Санкт–Петербургский государственный университет

Новикова Александра Сергеевна

Выпускная квалификационная работа

Генерация положения персонажа по скетч рисунку с использованием искусственных нейронных сетей

Уровень образования: бакалавриат
Направление 01.03.02 «Прикладная математика и информатика»
Основная образовательная программа СВ.5005.2018 «Прикладная математика, фундаментальная информатика и программирование»
Профиль «Современное программирование»

Научный руководитель:
профессор факультета МКН СПбГУ, д.ф.-м.н.
Александр Сергеевич Куликов

Рецензент:
руководитель команды анализа данных
ООО "Академджин"
Николай Евгеньевич Русских

Санкт-Петербург
2022 г.

Содержание

1. Введение 3
1.1. Определение позы 3
1.2. Скетч рисунок 3
1.3. Методы глубокого обучения 5
2. Постановка задачи 7
3. Обзор литературы 8
3.1. Определение позы 8
3.2. Трансферное обучение 9
4. Генерация данных 11
4.1. Существующие наборы данных 11
4.2. Датасет Mixamo 13
4.3. Blender 14
4.4. Рендеринг 15
4.5. Создание синтетического набора данных 17
5. Обучение 19
5.1. Deep High-Resolution Net 19
5.2. Основные модели 20
5.3. Трансферное обучение 23
6. Результаты 26
6.1. Выбор гиперпараметра: вес ошибки дискриминатора 26
6.2. Выбор архитектуры дискриминатора 27
6.3. Численные результаты 28
6.4. Выводы и дальнейшее развитие 31
7. Заключение 35
8. Благодарность 36
Список литературы 37

1. Введение

1.1. Определение позы

Определение позы - это классическая задача компьютерного зрения, суть которой состоит в нахождении нескольких ключевых точек, так называемого скелета, на изображении. Определение позы имеет множество практических применений в самых разных областях, начиная от занятий спортом и заканчивая кинематографом.

Значительный прогресс в решении этой задачи и в целом в области компьютерного зрения случился с приходом глубокого обучения и нейронных сетей. За последние десятилетия накопилось множество крупномасштабных наборов данных с размеченными фотографиями и видео из реального мира - что является основой любого метода глубокого обучения. Объём и разнообразие датасета сильно коррелирует с точностью модели - чем больше набор данных, тем лучше модель обобщается, и, соответственно, тем лучше производительность на практике. Обучаясь на датасетах с миллионами самых разных цветных изображений, нейронная сеть способна крайне точно выделить нужные признаки и на реальных фотографиях современные нейронные сети показывают очень высокие результаты.

1.2. Скетч рисунок

Однако часто аналогичные задачи возникают для нестандартных данных. Одним из таких примеров является определение положения по скетч рисунку. Скетч, то есть набросок рисунка, сильно отличается от реальной фотографии. Как правило, скетч - это чёрно-белый рисунок карандашом на белой бумаге, который состоит лишь из силуэта персонажа. То есть по своей сути - это набор нескольких чёрных линий на белом фоне. С точки зрения компьютера - это несравнимо малое количество информации по сравнению с цветной фотографией в высоком разрешении. Персонажи мультфильмов часто имеют нереалистичные или сильно искажённые пропорции. Сами художники также могут нарушать пропорции конечностей в зависимости от своего стиля рисования или просто из-за неточностей рисунка[1, 2, 3]. Помимо

этого, художники часто используют нелинейную перспективу[4] в своих рисунках, которая не соотносится с перспективой, получаемой фото или видео камерой. Поэтому обучения на фотографиях из реального мира становится недостаточно. Реальные фотографии не покрывают то многообразие форм и конфигураций, которые могут нарисовать художники. Для данной задачи нужны более специфичные алгоритмы глубокого обучения, нестандартные подходы и подходящий обучающий набор данных.

При этом определение положения по скетч рисунку - это очень распространённая задача при создании анимации, например, для компьютерных игр, кинофильмов и мультфильмов, приложений для телефона и так далее. До создания анимации художник рисует эскиз персонажа, который впоследствии служит референсом для создания 3D модели на компьютере. Создание этой модели происходит вручную и занимает очень много времени. Кроме того, эта работа требует от художника дополнительной квалификации и часто не является его рабочим интересом, забирая на себя много ресурсов и сил и отвлекая его от более творческих занятий. Именно поэтому автоматизация данного процесса является очень востребованной на практике задачей. Получение готовой модели лишь по скетч рисунку повышает эффективность работы художника и избавляет его от монотонной деятельности, а также ускоряет процесс создания анимации.

В генерации модели по скетч рисунку есть несколько подзадач. Например, можно восстанавливать 2D или 3D скелет, то есть находить положение нескольких ключевых точек, взаимосвязанных между собой, в двумерном или трёхмерном пространстве соответственно. Более трудной задачей является восстановление всей 3D модели, то есть получение множества вершин в трёхмерном пространстве, сгрупированных в геометрические примитивы (например, в треугольники), и приближенно описывающих поверхность объекта. Один из подходов к решению этой подзадачи и её усложнениям предлагает наука фотограмметрия, которая занимается трёхмерной реконструкцией по фотографиям. Но алгоритмам фотограмметрии требуется набор из нескольких фотографий с разных ракурсов. Эти алгоритмы не предназначены для восстановления модели лишь по одной фотографии (или одному скетч рисунку).

1.3. Методы глубокого обучения

Основным препятствием для применения алгоритмов глубокого обучения в данных задачах является отсутствие обучающего набора данных. Стоимость сбора такого датасета очень высока. И даже если удастся собрать большой набор настоящих рисунков-набросков разных художников, для их точной разметки понадобится очень много человеческого ресурса. А в некоторых случаях это и вовсе практически невозможно, например, для скетч рисунка не существует истинного положения трёхмерного скелета.

Один из способов решения этой проблемы - это синтетические данные, то есть данные, искусственно созданные компьютером. Например, для задачи поиска маркеров на фотографии можно было бы случайно брать задний фон и синтетически дорисовывать на нём маркеры в случайных местах. Так как мы сами рисуем маркеры, их положение на фотографии мы знаем, а следовательно разметка данных получается автоматически.Тем самым можно получить размеченный датасет любого размера. Этот способ давно используется в машинном обучении и во многих областях приводит к отличным результатам. Чем более похожими будут сгенерированные данные на реальные, тем более точный результат модель будет показывать на практике.

Потенциально, можно сгенерировать неограниченно большое количество синтетических данных. И качество модели, обученной на этих данных, можно увеличивать с увеличением объёма выборки. Однако, конечно, не получится сгенерировать данные, идеально похожие на реальные. Решением этой задачи занимается трансферное обучение - раздел глубокого обучения, целью которого является перенос знаний и результатов одной задачи на другую. В нашем случае, знание модели о том, как работать на синтетических данных, должно быть перенесено на данные из реального мира.

В данной работе мы хотим использовать синтетические данные для решения задачи определения положения по скетч рисунку. В следующем разделе 2 мы сформулируем постановку задачи. В разделе 3 представлен обзор существующих решений аналогичных задач. В разделе 4 подробно описан механизм генерации синтетических скетч рисунков. В разделе 5 описано применение используемых алгоритмов глубокого обучения и трансферного

обучения для решения данной задачи. В разделе 6 представлены полученные результаты, сформулированы выводы и возможные направления будущих исследований. В разделе 7 подведён итог всей работы.

2. Постановка задачи

Данная работа направлена на решение задачи определения положения двумерного скелета на изображении с эскизом гуманоидного существа (персонажа). Скелет представляет из себя заранее определенный набор ключевых точек (например, нос, шея, плечи, локти, бёдра, колени и так далее). Определение положения двумерного скелета означает нахождение двумерных координат каждой ключевой точки.

Не существует открытого и размеченного набора данных для решения данной задачи. Поэтому необходимо сначала сгенерировать синтетический набор данных для последующего обучения. На полученном наборе данных необходимо обучить модель и произвести её перенос на реальные данные.

Для тестирования мы собрали небольшой набор данных и разметили его вручную для измерения качества получаемых нами моделей.

3. Обзор литературы

3.1. Определение позы

В настоящее время главными методами решения задачи определения позы являются свёрточные нейронные сети. Есть два основных подхода: регрессия положения ключевых точек и оценка тепловых карт ключевых точек.

1. Регрессия положения ключевых точек.
   Скелет представляет собой набор из k ключевых точек. Для предсказания двумерного положения этих точек нужно предсказать $2k$ значений: x и y координату для каждой ключевой точки. Это можно сделать с помощью классических регрессионных методов [5]. В качестве ошибки чаще всего используется L_2 (MSE) или L_1 расстояния [6]. В таком случае может быть использована стандартная архитектура нейронной сети для классификации (например, сеть из семейства ResNet [7] или VGG [8]). Примером реализации подобного подхода является работа [9].

   При подобном подходе сеть слой за слоем уменьшает разрешение изображения и увеличивает высокоуровневость извлекаемых признаков. В отличие от классов в стандартной задачи классификации, предсказываемые ключевые точки в задаче определения позы обладают сильно большей локальностью: для точного предсказания положения, например, кисти руки, нейронной сети необходима локальная информация в окрестности. Поэтому классические архитектуры не являются наиболее удачным решением. В следующем пункте описан альтернативный подход, решающий эту проблему более точно.

