МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное   
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»  
(Самарский университет)

Институт информатики и кибернетики

Кафедра технической кибернетики

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Применение алгоритмов машинного обучения для задач захвата движения человека на видеоизображении

по программе бакалавриата по направлению подготовки   
01.03.02 Прикладная математика и информатика,

профиль «Компьютерные науки»

Обучающийся А.А. Сорока

(*подпись*)

Научный руководитель ВКР,

*доцент, к.ф.-м.н.*

Д.А. Савельев

(*подпись*)

Нормоконтролёр С.В. Суханов

(*подпись*)

Самара 2024

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное   
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

Институт информатики и кибернетики

Кафедра технической кибернетики

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заведующий кафедрой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Куприянов  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г. |
|  |  |

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ   
БАКАЛАВРА**

обучающемуся группы 6409-010302D ***Сорока Александру Александровичу***

Тема ВКР: ***Применение алгоритмов машинного обучения для задач захвата движения человека на видеоизображении***

утверждена приказом по университету от « 08 » 04 2021 г. № 333-ст .

Исходные данные: *двумерное волновое уравнение, метод разделения переменных Фурье, признаки сходимости функциональных рядов, метод вычислительного эксперимента, метод конечных разностей*.

Перечень вопросов, подлежащих разработке в ВКР:

* *получение решения краевой задачи для волнового уравнения в виде бесконечного ряда Фурье при различных способах описания входного импульса;*
* *получение оценки остатка ряда Фурье;*
* *разработка программного средства для численного моделирования и исследования погрешности;*
* *обеспечение контроля погрешности решения волнового уравнения;*
* *разработка программы разностного решения волнового уравнения и использование программного средства численного моделирования с помощью рядов Фурье для тестирования программы разностного решения.*

|  |  |
| --- | --- |
| Руководитель ВКР  *доцент*  Д.А. Савельев  (*подпись*)  « 15 » 02 2022 г. | Задание принял к исполнению  А.А. Сорока  (*подпись*)  « 15 » 02 2022 г. |

*Примечание: Задание в данном случае печатается* ***с двух сторон одного листа****, пункты задания и все подписи будут находиться на одном листе.*

РЕФЕРАТ

**Выпускная квалификационная работа бакалавра:** 63 c., 16 рисун-ков, 6 таблиц, 10 источников, два приложения.

**Презентация:** 12 слайдов Microsoft PowerPoint.

НЕЙРОННЫЕ СЕТИ, БЕЗМАРКЕРНЫЙ ЗАХВАТ ДВИЖЕНИЯ, ОБРАБОТКА ИЗОБРАЖЕНИЙ, АНИМАЦИЯ, НЕЙРОСЕТЕВОЙ АНАЛИЗ ДАННЫХ

Работа посвящена исследованию погрешности численного моделирования процесса распространения электромагнитного импульса в однородном планарном волноводе.

Значительная часть работы уделена нахождению аналитического решения для двух способов описания входного импульса, а также теоретическому и экспериментальному исследованию оценки остатка ряда.

Часть работы связана с построением разностной схемы и исследованием погрешности численного решения с помощью вычислительного эксперимента.

В работе приведены результаты численного моделирования процесса распространения импульса в планарном волноводе.

Установлен факт существенного преимущества одного способа задания входного импульса над другим.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 7](#_Toc165904354)

[1 Постановка задачи и цель работы 9](#_Toc165904355)

[1.1 Описание задачи захвата движения в общем смысле 9](#_Toc165904356)

[1.1.1 Захват движений с помощью специальных маркеров 10](#_Toc165904357)

[1.1.2 Безмаркерный захват движения 11](#_Toc165904358)

[1.2 Задача захвата движения с точки зрения математики 11](#_Toc165904359)

[1.3 Цель работы 12](#_Toc165904360)

[2 Алгоритмы машинного обучения для захвата движения 13](#_Toc165904361)

[2.1 Свёрточные нейронные сети 13](#_Toc165904362)

[2.1.1 Свёрточные слои 14](#_Toc165904363)

[2.1.2 Слои пулинга 14](#_Toc165904364)

[2.1.3 Функции активации 16](#_Toc165904365)

[2.1.4 Полносвязные слои 16](#_Toc165904366)

[2.2 Рекуррентные нейронные сети 18](#_Toc165904367)

[2.3 Сравнение CNN и RNN/LSTM для захвата движения 20](#_Toc165904368)

[3 Программная реализация алгоритма определения двумерных ключевых точек 22](#_Toc165904369)

[3.1 Подготовка датасета 22](#_Toc165904370)

[3.2 Общий подход и архитектура модели 23](#_Toc165904371)

[3.2.1 Backbone сети 23](#_Toc165904372)

[3.2.2 RoI Pooling 26](#_Toc165904373)

[3.2.3 Классификационные и регрессионные слои, блок ключевых точек 28](#_Toc165904374)

[3.3 Оптимизатор Stochastic Gradient Descent (SGD) 29](#_Toc165904375)

[3.4 Расписание скорости обучения с использованием MultiStepLR 30](#_Toc165904376)

[3.5 Функция потерь и процесс обучения 32](#_Toc165904377)

[3.6 Мониторинг, валидация и регуляризация 33](#_Toc165904378)

[4 Методы преобразования двумерных ключевых точек в трехмерные 36](#_Toc165904379)

[4.1 Трёхмерная реконструкция по нескольким изображениям 36](#_Toc165904380)

[4.2 Одиночное изображение с использованием глубины 37](#_Toc165904381)

[4.3 Использование учебных данных с аннотацией глубины 40](#_Toc165904382)

[4.4 Собственная реализация 40](#_Toc165904383)

[4.4.1 Описание модели MiDaS 40](#_Toc165904384)

[4.4.2 Преобразование 2D ключевых точек в 3D координаты 41](#_Toc165904385)

[5 Результаты и сравнительный анализ с существующими моделями 43](#_Toc165904386)

[5.1 Описание эксперимента 43](#_Toc165904387)

[5.2 Результаты работы модели определения двумерных ключевых точек 44](#_Toc165904388)

[5.3 Результаты получения трёхмерных ключевых точек 46](#_Toc165904389)

[ЗАКЛЮЧЕНИЕ 47](#_Toc165904390)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 48](#_Toc165904391)

[ПРИЛОЖЕНИЕ А Моделирование распространения импульса в среде с переменным коэффициентом преломления 49](#_Toc165904392)

[ПРИЛОЖЕНИЕ Б Фрагменты кода программы 50](#_Toc165904393)

[Б.1 Функция построения аналитического решения для кусочно-гладкого импульса 50](#_Toc165904394)

ВВЕДЕНИЕ

Современные исследования в области компьютерного зрения и машинного обучения актуальны благодаря широкому спектру применения этих технологий в различных индустриях, включая автоматизированное видеонаблюдение, интерактивные системы, спортивный анализ и реабилитацию после травм. Особенно значимыми становятся методы захвата и анализа движений человека, которые позволяют улучшить интерфейсы человеко-машинного взаимодействия и повысить точность биомеханических исследований.

Для анализа движений человека на видео часто используются алгоритмы машинного обучения, которые могут автоматически распознавать и классифицировать различные типы движений из видеоданных. Основным преимуществом этих алгоритмов является способность обучаться на больших объемах данных и адаптироваться к новым, ранее неизвестным условиям, что делает их идеально подходящими для задач компьютерного зрения.

В данной работе рассматривается задача разработки и апробации программного средства, основанного на методах машинного обучения, для захвата и анализа движения человека в видеопотоке.

Данная работа содержит шесть разделов.

В первом разделе рассматривается постановка задачи, вводятся базовые определения, исследуется применение систем захвата движения в реальных задачах и ставится цель работы.

Во втором разделе рассматриваются алгоритмы машинного обучения для решения задачи захвата движения человека. Приводится обоснование выбора определенного алгоритма для реализации собственного решения.

В третьем разделе проводится описание программной реализации алгоритма определения ключевых точек. Подробно описывается модель сети, а также другие компоненты, использующиеся при обучении.

В четвертом разделе приведено описание датасета, на котором происходит обучение модели.

И, наконец, в пятом разделе приведены результаты работы собственной модели определения ключевых точек. Также, провизуализирован результат совместной работы модели MiDaS с собственной моделью определения двумерных ключевых точек.

1. Постановка задачи и цель работы
   1. Описание задачи захвата движения в общем смысле

Захват движения (Motion Capture, MoCap) — это технология, которая позволяет записывать движение объектов, в особенности человеческого тела, и применять эти данные для анимации моделей в 3D-пространстве.

При анимации видеоигр, фильмов и создании контента для виртуальной реальности, технологии захвата движения (рисунок 1) играют ключевую роль, добавляя реализм и интерактивность. В спортивных науках, анализ видеозаписей тренировок и соревнований позволяет тренерам и спортсменам улучшать технику и стратегии. В медицине, особенно в реабилитации, захват движений используется для оценки и коррекции походки пациентов, что критически важно для успешного восстановления.

