

# Управление роботом с использованием нейронных сетей (практика)

**Пелевин Владимир**

к.пед.н., доцент, УрФУ

инженер, ООО Микроэлектроника и Робототехника

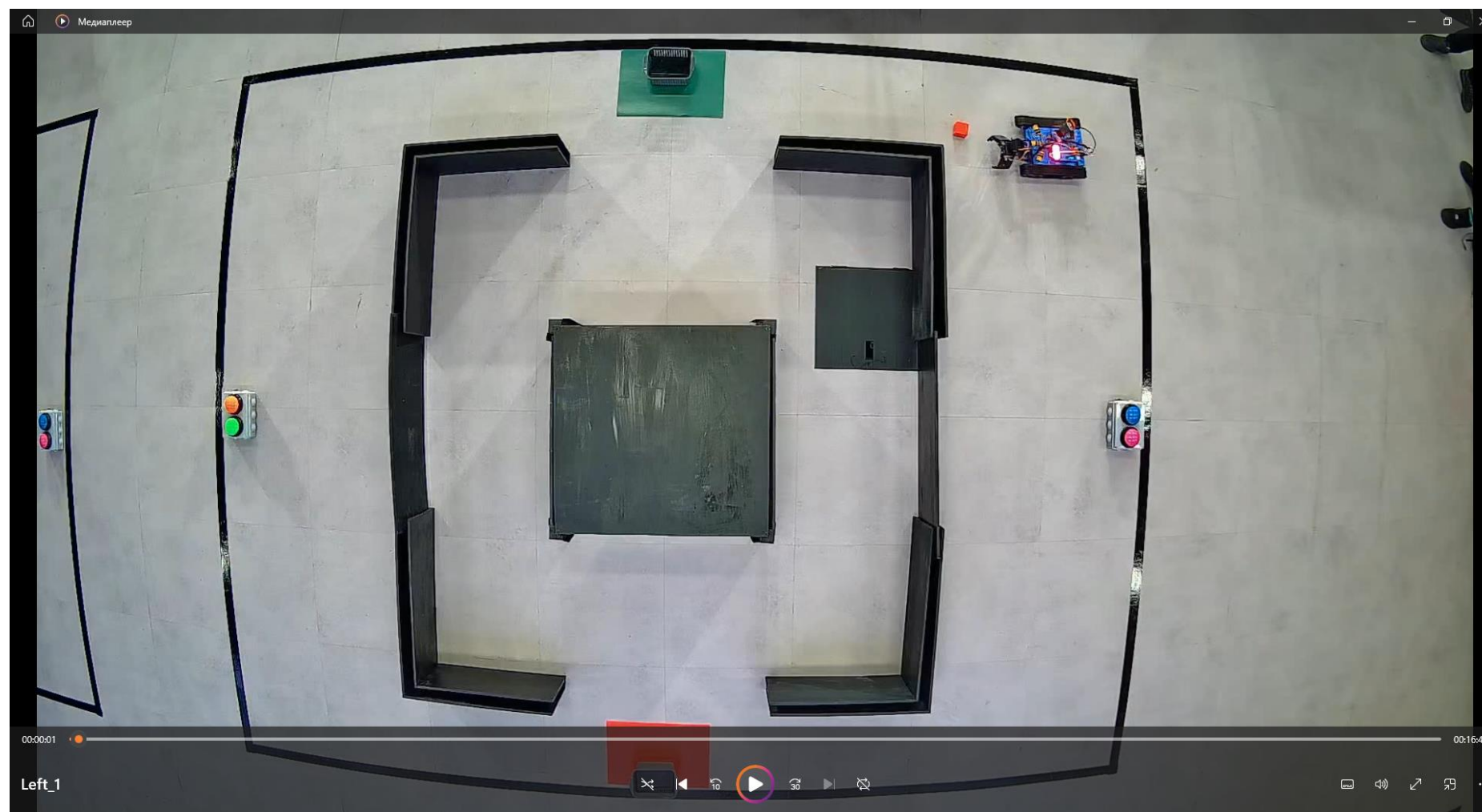
**Клюев Данил**

ассистент, УрФУ

# План работ на 1 занятие

1. Собрать данные для модели
2. Разметить данные через Label Studio

## Шаг 1 Сбор данных



```
def extract_frames(video_path, output_folder):
    if not os.path.exists(output_folder):
        os.makedirs(output_folder)

    video = cv2.VideoCapture(video_path)
    success, prev_image = video.read()
    count = 0

    while success:
        # Подготовка к сохранению кадра
        frame_filename = os.path.join(output_folder, f"frame_{count:04d}.jpg")
        cv2.imwrite(frame_filename, prev_image)

        # Считывание следующего кадра
        success, curr_image = video.read()

        # Проверяем, если удалось считать следующий кадр
        if not success:
            break

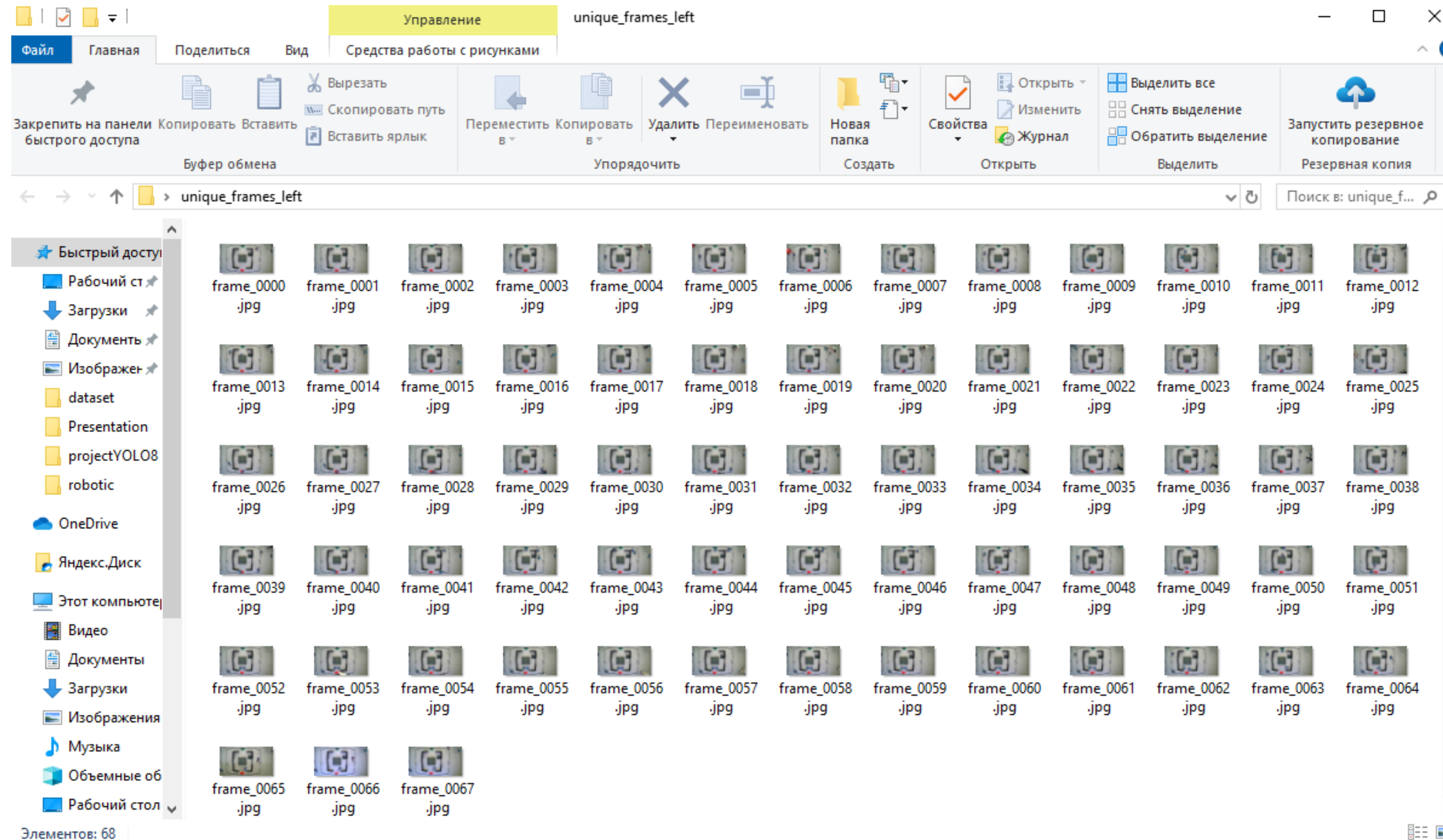