2. Оценка тепловых карт ключевых точек.
   Более продвинутым способом предсказания положения ключевых точек является оценка их тепловых карт [10]. Каждая ключевая точка вместо пары значений (x, y), представляется целой тепловой картой. Размер карты совпадает с размером исходного изображения. Температура в каждом пикселе тем выше, чем ближе пиксель к ключевой точке. Температура достигает максимума в том пикселе, в котором находится

ключевая точка (т.е. в пикселе с координатами (x, y)). Теперь нейронная сеть должна предсказывать не координаты, а всю тепловую карту целиком, а уже после этого на предсказанной тепловой карте можно найти точку максимума и считать это предсказанием ключевой точки.

В такой постановке задачи необходимо использовать так называемую Image-to-Image архитектуру - нейронную сеть, которая по изображению, т.е. тензору размера $H \times W \times C$, где H и W – это пространственные размеры изображения, а C – количество каналов (обычно 3), выдает другой тензор размера $H' \times W' \times C'$. Такими популярными архитектурами являются UNet [11], Hourglass[12], Deep High-Resolution [13].

3.2. Трансферное обучение

Так как для задачи определения позы по эскизу художника нет размеченного датасета достаточного размера, можно использовать синтетические данные и трансферное обучение.

В работах [14, 15, 16] описывается следующая идея трансферного обучения: модель обучают таким образом, что распределение признаков, извлеченных из основного обучающего датасета, совпадает с распределением признаков для датасета, на который осуществляется перенос. Если два распределения совпадают, то по признакам невозможно определить, из какого датасета они были получены, что означает, что модели придётся принимать решение о местоположении ключевых точек без этой информации. Поэтому можно предположить, что в этом случае качество переноса обучения будет лучше.

Основная идея в работе [14] состоит в том, чтобы выделить из исходного датасета набор экземпляров, распределение которых будет совпадать с распределением целевого датасета, на который осуществляется перенос. И далее использовать эти выявленные экземпляры для адаптации и поиска инвариантных признаков для двух датасетов.

В работе [15] поиск одинаковых признаков у двух датасетов происходит путём сопоставления исходного и целевого распределений в низкоразмерном пространстве. Данные проецируются в пространство более низкой размерно-

сти, где расстояние между ними будет мало. Таким образом, объекты одного датасета делают "похожими"на объекты другого датасета, но только в пространстве признаков, что улучшает качество переноса.

В работе [16] используется так называемый состязательный подход: обучается отдельная классифицирующая модель, называемая дискриминатором, цель которой определить по извлеченным признакам, какому датасету принадлежит объект. При обучении основной модели (в нашем случае определяющей позу), добавляется дополнительная функция потерь, которая заставляет часть модели, выделяющую признаки, стараться "обмануть" дискриминатор, то есть не дать ему определить датасет. Таким образом признаки основного датасета и датасета, на который осуществляется перенос, становятся похожи друг на друга. Далее, аналогично методам, уравнивающим два распределения, можно предположить, что перенос обучения будет хорошим.

4. Генерация данных

4.1. Существующие наборы данных

Одним из главных наборов данных в области компьютерного зрения является датасет ImageNet [17]. За последнее десятилетие преобладающее большинство исследований глубокого обучения в компьютерном зрении проводилось именно на данном наборе данных. ImageNet содержит более 20,000 различных классов, каждый из с которых в среднем представлен 1000 изображениями. Такое беспрецедентно большое количество размеченных данных позволило совершить прорывы в задачах классификации и обнаружении объектов. Более того, ImageNet используют и в других задачах. Очень часто с его помощью предобучают модели: сначала обучают классификационную модель на датасете ImageNet, затем убирают часть модели, отвечающую за классификацию (последний классификационный слой), и оставляют лишь часть, отвечающую за извлечение признаков. И затем добавляют нужную в конкретной задаче рабочую часть нейронной сети на место бывшей классификации. Такое предобучение в виде извлечения признаков на десятке миллионов разнообразных изображений ImageNet-а часто показывает хорошие результаты даже в очень специфичных задачах компьютерного зрения.

Классическим набором данных для задачи определения позы является датасет MS COCO [18], который включает в себя три основные исследовательские задачи анализа сцены: обнаружение объектов, контекстное взаимодействие объектов и точная двумерная локализация объектов. В задаче локализации ключевых точек набор данных для обучения, валидации и тестирования содержит более 200,000 изображений и более 250,000 человек с ключевыми точками. Все изображения - это фотографии людей в высоком разрешении из реального мира. На рисунке 1 представлен пример изображений из MS COCO датасета с наложенными поверх них размеченными ключевыми точками.

Аннотации к изображениям хранятся в формате JSON [19]. В файле с аннотациями хранится полная информация про набор данных и отдельно по каждому изображению: ширина и высота изображения в пикселях, название,

Рис. 1: Пример изображений датасета MS COCO.

номер и так далее. Отдельным JSON-полем в файле содержится информация о ключевых точках на всех изображениях. Во-первых, для каждого объекта (в данном случае для каждого человека на фотографии) предусмотрена ограничивающая рамка. В поле "bbox" хранятся координаты угла прямоугольника, который ограничивает положение человека на фотографии, и его высота и ширина. Во-вторых, для каждого объекта хранятся сами ключевые точки - массив длинной $3k$, где k - общее количество ключевых точек. Каждая ключевая точка имеет двумерные координаты x и y, а также флаг видимости v, определённый как: если v = 0, то ключевая точка не помечена (то есть можно считать, что её просто нет и x = y = 0); если v = 1, то точка не видна; если v = 2, то точка видна. Также в отдельном поле файла хранится общая информация про датасет для каждой категории (в данном случае есть лишь одна категория "человек"): набор имён ключевых точек (длины k) и скелет, определяющий связность ключевых точек (массив пар ключевых точек, используемый для визуализации). Пример части структуры JSON-файла с аннотациями ключевых точек представлен в листинге 1.

```
annotation{
    "keypoints": [x1,y1,v1,...],
    "num_keypoints": int,
    "bbox": [int],
```

```
    ...
}

categories[{
    "keypoints": [str],
    "skeleton": [edge],
    ...
}]
```

Листинг 1: Пример аннотации для ключевых точек

Большинство моделей глубокого обучения в данной области поддерживают формат MS COCO, поэтому чтобы следовать общепризнанным форматам, нам требуется сгенерировать синтетический набор данных соответствующий формату MS COCO.

4.2. Датасет Mixamo

Как обсуждалось ранее в разделе 1.2, для определения положения по скетч рисунку нужен специальный обучающий набор данных, который бы максимально походил на настоящие рисунки художников. Набор данных Mixamo [20] - это открытый набор трёхмерных моделей персонажей и анимаций к ним. Датасет содержит много разных персонажей из компьютерных игр и мультфильмов. Именно поэтому отлично подходит для использования в нашей задаче. Пример с сайта Mixamo с трёхмерной моделью персонажа Abe показан на рисунке 2.

Анимации представляют собой короткие действия, например, танец, бег, падение и тому подобное. В совокупности для каждого персонажа получается набор всевозможных поз и положений в пространстве. Было решено скачать несколько трёхмерных моделей персонажей из данного набора данных и анимации к ним, чтобы впоследствии превратить каждый кадр анимации в элемент датасета. Для каждого персонажа в Mixamo содержится более 2000 анимаций, каждая из которых в среднем содержит 100 кадров. Мы взяли 5 персонажей для создания датасета и все анимации к ним. Скачивая больше персонажей, можно улучшать качество и разнообразность набора данных, а

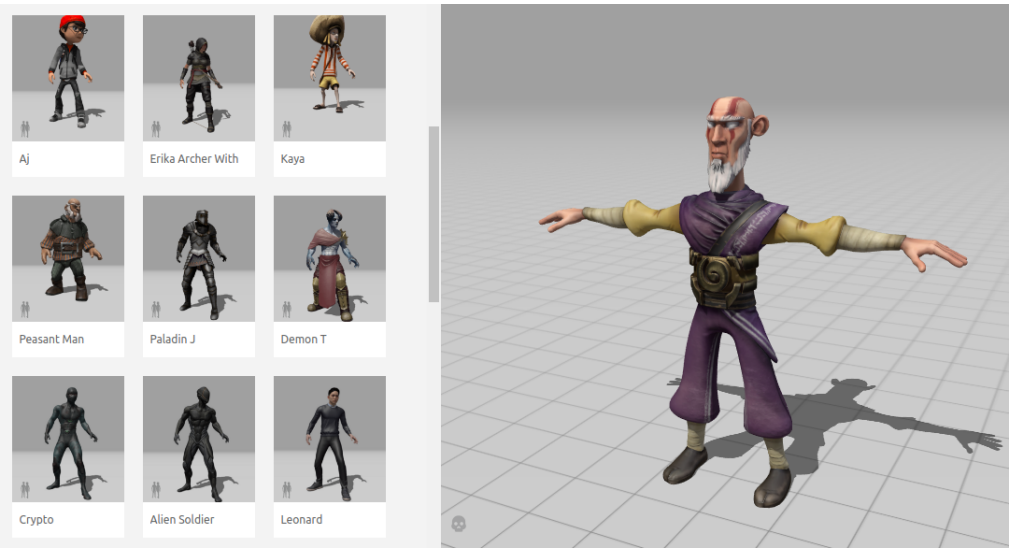

Рис. 2: Трёхмерная модель персонажа Abe с сайта Mixamo.

