

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-24

Рекун Д.А.

Проверила:

Андренко К.В.

Брест 2025

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

Вариант 2.

2	YOLOv10n	Номерные знаки авто: https://universe.roboflow.com/roboflow-universe-projects/license-plate-recognition-rxg4e/dataset/11
---	----------	--

1. Базируясь на своем варианте, ознакомится с выборкой для обучения детектора, выполнить необходимые преобразования данных для организации процесса обучения (если это нужно!);
2. Для заданной архитектуры нейросетевого детектора организовать процесс обучения для своей выборки. Оценить эффективность обучения на тестовой выборке (mAP);
3. Реализовать визуализацию работы детектора из пункта 1 (обнаружение знаков на отдельных фотографиях из сети Интернет);
4. Оформить отчет по выполненной работе, залить исходный код и отчет в соответствующий репозиторий на github.

Код программ:

```
lab3.py
import os
import yaml
import random
from pathlib import Path
import matplotlib.pyplot as plt
from PIL import Image
from ultralytics import YOLO
from roboflow import Roboflow

def safe_list_images(path):
    if not os.path.exists(path):
        print(f"[ОШИБКА] Путь не найден: {path}")
        return []
    return [
        os.path.join(path, f) for f in os.listdir(path)
        if f.lower().endswith((".jpg", ".jpeg", ".png"))
    ]

def load_yaml(path):
    with open(path, 'r', encoding='utf-8') as f:
        return yaml.safe_load(f)

def analyze_dataset(dataset_dir):
    print("\n==== АНАЛИЗ ДАТАСЕТА ====")

    dataset_dir = Path(dataset_dir)

    # Пути изображений
    train_p = dataset_dir / "train" / "images"
    val_p = dataset_dir / "valid" / "images"
    test_p = dataset_dir / "test" / "images"

    # Загрузка data.yaml
    yaml_path = dataset_dir / "data.yaml"
    data = load_yaml(yaml_path)
```

```

print(f"Классы: {data['nc']} ")
print(f"Названия классов: {data['names']} ")

# Подсчёты
for name, p in [("train", train_p), ("val", val_p), ("test", test_p)]:
    imgs = safe_list_images(str(p))
    print(f"{name}: {len(imgs)} изображений")

print("Анализ датасета завершён.\n")

def show_random_images(dataset_dir, count=6):
    dataset_dir = Path(dataset_dir)
    train_p = dataset_dir / "train" / "images"

    images = safe_list_images(str(train_p))
    if not images:
        print("Нет изображений для визуализации.")
        return

    chosen = random.sample(images, min(count, len(images)))

    plt.figure(figsize=(12, 8))
    for i, img_path in enumerate(chosen):
        img = Image.open(img_path)
        plt.subplot(2, 3, i + 1)
        plt.imshow(img)
        plt.title(os.path.basename(img_path))
        plt.axis("off")

    plt.tight_layout()
    plt.show()

def train_yolo(data_yaml, epochs=5):
    print("\n==== ОБУЧЕНИЕ YOLO ====")

    model = YOLO("yolol1n.pt")

    model.train(
        data=data_yaml,
        epochs=epochs,
        imgsz=640,
        batch=16,
        device="cpu",
        workers=8,
        project="runs/license_plate_detector",
        name="experiment",
        exist_ok=True
    )

    print("Обучение завершено.\n")
    return model

def validate_model(model, data_yaml):
    print("\n==== ВАЛИДАЦИЯ МОДЕЛИ ====")
    model.val(data=data_yaml)
    print("Валидация завершена.\n")

def run_prediction(model, img_path):
    print("\n==== ПРЕДСКАЗАНИЕ ====")
    results = model(img_path)
    results[0].show() # Покажет окно с bbox
    print("Предсказание завершено.\n")

def main():

```

```

print("Загрузка Roboflow...")
rf = Roboflow(api_key="Aw9Jet0111VWlnIJz9cL")
project = rf.workspace("roboflow-universe-projects").project("license-plate-recognition-rxg4e")
dataset = project.version(11).download("yolov11")

dataset_dir = dataset.location
data_yaml = os.path.join(dataset_dir, "data.yaml")

print(f"Датасет загружен в: {dataset_dir}")

analyze_dataset(dataset_dir)

show_random_images(dataset_dir)

model = train_yolo(data_yaml)

validate_model(model, data_yaml)

# Тестовое изображение
test_img = safe_list_images(os.path.join(dataset_dir, "test", "images"))
if test_img:
    run_prediction(model, test_img[0])

if __name__ == "__main__":
    main()

