

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ
по дисциплине «Анализ изображений и видео» для студентов
направления подготовки
09.03.04 Программная инженерия
Направленность (профиль)
«Инженерия сложных программных систем»

Ставрополь 2026

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
Лабораторная работа 1. Семантическая сегментация с U-Net или DeepLab на датасете Cityscapes.....	4
Лабораторная работа 2. Инстанс-сегментация с Mask R-CNN в Detectron2	12
Лабораторная работа 3. Распознавание сцен с EfficientNet и Vision Transformer.....	20
Лабораторная работа 4. Трекинг объектов в видео: SORT и Deep SORT.....	28
Лабораторная работа 5. Классификация действий в видео с I3D или TimeSformer	36
Лабораторная работа 6. Постобработка результатов детекции и сегментации.....	44
Лабораторная работа 7. Полный пайплайн анализа видео в ROS 2	52

ВВЕДЕНИЕ

Дисциплина «Анализ изображений и видео» направлена на изучение современных методов компьютерного зрения, применяемых для обработки статических изображений, видеопотоков и данных робототехнических систем. В рамках лабораторных работ последовательно рассматриваются задачи семантической и инстанс-сегментации, распознавания сцен, трекинга объектов, классификации действий, постобработки результатов нейросетевых моделей и интеграции видеоаналитики в ROS 2.

Выполнение данных лабораторных работ позволяет освоить полный цикл построения прикладных решений компьютерного зрения: подготовку исходных данных, выбор и запуск моделей глубокого обучения, визуализацию результатов, расчет метрик качества, анализ ошибок и оформление полученных результатов. Особое внимание уделяется не только получению предсказаний моделей, но и их практической интерпретации, воспроизводимости экспериментов и возможности использования результатов в составе более сложных информационных и робототехнических систем.

Настоящий документ объединяет лабораторные работы по дисциплине в единый файл. Сохранение исходной структуры и оформления отдельных работ необходимо для удобства проверки, последовательного изучения материала и дальнейшего использования документа при подготовке к защите лабораторных заданий.

Лабораторная работа 1. Семантическая сегментация с U-Net или DeepLab на датасете Cityscapes

Цель работы: запустить предобученную модель семантической сегментации, получить цветную карту классов для дорожной сцены, выполнить наложение маски на исходное изображение и вычислить метрику mIoU на небольшой выборке изображений.

Теоретическое обоснование

Семантическая сегментация относится к задачам плотного предсказания: модель должна присвоить класс каждому пикселю изображения. В отличие от классификации, где ответом является один или несколько классов для всего изображения, сегментация формирует двумерную карту меток того же пространственного размера, что и исходный кадр. Для автономных платформ такая карта позволяет отличать дорогу от тротуара, препятствия от фона, а также выделять области, безопасные для движения.

Архитектура U-Net построена по принципу encoder-decoder. Encoder постепенно уменьшает пространственное разрешение и извлекает высокоуровневые признаки, decoder восстанавливает разрешение, а skip connections передают детализацию границ из ранних слоев. DeepLab использует atrous-свёртки и блок ASPP, что позволяет учитывать контекст на разных масштабах без чрезмерного уменьшения карты признаков.

Для датасета Cityscapes обычно используются классы road, sidewalk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, sky, person, rider, car, truck, bus, train, motorcycle и bicycle. В лабораторной работе допускается использовать уменьшенный фрагмент датасета или подготовленный преподавателем набор из нескольких изображений и масок.

Метрика IoU вычисляется как отношение площади пересечения предсказанной и эталонной области класса к площади их объединения. mIoU

является средним значением IoU по всем анализируемым классам. Если класс отсутствует одновременно в предсказании и разметке, его обычно исключают из усреднения, чтобы не исказить итоговую оценку.

Визуализация результата является обязательной частью работы: студент должен сохранить исходное изображение, цветную карту классов и наложение сегментации с прозрачностью. Такая проверка позволяет увидеть ошибки, которые не всегда очевидны по одной числовой метрике: размытые границы, перепутанные классы и пропуски мелких объектов.

Таблица 1 – Основные компоненты лабораторной работы

Компонент	Назначение
Cityscapes / sample-набор	Исходные изображения городских сцен и эталонные маски классов.
U-Net / DeepLab	Нейросетевая модель, предсказывающая класс каждого пикселя.
Color palette	Преобразование индексов классов в цветную карту сегментации.
mIoU	Средняя Intersection over Union по классам; основная метрика качества сегментации.

Схема взаимодействия компонентов

Схема лабораторной работы № 1

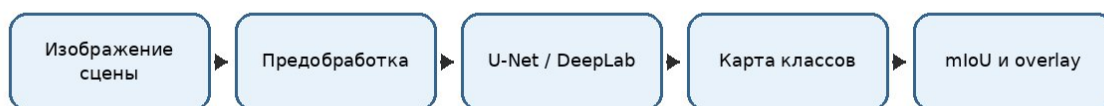


Рисунок 1 – Логическая схема выполнения лабораторной работы № 1

Пример выполнения задания

Ниже приведён учебный пример выполнения лабораторной работы. Его цель – показать последовательность проектных решений: как подготовить данные, какие компоненты создать, каким способом проверить корректность и какие результаты следует включить в отчёт.

Шаг 1. Подготовка окружения.

Создайте виртуальное окружение Python и установите библиотеки для работы с PyTorch, OpenCV, NumPy и segmentation_models_pytorch. Если GPU недоступен.

Листинг 1 – Подготовка окружения

```
python -m venv .venv
source .venv/bin/activate
pip install torch torchvision opencv-python pillow numpy matplotlib
pip install segmentation-models-pytorch albumentations tqdm
```

Шаг 2. Структура проекта.

Создайте каталог, в котором отдельно хранятся изображения, эталонные маски, веса модели и результаты. Это упрощает проверку отчёта и делает эксперимент воспроизводимым.

Листинг 2 – Структура проекта

```
segmentation_lab/
data/images/      # тестовые изображения Cityscapes
data/masks/       # эталонные маски с индексами классов
checkpoints/      # unet_resnet34_cityscapes.pth
outputs/          # mask.png, overlay.png, metrics.csv
infer_segmentation.py
```

Шаг 3. Загрузка модели.

В примере используется U-Net с encoder_name=resnet34. Число выходных каналов равно числу классов Cityscapes. Веса загружаются из checkpoint, подготовленного заранее или выданного преподавателем.

Листинг 3 – Загрузка модели

```
import torch
import segmentation_models_pytorch as smp

DEVICE = 'cuda' if torch.cuda.is_available() else 'cpu'
NUM_CLASSES = 19
```

```

model = smp.Unet(
    encoder_name='resnet34',
    encoder_weights=None,
    in_channels=3,
    classes=NUM_CLASSES
).to(DEVICE)
state = torch.load('checkpoints/unet_resnet34_cityscapes.pth',
map_location=DEVICE)
model.load_state_dict(state)
model.eval()

```

Шаг 4. Инференс и визуализация.

Изображение приводится к RGB, нормализуется и передаётся в модель. По выходному тензору выбирается класс с максимальным logit для каждого пикселя, после чего карта классов преобразуется в цветное изображение.

Листинг 4 – Инференс и визуализация

```

import cv2
import numpy as np
from PIL import Image

PALETTE = np.array([
    [128, 64, 128], [244, 35, 232], [ 70, 70, 70], [102, 102, 156],
    [190, 153, 153], [153, 153, 153], [250, 170, 30], [220, 220, 0],
    [107, 142, 35], [152, 251, 152], [ 70, 130, 180], [220, 20, 60],
    [255, 0, 0], [ 0, 0, 142], [ 0, 0, 70], [ 0, 60, 100],
    [ 0, 80, 100], [ 0, 0, 230], [119, 11, 32]
], dtype=np.uint8)

image = cv2.imread('data/images/frankfurt_000001.png')
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
resized = cv2.resize(rgb, (512, 256))
x = torch.from_numpy(resized).permute(2, 0, 1).float() / 255.0
x = x.unsqueeze(0).to(DEVICE)

with torch.no_grad():
    logits = model(x)
    pred = logits.argmax(1).squeeze().cpu().numpy().astype(np.uint8)
    color_mask = PALETTE[pred]
    overlay = cv2.addWeighted(resized, 0.55, color_mask, 0.45, 0)
    Image.fromarray(color_mask).save('outputs/pred_mask.png')
    Image.fromarray(overlay).save('outputs/overlay.png')

```

Шаг 5. Расчёт mIoU.

Для нескольких изображений вычислите IoU по каждому классу и среднее значение. В отчёте приведите таблицу с IoU для не менее чем пяти классов, присутствующих в сцене.

Листинг 5 – Расчёт mIoU

```
def mean_iou(pred, target, num_classes=19):
    values = []
    for cls in range(num_classes):
        p = pred == cls
        t = target == cls
        union = np.logical_or(p, t).sum()
        if union == 0:
            continue
        inter = np.logical_and(p, t).sum()
        values.append(inter / union)
    return float(np.mean(values)) if values else 0.0
```

Порядок выполнения работы

1. Изучить цель работы и теоретическое обоснование, выписать назначение основных компонентов схемы.
2. Подготовить рабочее окружение, структуру проекта и исходные данные в соответствии с примером выполнения.
3. Реализовать базовый пример, убедиться в корректности запуска на одном изображении или видеоролике.
4. Адаптировать решение под индивидуальный вариант: заменить данные, классы, параметры, модель или сценарий.
5. Сохранить промежуточные и итоговые результаты: изображения, видео, таблицы метрик, логи и фрагменты конфигурации.
6. Проанализировать качество результата, указать типичные ошибки алгоритма и возможные способы улучшения.

7. Оформить отчёт и подготовить ответы на вопросы для защиты.

Задания по вариантам

Номер варианта определяется по порядковому номеру студента в журнале группы. Если номер больше 14, используется формула: вариант = $((N - 1) \bmod 14) + 1$. Вариант выполняется на основе общего примера, но с изменённой предметной областью, входными данными, параметрами эксперимента и анализируемыми классами.