значит и качество обучаемых на нём моделей. В данной же работе все исследования проводились на датасете из 5 персонажей.

4.3. Blender

Для рендеринга трёхмерной модели в двумерный скетч был использован Blender [21] - набор инструментов для компьютерной 3D-графики с открытым исходным кодом.

Трёхмерная модель персонажа загружается в Blender в формате FBX. Модель состоит из двух основных частей: арматура (скелет) и полигональная сетка (меш). Скелет представляет собой набор костей, каждая из которых состоит из трёх частей: основание, тело и конец кости. Полигональная сетка состоит из множества вершин, рёбер и граней (чаще всего треугольных), которые описывают поверхность объекта. Меш объекта привязывается к арматуре: каждая вершина полигональной сетки зависит от каждой кости скелета с каким-то весом (возможно нулевым). При движении костей скелета, поверхность объекта также приходит в движение в зависимости от этих весов. Скелет всех персонажей почти полностью совпадает, за исключением некоторых незначительных костей. Основное отличие у персонажей происходит именно в полигональной сетке - для каждого персонажа она абсолютно уникальна.

Поза персонажа параметризуется углами суставов скелета. Задавая углы суставов, можно получить новую позу. И анимация - это последовательный набор поз. То есть анимация представляет собой арматуру и её движение для каждого кадра. Скелет анимации для данного персонажа полностью совпадает с его скелетом. Скелет персонажа привязывается к скелету анимации и вся трёхмерная модель приходит в движение в зависимости от движения скелета анимации. Набор анимаций для персонажей также почти полностью совпадает, с точностью до небольшого различия в костях скелета. Пример скелета из анимации и трёхмерной модели персонажа Abe показан на рисунке 3

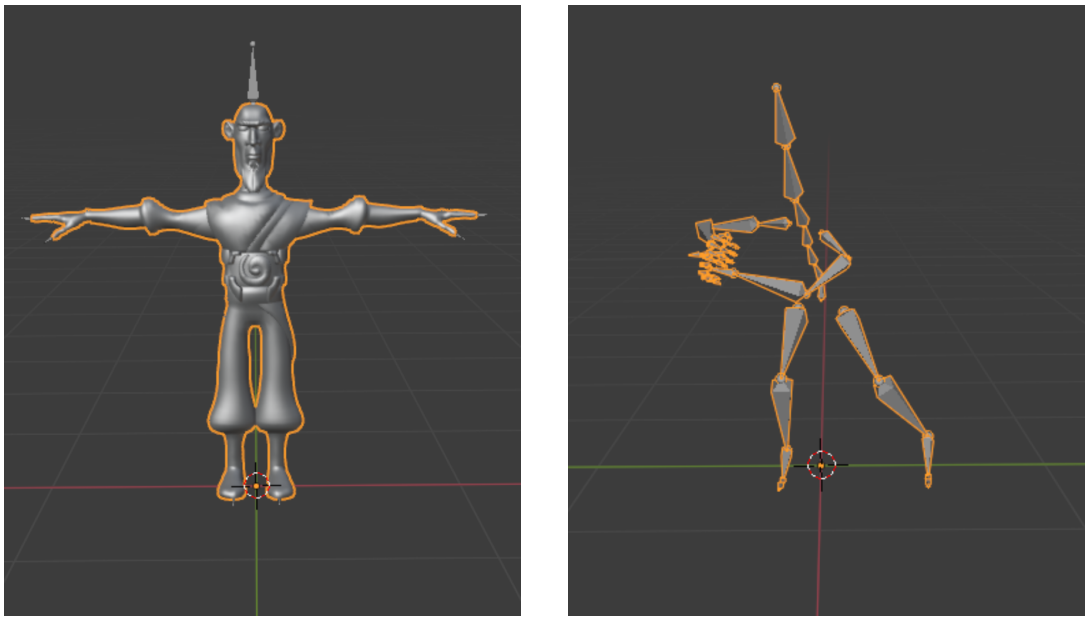

Рис. 3: Пример использования программы Blender. Слева: трёхмерная модель персонажа Abe: полигональная сетка, наложенная поверх скелета. Справа: скелет из анимации.

4.4. Рендеринг

Для создания датасета была написана программа на языке Python с использованием Blender Python API [22]. На вход программа получает список имён персонажей. Поочерёдно для каждого персонажа программа перебирает все анимации для него и делает следующее:

1. Загружается модель персонажа и текущая анимация. Модель привязы-

вается к скелету анимации. Далее в сцену добавляется 8 камер. Они расположены в углах правильного восьмиугольника, который параллелен оси XY и находится примерно на уровне середины спины персонажа. Расположение камер можно увидеть на рисунке 4.

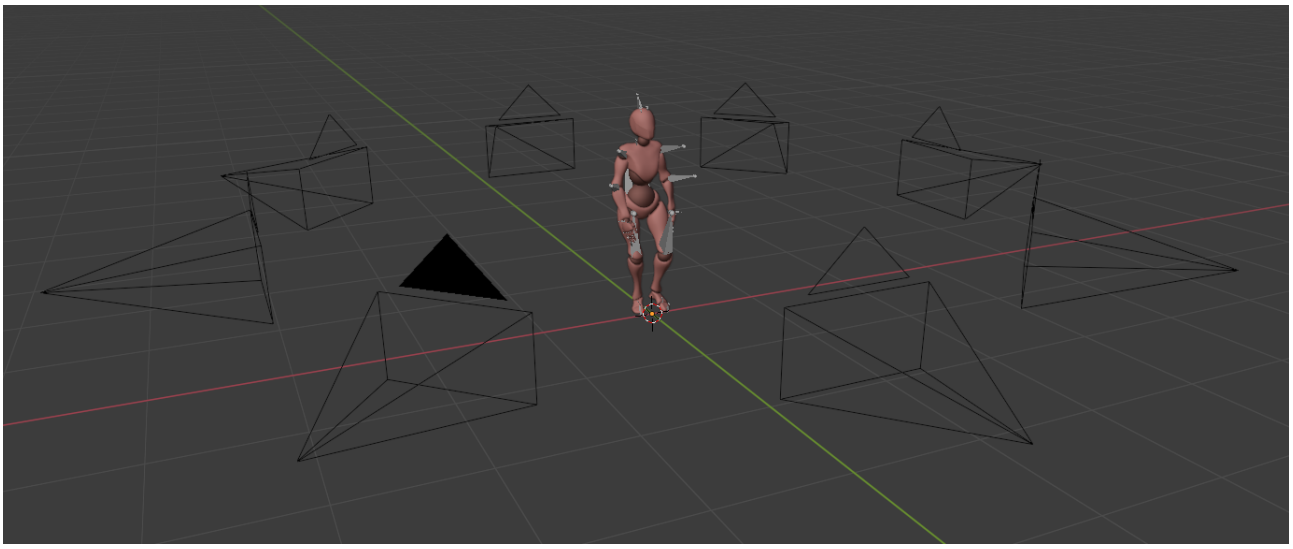

Рис. 4: Расположение камер в программе Blender при рендеринге

Рендер персонажа будет происходить для каждой добавленной камеры. С увеличением количества камер, конечно же, растёт и размер датасета. Но во-первых, понадобится сильно больше времени на рендеринг всего датасета. Во-вторых, качество датасета нелинейно зависит от количества камер, так как чем больше камер, тем более скоррелированными становятся элементы датасета. А чем более похожими становятся элементы датасета друг на друга, тем меньше увеличивается качество моделей глубокого обучения.

2. Для того, чтобы полученный рендер был похож на реальный скетч художника нужно чтобы рендерился только силуэт персонажа и, возможно, какие-то внутренние линии. Для этого создаётся отдельный материал, который заменяет материал меша персонажа. Новый материал создаётся таким образом, чтобы он не был виден при рендеринге (то есть выставляются флаги прозрачности, отражения задней стороны поверхности и т.д.). Аналогично происходит выставление флагов невидимости для костей скелета.

Далее, для того, чтобы получить границы поверхностей выставляются флаги силуэта, линий, рёбер и тому подобные.

3. Далее идёт перебор кадров. Два соседних кадра практически не отличаются друг от друга, поэтому чтобы сэкономить время на рендеринг можно брать кадры с какой-то частотой. Наш рендер выбирает примерно каждый 10 кадр.

4. Чтобы получить двумерные координаты скелета, совпадающие с полученным с помощью рендеринга силуэтом персонажа, нужно спроецировать трёхмерный скелет на текущую камеру. Проекция осуществляется с помощью однородных координат. Для проекции использовался классический метод проецирования сцены из компьютерного зрения [23].