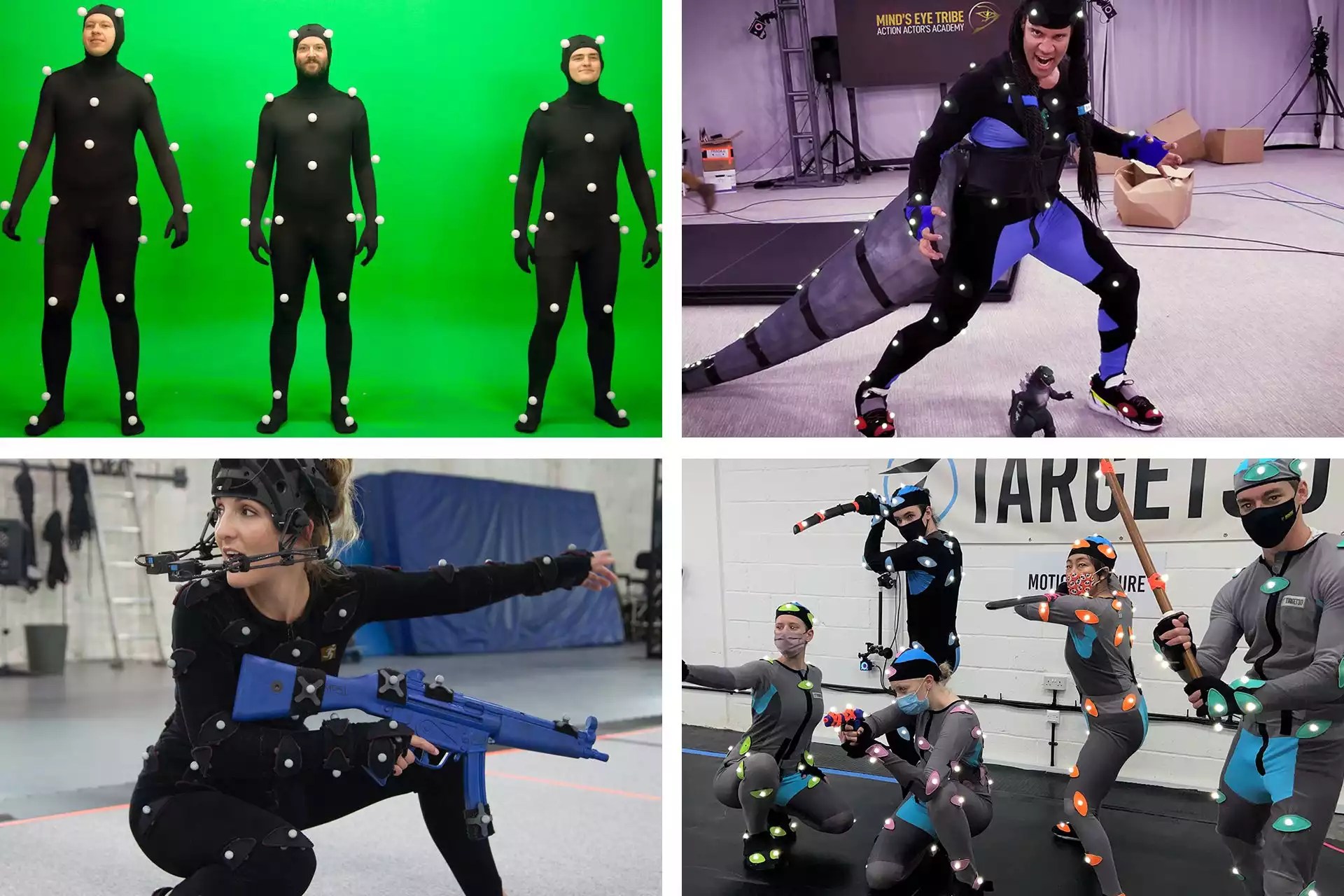


Рисунок 1 – Технология Motion Capture

* + 1. Захват движений с помощью специальных маркеров

Маркерный захват движения — это методика, при которой на теле актера или спортсмена закрепляются специальные маркеры, которые отслеживаются с помощью камер или других датчиков. Эти маркеры могут быть различных типов: отражающие, светящиеся или магнитные. В зависимости от системы захвата, маркеры отслеживаются камерами, которые расположены вокруг зоны захвата, и таким образом записывают точное положение маркеров в пространстве.

На первом этапе на тело испытуемого устанавливаются маркеры. Количество и расположение маркеров могут значительно варьироваться в зависимости от специфики задачи и системы захвата.

Прежде чем начать сессию захвата, необходимо калибровать систему. Это включает настройку камер или датчиков для оптимального захвата движения маркеров.

В процессе записи камеры считывают положение каждого маркера в пространстве. Данные с камер агрегируются и обрабатываются для создания трехмерной модели движения.

После записи сырые данные обрабатываются для построения анимационной модели. Это включает в себя удаление шумов, интерполяцию пропущенных данных и преобразование данных маркеров в координаты скелетной анимации.

Преимуществами захвата движения с помощью физических маркеров являются:

1. высокая точность и детализация захвата;
2. эффективность при сложных движениях, таких как акробатика или детальные мимические движения.

Но, такая система обладает и недостатками:

1. необходимость использования специального оборудования и пространства для захвата;
2. дискомфорт испытуемых из-за необходимости носить маркеры и ограничений, связанных с ними;
3. возможные ошибки из-за перекрытия маркеров или их потери в процессе движения.
   * 1. Безмаркерный захват движения

Бесмаркерный захват движения — это более современная и гибкая альтернатива, которая позволяет анализировать движения без прямого контакта с объектом захвата. Этот метод использует алгоритмы машинного зрения для анализа видеоизображений и выделения ключевых точек тела человека без физических маркеров.

В отличие от маркерных систем, бесмаркерные системы требуют только видеозапись с одной или нескольких камер. С помощью алгоритмов компьютерного зрения происходит распознавание формы тела, определение положения суставов и ключевых точек. На основе полученных данных строится модель движения, которая может быть использована для анимации или анализа.

Преимущества:

1. удобство и доступность использования;
2. возможность анализа естественных движений в реальных условиях без искажений, вызванных наличием маркеров или специального снаряжения.

Недостатки:

1. меньшая точность и детализация по сравнению с маркерными системами;
2. зависимость от качества и условий съемки.
   1. Задача захвата движения с точки зрения математики

Задача определения ключевых точек человеческого тела на видео может быть сформулирована как задача компьютерного зрения, в которой необходимо определить координаты предварительно определенных анатомических меток, таких как суставы, на изображении или в видеоряде. Математически это можно представить следующим образом.

Пусть обозначает кадр из видеопотока, а – набор ключевых точек, которые необходимо определить. Каждая точка описывается своими координатами на изображении . Задача алгоритма машинного обучения – максимизировать [Large-Margin Softmax Loss for Convolutional Neural Networks] вероятность правильного определения этих координат, основываясь на обучающем наборе данных, содержащем аннотированные изображения.

Процесс обучения модели заключается в минимизации функции потерь, которая оценивает разницу между предсказанными алгоритмом координатами и истинными координатами точек на обучающих данных. Одним из популярных выборов для этой функции потерь является сумма квадратов разностей между предсказанными и истинными значениями координат:

*,*

где и – предсказанные координаты точек, а и – истинные координаты.

* 1. Цель работы

Цель данной работы — исследовать применимость модели машинного обучения, способную с высокой точностью определять ключевые точки человеческого тела на изображениях и видео. Основные задачи включают улучшение точности захвата движения при различных условиях освещения, оптимизацию алгоритмов для работы в реальном времени на стандартном оборудовании и минимизацию нужды в ручной корректировке и аннотировании больших объемов данных.

Конечная цель исследования — не только создать эффективную техническую систему, но и продемонстрировать, как такие технологии могут быть интегрированы в реальные прикладные области, предоставляя значимую пользу в медицинских, спортивных и развлекательных приложениях.

1. Алгоритмы машинного обучения для захвата движения
   1. Свёрточные нейронные сети

Convolutional Neural Networks (CNN) являются основным инструментом в современном компьютерном зрении и имеют значительное применение в анализе визуальных данных. В контексте захвата движения, CNN применяются для автоматического распознавания и отслеживания человеческих ключевых точек в последовательности видеокадров.

Принцип работы CNN заключается в автоматическом извлечении признаков из входных изображений посредством операций свёртки, что делает их идеальными для задач, требующих обработку больших и сложных визуальных данных без необходимости ручного задания характеристик. CNN состоят из нескольких типов слоёв (рисунок 2) [СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ]:

1. свёрточные слои (convolutional layers);
2. слои пулинга (pooling layers);
3. функции активации;
4. полносвязные слои (fully connected layers).

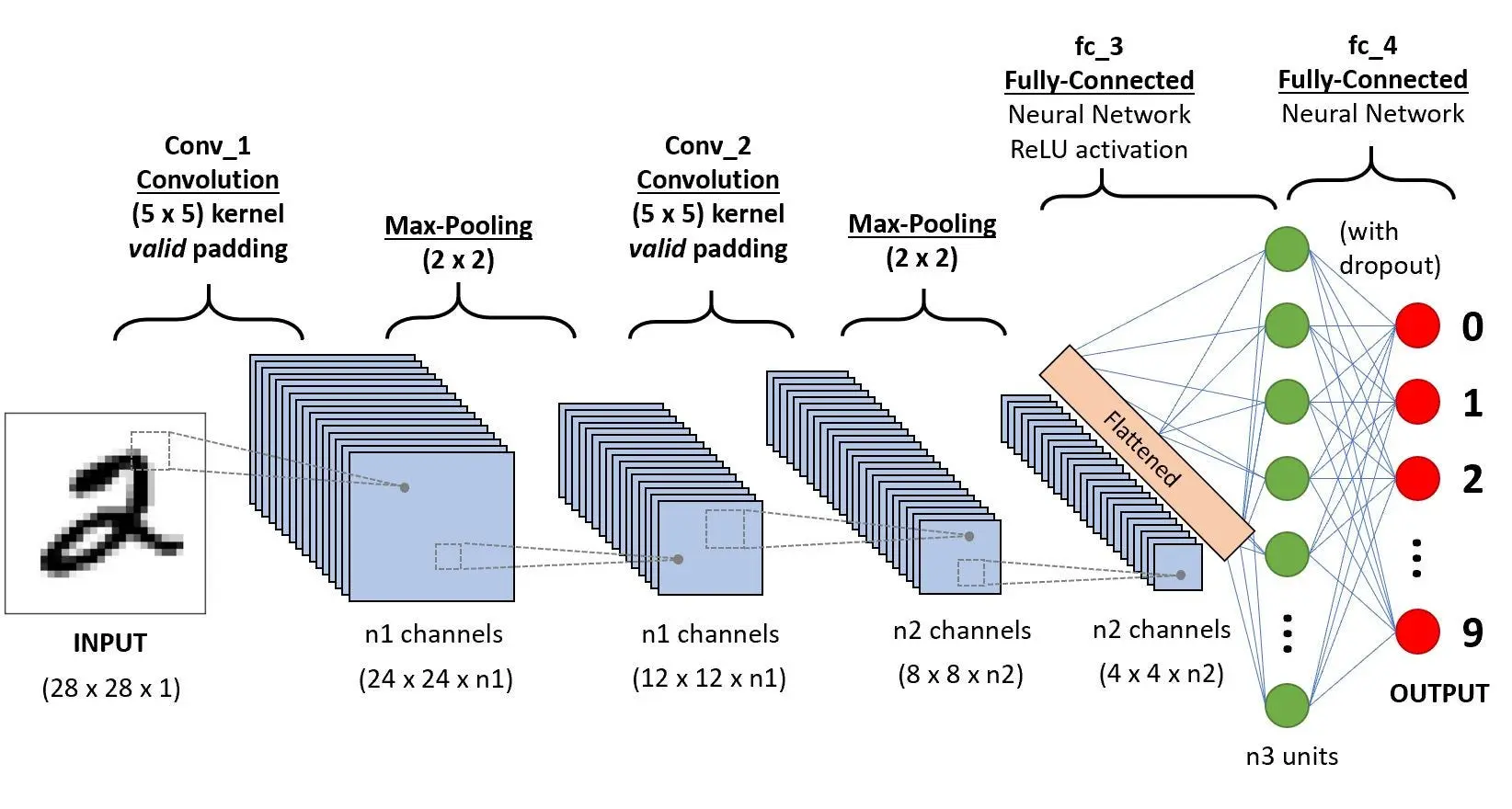


Рисунок 2 – Пример архитектуры свёрточной нейронной сети

* + 1. Свёрточные слои

Свёрточные слои являются фундаментальной составляющей большинства архитектур глубокого обучения, применяемых для анализа изображений и видео. Эти слои используют математическую операцию свертки для выделения важных признаков из входных данных, что делает их исключительно эффективными для задач компьютерного зрения.