```

lab3test.py

```

import os
import random
from pathlib import Path
from ultralytics import YOLO
from PIL import Image
import matplotlib.pyplot as plt

def safe_list_images(path):
    if not os.path.exists(path):
        print(f"[ОШИБКА] Путь не найден: {path}")
        return []
    return [
        os.path.join(path, f) for f in os.listdir(path)
        if f.lower().endswith((".jpg", ".jpeg", ".png"))
    ]

def predict_and_show_many(model, image_paths, cols=5):
    """Показывает N предсказаний в одной фигуре"""

    num_images = len(image_paths)
    rows = (num_images + cols - 1) // cols

    fig, ax = plt.subplots(rows, cols, figsize=.5 * cols, 5 * rows))
    ax = ax.flatten()

    for i, img_path in enumerate(image_paths):
        print(f"\n==== ПРЕДСКАЗАНИЕ для: {img_path} ====")

        result = model(img_path)[0]
        result.save(filename=f"temp_{i}.jpg")

        img = Image.open(f"temp_{i}.jpg")
        ax[i].imshow(img)
        ax[i].set_title(os.path.basename(img_path))
        ax[i].axis("off")

```

```
for box in result.boxes:
    cls = int(box.cls[0])
    conf = float(box.conf[0])
    xyxy = box.xyxy[0].tolist()
    print(f"Объект: class={cls}, conf={conf:.3f}, coords={xyxy}")

for j in range(i+1, rows * cols):
    ax[j].axis("off")

plt.tight_layout()
plt.show()

def main():
    dataset_dir = r"C:\4curs1sem\ОИИС\labs\lab3\dataset"
    test_images = Path(dataset_dir) / "test" / "images"

    imgs = safe_list_images(str(test_images))
    if not imgs:
        print("Нет изображений в test/images!")
        return

    selected_images = random.sample(imgs, 5)
    print("Выбраны изображения:")
    for img in selected_images:
        print(" •", img)

    model_path = r"C:\4curs1sem\ОИИС\labs\lab3\weights\best.pt"
    model = YOLO(model_path)

    predict_and_show_many(model, selected_images)

if __name__ == "__main__":
    main()
```

Вывод программ:

Lab3.py

```

PS C:\4cursisem\OMC> & "C:/Program Files/Python311/python.exe" c:/4cursisem\OMC\labs\lab3\lab3.py
Загрузка Roboflow...
Loading Roboflow workspace...
Loading Roboflow project...
Датасет загружен в: C:\4cursisem\OMC\License-Plate-Recognition-11

== АНАЛИЗ ДАТАСЕТА ==
Классы: 1
Названия классов: ['license_plate']
train: 7057 изображений
val: 2048 изображений
test: 1020 изображений
Анализ датасета завершён.

== ОБУЧЕНИЕ YOLO ==
New https://pypi.org/project/ultralytics/8.3.23 available Update with 'pip install -U ultralytics'
Ultralytics 8.3.232 Python-3.11.2 torch-2.6.0+cpu CPU (11th Gen Intel Core i3-111564 @ 3.00GHz)
engine:trainer: agnostic_nms=False, amp=True, augment=False, auto_augment=randaugment, batch=16, bgr=0.0, box=7.5, cache=False, cfg=None, class es=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:\4cursisem\License-Plate-Recognition-11\data.yaml, degrees=0.0, deterministic=True, device=cpu, dfl=1.5, dn=0.0, dropout=0.0, dynamic=False, embed=None, epochs=5, erasing=0.4, exist_ok=True, flipr=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=640, int8=False, iou=0.7, keras=False, kobj=1.0, line_width=None, lr=0.01, mask_ratio=4, max_det=300, mixup=0.0, mode=train, model=yolovin.pt, momentum=0.937, mosaic=1.0, multi_scale=False, name=experiment, nbs=64, nms=False, opset=None, optimize=False, optimizer='auto', output_layer=True, pretrain=True, preshape=False, plots=True, pose=12.0, pretrain=True, profile=False, project=runs/license_plate_detector, rect=False, resume=False, retina_masks=False, save=True, save_conf=False, save_crop=False, save_dir=C:\4cursisem\runs\license_plate_detector\experiment, save_fnames=False, save_json=False, save_period=1, save_txt=False, scale=0.5, seed=0, shear=0.0, show=False, show_box=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=8, workspace=None
Overriding model.yaml nc=80 with nc=1

      from    n   params   module           arguments
0       -1    1      464   ultralytics.nn.modules.conv.Conv   [3, 16, 3, 2]
1       -1    1      4672  ultralytics.nn.modules.conv.Conv  [16, 32, 3, 2]
2       -1    1     6640  ultralytics.nn.modules.block.C3k2  [32, 64, 1, False, 0.25]
3       -1    1     36992 ultralytics.nn.modules.conv.Conv  [64, 64, 3, 2]
4       -1    1     26688 ultralytics.nn.modules.block.C3k2  [64, 128, 1, False, 0.25]