   Трёхмерные модели датасета Mixamo содержат в среднем 60-70 ключевых точек. В задаче определения позы чаще всего используется около 20 ключевых точек, например, в датасете MS COCO [18] каждый человек описан скелетом с 17 ключевыми точками. Поэтому мы выделили лишь основные суставы. Всего 21, массив ключевых точек представлен в следующем фрагменте кода (у ключевых точек рук и ног по два экземпляра: левый и правый): 2.

```
skeleton_bones = ['Head', 'Neck', 'Spine', 'Spine1', 'Spine2',
                  'Shoulder', 'Arm', 'ForeArm', 'Hand',
                  'Hips', 'UpLeg', 'Leg', 'Foot']

```

Листинг 2: Итоговые ключевые точки датасета

5. Полученный рендер персонажа и двумерные координаты скелета сохраняются в файлы.

4.5. Создание синтетического набора данных

Чтобы ускорить процесс рендеринга, полученная программа была запущена на сервере на разных GPU отдельно для каждого персонажа. Рендеринг работал около 10 дней. Было получено более 200,000 изображений со скелетами. Пример изображений полученного датасета показан на рисунке 5

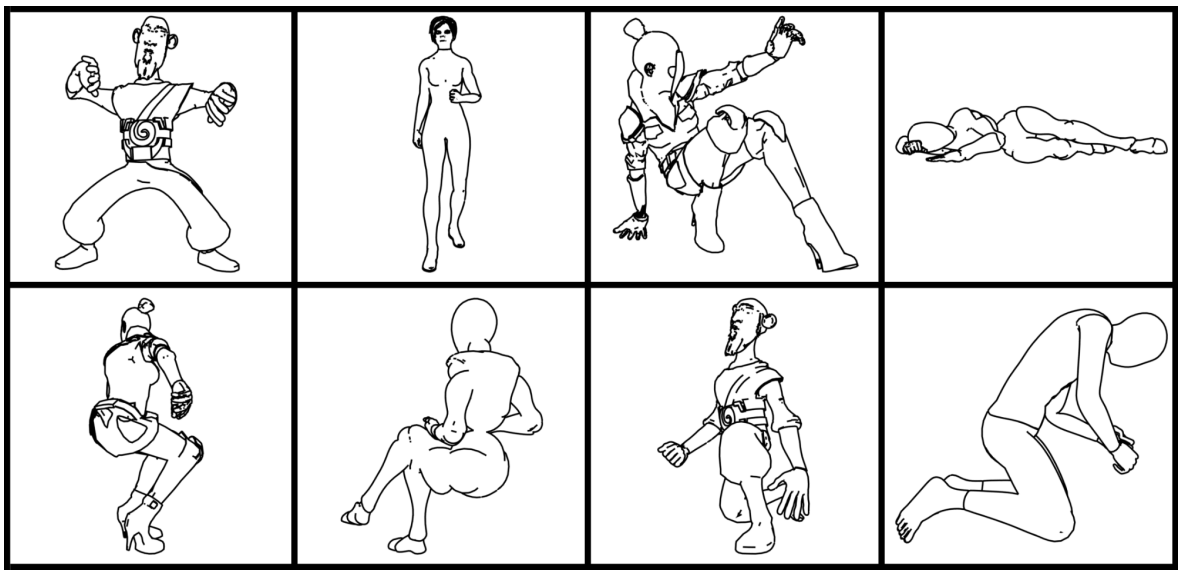

Рис. 5: Пример изображений синтетического набора данных

Далее был написан скрипт по преобразованию полученного датасета в формат MS COCO для его дальнейшего использования.

5. Обучение

5.1. Deep High-Resolution Net

Данная работа фокусируется на создание синтетического датасета, обучении на нём и после на улучшении модели с помощью трансферного обучения. Поэтому выбор модели для обучения - это параллельная задача, которая не является для нас главной (но, безусловно, в будущих исследованиях можно пытаться улучшать результаты именно за счёт изменения модели обучения).

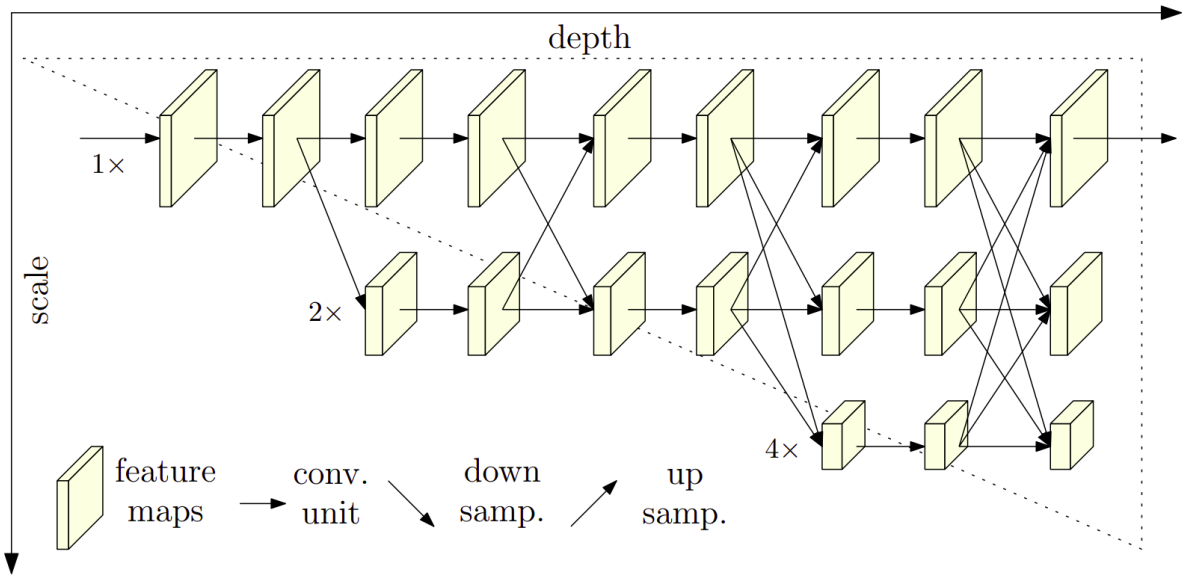

Рис. 6: Иллюстрация архитектуры сети Deep High-Resolution. Модель состоит из параллельных подсетей с высоким и низким разрешением с обменом информацией между этими подсетями. Горизонтальное и вертикальное направления соответствуют глубине сети и масштабу карт признаков соответственно.

Для обучения была выбрана архитектура Deep High-Resolution [13, 24, 25]. У данной модели есть ряд преимуществ:

- Архитектура модели направлена на решение задачи определения позы.
- Архитектура является одной из самых современных и лучших в задаче определения позы, поэтому учитывается весь предыдущий опыт в решении данной задачи. И модель показывает улучшения в результатах по сравнению с предыдущими моделями.

- Для данной модели существует реализация в открытом доступе на сайте Github.com [26]. У репозитория более 3700 звёзд и более 800 ответвлений, что свидетельствует об используемости и популярности.

Архитектура модели представлена на рисунке 6. Модель предсказывает тепловые карты ключевых точек (более подробно было описано в разделе 3.1). Модель улучшает идею классических нейронных сетей, адаптируя её к задаче определения позы: постепенно уменьшает разрешение изображения, переходя к всё более высокоуровневым признакам. При этом изображение в высоком разрешении также поддерживается на протяжении всей архитектуры. Благодаря связям между уровнями с разным разрешением и связям внутри каждого уровня, модель одновременно выделяет качественные высокоуровневые признаки (как в классических моделях) и не теряет информацию про изначальное изображение.

5.2. Основные модели

В качестве базовой модели, которая была отправной точкой для последующих моделей и улучшений, использовалась модель Deep High-Resolution, обученная на датасете MS COCO (про который более подробно было описано в разделе 4.1).

В датасете MS COCO для каждого человека предоставлено 17 ключевых точек. Они перечислены во фрагменте 3.

```
"keypoints": [
  "nose", "left_eye", "right_eye", "left_ear",
  "right_ear",
  "left_shoulder", "right_shoulder", "left_elbow",
  "right_elbow", "left_wrist", "right_wrist",
  "left_hip", "right_hip", "left_knee", "right_knee",
  "left_ankle", "right_ankle"
]
```

Листинг 3: Набор ключевых точек в датасете MS COCO

При этом модель, представленная в статье, предобучается на датасете ImageNet.

Далее были обучены две модели уже на сгенерированном датасете с синтетическими данными. В качестве предобучения использовалась и модель, обученная на ImageNet-e, и модель, обученная на MS COCO. Так как в сгенерированном датасете для каждого персонажа хранится 21 ключевая точка, то и модели предсказывают 21 точку.

Для валидации и тестирования было собрано два набора данных с размеченными настоящими скетч рисунками художников. В валидационном датасете более 800 изображений. В тестировочном датасете около 100 изображений. Датасеты похожи, но в валидационном датасете скетчи более разрежены: чаще всего это просто светлый фон и чёрный карандашный рисунок, из-за чего датасет больше похож на сгенерированный синтетический набор данных, а в тестировочном скетчи иногда цветные, более текстурные, периодически раскрашены внутри силуэта персонажа. В обоих датасетах разметка представляет собой 18 ключевых точек. Они перечислены во фрагменте 4.

```
"keypoints": [
  "Head", "Neck",
  "Right Shoulder", "Right Arm", "Right Hand", "Left
  Shoulder", "Left Arm", "Left Hand",
  "Spine", "Hips",
  "Right Upper Leg", "Right Leg", "Right Foot", "Left
   Upper Leg", "Left Leg", "Left Foot",
  "Left Toe", "Right Toe"
]
```

Листинг 4: Набор ключевых точек в тестировочном датасете

Так как во всех наборах данных (датасет MS COCO, тестировочный датасет, сгенерированный датасет с синтетическими данными) разное количество ключевых точек, то для тестирования и сравнения моделей было выделено 12 общих ключевых точек. 6 ключевых точек для рук (левые и правые плечи, локти и кисти) и 6 ключевых точек для ног (левые и правые бёдра, колени и стопы). Все последующие результаты тестирований и сравнений будут

происходить на этих 12 ключевых точках.