Таблица 2 – Индивидуальные задания по вариантам

Вариант	Индивидуальное задание
1	1) Подобрать 3-5 изображений по теме «городская дорога». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: автомобили, пешеходы, дорожные знаки. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
2	1) Подобрать 3-5 изображений по теме «парковка торгового центра». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: машины, тележки, люди. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
3	1) Подобрать 3-5 изображений по теме «аудитория университета». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: столы, стулья, доска, человек. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
4	1) Подобрать 3-5 изображений по теме «коридор учебного корпуса». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: двери, люди, указатели. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
5	1) Подобрать 3-5 изображений по теме «складское помещение». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: коробки, паллеты, погрузчик. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
6	1) Подобрать 3-5 изображений по теме «промышленный цех». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: рабочие, станки, средства защиты. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти

Вариант	Индивидуальное задание
	присутствующим классам и объяснить ошибки модели.
7	1) Подобрать 3-5 изображений по теме «парк или пешеходная зона». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: люди, велосипеды, деревья. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
8	1) Подобрать 3-5 изображений по теме «медицинский кабинет». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: оборудование, инструменты, персонал. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
9	1) Подобрать 3-5 изображений по теме «двор жилого комплекса». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: люди, машины, детская площадка. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
10	1) Подобрать 3-5 изображений по теме «перекрёсток». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: светофоры, транспорт, пешеходные переходы. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
11	1) Подобрать 3-5 изображений по теме «железнодорожная платформа». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: пассажиры, состав, таблички. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
12	1) Подобрать 3-5 изображений по теме «торговый зал». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: витрины, корзины, покупатели. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
13	1) Подобрать 3-5 изображений по теме «спортивная площадка». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: игроки, мяч, разметка. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.
14	1) Подобрать 3-5 изображений по теме «сцена с БПЛА». 2) Выполнить семантическую сегментацию с U-Net или DeepLab; отдельно выделить классы, связанные с объектами: дорога, деревья, здания сверху. 3) Построить overlay и цветную карту классов для каждого изображения. 4) Рассчитать mIoU минимум по пяти присутствующим классам и объяснить ошибки модели.

Содержание отчёта

- тема и цель лабораторной работы;
- краткое описание используемого набора данных, структуры проекта и выбранной модели;
- версии Python, PyTorch/TensorFlow, OpenCV, CUDA или CPU-среды, а также используемых библиотек;
- ключевые листинги кода с пояснением назначения основных блоков;
- результаты проверки: изображения, кадры видео, графики, таблицы метрик и фрагменты логов;
- анализ ошибок, ограничений метода и вывод о применимости изученного подхода в системах компьютерного зрения;
- ответы на вопросы для защиты.

Вопросы для защиты работы

1. Чем семантическая сегментация отличается от детекции объектов?
2. Почему U-Net использует skip connections?
3. Что показывает mIoU и почему одной Pixel Accuracy недостаточно?
4. Какие классы Cityscapes наиболее важны для автономного движения?
5. Какие ошибки сегментации опаснее всего для робота или беспилотной платформы?

Лабораторная работа 2. Инстанс-сегментация с Mask R-CNN в Detectron2

Цель работы: применить Mask R-CNN для выделения отдельных экземпляров объектов, визуализировать bounding box, класс, confidence и маску каждого объекта, сохранить бинарные маски и сравнить время инференса на CPU и GPU.

Теоретическое обоснование

Инстанс-сегментация решает более сложную задачу, чем семантическая сегментация: необходимо не только определить класс пикселя, но и разделить разные экземпляры одного класса. Например, две машины рядом должны получить две разные маски, хотя семантически они относятся к классу car.

Mask R-CNN расширяет Faster R-CNN отдельной веткой mask head. Сначала backbone и FPN строят карты признаков, затем RPN предлагает области-кандидаты, ROI Align извлекает признаки фиксированного размера, а финальные головы предсказывают class, bounding box и mask.

Detectron2 предоставляет готовые конфигурации и веса для моделей, обученных на COCO. Это удобно для учебной работы, поскольку студент может сосредоточиться на инференсе, визуализации, извлечении масок и анализе ограничений, а не на долгом обучении модели.

Для каждой найденной инстанции модель возвращает class id, confidence score, bounding box и бинарную маску. Confidence threshold влияет на баланс между пропусками и ложными срабатываниями: слишком высокий порог удаляет слабые, но правильные объекты; слишком низкий порог добавляет шум.

При сравнении CPU и GPU важно измерять не только один кадр, но и среднее время по серии повторов. Первый запуск часто включает загрузку

весов, компиляцию CUDA-ядра или прогрев памяти, поэтому его лучше исключать из расчёта.

Таблица 1 – Основные компоненты лабораторной работы

Компонент	Назначение
Detectron2	Фреймворк для задач detection и instance segmentation.
Mask R-CNN	Модель, формирующая bounding box, класс и отдельную маску объекта.
COCO classes	Набор классов, на котором обучена базовая модель.
Visualizer	Средство отрисовки рамок, масок, классов и confidence.
CPU/GPU benchmark	Сравнение времени инференса для разных устройств.

Схема взаимодействия компонентов

Схема лабораторной работы № 2

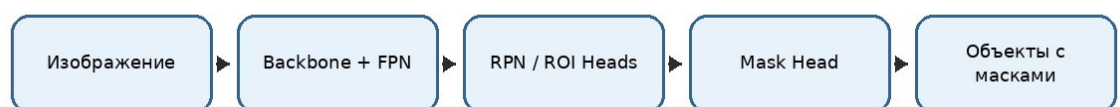


Рисунок 1 – Логическая схема выполнения лабораторной работы № 2

Пример выполнения задания

Ниже приведён учебный пример выполнения лабораторной работы. Его цель – показать последовательность проектных решений: как подготовить данные, какие компоненты создать, каким способом проверить корректность и какие результаты следует включить в отчёт.

Шаг 1. Установка окружения.

Выполняйте работу в Linux или Google Colab. Для локального запуска убедитесь, что версии PyTorch, CUDA и Detectron2 совместимы. Если GPU отсутствует, используйте CPU и уменьшите размер изображения.

Листинг 1 – Установка окружения

```
python -m venv .venv
source .venv/bin/activate
pip install torch torchvision opencv-python matplotlib
# Detectron2 устанавливается по инструкции для выбранной версии
PyTorch/CUDA.
# В Colab обычно используется wheel из официального репозитория
detectron2.
```

Шаг 2. Создание конфигурации модели.

Загрузите конфигурацию `mask_rcnn_R_50_FPN_3x` и установите порог уверенности. Для учебного примера достаточно порога 0.5, но в индивидуальном варианте его нужно варьировать.

Листинг 2 – Создание конфигурации модели

```
from detectron2.engine import DefaultPredictor
from detectron2.config import get_cfg
from detectron2 import model_zoo

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file(
    'COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml'))
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url(
    'COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml')
cfg.MODEL.DEVICE = 'cuda' # заменить на 'cpu', если GPU недоступен
predictor = DefaultPredictor(cfg)
```

Шаг 3. Инференс и визуализация.

Выполните предсказание на изображении с несколькими объектами и сохраните результат с рамками и масками. В отчёте нужно показать исходное изображение и изображение после отрисовки результатов.

Листинг 3 – Инференс и визуализация

```
import cv2
from detectron2.utils.visualizer import Visualizer, ColorMode
from detectron2.data import MetadataCatalog

image = cv2.imread('data/street_objects.jpg')
outputs = predictor(image)
```

```

metadata = MetadataCatalog.get(cfg.DATASETS.TRAIN[0])
vis = Visualizer(image[:, :, :-1], metadata=metadata,
                 instance_mode=ColorMode.IMAGE_BW)
result = vis.draw_instance_predictions(outputs['instances'].to('cpu'))
cv2.imwrite('outputs/mask_rcnn_result.png', result.get_image()[:, :, :-1])

```

Шаг 4. Сохранение бинарных масок.

Извлеките `pred_masks` и сохраните каждую маску отдельным PNG-файлом. Это показывает, что результатом является не только картинка для просмотра, но и данные, которые можно использовать в дальнейшей обработке.

Листинг 4 – Сохранение бинарных масок

```

import numpy as np
from pathlib import Path

Path('outputs/masks').mkdir(parents=True, exist_ok=True)
instances = outputs['instances'].to('cpu')
for i, mask in enumerate(instances.pred_masks.numpy()):
    class_id = int(instances.pred_classes[i])
    score = float(instances.scores[i])
    mask_img = (mask.astype(np.uint8) * 255)
    cv2.imwrite(f'outputs/masks/obj_{i:02d}_class_{class_id}_{score:.2f}.png',
                mask_img)

```

Шаг 5. Измерение скорости.

Сравните среднее время инференса на CPU и GPU. Если локального GPU нет, допустимо выполнить GPU-часть в Colab и приложить логи среды.

Листинг 5 – Измерение скорости

```

import time

def benchmark(predictor, image, repeats=20):
    for _ in range(3):
        _ = predictor(image) # прогрев
    t0 = time.perf_counter()
    for _ in range(repeats):
        _ = predictor(image)
    return (time.perf_counter() - t0) / repeats

```

```
avg_time = benchmark(predictor, image)
print(f'Average inference time: {avg_time:.4f} s')
```

Порядок выполнения работы

1. Изучить цель работы и теоретическое обоснование, выписать назначение основных компонентов схемы.
2. Подготовить рабочее окружение, структуру проекта и исходные данные в соответствии с примером выполнения.
3. Реализовать базовый пример, убедиться в корректности запуска на одном изображении или видеоролике.
4. Адаптировать решение под индивидуальный вариант: заменить данные, классы, параметры, модель или сценарий.
5. Сохранить промежуточные и итоговые результаты: изображения, видео, таблицы метрик, логи и фрагменты конфигурации.
6. Проанализировать качество результата, указать типичные ошибки алгоритма и возможные способы улучшения.
7. Оформить отчёт и подготовить ответы на вопросы для защиты.

Задания по вариантам

Номер варианта определяется по порядковому номеру студента в журнале группы. Если номер больше 14, используется формула: вариант = $((N - 1) \bmod 14) + 1$. Вариант выполняется на основе общего примера, но с изменённой предметной областью, входными данными, параметрами эксперимента и анализируемыми классами.

