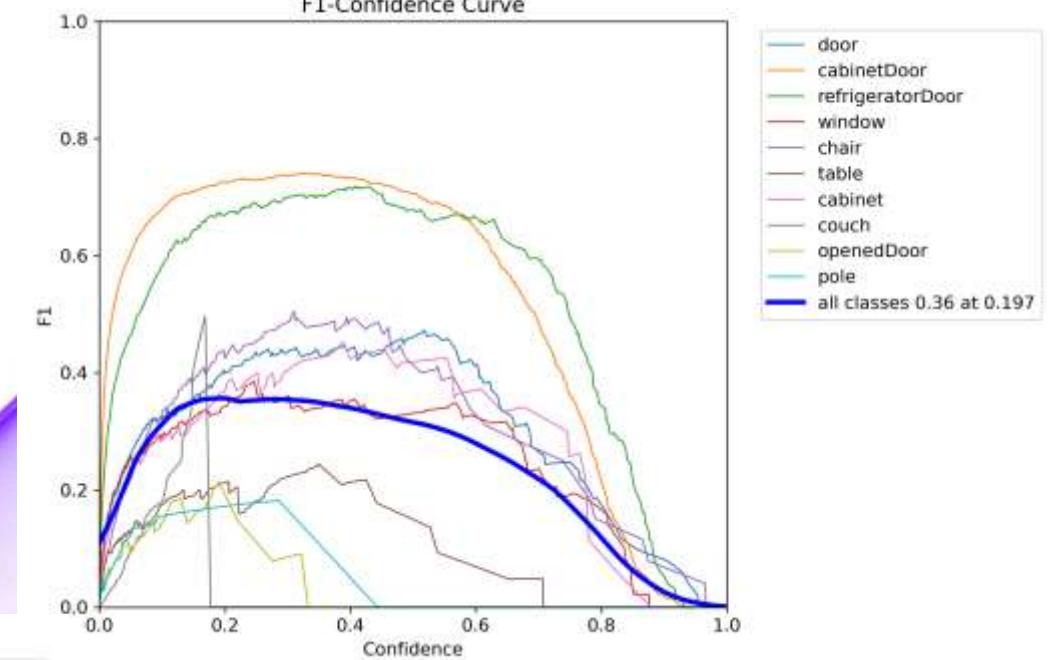
Новые параметры обучения



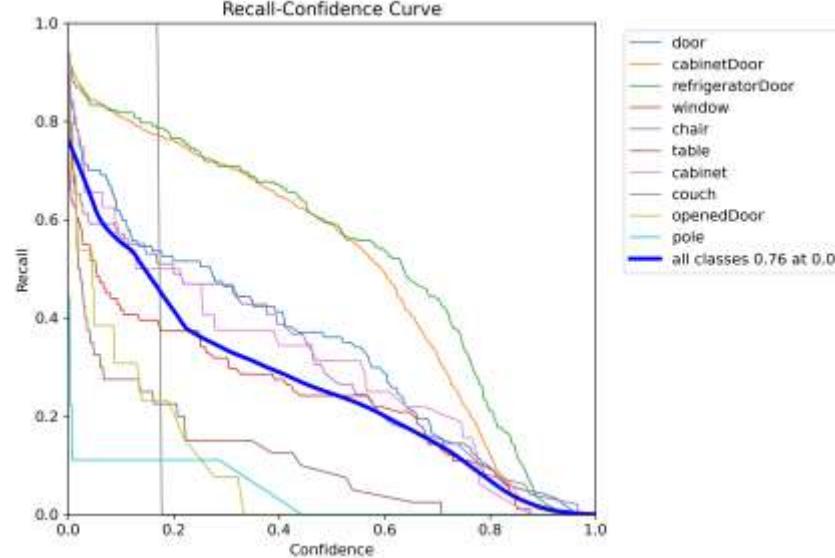
Precision-Recall Curve

