

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский Государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №3  
По дисциплине «ОИВИС»  
Тема: **“Обучение детекторов объектов”**

Выполнил:  
Студент 4 курса  
Группы ИИ-23  
Глухарев Д.Е.  
Проверила:  
Андренко К.В.

Брест 2025

**Цель:** осуществлять обучение нейросетевого детектора для решения задачи обнаружения заданных объектов

Вариант 5.

5	YOLOv10m	<b>Счетчики расхода воды:</b> <a href="https://universe.roboflow.com/kor3741-gmail-com/watermeteramv2/dataset/1">https://universe.roboflow.com/kor3741-gmail-com/watermeteramv2/dataset/1</a>
---	----------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Код программы:**

```
import os
import cv2
import requests
from PIL import Image
from ultralytics import YOLO
import yaml
import numpy as np
import torch

def main():
    """Основная функция выполнения лабораторной работы."""
    print("--- Проверка устройства ---")
    if torch.cuda.is_available():
        device = torch.cuda.current_device()
        gpu_name = torch.cuda.get_device_name(device)
        print(f"GPU обнаружено: {gpu_name}")
        print(f"Всего GPU: {torch.cuda.device_count()}")
        print(f"Используемое устройство: {device} ({gpu_name})")
        training_device = 'cuda'
    else:
        print("GPU не обнаружено. Используется CPU.")
        training_device = 'cpu'

    PROJECT_DIR = os.path.dirname(os.path.abspath(__file__))
    DATASET_PATH = PROJECT_DIR
    DATA_CONFIG_PATH = os.path.join(DATASET_PATH, "data.yaml")

    MODEL_NAME = "yolov10m.pt"

    SAVE_DIR = os.path.join(PROJECT_DIR, "runs", "detect", "water_meter_train")
    os.makedirs(SAVE_DIR, exist_ok=True)

    OUTPUT_DIR = os.path.join(PROJECT_DIR, "detection_results")
    os.makedirs(OUTPUT_DIR, exist_ok=True)

    print("\n--- Этап 1: Анализ датасета ---")
    print(f"Датасет находится в: {DATASET_PATH}")
    print(f"Конфигурация датасета: {DATA_CONFIG_PATH}")

    with open(DATA_CONFIG_PATH, 'r') as f:
        data_config = yaml.safe_load(f)
```

```

print(f"Количество классов: {data_config.get('nc', 'Не указано')}")
print(f"Названия классов: {data_config.get('names', 'Не указаны')}")
print(f"Путь к тренировочным изображениям (из config): {data_config.get('train', 'Не указан')}")
print(f"Путь к валидационным изображениям (из config): {data_config.get('val', 'Не указан')}")
print(f"Путь к тестовым изображениям (из config): {data_config.get('test', 'Не указан')}")

```

```

print("\n--- Этап 2: Обучение модели YOLOv10m ---")
print("Загрузка модели...")
model = YOLO(MODEL_NAME)
print(f"Модель будет использовать устройство: {next(model.model.parameters()).device}")
print(f"Запуск обучения на устройстве: {training_device}")

```

```

results = model.train(
    data=DATA_CONFIG_PATH,
    epochs=25, # Изменено на 25 эпох
    imgsz=608,
    batch=16,
    save_dir=SAVE_DIR,
    name="water_meter_experiment",
    exist_ok=True,
    device=training_device,
    workers=0 # <--- Вот этот параметр важен
)

```

```

print("\n--- Этап 3: Оценка эффективности на валидационной выборке ---")
# Укажите workers=0 и для валидации, если она использует DataLoader
validation_results = model.val(data=DATA_CONFIG_PATH, split='val', device=training_device,
workers=0)
print(f"Результаты валидации: {validation_results}")
print(f"mAP@0.5: {validation_results.box.map50:.4f}")
print(f"mAP@0.5:0.95 (среднее по IoU): {validation_results.box.map:.4f}")

```

```

print("\n--- Этап 4: Визуализация работы детектора на изображениях из датасета ---")

```

```

val_images_path = os.path.join(DATASET_PATH, data_config.get('val', 'valid/images')) # Предполагаем
формат из Roboflow

```

```

if not os.path.exists(val_images_path):
    print(f"Папка с валидационными изображениями не найдена: {val_images_path}")
    print("Попробуем использовать путь из data.yaml напрямую (возможно, он относительный).")
    val_images_path = os.path.join(os.path.dirname(DATA_CONFIG_PATH), data_config.get('val',
'valid/images'))
    if not os.path.exists(val_images_path):
        print(f"Папка с валидационными изображениями по пути из data.yaml тоже не найдена:
{val_images_path}")
        return # Выходим, если не можем найти изображения
    else:
        print(f"Путь к валидационным изображениям найден: {val_images_path}")

```

```

image_files = [f for f in os.listdir(val_images_path) if f.lower().endswith(('.jpg', '.jpeg', '.png', '.bmp'))]
if not image_files:
    print(f"В папке {val_images_path} не найдено изображений.")

