Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

По дисциплине «Обработка изображений в ИС»

Тема: «Обучение детекторов объектов»

Выполнила:

Студентка 4 курса

Группы ИИ-21

Соболева П.С.

Проверил:

Крощенко А.А.

Брест 2024

Цель: осуществлять обучение нейросетевого детектора для решения задачи обнаружения дорожных знаков.

14	YOLOv9t

Код программы:

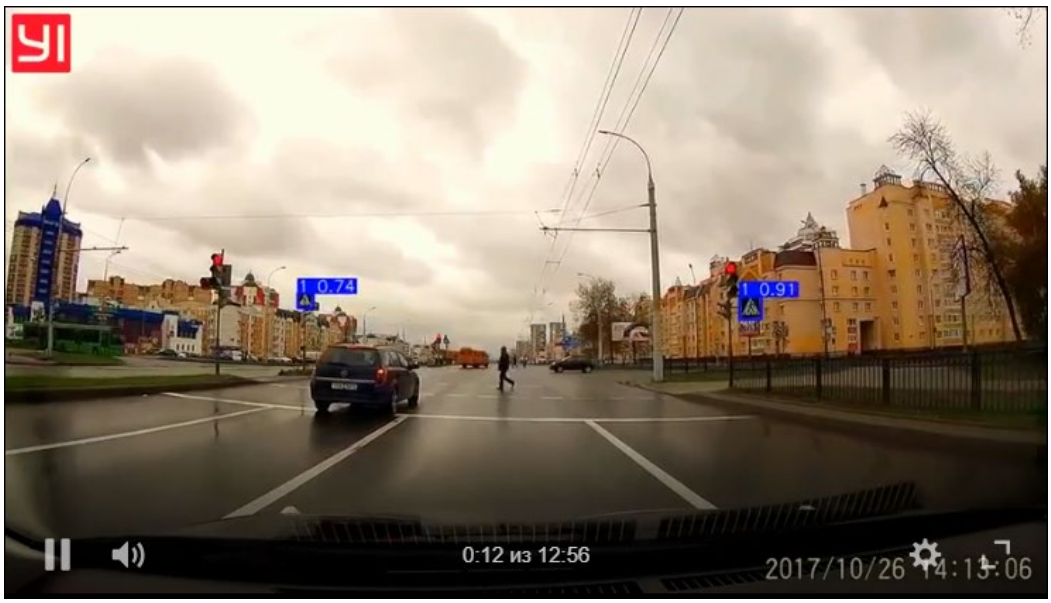
```
import os
import shutil
import pandas as pd
from sklearn.model_selection import train_test_split
from PIL import Image

# Пути к файлам и директориям
annotations_file_train = r"Polina\3\signs\rtsd-d3-gt\blue_rect\train_gt.csv"
annotations_file_test = r"Polina\3\signs\rtsd-d3-gt\blue_rect\test_gt.csv"
images_dir_train = r"Polina\3\signs\rtsd-d3-frames\train"
images_dir_test = r"Polina\3\signs\rtsd-d3-frames\test"
output_dir = r"Polina\3\src\yolo_data"
classes_file = r" Polina\3\src\classes.txt"

# Создание структуры папок для хранения данных
os.makedirs(f"{output_dir}/images/train", exist_ok=True)
os.makedirs(f"{output_dir}/images/val", exist_ok=True)
os.makedirs(f"{output_dir}/labels/train", exist_ok=True)
os.makedirs(f"{output_dir}/labels/val", exist_ok=True)

# Чтение аннотаций
df_train = pd.read_csv(annotations_file_train)
df_test = pd.read_csv(annotations_file_test)

# Создание сопоставления классов с идентификаторами
all_classes = pd.concat([df_train["sign_class"], df_test["sign_class"]]).unique()
class_mapping = {cls: idx for idx, cls in enumerate(sorted(all_classes))}

# Сохранение списка классов в файл
with open(classes_file, "w") as f:
    for cls in sorted(class_mapping):
        f.write(f"{cls}\n")

# Разделение данных на тренировочные и валидационные
train_files, val_files = train_test_split(df_train["filename"].unique(),
test_size=0.2, random_state=42)

def process_files(file_list, subset, df, images_dir):
    """
    Копирует изображения и формирует аннотации в формате YOLO для указанного набора
данных.
    """
    for filename in file_list:
        image_path = os.path.join(images_dir, filename)
```

```
            if not os.path.exists(image_path):
                print(f"Пропускаем отсутствующее изображение: {filename}")
                continue
            with Image.open(image_path) as image:
                img_width, img_height = image.size
            shutil.copy(image_path, f"{output_dir}/images/{subset}/{filename}")
            annotation_rows = df[df["filename"] == filename]
            label_path = os.path.join(output_dir, "labels", subset,
f"{os.path.splitext(filename)[0]}.txt")
            with open(label_path, "w") as f:
                for _, row in annotation_rows.iterrows():
                    class_id = class_mapping[row["sign_class"]]
                    x_center = (row["x_from"] + row["width"] / 2) / img_width
                    y_center = (row["y_from"] + row["height"] / 2) / img_height
                    bbox_width = row["width"] / img_width
                    bbox_height = row["height"] / img_height
                    f.write(f"{class_id} {x_center:.6f} {y_center:.6f} {bbox_width:.6f}
{bbox_height:.6f}\n")

# Обработка данных
process_files(train_files, "train", df_train, images_dir_train)
process_files(val_files, "val", df_train, images_dir_train)
process_files(df_test["filename"].unique(), "val", df_test, images_dir_test)

# Формирование файла data.yaml для YOLO
data_yaml_content = f"""
train: {output_dir}/images/train
val: {output_dir}/images/val
nc: {len(class_mapping)}  # Количество уникальных классов
names: {list(class_mapping.keys())}  # Перечень классов
"""
with open(f"{output_dir}/data.yaml", "w") as f:
    f.write(data_yaml_content)
print("Подготовка данных завершена успешно! Структура директорий и файлы созданы.")
```

