Интерпретация действий животного по его изображению во времени, близком к реальному

А.Д. Егоров [1], М.С. Резник [1]
[1] ФГАОУ ВО «Национальный исследовательский ядерный университет «МИФИ», 115409, Россия, г. Москва, Каширское шоссе, д. 31

Аннотация

Определение действий объекта – сложная и актуальная задача компьютерного зрения. Такую задачу можно решать с помощью информации о положении ключевых точек объекта. Обучение моделей, определяющих положение ключевых точек, требует большой объём данных, включающих в себя информацию о положении этих ключевых точек. В связи с недостатком данных для обучения представлен метод для получения дополнительных данных, а также алгоритм, позволяющий получать высокую точность распознавания действий животных на основании малого числа данных. Достигнутая точность определения положений ключевых точек на тестовой выборке составила 92,3 %. По положению ключевых точек определяется действие объекта. Сравниваются различные подходы к классификации действий по ключевым точкам. Точность определения действий объекта на изображении достигает 73,5 %.

Ключевые слова: компьютерное зрение, обнаружение животных, классификация действий, нейронная сеть, машинное обучение, опорные модели, классификация скелета, аугментация данных, Keypoint R-CNN, Mobile Net.

Цитирование: **Егоров, А.Д.** Интерпретация действий животного по его изображению во времени, близком к реальному / А.Д. Егоров, М.С. Резник // Компьютерная оптика. – 2023. – Т. 47, № 2. – С. 278-286. – DOI: 10.18287/2412-6179-CO-1138.

Citation: Egorov AD, Reznik MS. Near real-time animal action recognition and classification. Computer Optics 2023; 47(2): 278-286. DOI: 10.18287/2412-6179-CO-1138.

Введение

Компьютерное зрение – это востребованное научное направление в области искусственного интеллекта. Основная задача компьютерного зрения – анализ изображения или видеопотока (по сути представляющего из себя набор сменяющихся изображений), на котором требуется обнаружить интересующий объект и его свойства.

Одна из актуальных задач компьютерного зрения – задача распознавания действий объекта на основании его изображения или серии изображений [1 – 3]. Проводить распознавание действий при решении такой задачи можно как у механических [4], так и у живых объектов (люди [5, 6], животные [7], насекомые [3]). Решение рассматриваемой задачи необходимо для обеспечения возможности автоматизированного наблюдения за объектом и принятия необходимых решений на основании полученной информации [8].

Решение задачи определения действия объекта по видеопотоку или изображению можно разбить на подзадачи [9]: обнаружение объекта на изображении, определение положения его ключевых точек, классификация действия по положению ключевых точек. При этом для решения каждой из подзадач с использованием методов и моделей машинного обучения необходимы наборы данных [9, 10]. Для решения задачи распознавания действий человека создан набор данных COCO [11], для решения задачи распознавания действий животных на текущий момент эквивалентного по качеству набора данных в открытом доступе нет.

В данной работе предлагается решение задачи нахождения определенных животных по видеопотоку и интерпретации их действий, рассматриваются различные подходы организации набора данных и классификации действий по ключевым точкам.

1. Анализ проблемы

В качестве объекта рассматриваются животные. Задача определения действий объекта может быть решена с помощью различных подходов [12]. В этом параграфе рассмотрены решения в области определения действий объекта по его изображению, а также существующие современные наборы данных.

1.1. Анализ существующих работ

Задача определения действий животного на основе его изображения или видеопотока не является новой, однако она в первую очередь решалась в контексте распознавания действий животного, запечатлённого фото- или видеоловушкой в дикой природе [13]. При этом решения задачи могут быть разделены на два типа: без использования ключевых точек скелета найденного объекта и с использованием ключевых точек скелета найденного объекта. Далее будут рассмотрены примеры работ обоих типов.

Одной из первых работ (относится к первому типу) с использованием нейросетевых моделей на по-

добную тему можно считать [14], в работе использовались свёрточные нейронные сети, точность распознавания животных различных классов составила чуть более 30 %. В работе [15] задача определения действия животного разбита на несколько подзадач: определение факта наличия животного на изображении, локализация животного, идентификация вида, подсчёт количества животных в кадре; определение дополнительных атрибутов. Общая точность (accuracy) идентификации и детектирования животного в кадре в работе [15] составила 90,8 % – 91,9 % в зависимости от используемой модели (ResNet-152 [16] или ансамбль различных моделей). При этом в наборе данных было представлено 48 видов животных. Точность определения действия составила 76,2 %. Объём набора данных, использованного для обучения моделей, составляет 1,5 миллиона изображений. В [17] приводится описание того, как с помощью глубоких нейронных сетей (Faster R-CNN [35], YOLO v2.0) можно детектировать животных на изображениях. Лучшая полученная точность (accuracy) детектирования составила 93,0 % ± 3,2 %. В работе [18] точность (accuracy) классификации животного составила 87,3 % с использованием модели VGG-16. Для всех работ первого типа характерна потребность в большом объёме данных для обучения нейросетевых моделей. При этом решения в вышеперечисленных работах не могут работать в реальном времени, однако позволяют получить модель, которую можно использовать в различных ситуациях практически универсально. В табл. 1 приводится сравнение различных методов распознавания животных и их действий, а также ключевые результаты работы этих алгоритмов.