Таблица 2 – Индивидуальные задания по вариантам

Вариант	Индивидуальное задание
1	1) Подготовить изображение или набор изображений «городская дорога», где присутствуют перекрывающиеся объекты: автомобили, пешеходы, дорожные знаки. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
2	1) Подготовить изображение или набор изображений «парковка торгового центра», где присутствуют перекрывающиеся объекты:

Вариант	Индивидуальное задание
	машины, тележки, люди. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
3	1) Подготовить изображение или набор изображений «аудитория университета», где присутствуют перекрывающиеся объекты: столы, стулья, доска, человек. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
4	1) Подготовить изображение или набор изображений «коридор учебного корпуса», где присутствуют перекрывающиеся объекты: двери, люди, указатели. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
5	1) Подготовить изображение или набор изображений «складское помещение», где присутствуют перекрывающиеся объекты: коробки, паллеты, погрузчик. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
6	1) Подготовить изображение или набор изображений «промышленный цех», где присутствуют перекрывающиеся объекты: рабочие, станки, средства защиты. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
7	1) Подготовить изображение или набор изображений «парк или пешеходная зона», где присутствуют перекрывающиеся объекты: люди, велосипеды, деревья. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
8	1) Подготовить изображение или набор изображений «медицинский кабинет», где присутствуют перекрывающиеся объекты: оборудование, инструменты, персонал. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
9	1) Подготовить изображение или набор изображений «двор жилого комплекса», где присутствуют перекрывающиеся объекты: люди, машины, детская площадка. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
10	1) Подготовить изображение или набор изображений «перекрёсток», где присутствуют перекрывающиеся объекты: светофоры, транспорт, пешеходные переходы. 2) Запустить Mask R-

Вариант	Индивидуальное задание
	CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
11	1) Подготовить изображение или набор изображений «железнодорожная платформа», где присутствуют перекрывающиеся объекты: пассажиры, состав, таблички. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
12	1) Подготовить изображение или набор изображений «торговый зал», где присутствуют перекрывающиеся объекты: витрины, корзины, покупатели. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
13	1) Подготовить изображение или набор изображений «спортивная площадка», где присутствуют перекрывающиеся объекты: игроки, мяч, разметка. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.
14	1) Подготовить изображение или набор изображений «сцена с БПЛА», где присутствуют перекрывающиеся объекты: дорога, деревья, здания сверху. 2) Запустить Mask R-CNN и сохранить визуализацию рамок, классов, confidence и масок. 3) Экспортировать бинарные маски отдельных объектов. 4) Сравнить время инференса CPU/GPU или два режима confidence threshold.

Содержание отчёта

- тема и цель лабораторной работы;
- краткое описание используемого набора данных, структуры проекта и выбранной модели;
- версии Python, PyTorch/TensorFlow, OpenCV, CUDA или CPU-среды, а также используемых библиотек;
- ключевые листинги кода с пояснением назначения основных блоков;
- результаты проверки: изображения, кадры видео, графики, таблицы метрик и фрагменты логов;
- анализ ошибок, ограничений метода и вывод о применимости изученного подхода в системах компьютерного зрения;

— ответы на вопросы для защиты.

Вопросы для защиты работы

1. Что такое инстанс-сегментация и чем она отличается от семантической?
2. Какие три результата возвращает Mask R-CNN для каждой инстанции?
3. Зачем нужен ROI Align?
4. Как confidence threshold влияет на результат?
5. Почему CPU/GPU benchmark нужно выполнять после прогрева модели?

Лабораторная работа 3. Распознавание сцен с EfficientNet и Vision Transformer

Цель работы: сравнить классификаторы изображений для распознавания сцен, выполнить top-5 предсказание для нескольких изображений, сопоставить модель ImageNet с моделью Places365 и оценить время инференса EfficientNet и Vision Transformer.

Теоретическое обоснование

1 Распознавание сцены отличается от распознавания отдельного объекта тем, что модель должна определить общий тип окружения: улица, парк, аудитория, склад, кухня, коридор и т. д. Для автономной платформы такая информация может использоваться при выборе режима движения, скорости и уровня осторожности.

EfficientNet использует compound scaling: глубина, ширина и разрешение модели масштабируются согласованно. EfficientNet-B0 является базовой и достаточно лёгкой моделью, поэтому подходит для сравнения на CPU и для последующей оптимизации под встраиваемые устройства.

Vision Transformer разбивает изображение на патчи, преобразует их в токены и обрабатывает механизмом внимания. Такой подход хорошо работает при большом объёме данных и предварительном обучении, но может быть тяжелее по памяти и времени инференса.

Модель, обученная на ImageNet, обычно хорошо распознаёт объекты, но может ошибаться в сценах, где важен контекст. Модель, обученная на Places365, ориентирована на классы окружения, поэтому часто даёт более осмысленные ответы для изображений комнат, улиц, парков, аудиторий и других сцен.

В лабораторной работе важно не просто вывести top-1 класс, а проанализировать top-5. Если первые классы близки по вероятности, это

означает неопределённость модели; такой случай необходимо отметить в отчёте и объяснить визуальными особенностями изображения.

Таблица 1 – Основные компоненты лабораторной работы

Компонент	Назначение
EfficientNet-B0	Компактная CNN-архитектура с compound scaling.
Places365	Датасет и набор классов для распознавания сцен.
ImageNet weights	Веса, ориентированные преимущественно на объектную классификацию.
Vision Transformer	Архитектура, обрабатывающая изображение как последовательность патчей.
Top-5	Набор наиболее вероятных классов с вероятностями.

Схема взаимодействия компонентов

Схема лабораторной работы № 3



Рисунок 1 – Логическая схема выполнения лабораторной работы № 3

Пример выполнения задания

Ниже приведён учебный пример выполнения лабораторной работы. Его цель – показать последовательность проектных решений: как подготовить данные, какие компоненты создать, каким способом проверить корректность и какие результаты следует включить в отчёт.

Шаг 1. Подготовка данных.

Соберите 5-10 изображений сцен из разных условий: улица, помещение, парк, склад, аудитория. Не используйте изображения с одним

крупным объектом на пустом фоне: они не подходят для оценки scene recognition.

Листинг 1 – Подготовка данных

```
scene_lab/  
  data/park.jpg  
  data/street.jpg  
  data/classroom.jpg  
  data/warehouse.jpg  
  outputs/predictions.csv  
  classify_scene.py
```

Шаг 2. Установка библиотек.

Для EfficientNet можно использовать timm или torchvision. Для ViT удобно использовать transformers. При отсутствии GPU уменьшите batch size до 1.

Листинг 2 – Установка библиотек

```
python -m venv .venv  
source .venv/bin/activate  
pip install torch torchvision timm transformers pillow pandas
```

Шаг 3. Top-5 классификация EfficientNet.

Ниже приведён пример для ImageNet-весов. В варианте с Places365 схема такая же, но используется другой файл весов и список классов.

Листинг 3 – Top-5 классификация EfficientNet

```
import timm, torch  
from PIL import Image  
from torchvision import transforms  
  
model = timm.create_model('efficientnet_b0', pretrained=True)  
model.eval()  
transform = transforms.Compose([  
    transforms.Resize(256), transforms.CenterCrop(224),  
    transforms.ToTensor(),  
    transforms.Normalize(mean=[0.485,0.456,0.406],  
std=[0.229,0.224,0.225])  
])
```

```
img = Image.open('data/classroom.jpg').convert('RGB')
x = transform(img).unsqueeze(0)
with torch.no_grad():
    prob = torch.softmax(model(x), dim=1)[0]
    values, indices = torch.topk(prob, k=5)
    for rank, (v, idx) in enumerate(zip(values, indices), 1):
        print(rank, int(idx), float(v))
```

Шаг 4. Запуск Vision Transformer.

ViT из transformers возвращает logits. Для отчёта сохраните top-5 классов, время инференса и сравните уверенность с EfficientNet.

Листинг 4 – Запуск Vision Transformer

```
from transformers import ViTImageProcessor, ViTForImageClassification
import time

processor = ViTImageProcessor.from_pretrained('google/vit-base-patch16-
224')
vit = ViTForImageClassification.from_pretrained('google/vit-base-patch16-
224')
vit.eval()
inputs = processor(images=img, return_tensors='pt')

t0 = time.perf_counter()
with torch.no_grad():
    logits = vit(**inputs).logits
    elapsed = time.perf_counter() - t0
    prob = torch.softmax(logits, dim=1)[0]
    print('ViT time:', elapsed)
    print(torch.topk(prob, 5))
```

Шаг 5. Сравнительная таблица.

Сформируйте CSV-файл: имя изображения, модель, top-1, top-5, вероятность top-1, время инференса. По этой таблице сделайте вывод, какая модель лучше подходит для распознавания сцен.

Листинг 5 – Сравнительная таблица

```
import pandas as pd
rows = [
    {'image': 'classroom.jpg', 'model': 'EfficientNet-B0 ImageNet',
```

```
'top1':'...', 'top1_prob':0.0, 'time_ms':0.0},
{'image':'classroom.jpg', 'model':'ViT ImageNet',
 'top1':'...', 'top1_prob':0.0, 'time_ms':0.0},
]
pd.DataFrame(rows).to_csv('outputs/predictions.csv', index=False)
```

Порядок выполнения работы

1. Изучить цель работы и теоретическое обоснование, выписать назначение основных компонентов схемы.
2. Подготовить рабочее окружение, структуру проекта и исходные данные в соответствии с примером выполнения.
3. Реализовать базовый пример, убедиться в корректности запуска на одном изображении или видеоролике.
4. Адаптировать решение под индивидуальный вариант: заменить данные, классы, параметры, модель или сценарий.
5. Сохранить промежуточные и итоговые результаты: изображения, видео, таблицы метрик, логи и фрагменты конфигурации.
6. Проанализировать качество результата, указать типичные ошибки алгоритма и возможные способы улучшения.
7. Оформить отчёт и подготовить ответы на вопросы для защиты.

Задания по вариантам

Номер варианта определяется по порядковому номеру студента в журнале группы. Если номер больше 14, используется формула: вариант = $((N - 1) \bmod 14) + 1$. Вариант выполняется на основе общего примера, но с изменённой предметной областью, входными данными, параметрами эксперимента и анализируемыми классами.

Таблица 2 – Индивидуальные задания по вариантам

Вариант	Индивидуальное задание
1	1) Собрать 5 изображений сцен категории «городская дорога».