        # Преобразуем изображения в градации серого для сравнения
        prev_gray = cv2.cvtColor(prev_image, cv2.COLOR_BGR2GRAY)
        curr_gray = cv2.cvtColor(curr_image, cv2.COLOR_BGR2GRAY)

        # Вычисляем сходство
        similarity = ssim(prev_gray, curr_gray)

        # Если кадры не похожи, сохраняем текущий кадр
        if similarity < 0.9:
            prev_image = curr_image
            count += 1

    video.release()
    print(f"Extracted and saved {count} unique frames to '{output_folder}'.")

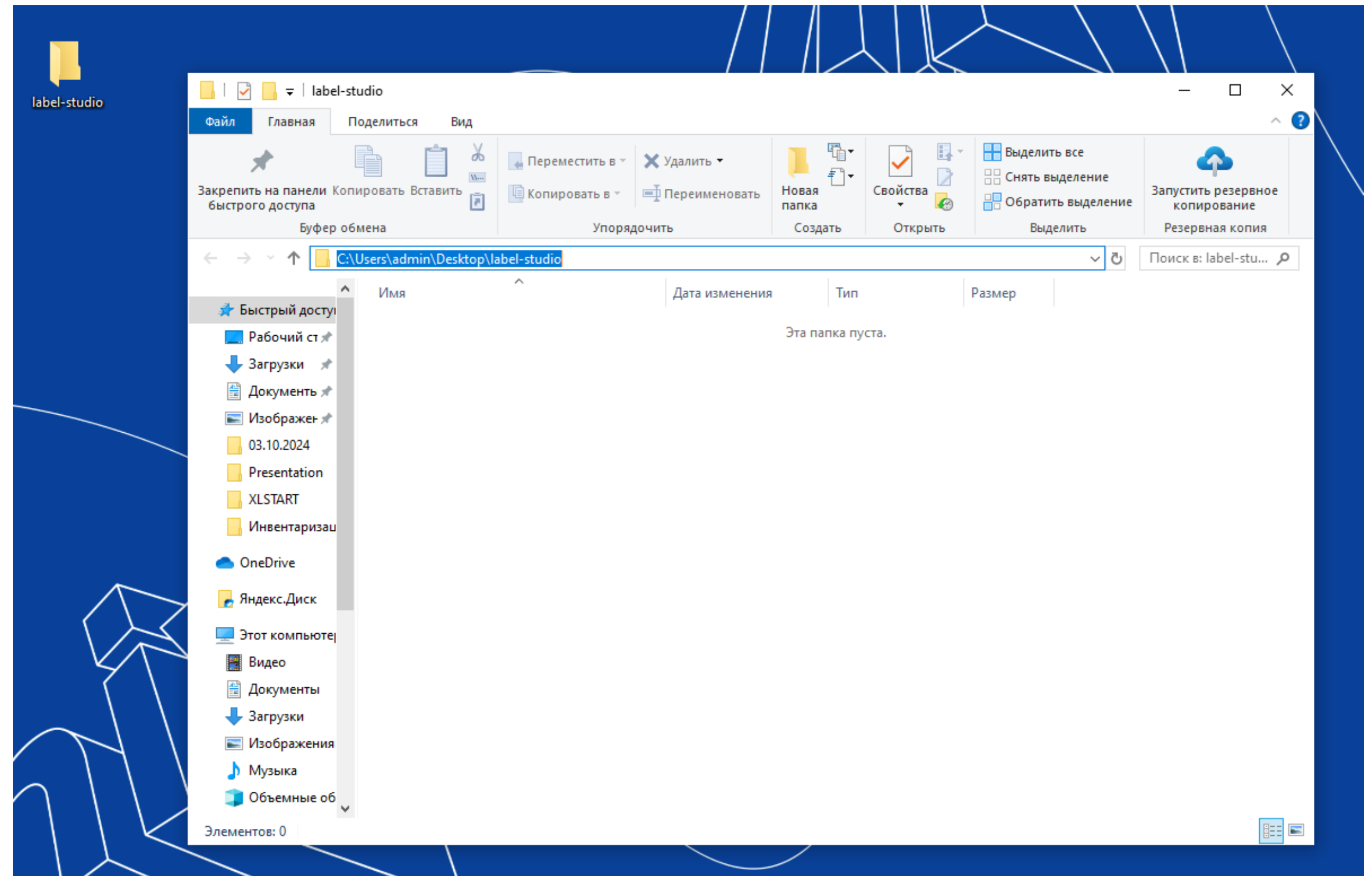
# Пример использования
video_file = 'C:/Users/UrFU/Desktop/projectYOL08/Left_1.avi' # Укажите путь к вашему видеофайлу
output_dir = 'unique_frames' # Название папки для уникальных кадров
extract_frames(video_file, output_dir)
```





## Шаг 2 Разметка изображений

- Создаем папку, где разместим label-studio.
- Копируем путь до папки, который нам пригодится далее



# Студкемп по робототехнике и ИИ

1. `cd` (путь до папки `label-studio`)