Результат (видео-демонстрацию) работы НС можно найти на YandexDisk:

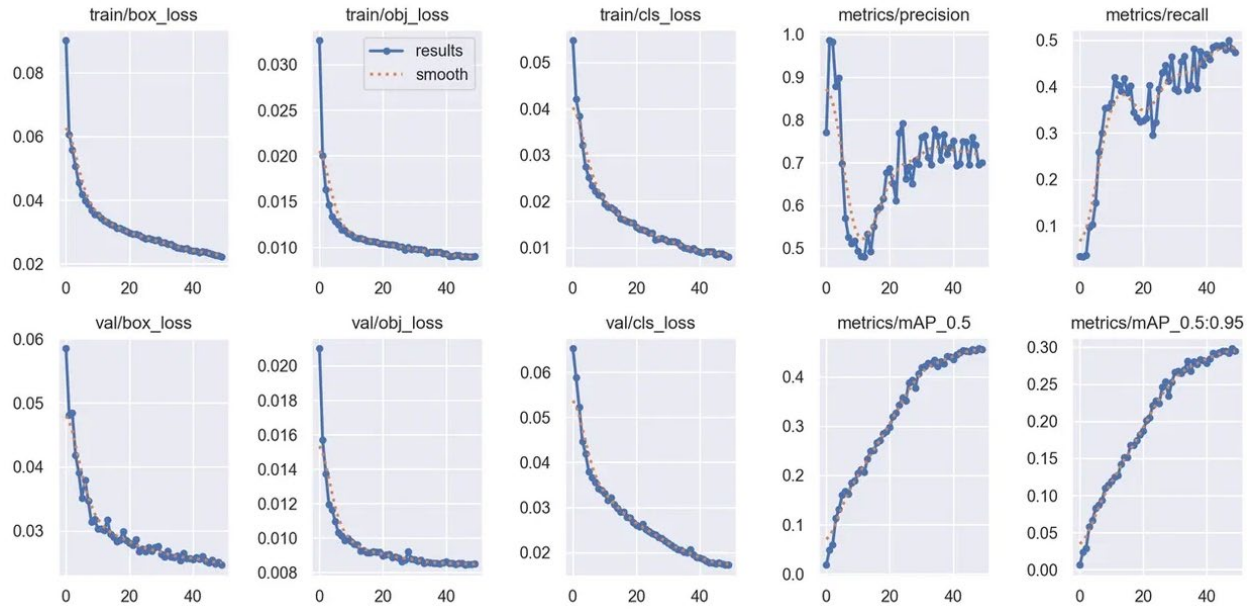
Ссылка на день:
https://disk.yandex.ru/i/06h0NBTR7HezLg
Ссылка на ночь:
https://disk.yandex.ru/i/-Jvm9_89goREdw

Кадры из видео-демонстрации(день/ночь):


1 0.74
1 0.91
0:12 из 12:56
2017/10/26 14:13:06


8 0.41
2017/10/28 20:11:42

Метрики:


train/box_loss
train/obj_loss
results
smooth
train/cls_loss
metrics/precision
metrics/recall
val/box_loss
val/obj_loss
val/cls_loss
metrics/mAP_0.5
metrics/mAP_0.5:0.95

Вывод: осуществила обучение НС, сконструированных на базе предобученных архитектур НС.