
Аннотация

Исследование методов доменной адаптации для
улучшения распознавания ключевых точек на теле
человека

Токарев Андрей Сергеевич

Краткое описание задачи и основных результатов,
мотивирующее прочитать весь текст

Содержание

1 Введение	3
2 Распознавание ключевых точек	7
2.1 Ключевые точки	7
2.2 Формальная постановка задачи	9
2.3 Распознавание позы человека	10
3 Доменная адаптация	14
3.1 Перенос знаний	14
3.2 Основные определения	16
3.3 Классификация методов доменной адаптации	19
4 Обзор существующих решений	23
4.1 Обзор моделей для распознавания ключевых точек	23
4.2 Обзор методов доменной адаптации на неразмеченных данных	33
5 Эксперимент	40
5.1 Описание эксперимента	40
5.2 Данные	44
5.3 Результаты эксперимента	49
6 Заключение	60
Список литературы	61

1 Введение

Современные технологии машинного обучения и компьютерного зрения продолжают активно развиваться, находя применение в самых разнообразных областях. Одно из направлений, активно развивающихся в последние годы, является решение задачи распознавания ключевых точек на теле человека (Keypoint Detection) или оценка позы человека (Human Pose Estimation). В настоящее время методы решения этой задачи могут иметь разнообразные практические применения.

Одной из возможностей использовать распознавание позы человека является виртуальная реальность. Оцифровка позы человека с помощью нейросетей позволяет сэкономить на закупке дорогостоящих костюмов. Можно установить несколько камер, которые будут восстанавливать позу человека и переносить ее компьютерное пространство. При добавлении генеративных алгоритмов можно создавать всевозможные аватары и погрузиться в "Оазис" из фильма Стивена Спилберга "Первому игроку приготовиться".

Другим, уже вполне реальным, применением данной технологии является использование её в качестве рефери на спортивных соревнованиях. Уже сейчас система полуавтоматического определения офсайда активно помогает судьям футбольных матчей по всему миру. Эта система функционирует на основе распознавания ключевых точек частей тела футболистов, которыми они могут играть в мяч, что позволяет определить, были ли нарушены правила или гол был забит чисто [1]. Таким образом, технология существенно повышает точность и объективность судейства, уменьшая количество ошибок и спорных моментов в игре.

Продолжая тему спорта, следует отметить, что оценка позы может быть использована для анализа тренировок и создания персональных помощников. Уже существуют несколько решений, направленных на ана-

лиз вашей игры в большой теннис, которые способны оценивать текущие результаты, указывать на области, требующие улучшения, и предлагать рекомендации по коррекции техники [2, 3]. Существует также проект MediaPipe от Google, предоставляющий публичные интерфейсы для анализа спортивной активности на основе распознавания ключевых точек [4]. Этот проект не ограничивается только оценкой и классификацией асан йоги. Он также включает функции для подсчёта количества повторяющихся упражнений, таких как отжимания, подтягивания и приседания.

Применений данной технологии можно придумать множество, но для их реализации необходима модель, которая будет работать быстро, поддерживающая режим реального времени, а также демонстрировать высокие показатели точности своей работы. Однако обучение модели и разработка алгоритма её работы представляют собой чрезвычайно сложный и трудоемкий процесс. Этот процесс требует значительных ресурсов, как со стороны специалистов, так и в плане вычислительной мощности. Сначала необходимо собрать и подготовить данные, затем обучить модель, настроить её параметры и протестировать на различных наборах данных, чтобы убедиться в её точности и надёжности. Это часто занимает много времени и требует значительных финансовых вложений.

В тоже время, датасеты, на которых обучаются модели часто имеют общий характер и могут вносить сильную погрешность в результаты при изменении общих характеристик входных данных. Например, модель, обученная распознавать объекты на дневных фотографиях, может показывать низкую точность наочных снимках из-за разницы в освещении и визуальных характеристиках. Поэтому важно проводить дополнительное обучение модели на данных, соответствующих конкретной задаче. Этот процесс также требует значительных усилий, аналогичных созда-

нию универсального решения. В связи с этим ученые задумались над тем, как можно уменьшить объем дополнительных работ по приспособлению модели к новым проблемам, не теряя в ее производительности. Таким образом появились алгоритмы доменной адаптации (англ. domain adaptation) и переноса обучения (англ. transfer learning).

Суть данных подходов состоит в том, чтобы преодолеть разрыв между исходным и целевым доменами данных. Это достигается путем выравнивания распределений данных, адаптации признаков и применения обученных моделей к новому дому с минимальной дополнительной обработкой. Таким образом, модели становятся более гибкими и способны эффективно работать в различных условиях, не требуя значительного объема новых данных или переработки архитектуры. В конечном итоге это не только ускоряет процесс внедрения, но и снижает затраты на разработку, поскольку уменьшает необходимость в полном цикле обучения новой модели.

Стоит отметить, что возможность адаптировать нейронную сеть к целевому набору данных имеет особую важность в задаче распознавания ключевых точек на теле человека. Различные приложения требуют адаптации к специфическим условиям съемки, будь то освещение, ракурс или качество изображения. А в разных видах спорта, акцент может быть смешен на разные части тела: в футболе важны ноги и корпус, а в баскетболе — руки и верхняя часть туловища. Отсюда требуется возможность быстрого и дешевого улучшения качества работы нейронной сети.

С учетом вышесказанного, напрашивается вывод, что тема данной работы является полезной и важной в нынешних реалиях. В разд. 4 произведен обзор как различных моделей распознавания ключевых точек

на теле человека (см в разд. 4.1), так и некоторых методов доменной адаптации (см в разд. 4.2). Также проведен эксперимент по применению *Progressive unsupervised learning (PUL)*, который хорошо себя показал в задачах детекции объектов и повторной идентификации, к оценке позы. Для него был собран и размечен целевой набор данных, описанный в разд. 5. Результаты эксперимента дают ход дальнейшим исследованиям применения PUL, за счет вариативности способов отбора псевдо-разметки.

2 Распознавание ключевых точек

С развитием технологий человечество начало ставить все более разнообразные задачи, для решения которых применялись сверточные нейронные сети (convolutional neural network, CNN). Одной из таких задач оказалось распознавание ключевых точек (Keypoint Detection). При этом распознавание ключевых точек на теле человека выделилось в отдельный раздел, известный как оценка позы (Pose Estimation). Далее рассмотрим формулировку этих задач.

2.1 Ключевые точки

Задача распознавания ключевых точек заключается в том, чтобы обнаружить и точно локализовать определенные точки или места внутри изображения или кадра видео. Эти точки, называемые ключевыми, могут быть определены для различных объектов, таких как лица, тела человека или других структур. Например, в случае распознавания лица ключевые точки могут включать углы глаз, кончик носа, уголки рта и другие характерные особенности лица. В контексте человеческого тела ключевыми точками могут быть суставы, такие как локти, колени, плечи и так далее. Для других объектов ими могут быть выбраны уникальные элементы, которые помогают идентифицировать и анализировать заданный предмет.

Другими словами, ключевые точки являются структурными, которые используются для определения положения и/или местонахождения объекта в пространстве. Они играют важную роль в задачах компьютерного зрения, таких как отслеживание движений, 3D-моделирование, анимация, медицинская визуализация и другие. Они могут использоваться для создания каркасов объектов, анализа их формы, измерения расстояний между различными частями и выполнения других аналитических

задач.

Более точно эти объекты можно определить следующим образом: *ключевые точки (КТ)* - это специфические, заранее определенные части распознаваемого объекта, которые имеют особое значение для дальнейшего анализа местоположения объекта на изображении. Каждая ключевая точка обычно соответствует определенной анатомической или структурной особенности, которая легко распознается и может служить ориентиром для алгоритмов обработки изображений. Иными словами, КТ необходимо обладать следующими характеристиками, чтобы можно было использовать их в качестве референсных для заданного объекта:

1. Уникальность

Точки должны быть уникальными и отличаться от других точек на изображении

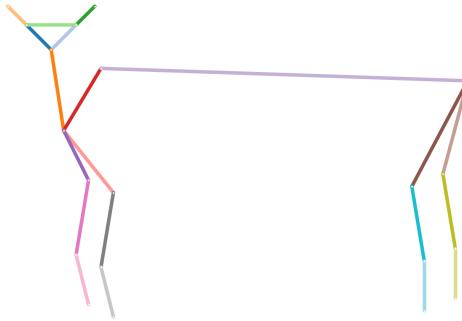
2. Инвариантность

Точки должны сохранять свою идентичность при общих преобразованиях изображения, таких как вращение, масштабирование и изменения условий освещения

3. Повторяемость

Точки должны быть обнаруживаемы в разных экземплярах одного и того же объекта или сцены

Для анализа структуры объекта и взаимосвязи между его различными КТ часто применяется схематичное описание, которое обеспечивает более наглядное визуальное представление. Этот метод помогает лучше понять анатомическую и функциональную структуру распознаваемого предмета. При этом саму схему, представляющую собой своего рода «скелет», принято называть *топологией*. Примеры различных топологий представлены на рис. 1.



(a) Animal keypoints



(b) Car keypoints

Рис. 1: Примеры топологий объектов от OpenPifPaf [5]

2.2 Формальная постановка задачи

Пусть дано изображение $I \in \mathbb{R}^{H \times W \times C}$, где $H \times W$ - размеры изображения, а C - количество каналов, равное 3 для RGB изображений.

Пусть также задан набор ключевых точек для данного изображения $K \in \mathbb{R}^{N_c \times N_k}$, где N_c - размерность предсказанных результатов, которая часто равняется размерности выходного пространства с добавлением координаты видимости точки, а N_k - количество ключевых точек.

Тогда задача T , выполняющая распознавание ключевых точек определяется как:

$$T = \{K, F_\theta(I)\},$$

где $F_\theta : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{N_c \times N_k}$ - функция предсказания нейросети с параметрами θ .

Для оценки предсказания $\hat{K} = F_\theta(I)$ вводится функция потерь, которую называют KeypointMSELoss:

$$L(\hat{K}, K) = \sum_{i=1}^{N_k} \|\hat{K}_i - K_i\|^2$$

Процесс обучения нейросети заключается в настройке параметров θ путем минимизации описанной функции потерь на тренировочном наборе размера N :

$$\theta = \operatorname{argmin}_{\theta} \frac{1}{N} \sum_{i=1}^N L(F_{\theta}(I_i), K_i)$$

Итоговая формулировка звучит следующим образом:

Необходимо найти такую функцию предсказания $F_{\theta} : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{N_c \times N_k}$, которая для входного изображения I и набора реальных ключевых точек K предсказывает ключевые точки \hat{K} , минимизируя ошибку по заданной функции потерь $L(\hat{K}, K)$.

2.3 Распознавание позы человека

На теле человека тоже можно выделить несколько ключевых точек, информация о которых дает возможность цифровизовать позу человека и использовать ее для аналитики с помощью методов машинного обучения. Именно для этого и была разработана задача распознавания ключевых точек на теле человека или, как ее часто называют в англоязычной литературе, задача оценки позы человека (Human Pose Estimation).

Расположение ключевых точек. Топология

Основной вопросом для НРЕ стал выбор набора ключевых точек и топологии, по которой они будут соединяться.

В первых работах были представлены различные примеры топологий, некоторые примеры представлены на рис. 2. Если точки туловища имеют большое количество пересечений, то точки головы сильно отличались: где-то учитывалось только положение головы (то есть добавлена верхняя точка головы), где-то рассматривались некоторые точки на лице, где-то голова вообще не учитывалась. Все зависело от задачи и возможностей исследователей.

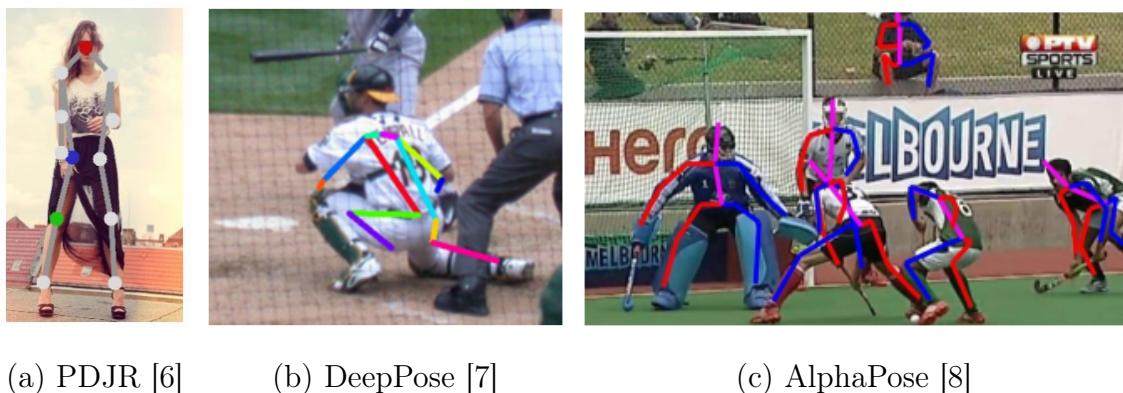


Рис. 2: Примеры различных топологий у первых решений задачи распознавания ключевых точек на теле человека

Позже в 2015 году Microsoft выпустило набор данных с детальным описанием 17 точек на теле человека и запустила соревнование по распознаванию этих точек [9, 10]. Исследователей это заинтересовало и они начали адаптировать свои модели под топологию, описанную в датасете COCO [9]. Хотя данные в наборах перестали обновлять после 2017 года, многие новые модели до сих пор оценивают по набору данных COCO. Отсюда и получилось, что данная топология стала основной для задачи оценки позы.

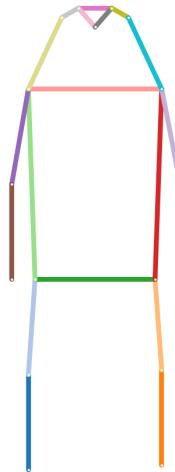


Рис. 3: Топология COCO

Подходы к распознаванию

Другим вопросом для описываемой задачи стал подход к распознаванию. Делать детекцию человека и уже на кропнутом изображении производить поиск или искать все возможные точки, а потом собирать их в скелет. Эти две идеи и сформировали два направления развития методов НРЕ. Немного о них:

1. Подход сверху-вниз (англ, *top-down*)

Для данного этапа вам понадобится дополнительная модель детекции, которая локализует человека на изображении, выделяя его в прямоугольную область. Затем этот прямоугольник передается в модель распознавания ключевых точек. Этот подход обеспечивает высокую точность предсказания КТ, но может быть чувствителен к ошибкам на этапе обнаружения и ориентации человека в кадре, что требует дополнительной предобработки изображений.

