

**Департамент образования и науки города Москвы**  
**Государственное автономное образовательное учреждение**  
**высшего образования города Москвы «Московский городской**  
**педагогический университет»**

**Лабораторный практикум**

Москва 2024

**Цель работы:** научиться определять позу человека на основе изображений, изучить способы анализа ключевых точек тела, познакомиться с методами машинного обучения для автоматического распознавания поз.

**Описание работы:** в рамках данной работы необходимо освоить базовые подходы к распознаванию позы человека на изображениях и видео. Работа направлена на понимание того, как алгоритмы машинного обучения могут извлекать ключевые точки тела (положение головы, рук, ног) и интерпретировать их для определения текущей позы человека.

### **Теоретический материал**

Распознавание поз человека – это задача компьютерного зрения, направленная на определение и анализ положения тела на изображениях или видео. Алгоритмы для распознавания позы выделяют ключевые точки тела (такие как положение головы, рук, локтей, коленей) и определяют взаимное расположение этих точек для интерпретации позы. Эти технологии находят применение в медицине, спорте, безопасности, развлечениях и других сферах, где важно отслеживать или анализировать движения человека. Для полноценного понимания теоретического материала, далее будут введены понятия и их определения, которые помогут решить практические задания.

Ключевые точки тела – это анатомические ориентиры на теле (например, плечи, локти, колени), которые используются для построения скелетной модели человека.

Скелетное представление – структура, связывающая ключевые точки тела линиями, визуально отображающая позу и положение частей тела.

Распознавание позы – процесс идентификации положения тела на основе координат ключевых точек.

Модель машинного обучения – алгоритм, обученный на большом количестве данных для автоматического выполнения задач (например, распознавания поз).

Модели для распознавания позы работают поэтапно, чтобы идентифицировать объекты, определить ключевые точки тела и интерпретировать их взаимное расположение. Далее будут приведены основные этапы работы с моделями данного типа.

Первый этап – это выделение интересующих объектов (например, человека) на изображении или в кадре видео. Система сканирует входные данные и определяет область, содержащую человека, используя алгоритмы распознавания объектов. Этот этап важен, так как точное определение объекта позволяет модели сосредоточиться на нужной области, игнорируя фон и другие неподходящие объекты. Для этого часто применяются архитектуры, сочетающие свёрточные нейронные сети (CNN) с механизмами поиска и классификации.

После обнаружения объекта модель определяет координаты ключевых частей тела, таких как голова, плечи, локти, запястья, колени и стопы. Каждая точка представляет собой пару координат  $(x, y)$  на изображении. Ключевые точки помогают задать структуру тела, которая впоследствии используется для интерпретации позы. Разные модели выделяют разное количество точек: например, простые модели могут работать с 14 точками, а более сложные – с 17 или даже 25 точками, включая мелкие детали, такие как пальцы рук.

Следующим этапом является процесс построения скелета. На этом этапе происходит соединение ключевых точек линиями, что создаёт скелетную модель позы человека. Связи между точками соответствуют сегментам тела (например, плечо соединяется с локтем, а локоть с запястьем). Полученный скелет визуализирует положение частей тела и помогает модели интерпретировать позу. Такой подход упрощает дальнейший анализ, позволяя распознавать не только присутствие человека, но и его действия или состояние (например, сидит, стоит, поднимает руку).

Если система обрабатывает видео, важно отслеживать изменения позы на последовательных кадрах. Анализируя последовательности координат

ключевых точек, модель может выявить изменения в положении тела и определить движения, такие как ходьба, бег или поднятие рук. Этот процесс требует высокой производительности и точности, так как небольшие ошибки могут привести к неверной интерпретации действий. Модели для отслеживания движения часто используют временные нейронные сети (например, LSTM) или специальные алгоритмы для анализа последовательностей данных.

В лабораторной работе в качестве примера приведена модель YOLOv8. YOLOv8 – одна из современных версий модели YOLO, которая поддерживает задачи распознавания объектов и поз. Её преимущества включают:

- Высокая скорость обработки, что позволяет использовать её для работы с видео в реальном времени.
- Гибкость и точность при распознавании сложных поз.
- Возможность интеграции с Google Colab для быстрого запуска и тестирования без необходимости установки на локальном компьютере.