Формат датасета MS COCO позволяет "не помечать" некоторые ключевые точки. Для преобразования датасетов к общему виду был написан код, который "не помечал" ненужные ключевые точки (выставлял для них видимость, равную 0).

В качестве валидации на синтетическом наборе данных пробовались разные варианты: выделять случайно часть тренировочных данных под валидацию, выделять от каждого персонажа поровну данных, выделять только одного персонажа. Первые два варианта показывали на валидации очень высокую точность из-за сильной скоррелированности синтетического датасета, поэтому в итоге использовался только последний вариант: в качестве валидации использовался один персонаж, обучение происходило на остальных без него.

Для сравнения моделей использовалась метрика Average Precision (AP). Вначале определим сходство по ключевым точкам для объекта (OKS: object keypoint similarity), как

$$OKS = \frac{\sum_i [exp(\frac{-d_i^2}{2s^2k_i^2})\delta(v_i > 0)]}{\sum_i [\delta(v_i > 0)]},$$

где d_i - евклидово расстояние по каждой ключевой точке между предсказанным положением и настоящим положением, v_i - флаг видимости ключевой точки (см. раздел 4.1), s - масштаб объекта, k_i - константа для каждой ключевой точки, контролирующая затухание. Для каждой ключевой точки ненормированная гауссиана со стандартным отклонением sk_i даёт сходство в диапазоне от 0 до 1. Эти сходства усредняются по всем видимым ключевым точкам (ключевым точкам, для которых $v_i > 0$).

Главным показателем, по которому мы сравниваем модели, является mAP - это AP для OKS, усреднённый по нескольким порогам:

$$mAP = AP\ at\ OKS = .50 : .05 : .95 \quad (1)$$

5.3. Трансферное обучение

Если использовать только методы глубокого обучения, описанные выше, то полученная модель может слишком сильно подстроиться под синтетический датасет и из-за этого будет плохо обобщаться на настоящие рисунки. Поэтому после стандартного обучения мы использовали методы трансферного обучения. Это методы, направленные на перенос знаний, полученных моделью от одного датасета, на другой датасет. В нашем случае мы хотим обучить нейронную сеть на синтетических данных таким образом, чтобы она выдавала точные предсказания и для реальных данных, для которых у нас нет разметки.

Более конкретно, этой задачей в области трансферного обучения занимается адаптация домена - область, изучающая предсказания в условиях сдвига между тренировочными и тестовыми распределениями датасетов. В качестве метода адаптации домена используется adverserial loss [16]. Для его применения необходим набор реальных данных без разметки. Мы собрали датасет с более чем 14,000 реальных скетч рисунков. Рисунки были взяты из поисковых систем, таких как Google, Pinterest и другие. Это неразмеченные данные, на их разметку требуется много временного и денежного ресурсов.

Идея метода adverserial loss состоит в том, чтобы обученная модель не могла определить, из какого датасета пришло изображение: из размеченного синтетического набора данных или из неразмеченного набора данных с реальными скетчами. То есть модель будет считать распределения двух этих датасетов идентичными. В таком случае это означает, что модель не пользуется признаками, которые доступны только для синтетических данных. А значит, для реальных данных модель должна выдавать столь же хороший результат.

Схематично архитектура модели изображена на рисунке 7.

К нейронной сети добавляется новая часть, называемая дискриминатором. Дискриминатор получает на вход признаки, которая выделила основная часть модели, и по этим признакам определяет, к какому из двух датасетов принадлежит изображение. То есть для всех изображений нейронная сеть после извлечения признаков определяет номер датасета. При этом для изоб-

Рис. 7: Архитектура нейронной сети с добавлением адаптации домена.

ражений, которые пришли из синтетического набора данных, нейросеть так же предсказывает двумерный скелет.

В качестве функции потерь для дискриминатора использовалась ошибка бинарной кросс-энтропии (Binary Cross Entropy [27]):

$$loss_{discr} = BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)),$$

где N - количество изображений в батче, y_i - номер датасета для i-ого изображения (0 либо 1), $p(y_i)$ - предсказанная вероятность класса для i-ого изображения. $loss_{discr}$ используется в методе обратного распространения ошибки при обновлении весов в дискриминаторе.

Для предсказанных тепловых карт двумерных скелетов для синтетических данных считается средняя квадратическая ошибка (MSE [6]):

$$loss_{skeleton} = MSE = \frac{1}{N}\sum_{i=1}^{N}(m_i - m_i')^2,$$

где N - количество изображений в батче, m_i - тепловая карта i-ого изображения, полученная путём применения двумерной гауссианы со стандартным отклонением в 1 пиксель и центром в настоящем положении каждой ключевой точки, m_i' - предсказанная тепловая карта. $loss_{skeleton}$ влияет на веса внутри части, отвечающей за предсказание двумерного скелета.

Две полученные ошибки складываются с некоторыми коэффициентами,

образуя общую ошибку

$$loss = loss_{skeleton} - \gamma\, loss_{discr}, \tag{2}$$

где γ - положительное число.

Таким образом, нейронная сеть учится одновременно выделять признаки, нужные для определения двумерного скелета (минимизируя $loss_{skeleton}$), и обманывать дискриминатор (максимизируя $\gamma\, loss_{discr}$), то есть не выделять признаки, которые бы помогали отличать синтетические данные от реальных.

Тем самым, нейронная сеть предсказывает двумерный скелет только по тем признакам, которые есть у обоих датасетов. А значит, с увеличением качества предсказаний для синтетических данных, будет расти и качество для реальных скетч рисунков.

В качестве дискриминатора пробовались разные классификационные архитектуры: стандартная линейная классификация, классификации с конволюционными слоями. Так же пробовались разные значения γ в формуле 2.

6. Результаты

Во всех экспериментах в качестве обучающего размеченного датасета использовались сгенерированные нами синтетические данные (см. раздел 4), состоящие из более чем 200,000 картинок для определения позы. Для обучения дискриминатора использовался датасет из более чем 14,000 неразмеченных реальных скетчей. Для валидации и тестирования использовались два размеченных датасета реальных скетчей из 800 и 100 картинок соответственно. На вход нейронной сети подаются изображения размера 224x192. Обучение всех моделей происходило в течении 10 эпох. Качество моделей измерялось на валидационной и тестовой выборках. У всех моделей были одинаковые гиперпараметры, представленные в таблице 1, за исключением веса ошибки дискриминатора γ (2) и начальных весов моделей.

В качестве бейзлайна мы выбрали модель, обученную на датасете MS COCO.

Оптимизатор	Adam
Шаг обучения	0.001
β_1	0.9
β_2	0.999
Размер изображений	224×198
Выделение признаков	Deep High-Resolution Network

Таблица 1: Гиперпараметры, используемые в экспериментах.

6.1. Выбор гиперпараметра: вес ошибки дискриминатора

Ключевую роль в нашей модели играет доменная адаптация, которая контролируется за счёт веса ошибки дискриминатора (параметр γ в уравнении (2)). Если параметр слишком маленький, то модель, извлекающая признаки, будет практически полностью игнорировать дискриминатор и доменная адаптация происходить не будет. Если параметр слишком высокий, то модель, извлекающая признаки, будет слишком сильно подстраиваться под дискриминатор, из-за чего качество оценки позы будет падать. Таким образом, необходимо выбрать золотую середину между этими двумя режимами.

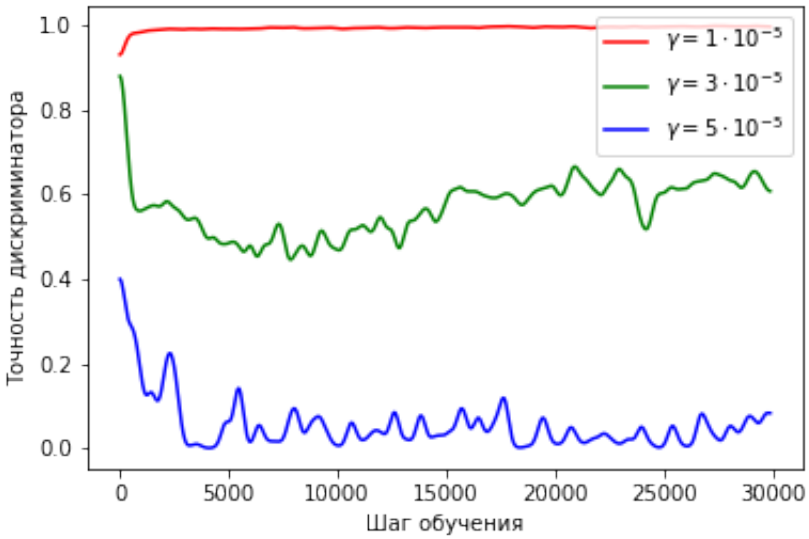

Рис. 8: График точности дискриминатора моделей с доменной адаптацией с разными γ.