2. Подход снизу-вверх (англ, *bottom-up*)

Для данного подхода вам не нужны помощники в виде детектора, но все равно имеет два этапа в своей работе. Первоначально модель распознает все ключевые точки на полученном на вход изображении, получая таким образом карту распределения КТ по фото. А вторым шагом модель собирает все полученные точки единый скелет. При детекции нескольких людей необходимо верно сопоставить их части тела. Для этого есть несколько способов, один из которых является построение полей сходства частей тела (part affinity fields), используемый в проекте OpenPose [11].

Несмотря на различия в описанных подходах, результатом их работы является массив данных размера $[N \times K \times 3]$, где N - количество распознанных человек, K - количество ключевых точек в топологии, а 3

представляет два номера пикселей на изображении и координату видимости точки. Последнее часто интерпретируется как уверенность модели в обнаруженной точке.

3 Доменная адаптация

В современном мире машинного обучения и искусственного интеллекта способность моделей адаптироваться к новым условиям и данным стала одной из ключевых задач. Традиционные методы обучения моделей предполагают, что данные, используемые для обучения и тестирования, имеют схожие характеристики и распределения. Однако в реальных приложениях часто возникает необходимость применять модели на данных, которые существенно отличаются от тех, на которых они были изначально обучены. Это приводит к снижению точности и эффективности моделей, что ставит под угрозу их практическое применение.

Для решения данной проблемы была разработана задача доменной адаптации, целью которой является создание методов для акклиматизации модели к целевым данным, отличающимся от исходных. В рамках этой задачи были разработаны подходы и техники, направленные на уменьшение расхождений между исходным и целевым доменами, что позволяет сохранять точность и производительность моделей в новых условиях. Эти методы способствуют переносу знаний, накопленных в исходном домене, на целевой, что относит данную задачу к области методов transfer learning.

3.1 Перенос знаний

Перенос обучения (англ. transfer learning) позволил уже существующим решениям выйти за пределы первоначально заданных задач. Этот подход позволил использовать различные архитектуры для решения новых, разнообразных проблем. К примеру, это позволило перенести опыт использования трансформеров из задач обработки естественного языка в задачи компьютерного зрения.

Помимо переноса знаний между различными областями нейронных

сетей, transfer learning предоставляет возможность использовать опыт, накопленный в процессе обучения модели, для работы с новыми данными. Эти данные могут представлять собой не только новые классы в задачах классификации или кластеризации, но и иметь значительные структурные различия, такие как язык и жанр для текстовых данных или стиль и качество для изображений. Такие методы, называемые адаптацией к новым доменам данных, позволяют существенно сократить время и ресурсы, направленные на решение новых, узкоспециализированных задач.

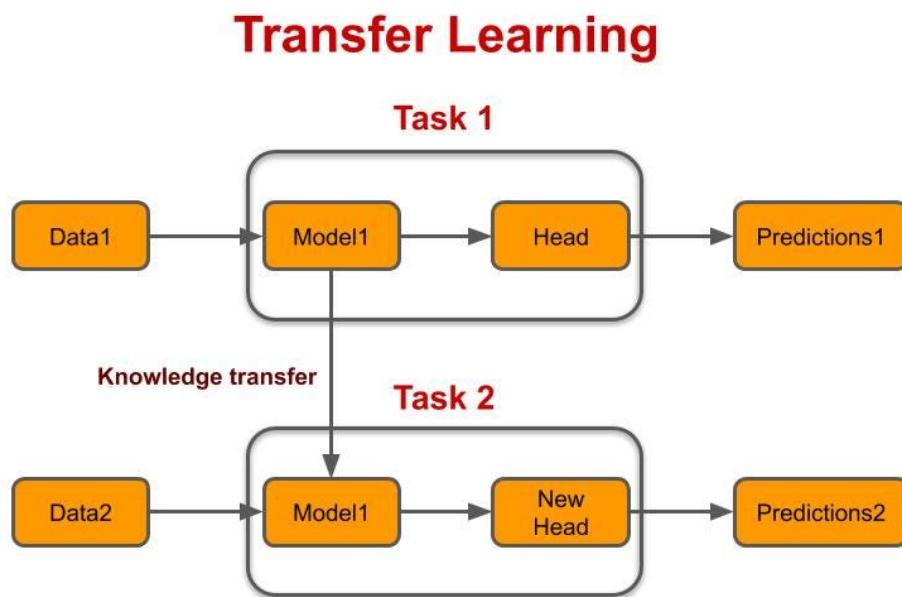


Рис. 4: Основная идея переноса знаний

Основная идея состоит в том, чтобы взять модель, которая уже научилась извлекать и интерпретировать общие признаки из большого набора данных, может эффективно адаптироваться к новым данным, требующим более специфических знаний (см рис. 4). Это аналогично тому, как человек, обладая общими знаниями в одной области, может быстрее освоить смежную сферу деятельности. Таким образом получается не только ускорить процесс обучения, но и повысить точность и эффектив-

ность модели при условии ограниченности в ресурсах.

3.2 Основные определения

Для дальнейшего повествования введем формальные обозначения для данной задачи.

Определение домена

Пусть существует пространство признаков χ и в нем задан набор данных X и частотное распределение вероятностей на нем $P(X)$:

$$X = X_1, \dots, X_n \in \chi. \quad (1)$$

В таком случае *доменом* D называют совокупность пространства признаков и частотного распределения вероятности на заданном наборе данных:

$$D = \{\chi, P(X)\} \quad (2)$$

Определение задачи

Пусть на пространстве признаков χ задан набор данных

$$X = \{X_1, \dots, X_n\} \in \chi. \quad (3)$$

Пусть существует пространство меток γ и для заданного набора данных существует соответствующий набор меток

$$Y = \{Y_1, \dots, Y_n\} \in \gamma \quad (4)$$

Тогда *задача* T определяется как

$$T = \{Y, f(X)\}, \quad (5)$$

где f - прогностическая функция зависимости целевой переменной, которую можно рассматривать как условное вероятностное распределение $P(Y|X)$.

Определение доменной адаптации

В задаче доменной адаптации подразумевается наличие не менее двух наборов данных:

1. Исходный домен (*Source*)

Представляет собой универсальный набор данных большого объема.

Обозначается следующим образом:

$$D^s = \{\chi^s, P(X^s)\}$$

На домене определена задача:

$$T^s = \{Y^s, f^s(X^s)\}$$

2. Целевой домен (*Target*)

Представляет собой набор данных маленького объема, к которому планируется адаптировать модель. Обозначается следующим образом:

$$D^t = \{\chi^t, P(X^t)\}$$

На домене определена задача:

$$T^t = \{Y^t, f^t(X^t)\}$$

В зависимости от того, как соотносятся между собой домены D^S и D^T и задачи T^s и T^t , ставятся различные задачи:

- Самым распространенным является случай совпадения задач и доменов, когда $D^s = D^t$ и $T^s = T^t$. В таком случае говорят о постановке задачи классического машинного обучения. Домен D^s называют обучающей выборкой, которая используется для обучения решения T , а домен D^t - тестовой выборкой, на которой производится оценка правильности полученного решения.

2. При совпадении данных $D^s = D^t$, но различной постановке задачи $T^s \neq T^t$ говорят о мультизадачном обучении. Часто такая постановка задачи подходит для адаптации модели классификации к новым меткам классов, которые представлены в целевой выборке, но отсутствуют в исходной.
3. При совпадении задач $T^s = T^t$, но различии в данных $D^s \neq D^t$ говорят о методах кросс-доменной адаптации. В таком случае ставится задача улучшения предсказания T^t с использованием информации, полученной в T^s .
4. При полном несовпадении данных $D^s \neq D^t$ и задач $T^s \neq T^t$ подразумевается, что пары (D, T) решают разные проблемы и адаптация происходит в индивидуальном порядке. Например, использование архитектуры трансформер, зарекомендовавшей себя в задачах обработки естественного языка, для задач компьютерного зрения.

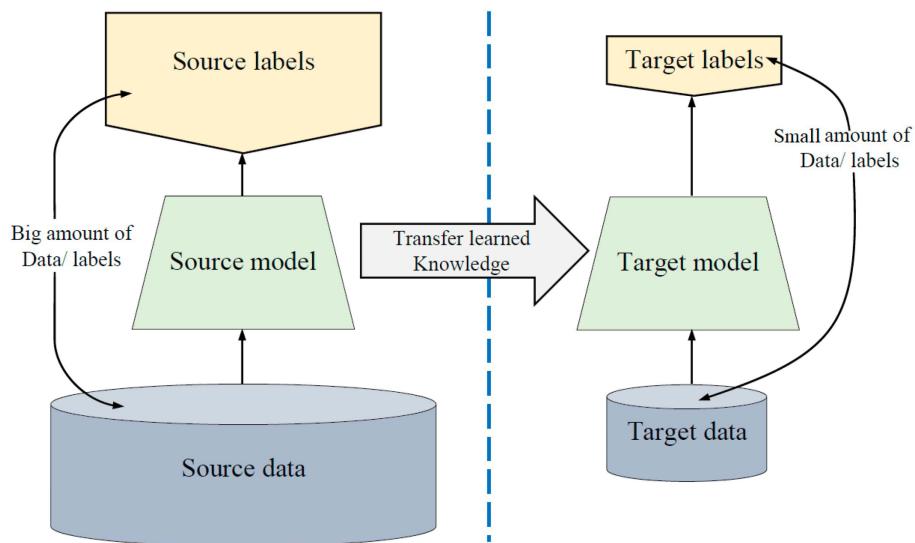


Рис. 5: Схематическое представление работы алгоритмов доменной адаптации

В текущей работе рассматривается третий вариант: $D^s \neq D^t$, но $T^s = T^t$. Схематически постановка задачи описана на рис. 5, а форму-

лировка задачи математическим языком выглядит следующим образом:

$$D^s, T^s \rightarrow D^t, T^t$$

3.3 Классификация методов доменной адаптации

Ранее методы, которые решают задачу *доменной адаптации* (англ. *domain adaptation*) были определены как методы переноса знаний, направленные на приспособление моделей $T^s = T^t = T$, обученных на данных из домена D^s , к данным из домена D^t , причем $D^s \neq D^t$. Данные могут различаться по различным характеристикам и в зависимости от различают несколько типов методов, которые схематически представлены на рис. 6.

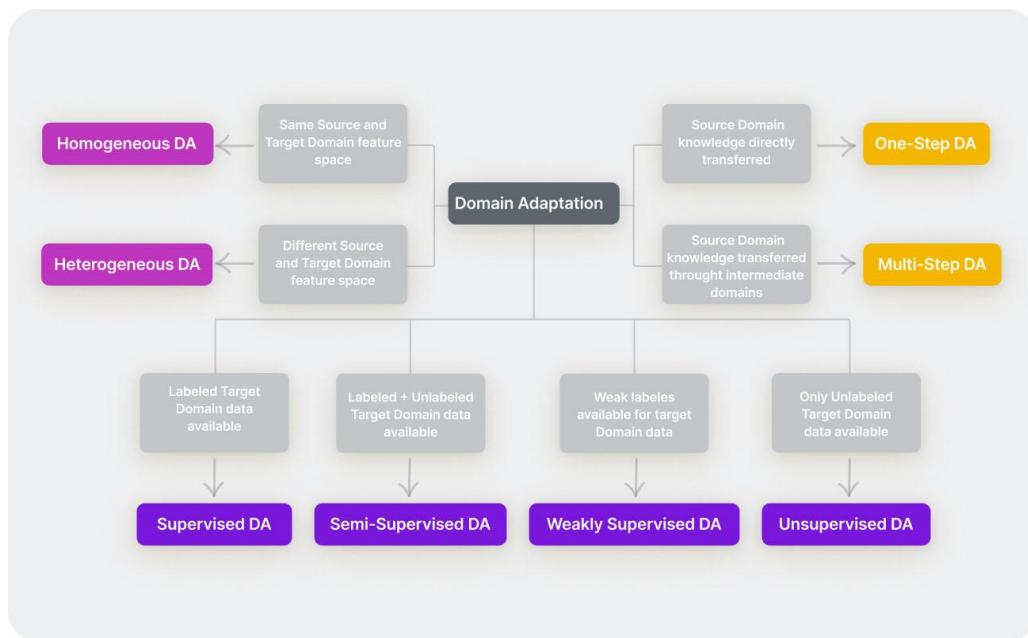


Рис. 6: Схематическое представление работы алгоритмов доменной адаптации

В зависимости от того, насколько сопоставляются распределения наборов данных $P(X)$ различают гомогенные и гетерогенные методы:

1. Гомогенные методы (*Homogeneous DA*)

Данный тип методов применяется, когда исходный и целевой домены имеют одинаковое пространство признаков $\chi^s = \chi^t = \chi$, но

их распределения отличаются $P(X^s) \neq P(X^t)$. Методы гомогенной адаптации фокусируются на выравнивании распределений признаков между доменами, чтобы модель, обученная на исходных данных D^s , могла эффективно работать на целевых данных D^t .

2. Гетерогенные методы (*Heterogeneous DA*)

В отличие от предыдущих способов, в текущем случае отличаются и признаковые пространства между доменами $\chi^s \neq \chi^t$. Это представляет собой более сложный сценарий, так как необходимо либо преобразовывать целевые данные в такое представление, которое будет сопоставимо с исходными, либо выуживать обобщенные признаки из обоих доменов, на которых модель будет считать данные похожими.

Также стоит обратить внимание насколько сильно отличаются распределения данных, так как от этого зависит сколько шагов необходимо будет производить между доменами:

1. Одношаговая доменная адаптация (*One-step DA*)

При небольших различиях между распределениями можно использовать прямой перенос знаний из D^s в D^t . В таком случае опыт, накопленный моделью в исходном домене, напрямую используется при адаптации модели T на целевом домене. Преимущества этого метода заключаются в его простоте и быстроте внедрения, так как он не требует промежуточных шагов.