Однако YOLOv8 требует вычислительных ресурсов, особенно при обработке видео высокого разрешения, и может потребовать адаптации параметров для конкретных задач.

Рассмотрим практическую сторону вопроса применения моделей распознавания поз. В медицинской практике и реабилитации модели распознавания поз применяются для отслеживания движений пациентов и контроля за правильностью выполнения упражнений. Это особенно важно при восстановлении после травм или операций, когда необходимо следить за прогрессом и корректировать технику движений. Такие системы могут быть интегрированы в программы реабилитации и использоваться как в клиниках, так и в домашних условиях.

В сфере безопасности такие модели применяются для мониторинга необычного поведения в общественных местах, таких как аэропорты, торговые

центры или стадионы. Алгоритмы распознавания поз могут выявлять подозрительные действия, например падения или агрессивные движения, что помогает оперативно реагировать на потенциальные инциденты и повышать уровень безопасности.

В индустрии развлечений модели используются для управления персонажами через движения тела. Виртуальные игры и приложения, основанные на движении, позволяют пользователям взаимодействовать с цифровым миром без использования традиционных контроллеров. Технология распознавания поз также применяется в создании анимаций и спецэффектов в кино и телевидении.

### **Лабораторная работа №1**

#### **Задание №1. Проведение сравнительного анализа**

Вам необходимо провести исследование доступных инструментов для отслеживания поз человека. Для этого выполните поиск в сети интернет и заполните Таблицу 1, включив в нее не менее 8 инструментов или моделей, которые могут использоваться для решения задач по распознаванию позы.

В таблице следует указать:

- Название модели/инструмента (например, YOLO).
- Краткое описание: ключевые особенности, принцип работы или целевое применение.
- Плюсы: преимущества использования (например, точность, простота интеграции, производительность).
- Минусы: возможные ограничения (например, требования к вычислительным ресурсам, сложность настройки).

Таблица 1. Сравнение инструментов для решения задач по определению и отслеживанию поз

№	Название	Описание	Плюсы	Минусы
---	----------	----------	-------	--------

1.	YOLO	Одноэтапная модель определения поз человека на изображении	Высокая скорость, простота в обучении	Не самая высокая точность
2.				

## Задание №2. Работа с облачной IDE

В этом задании вам предстоит познакомиться с облачной средой программирования Google Colab. Google Colab – это облачная среда программирования, предоставляющая доступ к GPU для работы с моделями машинного обучения. Для её использования не требуется регистрация или подключение к сторонним сервисам.

1. Откройте браузер на вашем персональном компьютере.
2. Перейдите на сайт: [colab.research.google.com](https://colab.research.google.com).
3. Создайте новый блокнот, следуя примеру на рисунке 1.

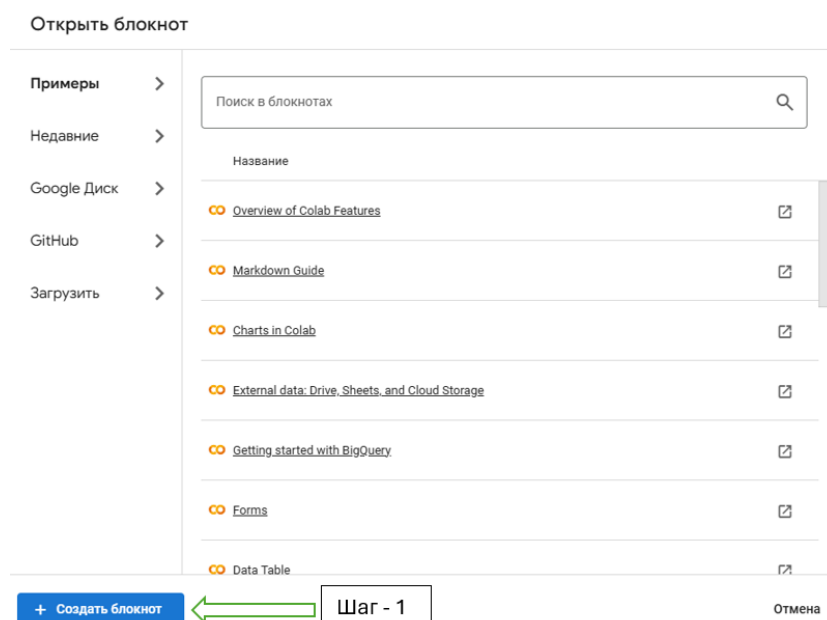


Рисунок 1. Первый шаг работы с IDE

Далее изучите основные элементы интерфейса Google Colab. Переименуйте свой блокнот и добавьте в него блоки для ввода кода и текста. Следуйте шагам, представленным на рисунке 2.

