МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное   
учреждение высшего образования   
«Южный федеральный университет»

Институт математики, механики   
и компьютерных наук им. И. И. Воровича

Кафедра прикладной математики и программирования

Дурманенко Всеволод Сергеевич

Оценка положения тела на видео

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
по направлению подготовки  
02.03.02 – Фундаментальная информатика и информационные технологии

**Научный руководитель** –   
доц. Демяненко Яна Михайловна

Допущено к защите:  
заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сидоров С. С.

Ростов-на-Дону – 2024

Оглавление

[Введение 3](#_Toc162624633)

[1. Постановка задачи 4](#_Toc162624634)

[2. Обзор подходов 5](#_Toc162624635)

[2.1. Методы решения задачи 5](#_Toc162624636)

[2.2. Двухэтапные методы 5](#_Toc162624637)

[2.2.1. Сверточная нейросеть 6](#_Toc162624638)

[2.2.2. R-CNN 6](#_Toc162624639)

[2.2.3. Fast R-CNN 8](#_Toc162624640)

[2.2.4. Faster R-CNN 10](#_Toc162624641)

[2.2.5. Mask R-CNN 11](#_Toc162624642)

[2.2.6. Сравнение алгоритмов 13](#_Toc162624643)

[2.3. Одноэтапные методы 14](#_Toc162624644)

[2.3.1. YOLO 14](#_Toc162624645)

[2.3.2. Модификации алгоритма YOLO 15](#_Toc162624646)

[2.4. Представление анатомии человека для компьютера 16](#_Toc162624647)

[2.4.1. Моделирование анатомии человека 16](#_Toc162624648)

[2.4.2. Работа алгоритма 17](#_Toc162624649)

[3. Основная часть 19](#_Toc162624650)

[3.1. Идея решения 19](#_Toc162624651)

[3.2. Разработка алгоритма 19](#_Toc162624652)

[3.3. Графический интерфейс 22](#_Toc162624653)

[3.4. Результаты работы алгоритма 23](#_Toc162624654)

[3.5. Недостатки и преимущества данного подхода 23](#_Toc162624655)

[Приложения 25](#_Toc162624656)

[Заключение 26](#_Toc162624657)

[Литература 27](#_Toc162624658)

# Введение

В современном мире использование компьютерного зрения и анализа видеоданных играет значительную роль в различных областях, таких как медицина, спорт, безопасность, робототехника и другие. Одним из важных аспектов анализа видео является оценка положения тела объектов на записи. Эта информация может быть критически важной для определения движений, поз, деформаций или других параметров, имеющих большое значение для конкретной области приложения.

Данная работа посвящена исследованию методов оценки положения тела на видео и их применению в различных сферах. В рамках исследования рассматриваются современные подходы к анализу видеоданных, а также методы машинного обучения, применяемые для решения задачи оценки положения тела.

Актуальность исследования обусловлена растущим интересом к проблемам компьютерного зрения и анализа видеоданных в различных областях, таких как медицина, спорт, безопасность и другие. Оценка положения тела на видео имеет широкий спектр применений, включая мониторинг движений для реабилитации пациентов, анализ движений спортсменов для улучшения их тренировочных программ, а также обнаружение аномальных или подозрительных действий для обеспечения безопасности. Разработка эффективных методов оценки положения тела на видео имеет практическое значение и потенциал для дальнейшего развития и улучшения существующих технологий.

# Постановка задачи

Целью данной работы является разработка и исследование метода оценки положения тела на видео с использованием современных технологий компьютерного зрения и машинного обучения. Для достижения этой цели поставлены следующие задачи:

* Исследовать существующие методы анализа видеоданных и оценки положения объектов на видео, применение данных методов для решения задачи оценки положения тела на видео.
* Реализовать метод оценки положения тела на основе полученных знаний и сравнить его эффективность с существующими методами.
* Провести анализ результатов и экспериментов, выявить преимущества и недостатки разработанного метода.

# Обзор подходов

## Методы решения задачи

Методы, основанные на регионах, также известные как двухэтапные методы, представляют собой подход, который состоит из двух этапов. На первом этапе используется селективный поиск или специальный слой нейронной сети для выделения регионов интереса, то есть областей с высокой вероятностью наличия объектов внутри себя. На втором этапе выбранные регионы подвергаются анализу классификатором, чтобы определить их принадлежность к исходным классам, а также регрессором, который уточняет местоположение ограничивающих рамок.