```

```

return

num_images_to_visualize = min(5, len(image_files)) # Визуализируем первые 5 изображений или все,
если их меньше
print(f"Будет визуализировано {num_images_to_visualize} изображений из валидационной
выборки.")

for i in range(num_images_to_visualize):
    image_filename = image_files[i]
    image_path = os.path.join(val_images_path, image_filename)

    try:
        print(f"Обработка изображения {i+1}: {image_filename}")
        image = cv2.imread(image_path)

        if image is None:
            print(f"Ошибка: не удалось загрузить изображение {image_path}")
            continue

        results = model(image)

        annotated_frame = results[0].plot() # plot() рисует боксы и метки

        output_filename = os.path.join(OUTPUT_DIR, f"detected_val_image_{i+1}_{image_filename}")
        cv2.imwrite(output_filename, annotated_frame)
        print(f"Результат сохранен как: {output_filename}")

    except Exception as e:
        print(f"Ошибка при обработке изображения {image_path}: {e}")

print("\n--- Лабораторная работа завершена ---")
print("1. Датасет проанализирован.")
print("2. Модель YOLOv10m обучена.")
print("3. Эффективность оценена (mAP).")
print("4. Результаты обнаружения на изображениях из датасета сохранены в папку
'detection_results'.")
print("Теперь вы можете оформить отчет и загрузить код и отчет в репозиторий.")

if __name__ == '__main__':
    main()

```

### Вывод программы:

```

C:\Users\gaming\PycharmProjects\PythonProject\.venv\Scripts\python.exe
"C:\Users\gaming\Desktop\ОИИС\Лаба 3.py"
--- Проверка устройства ---
GPU обнаружено: NVIDIA GeForce RTX 2080 Ti
Всего GPU: 1
Используемое устройство: 0 (NVIDIA GeForce RTX 2080 Ti)

```

```

--- Этап 1: Анализ датасета ---

```

Датасет находится в: C:\Users\gaming\Desktop\ОИИС  
 Конфигурация датасета: C:\Users\gaming\Desktop\ОИИС\data.yaml  
 Количество классов: 12  
 Названия классов: ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'counter', 'liters']  
 Путь к тренировочным изображениям (из config): ../train/images  
 Путь к валидационным изображениям (из config): ../valid/images  
 Путь к тестовым изображениям (из config): ../test/images

--- Этап 2: Обучение модели YOLOv10m ---

Загрузка модели...

Модель будет использовать устройство: cpu

Запуск обучения на устройстве: cuda

Ultralytics 8.3.227 Python-3.8.10 torch-2.4.1+cu118 CUDA:0 (NVIDIA GeForce RTX 2080 Ti, 11264MiB)