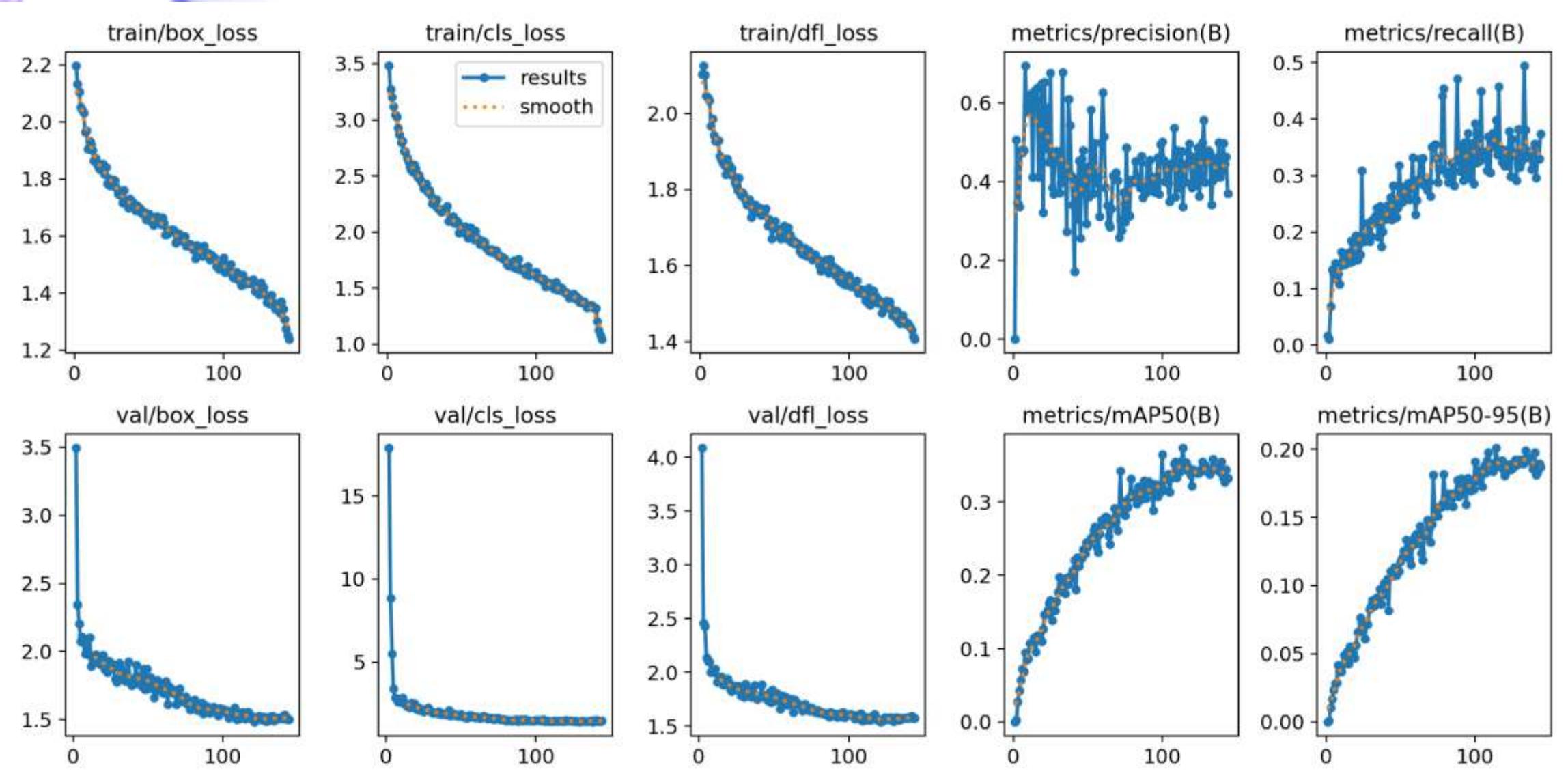


F1-Confidence Curve

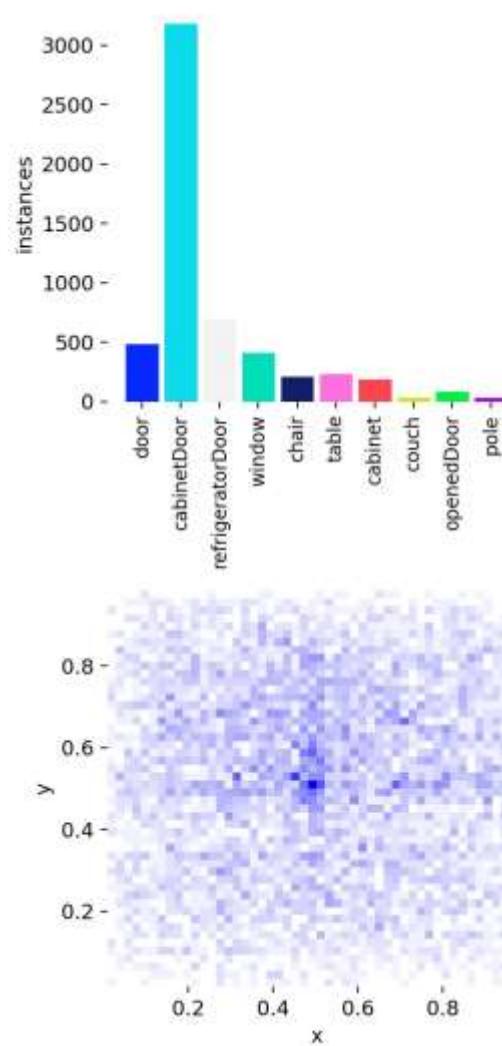