2. `py -m venv env`

```
Командная строка
Microsoft Windows [Version 10.0.19045.5011]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\admin>cd C:\Users\admin\Desktop\label-studio

C:\Users\admin\Desktop\label-studio>
```

label-studio

Файл Главная Поделиться Вид

Закрепить на панели быстрого доступа Копировать Вставить Буфер обмена Переместить в Удалить Копировать в Переименовать Упорядочить Новая папка Создать Свойства Открыть Выделить все Снять выделение Обратить выделение Выделить Запустить резервное копирование Резервная копия

label-studio

Имя	Дата изменения	Тип	Размер
env	12.10.2024 10:16	Папка с файлами	

```
Командная строка
Microsoft Windows [Version 10.0.19045.5011]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\admin>cd C:\Users\admin\Desktop\label-studio

C:\Users\admin\Desktop\label-studio>py -m venv env

C:\Users\admin\Desktop\label-studio>
```

1. `env\Scripts\activate.bat`
2. `pip install -U label-studio`
3. `(start) label-studio`

```
C:\Users\admin\Desktop\label-studio>env\Scripts\activate.bat  
(env) C:\Users\admin\Desktop\label-studio>
```

```
C:\Users\admin\Desktop\label-studio>env\Scripts\activate.bat  
(env) C:\Users\admin\Desktop\label-studio>pip install -U label-studio
```

```
Командная строка  
rtlib-metadata, httpcore, googleapis-common-protos, google-resumable-media, Django, deprecated, click, cffi, bleach, anyio  
, rq, requests-mock, pydantic, pandas, opentelemetry-api, nltk, httpx, grpcio-status, google-cloud-audit-log, google-auth,  
drf-generators, djangorestframework, django-user-agents, django-storages, django-model-utils, django-filter, django-debug  
-toolbar, django-csp, django-cors-headers, django-annoying, cryptography, botocore, azure-core, s3transfer, openai, label-  
studio-sdk, humansignal-drf-yasg, grpc-google-iam-v1, google-api-core, django-rq, azure-storage-blob, google-cloud-core, b  
oto3, google-cloud-storage, google-cloud-appengine-logging, google-cloud-logging, label-studio  
Successfully installed Django-3.2.25 Pillow-10.4.0 annotated-types-0.7.0 anyio-4.6.0 appdirs-1.4.4 asgiref-3.8.1 attr-0.3.  
1 attrs-24.2.0 azure-core-1.31.0 azure-storage-blob-12.23.1 bleach-5.0.1 boto-2.49.0 boto3-1.35.39 botocore-1.35.39 boxing  
-0.1.4 cachetools-5.5.0 certifi-2024.8.30 cffi-1.17.1 charset-normalizer-3.4.0 click-8.1.7 colorama-0.4.6 cryptography-43.  
0.1 defusedxml-0.7.1 deprecated-1.2.14 distro-1.9.0 django-annoying-0.10.6 django-cors-headers-3.6.0 django-csp-3.7 django  
-debug-toolbar-3.2.1 django-environ-0.10.0 django-extensions-3.1.0 django-filter-2.4.0 django-model-utils-4.1.1 django-ran  
ged-fileresponse-0.1.2 django-rq-2.5.1 django-storages-1.12.3 django-user-agents-0.4.0 djangorestframework-3.13.1 drf-dyna  
mic-fields-0.3.0 drf-flex-fields-0.9.5 drf-generators-0.3.0 expiringdict-1.2.2 google-api-core-2.21.0 google-auth-2.35.0 g  
oogle-cloud-appengine-logging-1.4.5 google-cloud-audit-log-0.3.0 google-cloud-core-2.4.1 google-cloud-logging-3.11.2 googl  
e-cloud-storage-2.18.2 google-crc32c-1.6.0 google-resumable-media-2.7.2 googleapis-common-protos-1.65.0 grpc-google-iam-v1  
-0.13.1 grpcio-1.66.2 grpcio-status-1.66.2 h11-0.14.0 htmlmin-0.1.12 httpcore-1.0.6 httpx-0.27.2 humansignal-drf-yasg-1.21  
.9 idna-3.10 ijson-3.3.0 importlib-metadata-8.4.0 inflection-0.5.1 isodate-0.7.2 jiter-0.6.1 jmespath-1.0.1 joblib-1.4.2 j  
sonschema-3.2.0 label-studio-1.13.1 label-studio-sdk-1.0.5 launchdarkly-server-sdk-8.2.1 lockfile-0.12.2 lxml-5.3.0 nltk-3  
.9.1 numpy-1.26.4 openai-1.51.2 opentelemetry-api-1.27.0 ordered-set-4.0.2 packaging-24.1 pandas-2.2.3 proto-plus-1.24.0 p  
rotobuf-5.28.2 pycopg2-binary-2.9.9 pyRFC3339-1.1 pyasn1-0.6.1 pyasn1-modules-0.4.1 pycparser-2.22 pydantic-2.9.2 pydanti  
c-core-2.23.4 pyparsing-3.1.2 pyrsistent-0.20.0 python-dateutil-2.9.0.post0 python-json-logger-2.0.4 pytz-2022.7.1 pyyaml-6.0.2 redis-3.5.  
3 regex-2024.9.11 requests-2.32.3 requests-mock-1.12.1 rq-1.10.1 rsa-4.9 rules-2.2 s3transfer-0.10.3 semver-3.0.2 sentry-s  
dk-2.16.0 setuptools-75.1.0 six-1.16.0 sniffio-1.3.1 sqlparse-0.5.1 tqdm-4.66.5 typing_extensions-4.12.2 tzdata-2024.2 ua  
parser-0.18.0 ujson-5.10.0 uritemplate-4.1.1 urllib3-1.26.20 user-agents-2.2.0 webencodings-0.5.1 wheel-0.40.0 wrapt-1.16.  
0 xmljson-0.2.1 zipp-3.20.2  
  
[notice] A new release of pip is available: 24.0 -> 24.2  
[notice] To update, run: python.exe -m pip install --upgrade pip  
(env) C:\Users\admin\Desktop\label-studio>label-studio_
```

# Студкемп по робототехнике и ИИ

Яндекс

УрФУ  
Институт  
радиоэлектроники  
и информационных  
технологий – РТФ



## Label Studio

A full-fledged open source solution for data labeling

### Did you know?

Label Studio has dozens of pre-built templates for all data types you can use to configure your labeling UI, from image classification to sentiment analysis to supervised LLM fine-tuning. [See all templates](#)



Brought to you by  
 HumanSignal

### Log in

Email Address

Password

☒ Keep me logged in this browser


Log in

Don't have an account? [Sign up](#)

### Sign Up

Email Address

Password

☒ Get the latest news from Heidi 

Create Account

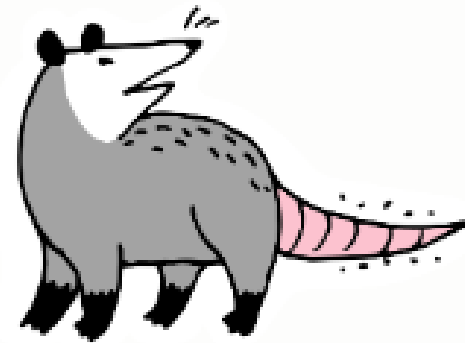
Already have an account? [Log in](#)



# Студкемп по робототехнике и ИИ

Яндекс

УрФУ  
Институт  
радиоэлектроники  
и информационных  
технологий – РТФ



## Heidi doesn't see any projects here!

Create one and start labeling your data.