engine\trainer: agnostic\_nms=False, amp=True, augment=False, auto\_augment=randaugment, batch=16, bgr=0.0, box=7.5, cache=False, cfg=None, classes=None, close\_mosaic=10, cls=0.5, compile=False, conf=None, copy\_paste=0.0, copy\_paste\_mode=flip, cos\_lr=False, cutmix=0.0, data=C:\Users\gaming\Desktop\data.yaml, degrees=0.0, deterministic=True, device=0, dfl=1.5, dnn=False, dropout=0.0, dynamic=False, embed=None, epochs=25, erasing=0.4, exist\_ok=True, fliplr=0.5, flipud=0.0, format=torchscript, fraction=1.0, freeze=None, half=False, hsv\_h=0.015, hsv\_s=0.7, hsv\_v=0.4, imgsz=608, int8=False, iou=0.7, keras=False, kobj=1.0, line\_width=None, lr0=0.01, lrf=0.01, mask\_ratio=4, max\_det=300, mixup=0.0, mode=train, model=yolov10m.pt, momentum=0.937, mosaic=1.0, multi\_scale=False, name=water\_meter\_experiment, nbs=64, nms=False, opset=None, optimize=False, optimizer=auto, overlap\_mask=True, patience=100, perspective=0.0, plots=True, pose=12.0, pretrained=True, profile=False, project=None, rect=False, resume=False, retina\_masks=False, save=True, save\_conf=False, save\_crop=False, save\_dir=C:\Users\gaming\Desktop\runs\detect\water\_meter\_experiment, save\_frames=False, save\_json=False, save\_period=-1, save\_txt=False, scale=0.5, seed=0, shear=0.0, show=False, show\_boxes=True, show\_conf=True, show\_labels=True, simplify=True, single\_cls=False, source=None, split=val, stream\_buffer=False, task=detect, time=None, tracker=botsort.yaml, translate=0.1, val=True, verbose=True, vid\_stride=1, visualize=False, warmup\_bias\_lr=0.1, warmup\_epochs=3.0, warmup\_momentum=0.8, weight\_decay=0.0005, workers=0, workspace=None

Overriding model.yaml nc=80 with nc=12

from	n	params	module	arguments
0	-1 1	1392	ultralytics.nn.modules.conv.Conv	[3, 48, 3, 2]
1	-1 1	41664	ultralytics.nn.modules.conv.Conv	[48, 96, 3, 2]
2	-1 2	111360	ultralytics.nn.modules.block.C2f	[96, 96, 2, True]
3	-1 1	166272	ultralytics.nn.modules.conv.Conv	[96, 192, 3, 2]
4	-1 4	813312	ultralytics.nn.modules.block.C2f	[192, 192, 4, True]
5	-1 1	78720	ultralytics.nn.modules.block.SCDown	[192, 384, 3, 2]
6	-1 4	3248640	ultralytics.nn.modules.block.C2f	[384, 384, 4, True]
7	-1 1	228672	ultralytics.nn.modules.block.SCDown	[384, 576, 3, 2]
8	-1 2	1689984	ultralytics.nn.modules.block.C2fCIB	[576, 576, 2, True]
9	-1 1	831168	ultralytics.nn.modules.block.SPPF	[576, 576, 5]
10	-1 1	1253088	ultralytics.nn.modules.block.PSA	[576, 576]
11	-1 1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6] 1	0	ultralytics.nn.modules.conv.Concat	[1]
13	-1 2	1993728	ultralytics.nn.modules.block.C2f	[960, 384, 2]
14	-1 1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
15	[-1, 4] 1	0	ultralytics.nn.modules.conv.Concat	[1]
16	-1 2	517632	ultralytics.nn.modules.block.C2f	[576, 192, 2]
17	-1 1	332160	ultralytics.nn.modules.conv.Conv	[192, 192, 3, 2]
18	[-1, 13] 1	0	ultralytics.nn.modules.conv.Concat	[1]
19	-1 2	831744	ultralytics.nn.modules.block.C2fCIB	[576, 384, 2, True]

20            -1 1 152448 ultralytics.nn.modules.block.SCDwn        [384, 384, 3, 2]  
21        [-1, 10] 1        0 ultralytics.nn.modules.conv.Concat        [1]  
22            -1 2 1911168 ultralytics.nn.modules.block.C2fCIB        [960, 576, 2, True]  
23        [16, 19, 22] 1 2294872 ultralytics.nn.modules.head.v10Detect        [12, [192, 384, 576]]  
YOLOv10m summary: 288 layers, 16,498,024 parameters, 16,498,008 gradients, 64.0 GFLOPs

Transferred 787/799 items from pretrained weights

Freezing layer 'model.23.dfl.conv.weight'

AMP: running Automatic Mixed Precision (AMP) checks...

AMP: checks passed

train: Fast image access (ping: 0.00.0 ms, read: 467.4126.2 MB/s, size: 41.6 KB)

train: Scanning C:\Users\gaming\Desktop\ОИИС\train\labels.cache... 3478 images, 0 backgrounds, 0 corrupt:

100% ————— 3478/3478 0.0s

val: Fast image access (ping: 0.00.0 ms, read: 509.1139.2 MB/s, size: 39.5 KB)

val: Scanning C:\Users\gaming\Desktop\ОИИС\valid\labels.cache... 389 images, 0 backgrounds, 0 corrupt:

100% ————— 389/389 0.0s

Plotting labels to C:\Users\gaming\Desktop\runs\detect\water\_meter\_experiment\labels.jpg...

optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best

'optimizer', 'lr0' and 'momentum' automatically...

optimizer: AdamW(lr=0.000625, momentum=0.9) with parameter groups 129 weight(decay=0.0), 142

weight(decay=0.0005), 141 bias(decay=0.0)

Image sizes 608 train, 608 val

Using 0 dataloader workers

Logging results to C:\Users\gaming\Desktop\runs\detect\water\_meter\_experiment

Starting training for 25 epochs...

