Столыпинский вестник

Научная статья

Original article

УДК 004.032.26

ДЕТЕКЦИЯ ЖЕСТОВ С ПОМОЩЬЮ YOLO

GESTURE DETECTION WITH YOLO

Скляр Александр Яковлевич, к.т.н, доцент**,** Российский технологический университет МИРЭА, г. Москва**,** E-mail: askliar@mail.ru

Высоцкая Анна Аркадьевна, Магистр 2 курса, Российский технологический университет МИРЭА, г. Москва**,** E-mail: anVys@mail.ru

Горячев Антон Александрович, Магистр 2 курса, Российский технологический университет МИРЭА, г. Москва**,** E-mail: gorjat.anton@mail.ru

Sklyar Alexander Yakovlevich, Ph.D., Associate Professor, Russian Technological University MIREA, Moscow, E-mail: askliar@mail.ru

Vysotskaya Anna Arkadievna, Master 2 courses, Russian Technological University MIREA, Moscow, E-mail: anVys@mail.ru

Goryachev Anton Alexandrovich, Master 2 courses, Russian Technological University MIREA, Moscow, E-mail: gorjat.anton@mail.ru

Аннотация. В данной статье рассматривается архитектура сверточной нейронной сети YOLO для детекции объектов на изображении или видеоряде. Представлена следующая цель для работы: определение нейронной сетью команд в режиме реального времени, которые в дальнейшем используются для взаимодействия с собакой-роботом. В статье представлен пример разметки

изображений для дообучения модели, а также результаты её обучения. На основе этого представлен пример взаимодействия с собакой-роботом UnitreeRobotics A1 с использованием обученной глубокой модели. В результате была получена сверточная нейронная сеть, позволяющая с высокой точностью и скоростью выполнять детекцию жестов на видеоряде.

Annotation. This article discusses the architecture of the YOLO convolutional neural network for detecting objects in an image or video sequence. The following goal for the work is presented: the definition by the neural network of commands in real time, which are further used to interact with the robot dog. The article presents an example of image markup for retraining the model, as well as the results of its training. Based on this, an example of interaction with a UnitreeRobotics A1 robot dog using a trained deep model is presented. As a result, a convolutional neural network was obtained, which makes it possible to perform gesture detection on a video sequence with high accuracy and speed.

Ключевые слова: нейронные сети, сверточные нейронные сети, робот, UnitreeRobotics A1, детекция объектов, YOLO, Transfer Learning.

Keywords: neural networks, convolutional neural networks, robot, UnitreeRobotics A1, object detection, YOLO, Transfer Learning.

Введение

Наука о данных или data science пытается понять, как можно из данных самой разной природы получить ответы на некоторые вопросы. Разделы науки о данных, которые освещают разные методы получения таких ответов, называются машинное обучение и анализ данных. Машинное и глубокое обучение являются вложенными друг в друга дисциплинами внутри искусственного интеллекта.

В настоящее время есть три основных активно развивающихся области исследования глубокого обучения: компьютерное зрение, алгоритмы обработки речи, понимание и генерация текста на естественном языке.

Самое зрелое и довольно давно развивающееся направление – это компьютерное зрение, которое занимается изучением того, как сделать так, чтобы

компьютеры могли «видеть». С помощью компьютерного зрения решаются задачи в различных областях. Например, в здравоохранении алгоритмы глубокого обучения позволяют определить по рентгеновскому снимку легких есть у человека пневмония или нет. Широко применяется компьютерное зрение во многих отраслях цифровой экономики, таких как «Умный город», автономные автомобили, видеонаблюдение и безопасность. Автономный автомобиль должен уметь отличать человека от дерева, а Face ID — владельца телефона от вора. В настоящее время активно развивается сфера робототехники, где так же необходимо компьютерное зрение. Роботы должны уметь детектировать объекты, жесты, движения для взаимодействия с окружающей средой и человеком.

Детекция объектов

Задача детекции объектов — задача, в рамках которой необходимо выделить несколько объектов на изображении с помощью нахождения координат ограничивающих рамок для каждого, заранее неизвестного объекта.

Одним из самых известных алгоритмов обнаружения объектов благодаря своей скорости и точности является YOLO (You Only Look Once).

Алгоритм YOLO был первой попыткой сделать возможной детекцию объектов в реальном времени. В рамках алгоритма YOLO исходное изображение сначала разбивается на сетку из N×N ячеек. Если центр объекта попадает внутрь координат ячейки, то эта ячейка считается ответственной за определение параметров местонахождения объекта. Каждая ячейка описывает несколько вариантов местоположения ограничивающих рамок для одного и того же объекта. Каждый из этих вариантов характеризуется пятью значениями — координатами центра ограничивающей рамки, его шириной и высотой, а также степени уверенности в том, что ограничивающая рамка содержит в себе объект. Также необходимо для каждой пары класса объектов и ячейки определить вероятность того, что ячейка содержит в себе объект этого класса.