К данным методом относятся алгоритмы R-CNN, Fast R-CNN, Faster R-CNN и Mask R-CNN.

Одноэтапные методы — это подход к обработке изображений, который не использует отдельный алгоритм для генерации регионов. Вместо этого они предсказывают координаты определенного количества ограничивающих рамок с различными характеристиками, такими как результаты классификации и степень уверенности, и затем корректируют местоположение рамок для более точного определения объектов на изображении.

К данным методам относятся алгоритмы YOLO, его усовершенствованные версии YOLOvx (x – номер версии), SSD.

## Двухэтапные методы

Рассмотрим развитие сверточных нейронных сетей и нейросетевых алгоритмов, начиная с идеи использования регионных предположений, которая привела от алгоритма R-CNN к более быстрому и эффективному алгоритму Mask R-CNN.

### Сверточная нейросеть

Сверточная нейронная сеть (CNN) представляет собой специализированный тип нейронной сети, применяемый в основном для обработки изображений и других данных с пространственной структурой, таких как аудио и видео. Основное отличие CNN от обычной нейронной сети заключается в способе обработки входных данных.

Основные концепции сверточных нейронных сетей включают в себя сверточные слои, которые обнаруживают локальные шаблоны, такие как грани и текстуры, путем скольжения окна фильтра по входным данным. Затем следует операция подвыборки, которая уменьшает размерность данных, сохраняя наиболее важные признаки.

После каждого сверточного или подвыборочного слоя обычно применяется нелинейная функция активации, такая как ReLU, для введения нелинейности в модель. И, наконец, после серии сверточных и подвыборочных слоев следует один или несколько полносвязных слоев, которые объединяют глобальные признаки для принятия окончательного решения.

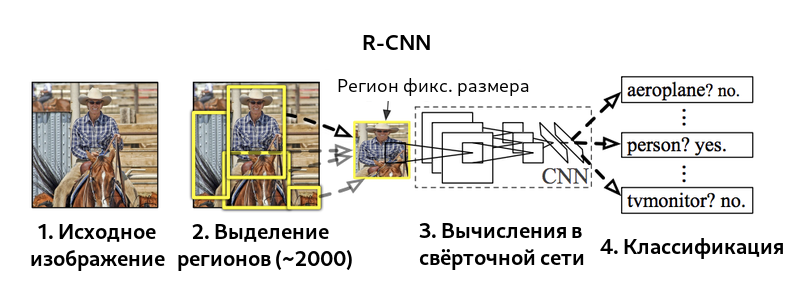
В целом, сверточная нейронная сеть отличается от обычной тем, что она эффективно обрабатывает данные с пространственной структурой, сохраняя локальные шаблоны и признаки, что делает ее более подходящей для задач компьютерного зрения и обработки изображений.

### R-CNN

Region-CNN (R-CNN) — это метод, который базируется на свёрточных нейронных сетях и отличается от традиционного подхода скользящих окон фиксированного размера. Вместо этого он применяет селективный поиск для обнаружения "регионов" на изображениях — прямоугольных рамок различных размеров, которые, вероятно, содержат объекты. Этот метод обеспечивает более быстрое и эффективное обнаружение объектов независимо от их размера, расстояния до камеры и угла обзора.

На первом этапе алгоритм генерирует примерно две тысячи регионов для каждого изображения. После этого найденные регионы преобразуются с использованием аффинных преобразований, паддингов или расширения ограничивающих рамок до размеров, необходимых для входа в сверточную нейронную сеть (CNN). Обычно для этой цели используется архитектура специальная архитектура, которая извлекает порядка 4096 признаков для каждого региона.

На последнем этапе векторы признаков регионов подаются на обработку в множество SVM (Support Vector Machine), каждый из которых проводит классификацию объектов для своего домена.

Селективный поиск также может быть обучен с использованием линейной регрессии для настройки параметров, таких как ширина, высота и центр. Этот метод, известный как bounding-box regression, улучшает точность выделения объекта. В процессе регрессии используются признаки, полученные после работы сверточной нейронной сети.