Create Project

Project Name

Data Import

Labeling Setup

Project Name

Robot\_Competition

Description

Optional description of your project

Workspace Enterprise

Select an option

Simplify project management by organizing projects into workspaces. [Learn more](#)

Unlock faster access provisioning

Streamline assigning staff to multiple projects by assigning them to workspaces in Label Studio Enterprise. [Learn more](#)



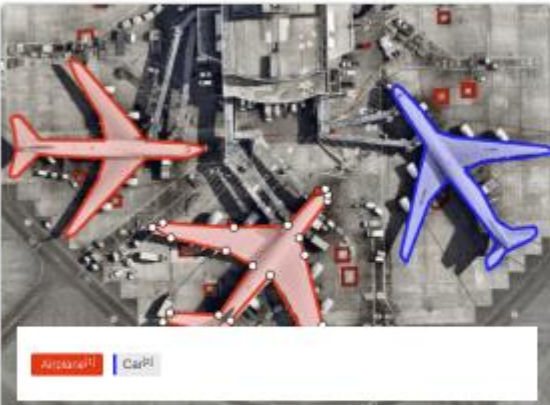
Project Name

Data Import

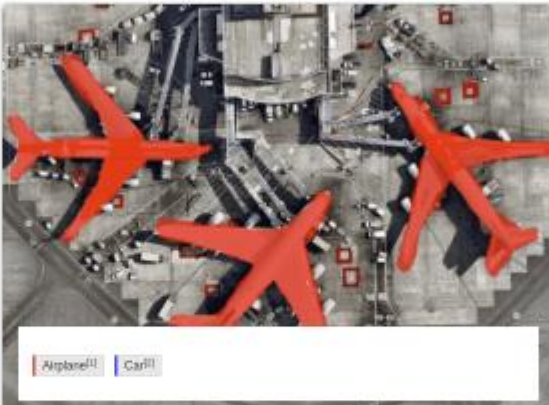
Labeling Setup

Delete


Save




Semantic Segmentation with Polygons



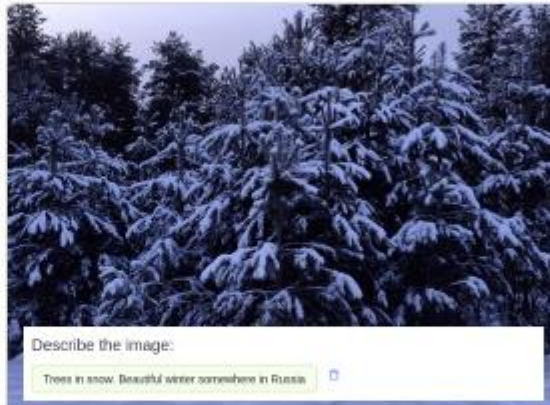
Semantic Segmentation with Masks



Object Detection with Bounding Boxes



Keypoint Labeling



Describe the image:  
Trees in snow. Beautiful winter somewhere in Russia

Image Captioning



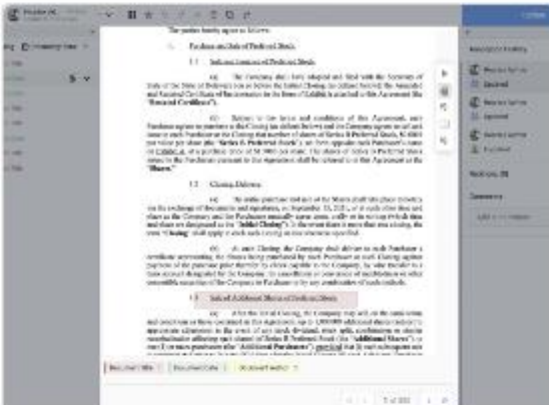



Image Classification




Inventory Tracking




Multi-page document annotation



Optical Character Recognition



Visual Genome



Please answer these questions:  
Q1: Are there an equal number of large  
A1: 1



Create Project

Project Name

Data Import

Labeling Setup

Delete

Save

## Labeling Interface

Browse Templates

Code

Visual

Configure data

Use image from < set manually > \$image

Add label names

Use new line as a separator to add multiple labels

Add

Labels (2)

- Airplane
- Car

Configure settings

Width of region borders 1

- ☐ Allow image zoom (ctrl+wheel)
- ☐ Show controls to zoom in and out
- ☐ Show controls to rotate image

Display labels: bottom

- ☐ Add filter for long list of labels

UI Preview



Airplane 1 Car 2

Regions

History

Relations

Info

Manual

By Time

Labeling Interface

Browse Templates

Code

Visual

Configure data

Use image from

<set manually>

\$image

Add label names

Use new line as a separator to add multiple labels

Add

Labels (5)

Robot

Buttons (G/O)

Buttons (B/P)

Cubic

Basket

Import

Export

List

Grid



## Import Data

Cancel

Import

Dataset URL

Add URL

or



Upload Files

Drag & drop files here  
or click to browse



Text	txt
Audio	wav, mp3, flac, m4a, ogg
Video	mpeg4/H.264 webp, webm*
Images	jpg, jpeg, png, gif, bmp, svg, webp
HTML	html, htm, xml
Time Series	csv, tsv
Common Formats	csv, tsv, txt, json

\* – Support depends on the browser  
\* – Use [Cloud Storages](#) if you want to import a large number of files