На данный момент существует 5 версий YOLO. YOLOv1 была выпущена в качестве исследовательской работы Джозефом Редмоном в 2015 году [1]. YOLOv2 является результатом совместной Джозефа Редмона, оригинального автора YOLO

и Али Фархади. Публикация на эту тему появилась в 2016 году. [2]. Следующей моделью от этих же авторов стала YOLOv3, которая является улучшением архитектуры YOLOv2 [3]. После этого работа над YOLO застопорилась. Следующая версия YOLOv4 была выпущена Алексеем Бочковским, Цзянь-Яо Вангом и Хун-Юанем Марком Ляо [4]. Вскоре после выхода YOLOv4 Гленн Джохер представил YOLOv5 с использованием фреймворка Pytorch.

Среди перечисленных детекторов значительно выделяется YOLOv5, в которой авторы указывают на наилучшее соотношение скорость-качество среди практически всех остальных нейросетевых детекторов.

Нейронная сеть YOLOv5 имеет несколько архитектур (Рисунок 1).

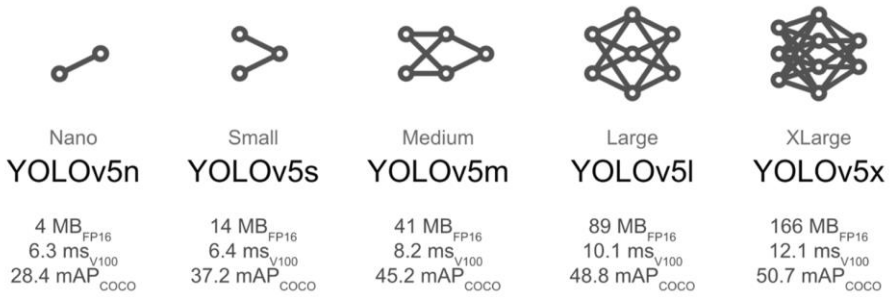

Рисунок 1 — Архитектуры YOLOv5

Самым маленьким и быстрым, следовательно, менее точным вариантом является YOLOv5n. Она содержит всего 1.9 миллионов параметров. Для сравнения, YOLOv5x использует 86.7 миллионов параметров, следовательно, является самой точной реализацией YOLO, но самой медленной.

Реализация модели для детекции кинологических жестов

Основная цель — это получить глубокую модель для того, чтобы взаимодействовать с собакой-роботом UnitreeRobotics A1 (Рисунок 2) [5].

Рисунок 2 – Собака-робот

Модель YOLO изначально обучена на наборе данных Microsoft COCO (Common Objects in Context). Этот набор данных содержит 91 класс и 2.5 миллиона размеченных изображений. YOLOv5n на данном наборе данных достигает точности mAP0.5 = 45.7 и скорости детекции на центральном процессоре (CPU), равную 45 миллисекундам для изображений размера 640 на 640 пикселей [6]. mAP0.5 (mean Average Precision) — это популярная метрика для измерения точности детекторов, которая определяется на интервале от 0 до 1.

Общая архитектура YOLO представлена на Рисунке 3

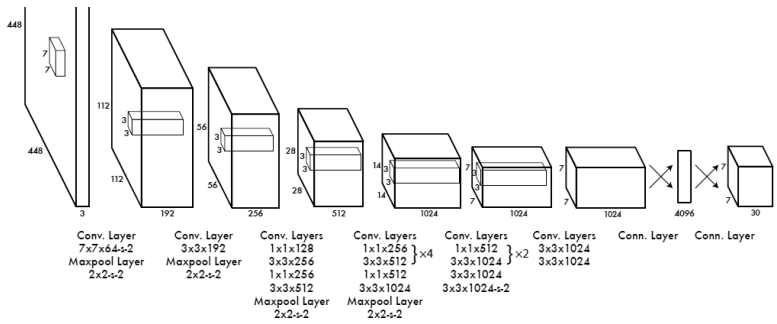

Рисунок 3 — Архитектура YOLO

Для такой цели необходима высокая скорость работы нейросети, именно поэтому используется самая маленькая версия модели — YOLOv5n.

Чтобы использовать данную архитектуру для решения задачи детекции жестов используется подход под названием Transfer Learning (трансферное обучение) [7]. Transfer Learning — это подраздел машинного обучения, целью которого является применение знаний, полученных из одной задачи, к другой целевой задаче. Нейросеть сначала обучается на большом объеме данных, в данном случае COCO, затем — на целевом наборе.