2. Многошаговая доменная адаптация (*Multi-step DA*)

Однако эффективность One-Step DA может снижаться, когда различия между исходным и целевым доменами слишком велики для

прямого переноса знаний. Для этих случаев и используются методы многошаговой адаптации. Они подразумевают поэтапное выполнение: знания из исходного домена сначала переносятся в один или несколько промежуточных доменов, прежде чем достигнуть целевого. Промежуточные шаги помогают постепенному выравниванию распределения данных, что улучшает адаптацию и повышает точность модели в целевом домене.

Последним маркером в классификации методов доменной адаптации является доступность размеченных данных для D^t . По данному признаку методы разделяются сразу на 4 группы:

1. *Supervised domain adaptation*

В этом случае обучение модели происходит с использованием как исходных, так и целевых данных, имеющих метки. Поэтому данные методы обычно достигают высокой точности, поскольку модель может явно учиться на целевых данных с метками.

2. *Semi-supervised domain adaptation*

К данным методам прибегают, когда в целевом домене присутствуют как размеченные данные, которые используются для начального обучения, так и неразмеченные данные, на которых есть возможность улучшать работу модели.

3. *Weakly supervised domain adaptation*

Когда данные целевого домена слабо размечены, то есть использовалась автоматическая система разметки, которая допускает ошибки, используются методы weakly supervised DA. Они направлены

либо на улучшение точности слабых меток, либо используют «мягкие» техники регуляризации, что позволяет учитывать возможность ошибки в данных.

4. *Unsupervised domain adaptation*

Самый сложный случай, когда нет возможности аннотировать домен и надо на сырых данных улучшить качество модели. Для методов этого типа часто используется кластеризация данных и итеративные подходы, которые помогают модели T улучшать собственные предсказания.

4 Обзор существующих решений

4.1 Обзор моделей для распознавания ключевых точек

В данном разделе будут рассмотрены несколько моделей распознавания ключевых точек на теле человека. Некоторые из них довольно старые, но показывают неплохие результаты. Большинство же представлены не более 5 лет назад и являются лидерами направления на сегодняшний день.

DeepPose

Данный представитель является самым старым решением из данной выборки и, одновременно, один из самых первых в целом. Статья "DeepPose: Human Pose Estimation via Deep Neural Networks"[7] была представлена исследователями из GOOGLE на конференции CVPR в 2014 году.

Исследователи разработали модель, представляющую собой каскад из DNN-регрессоров для локализации ключевых точек. Так как на тот момент не было выпущено общепринятых топологий, то в роли ключевых точек выступали суставы тела, а поза кодировалась их координатами, нормализованными на размер изображения.



Рис. 7: Архитектура сети DeepPose. [7]

Работа модели делилась на два этапа, которые схематически показаны на рис. 7. На первом производилась локализация точки. Далее ре-

зультаты переходили на второй этап, где пропускались через каскад из DNN, который производил уточнение предсказания. В результате получался относительно точный результат, который можно было использовать для дальнейших исследований.

OpenPose

OpenPose - это проект от лаборатории перцептивных вычислений университета Карнеги-Меллона в США. Проект включает в себя несколько моделей распознавания ключевых точек: на лице, руках, теле и различные их комбинации. Количество распознаваемых моделями точек доходит до 135 на одном человеке, а модель может распознавать сразу несколько человек на одном кадре. Скорость работы модели позволяет использовать ее для распознавания видео в реальном времени через веб-камеру. К сожалению, основной репозиторий проекта не поддерживается с ноября 2020 года.

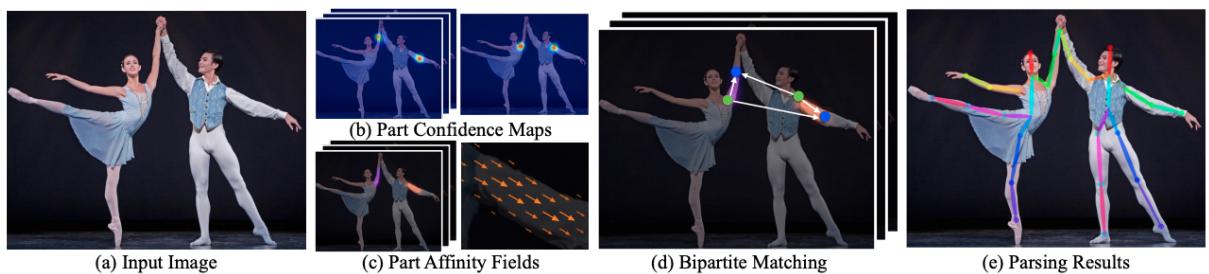


Рис. 8: Последовательность распознавания ключевых точек моделью OpenPose. [11]

На данный момент обратимся к модели, результаты которой описываются топологией из 25 точек. Данная модель одна из немногих в представленном обзоре, которая использует подход снизу вверх. Это достигается за счет использования двухступенчатой архитектуры нейросети (см рис. 8).

В рамках первого этапа (вторая колонка на рис. 8) нейросетью пред-

сказываются такие сущности, как карта достоверности обнаружения точки и карта двумерных векторных полей ориентации конечностей (англ. Part Affinity Fields или PAF). Изначально предсказание сущностей выполнялось параллельно, но взаимный анализ результатов предсказаний позволил увидеть возможность интуитивно предсказывать карты достоверности на основе PAFs. Это позволило значительно ускорить работу алгоритма.

Вторым этапом (третья колонка на рис. 8) происходит сопоставления точек и конечностей отдельным людям. Для увеличения точности и уменьшения времени построения скелетов используются графы соответствия, которые позволяют создать целостные и непротиворечивые представления поз людей на изображении.

HRNet

Решение на основе архитектуры HRNet, опубликованной в 2019 году, является одним из первых, представленных в проекте MMPose. Оно использует подход сверху-вниз, в котором для детекции используется модель из проекта MMDetection.

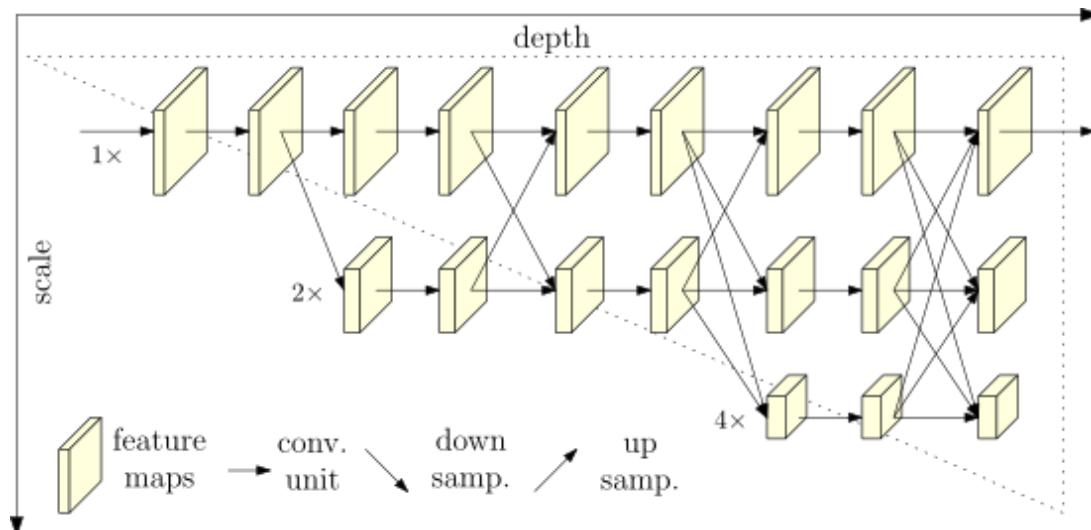


Рис. 9: Архитектура модели HrNet. [12]

Особенность архитектуры HRNet является возможность поддержи-

вать высокое разрешение изображений на всех этапах обработки, что позволяет достичь высокой точности и детальности предсказаний. Достигается это за счет параллельного использования нескольких ветвей с разным разрешением и постоянного обмена информацией между ними (см. рис. 9).

Все начинается с уровня слоев высокого разрешения, к которому параллельно добавляются слои с пониженным разрешением. Периодически между ветвями происходит обмен информацией с помощью fusion module. Этот процесс обеспечивает сбалансированное и детализированное представление на разных уровнях разрешения, что приводит к улучшению точности получаемого результата.

BlazePose

Архитектура нейросети BlazePose разработана в 2020 году исследователями от Google Research и известна своим использованием в работах проекта MediaPipe [13, 4]. Она предназначена для быстрого и точного распознавания ключевых точек тела в реальном времени, даже на мобильных устройствах и в условиях ограниченных вычислительных ресурсов. Расширенная до 33 точек топология модели помогает исполь-

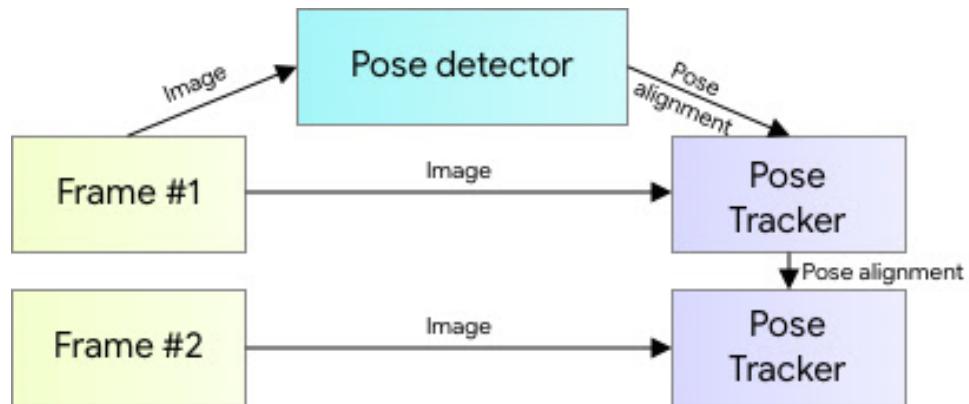


Рис. 10: Структура модели BlazePose для работы в реальном времени.

[13]

зователь нейросеть в разнообразных спортивных приложениях, таких как фитнес-трекеры и анализаторы асан йоги.

BlazePose использует top-down подход оценки позы, который оптимизирован для работы с видеопотоками. Схематично, структура нейросети представлена на рис. 10. На первом этапе необходимо найти человека на входном изображении, чем и занимается PoseDetector. Но он вызывается только для первого кадра и возвращает не только координаты области с человеком, а информацию об интересующей нас области (region of interest или ROI). Именно ROI дает возможность не вызывать каждый раз детектор, так как она изменяется на втором этапе работы сети и передается сразу для использования в Pose Tracker следующего кадра.

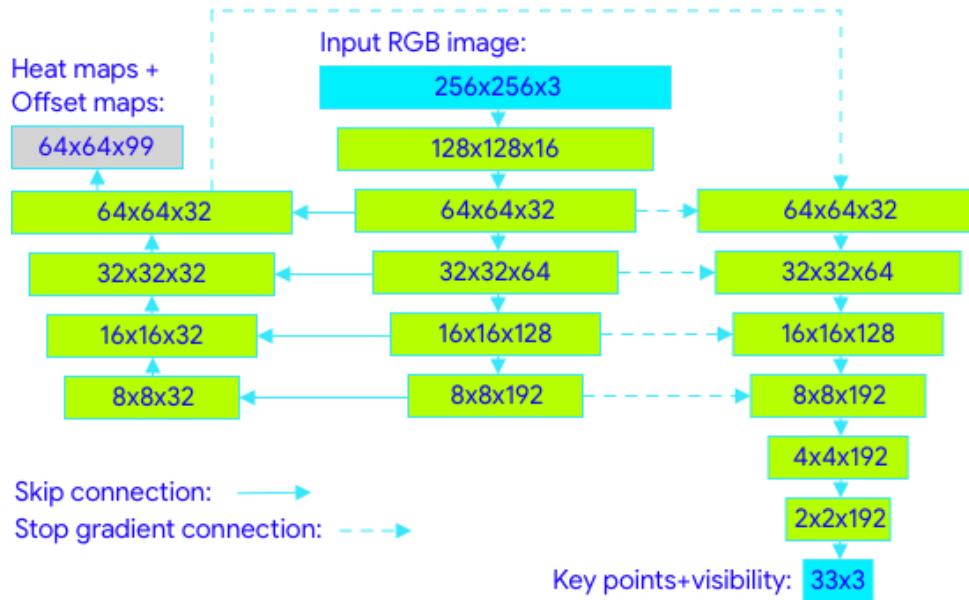


Рис. 11: Архитектура Pose Tracker. [13]

Следующим шагом является распознавание КТ в заданной области интереса. Путем использования нескольких пирамидальных архитектур производится комбинированный анализ тепловой карты и данных о смещении (см. рис. 11).

ViT Pose

ViT Pose - это модель для распознавания ключевых точек, которая была представлена в 2022 году командой исследователей из университета Сиднея [14]. В ее основе используется архитектура трансформера, которые изначально были разработаны для обработки последовательностей в задачах обработки естественного языка, адаптированного для задач компьютерного зрения (англ. Vision Transformer или ViT) [15]. Данный подход продемонстрировал высокую эффективность и конкурентоспособность по сравнению с традиционными сверточными нейронными сетями (CNN), что позволило область решений задачи оценки позы.

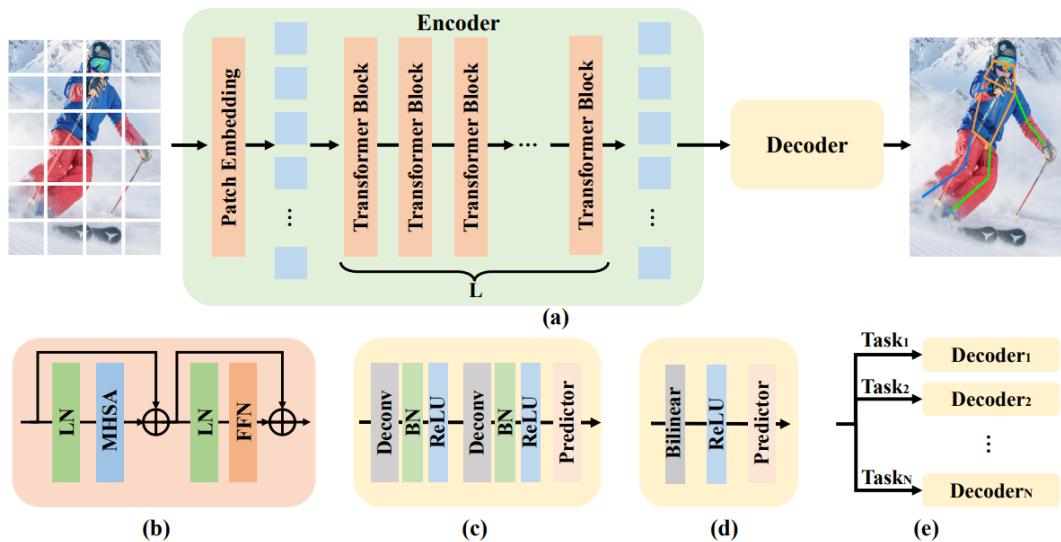


Рис. 12: Структура VitPose (a). Блок трансформера (b). Классический декодер (c). Простой декодер (d). Декодеры для нескольких наборов данных (e). [14]

Для создания анализируемой последовательности изображений разбивается на патчи размером 16×16 . Они проходят через линейные слои нейросети для получения эмбеддинга, к которому дополнительно добавляется позиционный эмбеддинг, чтобы модель могла учитывать пространственное расположение блоков. И потом уже полученная це-

почка векторов передается на вход магистральной части нейронной сети. Используя механизмы самовнимания (англ. self-attention), которые помогают использовать различные взаимосвязи между частями изображения, backbone возвращает признаковое описание изображения.