Label Studio

Projects / Robot\_Competition

Default

+

Actions

Columns

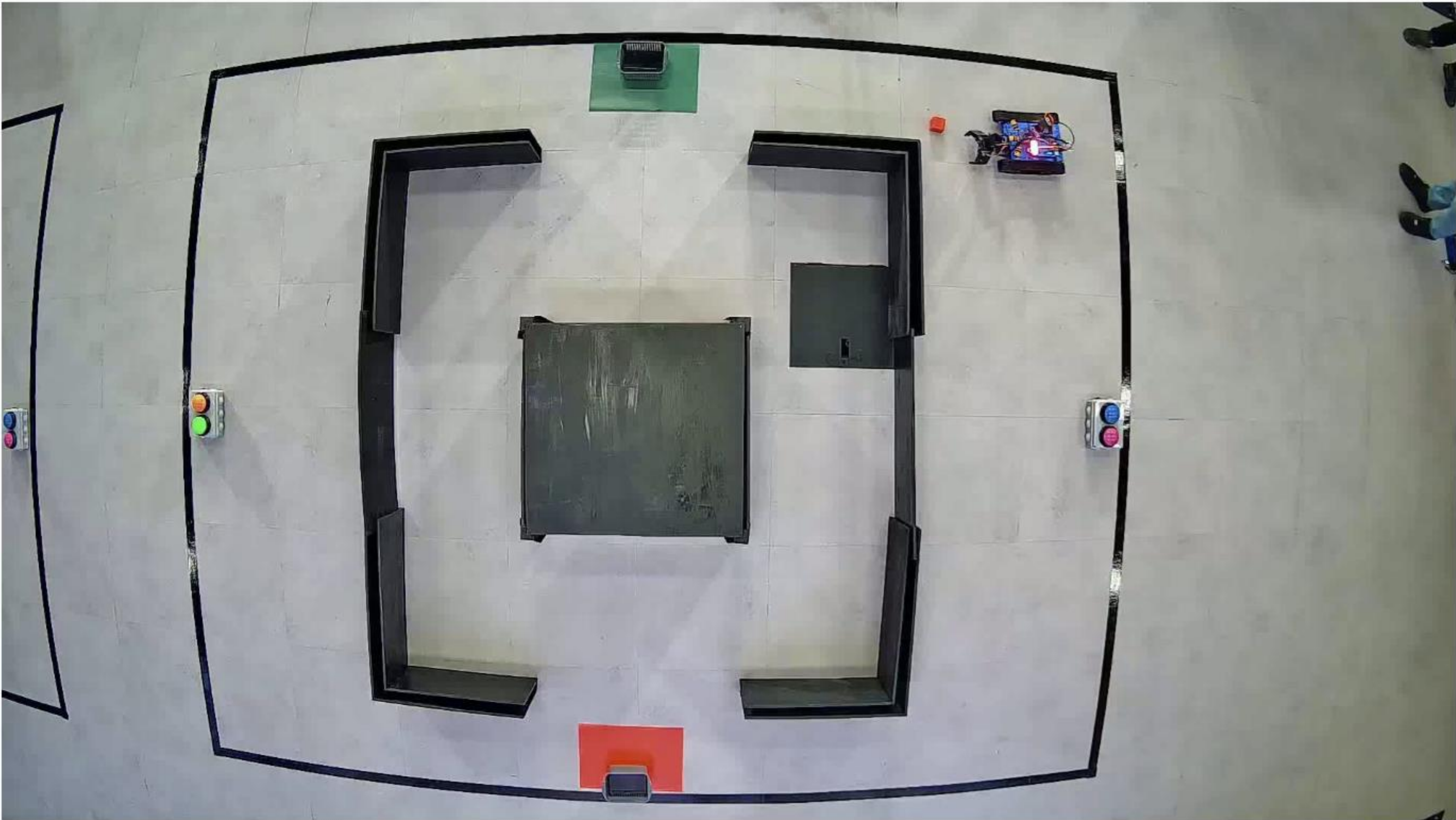
Filters

Order

not set

Label All Tasks

<input type="checkbox"/>	ID	Completed				Annotated by	image	img
<input type="checkbox"/>	48		0	0	0			<a href="#">↗</a>
<input type="checkbox"/>	49		0	0	0			<a href="#">↗</a>
<input type="checkbox"/>	50		0	0	0			<a href="#">↗</a>
<input type="checkbox"/>	51		0	0	0			<a href="#">↗</a>
<input type="checkbox"/>	52		0	0	0			<a href="#">↗</a>
<input type="checkbox"/>	53		0	0	0			<a href="#">↗</a>
<input type="checkbox"/>	54		0	0	0			<a href="#">↗</a>
<input type="checkbox"/>	55		0	0	0			<a href="#">↗</a>
<input type="checkbox"/>	56		0	0	0			<a href="#">↗</a>
<input type="checkbox"/>	57		0	0	0			<a href="#">↗</a>
<input type="checkbox"/>	58		0	0	0			<a href="#">↗</a>



Robot 1 Buttons (G/O) 2 Buttons (B/P) 3 Cubic 4 Basket 5

InfoHistory

Selection Details

➡

👤

⊕

🎲

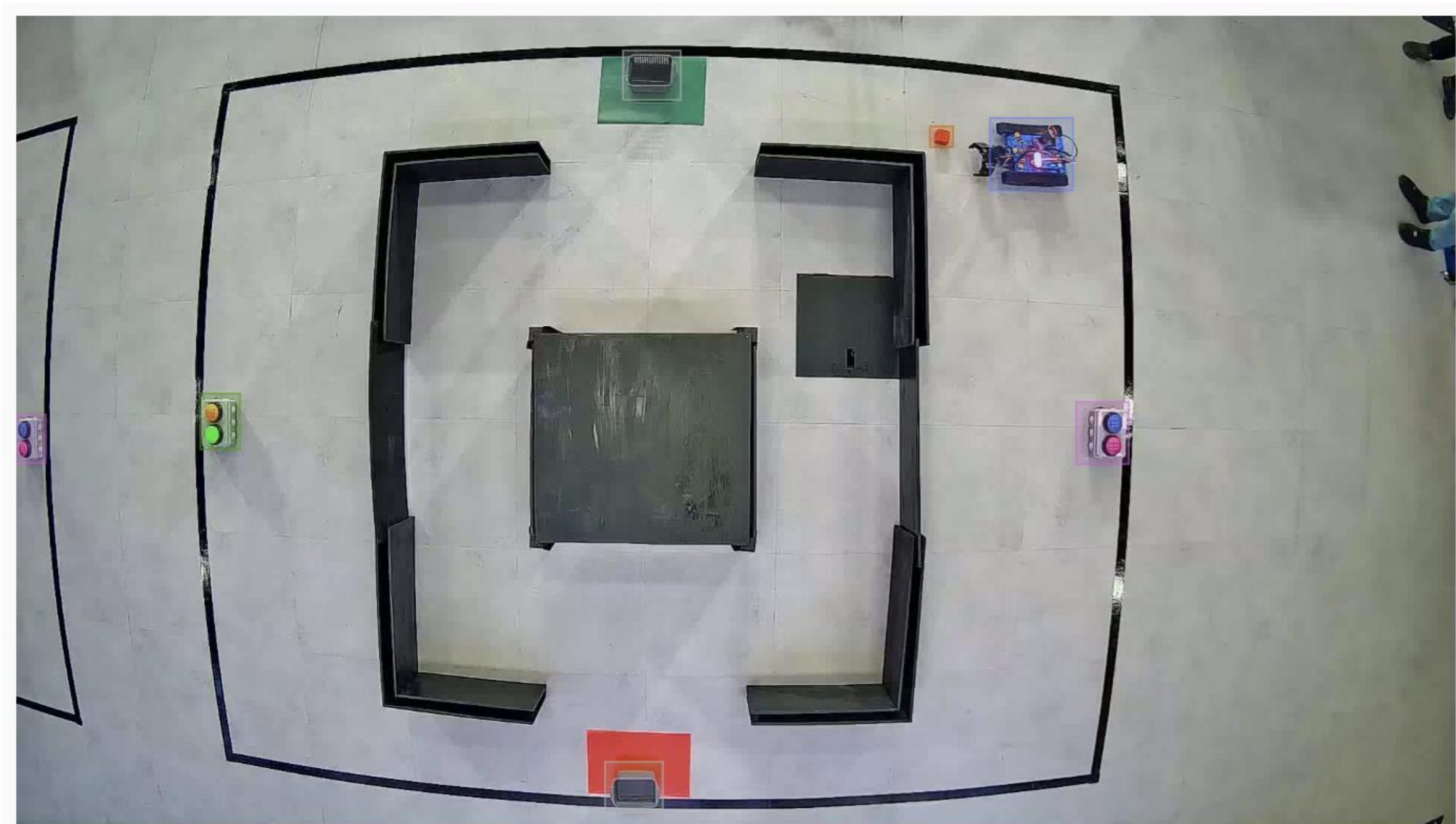
🔍

RegionsRelations

☰ Manual📅 By Time⬆️⬆️📶

Regions not added





Info

History

Selection Details

Regions

Relations

Manual

By Time

1 Robot

2 Buttons (G/O)

3 Buttons (B/P)

4 Buttons (B/P)

5 Cubic

6 Basket

7 Basket

Robot 1

Buttons (G/O) 2

Buttons (B/P) 3

Cubic 4

Basket 5

↶

↷

✕

⌵

Skip

Submit



# Задание

1. **Собрать данные, используя камеры или видеозаписи с этих камер**
2. **Разметить подготовленные данные**