```

```

      from    n   params   module           arguments
0       -1    1      464   ultralytics.nn.modules.conv.Conv   [3, 16, 3, 2]
1       -1    1      4672  ultralytics.nn.modules.conv.Conv  [16, 32, 3, 2]
2       -1    1     6640  ultralytics.nn.modules.block.C3k2  [32, 64, 1, False, 0.25]
3       -1    1     36992 ultralytics.nn.modules.conv.Conv  [64, 64, 3, 2]
4       -1    1     26088 ultralytics.nn.modules.block.C3k2  [64, 128, 1, False, 0.25]
5       -1    1     147712 ultralytics.nn.modules.conv.Conv  [128, 128, 3, 2]
6       -1    1      87040 ultralytics.nn.modules.block.C3k2  [128, 128, 1, True]
7       -1    1     295424 ultralytics.nn.modules.conv.Conv  [128, 256, 3, 2]
8       -1    1     346112 ultralytics.nn.modules.block.C3k2  [256, 256, 1, True]
9       -1    1     164088 ultralytics.nn.modules.block.SPPF  [256, 256, 5]
10      -1    1     249728 ultralytics.nn.modules.block.C2PSA  [256, 256, 1]
11      -1    1      86720  ultralytics.nn.modules.upsample  [None, 2, 'nearest']
12      [-1, 6]   0   ultralytics.nn.modules.concat  [1]
13      -1    1     111296 ultralytics.nn.modules.block.C3k2  [384, 128, 1, False]
14      -1    1      8   torch.nn.modules.upsampling.upsample  [None, 2, 'nearest']
15      [-1, 4]   1      0   ultralytics.nn.modules.concat  [1]
16      -1    1     32096 ultralytics.nn.modules.block.C3k2  [256, 64, 1, False]
17      -1    1     36992 ultralytics.nn.modules.conv.Conv  [64, 64, 3, 2]
18      [-1, 13]  1      0   ultralytics.nn.modules.concat  [1]
19      -1    1     86720  ultralytics.nn.modules.block.C3k2  [192, 128, 1, False]
20      -1    1     147712 ultralytics.nn.modules.conv.Conv  [128, 128, 3, 2]
21      [-1, 10]  1      0   ultralytics.nn.modules.concat  [1]
22      -1    1     378880 ultralytics.nn.modules.block.C3k2  [384, 256, 1, True]
23      [16, 19, 22] 1     438867 ultralytics.nn.modules.head.Detect  [1, [64, 128, 256]]
YOLO1n summary: 181 layers, 2,590,035 parameters, 2,590,019 gradients, 6.4 GFLOPs

Transferred 448/499 items from pretrained weights
Freezing layer 'model.23.dfl.conv.weight'
train: Fast image access (ping: 0.10.0 ms, read: 171.172.2 MB/s, size: 19.2 KB)
train: Scanning C:\4cursisem\OMC\License-Plate-Recognition-11\train\labels.cache... 7057 images, 5 backgrounds, 0 corrupt: 100% 7
train: Scanning C:\4cursisem\OMC\License-Plate-Recognition-11\train\labels.cache... 7057 images, 5 backgrounds, 0 corrupt: 100% 7
7/7057 0.0s
val: Fast image access (ping: 0.10.0 ms, read: 201.288.0 MB/s, size: 21.9 KB)
val: Scanning C:\4cursisem\OMC\License-Plate-Recognition-11\valid\labels.cache... 2048 images, 3 backgrounds, 0 corrupt: 100% 2048
val: Scanning C:\4cursisem\OMC\License-Plate-Recognition-11\valid\labels.cache... 2048 images, 3 backgrounds, 0 corrupt: 100% 2048
8/2048 0.0s