Свёрточный слой использует набор фильтров или ядер, которые перемещаются по всему входному изображению, применяя математическую операцию свертки. Эта операция позволяет извлекать изображения из входных данных, создавая карты признаков, которые представляют собой агрегированную информацию о наличии определённых характеристик в различных регионах изображения [Convolutional neural networks: an overview and application in radiology].

В качестве входных данных выступают многоканальные матрицы данных, где каждый канал соответствует определенному цветовому каналу (например, RGB). Свертка применяется отдельно к каждому каналу, а результаты суммируются для получения итоговой карты признаков для каждого фильтра.

Математическая формулировка свёртки [ТОЧЕЧНОЕ МОДЕЛИРОВАНИЕ ОПЕРАЦИИ СВЕРТКИ]: пусть – входное изображение с размерами , где – количество каналов (например, для RGB ). Фильтр свертки имеет размеры . Свертка вычисляется, как:

,

где – координаты элемента на выходной карте признаков, который получается после применения фильтра.

* + 1. Слои пулинга

Слои пулинга, или слои субдискретизации, представляют собой ключевой компонент сверточных нейронных сетей, используемых для уменьшения размерности пространственных данных. Эти слои следуют непосредственно за сверточными слоями и играют важную роль в обеспечении инвариантности сети к масштабированию и другим искажениям изображения.

Суть пулинга заключается в применении операции уменьшения размерности к отдельным сегментам карт признаков, полученных после сверточных слоев. Это достигается путем применения агрегирующей функции, такой как максимум или среднее, к каждому такому сегменту.

Самый распространенный тип пулинга — максимальный пулинг [Max-Pooling Dropout for Regularization of Convolutional Neural Networks]. В его рамках из каждого рассматриваемого подмножества входных данных выбирается максимальное значение. Например, если применять максимальный пулинг с размером окна 2x2 и шагом 2 к матрице признаков, каждое неперекрывающееся подокно 2x2 в этой матрице будет уменьшено до одного значения, равного максимальному из четырех.

Математически операция максимального пулинга для матрицы признаков с размером окна и шагом можем быть описана следующим образом:

1. Разделить матрицу на неперекрывающиеся подматрицы размером ;
2. Для каждой подматрицы найти максимальное значение:

,

где и – индексы в результирующей матрице признаков , которая будет иметь уменьшенные пространственные размеры.

Представим, что у нас есть матрица признаков следующего вида:

Применение операции максимального пулинга с размером окна и шагом приведет к следующей матрице:

Слои пулинга помогают уменьшить количество параметров и вычислений в сети, что способствует борьбе с переобучением и ускоряет обучение. Кроме того, благодаря уменьшению размерности, нейросеть становится менее чувствительной к местоположению объектов во входном изображении, что улучшает её способность к обобщению [СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ].

* + 1. Функции активации

Функции активации в искусственных нейронных сетях — это математические уравнения, которые определяют выходной сигнал нейрона на основе суммы входных сигналов. Эти функции необходимы для введения нелинейности в модель, что позволяет нейронной сети обучаться и выполнять более сложные задачи, чем просто линейная регрессия или классификация. Без нелинейности, какой бы сложной не была архитектура сети, она все равно оставалась бы линейным моделью, что значительно снижает её способность к обучению и аппроксимации функций.

Функция активации применяется к каждому нейрону в сети и определяет, насколько активным будет нейрон при данном входе. Проще говоря, это функция, которая включает или выключает нейрон. Рассмотрим этот процесс на примере одной из самых популярных функций активации — Rectified Linear Unit (ReLU) [ВЫБОР ФУНКЦИИ АКТИВАЦИИ НЕЙРОННОЙ СЕТИ В ЗАВИСИМОСТИ ОТ УСЛОВИЙ ЗАДАЧИ].

Формула для ReLU:

Эта функция активации принимает один вход и выдает , если положительное, и, если отрицательное.

ReLU широко используется, потому что она проста в вычислении и помогает уменьшить вероятность исчезающего градиента, что часто встречается при использовании таких функций активации, как сигмоид или гиперболический тангенс. Важно отметить, что ReLU активирует нейроны только тогда, когда на входе есть активация, что делает нейронные сети разреженными, увеличивая тем самым эффективность и уменьшая вычислительные затраты.

* + 1. Полносвязные слои

Полносвязные слои, также известные как плотные слои (dense layers), представляют собой основные строительные блоки в архитектуре нейронных сетей. В этих слоях каждый нейрон предыдущего слоя соединён с каждым нейроном следующего слоя, что создаёт полную связность между слоями. Это позволяет сети интегрировать информацию, полученную на предыдущих этапах, для выполнения конкретных задач, таких как классификация или регрессия [A ReLU dense layer to improve the performance of neural networks].

В контексте свёрточных нейронных сетей (CNN), полносвязные слои обычно располагаются в конце архитектуры после последовательности свёрточных и пулинг слоев. Основная функция этих слоев в CNN — синтезировать данные, извлечённые из предыдущих слоев, в предсказания, которые могут быть использованы для классификации или других типов вывода.

Каждый нейрон в полносвязном слое получает входы от всех активаций предыдущего слоя, умножает их на соответствующие веса, добавляет смещение (bias) и пропускает через функцию активации для получения выходного значения. Это можно математически представить следующим образом [Frequency Bias in Neural Networks for Input of Non-Uniform Density]:

,

где – вектор входных активаций из предыдущего слоя,

– матрица весов,

– вектор смещений,

– функция активации,

– вектор выходных активаций.

Рассмотрим полносвязный слой, который принимает входной вектор из трёх элементов и имеет два выходных нейрона. Предположим, что используется функция активации ReLU.

Инициализация:

,

,

Вычисление:

Выход первого нейрона:

Выход второго нейрона:

Полносвязные слои широко используются во многих областях машинного обучения и глубокого обучения, включая:

* **Классификация изображений:** после извлечения признаков через свёрточные и пулинг слои, полносвязные слои используются для классификации изображений на основе этих признаков.
* **Регрессия:** в задачах, где требуется предсказать непрерывные значения, такие как цены на дома или температура, полносвязные слои могут обрабатывать признаки для предсказания этих значений.
* **Усиление признаков в сложных задачах:** в задачах с множеством переменных, таких как распознавание речи или машинный перевод, полносвязные слои помогают интегрировать и абстрагировать информацию на высоком уровне.

Эти слои обеспечивают возможность выражения сложных взаимосвязей между входными данными и желаемыми выходными значениями, что делает их неотъемлемой частью многих архитектур глубокого обучения.

* 1. Рекуррентные нейронные сети

Рекуррентные нейронные сети (рисунок 3) представляют собой класс нейросетевых моделей, специализированных на обработке последовательных данных, благодаря своей уникальной способности передавать информацию через временные шаги. Это достигается за счёт внутреннего состояния (или "памяти"), которое позволяет сети удерживать информацию о предыдущих данных в последовательности, делая RNN особенно подходящими для задач, где контекст имеет решающее значение, например, в анализе временных рядов, обработке естественного языка, синтезе речи и других.

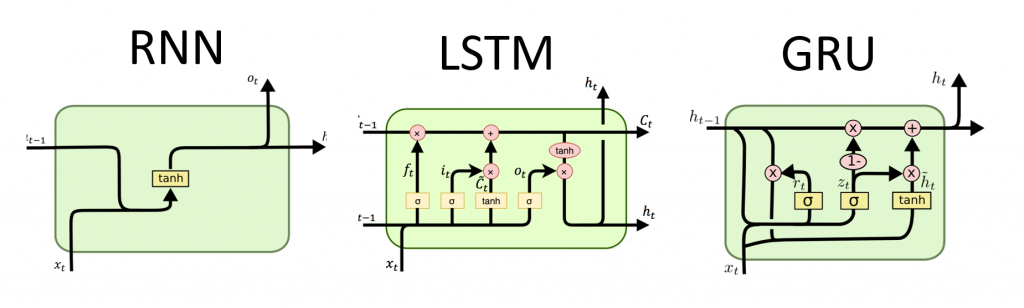


Рисунок 3 – Примерная архитектура рекуррентных нейронных сетей [A Scalable Data-driven platform to Detect Anomalies in Live Cellular Network Using Recurrent Convolutional Neural Network]

Одним из расширений RNN являются сети с долгой краткосрочной памятью (LSTM), которые предназначены для решения проблемы исчезающего градиента, с которой сталкиваются традиционные RNN. LSTM достигает этого за счёт использования затворов (ворот), которые контролируют поток информации.

Ворота забывания определяют, какая часть информации из предыдущего состояния ячейки должна быть забыта:

Входные ворота и кандидат ячейки определяют, какая новая информация будет сохранена в состоянии ячейки:

,

Обновление состояния ячейки комбинирует старое состояние с новыми кандидатами, обновляя текущее состояние ячейки:

Ворота вывода и выходное состояние определяют следующее скрытое состояние, которое будет использоваться в следующем временном шаге или для предсказания [Рекуррентные нейронные сети]:

RNN и их вариации, включая LSTM и GRU, находят широкое применение во многих областях, где требуется анализ последовательностей данных:

* RNN могут анализировать исторические данные временных рядов, такие как акции или погодные данные, для предсказания будущих значений;
* в задачах NLP, таких как машинный перевод или распознавание речи, RNN могут улавливать семантические и синтаксические зависимости в тексте;
* используя последовательности нот и аккордов, RNN способны генерировать музыкальные произведения;
* RNN используются для понимания и предсказания действий и событий в видеопотоках, что может быть применено в системах видеонаблюдения или интерактивных мультимедийных системах.
  1. Сравнение CNN и RNN/LSTM для захвата движения

Эффективность применения CNN и RNN/LSTM в этой задаче варьируется в зависимости от конкретных требований и условий применения.

Свёрточные нейронные сети идеально подходят для анализа визуальной информации благодаря своей способности эффективно обрабатывать пространственные отношения и текстуры на изображениях. В контексте MoCap, CNN могут быть использованы для точного определения ключевых точек и сегментов тела человека в различных позах и условиях освещения. Благодаря своим свёрточным слоям, CNN способны автоматически выделять важные признаки из видеоданных, что значительно упрощает процесс обучения и уменьшает необходимость в ручном извлечении признаков и их предварительной обработке.

Однако, основным недостатком CNN является их неспособность учитывать временные зависимости в данных. В контексте захвата движения, где важно анализировать последовательность движений и каждое последующее состояние может зависеть от предыдущих, этот недостаток может существенно ограничивать применимость CNN.

В отличие от CNN, рекуррентные нейронные сети (RNN) и их усовершенствованный вариант — LSTM (Long Short-Term Memory) — разработаны специально для работы с последовательными данными. Эти сети способны обрабатывать информацию о предыдущих состояниях, что делает их идеальными для задач, где необходимо учитывать контекст, таких как анализ сложных движений и жестов в MoCap. LSTM улучшает возможности базовых RNN за счет механизмов забывания и сохранения информации, что позволяет эффективно управлять потоком информации и избегать проблемы исчезающих градиентов.