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
1/25	7.25G	3.075	5.434	2.413	52	608: 100% ————— 218/218

2.0it/s 1:48

Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%
all	389	2059	0.501	0.608	0.595	0.311

13/13 2.8it/s 4.6s

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
2/25	7.25G	2.953	2.333	2.249	32	608: 100% ————— 218/218

2.1it/s 1:43

Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%
all	389	2059	0.613	0.662	0.661	0.328

13/13 2.9it/s 4.6s

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
3/25	7.3G	2.976	2.04	2.26	38	608: 100% ————— 218/218

2.1it/s 1:42

Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%
all	389	2059	0.752	0.756	0.809	0.398

13/13 2.8it/s 4.6s

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
4/25	7.25G	2.927	1.866	2.245	50	608: 100% ————— 218/218

2.1it/s 1:42

Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%		
13/13 2.9it/s 4.6s															
all								389	2059	0.842	0.824	0.881	0.46		
Epoch								GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
5/25								7.26G	2.903	1.723	2.223	76	608: 100%	218/218	
2.1it/s 1:42															
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%		
13/13 2.9it/s 4.5s															
all								389	2059	0.846	0.836	0.891	0.488		
Epoch								GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
6/25								7.29G	2.835	1.609	2.205	24	608: 100%	218/218	
2.1it/s 1:41															
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%		
13/13 2.8it/s 4.6s															
all								389	2059	0.854	0.833	0.883	0.432		
Epoch								GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
7/25								7.23G	2.799	1.517	2.185	41	608: 100%	218/218	
2.2it/s 1:41															
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%		
13/13 2.8it/s 4.6s															
all								389	2059	0.901	0.888	0.938	0.543		
Epoch								GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
8/25								7.22G	2.773	1.441	2.177	52	608: 100%	218/218	
2.1it/s 1:41															
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%		
13/13 2.8it/s 4.6s															
all								389	2059	0.887	0.888	0.925	0.492		
Epoch								GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
9/25								7.25G	2.736	1.414	2.164	56	608: 100%	218/218	
2.2it/s 1:41															
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%		
13/13 2.9it/s 4.6s															
all								389	2059	0.891	0.871	0.929	0.524		
Epoch								GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
10/25								7.27G	2.692	1.364	2.137	41	608: 100%		
218/218 2.2it/s 1:41															
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%		
13/13 2.8it/s 4.6s															
all								389	2059	0.92	0.903	0.949	0.561		
Epoch								GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size		
11/25								7.24G	2.705	1.331	2.135	32	608: 100%		
218/218 2.2it/s 1:41															



Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	
13/13 2.8it/s 4.6s														
all								389	2059	0.937	0.93	0.961	0.571	
Epoch								GPU_mem	box_loss	cls_loss	df_l_loss	Instances	Size	
19/25								7.25G	2.432	0.9337	2.129	34	608: 100%	
218/218 2.3it/s 1:37														
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	
13/13 2.8it/s 4.6s														
all								389	2059	0.926	0.926	0.96	0.569	
Epoch								GPU_mem	box_loss	cls_loss	df_l_loss	Instances	Size	
20/25								7.25G	2.408	0.912	2.133	34	608: 100%	
218/218 2.3it/s 1:37														
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	
13/13 2.9it/s 4.6s														
all								389	2059	0.946	0.928	0.966	0.596	
Epoch								GPU_mem	box_loss	cls_loss	df_l_loss	Instances	Size	
21/25								7.25G	2.387	0.89	2.119	36	608: 100%	218/218
2.3it/s 1:37														
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	
13/13 2.9it/s 4.6s														
all								389	2059	0.932	0.937	0.965	0.587	
Epoch								GPU_mem	box_loss	cls_loss	df_l_loss	Instances	Size	
22/25								7.25G	2.36	0.868	2.104	34	608: 100%	218/218
2.3it/s 1:37														
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	
13/13 2.9it/s 4.6s														
all								389	2059	0.951	0.926	0.97	0.605	
Epoch								GPU_mem	box_loss	cls_loss	df_l_loss	Instances	Size	
23/25								7.25G	2.295	0.8447	2.067	32	608: 100%	
218/218 2.3it/s 1:37														
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	
13/13 2.9it/s 4.5s														
all								389	2059	0.939	0.937	0.967	0.601	
Epoch								GPU_mem	box_loss	cls_loss	df_l_loss	Instances	Size	
24/25								7.24G	2.256	0.8264	2.058	26	608: 100%	
218/218 2.3it/s 1:36														
Class								Images	Instances	Box(P	R	mAP50	mAP50-95): 100%	
13/13 2.8it/s 4.6s														
all								389	2059	0.945	0.937	0.971	0.614	
Epoch								GPU_mem	box_loss	cls_loss	df_l_loss	Instances	Size	
25/25								7.25G	2.238	0.8074	2.051	32	608: 100%	
218/218 2.3it/s 1:37														