```

Lab3test.py

```

PS C:\4cursisem\OMC> & "C:/Program Files/Python311/python.exe" c:/4cursisem\OMC\labs\lab3\lab3test.py
Выбрали изображения:
+ C:\4cursisem\OMC\labs\lab3\dataset\test\images\CarLongPlateGen2122.jpg.rf.ea77d8011297883dc6f523d5b7aba099.jpg
+ C:\4cursisem\OMC\labs\lab3\dataset\test\images\xemay1817.jpg.rf.fe5ccb7bc9abfc8c3716bad7825ec6a.jpg
+ C:\4cursisem\OMC\labs\lab3\dataset\test\images\xemay2454.jpg.rf.89ab001efb297a107c9f84686e855955.jpg
+ C:\4cursisem\OMC\labs\lab3\dataset\test\images\CarLongPlate559.jpg.rf.9a872c09ff085b1bc247dcae4958f4051.jpg
+ C:\4cursisem\OMC\labs\lab3\dataset\test\images\xemay290.jpg.rf.2857a4156101a02bc99e49753514e099.jpg

== ПРЕДСКАЗАНИЕ ДЛЯ: C:\4cursisem\OMC\labs\lab3\dataset\test\images\CarLongPlateGen2122.jpg.rf.ea77d8011297883dc6f523d5b7aba099.jpg ==
image 1/1 C:\4cursisem\labs\lab3\dataset\test\images\CarLongPlateGen2122.jpg.rf.ea77d8011297883dc6f523d5b7aba099.jpg: 416x640 1 License_Plate
120.0ms
Speed: 3.4ms preprocess, 120.3ms inference, 1.5ms postprocess per image at shape (1, 3, 416, 640)
Объект: class=0, conf=0.763, coords=[237.4900360107422, 138.7948760986328, 343.2478942871094, 171.92926025390625]

== ПРЕДСКАЗАНИЕ ДЛЯ: C:\4cursisem\OMC\labs\lab3\dataset\test\images\xemay1817.jpg.rf.fe5ccb7bc9abfc8c3716bad7825ec6a.jpg ==
image 1/1 C:\4cursisem\labs\lab3\dataset\test\images\xemay1817.jpg.rf.fe5ccb7bc9abfc8c3716bad7825ec6a.jpg: 416x640 1 License_Plate, 106.2ms
Speed: 80.0ms preprocess, 106.2ms inference, 1.2ms postprocess per image at shape (1, 3, 416, 640)
Объект: class=0, conf=0.889, coords=[167.29904174804688, 108.38555908203125, 230.4669647216797, 160.454540673828]