Табл. 1. Сравнение ключевых алгоритмов и их результатов работы

Работа	Метод	Точность детектирования	Точность распознования
[14]	Мешок визуальных слов или сверточные нейронные сети	33,5 % – 38,3 %	–
[15]	Нейронные сети ResNet или ансамбли различных моделей	90,8 % – 91,9 %	76,2 %
[17]	Faster R-CNN, YOLO	89,8 % – 96,2 %	–
[18]	VGG-16	87,3 %	–
[19]	SSD, сиамская сеть и 3D ResNet-18	92 %	64% – 76 % (только визуальные методы)

Среди работ второго типа рассмотрим в первую очередь [3], в качестве объекта в которой выступает фруктовая дрозофила. В основе предлагаемого метода прогнозирования положения частей тела животных (LEAP estimates animal pose, LEAP) лежит 15-слойная свёрточная нейронная сеть, которая определяет местоположение каждой части тела объекта на изображении. В [21] для определения положения ключевых точек объектов используется предобученная свёрточная нейронная сеть на базе архитектуры ResNet. При этом в работе также используется механизм построения трёхмерных моделей объектов для более качественного определения ключевых точек скелетов. В [22] показано, что при большом количестве параметров моделей невозможно добиться производительности, близкой к работе в реальном времени. В [22] в качестве архитектуры используется модель, состоящая из энкодера и декодера, что в данном случае повышает точность работы. Однако для всех решений в рамках работ второго типа характерна невозможность сравнить точность между собой, так как все представленные решения направлены на работу с разными типами живых объектов и, по существу, решают отдельные частные задачи. Важно, что для использования моделей, обученных в работах второго типа, необходимо учитывать большое количество условий, ограничивающих использование предложенных моделей (особые методы съёмки объектов, определённые ракурсы), то есть методы не универсальны. Однако в силу небольшой вычислительной сложности представленные методы могут работать в режиме, близком к реальному времени.

Таким образом, ключевые факторы, определяющие характеристики используемого решения для задачи распознавания действий животного: универсальность/не универсальность решения; необходимость работы в реальном времени, имеющийся объём данных для обучения моделей.

1.2. Существующие наборы данных для задач распознавания действий

COCO (Common Object in Context) [11] является наиболее полным набором данных, используемым для обучения и тестирования нейронных сетей и других моделей машинного обучения, решающих задачи обнаружения, отслеживания, сегментации и определения позы объектов. Набор данных включает в себя около 330 тысяч изображений (более 200 тысяч размеченных), 1,5 миллиона экземпляров объектов 80 категорий. Для распознавания позы людей COCO предоставляет около 250 тысяч размеченных ключевых точек скелета людей. Однако в представленном наборе данных практически отсутствуют животные.

В [23] для распознавания ключевых точек людей используется модель, обученная на наборе данных COCO. Из всего числа ключевых точек – 35 % не имеют аннотации из-за различных факторов, включая окклюзию, усечение, недостаточно экспонированное изображение, размытый внешний вид и низкое разрешение экземпляров людей. Почти 50 % экземпляров в обучающем наборе данных COCO имеют по меньшей мере шесть необъявленных ключевых точек. Таким образом, набор данных COCO требует доработки и ручной разметки для эффективного использования.

Для работы с животными в качестве объектов можно использовать набор данных Animal Pose [24], содержащий 1000 изображений 5 видов животных (коты, коровы, собаки, лошади, овцы). Для каждого объекта приведена аннотация, содержащая ключевые точки скелета объекта (6 точек на голову, 12 – лапы/ноги, 1 – на хвост, 1 – на конец спины), описанные в виде трёх параметров: координата на изображении по оси X, координата на изображении по оси Y, видимость точки на изображении. Примеры изображений с соответствующими аннотациями из набора данных Animal Pose можно увидеть на рис. 1.

Рис. 1. Примеры изображений и аннотаций набора данных "Animal-pose"

2. Инструменты, необходимые для решения поставленной задачи

2.1. Обнаружение и классификация объекта

Изображения перед подачей моделям приводятся к разрешению с одинаковой шириной и высотой.

Так как на изображении может находиться несколько целевых объектов, в качестве модели для обнаружения и классификации объекта можно использовать свёрточные нейронные сети. Наиболее подходящим для работы в реальном времени [25] вариантом свёрточной нейронной сети для решения такой задачи на текущий момент признана сеть YOLO [26], для которой на текущий момент разработана 6 версия [27].

Изображение перед обработкой сверточной нейронной сетью делится на сетку, затем предсказываются ограничивающие области и вероятности наличия целевого объекта для каждого участка. Преимущества этого подхода заключаются в том, что сеть смотрит на все изображение сразу и учитывает контекст при обнаружении и распознавании объекта.

Конкурентами YOLO выступают модели Faster R-CNN [28] и SSD [29], однако в [30 – 32] показано, что YOLO наиболее точна в определении типов объектов и позволяет вести работу во времени, близком к реальному.