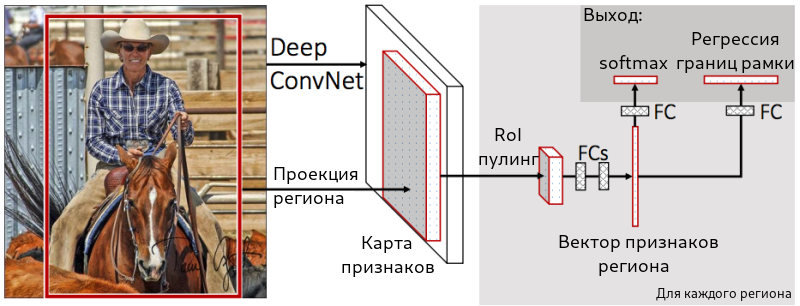
Из минусов данного алгоритма наиболее значительными являются следующие:

* **Высокая вычислительная сложность**: Генерация и классификация тысяч регионов для каждого изображения требует больших вычислительных ресурсов и времени. Это делает алгоритм медленным и неэффективным для реального времени.
* **Неустойчивость к изменениям масштаба**: R-CNN неэффективно работает с объектами различных масштабов из-за использования фиксированных размеров регионов. Это может привести к пропускам объектов малого размера или ложным срабатываниям на больших объектах.
* **Недостаточное использование пространственной информации**: R-CNN не учитывает пространственные отношения между объектами на изображении, что может привести к неточностям в локализации и классификации.
* **Неэффективное использование памяти**: Обработка большого количества регионов для каждого изображения требует значительного объема оперативной памяти, что может ограничивать масштабируемость алгоритма.

### Fast R-CNN

Fast R-CNN представляет собой улучшенную версию алгоритма R-CNN, который значительно ускоряет процесс обучения и инференса. В отличие от оригинального R-CNN, где классификация каждого из 2000 регионов выполняется отдельно, Fast R-CNN подает на вход сверточной нейронной сети (CNN) всё изображение целиком для получения общей карты признаков. Затем предложенные регионы выделяются на этой общей карте, что существенно сокращает количество операций свертки.

Чтобы привести регионы разного размера к фиксированному размеру, используется операция RoIPooling (Region of Interest Pooling). В этой операции регион разделяется на сетку ячеек, размер которой соответствует размеру выхода, и для каждой ячейки выбирается максимальное значение. Полученные регионы фиксированного размера становятся входом для полносвязного слоя, который осуществляет как классификацию, так и линейную регрессию для коррекции границ рамок.

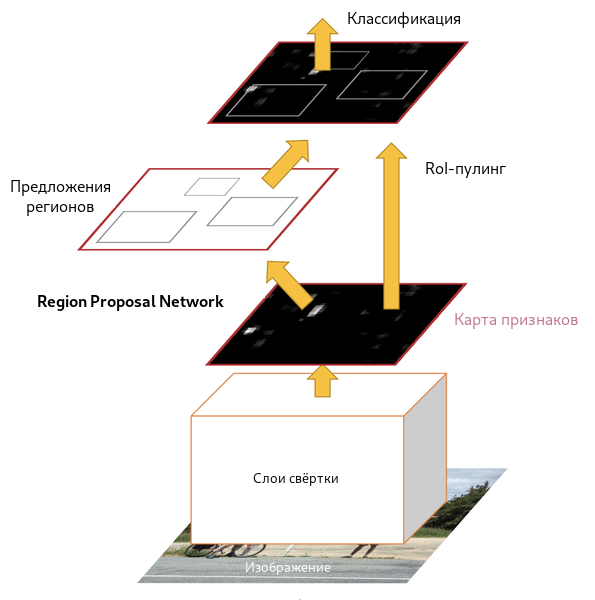
Основным отличием Fast R-CNN также является использование совместного обучения SVM для классификации, CNN и bounding box регрессора вместо их независимого обучения. Для этого применяется совместная функция потерь, которая учитывает все аспекты обучения модели: классификацию, распознавание объектов и коррекцию рамок. Это позволяет сети эффективно учиться на данных и значительно улучшает её производительность.