RNN и особенно LSTM требуют значительных вычислительных ресурсов для обучения, что может стать проблемой при работе с большими объемами данных, характерными для MoCap. Кроме того, настройка и оптимизация этих сетей может быть сложной из-за большого количества параметров и настроек, что увеличивает сложность моделей и время, необходимое для их тонкой настройки.

Выбор между CNN и RNN/LSTM для задач захвата движения должен базироваться на конкретных требованиях к задаче:

1. если важно пространственное распознавание и высокая точность в определении позы на отдельных кадрах, CNN будет предпочтительным выбором;
2. для анализа сложных движений, требующих учета временных зависимостей и последовательностей, более подходящим выбором окажется использование RNN или LSTM.

В идеальном сценарии комбинирование CNN для пространственного анализа и LSTM для обработки временной последовательности движений может дать наилучшие результаты. Такой подход позволит максимально использовать преимущества обеих технологий, обеспечивая точное распознавание поз и анализ сложных движений с учетом временной динамики.

1. Программная реализация алгоритма определения двумерных ключевых точек
   1. Подготовка датасета

Датасеты для обучения и оценки моделей компьютерного зрения играют критически важную роль в разработке и улучшении алгоритмов, способных анализировать и интерпретировать визуальную информацию. Ключевым аспектом этих датасетов является их способность предоставлять разнообразные данные, которые отражают множество сценариев использования в реальном мире.

Современные датасеты включают изображения с различными уровнями сложности. Это может включать в себя разнообразные фоны, условия освещения и конфигурации объектов. Например, уличные сцены содержат множество перекрывающихся объектов, разные типы движения и разнообразные погодные условия. Датасеты типа ImageNet или COCO включают изображения, которые захватывают эти сложности, помогая моделям учиться на данных, максимально приближенных к реальности [Computer vision for autonomous vehicles: Problems, datasets and state of the art].

Для распознавания и локализации объектов важно, чтобы датасеты включали широкий спектр объектов различных форм, размеров и категорий. Например, датасет Pascal VOC включает объекты, такие как велосипеды, машины, люди и животные, каждый из которых имеет уникальные атрибуты и формы. Это важно для обучения моделей распознавать и точно классифицировать объекты в различных контекстах.

Хорошо аннотированные датасеты предоставляют точные и подробные метки, которые необходимы для обучения и оценки моделей. Это включает не только метки классов, но и ограничивающие рамки, аннотации ключевых точек и маски сегментации. Например, датасет COCO [Microsoft coco: Common objects in context] предоставляет аннотации для сегментации на уровне пикселей и ключевые точки для анализа человеческой позы, что позволяет использовать его для задач, требующих детального понимания визуального контента.

Разнообразие в условиях съемки, таких как освещение, ракурс и качество изображения, также критически важно. Модели становятся устойчивыми к изменениям во входных данных, которые неизбежны в реальных приложениях. Датасеты, такие как ImageNet [Imagenet: A large-scale hierarchical image database], включают изображения с различным разрешением и качеством, что помогает улучшить устойчивость и надежность обученных моделей.

Датасеты должны отражать реальные сценарии применения, чтобы обеспечить практическую значимость разработанных моделей. Например, датасеты для автономного вождения, такие как KITTI, предоставляют данные, собранные с датчиков на автомобилях, что помогает в разработке систем, способных работать в реальных дорожных условиях.

В рамках данной работы был использован датасет COCO (Common Objects in Context), который является одним из стандартов в индустрии для задач детекции, сегментации и распознавания ключевых точек. COCO содержит более 330 тысяч изображений с более чем 200 тысячами меток, включающих ограничивающие рамки, сегментационные маски и ключевые точки. Этот датасет идеально подходит для наших нужд, так как предоставляет комплексные и разнообразные данные, которые могут помочь в точной настройке и оценке нашей модели.

Кроме того, используется аугментации данных и дополнительные техники предварительной обработки для улучшения обучения и обобщающей способности модели. Это включает в себя масштабирование, повороты и изменение цветовых настроек изображений, что позволяет модели лучше адаптироваться к разнообразным условиям и улучшить ее способность к обобщению на новые, невиданные данные.

* 1. Общий подход и архитектура модели
     1. Backbone сети

Backbone сеть в архитектуре Keypoint R-CNN является фундаментальным компонентом, который отвечает за первичное извлечение признаков из входных изображений. В контексте глубокого обучения, backbone — это обычно предварительно обученная сверточная нейронная сеть (CNN), которая преобразует исходные изображения в сложный набор признаков или карт признаков. Эти карты признаков служат основой для всех последующих этапов анализа и обработки, таких как детекция объектов, классификация и определение ключевых точек.

Распространенные архитектуры, используемые в качестве backbone в задачах компьютерного зрения, включают [Res2Net: A New Multi-scale Backbone Architecture]:

* ResNet (Residual Networks): Эта сеть использует так называемые "остаточные блоки", которые помогают обучать очень глубокие нейронные сети. Основное математическое выражение для слоя в ResNet выглядит следующим образом: , где – входной вектор признаков на уровне , – остаточная функция, а – веса, которые необходимо обучить [Resnet in Resnet: Generalizing Residual Architectures].
* VGG (Visual Geometry Group): Простая и мощная архитектура, основанная на повторении блоков, состоящих из сверточных слоев с маленьким размером фильтра (3x3), за которыми следуют слои пулинга. В VGG все слои используют одинаковый шаг и дополнение нулями, что позволяет сохранять пространственные размеры через слои [Going deeper in spiking neural networks: VGG and residual architectures].

Feature Pyramid Network (FPN) является дополнением к стандартному backbone и предназначена для улучшения детекции объектов различного размера. FPN создает иерархию признаков на разных уровнях разрешения, что позволяет модели лучше адаптироваться к объектам разных масштабов. В FPN каждый уровень пирамиды создается путем слияния информации из двух источников: нижнего уровня с высоким разрешением и верхнего уровня с более глубокой семантической информацией. Это достигается с помощью операций свертки и апсемплинга [Effective Fusion Factor in FPN for Tiny Object Detection]:

,

где – признаки на уровне пирамиды, – выходные данные сверточного слоя на уровне , и – операция апсемплинга [Guided Upsampling Network for Real-Time Semantic Segmentation].

В архитектуре Keypoint R-CNN используется ResNet50 в качестве backbone сети, основываясь на её эффективности (таблица 1) и универсальности в различных задачах компьютерного зрения. ResNet50, с её остаточными блоками, способствует упрощению обучения глубоких сетей, предотвращая исчезновение градиента, что критически важно для обеспечения стабильности и высокой производительности модели.

Таблица 1 - Эффективность оценки ключевых точек на валидационном наборе COCO [https://arxiv.org/pdf/1904.01169]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Backbone |  |  |  |  |  |
| ResNet-50 | 70,4 | 88,6 | 78,3 | 67,1 | 77,2 |
| Res2Net-50 | 71,5 | 89,0 | 79,3 | 68,2 | 78,4 |
| ResNet-101 | 71,4 | 89,3 | 79,3 | 68,1 | 78,1 |
| Res2Net-101 | 72,2 | 89,4 | 79,8 | 68,9 | 79,2 |
| Res2Net-vlb-50 | 72,2 | 89,5 | 79,7 | 58,5 | 79,4 |
| Res2Net-vlb-101 | 73,0 | 89,5 | 80,3 | 69,5 | 80,0 |

Таблица 1 представляет сравнение различных архитектур backbone на валидационном наборе данных COCO для оценки эффективности определения ключевых точек. Оценки включают общую точность (), точность при IoU пороге 0.5 () и 0.75 (), а также точность для средних () и больших () объектов.

ResNet-50 показывает средние результаты с общей точностью 70,4, что ниже, чем у более сложных моделей. Однако, данная модель уже интегрирована в архитектуру Keypoint R-CNN как *keypointrcnn\_resnet50\_fpn* в среде PyTorch, что делает её простой в использовании для начальных этапов разработки и экспериментов, несмотря на то, что она не обеспечивает максимально возможной точности.

Res2Net-50 и Res2Net-101 обеспечивают лучшую общую точность, чем их аналоги ResNet, со значениями 71,5 и 72,2 соответственно. Это указывает на то, что более сложные модификации ResNet архитектуры могут лучше справляться с задачами определения ключевых точек благодаря улучшенному извлечению признаков и более глубокой обработке информации.

Res2Net-vlb-50 и Res2Net-vlb-101 показывают наивысшие результаты во всех категориях, с достигающим 73,0 для Res2Net-vlb-101. Это подчеркивает преимущества использования очень глубоких сетей с расширенной архитектурой для сложных задач определения ключевых точек.

* + 1. RoI Pooling

RoI (Region of Interest) Pooling — это техника, используемая в сверточных нейронных сетях для извлечения фиксированного размера признаков из произвольно размерных регионов. Эта операция критически важна в задачах, где необходимо обработать локальные области изображения, такие как детекция объектов и сегментация. RoI Pooling был популяризирован архитектурами, такими как Fast R-CNN, для улучшения производительности моделей в этих задачах [Improvement in object detection using multi-scale RoI pooling and feature pyramid network].

На рисунке 4 можно посмотреть на результат операции определения RoI.