2.2. Определение положения ключевых точек объектов

После обнаружения объектов на изображении требуется определить положение их ключевых точек. Для поставленной задачи хорошо подходит Keypoint R-CNN [33] – двухэтапная [34] нейронная сеть, обнаруживающая ключевые точки объекта на изображении, модификация сети Mask R-CNN [35]. Mask R-CNN, в свою очередь, расширяет Faster R-CNN [35] путем добавления ветви для прогнозирования масок каждой области интереса (англ. region of interest, RoI), которые находит опорная модель (англ. backbone), схематично представленная на рис. 2. Предсказание масок происходит параллельно с классификацией и регрессией ограничивающего объект прямоугольника. Ветвь маски в Mask R-CNN – это небольшая свёрточная сеть, применяемая к каждому RoI и предсказывающая маску. Особенностью работы сети является алгоритм RoIAlign [35], который позволяет сохранять пространственное расположение признаков внутри области.

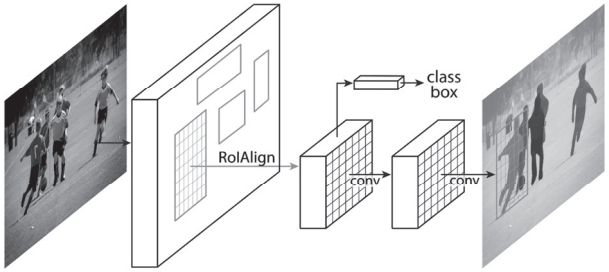

Рис. 2. Архитектура Keypoint R-CNN

Обучение Keypoint R-CNN проходит в совокупности с опорной моделью. При этом функция потерь, состоящая из трёх компонент (L_{cls} – ошибка определения класса, L_{box} – ошибка определения точного положения объекта, L_{mask} – ошибка определения ключевых точек), влияет на параметры опорной модели, схематично представленной на рис. 3.

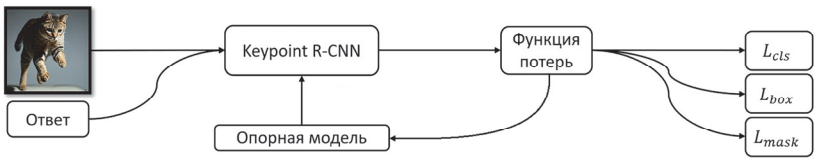

Рис. 3. Схема обучения Keypoint R-CNN

$$L_{cls}\left(y,\hat{y}\right)=-\sum_{i=1}^{k}y_i\log\hat{y}_i,$$

где y – предсказанный класс, $\hat{y}$ – истинный класс;

$$L_{box}\left(t^u,v\right)=\sum_{i\in x,y,w,h}smooth\left(t_i^u-v_i\right),$$

где t^u – предсказанные параметры bbox u-го RoI, v – истинные параметры bbox;

$$smooth\left(x\right)=\begin{cases}0{,}5x^2,\left|x\right|<1\\\left|x\right|-0{,}5,\left|x\right|\geq 1\end{cases},-сглаживание;$$

$$L_{mask}\left(y^k,\widehat{y^k}\right)=\frac{-1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\sigma\left(y_{ij}^k\right)log\,\widehat{y_{ij}^k}+$$

$$+\left(1-\sigma\left(y_{ij}^k\right)\right)\log(1-\widehat{y_{ij}^k})),$$

где m – размер стороны изображения, y – предсказанное значение, $\hat{y}$ – истинное значение для k-го класса.

2.3. Опорная модель

Для того чтобы архитектура Keypoint R-CNN определила местоположение ключевых точек, опорная модель должна найти области интереса, в которых могут находиться ключевые точки. Использование предобученных опорных моделей позволяет снизить количество необходимых для обучения данных. В качестве вариантов для опорной модели были рассмотрены семейство моделей архитектуры ResNet [36] и MobileNet [37]. В работе [38] приводится детальный анализ параметров опорных моделей в контексте задачи детектирования животных на изображении. Результаты работы сетей рассматриваемых семейств представлены на рис. 4. По оси x на рис. 4 отложена точность (mAP, формула приведена в [38]) работы моделей, по оси y приводится время обработки моделью одного кадра. С учётом полученных результатов принимается решение о целесообразности использования модели MobileNet v3 Large в качестве опорной в связи с сопоставимой с моделями семейства ResNet точностью, но большей производительностью.

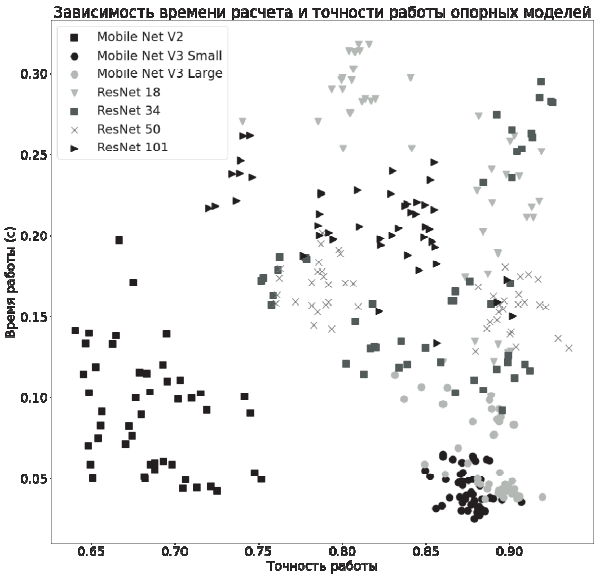

Рис. 4. Сравнение качества обнаружения ключевых точек объектов с использованием опорных моделей различных семейств

2.4. Этап классификации действий

После определения положения ключевых точек объекта происходит классификация его действия по позе. Для этой задачи подходят разные классические методы: случайный лес [39] и бустинговые алгоритмы [40 – 42]. Важным преимуществом алгоритма CatBoost [42] относительно других бустиновых алгоритмов является возможность обработки категориальных, а не только числовых признаков. В работе [43], хотя и с акцентом на «большие» данные, показано, что CatBoost оказывается более эффективен, чем аналоги.