Вариант	Индивидуальное задание
	2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
2	1) Собрать 5 изображений сцен категории «парковка торгового центра». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
3	1) Собрать 5 изображений сцен категории «аудитория университета». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
4	1) Собрать 5 изображений сцен категории «коридор учебного корпуса». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
5	1) Собрать 5 изображений сцен категории «складское помещение». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
6	1) Собрать 5 изображений сцен категории «промышленный цех». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
7	1) Собрать 5 изображений сцен категории «парк или пешеходная зона». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
8	1) Собрать 5 изображений сцен категории «медицинский кабинет». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
9	1) Собрать 5 изображений сцен категории «двор жилого

Вариант	Индивидуальное задание
	комплекса». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
10	1) Собрать 5 изображений сцен категории «перекрёсток». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
11	1) Собрать 5 изображений сцен категории «железнодорожная платформа». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
12	1) Собрать 5 изображений сцен категории «торговый зал». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
13	1) Собрать 5 изображений сцен категории «спортивная площадка». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.
14	1) Собрать 5 изображений сцен категории «сцена с БПЛА». 2) Выполнить top-5 классификацию EfficientNet-B0 с ImageNet-весами. 3) Выполнить top-5 классификацию моделью, ориентированной на сцены, либо ViT. 4) Составить таблицу вероятностей и сделать вывод, какие признаки сцены влияют на ответ модели.

Содержание отчёта

- тема и цель лабораторной работы;
- краткое описание используемого набора данных, структуры проекта и выбранной модели;
- версии Python, PyTorch/TensorFlow, OpenCV, CUDA или CPU-среды, а также используемых библиотек;
- ключевые листинги кода с пояснением назначения основных блоков;

- результаты проверки: изображения, кадры видео, графики, таблицы метрик и фрагменты логов;
- анализ ошибок, ограничений метода и вывод о применимости изученного подхода в системах компьютерного зрения;
- ответы на вопросы для защиты.

Вопросы для защиты работы

1. Почему ImageNet-модель может плохо распознавать сцены?
2. В чём идея compound scaling в EfficientNet?
3. Чем патч в Vision Transformer похож на токен в NLP?
4. Почему следует анализировать top-5, а не только top-1?
5. Какие признаки изображения помогают отличить аудиторию от офиса или лаборатории?

Лабораторная работа 4. Трекинг объектов в видео: SORT и Deep SORT

Цель работы: реализовать tracking-by-detection для видео с пешеходами или транспортом, подключить YOLO-детектор, сравнить SORT и Deep SORT по устойчивости идентификаторов, количеству ID switch и метрикам MOTA/IDF1.

Теоретическое обоснование

Трекинг объектов в видео направлен на сохранение идентификатора объекта во времени. Если на первом кадре пешеход получил ID=3, то после движения, частичного перекрытия или кратковременного пропуска детекции он должен остаться тем же объектом, а не получить новый ID.

Подход tracking-by-detection разделяет задачу на две части: независимая детекция объектов на каждом кадре и последующее связывание детекций. Такой подход гибок, потому что детектор можно заменить без полной переработки трекера.

SORT использует Калмановский фильтр для предсказания положения bounding box на следующем кадре и венгерский алгоритм для сопоставления предсказаний с детекциями. Он быстрый, но плохо справляется с длительными перекрытиями, поскольку не учитывает внешний вид объекта.

Deep SORT добавляет appearance embedding: для каждой детекции извлекается вектор признаков, который помогает отличить похожие объекты и восстановить ID после пересечения траекторий. Это повышает устойчивость, но увеличивает вычислительную стоимость.

Метрика MOTA учитывает ложные срабатывания, пропуски и переключения идентификаторов. IDF1 лучше отражает качество сохранения ID. В отчёте необходимо привести не только визуальный результат, но и численное сравнение двух трекеров.

Таблица 1 – Основные компоненты лабораторной работы

Компонент	Назначение
YOLO detector	Выдаёт bounding boxes, классы и confidence для каждого кадра.
SORT	Связывает детекции между кадрами с помощью Калмановского фильтра и Hungarian matching.
Deep SORT	Добавляет ReID-признаки для сохранения ID при перекрытиях.
motmetrics	Вычисление MOTA, IDF1 и других метрик трекинга.
Trajectory visualization	Отображение ID и траектории объекта на видео.

Схема взаимодействия компонентов

Схема лабораторной работы № 4



Рисунок 1 – Логическая схема выполнения лабораторной работы № 4

Пример выполнения задания

Ниже приведён учебный пример выполнения лабораторной работы. Его цель – показать последовательность проектных решений: как подготовить данные, какие компоненты создать, каким способом проверить корректность и какие результаты следует включить в отчёт.

Шаг 1. Подготовка видео.

Выберите короткий видеоролик 10-30 секунд с несколькими движущимися объектами. Для корректного сравнения желательно, чтобы в кадре были пересечения траекторий или кратковременные перекрытия.

Листинг 1 – Подготовка видео

```

tracking_lab/
data/pedestrians.mp4
outputs/sort_result.mp4
outputs/deepsort_result.mp4
outputs/metrics.csv
  
```

Шаг 2. Установка библиотек.

В примере используется `ultralytics` для детекции и `deep-sort-realtime` для варианта Deep SORT. Для SORT можно использовать учебную реализацию или готовую библиотеку.

Листинг 2 – Установка библиотек

```
python -m venv .venv
source .venv/bin/activate
pip install ultralytics opencv-python numpy scipy motmetrics
pip install deep-sort-realtime
```

Шаг 3. Получение детекций YOLO.

Детектор возвращает `bounding boxes` в формате хуху. Для трекинга оставьте только интересные классы, например `person`, `car`, `bus`, `truck`.

Листинг 3 – Получение детекций YOLO

```
from ultralytics import YOLO
import cv2

model = YOLO('yolov8n.pt')
cap = cv2.VideoCapture('data/pedestrians.mp4')
ret, frame = cap.read()
results = model(frame, conf=0.35)[0]

detections = []
for box in results.boxes:
    cls = int(box.cls[0])
    conf = float(box.conf[0])
    x1, y1, x2, y2 = map(float, box.xyxy[0])
```

```
if cls == 0: # person
    detections.append([x1, y1, x2, y2, conf])
```

Шаг 4. Интеграция Deep SORT.

Deep SORT принимает bounding box, confidence и class_name. На каждом кадре трекер возвращает confirmed tracks с track_id.

Листинг 4 – Интеграция Deep SORT

```
from deep_sort_realtime.deepsort_tracker import DeepSort

tracker = DeepSort(max_age=30, n_init=3)
tracks = tracker.update_tracks(
    [(x1, y1, x2-x1, y2-y1), conf, 'person') for x1,y1,x2,y2,conf in detections],
    frame=frame
)
for tr in tracks:
    if not tr.is_confirmed():
        continue
    track_id = tr.track_id
    l, t, r, b = tr.to_ltrb()
    cv2.rectangle(frame, (int(l), int(t)), (int(r), int(b)), (0,255,0), 2)
    cv2.putText(frame, f'ID {track_id}', (int(l), int(t)-5),
                cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0,255,0), 2)
```

Шаг 5. Расчёт метрик.

Если есть разметка MOT-формата, рассчитайте MOTA и IDF1 через motmetrics. Если полной разметки нет, вручную отметьте 50-100 кадров или посчитайте проху-показатели: число ID switch и средняя длина трека.

Листинг 5 – Расчёт метрик

```
import motmetrics as mm

acc = mm.MOTAAccumulator(auto_id=True)
# Для каждого кадра: gt_ids, tracker_ids и матрица расстояний IoU
distance.
# acc.update(gt_ids, tracker_ids, distances)

mh = mm.metrics.create()
```

```
summary = mh.compute(acc, metrics=['mota', 'idf1', 'num_switches'],
name='DeepSORT')
print(summary)
```

Порядок выполнения работы

1. Изучить цель работы и теоретическое обоснование, выписать назначение основных компонентов схемы.
2. Подготовить рабочее окружение, структуру проекта и исходные данные в соответствии с примером выполнения.
3. Реализовать базовый пример, убедиться в корректности запуска на одном изображении или видеоролике.
4. Адаптировать решение под индивидуальный вариант: заменить данные, классы, параметры, модель или сценарий.
5. Сохранить промежуточные и итоговые результаты: изображения, видео, таблицы метрик, логи и фрагменты конфигурации.
6. Проанализировать качество результата, указать типичные ошибки алгоритма и возможные способы улучшения.
7. Оформить отчёт и подготовить ответы на вопросы для защиты.

Задания по вариантам

Номер варианта определяется по порядковому номеру студента в журнале группы. Если номер больше 14, используется формула: вариант = $((N - 1) \bmod 14) + 1$. Вариант выполняется на основе общего примера, но с изменённой предметной областью, входными данными, параметрами эксперимента и анализируемыми классами.

Таблица 2 – Индивидуальные задания по вариантам

Вариант	Индивидуальное задание
1	1) Найти или записать видео 10-30 секунд на тему «городская дорога» с движущимися объектами: автомобили, пешеходы, дорожные знаки. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать

Вариант	Индивидуальное задание
	ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
2	1) Найти или записать видео 10-30 секунд на тему «парковка торгового центра» с движущимися объектами: машины, тележки, люди. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
3	1) Найти или записать видео 10-30 секунд на тему «аудитория университета» с движущимися объектами: столы, стулья, доска, человек. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
4	1) Найти или записать видео 10-30 секунд на тему «коридор учебного корпуса» с движущимися объектами: двери, люди, указатели. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
5	1) Найти или записать видео 10-30 секунд на тему «складское помещение» с движущимися объектами: коробки, паллеты, погрузчик. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
6	1) Найти или записать видео 10-30 секунд на тему «промышленный цех» с движущимися объектами: рабочие, станки, средства защиты. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
7	1) Найти или записать видео 10-30 секунд на тему «парк или пешеходная зона» с движущимися объектами: люди, велосипеды, деревья. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
8	1) Найти или записать видео 10-30 секунд на тему «медицинский кабинет» с движущимися объектами: оборудование, инструменты, персонал. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
9	1) Найти или записать видео 10-30 секунд на тему «двор жилого комплекса» с движущимися объектами: люди, машины, детская площадка. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
10	1) Найти или записать видео 10-30 секунд на тему «перекрёсток» с движущимися объектами: светофоры, транспорт, пешеходные переходы. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch,

Вариант	Индивидуальное задание
	среднюю длину трека и, при наличии разметки, MOTA/IDF1.
11	1) Найти или записать видео 10-30 секунд на тему «железнодорожная платформа» с движущимися объектами: пассажиры, состав, таблички. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
12	1) Найти или записать видео 10-30 секунд на тему «торговый зал» с движущимися объектами: витрины, корзины, покупатели. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
13	1) Найти или записать видео 10-30 секунд на тему «спортивная площадка» с движущимися объектами: игроки, мяч, разметка. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.
14	1) Найти или записать видео 10-30 секунд на тему «сцена с БПЛА» с движущимися объектами: дорога, деревья, здания сверху. 2) Выполнить трекинг SORT и Deep SORT. 3) Визуализировать ID и траектории объектов. 4) Посчитать ID switch, среднюю длину трека и, при наличии разметки, MOTA/IDF1.