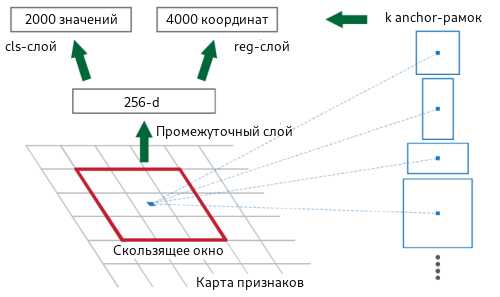
Несмотря на свои преимущества, метод Fast R-CNN имеет несколько недостатков:

* **Ограничения в скорости**: Хотя Fast R-CNN значительно ускоряет процесс обучения и инференса по сравнению с оригинальным R-CNN, его производительность все равно может быть недостаточной для решения задач в реальном времени, особенно при работе с большими объемами данных или на устройствах с ограниченными вычислительными ресурсами.
* **Ограничения на маленьких объектах**: Fast R-CNN может иметь трудности с обнаружением и классификацией маленьких объектов из-за потери деталей при использовании RoIPooling и ограничений связанных с размером регионов.
* **Чувствительность**: Фоновые помехи могу привести к снижению точности обнаружения.

### Faster R-CNN

Fast R-CNN, аналогично оригинальному алгоритму R-CNN, использует метод селективного поиска для обнаружения регионов. Хотя использование единоразовой свёртки значительно уменьшило время обучения на одном тестовом изображении, селективный поиск, который генерирует предложения регионов, остаётся узким местом в производительности Fast R-CNN. Авторы алгоритма Faster R-CNN предложили решить эту проблему, предложив использовать отдельный модуль, известный как сеть генерации регионов (Region Proposal Network, RPN). RPN выполняет функцию генерации регионов на основе признаков исходного изображения. После генерации регионов они проходят через два полносвязных слоя: слой регрессии рамок (reg layer), который прогнозирует значения смещения для ограничивающих рамок, и слой классификации рамок (cls layer), который классифицирует изображения в предложенных областях. Важную роль также играют ключевые рамки (anchor boxes) — рамки с различными положениями и размерами для скользящего окна. Для каждого масштаба обычно выбирают три ключевые рамки: квадратную, прямоугольную с горизонтальной ориентацией и прямоугольную с вертикальной ориентацией. Ключевые рамки перемещаются скользящим окном с учётом их масштаба для генерации регионов. Для сгенерированных регионов вычисляются вероятности нахождения объекта внутри рамки с помощью слоя cls, а слой reg отвечает за смещение их местоположения. После прохождения через слой RPN следует RoIPooling, как и в алгоритме Fast R-CNN, для приведения регионов к одному размеру и последующей классификации и коррекции границ ограничивающих рамок. Поскольку как сеть в целом, так и RPN занимаются классификацией и регрессией границ, функция потерь учитывает как окончательное решение по классификации и регрессии координат, так и вычисления, выполненные RPN.

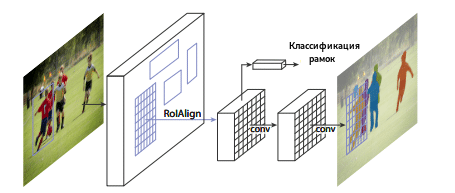




Несмотря на то, что метод Faster R-CNN внес значительные улучшения по сравнению с предыдущими версиями алгоритма R-CNN, он все же имеет недостаток. Этим недостатком является то, что вычисления могут быть дорогостоящими из-за необходимости обнаружения областей и нескольких этапов обработки.

### Mask R-CNN

Mask R-CNN — это улучшение алгоритма Faster R-CNN, разработанное в 2017 году, которое позволяет не только обнаруживать объекты и классифицировать их, но и выполнять их сегментацию на уровне пикселей. В дополнение к традиционным для алгоритмов семейства R-CNN меткам класса и координатам ограничивающей рамки Mask R-CNN добавляет маску объекта - матрицу принадлежности пикселей данному объекту. Для каждого класса маски предсказываются с помощью классификации без предварительного знания о содержании региона, используя отдельный классификатор на последнем уровне сети. Mask R-CNN внес несколько архитектурных изменений по сравнению с Faster R-CNN, включая использование RoIAlign вместо RoIPooling. RoIPooling хорошо масштабирует ограничивающие рамки, но для сегментации масок он недостаточно точен. RoIAlign не использует округление сдвигов для пулинга, а сохраняет значения с плавающей точкой, используя билинейную интерполяцию, что обеспечивает более точную сегментацию объектов.