Для дообучения модели был собран набор данных с 3 классами жестов: 0 — COOL, 1 — PISTOL, 2 — SHAKE. Исходные данные — это видео с разрешением 1920*1080 с частотой 30 кадров в секунду с выполнением трех различных движений. Длительность каждого видеоряда около 3-х минут. Далее было произведено разбиение видеоряда на кадры. Удалены лишние изображения, на

которых нет объектов детекции. Далее была произведена разметка изображений. В результате обработки данных были получены:

- папка с изображениями для каждого класса;
- папка с txt файлами, в которых содержится информация о классе и координаты ограничивающей рамки.

Пример разметки изображений представлен на Рисунке 4.

Рисунок 4 — Разметка изображений

В результате в обучающем наборе содержится 5453 изображений, а на валидационном — 1361 изображение.

Далее производилось дообучение модели YOLOv5n. Оптимизатор, который использовался во время обучения — SGD (стохастический градиентный спуск). Количество эпох обучения: 32. Функция активации — SiLU, которая представлена формулой

$$f(x) = x \cdot \frac{1}{1+e^{-x}}, \quad (1)$$

где x — линейная комбинация данных и весов.

Функция потерь состоит из линейной комбинации трех метрик L с балансовыми коэффициентами λ (2).

$$\lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc}, \quad (2)$$

где L_{cls} — ошибка ограничивающего прямоугольника (средний квадрат ошибок);

L_{obj} — уверенность в присутствии объекта (бинарная кросс-энтропия);

L_{loc} — ошибка классификации (кросс-энтропия).

Процесс обучения модели на видеокарте Tesla K80 от NVIDIA с технологией CUDA представлен на Рисунке 5.

```
Epoch   gpu_mem       box       obj       cls    labels  img_size
 0/31     4.67G   0.07973   0.02344   0.02576        15       640: 100% 122/122 [03:58<00:00,  1.96s/it]
           Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 16/16 [00:17<00:00,  1.09s/it]
             all       1361       1361      0.357      0.594      0.449      0.157

Epoch   gpu_mem       box       obj       cls    labels  img_size
 1/31     4.67G   0.05547   0.01191  0.007109        14       640: 100% 122/122 [03:55<00:00,  1.93s/it]
           Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 16/16 [00:16<00:00,  1.05s/it]
             all       1361       1361       0.68       0.88       0.79      0.403

Epoch   gpu_mem       box       obj       cls    labels  img_size
 2/31     4.67G   0.04366  0.009046  0.003883        10       640: 100% 122/122 [03:55<00:00,  1.93s/it]
           Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 16/16 [00:16<00:00,  1.06s/it]
             all       1361       1361      0.795      0.821      0.889      0.389

Epoch   gpu_mem       box       obj       cls    labels  img_size
 3/31     4.67G   0.03557  0.008173  0.002865        12       640: 100% 122/122 [03:55<00:00,  1.93s/it]
           Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 16/16 [00:16<00:00,  1.05s/it]
             all       1361       1361      0.967      0.978      0.989      0.634

Epoch   gpu_mem       box       obj       cls    labels  img_size
 4/31     4.67G   0.03039  0.007363  0.002058        13       640: 100% 122/122 [03:55<00:00,  1.93s/it]
           Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 16/16 [00:16<00:00,  1.05s/it]
             all       1361       1361      0.977      0.977       0.99      0.681

Epoch   gpu_mem       box       obj       cls    labels  img_size
 5/31     4.67G     0.027  0.006811  0.001543        13       640: 100% 122/122 [03:55<00:00,  1.93s/it]
           Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 16/16 [00:16<00:00,  1.06s/it]
             all       1361       1361      0.996      0.995      0.995      0.693

Epoch   gpu_mem       box       obj       cls    labels  img_size
 6/31     4.67G    0.0251  0.006553  0.001413        16       640: 100% 122/122 [03:55<00:00,  1.93s/it]
           Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 16/16 [00:16<00:00,  1.05s/it]
             all       1361       1361       0.98      0.989      0.994      0.717
```

Рисунок 5 — Процесс обучения модели

Технология CUDA — это программно-аппаратная вычислительная архитектура NVIDIA, которая даёт возможность организации доступа к набору инструкций графического ускорителя и управления его памятью при организации параллельных вычислений.

Метрики качества модели на тренировочной и на валидационной выборках представлены на Рисунке 6.