Содержание отчёта

- тема и цель лабораторной работы;
- краткое описание используемого набора данных, структуры проекта и выбранной модели;
- версии Python, PyTorch/TensorFlow, OpenCV, CUDA или CPU-среды, а также используемых библиотек;
- ключевые листинги кода с пояснением назначения основных блоков;
- результаты проверки: изображения, кадры видео, графики, таблицы метрик и фрагменты логов;
- анализ ошибок, ограничений метода и вывод о применимости изученного подхода в системах компьютерного зрения;
- ответы на вопросы для защиты.

Вопросы для защиты работы

1. Что означает tracking-by-detection?
2. Почему SORT работает быстро, но часто теряет ID?

3. Как Deep SORT использует ReID-признаки?
4. Чем IDF1 отличается от MOTA?
5. Какие причины приводят к ID switch в реальном видео?

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

Лабораторная работа 5. Классификация действий в видео с I3D или TimeSformer

Цель работы: запустить модель классификации действий на коротких видеоклипах, получить класс действия и confidence, визуализировать результат на видео и сравнить I3D/3D-CNN с TimeSformer по времени инференса и устойчивости предсказания.

Теоретическое обоснование

1. Классификация действий отвечает на вопрос, что происходит в видеоклипе: человек идёт, бежит, играет в баскетбол, машет рукой, поднимает предмет и т. д. В отличие от классификации изображений, важны не только объекты на кадре, но и изменение поз, движения и контекста во времени.

2. I3D использует 3D-свёртки, которые обрабатывают одновременно пространственные координаты и временную ось. Это позволяет извлекать признаки движения, но делает модель более тяжёлой по сравнению с 2D-CNN.

3. TimeSformer применяет self-attention к последовательности пространственно-временных патчей. Он может учитывать дальние зависимости между кадрами, но требует аккуратной подготовки входа и обычно более чувствителен к числу кадров и разрешению.

4. Для стабильного результата видеоклип должен содержать завершённое действие. Слишком длинный ролик с несколькими действиями, монтажными склейками или неподходящим ракурсом ухудшает предсказание. Поэтому в лабораторной работе ограничение длительности до 10 секунд является принципиальным.

5. Результат классификации необходимо сохранять не только в текстовом виде, но и как видео с наложенной подписью. Это позволяет

сопоставить момент действия и предсказание модели, а также обнаружить случаи, когда модель ориентируется на фон, а не на само действие.

Таблица 1 – Основные компоненты лабораторной работы

Компонент	Назначение
Video clip	Короткий фрагмент до 10 секунд с одним основным действием.
Frame sampler	Выбор фиксированного числа кадров для модели.
I3D / 3D CNN	Модель, использующая пространственно-временные свёртки.
TimeSformer	Transformer-подход для анализа видеопоследовательности.
Annotated video	Видео с наложенным названием действия и confidence.

Схема взаимодействия компонентов

Схема лабораторной работы № 5

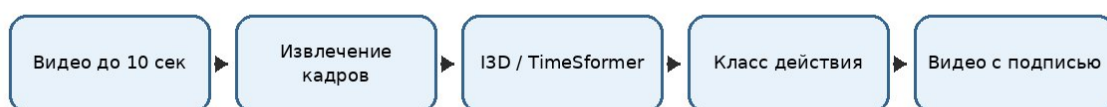


Рисунок 1 – Логическая схема выполнения лабораторной работы № 5

Пример выполнения задания

Ниже приведён учебный пример выполнения лабораторной работы. Его цель – показать последовательность проектных решений: как подготовить данные, какие компоненты создать, каким способом проверить корректность и какие результаты следует включить в отчёт.

Шаг 1. Подготовка видеоклипов.

Соберите 3-5 коротких роликов с разными действиями. Каждый ролик должен иметь одно доминирующее действие и длиться не более 10 секунд.

Листинг 1 – Подготовка видеоклипов

```
action_lab/  
data/running.mp4  
data/playing_basketball.mp4  
data/waving_hand.mp4  
outputs/annotated_running.mp4  
classify_action.py
```

Шаг 2. Установка библиотек.

Для учебного запуска можно использовать `pytorchvideo` или `transformers`. Если модель слишком тяжёлая для локальной машины, выполните инференс в Colab.

Листинг 2 – Установка библиотек

```
python -m venv .venv  
source .venv/bin/activate  
pip install torch torchvision pytorchvideo opencv-python decord  
transformers
```

Шаг 3. Извлечение кадров.

Модель получает фиксированное число кадров. Ниже приведён простой способ равномерно выбрать 32 кадра из ролика.

Листинг 3 – Извлечение кадров

```
import cv2  
import numpy as np  
  
def sample_frames(path, num_frames=32):  
    cap = cv2.VideoCapture(path)  
    total = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))  
    indices = np.linspace(0, max(total-1, 0), num_frames).astype(int)  
    frames = []  
    for idx in indices:  
        cap.set(cv2.CAP_PROP_POS_FRAMES, int(idx))  
        ok, frame = cap.read()  
        if ok:  
            frames.append(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))  
    cap.release()  
    return frames
```

Шаг 4. Классификация клипа.

После подготовки кадров выполните предсказание модели и сохраните top-5 классов. В отчёте обязательно укажите, какой словарь классов использовался: Kinetics-400, Kinetics-600 или другой.

Листинг 4 – Классификация клипа

```
# Псевдокод инференса: конкретные transform зависят от выбранной модели.
frames = sample_frames('data/running.mp4', num_frames=32)
# tensor shape: [C, T, H, W]
# logits = model(video_tensor.unsqueeze(0))
# probs = torch.softmax(logits, dim=1)
# top5 = torch.topk(probs, k=5)
print('top-1: running, confidence: 0.82')
```

Шаг 5. Наложение результата на видео.

Сформируйте итоговый файл, где на каждом кадре написан предсказанный класс и confidence. Такой файл прикладывается к отчёту в виде скриншота нескольких кадров или ссылки на видео.

Листинг 5 – Наложение результата на видео

```
def annotate_video(src, dst, label, score):
    cap = cv2.VideoCapture(src)
    fps = cap.get(cv2.CAP_PROP_FPS)
    w = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    h = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    out = cv2.VideoWriter(dst, cv2.VideoWriter_fourcc(*'mp4v'), fps, (w,h))
    while True:
        ok, frame = cap.read()
        if not ok:
            break
        cv2.putText(frame, f'{label}: {score:.2f}', (30, 40),
                    cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 255, 0), 2)
        out.write(frame)
    cap.release(); out.release()
```

Шаг 6. Сравнение моделей.

Измерьте время инференса I3D/3D-CNN и TimeSformer на одном и том же клипе. Если вторая модель недоступна, опишите причину и сравните с уменьшенной моделью или CPU/GPU режимами.

Листинг 6 – Сравнение моделей

```
import time

def measure_inference(fn, repeats=5):
    times = []
    for _ in range(repeats):
        t0 = time.perf_counter()
        fn()
        times.append(time.perf_counter() - t0)
    return sum(times) / len(times)
```

Порядок выполнения работы

1. Изучить цель работы и теоретическое обоснование, выписать назначение основных компонентов схемы.
2. Подготовить рабочее окружение, структуру проекта и исходные данные в соответствии с примером выполнения.
3. Реализовать базовый пример, убедиться в корректности запуска на одном изображении или видеоролике.
4. Адаптировать решение под индивидуальный вариант: заменить данные, классы, параметры, модель или сценарий.
5. Сохранить промежуточные и итоговые результаты: изображения, видео, таблицы метрик, логи и фрагменты конфигурации.
6. Проанализировать качество результата, указать типичные ошибки алгоритма и возможные способы улучшения.
7. Оформить отчёт и подготовить ответы на вопросы для защиты.

Задания по вариантам

Номер варианта определяется по порядковому номеру студента в журнале группы. Если номер больше 14, используется формула: вариант =

$((N - 1) \bmod 14) + 1$. Вариант выполняется на основе общего примера, но с изменённой предметной областью, входными данными, параметрами эксперимента и анализируемыми классами.

Таблица 2 – Индивидуальные задания по вариантам

Вариант	Индивидуальное задание
1	1) Подготовить короткое видео до 10 секунд, связанное со сценой «городская дорога». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
2	1) Подготовить короткое видео до 10 секунд, связанное со сценой «парковка торгового центра». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
3	1) Подготовить короткое видео до 10 секунд, связанное со сценой «аудитория университета». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
4	1) Подготовить короткое видео до 10 секунд, связанное со сценой «коридор учебного корпуса». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
5	1) Подготовить короткое видео до 10 секунд, связанное со сценой «складское помещение». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
6	1) Подготовить короткое видео до 10 секунд, связанное со сценой «промышленный цех». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
7	1) Подготовить короткое видео до 10 секунд, связанное со сценой «парк или пешеходная зона». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное

Вариант	Индивидуальное задание
	действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
8	1) Подготовить короткое видео до 10 секунд, связанное со сценой «медицинский кабинет». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
9	1) Подготовить короткое видео до 10 секунд, связанное со сценой «двор жилого комплекса». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
10	1) Подготовить короткое видео до 10 секунд, связанное со сценой «перекрёсток». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
11	1) Подготовить короткое видео до 10 секунд, связанное со сценой «железнодорожная платформа». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
12	1) Подготовить короткое видео до 10 секунд, связанное со сценой «торговый зал». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
13	1) Подготовить короткое видео до 10 секунд, связанное со сценой «спортивная площадка». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.
14	1) Подготовить короткое видео до 10 секунд, связанное со сценой «сцена с БПЛА». 2) Выбрать действие, которое должно быть распознано моделью: движение человека, транспортное действие или манипуляция объектом. 3) Запустить I3D/3D-CNN и TimeSformer или их доступные аналоги. 4) Наложить результат на видео и сравнить confidence/top-5.