Для извлечения признаков и локализации ключевых точек из результатов магистральной части используется два типа декодеров. Классический декодер использует несколько блоков для повышения дискретизации тепловой карты ключевых точек, а для локализации координат используется сверточный слой в качестве выходного слоя. Другой декодер, называемый в работе «простым» [14], использует один слой билинейной интерполяции для повышения детализации результатов, а для получения тепловых карт используется функция активации RELU. Предсказателем также выступает сверточный слой.

Simcc

Локализация ключевых точек на основе анализа тепловых карт достоверности является весь распространенным подходом в задаче оценки позы. Но даже этот подход имеет некоторые недостатки: плохие результаты на изображениях низкого разрешения, вычислительно тяжелые слои повышения дискретизации тепловых карт и дополнительная постобработка для уменьшения ошибок квантования. Чтобы избежать их исследователями был разработан алгоритм Simple Coordinate Classification или SimCC [16], который предлагает новый подход к оценке поз человека с помощью классификации отдельно горизонтальных и вертикальных координат. Схематическое описание алгоритма представлено на рис. 13.

Для извлечения признаков в данном алгоритме используется магистральная нейронная сеть, которой могут быть как сверточные нейронные сети, так и трансформеры. Она возвращает признаковое описание необходимого количества ключевых точек.

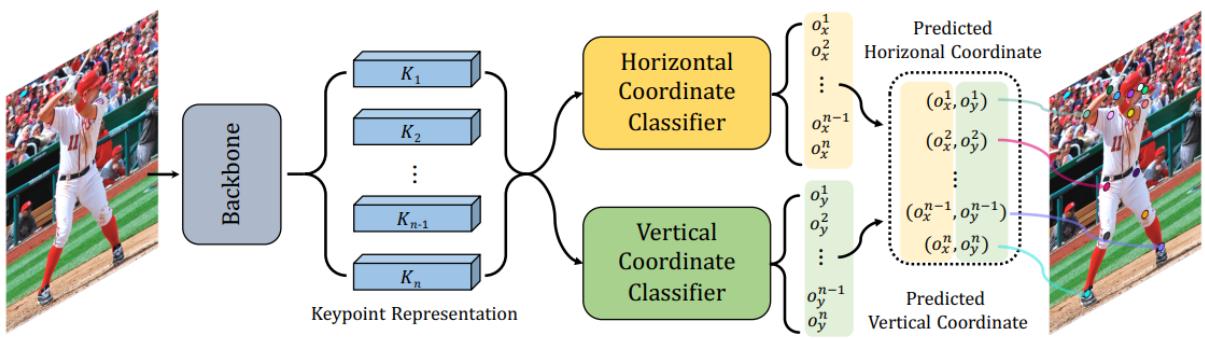


Рис. 13: Структура SimCC [16]

Для дальнейшего шага все координаты дискретизируются на более мелкие бины, чтобы избежать ошибок квантования. Далее классификатор выставляет метки всем бинам, причем делает это независимо для вертикального и горизонтального направления. А для обучения модели используется функция потерь на основе дивергенции Кульбака-Лейблера (англ. Kullback-Leibler divergence).

В итоге алгоритм показывает немного более лучшие результаты, в сравнение с решениями на основе анализа тепловых карт, но сильно уменьшает количество проводимых операций, чем значительно улучшает скорость работы модели.

YoloPose

Yolo-Pose - это семейство решений на основе архитектуры Yolo, которое представляет собой новаторский подход для одновременного обнаружения нескольких человек на фото и распознавания их скелета. Алгоритм был предъявлен публике в 2022 году исследователями из Техаса [17]. В своей статье они использовали архитектуру Yolo5, которая показывала хорошие результаты распознавания ключевых точек на датасете COCO. На сегодняшний день к семейству Yolo-Pose подтянули и других представителей популярной архитектуры, таких как Yolo8, YoloX и Yolo-NAS [18, 19].

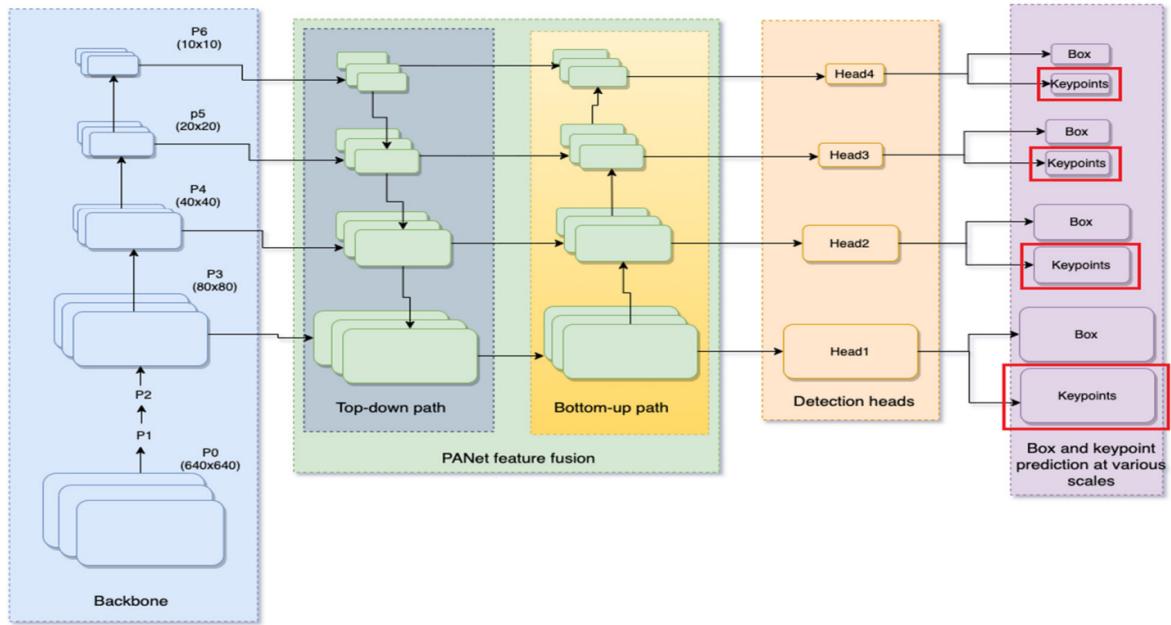


Рис. 14: Архитектура Yolo-Pose на основе модели Yolo5. [17]

Авторы решили пересмотреть использование различных подходов к распознаванию ключевых точек и совместили лучшее из обоих вариантов. Большинство подходов сверху-вниз используют тепловые карты для предсказания координат ключевых точек. Как уже было сказано в разд. 4.1, детализация тепловых карт занимает множество вычислительных ресурсов. А ещё производительность этих решений напрямую зависит от количества человек на изображении, так как все они распознаются по отдельности. В свою очередь подходы снизу-вверх имеют сложность при сопоставлении ключевых точек отдельным людям. В итоге получилось построить решение, которое производит поиск человека на фото, присваивает ему некоторую якорную точку (англ. anchor point), с которой впоследствии ассоциируются все ключевые точки и ограничивающая рамка. А разделение на две сущности: bbox и keypoints происходит на этапе обработки полученных признаков (head) и предсказания результатов.

Полученный алгоритм сквозного обучения не мог работать на существующих L1 метриках, поэтому исследователи оптимизировали мет-

рику OKS для обучения моделей (о метрике рассказано в разд. 5). Это дало возможность учитывать весовые коэффициенты точек и масштаб объекта при обучении.

Но у моделей этого семейства есть и минус. Это требования большого количества вычислительных ресурсов для обучения моделей. Что делает практически невозможным улучшение модели простым обывателем.

RTMPose

Из текущей подборки данная архитектура является самой молодой. Алгоритм создан в 2023 году и специально оптимизирован для распознавания поз нескольких человек в реальном времени, о чем говорит его название: «RTMPose: Real-Time Multi-Person Pose Estimation» [20]. Он вобрал в себя лучшие методики последних лет и показывает очень хорошие результаты как в точности, так и в скорости инференса.

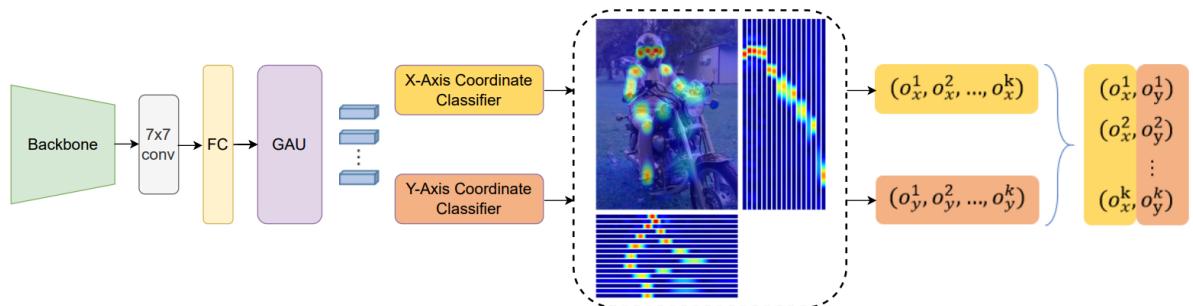


Рис. 15: Общая архитектура RTMPose. [20]

RTMPose поддерживает top-down подход, поэтому для работы ему необходим детектор. В рамках проекта MMPose используется детектор RTMDet, разработанный в той же лаборатории [21, 22]. Для ускорения работы нейросети на видеозаписях используется идея, предложенная в BlazePose [13]: использование детектора только на первом кадре, при работе алгоритма вносить изменения в ROI и уже измененные прямоугольники передавать для следующего кадра без задействования дополн-

нительной модели. Это позволяет получать более высокие показатели скорости работы в реальном времени.

Рассмотрим архитектуру нейронной сети, которая представлена на рис. 15. Для извлечения признаков используется базовая нейронная сеть CSPNeXt, которая показывала хорошие результаты в задаче детекции объектов. Для выуживания координат из полученной карты признаков используют описанный ранее алгоритм SimCC [16]. Таким образом с помощью классификаторов предсказываются отдельно горизонтальные и вертикальные координаты точек, что показывало 69.7 AP на датасете COCO. Однако и текущие результаты получилось улучшить.

Было замечено, что при увеличении размерности полученных признаков улучшается результат предсказания. Поэтому между магистральной сетью и классификаторами добавили полно связный слой, который увеличивает размерность полученных представлений ключевых точек. Также после этого добавлен модель самовнимания, основанный на Gated Attention Unit (GAU). Это изменение обеспечило более внимательное использование глобальной и локальной пространственной информации.

В итоге предложенный алгоритм достиг 75.8 AP на датасете COCO и скорости 90 кадров в секунду без использования графических ускорителей.

4.2 Обзор методов доменной адаптации на неразмеченных данных

Progressive Unsupervised Learning

Встречаясь с чем-то новым, люди могут несколько раз рассматривать, пробовать и анализировать новый предмет, прежде чем получат необходимые знания о нем. Похожую схему использует алгоритм прогрессивного обучения без учителя (англ. progressive unsupervised learning или

PUL). Его концепция позволяет избавиться от необходимости аннотировать данные для дообучения нейронных сетей на них, тем самым позволяя моделям адаптироваться к новому домену. Первоначально модель была применена в задаче отслеживания объектов [23], а позже использована в задаче повторной идентификации человека [24].

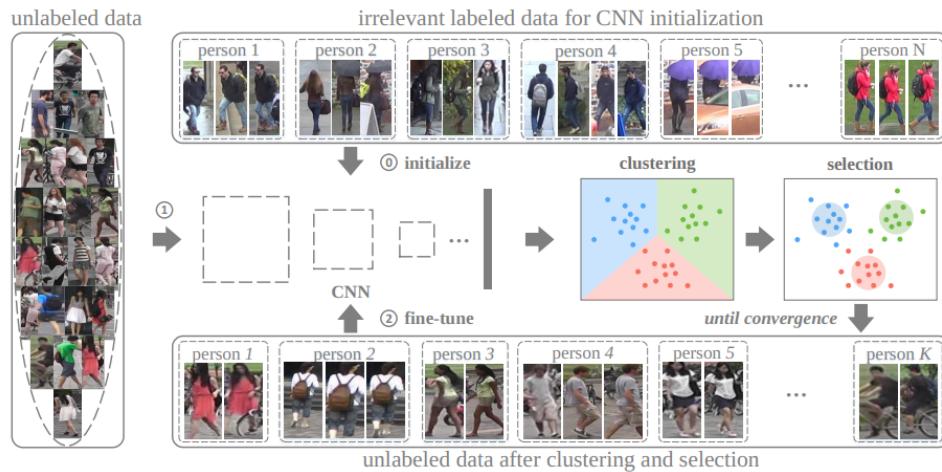


Рис. 16: PUL [24]

Основная концепция алгоритма, как метода unsupervised domain adaptation, состоит в отсутствии необходимости размечать данные, а адаптироваться к ним посредством итеративного обучения на псевдо-разметке. Опишем действия, которые будут выполняться повторно.

1. Входные данные

Для начала работы алгоритма необходимо иметь некоторое предварительно обученное состояние модели.