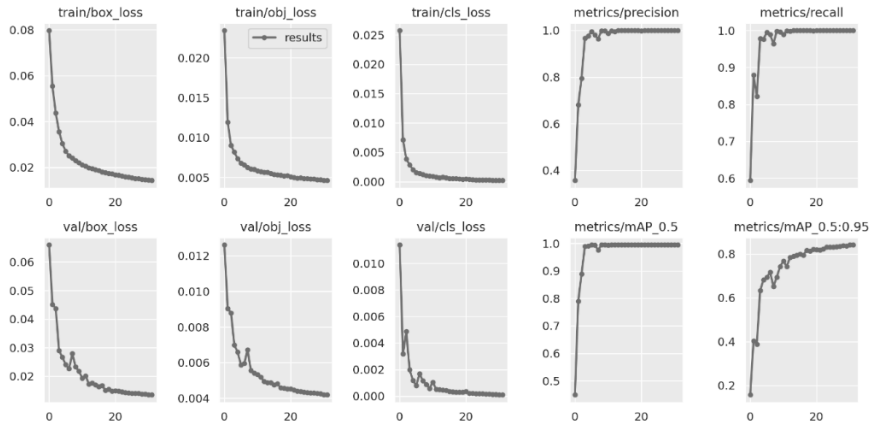

Рисунок 6 — Метрики качества

Результат обучения модели для каждого класса на валидационном наборе изображений представлен на Рисунке 7.

```
Model summary: 213 layers, 1763224 parameters, 0 gradients, 4.2 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 16/16 [00:20<00:00,  1.26s/it]
                 all       1361       1361          1          1      0.995      0.842
                COOL       1361        505          1          1      0.995      0.774
              PISTOL       1361        461          1          1      0.995      0.833
               SHAKE       1361        395          1          1      0.995      0.918
```

Рисунок 7 — Результат обучения

Наилучшую точность имеет класс SHAKE, далее класс PISTOL и класс COOL.

При взаимодействии с собакой-роботом были выбраны следующие движения:

- COOL – собака-робот «отряхивается»;
- PISTOL – собака-робот присаживает на все «лапы»;
- SHAKE – собака-робот присаживается на задние «лапы».

На Рисунках 8-9 представлен результат работы модели при взаимодействии с роботом A1 в режиме реального времени.

Рисунок 8 — Результат 1

Рисунок 9 — Результат 2

Модель имеет точность достаточную для выполнения поставленных задач, не смотря на выбранную архитектуру нейронной сети с наименьшим количеством настраиваемых параметров. Вместе с этим обеспечивается высокая скорость детекции объектов на изображении. Обученная нейронная сеть детектирует классы в среднем со скоростью 12 кадров в секунду на центральном процессоре.

ЗАКЛЮЧЕНИЕ

В результате была реализовала глубокая модель нейронной сети, которая позволяет детектировать жесты в реальном времени для взаимодействия с роботом UnitreeRobotics A1. Рассмотрены основные виды детекторов объектов на изображении, а также метрики, позволяющие оценивать качество модели.

Литература

1. Joseph Redmon. You Only Look Once: Unified, Real-Time Object Detection / Santosh Divvala, Ross Girshick, Ali Farhadi // IEEE Conference on Computer Vision and Pattern Recognition. — 2015 г.
2. Joseph Redmon. YOLO9000: Better, Faster, Stronger / Ali Farhadi // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016 г.
3. Joseph Redmon. YOLOv3: An Incremental Improvement / Ali Farhadi // Computer Science. — 2018 г.
4. Alexey Bochkovskiy. YOLOv4: Optimal Speed and Accuracy of Object Detection / Chien-Yao Wang, Hong-Yuan Mark Liao // Computer Science. — 2020 г.
5. Робот UnitreeRobotics A1 [Электронный ресурс]. — URL: https://www.unitree.com/products/a1/
6. Официальный сайт YOLO [Электронный ресурс]. — URL: https://github.com/ultralytics/yolov5
7. Трансферное обучение [Электронный ресурс]. — URL: https://habr.com/ru/company/binarydistrict/blog/428255/

References

1. Joseph Redmon. You Only Look Once: Unified, Real-Time Object Detection / Santosh Divvala, Ross Girshick, Ali Farhadi // IEEE Conference on Computer Vision and Pattern Recognition. — 2015
2. Joseph Redmon. YOLO9000: Better, Faster, Stronger / Ali Farhadi // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016
3. Joseph Redmon. YOLOv3: An Incremental Improvement / Ali Farhadi // Computer Science. — 2018
4. Alexey Bochkovskiy. YOLOv4: Optimal Speed and Accuracy of Object Detection / Chien-Yao Wang, Hong-Yuan Mark Liao // Computer Science. — 2020
5. UnitreeRobotics A1 robot [Electronic resource]. - URL: https://www.unitree.com/products/a1/
6. Official website of YOLO [Electronic resource]. URL: https://github.com/ultralytics/yolov5

7. Transfer learning [Electronic resource]. - URL: https://habr.com/ru/company/binarydistrict/blog/428255/

Для цитирования: Скляр А. Я., Высоцкая А. А., Горячев А. А. ДЕТЕКЦИЯ ЖЕСТОВ С ПОМОЩЬЮ YOLO// Научный сетевой журнал «Столыпинский вестник» №9/2022.