# План работ на 2 занятие

1. Подготовить модель через YOLOv8 на основе своего датасета
2. В консоли своего IDE вывести x и y центров bounding box

# Шаг 1 Подготовка модели

Tasks: 68 / 68   Annotations: 68   Predictions: 0

Import

Export

List

Grid

Export data

You can export dataset in one of the following formats:

☐ JSON

List of items in raw JSON format stored in one JSON file. Use to export both the data and the annotations for a dataset. It's Label Studio Common Format

☐ JSON-MIN

List of items where only "from\_name", "to\_name" values from the raw JSON format are exported. Use to export only the annotations for a dataset.

☐ CSV

Results are stored as comma-separated values with the column names specified by the values of the "from\_name" and "to\_name" fields.

☐ TSV

Results are stored in tab-separated tabular file with column names specified by "from\_name" "to\_name" values

☐ COCO

Popular machine learning format used by the COCO dataset for object detection and image segmentation tasks with polygons and rectangles. 

image segmentation

object detection

☐ Pascal VOC XML

Popular XML format used for object detection and polygon image segmentation tasks. 

image segmentation

object detection

☒ YOLO

Popular TXT format is created for each image file. Each txt file contains annotations for the corresponding image file, that is object class, object coordinates, height & width. 

image segmentation

object detection

☐ YOLOv8 OBB

Popular TXT format is created for each image file. Each txt file contains annotations for the corresponding image file. The YOLO OBB format designates bounding boxes by their four corner points with coordinates normalized between 0 and 1, so it is possible to export rotated objects. 

image segmentation

object detection

☐ CONLL2003

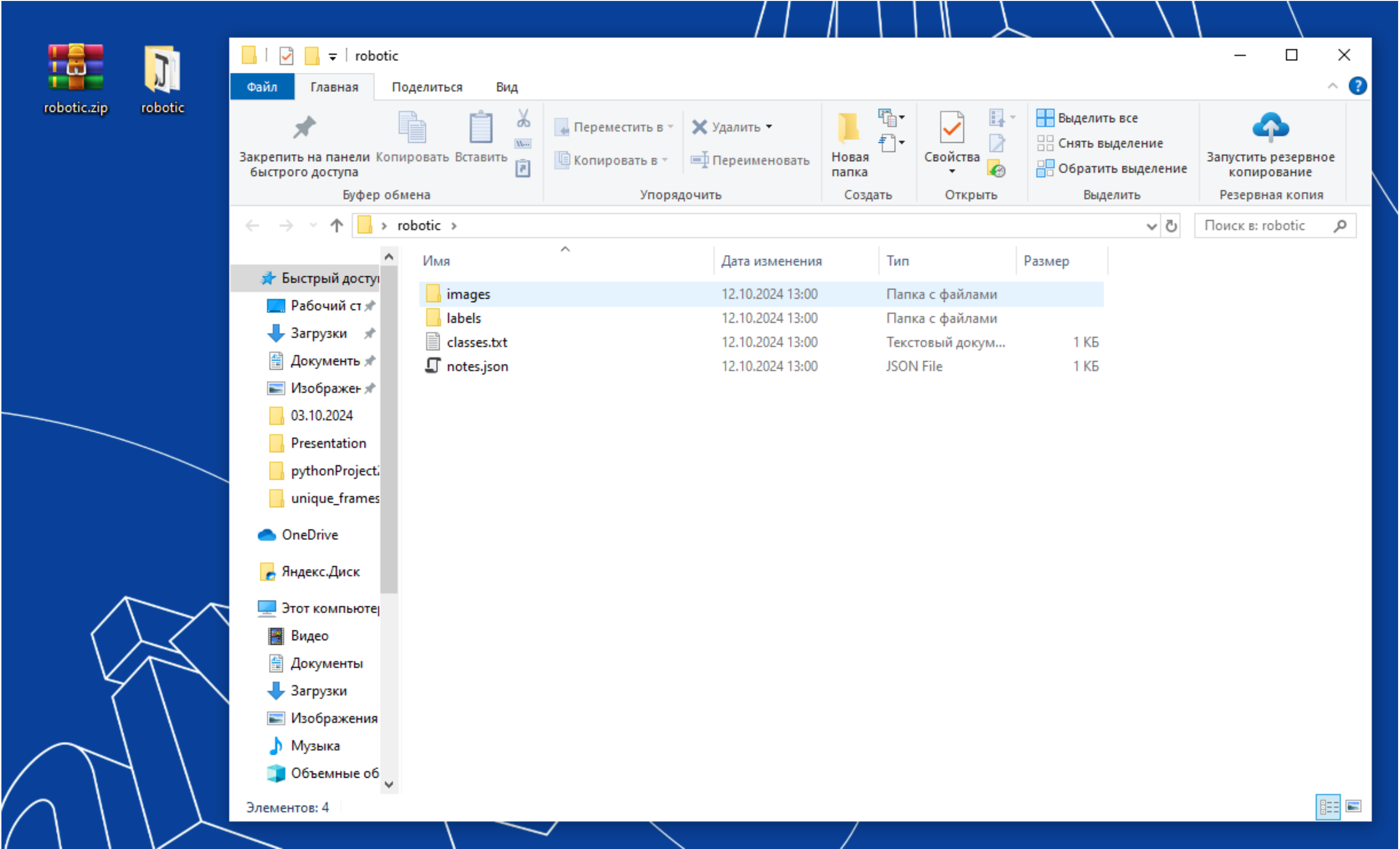
Popular format used for the CoNLL-2003 named entity recognition challenge. 