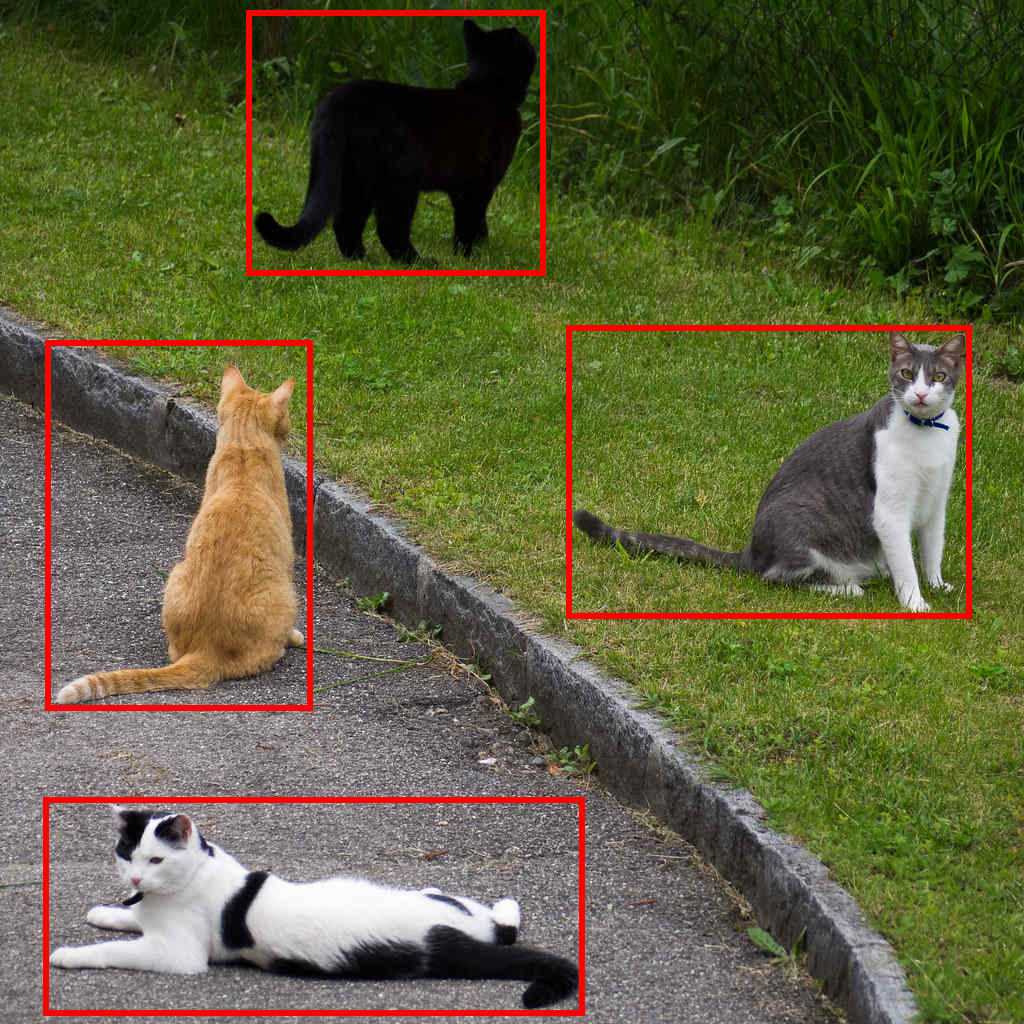


Рисунок 4 – Визуализация результатов работы определения RoI

Рассмотрим входную карту признаков размером , где и обозначают высоту, ширину и глубину карты признаков соответственно. Пусть дано регионов интереса, каждый из которых задан четырьмя координатами: , где и обозначают верхний левый и нижний правый углы региона на карте признаков.

Для каждого региона RoI Pooling операция выполняется следующим образом [ROI Pooled Correlation Filters for Visual Tracking]:

1. регион преобразуется к фиксированному размеру , который задается заранее. Это делается для стандартизации выходных данных, позволяя использовать их на следующих слоях нейронной сети;
2. преобразованный регион делится на ячеек равного размера;
3. в каждой ячейке выполняется операция пулинга для получения одного значения признака. Пулинг [A Theoretical Analysis of Feature Pooling in Visual Recognition] определяется, как:

,

где – это значение признака в ячейке , а – множество пикселей в ячейке входной карты признаков.

* + 1. Классификационные и регрессионные слои, блок ключевых точек

После извлечения признаков с помощью RoI Pooling, следующий шаг заключается в использовании классификационных и регрессионных слоев для определения класса объектов и точного позиционирования их ограничивающих рамок. Классификационный слой применяет функцию softmax для оценки вероятности принадлежности каждого предложенного региона к одному из возможных классов объектов:

,

где – вероятность класса для региона , – сырые оценки классификации для класса , а – общее количество классов.

Регрессионный слой затем адаптирует ограничивающие рамки, предложенные RPN, для того чтобы они как можно точнее обрамляли детектируемый объект. Это достигается путем корректировки четырех параметров рамки: центра , ширины и высоты . Преобразование координат рамки выражается через:

,

где - предсказания регрессии относительно исходной рамки [A High-Speed and Low-Complexity Architecture for Softmax Function in Deep Learning].

После классификации объектов и уточнения их рамок, выполняется задача определения ключевых точек. Каждая детектированная рамка объекта снабжается набором признаков, из которых затем предсказываются ключевые точки. Это достигается с помощью регрессии для каждой точки внутри рамки:

,

где – координаты предсказанной ключевой точки, – исходные координаты точки в рамке, а – смещения, предсказанные сетью [Local keypoint-based Faster R-CNN].

* 1. Оптимизатор Stochastic Gradient Descent (SGD)

Оптимизатор Stochastic Gradient Descent (SGD) — это один из основных методов оптимизации, используемых в обучении нейронных сетей и других вычислительных задачах, связанных с минимизацией функций. SGD модифицирует традиционный метод градиентного спуска путём обновления параметров модели на основе оценки градиента, полученной из случайно выбранного подмножества всего набора данных [Is local SGD better than minibatch SGD?].

Формула для обновления весов в SGD:

,

где – параметры модели на шаге ,

– скорость обучения, положительный коэффициент, определяющий размер шага в пространстве параметров,

– градиент функции потерь по параметрам на шаге , полученный из текущего батча данных.

В классическом градиентном спуске (Gradient Descent, GD) для обновления параметров модели используется весь набор данных для вычисления градиента функции потерь. Это требует значительных вычислительных ресурсов и времени, особенно при работе с большими данными. В отличие от GD, SGD обновляет параметры, используя только один обучающий пример за раз. Это делает SGD значительно быстрее, особенно в контекстах, где данные имеют высокую избыточность [SGD Generalizes Better Than GD (And Regularization Doesn’t Help)].

Быстрота итераций также является ключевым преимуществом SGD. Каждое обновление параметров происходит гораздо быстрее, поскольку для этого необходимо вычислить градиенты только по одному образцу данных.

SGD идеально подходит для задач, где данные поступают последовательно, например, в системах рекомендаций или для персонализированных приложений, где модель должна адаптироваться к новым пользователям или меняющимся предпочтениям пользователя в реальном времени. Это обеспечивается возможностью модели обновляться постоянно, по мере поступления каждого нового данных, без необходимости повторного обучения с нуля.

Помимо достоинств, SGD обладает и рядом недостатков. Проблемы сходимости — один из основных недостатков SGD. Из-за того, что градиенты вычисляются исходя из одного образца, результаты могут сильно колебаться между итерациями. Это может привести к тому, что процесс обучения будет менее стабильным и потребует большего количества итераций для достижения сходимости, особенно если выбранный размер шага обучения не оптимален.

Риск застревания в локальных минимумах или седловых точках особенно актуален в контекстах, где функция потерь не является выпуклой. В таких случаях SGD может "застрять", не найдя глобальный минимум ошибки, который оптимален для всех данных. Это связано с тем, что локальные шумы и несовершенства данных могут ввести модель в заблуждение, указывая на неправильное направление обновлений.

* 1. Расписание скорости обучения с использованием MultiStepLR

MultiStepLR — это метод адаптации скорости обучения в процессе тренировки нейронной сети, реализуемый в библиотеке *PyTorch* через модуль *torch.optim.lr\_scheduler*. Этот метод позволяет планомерно уменьшать скорость обучения в предопределённые моменты тренировки, что часто приводит к более эффективному и стабильному обучению моделей глубокого обучения.

Скорость обучения (learning rate, LR) — один из наиболее важных гиперпараметров при тренировке нейронных сетей. Она определяет величину шага, с которым обновляются веса модели в направлении антиградиента функции потерь. Слишком большая скорость обучения может привести к тому, что обучение будет "перепрыгивать" через минимумы функции потерь, в то время как слишком маленькая скорость замедлит процесс обучения и может привести к застреванию в локальных минимумах.

*MultiStepLR* работает, уменьшая скорость обучения на заданный коэффициент γ в заданные моменты времени, что обычно совпадает с эпохами. Эти "шаги" или "ступеньки" уменьшения скорости обучения задаются списком эпох. Когда текущая эпоха обучения достигает одной из указанных эпох в списке, скорость обучения умножается на коэффициент .

Преимущества использования MultiStepLR включают возможность настройки контрольных точек для изменения скорости обучения, что позволяет адаптировать процесс обучения под конкретные этапы развития модели. Это особенно важно, поскольку разные фазы тренировки могут требовать различной скорости обучения для оптимальной эффективности. Кроме того, использование MultiStepLR способствует преодолению плато в обучении, когда прогресс модели замедляется и стандартная постоянная скорость обучения не приносит желаемых результатов. Постепенное уменьшение скорости обучения может также улучшить сходимость, позволяя модели более стабильно и глубоко аппроксимировать оптимальные значения весов, что снижает риск осцилляций вокруг минимума функции потерь.

Однако использование MultiStepLR не лишено недостатков. Одной из основных проблем является необходимость ручной настройки моментов изменения скорости и коэффициента уменьшения, что может потребовать значительных усилий и времени на эксперименты, особенно в начале обучения. Эффективность конкретного расписания скорости обучения может также значительно варьироваться в зависимости от специфики задачи и используемых данных, что ограничивает универсальность данного подхода. Кроме того, существует риск застревания в локальных минимумах или на седловых точках, если скорость обучения будет снижена слишком рано или если начальная скорость обучения была установлена слишком низкой, что может препятствовать достижению глобального минимума функции потерь [Effective Neural Network Training with Adaptive Learning Rate based on Training Loss].

* 1. Функция потерь и процесс обучения

Функция потерь, используемая в данной модели, включает в себя несколько компонентов, предназначенных для различных аспектов задачи. Основные используемые функции потерь:

* Smooth L1 Loss - это вариант функции потерь, который обычно используется для задач регрессии, таких как определение ограничивающих рамок в задачах детекции объектов. Эта функция потерь менее чувствительна к выбросам по сравнению с традиционной L2 потерей благодаря своей способности уменьшать влияние больших ошибок на процесс обучения. Формула Smooth L1 Loss представлена как [Oriented Object Detection in Aerial Images Based on the Scaled Smooth L1 Loss Function]:

,

где – разница между предсказанным значением и истинным значением.

* Cross-Entropy Loss - часто применяется в задачах классификации. Эта функция измеряет различие между двумя вероятностными распределениями: предсказанным и истинным. Для двухклассовой классификации функция потерь вычисляется как [Cross-entropy loss functions: Theoretical analysis and applications]:

,

где – предсказанная вероятность принадлежности объекта к классу, а – истинная метка класса (0 или 1).

Процесс обучения модели разделяется на несколько ключевых этапов:

1. прямое распространение: на этом этапе входные данные подаются в модель, проходят последовательно через все слои сети, в результате чего модель генерирует предсказание. Каждый слой сети преобразует входные данные согласно своим обученным параметрам и передает результат следующему слою;
2. вычисление потерь: после получения предсказаний на выходе модели рассчитывается значение функции потерь. Это значение представляет собой оценку того, насколько предсказания модели отличаются от истинных значений.;
3. обратное распространение: с помощью этого метода производится расчет градиентов функции потерь по всем параметрам модели — от выходных слоев к входным. Этот процесс помогает определить вклад каждого параметра в ошибку и определить, как необходимо изменить параметры, чтобы минимизировать функцию потерь;
4. оптимизация: на основе вычисленных градиентов и с использованием метода SGD параметры модели обновляются таким образом, чтобы функция потерь уменьшалась. Оптимизация является итеративным процессом, и каждое обновление параметров приближает модель к оптимальному решению задачи.
   1. Мониторинг, валидация и регуляризация

Одним из основных этапов в процессе обучения является валидация модели. Валидация (рисунок 5) позволяет оценить, как модель будет работать в реальных условиях, используя тестовый набор данных, который не участвовал в обучении. Этот процесс включает в себя периодическую оценку модели на этом наборе данных, чтобы мониторить такие метрики, как точность классификации, точность локализации объектов (в случае задач детекции или сегментации), а также другие метрики, специфичные для конкретной задачи, такие как площадь под ROC-кривой (AUC) или среднее значение точности пересечения по объединению (IoU). Значения IoU после первой эпохи можно увидеть на рисунке 6.