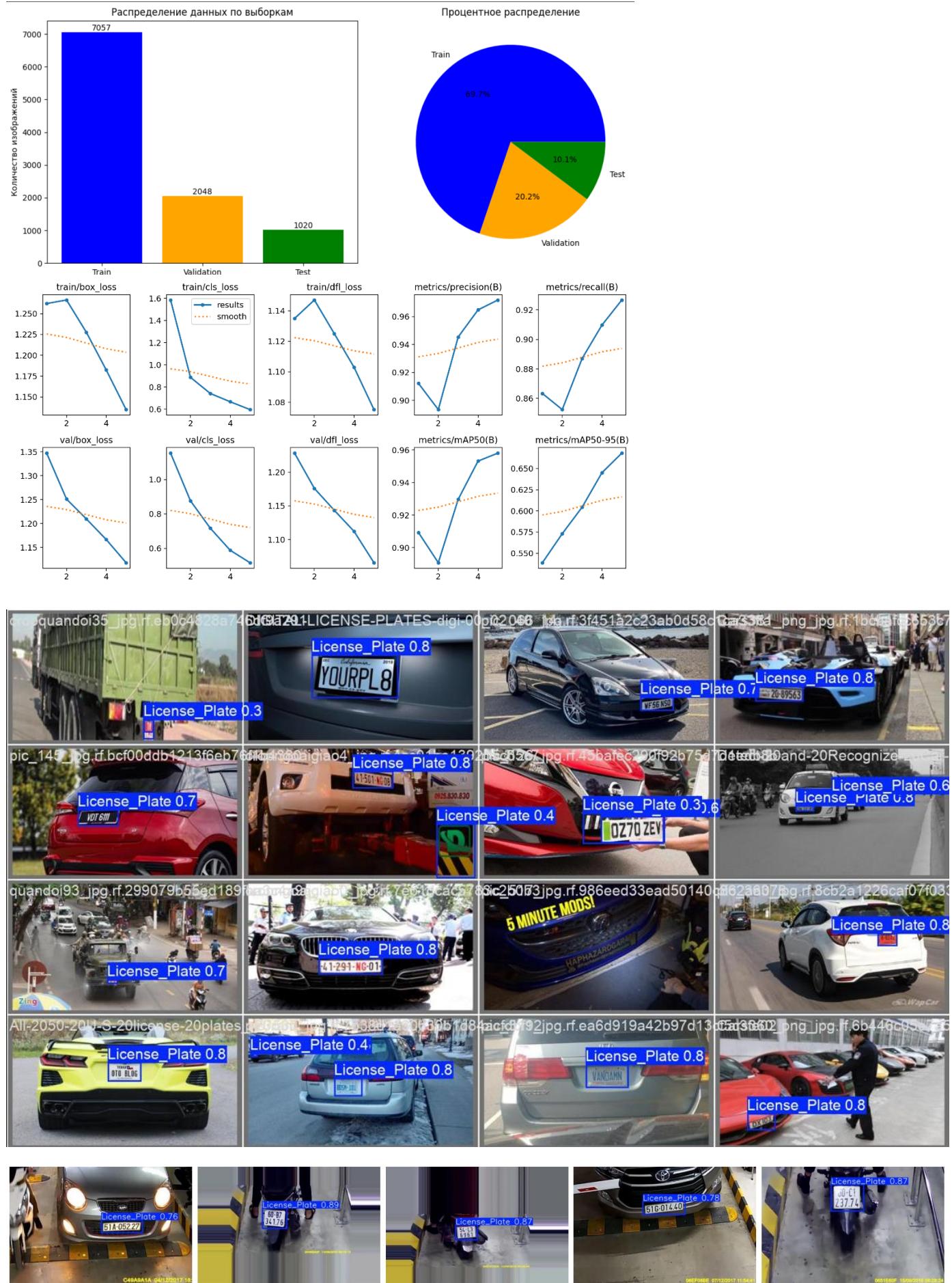
== ПРЕДСКАЗАНИЕ ДЛЯ: C:\4cursisem\OMC\labs\lab3\dataset\test\images\xemay2454.jpg.rf.89ab001efb297a107c9f84686e855955.jpg ==
image 1/1 C:\4cursisem\labs\lab3\dataset\test\images\xemay2454.jpg.rf.89ab001efb297a107c9f84686e855955.jpg: 416x640 1 License_Plate, 91.4ms
Speed: 2.0ms preprocess, 91.4ms inference, 1.0ms postprocess per image at shape (1, 3, 416, 640)
Объект: class=0, conf=0.872, coords=[188.4561767578125, 151.8695343017578, 236.87672424316486, 189.5818328857422]

== ПРЕДСКАЗАНИЕ ДЛЯ: C:\4cursisem\OMC\labs\lab3\dataset\test\images\CarLongPlate559.jpg.rf.9a872c09ff085b1bc247dcae4958f4051.jpg ==
image 1/1 C:\4cursisem\labs\lab3\dataset\test\images\CarLongPlate559.jpg.rf.9a872c09ff085b1bc247dcae4958f4051.jpg: 416x640 1 License_Plate, 84.0ms
Speed: 2.3ms preprocess, 84.0ms inference, 1.1ms postprocess per image at shape (1, 3, 416, 640)
Объект: class=0, conf=0.781, coords=[179.9219512939453, 98.04145812988281, 288.4876403808594, 124.04878997802734]

== ПРЕДСКАЗАНИЕ ДЛЯ: C:\4cursisem\OMC\labs\lab3\dataset\test\images\xemay290.jpg.rf.2857a4156101a02bc99e49753514e099.jpg ==
image 1/1 C:\4cursisem\labs\lab3\dataset\test\images\xemay290.jpg.rf.2857a4156101a02bc99e49753514e099.jpg: 416x640 1 License_Plate, 89.6ms
Speed: 2.7ms preprocess, 89.6ms inference, 1.1ms postprocess per image at shape (1, 3, 416, 640)
Объект: class=0, conf=0.873, coords=[179.2661895751953, 47.43080520629883, 263.3797302246094, 115.3881607055664]

```

Результаты



Вывод: осуществил обучение нейросетевого детектора YOLOv10n для решения задачи обнаружения номерных знаков машин.