Recall-Confidence Curve

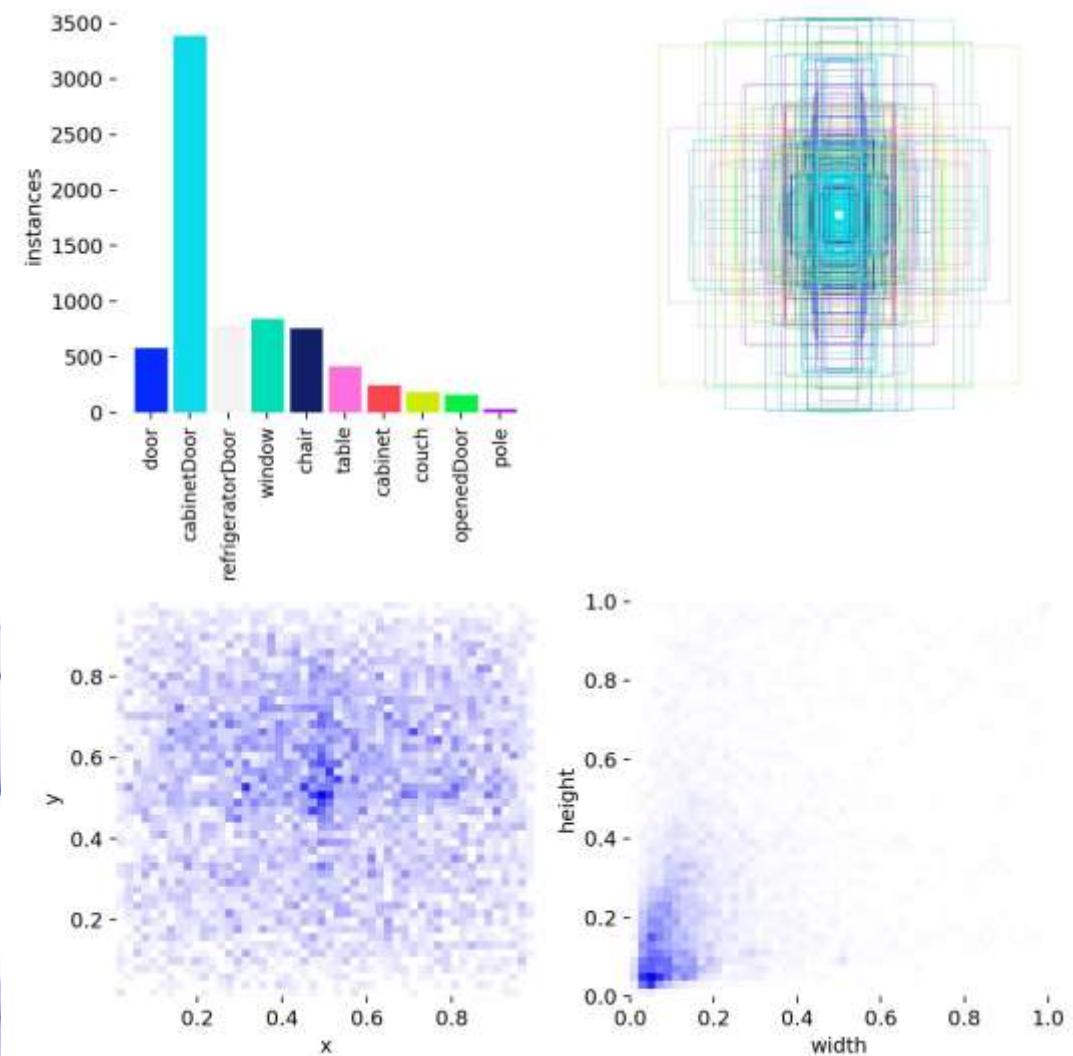




До



После



До(после дообучения исходного датасета)