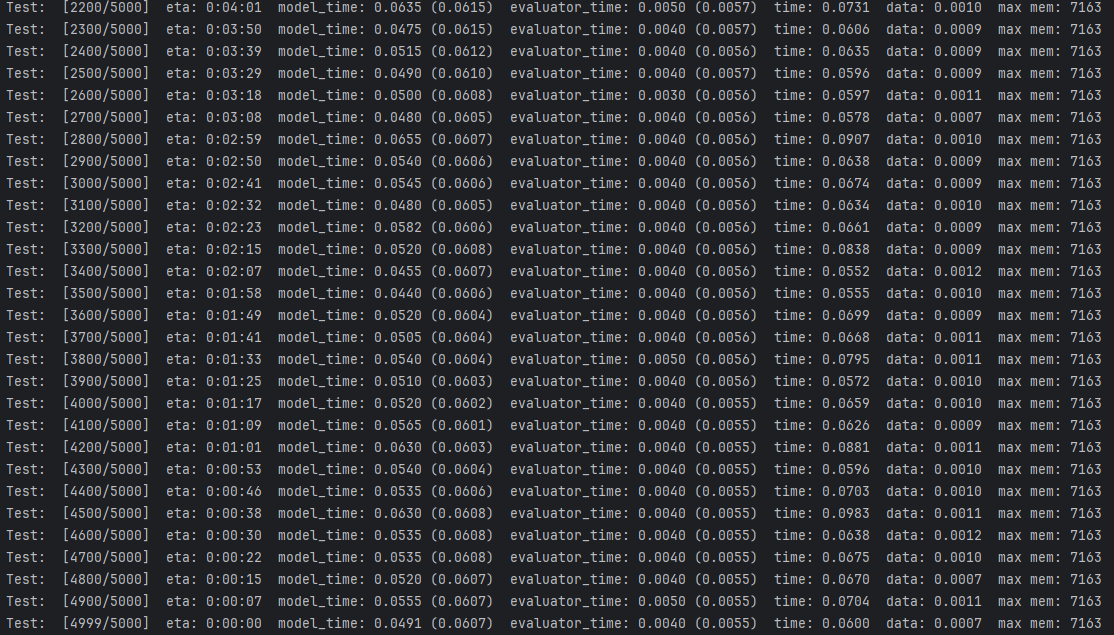


Рисунок 5 – Валидация модели

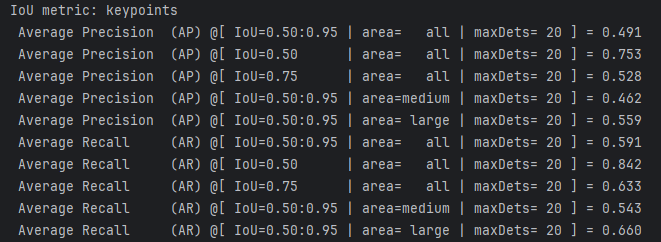


Рисунок 6 – IoU метрики

Проведение регулярных проверок в процессе обучения помогает выявить такие проблемы, как переобучение, когда модель хорошо работает на тренировочных данных, но плохо на тестовых. Это также позволяет разработчикам принять меры для корректировки параметров обучения, выбора модели или самого процесса обучения в реальном времени.

Для предотвращения переобучения часто используются методы регуляризации. Одним из самых популярных методов является L2 регуляризация [L2 regularization versus batch and weight normalization], известная также как Weight Decay. Этот метод добавляет к функции потерь член, пропорциональный квадрату нормы вектора весов модели:

,

где – исходная функция потерь,

– вектор весов модели,

– коэффициент регуляризации, который контролирует степень влияния штрафа на итоговую функцию потерь.

Этот штраф за большие веса помогает предотвратить переобучение, делая модель менее чувствительной к небольшим колебаниям во входных данных, тем самым способствуя лучшей обобщающей способности.

Также широко используется регуляризация Dropout [Dropout: a simple way to prevent neural networks from overfitting], при которой случайным образом исключается часть нейронов в процессе обучения, что помогает уменьшить зависимость модели от конкретных атрибутов входных данных и тем самым увеличить её способность к обобщению.

Комбинация регулярного мониторинга и валидации модели во время её обучения с применением методов регуляризации создает мощную систему, способную эффективно бороться с переобучением. Мониторинг позволяет оперативно отслеживать эффективность модели и делать корректировки в процессе обучения, в то время как регуляризация обеспечивает дополнительную устойчивость модели к переобучению.

1. Методы преобразования двумерных ключевых точек в трехмерные
   1. Трёхмерная реконструкция по нескольким изображениям

Мультивидовая стереоскопия (рисунок 7) является одним из наиболее эффективных подходов для создания детализированных трехмерных моделей из двухмерных изображений. Процесс включает в себя анализ нескольких снимков объекта, сделанных с разных ракурсов, что позволяет воссоздать объемные структуры с высокой точностью [Stereoscopy and the human visual system].

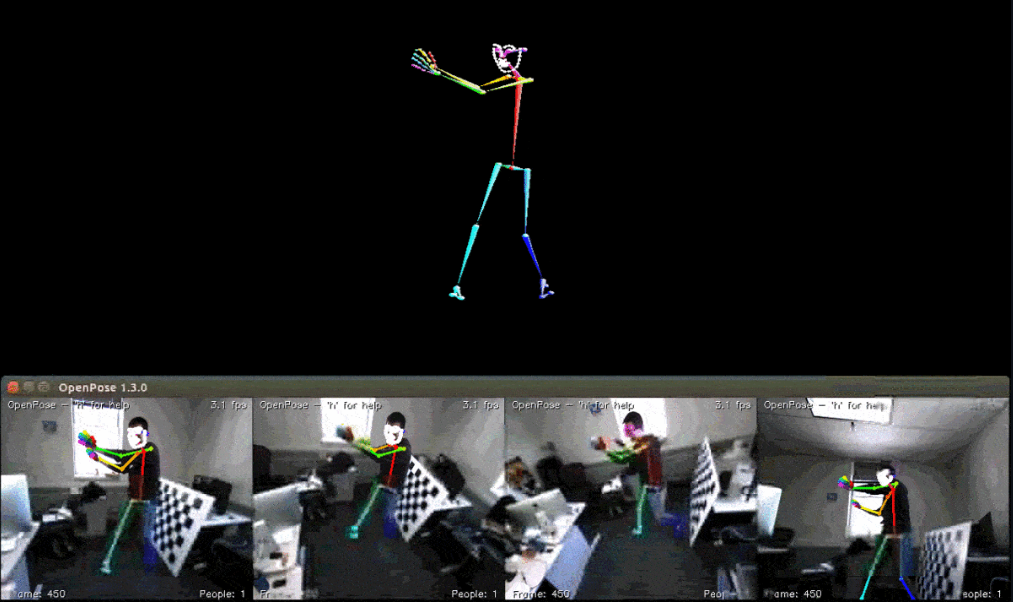


Рисунок 7 – Реконструкция 3D скелета из нескольких изображений

На начальном этапе происходит сбор данных. Данный этап является критически важным этапом, поскольку качество и количество изображений напрямую влияют на точность и детализацию трехмерной модели. Изображения должны быть сняты так, чтобы между ними было достаточное перекрытие, что позволит алгоритмам сопоставления точек находить соответствия с высокой точностью. Обычно используются специализированные камеры, расположенные под разными углами относительно объекта.

Затем, алгоритмы компьютерного зрения, такие как SIFT (Scale-Invariant Feature Transform) или SURF (Speeded Up Robust Features), используются для обнаружения уникальных особенностей на изображениях. Эти особенности должны быть инвариантны к масштабированию, вращению и изменениям освещения, чтобы обеспечить их надежное сопоставление между разными снимками.

Финальным этапом является триангуляция. Триангуляция - это процесс определения положения точки в трехмерном пространстве по её проекциям на несколько изображений. Для каждой сопоставленной точки строится система уравнений, основанная на параметрах проекции каждой камеры и координатах точек на изображениях. Решение этой системы позволяет найти координаты точки в глобальной системе координат. Математически этот процесс можно описать через минимизацию ошибки воспроизведения:

,

где – проекция точки на изображение , а – матрица проекции камеры .

Мультивидовая стереоскопия широко используется в киноиндустрии для создания реалистичных CGI (computer-generated imagery) сцен, в археологии для реконструкции артефактов, в робототехнике для создания точных карт окружающей среды, и в медицине для создания трехмерных моделей анатомических структур [3D tracking of human motion using visual skeletonization and stereoscopic vision].

* 1. Одиночное изображение с использованием глубины

Использование одиночного изображения с картой глубины актуально, когда доступ к мультивидовым настройкам ограничен или невозможен. Этот метод позволяет эффективно восстанавливать пространственную структуру объектов и сцен из одного изображения, используя информацию о глубине каждой точки [Deep learning for monocular depth estimation: A review].

Первым этапом является получение карты глубины. Основная идея данного подхода состоит в том, что глубина пикселя на изображении известна. Но вот возможностей ее получения несколько:

* Стереокамеры: используют две камеры для захвата двух видов одной и той же сцены под немного разными углами. Глубина рассчитывается на основе различий между этими двумя видами, опираясь на принципы триангуляции;
* LiDAR: системы, основанные на LiDAR (рисунок 8), излучают лазерные лучи и измеряют время, необходимое им для возвращения после отражения от объектов. Это время преобразуется в расстояние, создавая высокоточные карты глубины;
* современные методы глубокого обучения могут предсказывать карту глубины (рисунок 9) непосредственно из одиночного изображения на основе обучения модели на больших наборах данных с аннотированной глубиной.

Следующим этапом является преобразование 2D ключевых точек в 3D. Используя карту глубины, преобразование 2D пиксельных координат изображения в 3D пространственные координаты осуществляется через процесс, известный как проекция обратного вида (back-projection). Этот процесс учитывает внутреннюю калибровку камеры, включая фокусное расстояние и центр проекции, для перевода пиксельных координат в реальные координаты. Математически это описывается следующими уравнениями:

,

где – координаты пикселя на изображении, – координаты центра проекции, – фокусное расстояние камеры, – значение глубины для пикселя [Depth estimation from single image and semantic prior].