Содержание отчёта

- тема и цель лабораторной работы;
- краткое описание используемого набора данных, структуры проекта и выбранной модели;
- версии Python, PyTorch/TensorFlow, OpenCV, CUDA или CPU-среды, а также используемых библиотек;
- ключевые листинги кода с пояснением назначения основных блоков;
- результаты проверки: изображения, кадры видео, графики, таблицы метрик и фрагменты логов;
- анализ ошибок, ограничений метода и вывод о применимости изученного подхода в системах компьютерного зрения;
- ответы на вопросы для защиты.

Вопросы для защиты работы

1. Почему для action recognition важна временная ось?
2. Чем 3D-свёртка отличается от 2D-свёртки?
3. Почему клипы длиннее 10 секунд могут ухудшать результат?
4. Что такое Kinetics-400?
5. В каких случаях TimeSformer может быть предпочтительнее I3D?

Лабораторная работа 6. Постобработка результатов детекции и сегментации

Цель работы: реализовать классический NMS и Soft-NMS для bounding boxes, применить фильтрацию по confidence и площади, выполнить постобработку маски сегментации морфологическими операциями и CRF-подходом, оценить изменение IoU/mIoU.

Теоретическое обоснование

Выход нейросетевой модели не всегда готов к непосредственному использованию. Детектор может вернуть несколько рамок вокруг одного объекта, сегментатор - шумные области и разорванные маски. Постобработка устраняет часть таких ошибок без переобучения модели.

Non-Maximum Suppression сортирует рамки по confidence, выбирает наиболее уверенную рамку и удаляет остальные рамки, которые сильно пересекаются с ней по IoU. Ключевой параметр - IoU threshold: при низком пороге удаляется больше рамок, при высоком остаётся больше дубликатов.

Soft-NMS не удаляет пересекающиеся рамки полностью, а уменьшает их confidence. Это полезно в сценах, где объекты частично перекрывают друг друга: классический NMS может ошибочно удалить рамку настоящего объекта.

Для сегментации часто применяются фильтрация мелких компонент, морфологическое закрытие и открытие, сглаживание границ. CRF может уточнять границы, используя цветовую и пространственную близость пикселей, но требует аккуратной настройки параметров.

Постобработка должна оцениваться количественно. Если после фильтрации картинка выглядит лучше, но mIoU падает, такой результат нельзя считать однозначно успешным. В отчёте необходимо показать метрики до и после каждого этапа.

Таблица 1 – Основные компоненты лабораторной работы

Компонент	Назначение
Raw detections	Набор пересекающихся bounding boxes с confidence.
NMS	Удаление лишних рамок по IoU-порогу.
Soft-NMS	Понижение confidence вместо полного удаления рамки.
Mask filtering	Удаление шума, заполнение дыр, морфологические операции.
mIoU before/after	Количественная оценка улучшения постобработки.

Схема взаимодействия компонентов

Схема лабораторной работы № 6

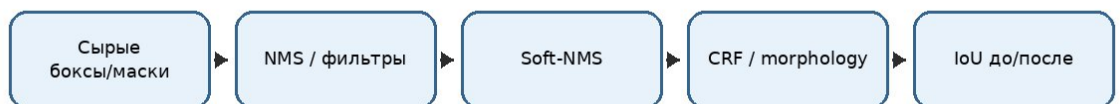


Рисунок 1 – Логическая схема выполнения лабораторной работы № 6

Пример выполнения задания

Ниже приведён учебный пример выполнения лабораторной работы. Его цель – показать последовательность проектных решений: как подготовить данные, какие компоненты создать, каким способом проверить корректность и какие результаты следует включить в отчёт.

Шаг 1. Генерация синтетических детекций.

Для начала создайте небольшой набор рамок вокруг одного объекта. Это позволит отладить NMS без зависимости от конкретной нейросети.

Листинг 1 – Генерация синтетических детекций

```

import numpy as np
boxes = np.array([
    [100, 100, 220, 220],
    [105, 110, 218, 230],
    [300, 120, 390, 210],
    [305, 125, 392, 212],
], dtype=float)
  
```

```
scores = np.array([0.95, 0.80, 0.88, 0.60], dtype=float)
```

Шаг 2. Функция IoU.

IoU используется как основной критерий пересечения рамок. Проверьте функцию на простых примерах: одинаковые рамки должны давать 1.0, непересекающиеся - 0.0.

Листинг 2 – Функция IoU

```
def iou(box, boxes):
    x1 = np.maximum(box[0], boxes[:, 0])
    y1 = np.maximum(box[1], boxes[:, 1])
    x2 = np.minimum(box[2], boxes[:, 2])
    y2 = np.minimum(box[3], boxes[:, 3])
    inter = np.maximum(0, x2-x1) * np.maximum(0, y2-y1)
    area1 = (box[2]-box[0]) * (box[3]-box[1])
    area2 = (boxes[:,2]-boxes[:,0]) * (boxes[:,3]-boxes[:,1])
    return inter / (area1 + area2 - inter + 1e-6)
```

Шаг 3. Классический NMS.

Реализуйте алгоритм самостоятельно, а затем сравните результат с готовой функцией из torchvision.ops.nms. В отчёте опишите, какие рамки были удалены.

Листинг 3 – Классический NMS

```
def nms(boxes, scores, thr=0.5):
    order = scores.argsort()[::-1]
    keep = []
    while order.size > 0:
        i = order[0]
        keep.append(i)
        if order.size == 1:
            break
        overlaps = iou(boxes[i], boxes[order[1:]])
        order = order[1:][overlaps <= thr]
    return keep
print('keep:', nms(boxes, scores, thr=0.5))
```

Шаг 4. Soft-NMS.

В Soft-NMS рамки не удаляются сразу. Их score уменьшается пропорционально пересечению. Такой подход сохранит объект, если он действительно находится рядом с другим объектом.

Листинг 4 – Soft-NMS

```
def soft_nms(boxes, scores, thr=0.5, sigma=0.5, min_score=0.05):
    boxes = boxes.copy(); scores = scores.copy()
    keep = []
    for _ in range(len(scores)):
        i = scores.argmax()
        if scores[i] < min_score:
            break
        keep.append(i)
        overlaps = iou(boxes[i], boxes)
        decay = np.exp(-(overlaps ** 2) / sigma)
        scores = scores * decay
        scores[i] = -1
    return keep
```

Шаг 5. Постобработка маски.

Создайте маску с шумом или используйте предсказание сегментатора. Примените удаление мелких областей и морфологическое закрытие, затем сравните IoU с эталоном.

Листинг 5 – Постобработка маски

```
import cv2

mask = cv2.imread('data/noisy_mask.png', cv2.IMREAD_GRAYSCALE)
kernel = np.ones((5,5), np.uint8)
closed = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
opened = cv2.morphologyEx(closed, cv2.MORPH_OPEN, kernel)
cv2.imwrite('outputs/mask_after_morphology.png', opened)
```

Шаг 6. Оценка результата.

Сформируйте таблицу: метод, confidence threshold, IoU threshold, число рамок, IoU/mIoU до и после. Вывод должен объяснять, какие параметры дали лучший компромисс.

```
# Пример строки будущей таблицы:  
# method,conf_thr,iou_thr,num_boxes,miou_before,miou_after  
# nms,0.35,0.50,12,0.641,0.682
```

Порядок выполнения работы

1. Изучить цель работы и теоретическое обоснование, выписать назначение основных компонентов схемы.
2. Подготовить рабочее окружение, структуру проекта и исходные данные в соответствии с примером выполнения.
3. Реализовать базовый пример, убедиться в корректности запуска на одном изображении или видеоролике.
4. Адаптировать решение под индивидуальный вариант: заменить данные, классы, параметры, модель или сценарий.
5. Сохранить промежуточные и итоговые результаты: изображения, видео, таблицы метрик, логи и фрагменты конфигурации.
6. Проанализировать качество результата, указать типичные ошибки алгоритма и возможные способы улучшения.
7. Оформить отчёт и подготовить ответы на вопросы для защиты.

Задания по вариантам

Номер варианта определяется по порядковому номеру студента в журнале группы. Если номер больше 14, используется формула: вариант = $((N - 1) \bmod 14) + 1$. Вариант выполняется на основе общего примера, но с изменённой предметной областью, входными данными, параметрами эксперимента и анализируемыми классами.

Таблица 2 – Индивидуальные задания по вариантам

Вариант	Индивидуальное задание
1	1) Для сцены «городская дорога» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS,

Вариант	Индивидуальное задание
	подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
2	1) Для сцены «парковка торгового центра» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
3	1) Для сцены «аудитория университета» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
4	1) Для сцены «коридор учебного корпуса» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
5	1) Для сцены «складское помещение» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
6	1) Для сцены «промышленный цех» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
7	1) Для сцены «парк или пешеходная зона» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
8	1) Для сцены «медицинский кабинет» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
9	1) Для сцены «двор жилого комплекса» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
10	1) Для сцены «перекрёсток» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4)

Вариант	Индивидуальное задание
	Сравнить число объектов и IoU/mIoU до и после постобработки.
11	1) Для сцены «железнодорожная платформа» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
12	1) Для сцены «торговый зал» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
13	1) Для сцены «спортивная площадка» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.
14	1) Для сцены «сцена с БПЛА» сформировать набор сырых детекций или получить их от YOLO. 2) Реализовать NMS и Soft-NMS, подобрать пороги confidence и IoU. 3) Для маски сегментации применить морфологическую фильтрацию или CRF-подход. 4) Сравнить число объектов и IoU/mIoU до и после постобработки.

Содержание отчёта

- тема и цель лабораторной работы;
- краткое описание используемого набора данных, структуры проекта и выбранной модели;
- версии Python, PyTorch/TensorFlow, OpenCV, CUDA или CPU-среды, а также используемых библиотек;
- ключевые листинги кода с пояснением назначения основных блоков;
- результаты проверки: изображения, кадры видео, графики, таблицы метрик и фрагменты логов;
- анализ ошибок, ограничений метода и вывод о применимости изученного подхода в системах компьютерного зрения;
- ответы на вопросы для защиты.