46 4 doors, 18 cabinetDoors, 1 refrigeratorDoor, 1 table, 1 openedDoor, 29.1ms
Speed: 2.9ms preprocess, 59.1ms inference, 2.3ms postprocess per image at shape
(1, 3, 480, 640)

Результат детекции объектов

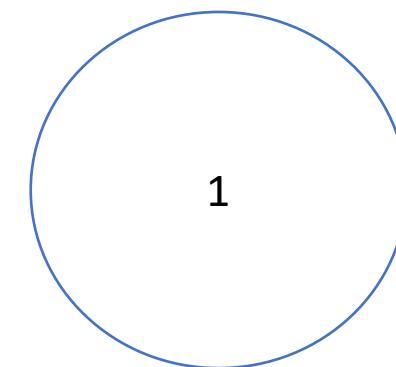
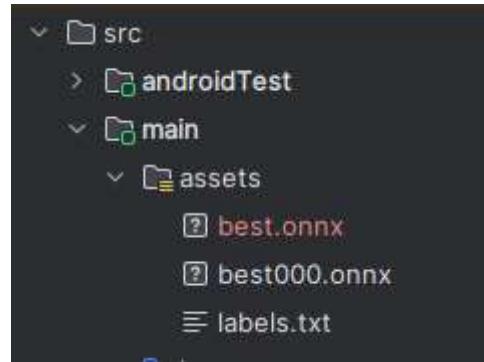


После(обучение 3 раз с параметрами и +новый датасет собранный)

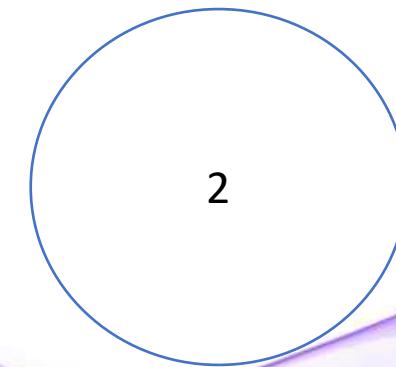
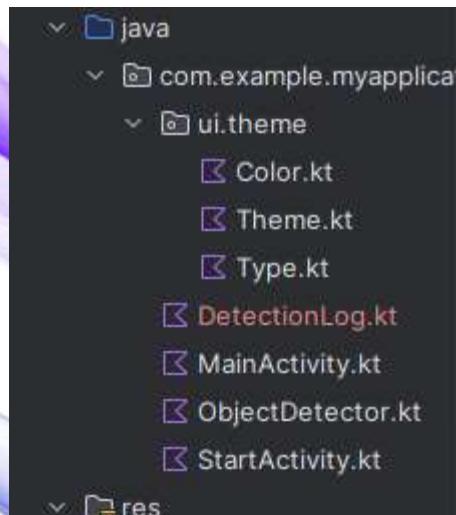
Speed: 2.7ms preprocess, 56.0ms inference, 2.0ms postprocess per image at shape
(1, 3, 480, 640)

Результат детекции объектов

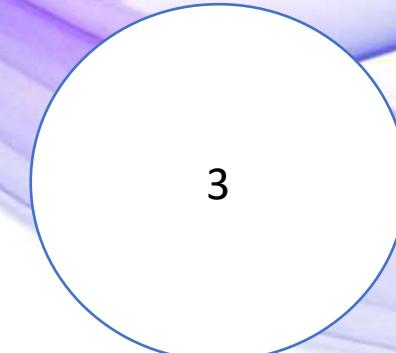




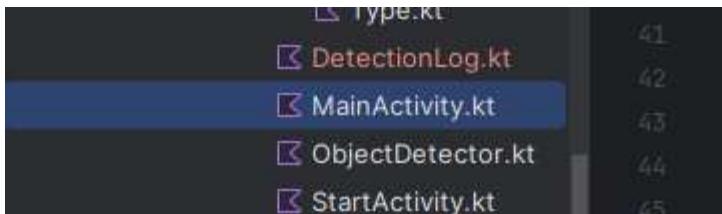
Модель машинного обучения в формате best.onnx
Классы



Основные элементы программы
DetectionLog.kt Запись логов 10 до фото и 10 после
MainActivity.kt Основной код программы
ObjectDetector.kt Детекция объектов
StartActivity.kt Стартовая программа(окно запуска)