2. Псевдо-разметка данных

Неразмеченные данные прогоняются через модель для получения некоторых результатов. Из-за того, что модель имеет некоторую ошибку, текущие результаты называются псевдо-разметкой.

3. Фильтрация невалидных результатов

Модель может значительно ошибаться в предсказании псевдо-разметки,

поэтому необходимо обозначить некую функцию фильтрации шумных значений. Например, при применении PUL к задаче повторной идентификации [24], была построена интересная функция фильтрации на основе кластеризации вектора признаков изображения и отбрасывания всех значений, значительно далеких от центра кластера.

4. Обучения модели

Производится дообучение модели на полученных данных. Это состояние модели передается на вход следующей итерации алгоритма.

Маркерами для остановки итеративного, помимо заранее определенного количества итераций, процесса могут служить:

- Стабильность кластеров

Когда кластеры псевдо размеченных данных стабилизируются и изменения в них между итерациями становятся минимальными.

- Устойчивость к шуму

Когда количество зашумленных данных становится минимальным или приемлемым.

- Сходимость функции потерь

Если функция потерь, используемая для обучения модели, перестает значительно уменьшаться между итерациями, то это указывает на сходимость модели.

- Критерий качества результатов

Когда достигается определенное заранее выбранное значение метрики качества на целевых данных. Данный метод хорош при небольшом количестве размеченных данных на целевом домене.

Regressive Domain Adaptation

Большинство современных алгоритмов доменной адаптации без учителя изначально созданы для применения в задачах классификации и могут не работать в задачах регрессии, например к задаче распознавания ключевых точек. Поэтому был предложен алгоритм регрессивной доменной адаптации (англ. Regressive Domain Adaptation или RegDA) [25].

В решениях задачи распознавания ключевых точек имеется две части: магистральная нейронная сеть, которая используется для генерации признаков, и извлекающий алгоритм (англ. extractor) или регрессор, который из признаков предсказывает координаты ключевых точек. Первую часть будем в дальнейшем обозначать ψ , а последнюю - f .

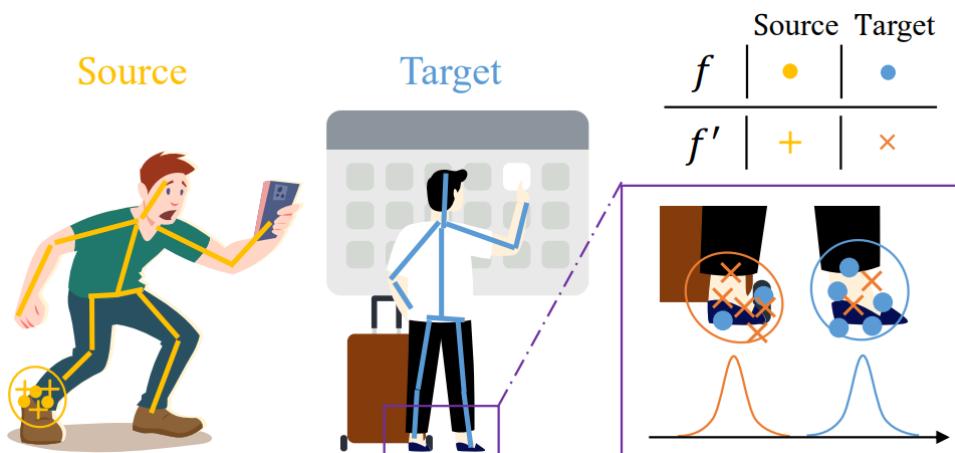
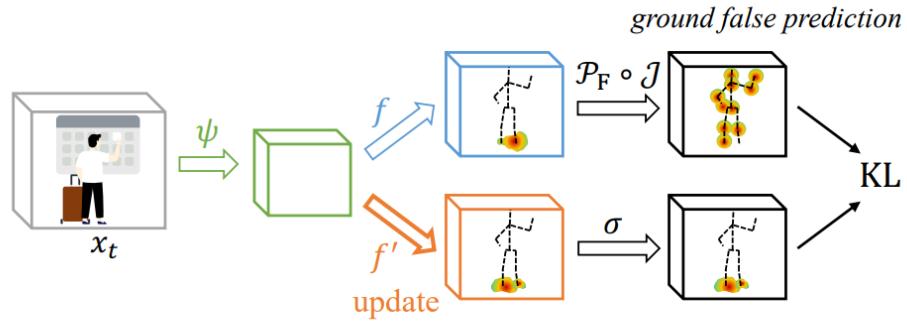


Рис. 17: Прогнозирование результатов f и f' на данных исходного (source) и целевого (target) доменов. [25]

Суть алгоритма состоит в использовании состязательного регрессора f' , который обучается допускать ошибки в предсказаниях на целевом домене, без потери качества на исходном. Другими словами максимизируется расхождение в предсказаниях с помощью расстояния Кульбака-Лейблера. Чтобы уменьшить вычислительную сложность обучения состязательного регрессора авторы предложили строить вероятностные карты на основе предсказания f . Таким образом f' не просто вносил шум,

предсказывая точку рядом с ее значением, а допускал ошибку в предсказании самой ключевой точки (см. рис. 17).

Objective 2: **Maximize disparity on target** (Fix ψ and f , update f')



Objective 3: **Minimize disparity on target** (Fix f , f' , update ψ)

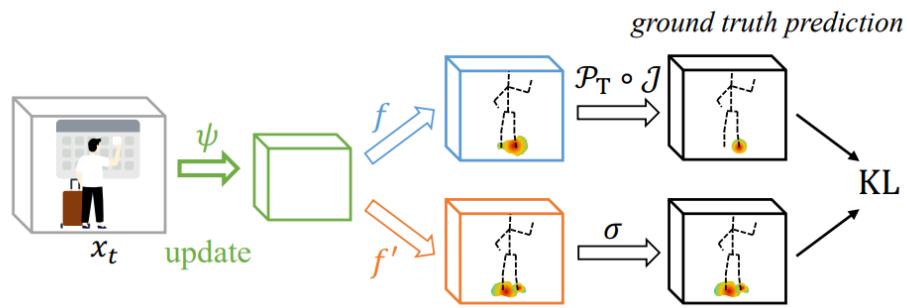


Рис. 18: Обучение состязательного регрессора и обучение генератора признаков.[25]

После получения достаточного расхождения между предсказаниями двух регрессоров происходит обучение генератора признаков. Во время этого процесса генератор старается уменьшить полученную разницу в распределениях между извлекающими алгоритмами. Таким образом, около верных предсказаний создаются дополнительные опознавательные знаки, которые, в итоге, улучшают результат. Описанные шаги представлены на рис. 18

UDA PoseEstimation

Авторы следующего алгоритма задались вопросом адаптации распознавания ключевых точек через перенос стиля изображений между домена-

ми [26], причем в обоих направлениях. Дополнительно к этому используется парадигма среднего учителя (англ. MeanTeacher) для генерации псевдо-разметок целевым данным. Рассмотрим детальнее шаги алгоритма, представленного на рис. 19.

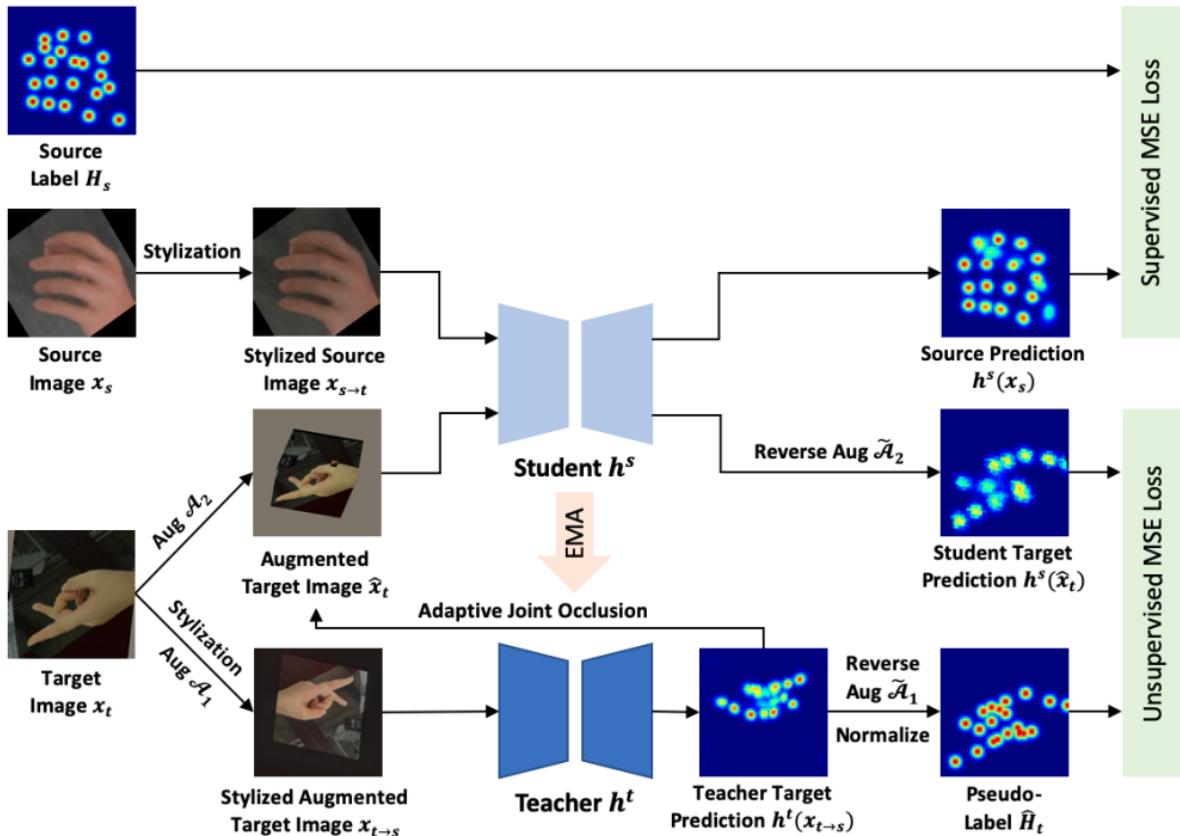


Рис. 19: Идея работы алгоритма UDA Pose estimation. [26]

Исследователи из МИТ использовали алгоритм AdaIN [27] для переноса стиля сразу в нескольких местах. В первом случае происходит трансфер из исходного домена в целевой. Таким образом получается часть ошибки, которая формируется на размеченных данных L_{sup} .

Другой перенос происходит в другом направлении, из целевого в исходный, для предсказания псевдо-меток обучения изображения из неразмеченного домена. Псевдо-метки получаются с модели среднего учителя, получаемой с помощью усреднения весом студенческой модели. Учитывая, что сгенерированные псевдо-метки могут иметь ошибку, применяется алгоритм фильтрации наиболее шумных результатов через порог до-

верия. Прогоняя целевое изображение через модель студента мы получаем некоторое предсказание. Сравнив его с полученными псевдо-метками, мы получаем часть ошибки на неразмеченных данных L_{unsup} .

Итоговая ошибка выступает комбинацией двух частей $L = L_{sup} + \lambda L_{unsup}$ и позволяет модели обучаться сразу на обоих доменах данных.

5 Эксперимент

5.1 Описание эксперимента

В рамках данного исследования было решено провести эксперимент по адаптации моделей распознавания ключевых точек на теле человека к целевому набору данных. В представленном разделе будет рассказано про выбранные модели, алгоритм доменной адаптации, описание исходного и целевого наборов данных и метрик оценивания результатов работы нейронных сетей.

Выбор модели для эксперимента

Многие модели из представленных в разд. 4 были представлены в рамках проекта MMPose, что значительно облегчило эксперимент с точки зрения подготовки пайплайнов предобработки данных и построения конфигураций нейронных сетей. Также критерием отбора стала возможность обучения модели в условиях ограниченных вычислительных ресурсов, так как адаптация проводилась в рамках платформы Google Colab.

Название модели	Количество параметров	Размер входящего изображения	Время предсказания
HRNet	28,5 М	256×192	203,7 мс
ViTPose	90 М	256×192	269,8 мс
SimCC + Resnet	36,8 М	256×192	101 мс
RTMPose	13 М	256×192	55,4 мс

Таблица 1: Характеристики моделей, выбранных для эксперимента.

В итоге было выбрано 4 модели, описание которых представлено в

табл. 1. Они будут обучены на исходном домене в течении 20 эпох. От этого состояния будет начинать при проведении адаптации.

Описание метода доменной адаптации

В рамках эксперимента будет оптимизирован метод Progressive Unsupervised Learning [23] для задачи распознавания ключевых точек.

Основной сложностью выступает выбор функции фильтрации невалидных результатов. И в рамках эксперимента предлагается использовать фильтрацию по уровню уверенности модели в предсказанной ключевой точке. Средняя уверенность полученных из предсказания результатов для одного изображения будет сравниваться с заранее заданным пороговым значением. Все значения, перешедшие этот порог попадают в псевдо-обучающую выборку.

Заметим, что для точек, которые не являются видимыми, уверенность сильно меньше, что может портить среднюю уверенность для фотографии. Таким образом получаем ещё одну функцию фильтрации - по средней уверенности для видимых точек результата.

На каждой итерации будет проводиться дообучение модели в течение 10 эпох. Количество итераций будет ограничено либо сходимостью метрики качества, либо количеством 10 штук.

Метрики оценки качества распознавания

Для проведения количественной оценки работы алгоритма во всех случаях необходимо использовать метрики оценки предсказания. Выбранные метрики дают полную оценку того, насколько хорошо модель работает, как с точки равнозначности всех точек в топологии, так и с учетом их веса в скелете.

1. Percentage of Correct Keypoints

Первой рассмотрим метрику РСК, которая равнозначно воспринимает все ключевые точки. Для нее важно попало ли предсказания в окрестности реального результата, причем размер окрестности может быть выбран как фиксированный для всего тестового набора, так и зависеть от высоты человека на изображении.

Математическую формулу метрики можно представить в следующем виде:

$$PCK = \frac{\sum_{i=1}^n \text{bool}(d_i < \text{threshold} * \text{body_height})}{n}, \quad (6)$$

где d_i - расстояние между предсказанной и правильной точкой,
 threshold - порог, задаваемый исследователем,
 body_height - высота прямоугольника, внутри которого находится человек,
 $\text{bool}(\ast)$ - логическое условие, возвращает 1, если оно верно и 0 в ином случае,
 n - размер выборки.