### Сравнение алгоритмов

Ниже приведена таблица сравнения ключевых характеристик алгоритмов R-CNN, Fast R-CNN, Faster R-CNN и Mask R-CNN.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Характеристика | R-CNN | Fast R-CNN | Faster R-CNN | Mask R-CNN |
| Скорость обработки | Очень низкая | Высокая | Очень высокая | Высокая |
| Точность обнаружения | Высокая | Очень высокая | Очень высокая | Очень высокая |
| Скорость обучения | Низкая | Средняя | Высокая | Средняя |
| Объем памяти | Высокий | Средний | Средний | Высокий |
| Дополнительные затраты |  | ROI pooling | RPN | Маски для сегментации |

## Одноэтапные методы

Семейство алгоритмов быстрой детекции, в отличие от R-CNN, обеспечивает высокую скорость работы за счет отказа от использования предсказанных регионов. Это особенно важно в областях, где требуется быстродействие, например, в беспилотном управлении автомобилем.

### YOLO

Одним из одноэтапных алгоритмов является алгоритм YOLO (You Look Only Once). Данный алгоритм был предложен в 2016 году и представляет собой новаторский подход к обнаружению объектов в реальном времени. В отличие от алгоритмов семейства R-CNN, YOLO разбивает изображение на сетку из N×N ячеек, и каждая ячейка отвечает за определение параметров местоположения объекта. Если центр объекта попадает внутрь координат ячейки, она считается ответственной за определение параметров объекта. Вместо предложения регионов и вычисления решения для каждого отдельного региона, YOLO генерирует несколько вариантов местоположения ограничивающих рамок для каждой ячейки. Каждый вариант описывается пятью значениями: координатами центра ограничивающей рамки, ее шириной и высотой, а также степенью уверенности в наличии объекта в рамке. Для каждой пары класса объектов и ячейки определяется вероятность наличия объекта данного класса в этой ячейке. Таким образом, финальный слой сети работает с тензором размерности N×N×(5B+C), где B - количество предсказываемых ограничивающих рамок для каждой ячейки, а C - количество классов объектов.

Алгоритм YOLO обладает высокой скоростью работы благодаря разбиению на постоянное количество ячеек и предсказанию параметров объектов в каждой ячейке одновременно. Однако его недостатком является невысокое качество распознавания объектов сложной формы или группы маленьких объектов из-за ограниченного числа кандидатов для ограничивающих рамок.

### Модификации алгоритма YOLO

* YOLOv2, выпущенный в 2016 году, улучшил исходную модель, включив пакетную нормализацию, поля привязки и кластеры измерений.
* YOLOv3, запущенный в 2018 году, еще больше повысил производительность модели за счет более эффективной магистральной сети, нескольких якорей и объединения пространственных пирамид.
* YOLOv4 был выпущен в 2020 году и представил такие инновации, как увеличение данных Mosaic, новую головку обнаружения без привязки и новую функцию потерь.
* YOLOv5 еще больше улучшил производительность модели и добавил новые функции, такие как оптимизация гиперпараметров, интегрированное отслеживание экспериментов и автоматический экспорт в популярные форматы экспорта.
* YOLOv6 был открыт компанией Meituan в 2022 году и используется во многих автономных роботах-доставщиках компании.
* В YOLOv7 добавлены дополнительные задачи, такие как оценка позы в наборе данных ключевых точек COCO.
* YOLOv8 — это последняя версия YOLO от Ultralytics. Являясь передовой, современной (SOTA) моделью, YOLOv8 развивает успех предыдущих версий, предлагая новые функции и улучшения для повышения производительности, гибкости и эффективности. YOLOv8 поддерживает полный спектр задач искусственного интеллекта, включая обнаружение, сегментацию, оценку позы, отслеживание и классификацию. Эта универсальность позволяет пользователям использовать возможности YOLOv8 в различных приложениях и областях.

## Представление анатомии человека для компьютера

Определение положения и ориентации различных частей человеческого тела на изображениях или видео - это ключевая задача в области компьютерного зрения и искусственного интеллекта. Эта задача имеет широкий спектр применений, включая разработку систем для управления роботами, анализа видеонаблюдения, виртуальной реальности, медицинской диагностики и даже в области спортивных технологий. Оценка положения тела позволяет компьютерным системам понимать и интерпретировать человеческое движение, что является важным шагом к созданию более умных и автономных устройств.