	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%
13/13	2.9it/s	4.6s					
	all	389	2059	0.945	0.942	0.973	0.618

25 epochs completed in 0.736 hours.

Optimizer stripped from C:\Users\gaming\Desktop\runs\detect\water\_meter\_experiment\weights\last.pt,  
33.5MB

Optimizer stripped from C:\Users\gaming\Desktop\runs\detect\water\_meter\_experiment\weights\best.pt,  
33.5MB

Validating C:\Users\gaming\Desktop\runs\detect\water\_meter\_experiment\weights\best.pt...

Ultralytics 8.3.227 Python-3.8.10 torch-2.4.1+cu118 CUDA:0 (NVIDIA GeForce RTX 2080 Ti, 11264MiB)

YOLOv10m summary (fused): 136 layers, 15,320,116 parameters, 0 gradients, 58.9 GFLOPs

	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%
13/13	2.8it/s	4.6s					
	all	389	2059	0.944	0.942	0.973	0.618
	0	151	196	0.851	0.875	0.908	0.533
	1	122	161	0.89	0.938	0.954	0.599
	2	97	106	0.966	0.943	0.987	0.595
	3	124	148	0.968	0.946	0.99	0.642
	4	148	184	0.984	0.967	0.982	0.616
	5	96	111	0.942	0.946	0.982	0.584
	6	125	151	0.935	0.948	0.987	0.624
	7	98	124	0.983	0.903	0.962	0.61
	8	60	67	0.949	0.94	0.986	0.585
	9	58	60	0.932	0.917	0.958	0.622
	counter	367	373	0.984	0.987	0.994	0.817
	liters	375	378	0.949	0.991	0.988	0.594

Speed: 0.3ms preprocess, 4.3ms inference, 0.0ms loss, 0.3ms postprocess per image

Results saved to C:\Users\gaming\Desktop\runs\detect\water\_meter\_experiment

--- Этап 3: Оценка эффективности на валидационной выборке ---

Ultralytics 8.3.227 Python-3.8.10 torch-2.4.1+cu118 CUDA:0 (NVIDIA GeForce RTX 2080 Ti, 11264MiB)

YOLOv10m summary (fused): 136 layers, 15,320,116 parameters, 0 gradients, 58.9 GFLOPs