Для обучения классификатора CatBoost используют два вида функций потерь в зависимости от количества классов, на которые нужно провести классификацию:

- для бинарной классификации используется функция потерь Logloss [44];
- для многоклассовой классификации используется функция потерь MultiClass [45].

3. Описание разработанного алгоритма нахождения объектов в видеопотоке

Для нахождения объекта в видеопотоке и классификации его действий был разработан следующий алгоритм:

1. Модель YOLO обнаруживает, определяет границы и классифицирует объекты на поступившем на вход изображении.
2. Обученная модель Keypoint R-CNN с подобранной опорной моделью определяет позицию ключевых точек каждого из найденных объектов.
3. Бустинговый алгоритм проводит классификацию действий объекта.

Модуль Keypoint R-CNN может самостоятельно находить объекты на изображении и классифицировать их. YOLO используется для того, чтобы обработать изображение отдельного животного в кадре и не рассматривать животных, которых не было в используемом наборе данных [24]. Сравнение результатов работы детектирования и классификации объектов с использованием модели YOLO и без использования YOLO приводится в подпараграфе 4.2.

Обучение моделей требует предварительной подготовки набора данных, а также подбора гиперпараметров.

Рассмотренные модели обучались в веб-сервисе Google Colaboratory на языке Python. Программа написана на языке Python версии 3.8. При обучении и в программе использовались следующие библиотеки: PyTorch, OpenCV, NumPy, CatBoost, Pandas.

В качестве набора данных для обучения принято решение использовать набор данных Animal pose [24].

3.1. Подготовка данных для обучения

Из исходного набора данных были удалены дубликаты (осталось 967 изображений). Далее картинки вручную были распределены по видам деятельности объектов (животных). Первоначальное распределение по классам действий (см. рис. 5): 411 объектов на изображении стоят, 80 – сидят, 55 – лежат, 95 – идут, 67 – бегут, 15 – прыгают, 45 – с чем-то взаимодействуют, 29 – спят и 36 – едят. Остальные объекты принадлежат классу "nothing" (недостаточно информации для определения действия). Таких объектов 134.

Для получения более равномерного распределения была произведена переклассификация данных. Типы действий в новых распределениях животных представлены в табл. 2.

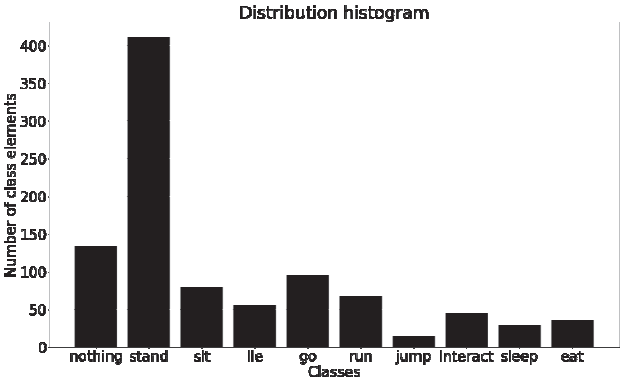

Рис. 5. Первоначальное распределение по классам действий

Табл. 2. Распределение животных по типам действий

	Распределение без nothing (рис. 6)	Распределение с nothing (рис. 7)
Стоит	550	543
Сидит	102	79
Лежит	103	105
Идёт	129	90
Бежит	83	76
Nothing	–	74

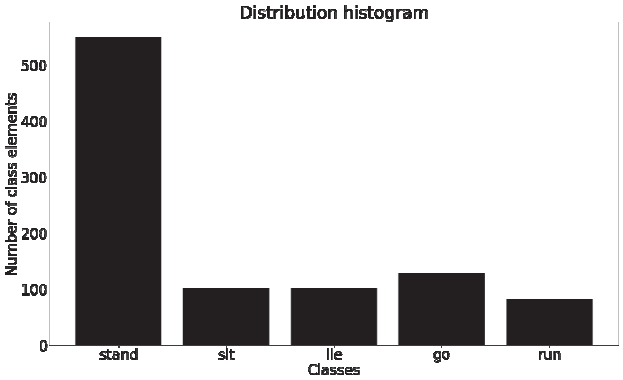

Рис. 6. Распределение по классам действий без "nothing"

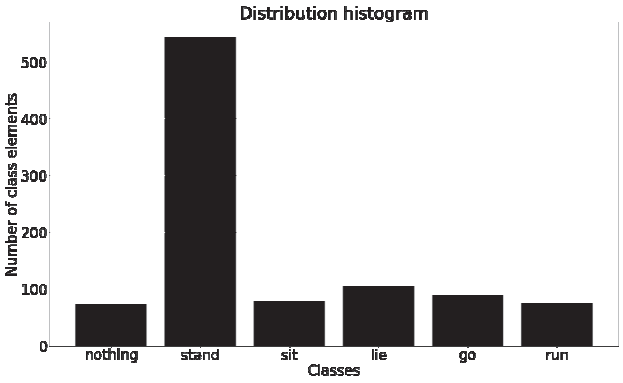

Рис. 7. Распределение по классам действий с добавлением класса "nothing"

3.2. Создание дополнительных данных

Чтобы сделать равномерный по видам деятельности набор данных, необходимо дополнить классы с малым количеством объектов новыми данными. Чтобы увеличить количество доступных данных, предложен специальный метод аугментации данных, который основан на случайном изменении положения точки на изображении. Каждая точка изображения сдвигается относительно исходной позиции на некоторую величину так, чтобы новые координаты не превышали размер изображения.