2. Object Keypoint Similarity

OKS была представлена для оценки решений задачи распознавания ключевых точек в рамках соревнования COCO [10]. Авторы старались провести аналогию с метрикой Intersection over Union (IoU) для задачи детекции объектов, чтобы можно было пользоваться метрикой average precision, о которой будет рассказано далее.

В отличие от предыдущей метрики, OKS не считает все точки равнозначными. Для этого проводится процедура нормализации расстояния между предсказанной и реальной точками. Одним из шагов нормализации является учет размера детектируемого объекта,

что показывает различия для детекции скелета людей на заднем и переднем планах. Вторым шагом нормализации является учет дисперсии данной ключевой точки.

Математическая формула метрики выглядит следующим образом:

$$OKS = \frac{\sum_i \exp\left(-d_i^2/2s^2k_i^2\right) \delta(v_i > 0)}{\sum_i \delta(v_i > 0)}, \quad (7)$$

где d_i - расстояние между предсказанной и правильной точкой,

s - масштаб объекта,

k_i - константа ключевой точки, контролирующая спад,

v_i - видимость точки по аннотации COCO, где 0 обозначает, что точка не была размечена.

3. Average Precision

AP - метрика, широкоиспользуемая для оценки качества моделей в задаче классификации и обнаружения объектов. Она измеряет точность нейронной сети, принимая во внимание как ее способность правильно классифицировать объекты, так и ее способность точно их локализовать.

Исходя из названия метрика использует такие понятия как Precision, показывающее долю правильно предсказанных положительных примеров среди всех положительно предсказанных результатов, и Recall, показывающее долю правильно предсказанных положительных примеров среди всех фактических положительных примеров.

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}$$

При использовании OKS верно положительным результатом является тот, для которого метрика OKS больше заданного порогового

значения. Неверно положительными являются те результаты, метрика OKS для которых не перешла заранее заданный порог. Неверно отрицательными являются все реальные данные, для которых результата получено не было.

$$TP \text{ при } OKS > threshold \text{ и } FP \text{ при } OKS \leq threshold$$

Используя описанные пояснения можно рассчитать значения метрик точность и полнота. С их помощью строится кривая precision-recall, которая показывает изменения точности в зависимости от полноты. Площадь под ее графиком и будет являться значением метрики AP. Часто она высчитывается интегрирования кривой precision-recall методом трапеций, который показан на рис. 20.

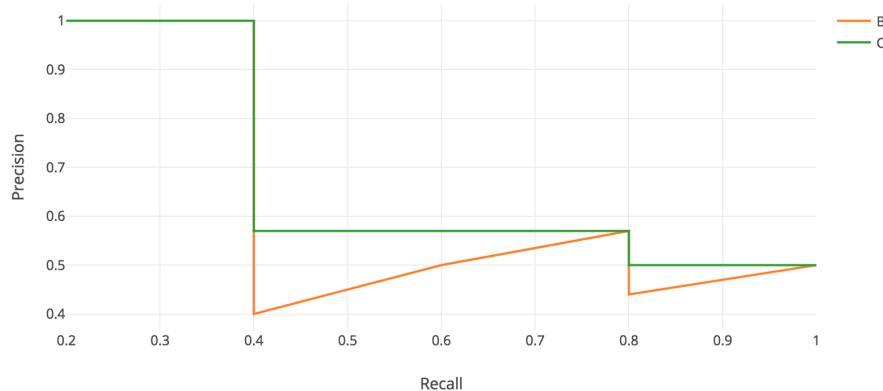


Рис. 20: Применения метода трапеций для интегрирования кривой precision-recall.

Для задачи детекции обычно применяется несколько порогов: $AP^{0.5}$, $AP^{0.75}$ и $AP = mAP = meanAP^{0.5:0.05:0.95}$.

5.2 Данные

По условиям задачи доменной адаптации необходимо найти два набора данных для эксперимента. Далее приведем информацию о выбранных доменах и их характеристиках.

Исходный домен

В качестве исходного домена выбран набор данных Common Objects in Context [9]. COCO — это крупный датасет, широко используемый в области компьютерного зрения для задач распознавания объектов, сегментации, и создания описательных подписей к изображениям. Он был создан Microsoft и с тех пор стал стандартом для обучения и оценки алгоритмов компьютерного зрения.



Рис. 21: Примеры изображений из набора данных COCO. [10]

Учитывая, что в рамках соревнований COCO была и задача детекции ключевых точек (Keypoint detection) [10], то часть этого набора данных была размечена под нее. Если быть точным, то датасет включает более 250 тысяч аннотаций людей на различных изображениях. Формат аннотаций включает в себя:

1. *id* - уникальный номер аннотации;
2. *image_id* - уникальный номер изображения, которому принадлежит данная аннотация;
3. *category_id* - уникальный номер категории, к которой относится

данная аннотация. Для задачи оценки позы везде выставляется категория person;

4. *keypoints* - массив из 17 ключевых точек, для каждой из которых указаны координаты (x, y) на изображении, а также информация о видимости. Точки, которые не представлены на изображении заполняются нулями;
5. *num_keypoints* - здесь содержится информация о количестве размещенных точек для данной аннотации;
6. *bbox* - информация об ограничивающем человека прямоугольнике. Значения внутри лежат в следующем формате: $[x, y, width, height]$;
7. *area* площадь сегментированного человека. Значение необходимо при высчитывании метрики OKS;
8. *iscrowd* - информация о том, одиночный человек представлен на изображении или толпа людей.

Также в рамках задачи Keypoint Detection была введена метрика OKS и метрика mAP, о которых было рассказано ранее. Они представляют собой единые критерии для оценки моделей, что облегчает сравнение и улучшение результатов различных алгоритмов, поэтому регулярно используются для оценки новых методов и технологий.

В рамках задачи были выбраны 8000 аннотаций, которые содержат все 17 ключевых точек топологии COCO. На них и было произведено обучение моделей для получения бейзлайнов эксперимента.

Целевой домен

В качестве целевого набора данных был собран отдельный набор данных боксеров. В наборе данных представлены 2 человека, снятые с 3 ракур-

сов: профиль, анфас и 3/4. Датасет состоит из 10 видеозаписей, которые содержат порядка 10 тысяч кадров. Для проведения эксперимента выбрано 2,6 тысячи изображений, которые были впоследствии размечены. Из них для тестовой выборки отобрано около 420 изображений, а оставшиеся 2200 составили обучающую выборку, на которой и проводилась адаптация.



Рис. 22: Примеры изображений целевого домена.

В рамках задачи по аннотированию собранных данных была разработана система полуавтоматической разметки изображений pose-markup [28]. Она представляет собой предобученную модель распознавания ключевых точек и инструмент для визуальной корректировки данных экспертом.

Для автоматической части использовалась модель BlazePose от проекта MediaPipe [4]. Выбор сделан благодаря высоким характеристикам скорости инференса результатов и их точности у данного решения. А также для того, чтобы избежать корреляции размеченных данных с предсказаниями, которые будут оцениваться в рамках эксперимента. Результат, возвращаемый моделью был преобразован к формату аннотаций COCO, который был описан выше и сохранен в формате JSON.

Как можно видеть на рис. 23, модель имеет неточности, которые

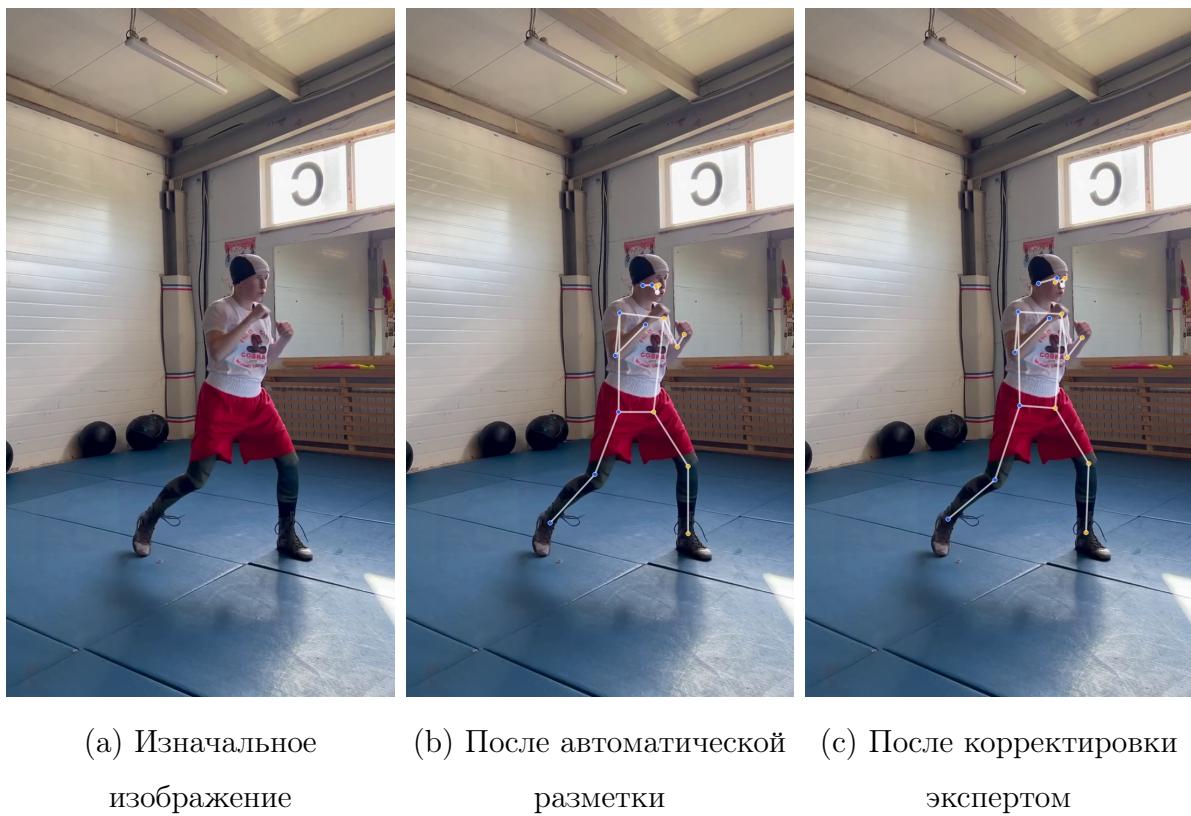


Рис. 23: Пример работы системы полуавтоматической разметки данных.

необходимо было исправить эксперту. Для этой цели использовалась вторая часть программы - инструмент для визуальной корректировки данных экспертом.

Проанализировав результаты внесенных изменений, получили, что каждый кадр требовал правки ключевых точек. Средние показатели изменений на кадр:

- Собирая статистику любых изменений в точках, даже сдвига на соседний пиксель, получаем, что в среднем 14.9 точек на кадре были изменены. Распределение этих исправлений по топологии представлено на рис. 24а.
- Если считать точки, которые были значительно сдвинуты (5 и более пикселей), то их в среднем приходилось 6.5 на кадр. Большинство из них были сосредоточены на лице и левой конечности человека. Более детальное распределение изменений по топологии представлено на

рис. 24b.

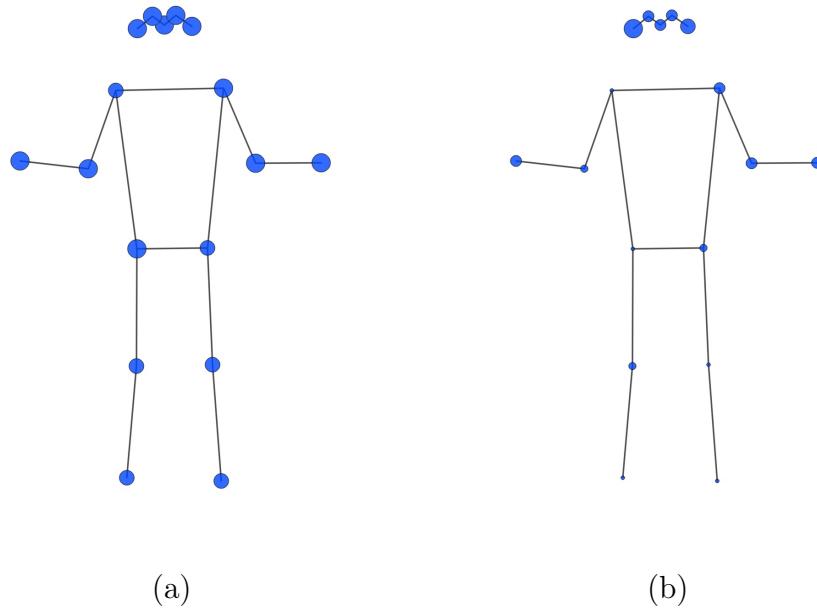


Рис. 24: Схематическое представление изменений в точках, внесенных экспертом.

5.3 Результаты эксперимента

В рамках эксперимента для выбранных ранее моделей был применен алгоритм доменной адаптации PUL, который показал смешанные результаты на разных архитектурах. В качестве функции фильтрации было выбрано отсеивание результатов предсказания поз на основе средней уверенности в предсказанных точках. Порог уверенности для каждой модели выбирался исходя из результатов первых итераций алгоритма. Так как на первых из них уверенность модели в своих результатах падала из-за внесения обучения на псевдо-данных. Влияние фактора подбора порога будет оценено далее.

адаптированная модель сравнивается с базовыми весами, которые обучены на исходном домене, а также с дообученной на размеченных це-

левых данных. Это позволит оценить полезность и применимость работы алгоритма доменной адаптации.

Тестирование моделей производилось с использованием графического ускорителя NVIDIA MX250 с 2 Гб памяти. Так как объемов памяти этого ускорителя не хватало для обучения моделей, для обучения моделей была совершена миграция в облачный сервис Google Colab с использованием предоставляемого там графического ускорителя NVIDIA Tesla T4 с 16 Гб памяти.

HRNet

В рамках первого эксперимента использовалась архитектура HRNet. Так как уверенность на бейзлайне не была высокой, то пришлось выставить порог уверенности в 0.4. В связи с этим для итеративных процессов отбирались данные, которые не являются точными и вносят большую ошибку. Это стало причиной того, что с каждой итерацией модель показывала все более плохие результаты, которые можно увидеть на рис. 25.