### Моделирование анатомии человека

В большинстве методов используется жесткая кинематическая модель N-суставов, в которой человеческое тело представлено как объект с суставами и конечностями, содержащими кинематическую структуру тела и информацию о форме тела.

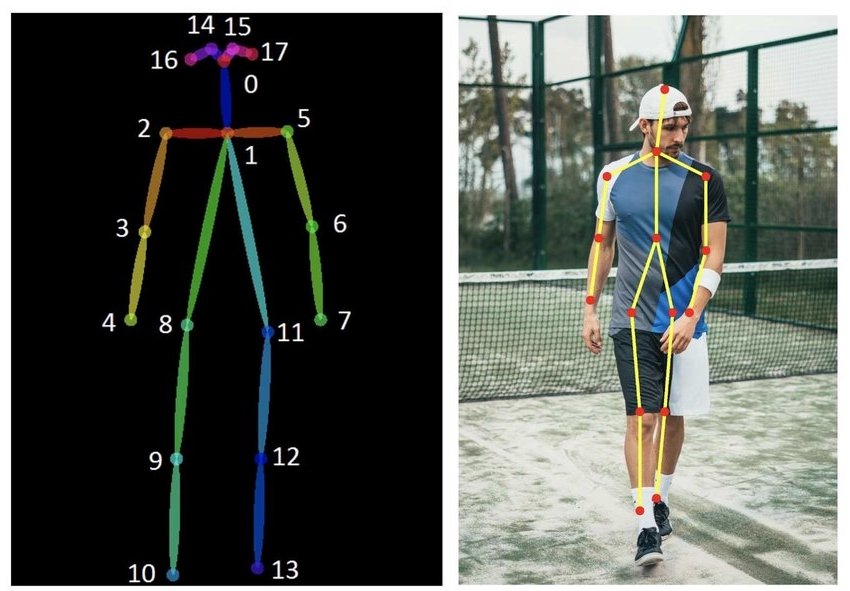
Существуют три типа моделирования анатомии человека:

* **Kinematic model** (Скелетная модель) - это модель также известная как скелетно-ориентированная модель, и она применяется для анализа позы в форматах 2D и 3D. Эта удобная и интуитивно понятная модель человеческого тела включает набор суставов и ориентацию конечностей для описания структуры тела. Таким образом, модели, оценивающие положение скелета, используются для улавливания взаимосвязей между различными частями тела. Однако кинематические модели имеют ограничения в передаче информации о текстуре или форме.
* **Planar model** (Плоская модель) – это модель, известная также как модель на основе контура, применяется для оценки 2D позы. Плоские модели используются для представления внешнего вида и формы человеческого тела. Обычно различные части тела представлены несколькими прямоугольниками, которые приближены к контуру человеческого тела. Хорошим примером такой модели является модель активных форм, которая используется для охвата полного контура человеческого тела и его деформации с помощью анализа главных компонентов.
* **Volumetric model** (Объемная модель) – это объемная модель, применяемая для трехмерной оценки положения, представляет собой важный инструмент в анализе поз. Они предназначены для моделирования статистических и артикулированных форм человеческого тела и поз. Эти модели могут быть использованы для вывода различных характеристик и параметров позы.



### Работа алгоритма

Оценка положения основывается на использовании позы и ориентации для прогнозирования и отслеживания расположения человека или объекта. Методы машинного обучения позволяют программам анализировать пространственные позиции тела на изображениях или видео. Обычно это двухэтапный процесс, который сначала определяет ограничивающие рамки вокруг человека, а затем оценивает позу в каждой рамке. Для этого используются ключевые точки человека или объекта, такие как локти, колени, запястья и т.д. Существуют два типа подходов: однопозиционный и многопозиционный. Однопозиционный подход применяется для оценки положения одного объекта на сцене, в то время как многопозиционный используется для обнаружения положений нескольких объектов.