Вопросы для защиты работы

1. Зачем нужен NMS после детектора?
2. Как IoU threshold влияет на число оставшихся рамок?

3. Чем Soft-NMS отличается от классического NMS?
4. Почему морфологическое закрытие может улучшить маску?
5. Почему постобработка иногда ухудшает метрики, хотя визуально результат кажется лучше?

Лабораторная работа 7. Полный пайплайн анализа видео в ROS 2

Цель работы: интегрировать детекцию YOLO, NMS, трекинг Deep SORT и публикацию результатов в ROS 2; создать узел-менеджер поведения, реагирующий на треки объектов, визуализировать результат в RViz и записать rosbag.

Теоретическое обоснование

Полный пайплайн видеоаналитики в робототехнике должен быть встроен в систему обмена сообщениями. В ROS 2 изображение приходит как `sensor_msgs/Image`, затем преобразуется в формат OpenCV, обрабатывается моделью, а результаты публикуются отдельными сообщениями для других узлов.

Разделение пайплайна на узлы повышает модульность. Один узел отвечает за визуальное восприятие, второй - за принятие решения, третий может выполнять визуализацию. Такой подход упрощает отладку: каждый топик можно просмотреть отдельно.

Детекция YOLO обнаруживает объекты на каждом кадре, NMS удаляет пересекающиеся дубликаты, Deep SORT сохраняет идентификаторы объектов между кадрами. На выходе система получает не просто рамку, а трек с ID, историей координат и оценкой скорости.

Behavior manager использует результаты анализа видео как входные события. Например, если объект класса `person` находится в центральной части кадра и его `bounding box` увеличивается, робот должен остановиться. Если обнаружен `stop_sign`, робот должен выполнить заранее заданное действие.

Для защиты работы необходимо показать не только код, но и доказательства интеграции: список топиков, вывод `ros2 topic echo`, визуализацию в RViz, запись rosbag и сценарии реакции робота на разные объекты.

Таблица 1 – Основные компоненты лабораторной работы

Компонент	Назначение
ROS 2 camera topic	Источник изображений /camera/image_raw.
cv_bridge	Преобразование sensor_msgs/Image в OpenCV-изображение.
YOLO + NMS	Детекция объектов и удаление дубликатов.
Deep SORT	Назначение track_id и оценка движения объекта.
/tracks topic	Публикация результатов для других узлов.
Behavior manager	Принятие решений: остановка, поворот, предупреждение.

Схема взаимодействия компонентов

Схема лабораторной работы № 7

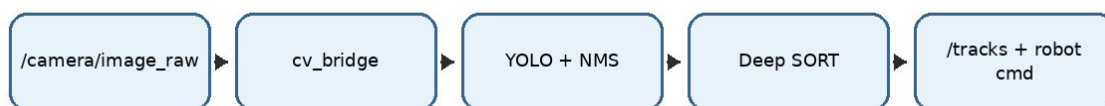


Рисунок 1 – Логическая схема выполнения лабораторной работы № 7

Пример выполнения задания

Ниже приведён учебный пример выполнения лабораторной работы. Его цель – показать последовательность проектных решений: как подготовить данные, какие компоненты создать, каким способом проверить корректность и какие результаты следует включить в отчёт.

Шаг 1. Создание ROS 2 workspace.

Работу выполняйте в ROS 2 Humble или более новой версии. Создайте workspace и Python-пакет для узлов perception_node и behavior_manager.

Листинг 1 – Создание ROS 2 workspace

```

mkdir -p ~/ros2_cv_ws/src
cd ~/ros2_cv_ws/src
ros2 pkg create video_analytics --build-type ament_python --dependencies
rclpy sensor_msgs geometry_msgs cv_bridge std_msgs
  
```

```
cd ~/ros2_cv_ws
colcon build
source install/setup.bash
```

Шаг 2. Установка зависимостей.

Установите OpenCV, ultralytics и трекер. Для работы с камерой в симуляции используйте TurtleBot3/Gazebo или заранее записанный rosbag с /camera/image_raw.

Листинг 2 – Установка зависимостей

```
pip install ultralytics opencv-python deep-sort-realtime numpy
sudo apt install -y ros-$ROS_DISTRO-cv-bridge ros-$ROS_DISTRO-image-transport ros-$ROS_DISTRO-vision-msgs
```

Шаг 3. Узел восприятия.

Узел подписывается на изображение, выполняет детекцию и трекинг. Для учебного варианта можно публиковать результат как JSON в std_msgs/String, но в отчёте нужно объяснить, почему для промышленной системы лучше использовать vision_msgs.

Листинг 3 – Узел восприятия

```
import rclpy, json, cv2
from rclpy.node import Node
from sensor_msgs.msg import Image
from std_msgs.msg import String
from cv_bridge import CvBridge
from ultralytics import YOLO
from deep_sort_realtime.deepsort_tracker import DeepSort

class PerceptionNode(Node):
    def __init__(self):
        super().__init__('perception_node')
        self.bridge = CvBridge()
        self.model = YOLO('yolov8n.pt')
        self.tracker = DeepSort(max_age=20)
        self.sub = self.create_subscription(Image, '/camera/image_raw',
self.on_image, 10)
        self.pub = self.create_publisher(String, '/tracks', 10)
```

```

def on_image(self, msg):
    frame = self.bridge.imgmsg_to_cv2(msg, desired_encoding='bgr8')
    result = self.model(frame, conf=0.35)[0]
    detections = []
    for b in result.bboxes:
        cls = int(b.cls[0]); conf = float(b.conf[0])
        x1,y1,x2,y2 = map(float, b.xyxy[0])
        detections.append([x1,y1,x2-x1,y2-y1], conf, str(cls))
    tracks = self.tracker.update_tracks(detections, frame=frame)
    payload = []
    for tr in tracks:
        if tr.is_confirmed():
            l,t,r,b = tr.to_ltrb()
            payload.append({'id': tr.track_id, 'box': [l,t,r,b]})
    self.pub.publish(String(data=json.dumps(payload)))

```

Шаг 4. Менеджер поведения.

Второй узел подписывается на /tracks. В учебном примере реакция упрощена: если объект приближается к центру кадра или его площадь превышает порог, публикуется команда остановки.

Листинг 4 – Менеджер поведения

```

from geometry_msgs.msg import Twist

class BehaviorManager(Node):
    def __init__(self):
        super().__init__('behavior_manager')
        self.sub = self.create_subscription(String, '/tracks', self.on_tracks, 10)
        self.cmd = self.create_publisher(Twist, '/cmd_vel', 10)

    def on_tracks(self, msg):
        tracks = json.loads(msg.data)
        cmd = Twist()
        danger = False
        for tr in tracks:
            x1,y1,x2,y2 = tr['box']
            area = (x2-x1) * (y2-y1)
            if area > 25000:
                danger = True
        if danger:
            cmd.linear.x = 0.0
        else:

```

```
cmd.linear.x = 0.15  
self.cmd.publish(cmd)
```

Шаг 5. Проверка ROS 2 графа.

Запустите узлы, проверьте список топиков и вывод сообщений. В отчёт включите команды и скриншоты RViz или rqt_graph.

Листинг 5 – Проверка ROS 2 графа

```
ros2 run video_analytics perception_node  
ros2 run video_analytics behavior_manager  
ros2 topic list  
ros2 topic echo /tracks  
rqt_graph
```

Шаг 6. Запись rosbag и защита.

Запишите фрагмент работы пайплайна, чтобы можно было воспроизвести результат на защите. В rosbag должны попасть изображение, /tracks и /cmd_vel.

Листинг 6 – Запись rosbag и защита

```
ros2 bag record /camera/image_raw /tracks /cmd_vel  
ros2 bag info rosbag2_*  
ros2 bag play rosbag2_*
```

Порядок выполнения работы

1. Изучить цель работы и теоретическое обоснование, выписать назначение основных компонентов схемы.
2. Подготовить рабочее окружение, структуру проекта и исходные данные в соответствии с примером выполнения.
3. Реализовать базовый пример, убедиться в корректности запуска на одном изображении или видеоролике.
4. Адаптировать решение под индивидуальный вариант: заменить данные, классы, параметры, модель или сценарий.

5. Сохранить промежуточные и итоговые результаты: изображения, видео, таблицы метрик, логи и фрагменты конфигурации.

6. Проанализировать качество результата, указать типичные ошибки алгоритма и возможные способы улучшения.

7. Оформить отчёт и подготовить ответы на вопросы для защиты.

Задания по вариантам

Номер варианта определяется по порядковому номеру студента в журнале группы. Если номер больше 14, используется формула: вариант = $((N - 1) \bmod 14) + 1$. Вариант выполняется на основе общего примера, но с изменённой предметной областью, входными данными, параметрами эксперимента и анализируемыми классами.