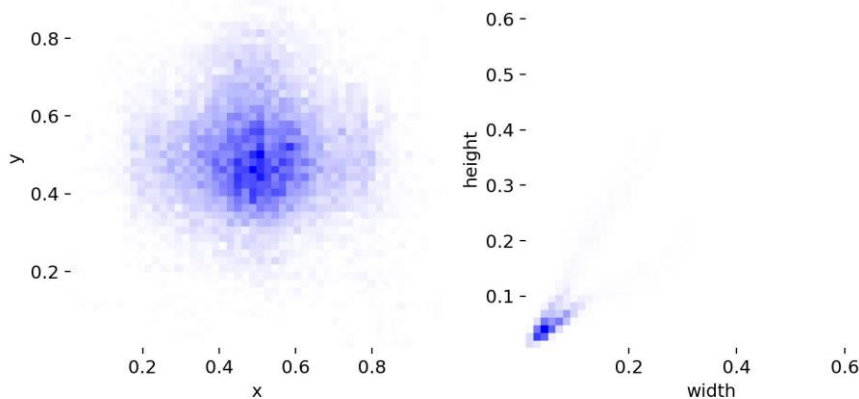
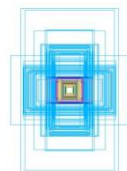
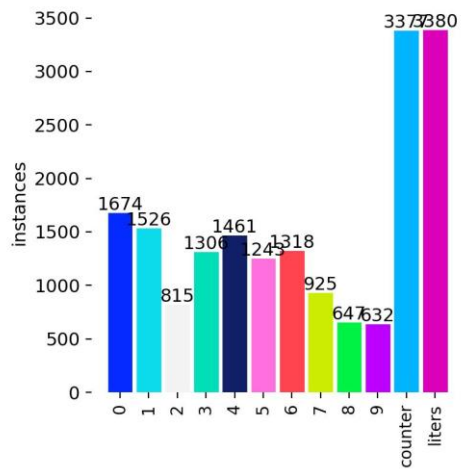
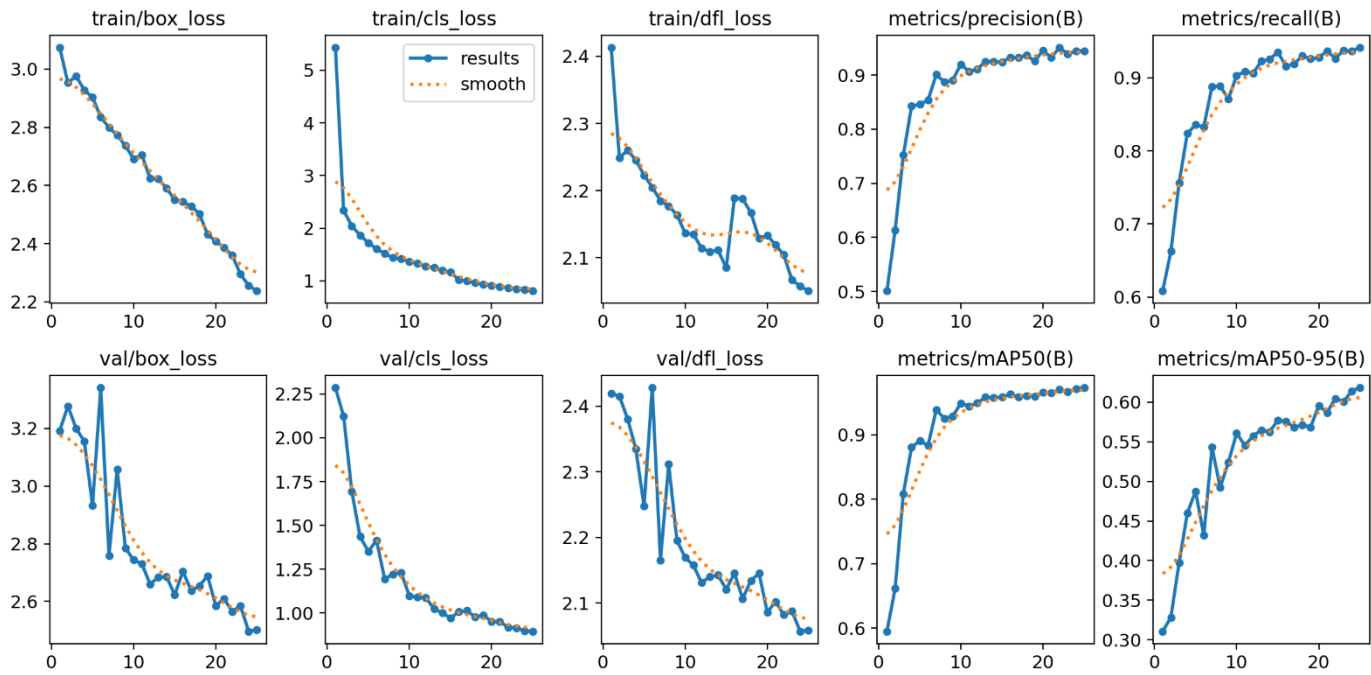
val: Fast image access (ping: 0.00.0 ms, read: 567.6102.3 MB/s, size: 42.2 KB)