На рисунке 8 представлен график точности дискриминатора при разных γ для одной и той же архитектуры сети. При $\gamma = 1 \cdot 10^{-5}$ точность дискриминатора равна единице, то есть он идеально распознает два датасет по признакам, которые получает от модели, выделяющей признаки, что означает, что γ слишком мала. При $\gamma = 5 \cdot 10^{-5}$ точность дискриминатора примерно равна нулю, то есть γ можно уменьшить. При $\gamma = 3 \cdot 10^{-5}$ достигается среднее поведение между двумя крайними случаями: точность дискриминатора больше нуля и меньше единицы, то есть поведения модели, выделяющей признаки, и дискриминатора сбалансированы.

6.2. Выбор архитектуры дискриминатора

В нашей работе мы исследовали два варианта архитектуры дискриминатора: многослойный перцептрон и свёрточная сеть.

Многослойный перцептрон с несколькими полносвязными слоями является стандартным выбором для случая, когда модель, извлекающая признаки, возвращает d-мерный вектор (где d достаточно мало, примерно 512). В нашем же случае модель, извлекающая признаки, возвращает трёхмерные тензоры размера $64 \times 48 \times 32$, поэтому сразу применять полносвязные слои нельзя - они получатся слишком большими. Поэтому перед ними мы применяем один

свёрточный слой, после которого идёт слой Average Pooling.

Свёрточная архитектура состоит из серии свёрточных слоёв со смещениями (strides), которые постепенно уменьшают пространственные размеры тензора признаков, пока не доводят их до 1×1.

Примеры обоих вариантов архитектур дискриминатора представлены в листингах 5 и 6.

```
                Conv2d(256, kernel_size=1),
                AdaptiveAvgPool2d(1),
                Flatten(),
                Linear(in_features=256, out_features=1)

```

Листинг 5: Пример полносвязного дискриминатора с предшествующим конволюционным слоем

```
                Conv2d(64, kernel_size=(4, 3), stride=(4, 3)),
                BatchNorm2d(64),
                ReLU(),
                Conv2d(128, kernel_size=2, stride=2),
                BatchNorm2d(128),
                ReLU(),
                Conv2d(256, kernel_size=2, stride=2),
                BatchNorm2d(256),
                ReLU(),
                Conv2d(512, kernel_size=4, stride=4),
                BatchNorm2d(512),
                ReLU(),
                Flatten(),
                Linear(in_features=512, out_features=1)

```

Листинг 6: Пример свёрточного дискриминатора

Сглаженный график точности на валидации 4 архитектур (два многослойных перцептрона и две свёртночные нейросети) представлен на рисунке 9. Архитектуры отличаются количеством линейных и свёрточных слоёв. Лучшие результаты показала архитектура, указанная в листинге 5

6.3. Численные результаты

В работе мы рассматриваем два варианта предобучения модели, которая извлекает признаки: стандартный способ предобучения на датасете ImageNet

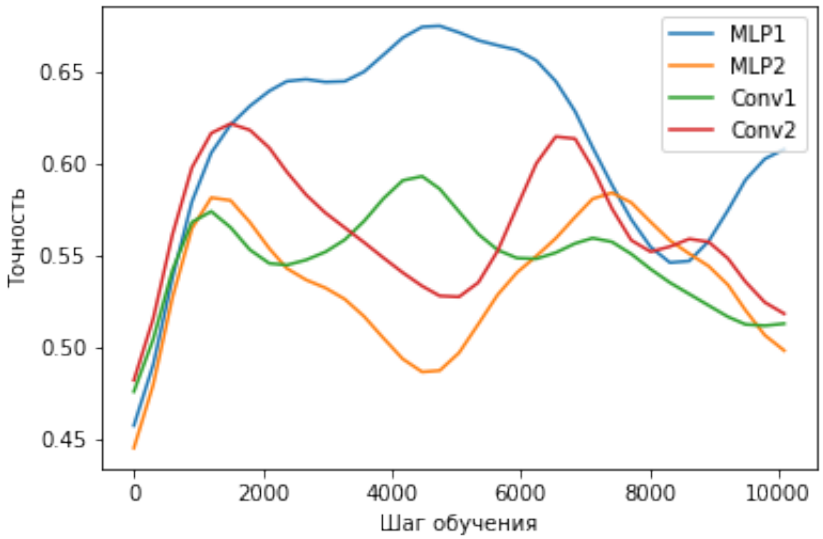

Рис. 9: Сглаженный график точности 4 архитектур, где MLP1 - это многослойный перцептрон с 1 линейным слоем, MLP2 - это многослойный перцептрон с 4 линейными слоями, Conv1 - свёрточная архитектура с удвоением количества выходных каналов(указанная в листинге 6), Conv2 - свёрточная архитектура без удвоения выходных каналов.

(классификация на 1000 классов) и предобучение на датасете MS COCO (определение позы).

На рисунке 10 изображены сглаженные графики точности определения позы (mAP 1) на валидационном множестве для пяти различных моделей:

1. **Baseline**. Бейзлайн модель: обучение на COCO
2. **ImageNet-Sketch**. Начальные веса - модель, предобученная на ImageNet, дообучение на синтетическом датасете, без доменной адаптации
3. **ImageNet-Sketch-Domain**. Начальные веса - модель, предобученная на ImageNet, дообучение на синтетическом датасете, с доменной адаптацией
4. **COCO-Sketch**. Начальные веса - модель, предобученная на COCO, дообучение на синтетическом датасете, без доменной адаптации
5. **COCO-Sketch-Domain**. Начальные веса - модель, предобученная на COCO, дообучение на синтетическом датасете, с доменной адаптацией

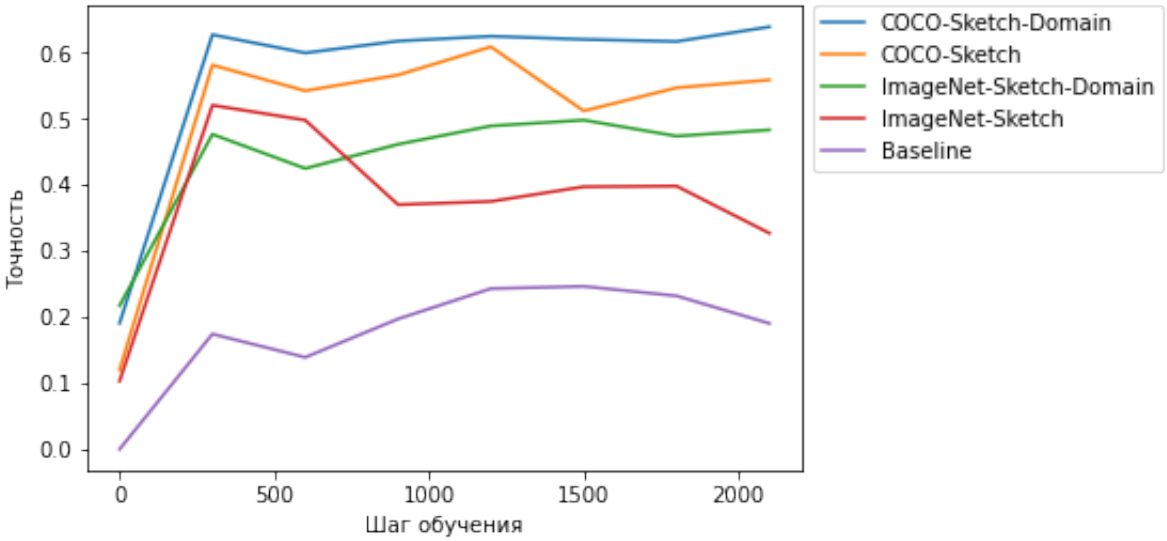

Рис. 10: Сглаженные графики точности 5 моделей, описанные в разделе 6.3

Лучшие результаты на валидации показывают модели, предобученные на COCO датасете. Для них (а также для бейзлайн модели) мы посчитали качество на валидационном и тестовом датасетах. Результаты представлены в таблице 2.

Модель	mAP на валидационном датасете	mAP на тестовом датасете
Baseline	0.553	0.520
COCO-Sketch	0.630	0.601
COCO-Sketch-Domain	**0.721**	**0.750**

Таблица 2: Результаты главных моделей на валидационном и тестовом датасетах

Тестовый датасет отличается от валидационного большей вариативностью рисунков. Полученная главная модель (COCO-Sketch-Domain) показывает сильно лучшие результаты, относительно двух других моделей (Baseline и COCO-Sketch) на обоих датасетах. Так как в ней применяется трансферное обучение и при обучении используется датасет с 14,000 изображений реальных скетчей, взятых из интернета, где все рисунки очень разнообразны, модель получилась более обобщаемой на реальные данные.