sequence labeling

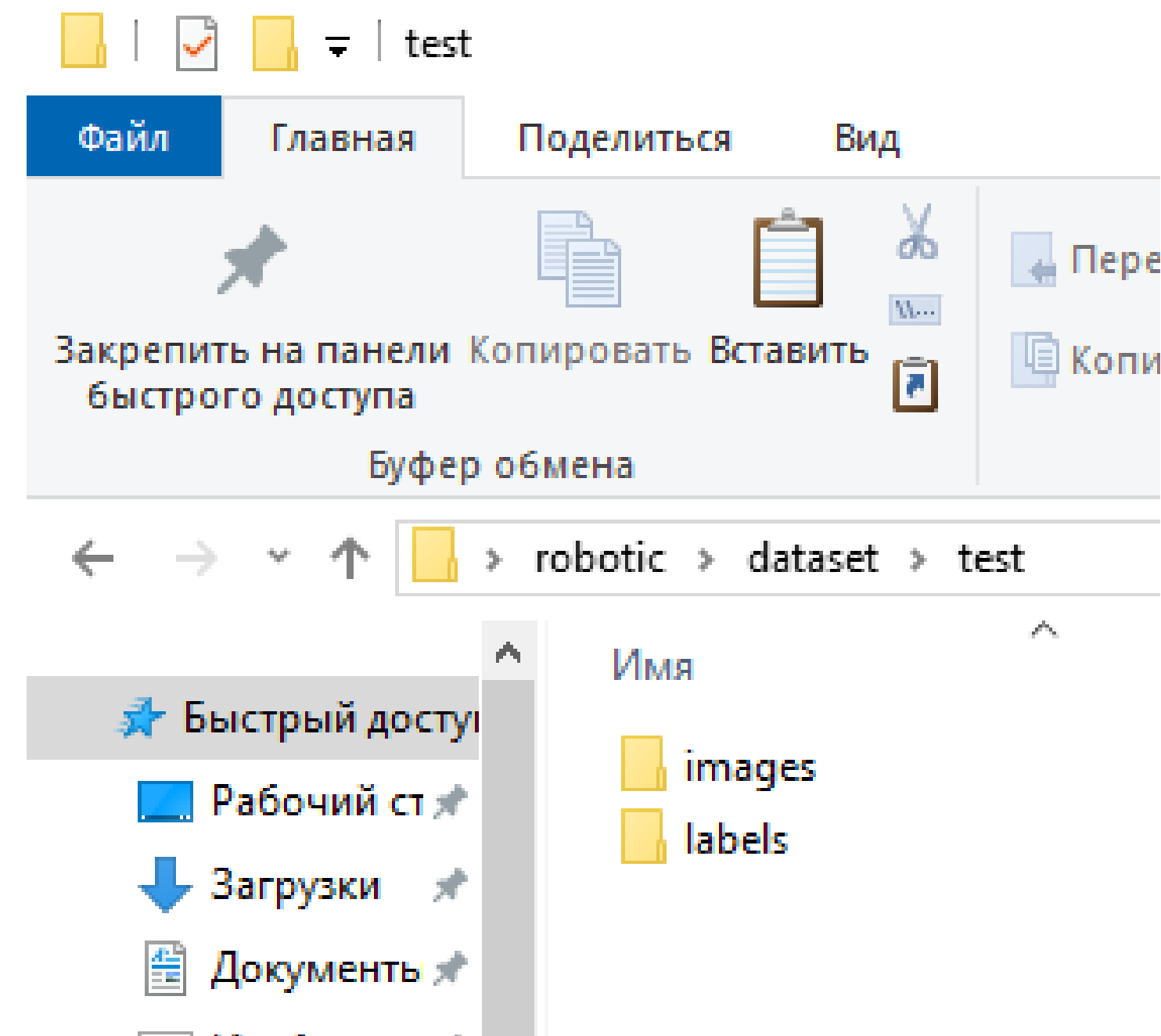
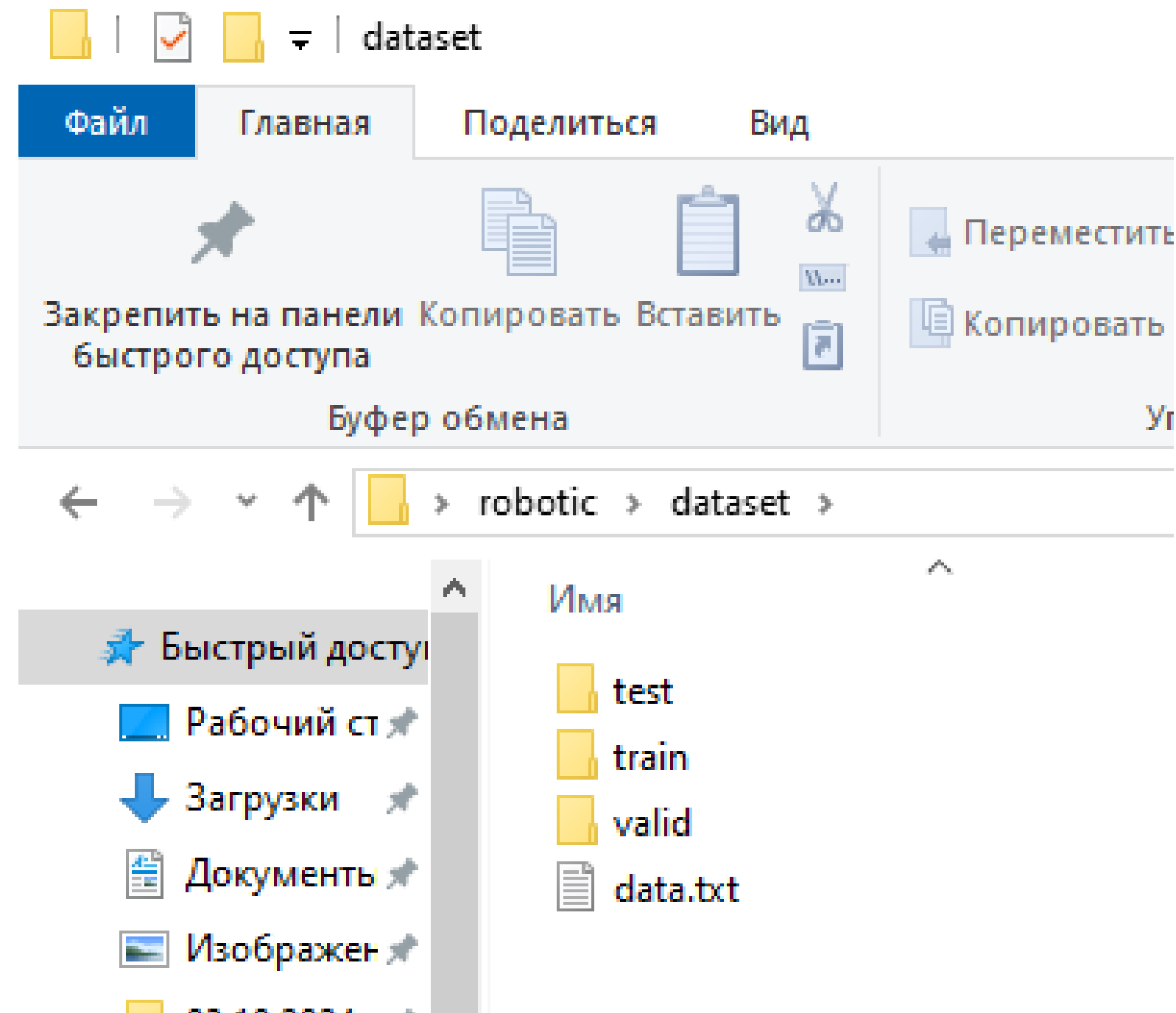
text tagging