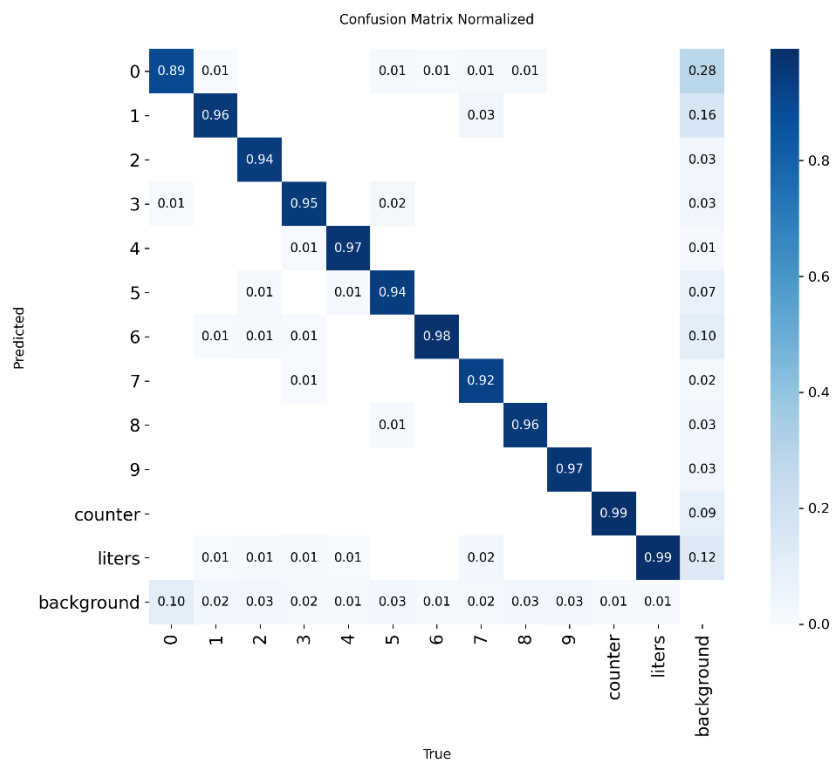
val: Scanning C:\Users\gaming\Desktop\ОИИ\valid\labels.cache... 389 images, 0 backgrounds, 0 corrupt:

100%			389/389	0.0s			
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100%
25/25	4.2it/s	6.0s					
	all	389	2059	0.945	0.942	0.973	0.618
	0	151	196	0.851	0.875	0.908	0.534
	1	122	161	0.89	0.938	0.954	0.601
	2	97	106	0.966	0.943	0.986	0.599
	3	124	148	0.968	0.946	0.99	0.639
	4	148	184	0.984	0.967	0.982	0.616
	5	96	111	0.941	0.946	0.982	0.584
	6	125	151	0.935	0.948	0.987	0.624
	7	98	124	0.983	0.903	0.962	0.609
	8	60	67	0.951	0.94	0.986	0.579
	9	58	60	0.932	0.916	0.958	0.615
	counter	367	373	0.984	0.987	0.994	0.816
	liters	375	378	0.949	0.991	0.988	0.593

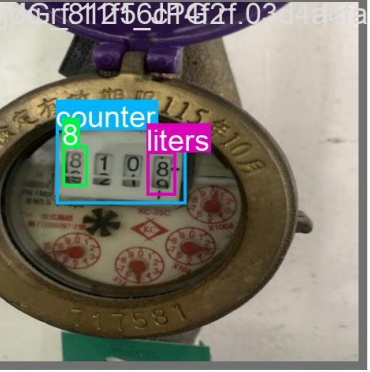
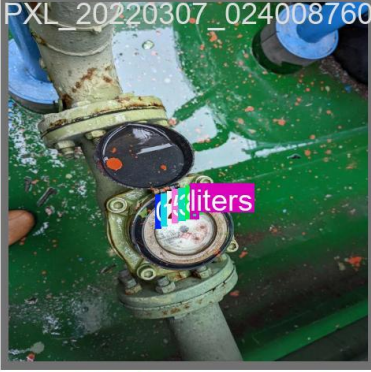
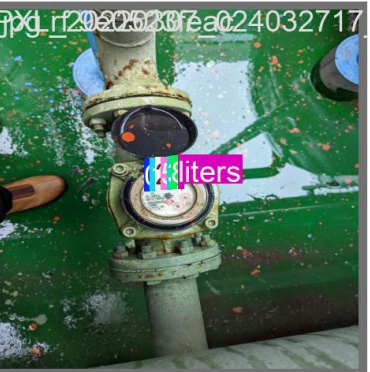
Speed: 0.5ms preprocess, 8.2ms inference, 0.0ms loss, 0.2ms postprocess per image  
Results saved to C:\Users\gaming\Desktop\\runs\detect\val