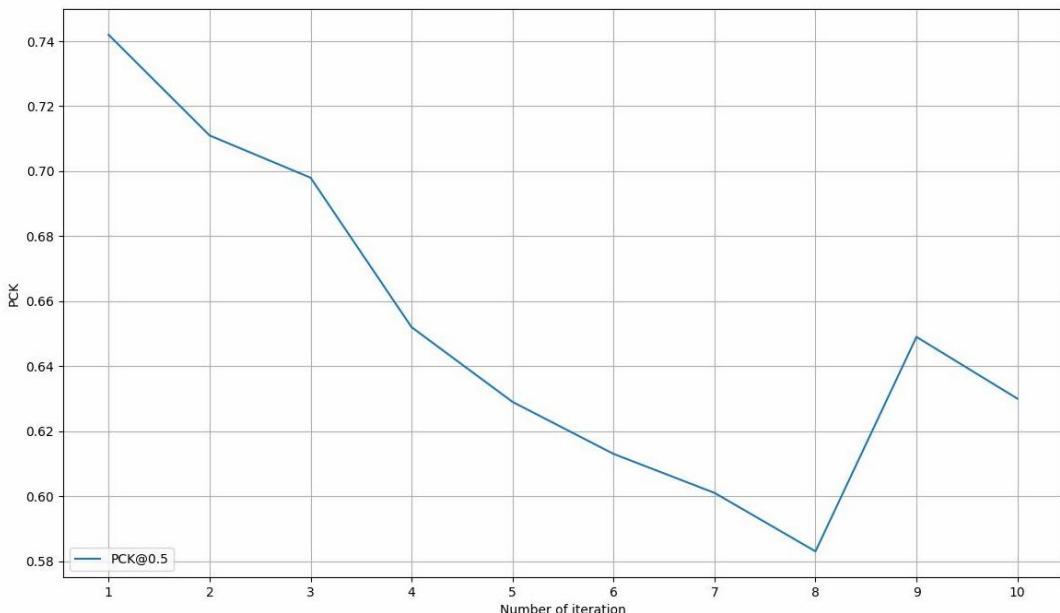


Рис. 25: Зависимость метрики РСК на целевом домене от номера итерации.

Остальные результаты для данной модели не снимались ввиду плохих результатов адаптации. В дальнейшем значение порога будет выбираться не ниже 0.5.

ViT Pose

Исходя из опыта предыдущего эксперимента, порог уверенности был выбран равным 0.7. Особенностью данного эксперимента явилось то, что модель быстро достигла высокой уверенности в своих результатах, из-за чего объемы псевдо-разметок были равными объему всей выборки для адаптации. Из рис. 26 видно, что итеративный процесс можно было остановить на 4 этапе, что могло сэкономить время эксперимента. Это состояние и будет считаться адаптированным в рамках данного эксперимента.

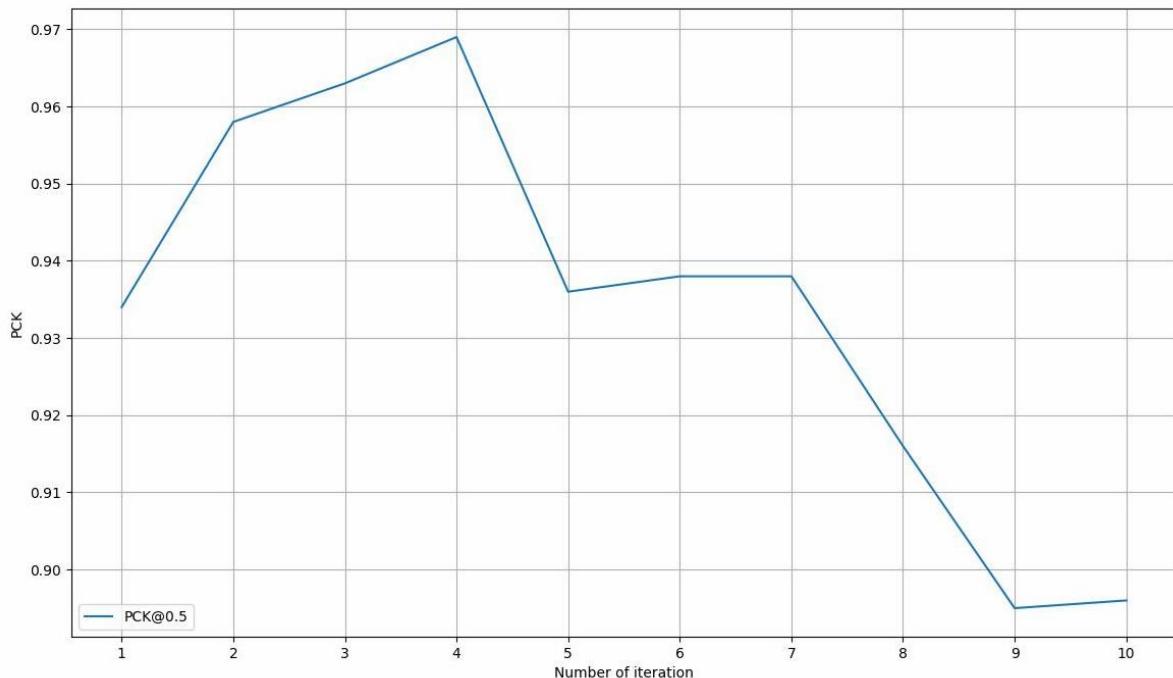


Рис. 26: Зависимость метрики РСК на целевом домене от номера итерации для модели ViTPose.

Метрики, снятые на разных версиях модели, показаны в табл. 2. Можно заметить, что адаптированная модель предсказывает больше то-

чек, по сравнению с другими моделями, но точность предсказания показывает значительно более плохие результаты.

	PCK@0.05	PCK@0.25	PCK@0.5
Baseline	0.147	0.725	0.902
Adapted model	0.118	0.623	0.969
Finetune model	0.21	0.789	0.948
	Время обучения	Время предсказания	
Baseline	118 мин	250,2 мс	
Adapted model	10 мин	248,3 мс	
Finetune model	30,5 мин	246,5 мс	

Таблица 2: Сравнительная статистика нескольких состояний модели ViTPose.

На рис. 27 показано распределение адаптированной метрики РСК применительно ко всем точкам топологии в отдельности. Как можно заметить на рис. 27a, у базового состояния возникли проблемы с точками внешних конечностей, но и дообучение, и адаптация улучшили их предсказание. Но при уменьшении радиуса допустимой ошибки, мы наблюдаем на рис. 27b, что адаптированная модель показывает хорошие результаты только для точек нижних конечностей. Причем точность предсказания точек головы ухудшается для обоих попыток изменения модели.

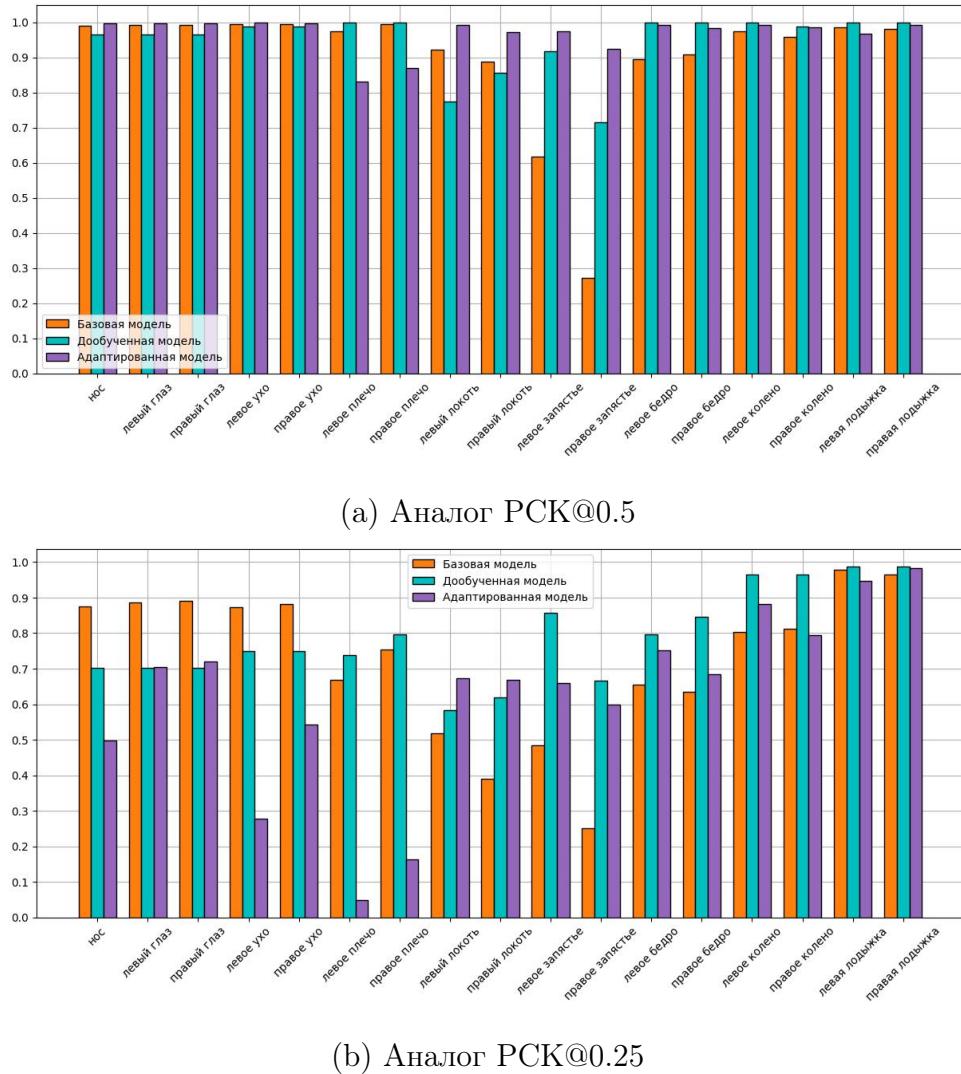


Рис. 27: Распределение корректно предсказанных точек по всей топологии для модели ViTPose.

SimCC

Для текущей модели был выбран порог достоверности равный 0.6. Тренд основной метрики качества итеративного обучения, представленный на рис. 28а, не является монотонным и, как и в предыдущем эксперименте, наибольшее значение имеет в первой половине итераций. Но при анализе более точных результатов на рис. 28б видим, что точность модели продолжает улучшаться.



Рис. 28: Зависимость метрики РСК на целевом домене от номера итерации для алгоритма SimCC с магистральной сетью ResNet.

Данный эксперимент показывает наилучшее качество улучшения результатов предсказаний с помощью алгоритма адаптации. Проигрывает он только дообученной модели в количестве предсказаний с низкой возможной ошибкой.

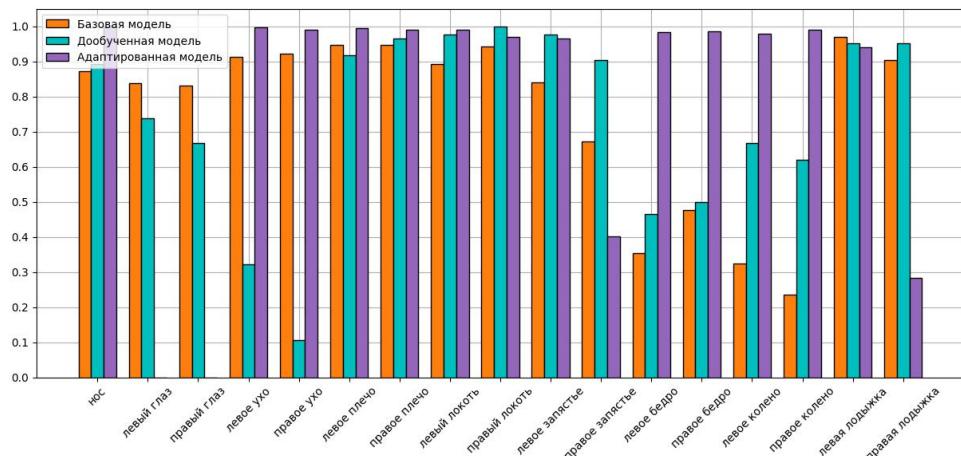
Также из табл. 3 видно, что дообучение модели не сильно увеличивает количество точек, которые можно считать распознанными. Также видно, что адаптация и дообучение увеличивают высокую точность предсказаний с 1,5 и почти в 3 раза соответственно.

	PCK@0.05	PCK@0.25	PCK@0.5
Baseline	0.024	0.429	0.758
Adapted model	0.037	0.445	0.792
Finetune model	0.062	0.411	0.76

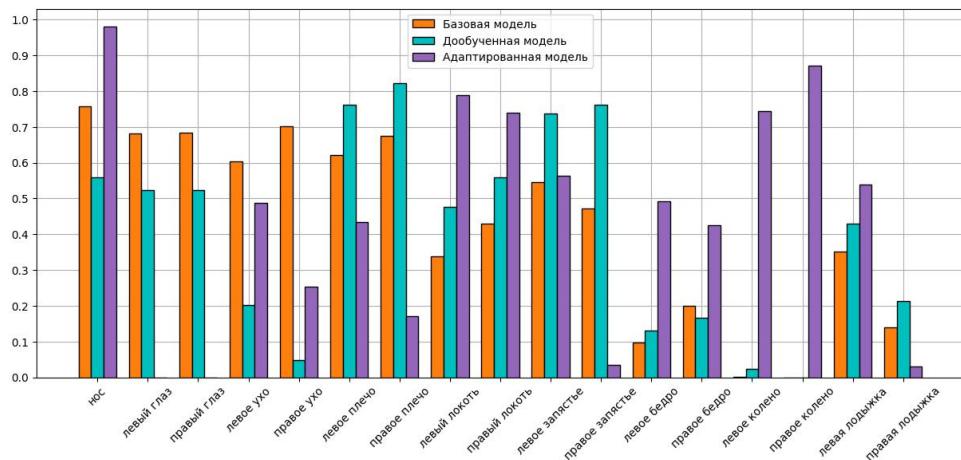
	Время обучения	Время предсказания
Baseline	80 мин	83,5 мс
Adapted model	7,75 мин	82,5 мс
Finetune model	24 мин	82,3 мс

Таблица 3: Сравнительная статистика нескольких состояний для алгоритма SimCC с магистральной сетью ResNet.