named entity recognition

Export







train – 80%|test – 10%|valid – 10%

classes.txt – Блокнот

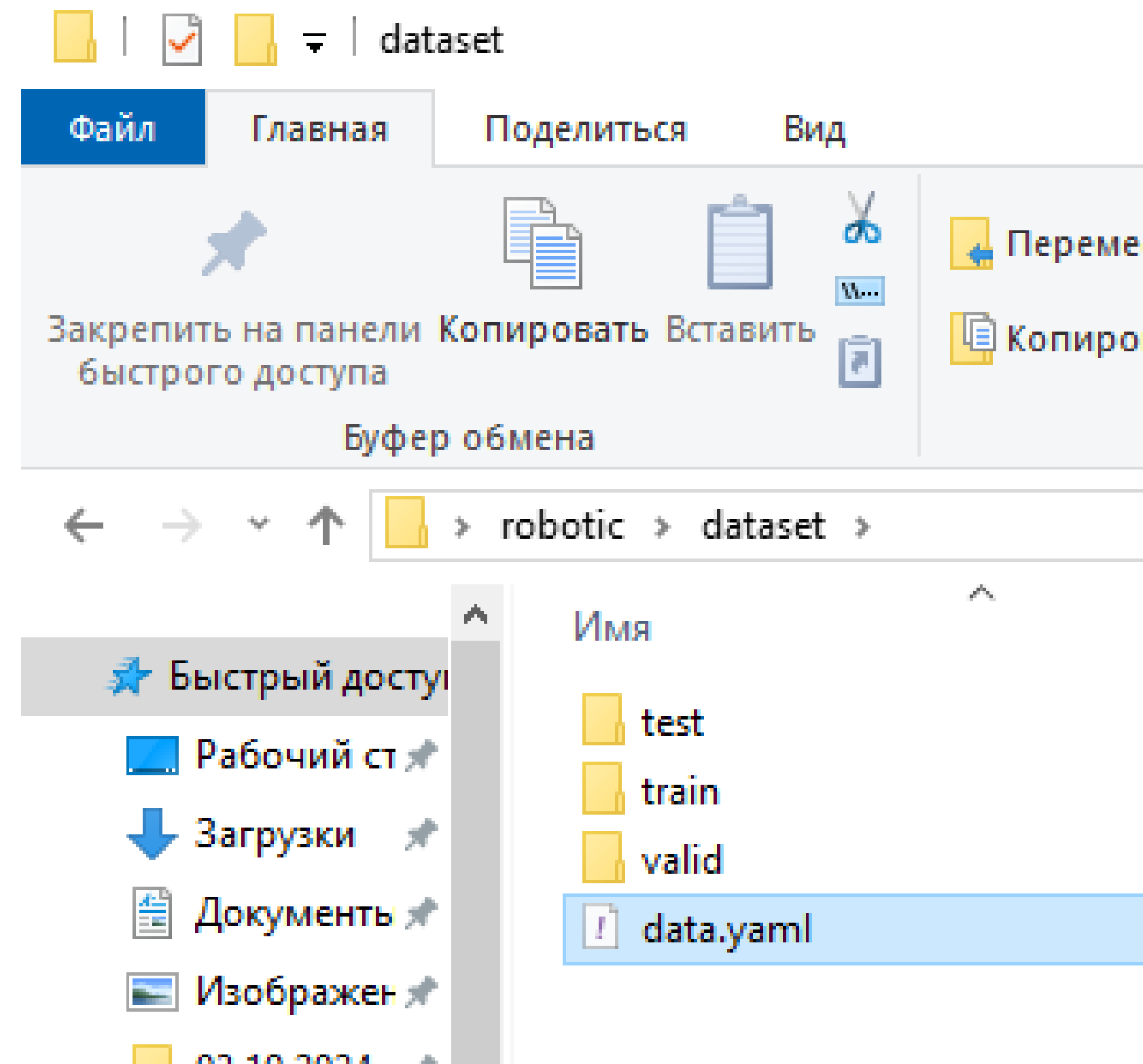
Файл Правка Формат

Basket  
Buttons (B/P)  
Buttons (G/O)  
Cubic  
Robot

data.txt – Блокнот

Файл Правка Формат Вид Справка

```
train: ../train/images  
val: ../valid/images  
test: ../test/images  
  
nc: 5  
names: ['Basket', 'Buttons (B/P)', 'Buttons (G/O)', 'Cubic', 'Robot']
```



Наш датасет на наших данных готов!

## Шаг 2 Получение центра Bounding Box

```
# Получаем список названий классов из модели
class_names = model.names # Это словарь {id: "class_name"}

# Итерируемся по каждому обнаруженному объекту
for box in result.bboxes:
    # Извлекаем координаты и конвертируем их в стандартные Python числа
    xywh_tensor = box.xywh[0]
    xywh = xywh_tensor.cpu().detach().tolist() # Преобразуем в список Python

    # Извлекаем ID класса и конвертируем в int
    class_id_tensor = box.cls[0]
    class_id = int(class_id_tensor.cpu().detach().item())

    # Получаем название класса по ID
    class_name = class_names.get(class_id, "Unknown")

    # Форматируем координаты для вывода
    x, y, w, h = xywh
    print(f"Box (x, y, w, h): ({x:.2f}, {y:.2f}, {w:.2f}, {h:.2f}), Class ID: {class_id}, Class Name: {class_name}")

# Помещаем обработанный кадр в очередь результатов
result_queue.put(plotted_image)
```



```
0: 384x640 2 Baskets, 1 Buttons (B/P), 1 Buttons (G/O), 1 Cubic, 1 Robot, 86.5ms
Speed: 2.0ms preprocess, 86.5ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)
Box (x, y, w, h): (903.89, 124.07, 74.47, 67.13), Class ID: 4, Class Name: Robot
Box (x, y, w, h): (968.65, 370.76, 45.27, 53.47), Class ID: 1, Class Name: Buttons (B/P)
Box (x, y, w, h): (565.07, 52.67, 55.47, 44.03), Class ID: 0, Class Name: Basket
Box (x, y, w, h): (551.03, 686.83, 54.06, 42.80), Class ID: 0, Class Name: Basket
Box (x, y, w, h): (179.93, 362.80, 40.30, 55.98), Class ID: 2, Class Name: Buttons (G/O)
Box (x, y, w, h): (822.29, 107.58, 18.78, 18.93), Class ID: 3, Class Name: Cubic
```

## Задание

1. Обучить свою модель на размеченном датасете через YOLOv8
2. Вывести x и y центров bounding box для робота в консоли

# Спасибо за внимание!

**Клюев Данил**

ассистент, УрФУ

**Пелевин Владимир**

к.пед.н., доцент, УрФУ

инженер, ООО Микроэлектроника и  
Робототехника