AndroidManifest.xml с основными параметрами sync



Ключевые функции:

Захват изображения с камеры (ImageCapture)

Анализ кадров в реальном времени (ImageAnalysis)

Преобразование ImageProxy в Bitmap для обработки

Масштабирование и нормализация изображения для модели (bitmapToFloatArray)

Отрисовка bounding boxes и меток классов

Сохранение фото с обнаруженными объектами (saveImageWithOverlay)

MA My Application0404 Version control

Pixel 6a app

MainActivity.kt Pixel 6a API 36

Project

MyApplication0404 [My Application0404] C:\Users\PC\Android Studio Projects\My Application0404

- .gradle
- .idea
- .kotlin
- app
 - build
 - src
 - androidTest
 - main
 - assets
 - best.onnx
 - labels.txt
 - java
 - res
 - drawable
 - ic_google.png
 - ic_launcher_background.xml
 - ic_launcher_foreground.xml
 - ic_logo.jpg
 - splash_image.jpg
 - mipmap-anydpi-v26
 - mipmap-hdpi
 - mipmap-mdpi
 - mipmap-xhdpi
 - mipmap-xxhdpi
 - mipmap-xxxhdpi
 - values
 - xml
 - AndroidManifest.xml
 - test [unitTest]
 - .gitignore
 - build.gradle.kts
 - proguard-rules.pro
 - gradle
 - .gitignore
 - build.gradle.kts
 - gradle.properties
 - gradlew

51 f 6 2 7 7 51
56 var resultText
57 var detectionResults
58 var imageWidth
59 var imageHeight
60
61 AndroidView(61
62 factory = - 62
63 val pre 63
64 lau 64
65) 65
66) 66
67 } 67
68 } 68
69
70 cameraF 69
71 val 71
72 val 72
73) 73
74 } 74
75
76 val 75
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96