Предположим, что рассматривается изображение I размера $I_w = w$ пикселей по ширине и размера $I_h = h$ пикселей по высоте. На изображении отмечено n точек, у каждой из которых есть фиксированная позиция (x_i, y_i). Каждая точка обозначает опорную точку скелета животного. Тогда на основании этих данных можно создать новый набор точек, который смещается относительно исходного на значение, определяемое произведением малого коэффициента δ на ширину и высоту изображения. То есть $\forall i \in [1, n] : \left(x_i^{new} = x_i \pm \delta \cdot w, y_i^{new} = y_i \pm \delta \cdot h\right)$. Таким образом можно значительно расширить набор исходных данных, что позволит увеличить тестовую выборку. На рис. 8 представлен пример сетки для аугментации данных. Красным отмечена исходная точка, синей линией – условное направление соединения с другими точками скелета.

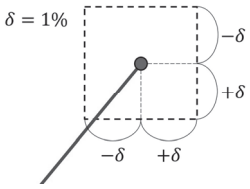

Рис. 8. Сетка для аугментации данных на примере крайней точки скелета

3.3. Обработка видеопотока

Видеопоток обрабатывается в цикле покадрово. Для нахождения всех исследуемых животных в кадре используется модель YOLO V6. Изображения с найденными животными вырезаются по области чуть большей полученной ограничивающей области и подаются на вход Keypoint R-CNN отдельно (см. рис. 9), как это сделано в [21]. Было замечено, что, если подавать всю ограничивающую область, выбираются области интереса, которые почти всегда меньше всей картинки, из-за чего некоторые части тела не попадают в область. Поэтому было решено производить паддинг (создать пустые поля вокруг изображения, заполненные пикселями белого цвета) найденной области перед подачей.

Рис. 9. Раздельное обнаружение ключевых точек

3.4. Обучение моделей

3.4.1. Keypoint R-CNN

Модель Keypoint R-CNN обучалась на рассмотренном выше наборе данных для обнаружения клю-

чевых точек животных на изображениях. В качестве опорной модели были выбраны предобученные MobileNet различных версий (V2, V3 small, V3 large).

Для якорной генерации (создание ограничивающих областей, в которых могут находиться искомые объекты), используемой в модели, были взяты следующие размеры: 16, 32, 64, 128 и 256, и соотношения сторон: 0,5, 1,0 и 2,0. Во время обучения был использован метод оптимизации градиентного спуска Adam [46] со скоростью обучения 0,001.

Обучающая выборка включала в себя 800 экземпляров, а тестовая – 167. Изображения приводились к размеру 256×256, для этого пересчитывались координаты ключевых точек и параметры ограничивающих областей. Обучение модели состояло из 100 эпох, и данные подавались партиями по 6 изображений. Первичные результаты работы можно видеть на рис. 10.

Рис. 10. Первичные результаты работы Keypoint R-CNN

3.4.2. Алгоритмы бустинга

Модель машинного обучения на основании бустингового алгоритма для классификации действия по параметрам ключевых точек также обучалась на рассмотренном выше наборе данных. В качестве параметров обучения были выбраны значения по умолчанию. Итоговый результат работы алгоритма представлен на рис. 11.

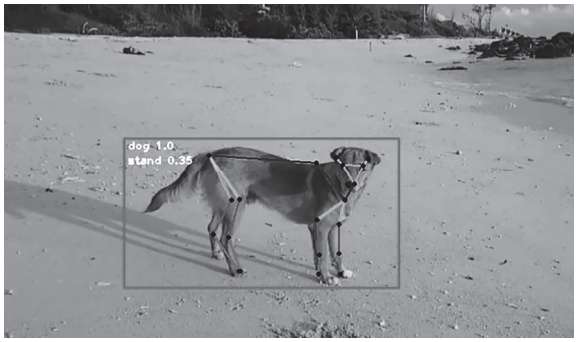

Рис. 11. Пример работы алгоритма с одноэтапной классификацией и использованием YOLO для детектирования и классификации объектов

В рамках экспериментальных исследований были рассмотрены 2 вида классификации (описаны на примере CatBoost Classifier):

- классификация в один этап происходит по всем классам выборки. Был обучен CatBoost Classifier, выполняющий такую операцию. Функция потерь, используемая при обучении, – MultiClass;
- классификация в два этапа. Сначала определяется, принадлежит ли объект классу “stand” (животное стоит), затем определяется, к какому из оставшихся классов принадлежит объект. Были обучены две модели, выполняющие эти операции независимо. Для первой модели была выбрана функция потерь Logloss, а для второй – MultiClass.