# Основная часть

## Идея решения

В первую очередь для решения задачи классификации тела человека на видео необходимо определить саму фигуру человека и ключевые точки на ней. Для определения самого положения тела необходимо рассчитать углы между определенными ключевыми точками и их расположение относительно друг друга. У каждой человеческой позы есть отличительные черты, благодаря которым можно ее классифицировать:

* Стоит – угол между бедром, коленом и лодыжкой больше 170 градусов, туловище находится между головой и ногами относительно оси X.
* Сидит – угол между бедром, коленом и лодыжкой меньше 170 градусов, при этом таз расположен на одной прямой с коленом.
* Лежит – ноги, туловище и плечи находятся практически на одной прямой относительно оси Y.
* Идет – угол между ногами меньше 70 градусов, стопы развернуты в одну сторону.
* Бежит – угол между ногами больше 70 градусов, руки согнуты в локтях, стопы развернуты в одну сторону.
* Наклоняется – голова находится между тазом и стопой относительно оси Y.

В связи с тем, что некоторые условия пересекаются с другими, необходимо начинать рассмотрение возможного положения от более частного к более общему случаю.

## Разработка алгоритма

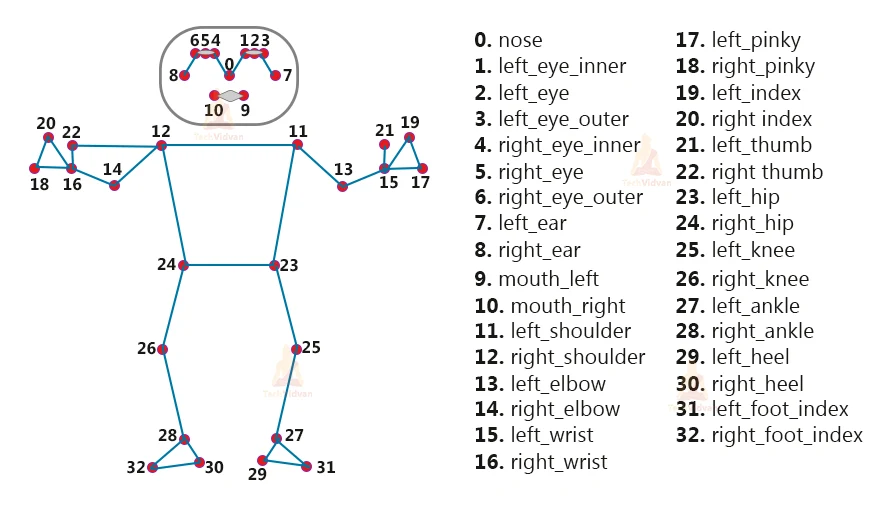
Для решения поставленной задачи был использован язык программирования Python с использованием библиотек Mediapipe и OpenCV. Mediapipe предоставляет различные инструменты для анализа изображений и видео, включая распознавание объектов, действий и ключевых точек на изображениях. OpenCV является популярной библиотекой компьютерного зрения, которая предоставляет широкий спектр инструментов для обработки изображений и видео.

Обнаружение позы человека в библиотеке Mediapipe осуществляется с использованием нейронных сетей и алгоритмов компьютерного зрения. Для этого библиотека использует модель PoseNet, которая работает на основе сверточных нейронных сетей.

Процесс обнаружения скелетной модели человека включает в себя несколько шагов:

* **Предварительная обработка изображения**: Входное изображение подается на вход модели PoseNet, после чего происходит предварительная обработка, включая изменение размера изображения и нормализацию пикселей.
* **Использование нейронной сети**: Изображение проходит через нейронную сеть PoseNet, которая состоит из нескольких слоев свертки и pooling, позволяющих выявлять ключевые точки тела человека.
* **Обнаружение ключевых точек**: Модель PoseNet определяет позу человека, выделяя ключевые точки, такие как нос, глаза, плечи, локти, запястья, таз, колени и щиколотки.
* **Отрисовка сегментов**: модель соединяет обнаруженные ключевые точки сегментами, чтобы создать полную скелетную модель человека на изображении.

Таким образом, обнаружение модели человека в библиотеке Mediapipe происходит путем анализа изображения с помощью нейронных сетей и выделения ключевых точек тела, что позволяет точно определить скелетную модель человека на изображении или видео.



В результате получаем координаты точек и коэффициенты их видимости. Формат, в котором Mediapipe предоставляет данные, не очень удобен для дальнейшей работы. В связи с этим был создан класс, в котором есть словарь в формате string:Point. Ключом здесь является название части тела, а значением – точка, содержащая координаты расположения этой части тела. Помимо словаря в данном классе есть функции для определения самого положения тела человека.