10:18 Fri, Apr 4

Play Store Gmail Photos YouTube

Phone Chat Chrome

G

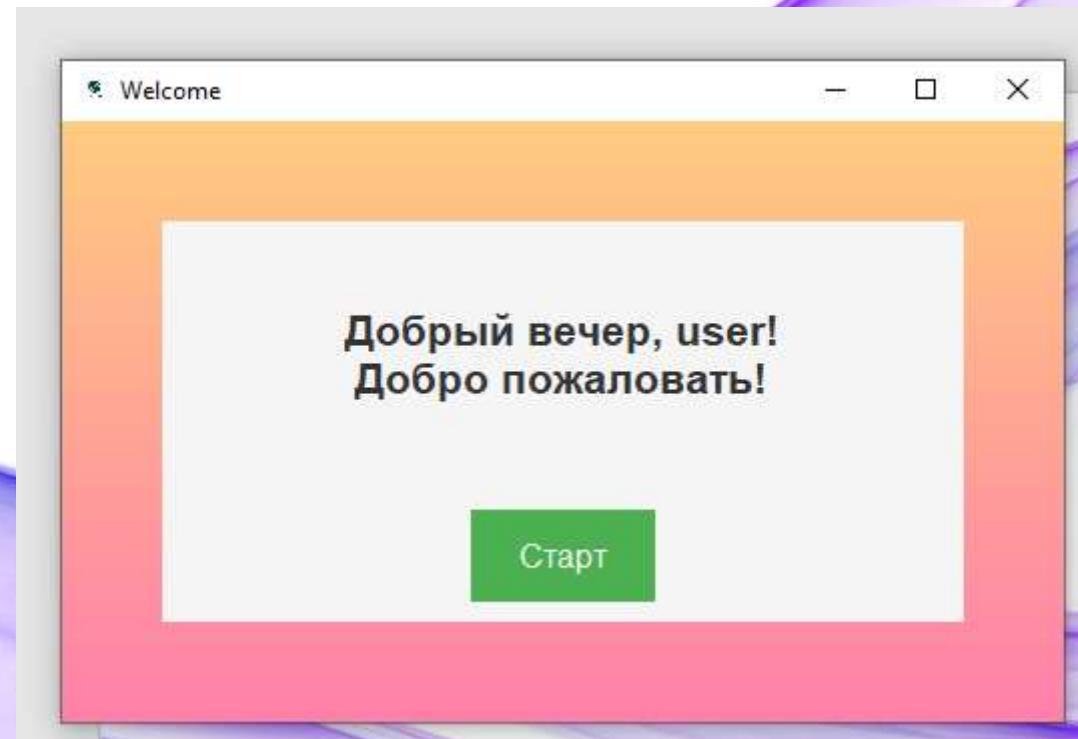
63.1 LF UTF-8 4 spaces



- 1) Модель машинного обучения обучена
- 2) Модель определяет предметы
- 3) Реализована функция захвата результата(снимок экрана сохраняется в галерею с оверлеем идентификации)
- 4) Есть стартовая функция и главная функция
- 5) Данные сохраняются в сетевую папку, автоматически соединяясь с ПК-приложением
- 6) ПК-приложение получает изображения и позволяет работу с ними

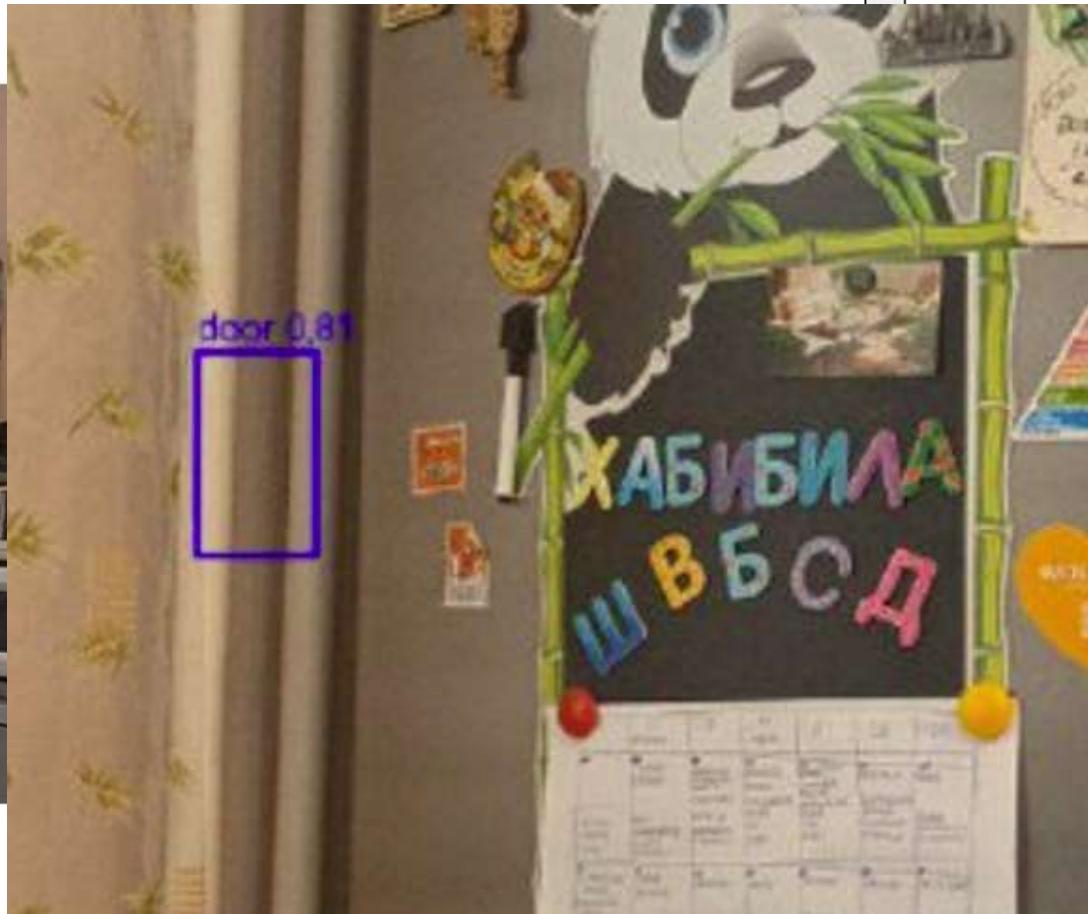


Desktop-приложение



Desktop-приложение

Логи



быть

Изображение 1 из 1

Предыдущее

Следующее

Применение

Технологии распознавания в помощь человеку

The diagram features three horizontal rectangular boxes of different colors: yellow, green, and blue. Each box contains white text describing an application of object detection technology. Above each box is a white circle with a colored outline (yellow, green, or blue) connected by a thin blue line to the top edge of its respective box. The background consists of several overlapping, translucent purple and blue curved bands.