Примеры предсказаний трёх моделей, результаты которых предоставлены в таблице 2, изображены на рисунках 11 и 12. И даже невооружённым глазом видно, насколько модель, обученная на синтетических данных (COCO-Sketch), улучшает результат по сравнению с моделью, обученной толь-

ко на COCO датасете (Baseline). И насколько модель с доменной адаптацией (COCO-Sketch-Domain) ещё значительнее улучшает оба этих результата.

Например, на верхнем примере рисунка 11 модель Baseline совсем неточно предсказывает стопы и плечи. А модель COCO-Sketch-Domain очень точно предсказывает и то, и другое. На верхнем примере рисунка 12 модели Baseline и COCO-Sketch не предсказывают одну из рук, но при этом модель COCO-Sketch, в отличие от Baseline, более точно предсказывает нижнюю часть туловища: бёдра, колени и стопы не слились в одну точку и предсказаны для обеих ног). И модель COCO-Sketch-Domain очень точно предсказывает как верхнюю часть туловища, так и нижнюю.

6.4. Выводы и дальнейшее развитие

Полученная модель с доменной адаптацией показывает очень высокие результаты, сильно опережая в точности остальные полученные модели. При этом она может быть улучшена, путём расширения набора реальных неразмеченных скетчей. Сбор такого датасета затрагивает намного меньше ресурсов, так как не тратится время на разметку.

Полученный в результате данной работы сгенерированный набор синтетических данных, безусловно, можно использовать и в дальнейших исследованиях. Более того, датасет соответствует формату MS COCO, что позволяет внедрять его в любую классическую модель, поддерживающую этот формат, без каких либо усилий и изменений.

В ходе работы был написан программный код для генерации этого датасета. Данный код может быть легко преобразован для улучшения и обогащения набора данных: в коде легко изменяется количество и положение камер (ракурсов), которые использовались при рендеринге. Таким образом, можно легко увеличить количество данных. Более того, рендеринг можно запускать с любыми трёхмерными моделями персонажей, если модели имеют тот же вид, что и в датасете Mixamo. Для этого достаточно передать название директорий с трёхмерными моделями в качестве аргументов при запуске кода. Тем самым датасет также может быть расширен.

Существуют способы предсказания трёхмерного скелета по двумерным

координатам. В статье [28] представлены эти способы и описано, как много информации о трёхмерной форме тела несут в себе двумерные ключевые точки.

Более того, можно легко перейти напрямую к трёхмерной модели. Программный код по генерации датасета позволяет быстро перейти к трёхмерному скелету. Достаточно лишь убрать проецирование скелета на плоскость камеры и добавить его сохранение в отдельный файл в формате 3D.

Таким образом, в данной работе выполнены все поставленные задачи: был сгенерирован датасет, обучены нейронные сети с применением трансферного обучения и без, получена модель для генерации положения персонажа по скетч рисунку с очень высокой точностью. В будущих исследованиях можно развивать и улучшать полученные модели и синтетический датасет. А также перейти в 3D для определения трёхмерной модели персонажа.

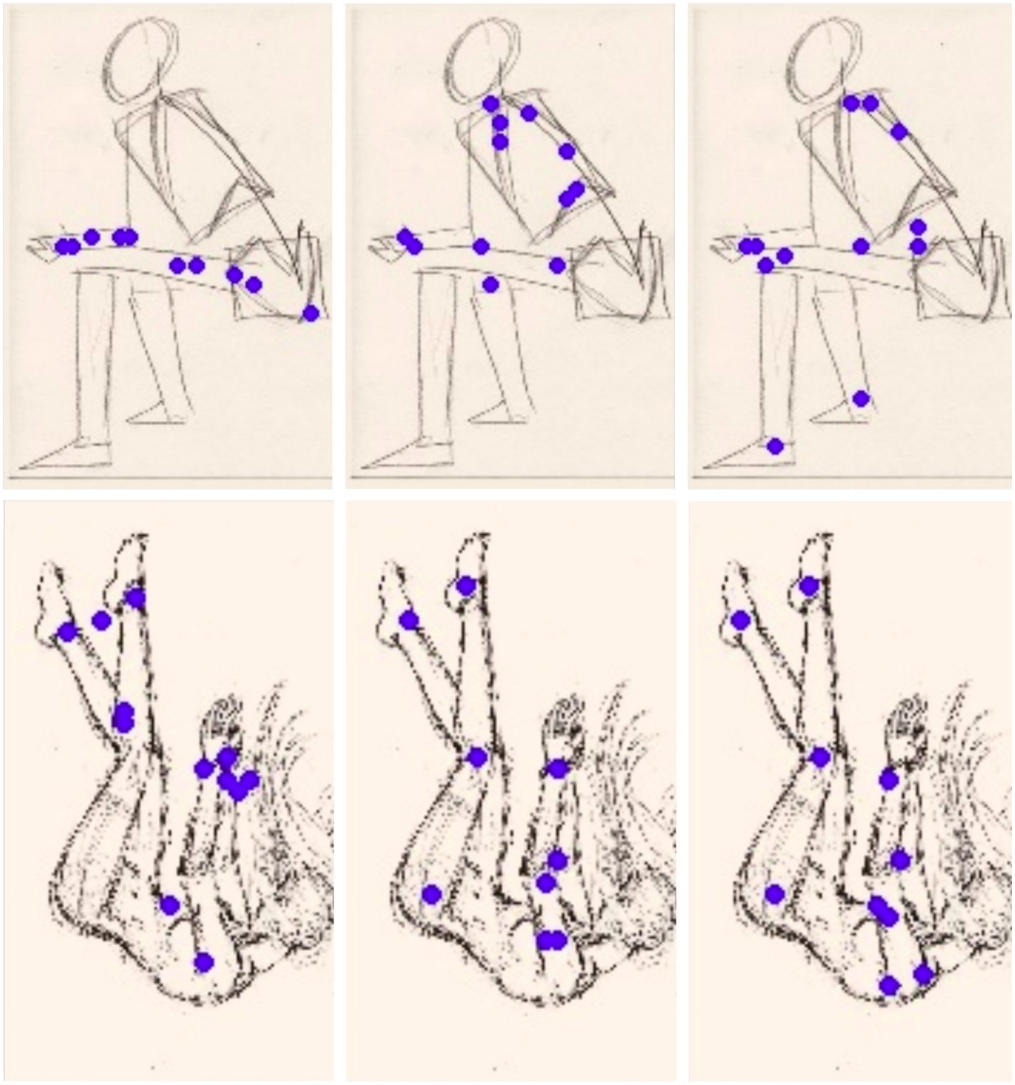

Рис. 11: Примеры предсказаний трёх моделей, результаты которых предоставлены в таблице 2. Левый столбец изображений - предсказания модели Baseline, центральный столбец - предсказания модели COCO-Sketch, правый столбец - предсказания модели COCO-Sketch-Domain