Для определения угла были написаны две функции get\_angle и angle\_exists. Первая отвечает за расчет самого угла, а вторая проверяет его существование по коэффициентам видимости.

def get\_angle(a: Point, b: Point, c: Point) -> float:

    deg = math.degrees(math.atan2(c.y - b.y, c.x - b.x) - math.atan2(a.y - b.y, a.x - b.x))

    return deg + 360 if deg < 0 else deg

def angle\_exists(a: Point, b: Point, c: Point) -> bool:

    if a.visibility < 0.9 or b.visibility < 0.9 or c.visibility < 0.9:

        return False

    return True

В основном файле импортируются необходимые модули. Создается экземпляр класса Pose с установленными параметрами для уверенности в обнаружении. Загружается изображение, преобразуется в RGB формат и обрабатывается с помощью метода process объекта Pose, чтобы обнаружить человека на изображении.

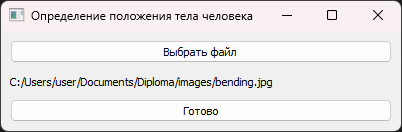
Затем создается объект класса Human, в который передаются обнаруженные ключевые точки человеческого тела. Вызывается метод estimate\_pose для определения типа позы человека. Результат выводится в консоль. Помимо консоли, результат можно посмотреть на итоговом изображении. Для этого используется функция из OpenCV, обернутая в функцию put\_test(image, text), где параметрами являются итоговое изображение и текст, который необходимо поместить в левую верхнюю часть экрана.

Методы draw\_landmarks из mp\_drawing используются для визуализации ключевых точек на изображении. Наконец, итоговое изображение выводится на экран.

## Графический интерфейс

Для создания графического интерфейса была использована библиотека PyQt5. PyQt5 – это набор Python-обёрток над Qt, мощным фреймворком для создания графических интерфейсов. Выбор был сделан в пользу данной библиотеки из-за простоты её использовании и кроссплатформенности.

Для создания графического окна был создан класс MainWindow, унаследованный от QMainWindow. Внутри вызывается конструктор базового класса и задается расположение по центру окна. Также создаются две кнопки и одна метка: одна кнопка используется для выбора файла из проводника, другая – для вызова функции, в которой применяется сам алгоритм. Метка используется как подсказка, показывающая путь к выбранному файлу. Для безопасности выбора файла используются фильтры, которые позволяют выбрать только файлы с расширением .mp4 или .jpg.



## Результаты работы алгоритма

…

## Преимущества и недостатки данного подхода

Ниже представлены преимущества и недостатки выбранного подхода на основе ключевых точек.

Преимущества:

* Относительная простота реализации: Использование библиотеки Mediapipe и расчет углов между ключевыми точками относительно друг друга является довольно простым в реализации. Для добавления новых положения тела для распознавания не нужно собирать огромный датасет для обучения. Достаточно найти отличительные черты данного положения тела от других.
* Контролируемый процесс: При использовании данного подхода мы имеем контроль над тем, какие ключевые точки выбирать и как рассчитывать углы между ними. Это позволяет легче интерпретировать и отслеживать процесс определения положения тела.
* Прозрачность: Поскольку вы рассчитываете углы между ключевыми точками, процесс определения положения тела человека может быть более прозрачным и понятным для других.

Недостатки:

* Чувствительность к шуму и ошибкам: Данный подход более чувствителен к шуму на изображении или к ошибкам в определении ключевых точек. Даже небольшие ошибки в определении положения ключевых точек могут существенно повлиять на расчет углов и, следовательно, на определение положения тела.
* Ограниченная обобщаемость: Данный подход является менее обобщаемым, чем нейронные сети, основанные на дообучении. Например, если нужно определять положение тела для разных типов людей (например, разные возрастные группы или телосложения), данный метод может потребовать дополнительной настройки или модификации.

# Приложения

# Заключение

Таким образом, проведенное исследование по обнаружению положения тела человека на видео показало эффективность разработанных алгоритмов и моделей. Полученные результаты демонстрируют высокую точность и надежность при распознавании поз и движений человека. В дальнейшем исследовании можно продолжить работу над улучшением алгоритмов и расширением функциональности системы для более точного и широкого применения в реальном мире.

# Литература

…