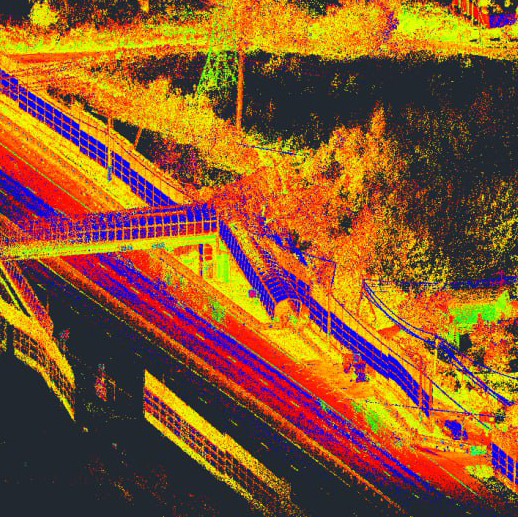


Рисунок 8 – Лидарная съемка

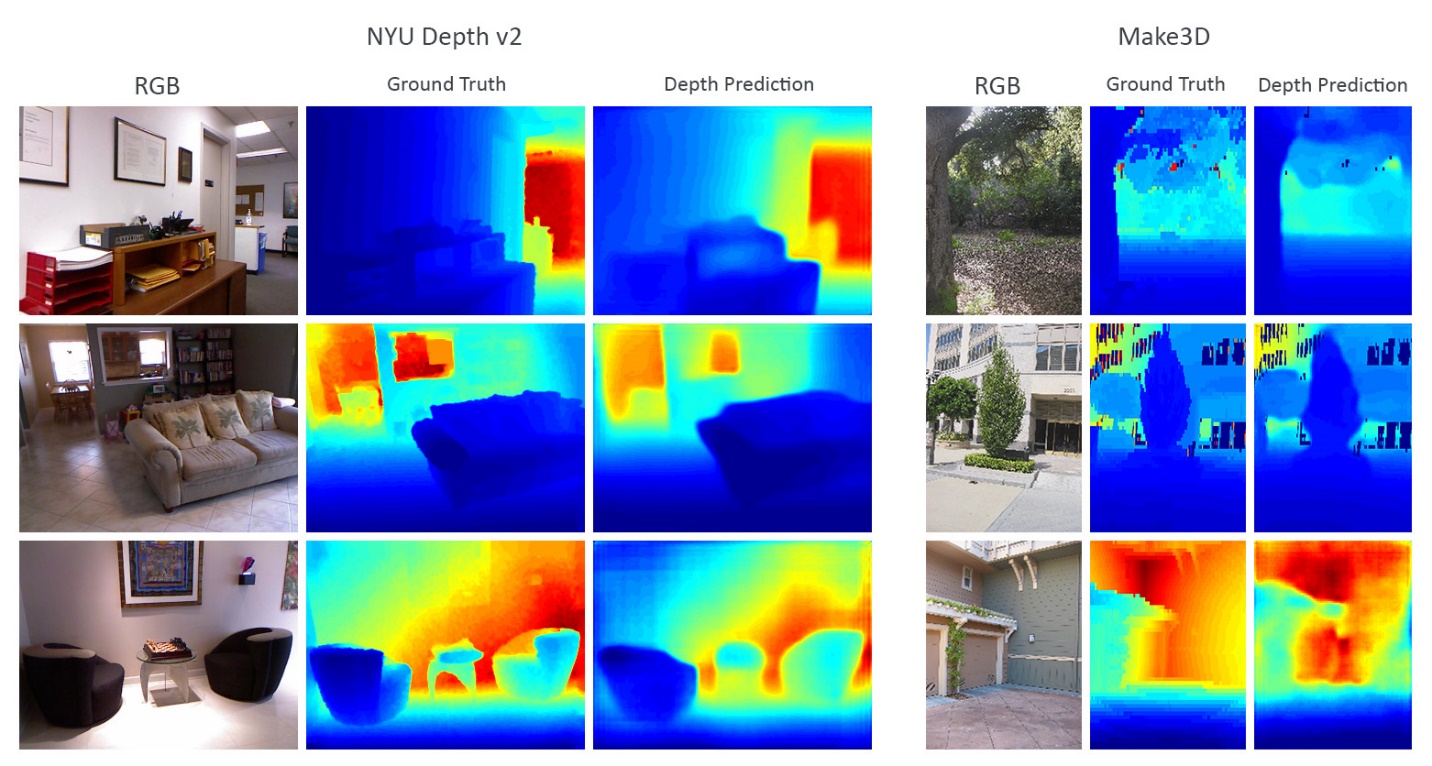


Рисунок 9 – Карта глубины, предсказанная с помощью нейросетей

* 1. Использование учебных данных с аннотацией глубины

В данном методе предполагается, что данные для обучения модели предварительно дополнительно размечены значениями глубины. Он имеет множество преимуществ, таких как: отсутствие необходимости в дополнительных устройствах, высокая мобильность и возможность работы в реальном времени. Но при этом обладает и недостатками: качество напрямую зависит от качество данных для обучения.

* 1. Собственная реализация
     1. Описание модели MiDaS

Для реализации собственного решения была выбрана модель определения глубины MiDaS.

MiDaS (Monocular Depth Estimation via Deep Learning) — это нейросетевая модель, разработанная для оценки глубины сцены из одиночных изображений. Модель была обучена на множестве данных, содержащих изображения с разнообразными сценами, что позволяет ей эффективно определять глубину в различных условиях освещения и композиций.

Принцип ее работы заключается в том, что она использует предобученную на датасете ImageNet сеть для извлечения признаков из входного изображения. Затем, на основе признаков модель предсказывает карту глубины, где каждому пикселю изображения соответствует значение глубины (рисунок 10). Это достигается за счет использования слоев, специально адаптированных для работы с регрессией глубины [Midas v3. 1--a model zoo for robust monocular relative depth estimation].



Рисунок 10 – Результаты работы модели MiDaS

* + 1. Преобразование 2D ключевых точек в 3D координаты

Для трансформации 2D ключевых точек в их 3D координаты используется карта глубины, сгенерированная моделью MiDaS. Каждой ключевой точке, определённой на изображении, соответствует пиксель на карте глубины. Из этого пикселя извлекается значение глубины, которое представляет собой расстояние от камеры до объекта в данной точке.

На практике, значения глубины могут содержать шумы или неточности из-за ограничений самой модели глубины или из-за визуальных аномалий в исходном изображении. Эти неточности могут привести к ошибкам в определении истинного 3D положения точек. Чтобы уменьшить эти ошибки, применяется процесс коррекции [2d3d-matchnet: Learning to match keypoints across 2d image and 3d point cloud]:

1. вычисляется евклидово расстояние между соседними ключевыми точками в проекции на плоскость изображения и в пространстве предсказанных глубин;
2. если расстояние между точками превышает заданный порог, предполагается, что между точками произошла ошибка в оценке глубины;
3. координаты точек корректируются, чтобы минимизировать визуальные искажения, такие как неестественные изгибы или разрывы скелета.
4. Результаты и сравнительный анализ с существующими моделями
   1. Описание эксперимента

В рамках экспериментальной проверки эффективности разработанной модели для распознавания ключевых точек человеческого тела на изображении был реализован комплексный подход, включающий загрузку и обработку изображений, применение предварительно обученной модели и визуализацию результатов. Для достижения максимальной точности и наглядности результатов эксперименты проводились с использованием специализированных библиотек и алгоритмов компьютерного зрения.

Ядро эксперимента составляют PyTorch и его модуль для работы с изображениями torchvision, которые предоставляют инструменты для глубокого обучения и обработки изображений. Это включает в себя загрузку предварительно обученных моделей, операции с тензорами и их обработку.

Для распознавания ключевых точек использовалась модель Keypoint R-CNN, адаптированная для задачи с использованием архитектуры ResNet-50 в качестве основы. Эта модель была обучена на отдельных данных и загружена из сохраненного состояния (checkpoint), что позволило применить ее для локализации ключевых точек на новых изображениях без необходимости повторного обучения.

Исходные изображения загружались в формате RGB, после чего преобразовывались в тензоры с помощью функций библиотеки torchvision. Это преобразование включало нормализацию данных и добавление размерности батча для соответствия входным требованиям модели.

Для визуализации результатов распознавания использовались предопределенные соединения между ключевыми точками (соединения рук, туловища, ног и головы), что позволило наглядно демонстрировать положение и ориентацию человеческого тела на изображении.

Для отрисовки ключевых точек и соединений на изображении использовалась библиотека OpenCV, а результаты визуализировались с помощью Matplotlib. Это обеспечило возможность вывода обработанного изображения непосредственно в среде разработки или в интерактивном режиме.

Для улучшения визуализации и понимания трехмерного пространственного распределения ключевых точек человеческого тела, к эксперименту было добавлено использование технологии визуализации глубины. Это включало применение модели оценки глубины MiDaS, обученной предсказывать глубину из одного изображения. Модель позволяла создавать карту глубины, которая использовалась для корректировки и уточнения положения ключевых точек в трехмерном пространстве, улучшая точность визуализации трехмерных скелетов.

Ключевые точки, распознанные с помощью модели Keypoint R-CNN, в дополнение к двухмерным координатам, теперь могли быть обогащены данными о глубине, полученными из карты глубины. Это позволило проводить визуализацию не только в двух измерениях, но и в трехмерной перспективе.

Визуализация трехмерных скелетов проводилась с использованием Matplotlib и его трехмерного модуля *mpl\_toolkits.mplot3d*.

* 1. Результаты работы модели определения двумерных ключевых точек

Для наглядного представления результатов были созданы графики, показывающие динамику изменения потерь по всем компонентам модели на протяжении эпохи. Графики позволяют визуально оценить скорость сходимости модели и эффективность алгоритма обучения.



Рисунок 11 – Результат работы собственной модели для определения двумерных ключевых точек

* 1. Результаты получения трёхмерных ключевых точек

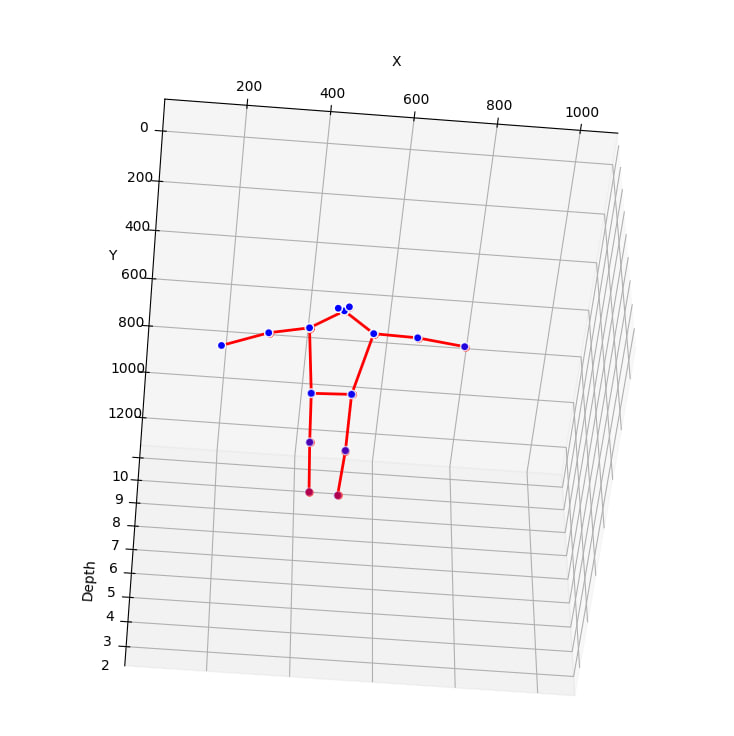


Рисунок 12 – Результат совместной работы собственной модели определения двумерных ключевых точек с моделью MiDaS

ЗАКЛЮЧЕНИЕ

В рамках данной работы было проведено масштабное исследование применимости алгоритмов машинного обучения для захвата и анализа движений человека, особое внимание уделено использованию сверточных нейронных сетей (CNN) и рекуррентных нейронных сетей (RNN) в контексте обработки изображений и видео. Исследование продемонстрировало значительные успехи в области компьютерного зрения, позволяющие точно и эффективно анализировать человеческие движения без использования маркеров.