Классификация в два этапа предложена как один из вариантов классификации в связи с тем, что используемый набор данных сильно разбалансирован по типам действий животных: стоящих животных больше, чем любых других животных.

4. Результаты

4.1. Обнаружение ключевых точек

В рамках работы были протестированы разные вариации MobileNet и ResNet, используемые в качестве опорной модели (результаты тестирования см. в [38]). В качестве метрики точности взято среднее относительное расстояние от предсказанной точки до истинной:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\sqrt{\left(k_{ix}-\hat{k}_{ix}\right)^2+\left(k_{iy}-\hat{k}_{iy}\right)^2}}{m},$$

где n – количество ключевых точек, k_i – предсказанная i-я ключевая точка, $\hat{k_i}$ – истинная i-я ключевая точка, m – длина стороны изображения.

4.2. Классификация действий

Средняя точность (accuracy) классификации в случае обучения с использованием данных с первоначального распределения по всем классам действий достигает 66,3 %. Такой результат обусловлен наличием классов с очень маленьким количеством объектов.

На новом распределении (рис. 6) бустинговые алгоритмы [40 – 42] были сравнены по точности классификации при различных видах классификации: разбиение на все классы, определение принадлежности к одному классу (самому многочисленному в выборке) и определение принадлежности к оставшимся классам (всем, кроме самого многочисленного); и при различных условиях обучения: учитывая / не учитывая класс животного, с созданием / без создания дополнительных данных (см. табл. 3). Определение принадлежности к классу “stand” и к остальным классам (без “stand”) проводилось без создания дополнительных данных, так как в этих случаях выборка равномерно разбита на классы. В табл. 4 можно видеть результаты сравнения для двухэтапной и одноэтапной классификации.

Классификатор CatBoost [42] показал наилучшие результаты при всех видах классификации и условиях обучения. Отметим, что учёт класса животного и создание дополнительных данных при неравномерном разбиении положительно влияет на точность классификации.

Работа алгоритма была протестирована на исходном наборе данных с использованием YOLO и без

неё. В среднем обработка кадра, на котором заведомо есть обрабатываемое животное, происходит за 0,196 секунды с использованием YOLO и за 0,163 секунды только с Keypoint R-CNN. Учитывая скорость передвижения животных в видеопотоке, было принято решение обрабатывать только 4 кадра в секунду.

Табл. 3. Сравнение классификаторов при различных видах классификации и различных условиях обучения

Точность работы		XGBoost		
		все классы	только класс stand	классы без stand
С учётом класса животного	без аугментации	52,5 %	65,3 %	51,7 %
	с аугментацией	61,3 %	–	–
Без учёта класса животного	без аугментации	50,0 %	60,2 %	49,8 %
	с аугментацией	61,1 %	–	–
		LightGBM		
С учётом класса животного	без аугментации	52,4 %	58,8 %	50,1 %
	с аугментацией	58,4 %	–	–
Без учёта класса животного	без аугментации	47,8 %	56,4 %	47,2 %
	с аугментацией	52,6 %	–	–
		CatBoost		
С учётом класса животного	без аугментации	69,7 %	75,5 %	68,7 %
	с аугментацией	69,7 %	–	–
Без учёта класса животного	без аугментации	68,4 %	73,5 %	67,5 %
	с аугментацией	69,0 %	–	–

Табл. 4. Сравнение распределений при различных видах классификации и различных условиях обучения

Точность		Распределение без nothing	Распределение с nothing
В один этап	без добавления новых данных	69,7 %	66,5 %
	с добавлением новых данных	69,7 %	73,5 %
В два этапа		67,1 %	67,7 %

На тестовых данных, где хотя бы одно исследуемое животное есть в течение всего времени, среднее количество кадров в секунду при такой обработке – 27,4 с использованием YOLO и 33,8 – только с Keypoint R-CNN.

Использование обеих нейронных сетей позволяет достичь точности определения действия 73,5 %, а при использовании только модели Keypoint R-CNN точность снижается до 67,7 %. Можно сделать вывод, что модель YOLO значительно увеличивает качество работы, не сильно снижая производительность.

Тестирования проходили на устройстве с видеокартой 2×NVIDIA Tesla V100 SXM2 32 GB и процессором 2×6252 Intel Xeon 2.1 GHz, 192 GB RAM (DDR4 2933 MHz).

Заключение

В работе исследовались нахождение определенных животных по видеопотоку и интерпретация их действий, были рассмотрены различные подходы организации набора данных и классификации действий по ключевым точкам.

- Для применения в работе был разработан метод аугментации данных для ключевых точек скелета животного на основании их исходного положения.
- Для повышения точности классификации был разработан алгоритм из нескольких ступеней (учитывающий класс животного, тип его деятельности), на основании которого строился одно- или двухступенчатый алгоритм классификации.
- Подготовлен набор данных для обучения моделей из менее, чем 1000 исходных изображений животных с размеченными классами и ключевыми точками, типом действия.
- Показано, что аугментация данных приводит к улучшению точности работы алгоритмов машинного обучения.
- Показано, что учёт класса животного приводит к улучшению точности работы алгоритмов машинного обучения.
- Показано, что классификация наиболее простого действия (животное «стоит») возможна с точностью 75,5 %.
- Наилучший результат – точность классификации действий животного 73,5 %. Для его получения использовалась модель, обученная на аугментированной выборке из малого количества исходных данных.
- Лучший результат получен с помощью обучения нейронной сети для обнаружения ключевых точек Keypoint-RCNN с MobileNet V3 Large в качестве опорной модели и классификатора действия на основе CatBoost.