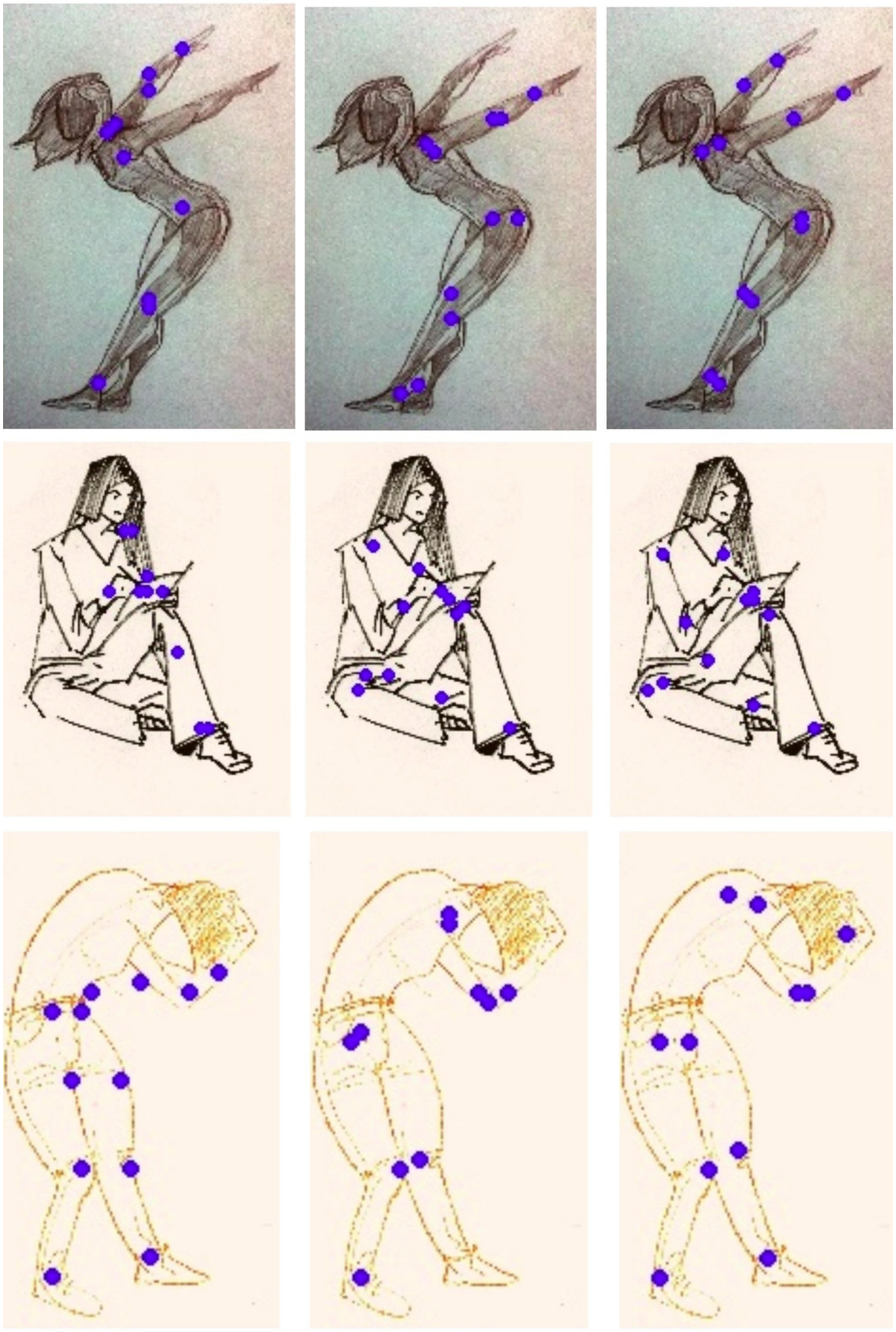

Рис. 12: Примеры предсказаний трёх моделей, результаты которых предоставлены в таблице 2. Левый столбец изображений - предсказания модели Baseline, центральный столбец - предсказания модели COCO-Sketch, правый столбец - предсказания модели COCO-Sketch-Domain

7. Заключение

В работе исследовалась задача определения положения персонажа по скетч рисунку художника.

Был сгенерирован синтетический набор данных со скетч рисунками. На полученном датасете было обучено и протестировано несколько классических моделей задачи определения положения. Для улучшения результата были применены методы трансферного обучения, в результате чего классические модели были расширены дополнительным классификатором, были изменены функции обучения и валидации.

В ходе работы был проведён ряд эксперементов: обучение происходило с разными гиперпараметрами моделей, использовались разные архитектуры классификатора для трансферного обучения. Полученный набор моделей был протестирован. Модели были сравнены по разным метрикам, а также по полученным изображениям с предсказаниями.

Обучение на синтетических данных дало прирост в качестве по сравнению с обучением на классическом датасете, что доказало применимость данного метода в задаче определения положения и целесообразность дальнейших исследований.

Методы трансферного обучения позволили ещё сильнее увеличить качество предсказаний нейронной сети. В результате чего была получена модель, предсказывающая двумерный скелет персонажа по одному скетч рисунку, с отличным качеством, которое сильно выше качества классических моделей. Более того, качество полученной модели можно легко увеличивать, не тратя ресурсов на разметку реальных скетчей.

В завершение, были обсуждены возможные способы применения обученных моделей и дальнейшие пути улучшений и исследований.

8. Благодарность

Автор выражает особую благодарность Кириллу Бродту за регулярное внимание и наставления к данной работе.

Список литературы

[1] Walt Stanchfield, Leo Brodie: *Gesture Drawing for Animation*. Independently published, 1 редакция, 2020, ISBN 979-8694892155.

[2] Hogarth, Burne: *Dynamic Figure Drawing*. Watson-Guptill, 1996.

[3] Thomas, Frank и Ollie Johnston: *The Illusion of Life: Disney Animation*. Disney Editions, New York, N.Y., 1st hyperi редакция, 1981, ISBN 0-7868-6070-7.

[4] Singh, Karan: *A Fresh Perspective*. В *Proceedings of the Graphics Interface 2002 Conference, May 27-29, 2002, Calgary, Alberta, Canada*, страницы 17–24, May 2002. http://graphicsinterface.org/wp-content/uploads/gi2002-3.pdf.

[5] Specht, Donald F.: *A general regression neural network*. IEEE Trans. Neural Networks, 2(6):568–576, 1991. https://doi.org/10.1109/72.97934.

[6] Lee, Ching-Pei и Chih-Jen Lin: *A Study on L2-Loss (Squared Hinge-Loss) Multiclass SVM*. Neural Comput., 25(5):1302–1323, 2013. https://doi.org/10.1162/NECO_a_00434.

[7] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, и Jian Sun: *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. В *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, страницы 1026–1034. IEEE Computer Society, 2015. https://doi.org/10.1109/ICCV.2015.123.

[8] Simonyan, Karen и Andrew Zisserman: *Very Deep Convolutional Networks for Large-Scale Image Recognition*. В Bengio, Yoshua и Yann LeCun (редакторы): *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. http://arxiv.org/abs/1409.1556.

[9] Toshev, Alexander и Christian Szegedy: *DeepPose: Human Pose Estimation via Deep Neural Networks*. В *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, страницы 1653–1660. IEEE Computer Society, 2014. https://doi.org/10.1109/CVPR.2014.214.

[10] Chu, Xiao, Wanli Ouyang, Hongsheng Li, и Xiaogang Wang: *Structured Feature Learning for Pose Estimation*. В *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, страницы 4715–4723. IEEE Computer Society, 2016. https://doi.org/10.1109/CVPR.2016.510.

[11] Ronneberger, Olaf, Philipp Fischer, и Thomas Brox: *U-Net: Convolutional Networks for Biomedical Image Segmentation*. В Navab, Nassir, Joachim Hornegger, William M. Wells, и Alejandro F. Frangi (редакторы): *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, страницы 234–241, Cham, 2015. Springer International Publishing, ISBN 978-3-319-24574-4.

[12] Newell, Alejandro, Kaiyu Yang, и Jia Deng: *Stacked Hourglass Networks for Human Pose Estimation*. В Leibe, Bastian, Jiri Matas, Nicu Sebe, и Max Welling (редакторы): *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, том 9912 из *Lecture Notes in Computer Science*, страницы 483–499. Springer, 2016. https://doi.org/10.1007/978-3-319-46484-8_29.

[13] Sun, Ke, Bin Xiao, Dong Liu, и Jingdong Wang: *Deep High-Resolution Representation Learning for Human Pose Estimation*. В *CVPR*, 2019.

[14] Gong, Boqing, Kristen Grauman, и Fei Sha: *Connecting the Dots with Landmarks: Discriminatively Learning Domain-Invariant Features for Unsupervised Domain Adaptation*. В *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, том 28 из *JMLR Workshop and Conference Proceedings*, стра-

ницы 222–230. JMLR.org, 2013. http://proceedings.mlr.press/v28/gong13.html.

[15] Baktashmotlagh, Mahsa, Mehrtash T. Harandi, Brian C. Lovell, и Mathieu Salzmann: *Unsupervised Domain Adaptation by Domain Invariant Projection*. В *2013 IEEE International Conference on Computer Vision*, страницы 769–776, 2013.

[16] Ganin, Yaroslav, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, и Victor S. Lempitsky: *Domain-Adversarial Training of Neural Networks*. J. Mach. Learn. Res., 17:59:1–59:35, 2016. http://jmlr.org/papers/v17/15-239.html.

[17] Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, и Li Fei-Fei: *ImageNet: A large-scale hierarchical image database*. В *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, страницы 248–255. IEEE Computer Society, 2009. https://doi.org/10.1109/CVPR.2009.5206848.

[18] Lin, Tsung-Yi, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, и C. Lawrence Zitnick: *Microsoft COCO: Common Objects in Context*. В Fleet, David J., Tomás Pajdla, Bernt Schiele, и Tinne Tuytelaars (редакторы): *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, том 8693 из *Lecture Notes in Computer Science*, страницы 740–755. Springer, 2014. https://doi.org/10.1007/978-3-319-10602-1_48.

[19] Pezoa, Felipe, Juan L Reutter, Fernando Suarez, Martín Ugarte, и Domagoj Vrgoč: *Foundations of JSON schema*. В *Proceedings of the 25th International Conference on World Wide Web*, страницы 263–273. International World Wide Web Conferences Steering Committee, 2016.

[20] https://www.mixamo.com/#/.

[21] Community, Blender Online: *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. http://www.blender.org.

[22] *Blender 3.1 python API documentation*, May 2022. https://docs.blender.org/api/current/index.html.

[23] https://courses.cs.washington.edu/courses/cse455/09wi/Lects/lect5.pdf.

[24] Xiao, Bin, Haiping Wu, и Yichen Wei: *Simple Baselines for Human Pose Estimation and Tracking*. В *European Conference on Computer Vision (ECCV)*, 2018.

[25] Wang, Jingdong, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, и Bin Xiao: *Deep High-Resolution Representation Learning for Visual Recognition*. TPAMI.

[26] Leoxiaobin: *Leoxiaobin/deep-high-resolution-net.pytorch: The project is an official implementation of our CVPR2019 paper "Deep high-resolution representation learning for human pose estimation"*. https://github.com/leoxiaobin/deep-high-resolution-net.pytorch.

[27] Cox, D. R.: *The Regression Analysis of Binary Sequences*. Journal of the Royal Statistical Society: Series B (Methodological), 20(2):215–232, июль 1958. https://doi.org/10.1111/j.2517-6161.1958.tb00292.x.

[28] Bogo, Federica, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, и Michael J. Black: *Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image*, 2016.