Основной фокус работы был направлен на разработку и апробацию модели Keypoint R-CNN, адаптированной для задач захвата движения, где ключевую роль играет архитектура ResNet-50 как backbone. Эта модель позволила не только точно локализовать ключевые точки человеческого тела на двумерных изображениях, но и расширить анализ до трехмерного пространства с помощью интеграции алгоритмов оценки глубины изображения.

Среди основных вызовов, с которыми столкнулись в ходе исследования, следует отметить сложности связанные с точностью локализации ключевых точек в условиях низкой контрастности и при частичном перекрытии объектов. Решение данных проблем было частично найдено за счет внедрения корректировки ключевых точек на основе анализа глубины, что позволило значительно повысить точность распознавания движений.

Перспективы развития данной области весьма обширны. Возможность использования алгоритмов для захвата движения без специальных маркеров открывает новые направления в реабилитационной медицине, спортивных науках и развлекательной индустрии. Например, точный захват движений позволит создавать более сложные и интерактивные системы виртуальной реальности, а также способствовать более глубокому анализу техники спортсменов для предотвращения травм.

Дальнейшие исследования могут быть направлены на улучшение алгоритмов обработки изображений с низким разрешением, а также на разработку новых методов обучения моделей, способных адаптироваться к различным условиям освещения и фона. Также важным аспектом остается минимизация времени обработки данных в реальном времени для создания систем, способных функционировать в динамичных, непредсказуемых условиях.

Также целесообразным является использование рекуррентных нейронных сетей (RNN), включая LSTM и GRU, для анализа временных последовательностей движения. Эти технологии могут значительно улучшить способность системы распознавать и интерпретировать сложные последовательности движений, что особенно важно в задачах, где необходимо учитывать динамику и предыдущее состояние для предсказания будущих поз.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Козлова, Е.С. Моделирование распространения короткого двумерного импульса света / Е.С. Козлова, В.В. Котляр // Компьютерная оптика. − 2012. − Т. 36, № 2. − C. 158-164.
2. Козлова, Е.С. Моделирование предвестников Зоммерфельда и Бриллюэна в среде с частотной дисперсией на основе разностного решения волнового уравнения / Е.С. Козлова, В.В. Котляр // Компьютерная оптика. − 2013. − Т. 37, № 2. − C. 146-154.
3. Самарский, А.А. Теория разностных схем : учеб. пособие / А.А. Самарский. − Москва : Наука, 1977. − 656 с.
4. Самарский, А.А. Введение в численные методы : учеб. пособие для вузов / А.А. Самарский. − Москва : Наука, 1987. − 288 с.
5. Евтихов, М.Г. Соотношения Снеллиуса и Френеля для электромагнитных волн с постоянной линейной поляризацией // Радиотехника и электроника. − 2010. − Т. 55, № 8. − C. 915-922.
6. Тихонов, А.Н. Уравнения математической физики : учеб. пособие / А.Н. Тихонов, А.А. Самарский. − Москва : Наука, 1972. − 736 с.
7. Зайцев, В.Ф. Метод разделения переменных в математической физике : учеб. пособие / В.Ф. Зайцев, А.Д. Полянин. − Санкт-Петербург : Книжный Дом, 2009. − 92 с.
8. Владимиров, В.С. Уравнения математической физики : учеб. для вузов / В.С. Владимиров, В.В. Жаринов. − Москва : Физматлит, 2004. − 400 с.
9. Волков, И.К. Случайные процессы : учеб. для вузов / И.К. Волков, С.М. Зуев, Г.М. Цветкова. − Москва : Изд-во МГТУ им. Баумана, 1999. − 448 с.
10. Фихтенгольц, Г.М. Курс дифференциального и интегрального исчисления. В 3 т. Т. 3 / Г.М. Фихтенгольц. – Москва : Физматлит, 2001. – 662 с.

ПРИЛОЖЕНИЕ А  
Моделирование распространения импульса в среде   
с переменным коэффициентом преломления

Предположим, что на пути распространения импульса размещен оптический клин из материала, имеющего более высокий коэффициент преломления. Схематическое изображение такого волновода показано на рисунке А.1.

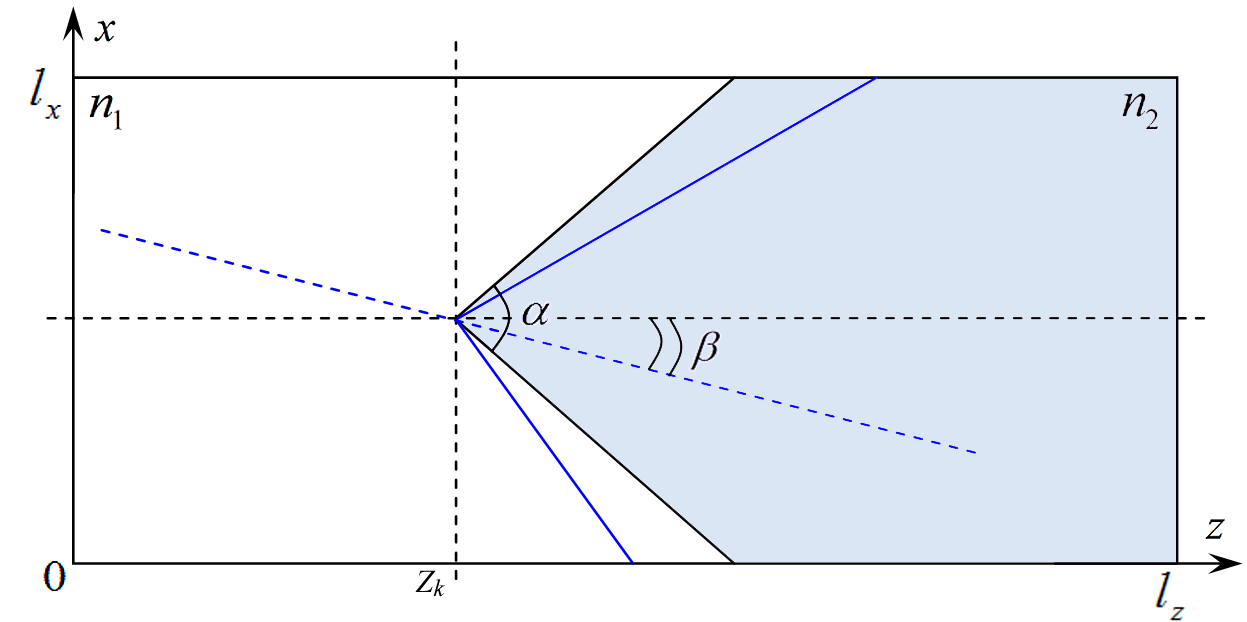


Рисунок А.1 – Изображение волновода с клином

Разработанная и оттестированная программа была применена для расчета процесса прохождения импульса через такой клин.

Проведем сравнение разностных решений для каждого случая. Зафиксируем:  Задавая уровень погрешности , будем изменять сетку таким образом, чтобы схема была устойчивой и фактическая погрешность практически совпадала с . Результаты приведены в таблице А.1.

Таблица А.1 – Сравнение разностных решений

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | 0,02 | 0,01 | 0,004 |
| Количество узлов сетки | Кусочно-гладкий импульс | 4090601 | 64682001 | 1029287201 |
| Гладкий импульс | 17576 | 262701 | 4090601 |
| Разница (во сколько раз) | 232,7 | 246,2 | 251,6 |
| Время выполнения, мс | Кусочно-гладкий импульс | 761 | 12107 | 135537 |
| Гладкий импульс | 5 | 50 | 766 |
| Разница (во сколько раз) | 152,2 | 242,1 | 176,9 |

ПРИЛОЖЕНИЕ Б  
Фрагменты кода программы

Б.1 Функция построения аналитического решения для кусочно-гладкого импульса

public double[] getAnalyticalResult1()

{

double[] res = new double[G + 1];

double sum = 0;

double z = 0;

double k = 0;

double wm = 0;

double sinwmt = 0;

double coswmt = 0;

double w = (2 \* Math.PI \* c) / lambda;

double w2 = w \* w;

double[] wm2 = new double[count + 1];

double x = lx / 2;

double a2 = (eps \* w2) / (c \* c) - (Math.PI \* Math.PI) / (lx \* lx);

double b2 = (c \* c \* a2) / eps;

double sinwt = Math.Sin((2 \* Math.PI \* tmkm \* Math.Sqrt(eps)) / lambda);

for (int m = 0; m <= count; m++)

{

wm2[m] = ((c \* c \* Math.PI \* Math.PI) / eps) \* ((1 / (lx \* lx)) + (((1 + 2 \* m) \*

(1 + 2 \* m)) / (4 \* lz \* lz)));

}

if (tmkm <= tmkm1)

{

for (int g = 0; g <= G; g++)

{

z = hg \* g;

sum = 0;

k = 0;

sinwmt = 0;

wm = 0;

for (int m = 0; m <= count; m++)

{

wm = Math.Sqrt(wm2[m]);

sinwmt = Math.Sin(Math.Sqrt(wm2[m]) \* tmkm \* Math.Sqrt(eps) / c);

k = (b2 / (wm2[m] - w2)) \* (wm \* sinwt - w \* sinwmt) - w \* sinwmt;

sum += Math.Sin(Math.PI \* x / lx) \* Math.Sin(Math.PI \* (1 + 2 \* m) \* z / (2

\* lz)) \* k\*(1-Math.Cos(Math.PI\*(1 + 2 \*m)/2))\*(4/(wm\*Math.PI\*(1 + 2 \* m)));

}

res[g] = sum + sinwt \* Math.Sin(Math.PI \* x / lx);

}

}

else

{

for (int g = 0; g <= G; g++)

{

z = hg \* g;

sum = 0;

k = 0;

sinwmt = 0;

coswmt = 0;