В работе показано, что использование малого количества данных при правильном подходе к их аугментации позволяет добиться большей точности работы моделей машинного обучения, чем в случае отсутствия аугментации. Дальнейший вектор работы должен быть направлен на использование моделей машинного обучения, использующих последовательности кадров для одновременного анализа типа действий.

References

[1] Zhou J, Lin K-Y, Li H, Zheng W-S. Graph-based high-order relation modeling for long-term action recognition. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2021: 8984-8993.

[2] Wang L, Tong Z, Ji B, Wu G. TDN: Temporal Difference Networks for efficient action recognition. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2021: 1895-1904.

[3] Pereira TD, et al. Fast animal pose estimation using deep neural networks. Nat Methods 2019; 16(1): 117-125.

[4] Yu L, et al. Traffic danger recognition with surveillance cameras without training data. 2018 15th IEEE Int Conf on Advanced Video and Signal Based Surveillance (AVSS) 2018: 378-383.

[5] Shu X, et al. Concurrence-aware long short-term sub-memories for person-person action recognition. Proc IEEE Conf on Computer Vision and Pattern Recognition Workshops 2017: 2176-2183.

[6] Seredin OS, Kopylov AV, Surkov EE. The study of skeleton description reduction in the human fall-detection task. Computer Optics 2020; 44(6): 951-958. DOI: 10.18287/2412-6179-CO-753.

[7] Graving JM, Chae D, Naik H, Li L, Koger B, Costelloe BR, Couzin ID. DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. eLife 2019; 8: e47994.
[8] Shinde S, Kothari A, Gupta V. YOLO based human action recognition and localization. Procedia Comput Sci 2018; 133: 831-838.
[9] Lalitha B, Gomathi V. Review based on image understanding approaches. 2019 IEEE Int Conf on Electrical, Computer and Communication Technologies (ICECCT) 2019: 1-8.
[10] Josyula R, Ostadabbas S. A review on human pose estimation. arXiv Preprint. 2021. Source: ⟨https://arxiv.org/abs/2110.06877⟩.
[11] Lin T-Y, et al. Microsoft COCO: Common objects in context. In Book: Fleet D, Pajdla T, Schiele B, Tuytelaars T, eds. Computer Vision -- ECCV 2014. Part V. Cham: Springer; 2014: 740-755.
[12] Tuia D, Kellenberger B, Beery S, et al. Perspectives in machine learning for wildlife conservation. Nat Commun 2022; 13: 792.
[13] Li W, Swetha S, Shah M. Wildlife action recognition using deep learning. Source: ⟨https://www.crcv.ucf.edu/wp-content/uploads/2018/11/Weining_L_Report.pdf⟩.
[14] Chen G, Han TX, He Z, Kays R, Forrester T. Deep convolutional neural network based species recognition for wild animal monitoring. 2014 IEEE Int Conf on Image Processing (ICIP) 2014: 858-862.
[15] Norouzzadeh MS, et al. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. PNAS 2018; 115(25): E5716-E5725.
[16] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016: 770-778.
[17] Schneider S, Taylor GW, Kremer S. Deep learning object detection methods for ecological camera trap data. 2018 15th Conf on Computer and Robot Vision (CRV) 2018: 321-328.
[18] Bain M, Nagrani A, Schofield D, Berdugo S, Bessa J, Owen J, Hockings KJ, Matsuzawa T, Hayashi M, Biro D, Carvalho S, Zisserman A. Automated audiovisual behavior recognition in wild primates. Sci Adv 2021; 7(46): eabi4883.
[19] Schindler F, Steinhage V. Identification of animals and recognition of their actions in wildlife videos using deep learning techniques. Ecol Inform 2021; 61: 101215.
[20] Nath T, Mathis A, Chen AC, Patel A, Bethge M, Mathis MW. Using DeepLabCut for 3D markerless pose estimation across species and behaviors. Nat Protoc 2019; 14(7): 2152-2176.
[21] Zhang J, Chen Z, Tao D. Towards high performance human keypoint detection. Int J Comput Vis 2021; 129(9): 2639-2662.
[22] Cao J, et al. Cross-domain adaptation for animal pose estimation. Proc IEEE/CVF Int Conf on Computer Vision 2019: 9498-9507.
[23] Dewi C, et al. Yolo V4 for advanced traffic sign recognition with synthetic training data generated by various GAN. IEEE Access 2021; 9: 97228-97242.
[24] Redmon J, et al. You only look once: Unified, real-time object detection. Proc IEEE Conf on Computer Vision and Pattern Recognition 2016: 779-788.
[25] meituan/YOLOv6. Source: ⟨https://github.com/meituan/YOLOv6⟩.
[26] Ren S, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 2015; 28: 91-99.
[27] Liu W, et al. SSD: Single shot multibox detector. In Book: Leibe B, Matas J, Sebe N, Welling M, eds. Computer Vision – ECCV 2016. Cham: Springer; 2016: 21-37.
[28] Kim J-a, Sung J-Y, Park S-h. Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition. 2020 IEEE Int Conf on Consumer Electronics-Asia (ICCE-Asia) 2020: 1-4.
[29] Dr Viraktamath SV, Neelopant A, Navalgi P. Comparison of YOLOv3 and SSD algorithms. Int J Eng Res Technol 2021; 10(02): 193-196.
[30] Sree BB, Bharadwaj VY, Neelima N. An inter-comparative survey on state-of-the-art detectors – R-CNN, YOLO, and SSD. In Book: Reddy ANR, Marla D, Favorskaya MN, Satapathy SC, eds. Intelligent manufacturing and energy sustainability. Singapore: Springer; 2021: 475-483.
[31] Ding X, et al. Local keypoint-based Faster R-CNN. Appl Intell 2020; 50(10): 3007-3022.
[32] Vizilter YV, Gorbatsevich VS, Moiseenko AS. Single-shot face and landmarks detector. Computer Optics 2020; 44(4): 589-595. DOI: 10.18287/2412-6179-CO-674.
[33] He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. arXiv Preprint. 2017. Source: ⟨https://arxiv.org/abs/1703.06870⟩.
[34] Targ S, Almeida D, Lyman K. Resnet in Resnet: Generalizing residual architectures. arXiv Preprint. 2016. Source: ⟨https://arxiv.org/abs/1603.08029⟩.
[35] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv Preprint. 2017. Source: ⟨https://arxiv.org/abs/1704.04861⟩.
[36] Egorov AD, Reznik MS. Selection of hyperparameters and data augmentation method for diverse backbone models mask R-CNN. 2021 IV Int Conf on Control in Technical Systems (CTS) 2021: 249-251.
[37] Breiman L. Random forests. Mach Learn 2001; 45(1): 5-32.
[38] Ke G, et al. LightGBM: A highly efficient gradient boosting decision tree. Adv Neural Inf Process Syst 2017; 30: 3146-3154.
[39] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. Proc 22nd ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining (KDD '16) 2016: 785-794.
[40] Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. CatBoost: unbiased boosting with categorical features. arXiv Preprint. 2017. Source: ⟨https://arxiv.org/abs/1706.09516⟩.
[41] Hancock JT, Khoshgoftaar TM. CatBoost for big data: an interdisciplinary review. J Big Data 2020; 7(1): 94.
[42] Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). Ann Stat 2000; 28(2): 337-407.
[43] Hastie T, et al. Multi-class AdaBoost. Stat Interface 2009; 2(3): 349-360.
[44] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv Preprint. 2014. Source: ⟨https://arxiv.org/abs/1412.6980⟩.