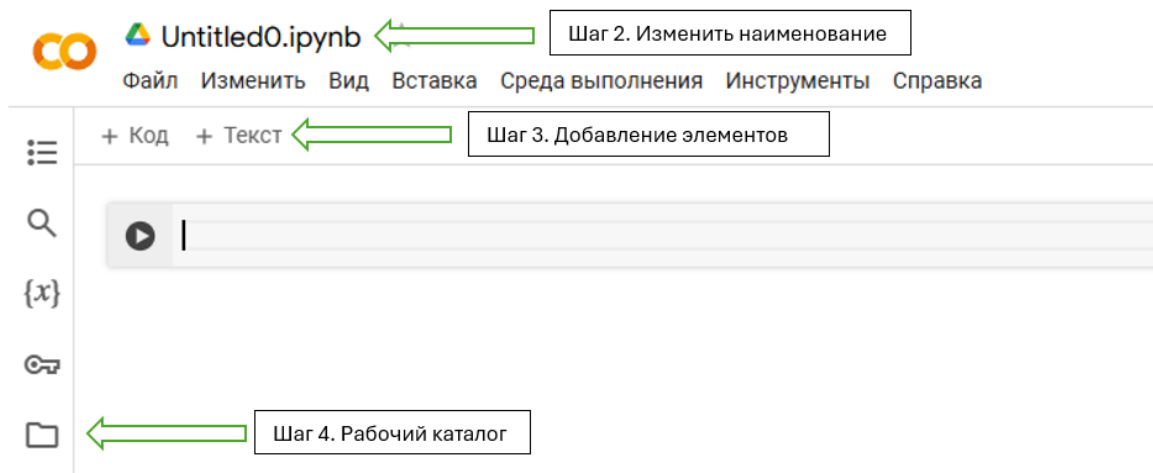


Рисунок 2. Основные элементы интерфейса

На заключительном этапе запустите первую ячейку с кодом. Для этого введите Python-код в ранее добавленную ячейку и нажмите на значок запуска. Если все шаги выполнены правильно, вы получите результат, как показано на рисунке 3.



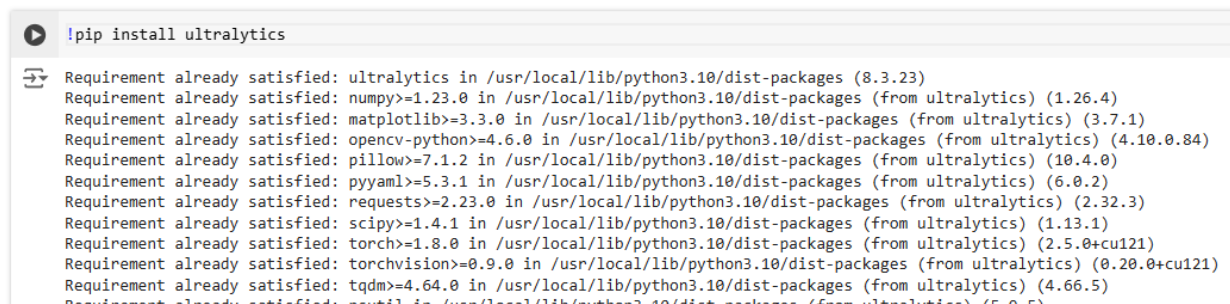
Рисунок 3. Запуск ячейки кода

### Задание №3. Реализация собственной модели распознавания поз

В этом задании вам предстоит использовать собственную модель для распознавания поз на базе YOLOv8. Следуйте инструкциям:

1. Используйте подготовленную рабочую среду из предыдущего задания в Google Colab.
2. Подготовьте одно или несколько изображений, на которых присутствует человек. Вы можете использовать свои фотографии, изображения из открытых источников.
3. Загрузите изображения в Colab и примените модель YOLOv8 для распознавания ключевых точек и построения скелета человека.

На рисунке 4 показан процесс установки необходимых библиотек. Для этого введите соответствующий код в ячейку и запустите её.