Ассистивные технологии

Использование для детекции объектов
быстroredвижущихся и систем реального времени

ИСТОЧНИКИ:

- [Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library - Adrian Kaehler, Gary Bradski - Google Книги](#)
- [Pressure-Driven Micro-Casting for Electrode Fabrication and Its Applications in Wear Grain Detections - PMC \(nih.gov\)](#)
- [Frontiers | The Argus-II Retinal Prosthesis Implantation; From the Global to Local Successful Experience \(frontiersin.org\)](#)
- [Polymers | Free Full-Text | Medical-Grade Polyamide 12 Nanocomposite Materials for Enhanced Mechanical and Antibacterial Performance in 3D Printing Applications \(mdpi.com\)](#)
- [diamond and bromine - Search Results - PubMed \(nih.gov\)](#)
- [Machine Learning in Computer Vision - ScienceDirect](#)
- [121_B01656life1003_804_813-libre.pdf \(d1wqtxts1xzle7.cloudfront.net\)](#)
- [A Survey on Performance Metrics for Object-Detection Algorithms | IEEE Conference Publication | IEEE Xplore](#)
- [Image-Adaptive YOLO for Object Detection in Adverse Weather Conditions | Proceedings of the AAAI Conference on Artificial Intelligence](#)
- [A review of object detection based on deep learning | Multimedia Tools and Applications \(springer.com\)](#)
- [Target Detection, Tracking and Avoidance System for Low-cost UAVs using AI-Based Approaches | IEEE Conference Publication | IEEE Xplore](#)
- [Deep Learning for Image Processing Applications - Google Книги](#)
- [Comparative analysis of image classification algorithms based on traditional machine learning and deep learning - ScienceDirect](#)
- [Machine \(Deep\) Learning Methods for Image Processing and Radiomics | IEEE Journals & Magazine | IEEE Xplore](#)
- [Machine learning and deep learning | Electronic Markets \(springer.com\)](#)
- [Open Access proceedings Journal of Physics: Conference series \(iop.org\)](#)
- [IEEE Xplore Full-Text PDF:](#)
- [International Journal of Innovative Technology and Exploring Engineering \(IJITEE\) \(researchgate.net\)](#)
- [Introduction to Algorithms, Third Edition \(edutechlearners.com\)](#)
- [Microelectrode Array | Axion Biosystems](#)
- [Pressure-Driven Micro-Casting for Electrode Fabrication and Its Applications in Wear Grain Detections - PMC \(nih.gov\)](#)
- [Медицинское литье под давлением: процесс, материалы и применение \(proleantech.com\)](#)
- [Установка по нанесению покрытий из металлов на объемные изделия \(epos-nsk.ru\)](#)
- [Искусственные алмазы: альтернатива настоящим или суррогат? | Добывающая промышленность \(deprom.online\)](#)
- [Трубчатая Печь Cvd С Разделенной Камерой И Вакуумной Станцией Cvd Машины - Kintek Solution \(kindle-tech.com\)](#)
- [▷ Решения для литья под давлением электронных деталей - ENGEL \(engelglobal.com\)](#)
- [Микроинжекция, микро литье компонентов, высокоскоростное литье под давлением. | Производство автомобильных деталей высокой сложности с применением точных техник литья под давлением | FORESHOT](#)
- [Литье под давлением микрокомпонентов - новаторство в производстве крошечных деталей \(toplast.ru\)](#)
- [Machine Learning in Computer Vision - ScienceDirect](#)
- <https://www.drugdeliverybusiness.com/second-sight-merges-nano-precision-rebrands-vivani/#:~:text=Second%20Sight%20Medical%20merges%20with%20Nano%20Precision%20Medical%2C%20rebrands%20as%20Vivani%20Medical&text=Second%20Sight%20Medical%20>
- <https://www.universiteitleiden.nl/en/news/2020/10/3d-printed-microboat>
- <https://www.materialstoday.com/materials-down-under-dr-kate-fox/>
- <https://pubmed.ncbi.nlm.nih.gov/33435443/>
- <https://paar.ru/usage/articles/mekhanicheskie-svoystva-plenok-iz-ultrananolistallicheskogo-almaza-uncd-s-vliyaniem-plotnosti-nukle/>
- <https://www.mdpi.com/2073-4360/14/3/440>
- <https://link.springer.com/article/10.1007/s13204-023-02966-4>
- <https://www.thingiverse.com/thing:5675700>



Спасибо за внимание!