Сведения об авторах

Егоров Алексей Дмитриевич, 1992 года рождения, в 2014 году окончил Национальный исследовательский ядерный университет «МИФИ» по направлению «Прикладная математика и информатика». Работает ассистентом в Институте интеллектуальных кибернетических систем НИЯУ МИФИ. Область научных интересов: обработка графических изображений, компьютерное зрение, машинное обучение. E-mail: *adegorov@mephi.ru* .

Резник Максим Семёнович, 2002 года рождения, с 2020 года обучается в Национальном исследовательском ядерном университете «МИФИ» по направлению «Прикладная математика и информатика». Работает лаборантом и руководителем проектной деятельности в Институте интеллектуальных кибернетических систем НИЯУ МИФИ. Область научных интересов: моделирование, компьютерное зрение, машинное обучение.
E-mail: *maximrez@mail.ru* .

ГРНТИ: 28.23.15
Поступила в редакцию 1 апреля 2022 г. Окончательный вариант – 3 октября 2022 г.

Near real-time animal action recognition and classification

A.D. Egorov [1], M.S. Reznik [1]
[1] National Research Nuclear University MEPhI,
115409, Moscow, Russia, Kashirskoe Shosse, 31

Abstract

In computer vision, identification of actions of an object is considered as a complex and relevant task. When solving the problem, one requires information on the position of key points of the object. Training models that determine the position of key points requires a large amount of data, including information on the position of these key points. Due to the lack of data for training, the paper provides a method for obtaining additional data for training, as well as an algorithm that allows highly accurate recognition of animal actions based on a small number of data. The achieved accuracy of determining the key points positions within a test sample is 92%. Positions of the key points define the action of the object. Various approaches to classifying actions by key points are compared. The accuracy of identifying the action of the object in the image reaches 72.9 %.

Keywords: computer vision, machine learning, animal recognition, action recognition, data augmentation, Keypoint R-CNN, Mobile Net.

Citation: Egorov AD, Reznik MS. Near real-time animal action recognition and classification. Computer Optics 2023; 47(2): 278-286. DOI: 10.18287/2412-6179-CO-1138.

Authors' information

Alexey Dmitrievich Egorov (b. 1992) graduated from National Research Nuclear University MEPhI in 1992, majoring in Applied Mathematics and Informatics. Currently he works as the assistant at the Institute of Cyber Intellectual Systems of National Research Nuclear University. Research interests are computer graphics processing, computer vision, and machine learning. E-mail: *adegorov@mephi.ru* .

Masksim Semenovich Reznik (b. 2002) study at National Research Nuclear University MEPhI. Works as laboratory assistant and as project manager in the Institute of Cyber Intellectual Systems of National Research Nuclear University. Research interests: modeling, computer vision, and machine learning. E-mail: *maximrez@mail.ru* .

Received April 1, 2022. The final version – October 3, 2022.