Таблица 2 – Индивидуальные задания по вариантам

Вариант	Индивидуальное задание
1	1) Настроить ROS 2 пайплайн для сценария «городская дорога». 2) Детектировать и сопровождать объекты: автомобили, пешеходы, дорожные знаки. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
2	1) Настроить ROS 2 пайплайн для сценария «парковка торгового центра». 2) Детектировать и сопровождать объекты: машины, тележки, люди. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
3	1) Настроить ROS 2 пайплайн для сценария «аудитория университета». 2) Детектировать и сопровождать объекты: столы, стулья, доска, человек. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
4	1) Настроить ROS 2 пайплайн для сценария «коридор учебного корпуса». 2) Детектировать и сопровождать объекты: двери, люди, указатели. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
5	1) Настроить ROS 2 пайплайн для сценария «складское помещение». 2) Детектировать и сопровождать объекты: коробки, паллеты, погрузчик. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
6	1) Настроить ROS 2 пайплайн для сценария «промышленный цех». 2) Детектировать и сопровождать объекты: рабочие, станки, средства защиты. 3) Публиковать /tracks и реализовать правило

Вариант	Индивидуальное задание
	поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
7	1) Настроить ROS 2 пайплайн для сценария «парк или пешеходная зона». 2) Детектировать и сопровождать объекты: люди, велосипеды, деревья. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
8	1) Настроить ROS 2 пайплайн для сценария «медицинский кабинет». 2) Детектировать и сопровождать объекты: оборудование, инструменты, персонал. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
9	1) Настроить ROS 2 пайплайн для сценария «двор жилого комплекса». 2) Детектировать и сопровождать объекты: люди, машины, детская площадка. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
10	1) Настроить ROS 2 пайплайн для сценария «перекрёсток». 2) Детектировать и сопровождать объекты: светофоры, транспорт, пешеходные переходы. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
11	1) Настроить ROS 2 пайплайн для сценария «железнодорожная платформа». 2) Детектировать и сопровождать объекты: пассажиры, состав, таблички. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
12	1) Настроить ROS 2 пайплайн для сценария «торговый зал». 2) Детектировать и сопровождать объекты: витрины, корзины, покупатели. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
13	1) Настроить ROS 2 пайплайн для сценария «спортивная площадка». 2) Детектировать и сопровождать объекты: игроки, мяч, разметка. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.
14	1) Настроить ROS 2 пайплайн для сценария «сцена с БПЛА». 2) Детектировать и сопровождать объекты: дорога, деревья, здания сверху. 3) Публиковать /tracks и реализовать правило поведения робота: остановка, поворот или предупреждение. 4) Записать rosbag и показать rqt_graph/RViz с результатами.

Содержание отчёта

- тема и цель лабораторной работы;
- краткое описание используемого набора данных, структуры проекта и выбранной модели;
- версии Python, PyTorch/TensorFlow, OpenCV, CUDA или CPU-среды, а также используемых библиотек;
- ключевые листинги кода с пояснением назначения основных блоков;
- результаты проверки: изображения, кадры видео, графики, таблицы метрик и фрагменты логов;
- анализ ошибок, ограничений метода и вывод о применимости изученного подхода в системах компьютерного зрения;
- ответы на вопросы для защиты.

Вопросы для защиты работы

1. Зачем в ROS 2 используется `cv_bridge`?
2. Почему результаты анализа лучше публиковать отдельным топиком?
3. Как связаны детекция, NMS и трекинг в одном пайплайне?
4. Почему Behavior Manager не должен напрямую зависеть от YOLO-кода?
5. Какие данные нужно записать в `rosvbag` для воспроизводимой защиты?

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по организации и проведению самостоятельной работы по дисциплине
«Анализ изображений и видео» для студентов
направления подготовки
09.03.04 Программная инженерия
Направленность (профиль)
«Разработка и сопровождение программного обеспечения»

Ставрополь 2026

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ	3
МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА.....	4
МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ.....	5
ВОПРОСЫ ДЛЯ СОБЕСЕДОВАНИЯ ПО САМОСТОЯТЕЛЬНОМУ ИЗУЧЕНИЮ ЛИТЕРАТУРЫ.....	6
КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ.....	7

ВВЕДЕНИЕ

Настоящие методические указания определяют порядок организации и проведения самостоятельной работы студентов по дисциплине «Анализ изображений и видео» и являются дополнением к методическим указаниям по выполнению лабораторных работ. Самостоятельная работа рассматривается как обязательная составляющая образовательного процесса, обеспечивающая закрепление и углубление знаний, полученных на аудиторных занятиях.

Цель самостоятельной работы — сформировать у студентов устойчивые навыки самостоятельного освоения современных методов анализа изображений и видео, включая семантическую и инстанс-сегментацию, распознавание сцен, трекинг объектов, классификацию действий, постобработку результатов нейросетевых моделей и интеграцию видеоаналитики в ROS 2.

Указания содержат общую характеристику видов самостоятельной работы, рекомендации по изучению теоретического материала, порядок подготовки к выполнению практических (лабораторных) работ, перечень вопросов для собеседования и критерии оценивания сформированности компетенций.

ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа студента по дисциплине включает проработку теоретического материала, подготовку к лабораторным работам, самостоятельное выполнение индивидуальных заданий и подготовку к их защите. Основные виды самостоятельной работы и формы контроля приведены в таблице 1.

Таблица 1 — Виды самостоятельной работы и формы контроля

Вид самостоятельной работы	Содержание деятельности	Форма контроля
Проработка теоретического материала	Изучение лекций, методических указаний, учебной литературы и документации по библиотекам компьютерного зрения и глубокого обучения	Собеседование, тестирование
Подготовка к лабораторной работе	Изучение теоретического обоснования и примера, подготовка рабочего окружения, исходных изображений, видеоданных и зависимостей	Допуск к выполнению работы
Выполнение индивидуального задания	Самостоятельное выполнение варианта задания, запуск моделей, сохранение визуализаций, метрик и логов, оформление отчёта	Проверка отчёта
Подготовка к защите работы	Подготовка ответов на контрольные вопросы, объяснение выбранных моделей, параметров, метрик и полученных результатов	Защита отчёта

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА

Изучение теоретического материала рекомендуется проводить последовательно, в соответствии с порядком следования лабораторных работ. По каждой теме необходимо ознакомиться с основными понятиями компьютерного зрения, разобрать приведённый в работе пример, подготовить исходные изображения или видеоданные и проверить понимание материала по вопросам для самоконтроля. Перечень тем для самостоятельного изучения приведён в таблице 2.

Таблица 2 — Темы для самостоятельного изучения теоретического материала

№	Тема для самостоятельного изучения	Вопросы для проработки
1	Семантическая сегментация с U-Net или DeepLab на датасете Cityscapes	плотное предсказание, классы Cityscapes, цветовая карта, overlay, IoU и mIoU
2	Инстанс-сегментация с Mask R-CNN в Detectron2	отличие семантической и инстанс-сегментации, bounding box, class id, confidence, бинарные маски
3	Распознавание сцен с EfficientNet и Vision Transformer	scene recognition, ImageNet и Places365, top-5, вероятности классов, время инференса
4	Трекинг объектов в видео: SORT и Deep SORT	tracking-by-detection, Калмановский фильтр, ReID-признаки, ID switch, MOTA и IDF1
5	Классификация действий в видео с I3D или TimeSformer	выбор кадров, пространственно-временные признаки, 3D-CNN, transformer-подход, annotated video
6	Постобработка результатов детекции и сегментации	NMS, Soft-NMS, фильтрация по confidence и площади, морфологические операции, IoU до и после
7	Полный пайплайн анализа видео в ROS 2	sensor_msgs/Image, cv_bridge, YOLO, Deep SORT, публикация /tracks, behavior manager, rosbag

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

Выполнение каждой лабораторной работы осуществляется самостоятельно в соответствии с полученным вариантом индивидуального задания. Рекомендуется придерживаться последовательности этапов, приведённой в таблице 3.

Таблица 3 — Этапы выполнения лабораторной работы

Этап	Содержание деятельности
Подготовительный	Изучение теоретического обоснования и примера, подготовка рабочего окружения Python/ROS 2, установка библиотек, уточнение варианта задания
Основной	Самостоятельное выполнение индивидуального задания, запуск моделей и алгоритмов, фиксация параметров, команд, фрагментов кода и полученных результатов
Оформление отчёта	Подготовка отчёта в соответствии с требованиями работы: цель, ход выполнения, листинги, скриншоты, изображения, таблицы метрик и выводы
Защита	Ответы на контрольные вопросы, обоснование выбранных моделей, параметров обработки, метрик качества и полученных результатов

Отчёт оформляется в текстовом редакторе, листинги программ, команд и конфигураций приводятся моноширинным шрифтом. По итогам работы формулируются выводы, отражающие достигнутые результаты, значения метрик, особенности визуализации и приобретённые навыки анализа изображений и видео.

ВОПРОСЫ ДЛЯ СОБЕСЕДОВАНИЯ ПО САМОСТОЯТЕЛЬНОМУ ИЗУЧЕНИЮ ЛИТЕРАТУРЫ

Для подготовки к собеседованию по результатам самостоятельного изучения литературы рекомендуется проработать следующие вопросы:

1. Чем семантическая сегментация отличается от детекции объектов и инстанс-сегментации?
2. Что показывает метрика IoU/mIoU и почему визуальная проверка результата не менее важна, чем числовая оценка?
3. Каково назначение архитектур U-Net и DeepLab при решении задачи плотного предсказания?

4. Какие результаты возвращает Mask R-CNN для каждой найденной инстанции объекта?
5. Как confidence threshold влияет на количество найденных объектов, ложные срабатывания и пропуски?
6. Почему при сравнении CPU и GPU необходимо выполнять прогрев модели и усреднять время инференса?
7. Чем распознавание сцены отличается от классификации отдельного объекта на изображении?
8. Для чего анализируется top-5 предсказаний и почему модель ImageNet может ошибаться при распознавании сцен?
9. В чём заключается подход tracking-by-detection и как SORT связывает детекции между кадрами?
10. Зачем Deep SORT использует признаки внешнего вида объекта и как это влияет на число ID switch?
11. Чем классификация действий в видео отличается от классификации отдельных кадров?
12. Для чего применяются NMS, Soft-NMS и морфологическая обработка масок?
13. Какие показатели следует сравнивать до и после постобработки результатов детекции и сегментации?
14. Как организуется полный пайплайн анализа видео в ROS 2 и зачем сохранять rosbag при защите работы?

КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Оценивание результатов самостоятельной работы осуществляется по итогам выполнения и защиты лабораторных работ, а также собеседования. Критерии оценивания приведены в таблице 4.

Таблица 4 — Критерии оценивания

Оценка	Критерии оценивания
«Отлично»	Задание выполнено в полном объёме и без ошибок; отчёт оформлен в соответствии с требованиями; студент свободно владеет теоретическим материалом, обоснованно отвечает на вопросы и объясняет выбор моделей, параметров и метрик
«Хорошо»	Задание выполнено в полном объёме с незначительными недочётами; отчёт оформлен с небольшими замечаниями; студент владеет материалом, но допускает отдельные неточности в объяснении результатов
«Удовлетворительно»	Задание выполнено частично; отчёт содержит ошибки и недочёты; студент испытывает затруднения при объяснении алгоритмов, параметров обработки и полученных метрик
«Неудовлетворительно»	Задание не выполнено или выполнено с грубыми ошибками; отчёт отсутствует или не соответствует требованиям; студент не владеет теоретическим материалом