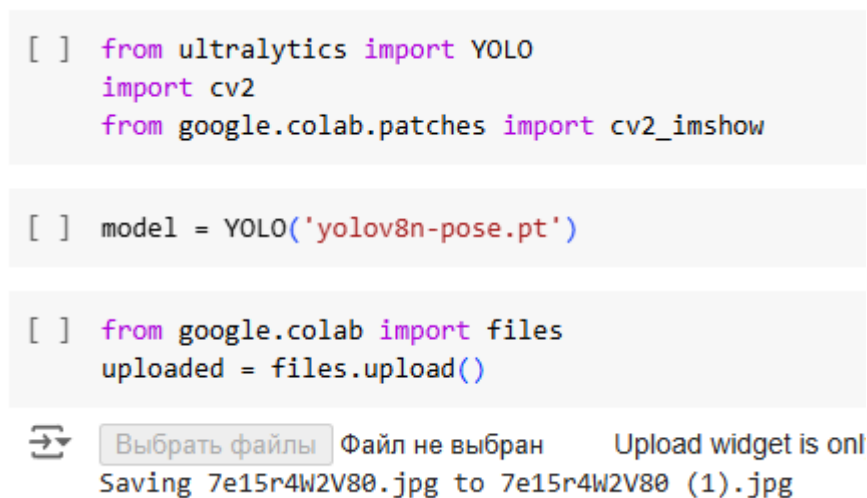


```
!pip install ultralytics

Requirement already satisfied: ultralytics in /usr/local/lib/python3.10/dist-packages (8.3.23)
Requirement already satisfied: numpy>=1.23.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (1.26.4)
Requirement already satisfied: matplotlib>=3.3.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (3.7.1)
Requirement already satisfied: opencv-python>=4.6.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (4.10.0.84)
Requirement already satisfied: pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (10.4.0)
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (6.0.2)
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (2.32.3)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (1.13.1)
Requirement already satisfied: torch>=1.8.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (2.5.0+cu121)
Requirement already satisfied: torchvision>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (0.20.0+cu121)
Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (4.66.5)
```

Рисунок 4. Установка библиотек

После выполнения предыдущего шага создайте несколько ячеек с кодом и последовательно заполните, и запустите их. Первая ячейка — отвечает за импорт ранее установленных библиотек в рабочую среду. Вторая ячейка — выполняет объявление модели и её конфигурации. Третья ячейка — позволяет загрузить изображения с персонального компьютера в рабочий каталог. Все этапы подробно показаны на рисунке 5.



```
[ ] from ultralytics import YOLO
import cv2
from google.colab.patches import cv2_imshow

[ ] model = YOLO('yolov8n-pose.pt')

[ ] from google.colab import files
uploaded = files.upload()
```

Выбрать файлы    Файл не выбран    Upload widget is onl  
Saving 7e15r4W2V80.jpg to 7e15r4W2V80 (1).jpg

Рисунок 5. Подготовка данных

Далее необходимо передать модели загруженный файл, получить результат и проанализировать его.

1. Добавьте несколько новых ячеек с кодом в рабочую область.
2. Введите код, приведённый на рисунке 6.



На втором этапе (ячейки с кодом) модель уже выдаёт результат, но он может быть недостаточно информативным и не отображать скелет. Чтобы получить визуализацию скелета, добавьте третью ячейку и измените параметры модели. Подберите оптимальные параметры, чтобы улучшить качество распознавания и анализа.

```
[ ] for filename in uploaded.keys():
    img_path = filename

[ ] results = model(img_path)

image 1/1 /content/7e15r4w2v80 (1).jpg: 544x640 2 persons, 372.1ms
Speed: 9.1ms preprocess, 372.1ms inference, 2.6ms postprocess per image at shape (1, 3, 544, 640)

annotated_img = results[0].plot()
# Можно изменить размер изображения (например, 740, 540)
resized_img = cv2.resize(annotated_img, (740, 540), interpolation=cv2.INTER_LINEAR)
cv2.imshow(resized_img)
```

Рисунок 6. Подготовка данных

Используя свое изображение интерпретируйте результаты модели, укажите какое количество человек было на фото изначально, сколько модели удалось распознать и с какой вероятностью. Укажите какие точки скелета модель смогла распознать на загруженной фотографии.