

Table of content:

- Download urn pics
- Urn annotation
- Download the ready dataset (YOLOv5 or COCO format annotations).
- YOLO train (see the notebook in the same folder), here just for completeness

1. Download urn pics:

A. Installation instructions:

<https://pypi.org/project/yandex-images-download/>
<https://pypi.org/project/lxml/>
sudo apt install openssl
if use Firefox install this driver
<https://github.com/mozilla/geckodriver/releases/tag/v0.26.0>

B. Alternative downloader (not tested):

<https://github.com/scirag/selenium-image-crawler>

C. Best examples:

```
yandex-images-download Firefox --keywords "Веб-камеры на избирательных участках" --limit 500 --driver-path  
'/home/your_path/projects/cv/yandex_pics/geckodriver'
```

secondary examples (not as good as they have unrealistic high quality):

```
yandex-images-download Firefox --keywords "избирательная урна" --limit 500 --driver-path  
'/home/your_path/projects/cv/yandex_pics/geckodriver'
```

```
yandex-images-download Firefox --keywords "избирательный участок" --limit 500 --driver-path  
'/home/your_path/projects/cv/yandex_pics/geckodriver'
```

```
yandex-images-download Firefox --keywords "урна для голововаания" --limit 500 --driver-path  
'/home/your_path/projects/cv/yandex_pics/geckodriver'
```

2. Urn annotation

Use: <https://github.com/tzutalin/labelImg>

LabelImg is a graphical image annotation tool

Annotations are saved as XML files in PASCAL VOC format, the format used by [ImageNet](#). Besides, it also supports YOLO format

An example of annotation

<https://github.com/darpan-jain/crowd-counting-using-tensorflow>

Use <https://roboflow.ai/> to save annotated files in both PASCAL VOC and YOLO format.

2. Download the ready dataset

The dataset includes 995 images. Urns are annotated either in YOLOv5 PyTorch format or in COCO format.

The following pre-processing was applied to each image:

- * Auto-orientation of pixel data (with EXIF-orientation stripping)
- * Resize to 416x416 (Stretch)

The following augmentation was applied to create 3 versions of each source image:

- * Random brightness adjustment of between -25 and +25 percent
- * Random exposure adjustment of between -25 and +25 percent
- * Random Gaussian blur of between 0 and 0.75 pixels

Urns are annotated in YOLO v5 PyTorch format

```
curl -L "https://app.roboflow.ai/ds/SESRKBTPJ8?key=P0A7b9KUtr" > roboflow.zip; unzip roboflow.zip; rm roboflow.zip
```

Urns are annotated in COCO format

```
curl -L "https://app.roboflow.ai/ds/SWoKG6vwDz?key=kj0J1ymOpt" > roboflow.zip; unzip roboflow.zip; rm roboflow.zip
```

4. YOLO training on a custom object (urn)

See notebook in the same folder for training and inference on a custom object. Below notes just for completeness.

change in [yolov5/models/yolov5s.yaml](#)

NC=1

train examples:

```
python train.py --img 416 --batch 16 --epochs 100 --data ./data/urn.yaml --cfg ./models/yolov5s.yaml --weights 'yolov5s.pt' --device 0
```

```
python train.py --img 640 --batch 8 --epochs 30 --data ./data/urn.yaml --cfg ./models/yolov5s.yaml --weights " " --device 0
```

```
python detect.py --source /home/your_path/projects/cv/datasets/election/election_2018_sample_2.mp4 --weights weights/best.pt --conf 0.4 --classes 0
```

train yolo in colab

<https://blog.roboflow.ai/how-to-train-yolov5-on-a-custom-dataset/>

train detectron on colab

<https://blog.roboflow.ai/how-to-train-detectron2/>