В рамках детального предсказанных точек на рис. 29, можно заметить, что адаптированная модель очень плохо предсказывает точки глаз, правых запястья и лодыжки. Для метрики в большим трешхолдом (см. рис. 29a) в остальных точках адаптированная модель либо показывает наилучшие результаты, либо сопоставимые с базовой моделью. Но для при более точном распознавании (см. рис. 29b) заметно ухудшение предсказаний для точек лица, кроме носа, и плеч.



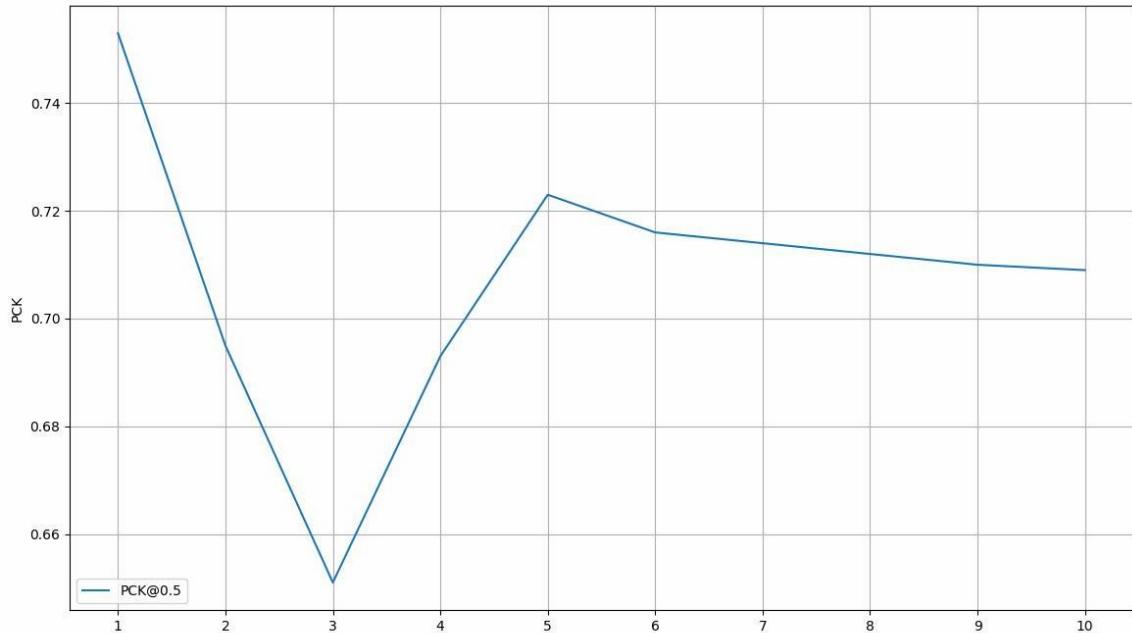
(a) Аналог РСК@0.5



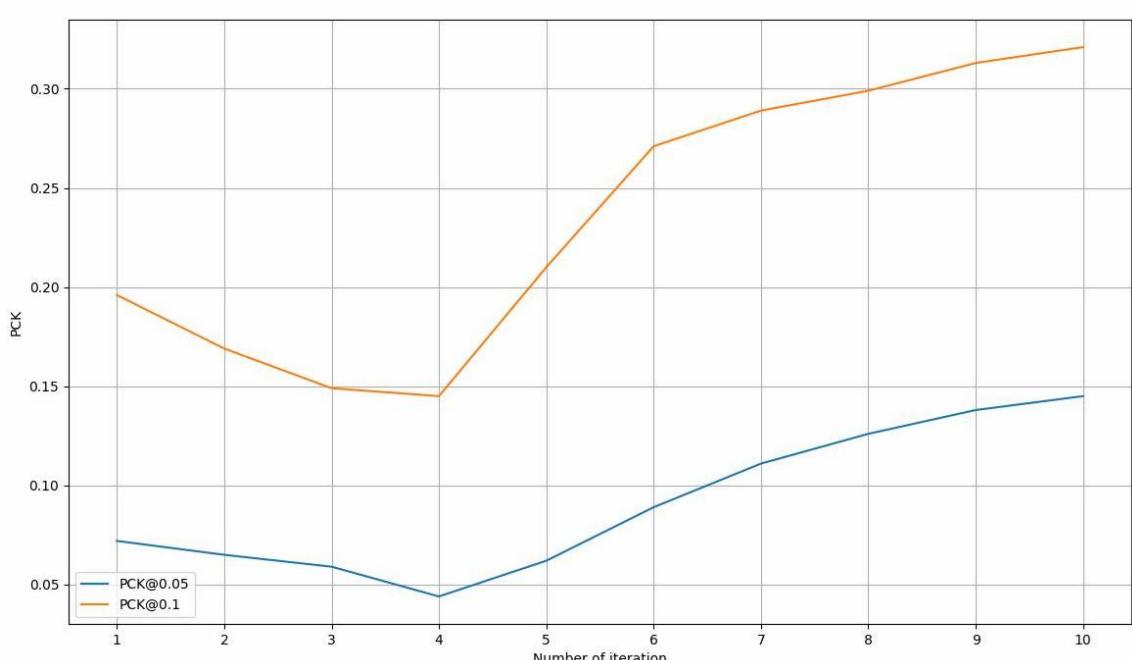
(b) Аналог РСК@0.25

Рис. 29: Распределение корректно предсказанных точек по всей топологии для алгоритма SimCC с магистральной сетью ResNet.

RTMPose



(a) PCK@0.5



(b) PCK@0.05 и PCK@0.1

Рис. 30: Зависимость метрики РСК на целевом домене от номера итерации для модели RTMPose.

	PCK@0.05	PCK@0.25	PCK@0.5
Baseline	0.282	0.882	0.976
Adapted model	0.145	0.652	0.709
Finetune model	0.076	0.709	0.922
	Время обучения	Время предсказания	
Baseline	73 мин	42,4 мс	
Adapted model	7,6 мин	41,7 мс	
Finetune model	22 мин	41,8 мс	

Таблица 4: Сравнительная статистика нескольких состояний для модели RTMPose.

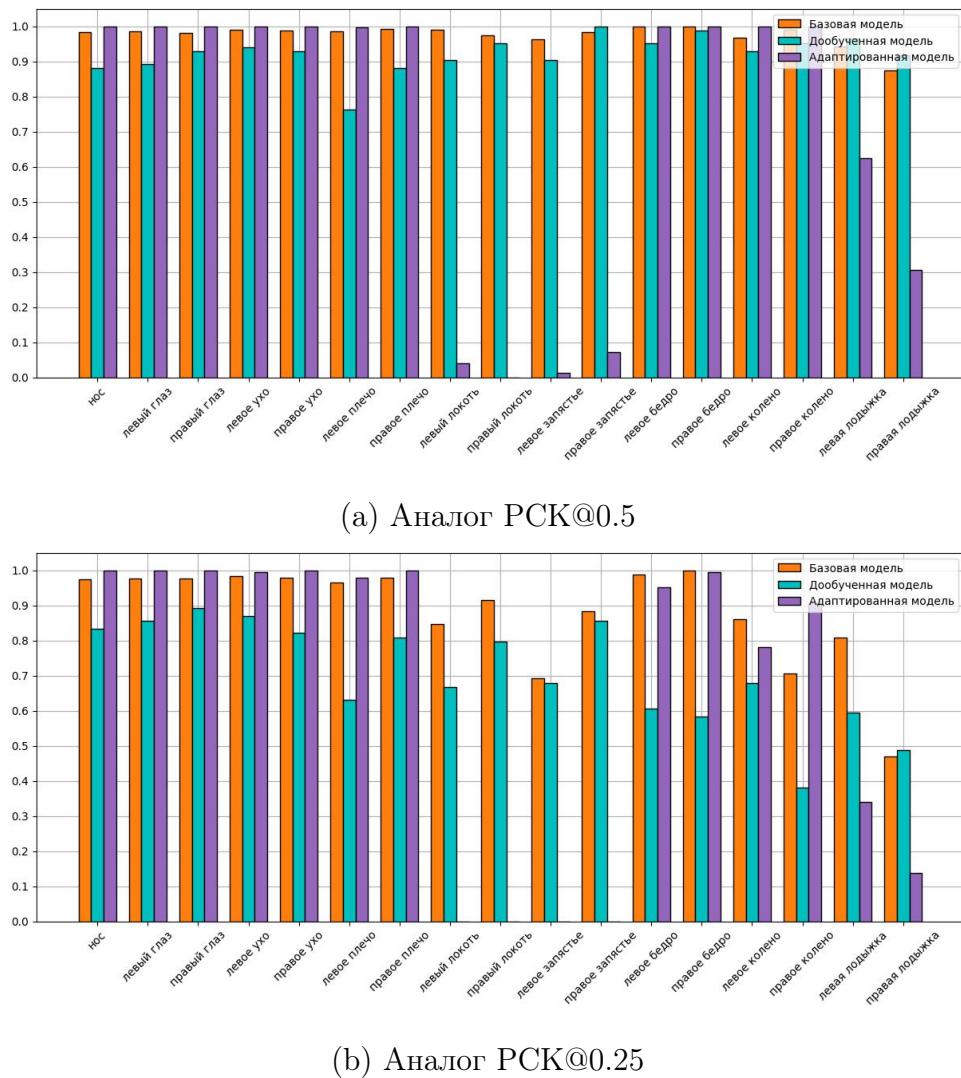


Рис. 31: Распределение корректно предсказанных точек по всей топологии для модели RTMPose.

6 Заключение

Необходимо рассказать о результатах исследования и о том, что мы получили. Привести сравнения, если таковые будут иметь место. Закинуть удочку для будущих исследований.

Список литературы

- [1] *FIFA*. Semi-automated offside technology to be used at FIFA World Cup 2022. — <https://www.fifa.com/fifaplus/en/articles/semi-automated-offside-technology-to-be-used-at-fifa-world-cup-2022> — 2022.
- [2] Player pose analysis in tennis video based on pose estimation / Ryunosuke Kurose, Masaki Hayashi, Takeo Ishii, Yoshimitsu Aoki // 2018 International Workshop on Advanced Image Technology (IWAIT). — 2018. — Pp. 1–4.
- [3] *Thorpe, James*. Pose estimation: utilising AI to improve tennis technique. — <https://sportretina.com/blog/pose-estimation-utilising-ai-to-improve-tennis-technique/>. — 2023.
- [4] *google.github.io*. MediaPipe.Home. — <https://google.github.io/mediapipe/>.
- [5] *Kreiss, Sven*. OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association / Sven Kreiss, Lorenzo Bertoni, Alexandre Alahi // *IEEE Transactions on Intelligent Transportation Systems*. — 2022. — Vol. 23, no. 8. — Pp. 13498–13511.
- [6] Human Pose Estimation Using Body Parts Dependent Joint Regressors / Matthias Dantone, Juergen Gall, Christian Leistner, Luc Van Gool // 2013 IEEE Conference on Computer Vision and Pattern Recognition. — 2013. — Pp. 3041–3048.
- [7] *Toshev, Alexander*. DeepPose: Human Pose Estimation via Deep Neural Networks / Alexander Toshev, Christian Szegedy // 2014 IEEE

- Conference on Computer Vision and Pattern Recognition. — 2014. — Pp. 1653–1660.
- [8] RMPE: Regional Multi-person Pose Estimation / Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, Cewu Lu // ICCV. — 2017.
- [9] Lin, Tsung-Yi. Microsoft COCO: Common Objects in Context. — 2014. <https://arxiv.org/abs/1405.0312>.
- [10] Tsung-Yi Lin Matteo Ruggero Ronchi, Alexander Kirillov. COCO 2020 Keypoint Detection Task. — <https://cocodataset.org/#keypoints-2020>.
- [11] OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields / Z. Cao, G. Hidalgo Martinez, T. Simon et al. // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2019.
- [12] Deep High-Resolution Representation Learning for Human Pose Estimation / Ke Sun, Bin Xiao, Dong Liu, Jingdong Wang // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2019.
- [13] Bazarevsky, Valentin. BlazePose: On-device Real-time Body Pose tracking. — 2020. <https://arxiv.org/abs/2006.10204>.
- [14] ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation / Yufei Xu, Jing Zhang, Qiming Zhang, Dacheng Tao // Advances in Neural Information Processing Systems. — 2022.
- [15] Dosovitskiy, Alexey. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. — 2021.
- [16] Li, Yanjie. SimCC: a Simple Coordinate Classification Perspective for Human Pose Estimation. — 2021.

- [17] Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss / Debapriya Maji, Soyeb Nagori, Manu Mathew, Deepak Poddar // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2022. — Pp. 2637–2646.
- [18] *Reis, Dillon.* Real-Time Flying Object Detection with YOLOv8. — 2024.
- [19] Yolox: Exceeding yolo series in 2021 / Zheng Ge, Songtao Liu, Feng Wang et al. // arXiv preprint arXiv:2107.08430. — 2021.
- [20] *Jiang, Tao.* RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose. — 2023. <https://arxiv.org/abs/2303.07399>.
- [21] *Contributors, MMPose.* OpenMMLab Pose Estimation Toolbox and Benchmark. — <https://github.com/open-mmlab/mmpose>. — 2020.
- [22] *Lyu, Chengqi.* RTMDet: An Empirical Study of Designing Real-Time Object Detectors. — 2022.
- [23] *Wu, Qiangqiang.* Progressive Unsupervised Learning for Visual Object Tracking / Qiangqiang Wu, Jia Wan, Antoni B. Chan // 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). — 2021. — Pp. 2992–3001.
- [24] *Fan, Hehe.* Unsupervised Person Re-identification: Clustering and Fine-tuning. — 2017.
- [25] Regressive Domain Adaptation for Unsupervised Keypoint Detection / Junguang Jiang, Yifei Ji, Ximeい Wang et al. // CVPR. — 2021.
- [26] A Unified Framework for Domain Adaptive Pose Estimation / Donghyun Kim, Kaihong Wang, Kate Saenko et al. // The European Conference on Computer Vision (ECCV). — 2022.

- [27] *Huang, Xun.* Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization / Xun Huang, Serge Belongie // ICCV. — 2017.
- [28] *Tokarew, Andrey.* Полуавтоматическая система разметки ключевых точек. — <https://github.com/andrwtokar/pose-markup>.
- [29] Domain Adaptation: Challenges, Methods, Datasets, and Applications / Peeyush Singhal, Rahee Walambe, Sheela Ramanna, Ketan Kotecha // IEEE Access. — 2023. — Vol. 11. — Pp. 6973–7020.