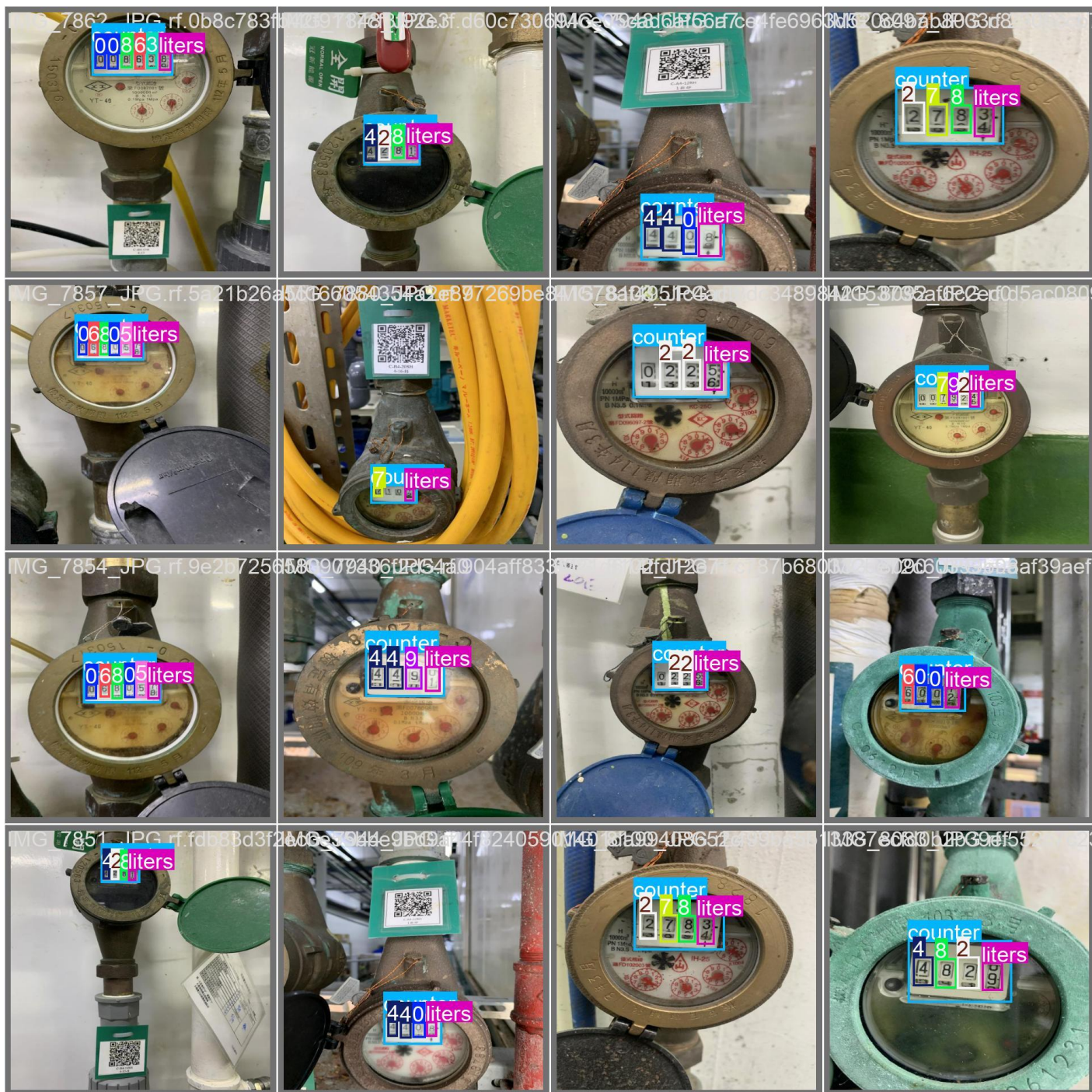
**Результаты оценки:**  
mAP50-95: 0.618  
mAP50: 0.973











Вывод: осуществил обучение нейросетевого детектора для решения задачи обнаружения заданных объектов.