

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра теоретических основ
компьютерной безопасности и
криптографии

Обнаружение объектов на изображении

АВТОРЕФЕРАТ

дипломной работы

студента 6 курса 631 группы

специальности 10.05.01 Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Раевского Сергея Владимировича

Научный руководитель

зав. кафедрой, д. ф.-м. н., доцент

М. Б. Абросимов

23.01.2021 г.

Заведующий кафедрой

д. ф.-м. н., доцент

М. Б. Абросимов

23.01.2021 г.

Саратов 2021

ВВЕДЕНИЕ

Машинное обучение с каждым днем занимает всё большее место в нашей жизни ввиду огромного спектра его применений. Начиная от анализа образов (например, изображений) и заканчивая самоуправляемыми автомобилями. Всё больше задач перекладывается на самообучаемые машины ввиду увеличения объёма данных, так как достаточно хорошо обученная нейронная сеть способна обрабатывать информацию значительно быстрее человека, но не хуже него в плане качества.

В контексте информационной безопасности машинное обучение может использоваться для обнаружения нежелательной или запрещённой информации, например, обнаружение посторонних лиц на защищаемой территории.

В рамках данной работы рассматривается понятие машинного обучения и искусственной нейронной сети, общая постановка задачи обучения по прецедентам, типология задач обучения по прецедентам, объясняется задача классификации и типология этой задачи, осуществляется формальная постановка задачи классифицирования объектов, даётся понятие компьютерного зрения, задач, решаемых с помощью компьютерного зрения, ставится задача обнаружения объектов на изображении, приводится весь необходимый аппарат для решения этой задачи (свёрточные нейронные сети и вспомогательные функции) и рассматриваются различные модели, в том числе и наиболее оптимальная на данный момент – модель YOLO.

В качестве практической части создан и описан фреймворк (каркас) на языке программирования Python с использованием различных дополнительных технологий, который позволяет использовать API для создания различных моделей машинного обучения, решающих задачу обнаружения объектов на изображении. Также на основе созданного фреймворка реализована модель YOLO, позволяющая осуществлять обнаружение объектов на изображении и видео.

Дипломная работа состоит из введения, 6 разделов, заключения, списка использованных источников и приложения. Общий объем работы – 43 страницы, из них 36 страниц – основное содержание, включая 13 рисунков, список использованных источников из 13 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

1 Первый раздел

Первый раздел дипломной работы посвящён рассмотрению основных понятий, связанных с машинным обучением.

Машинное обучение (англ. machine learning, ML) – класс методов искусственного интеллекта, отличительной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества схожих задач. Для построения таких методов используются средства математической статистики, методов оптимизации, численных методов, теории графов, теории вероятностей, различные техники работы с данными в цифровой форме и др.

Данный раздел содержит четыре подраздела.

В первом из них происходит постановка задачи обучения по прецедентам.

Во втором подразделе даётся определение искусственной нейронной сети и приводятся этапы решения задач с помощью нейронных сетей.

Нейронная сеть (искусственная нейронная сеть, ИНС) – математическая модель, её аппаратная и/или программная реализация, построенная на основе организации и функционирования биологических нейронных сетей (сети нервных клеток живых организмов). Это понятие появилось в результате изучения процессов, протекающих в мозге, и при попытке воссоздать эти процессы.

В третьем подразделе приводится типология задач обучения по прецедентам, которые исходят из случая использования нейронных сетей: обучение с учителем, обучение без учителя, частичное обучение.

В четвёртом подразделе даётся понятие глубокого обучения и выделяются его особенности.

Глубокое обучение (англ. deep learning) – это совокупность широкого семейства методов машинного обучения, основанных на имитации работы мозга живого организма в процессе обработки данных, восприятия образов, принятия решений и прочих.

2 Второй раздел

Второй раздел дипломной работы посвящён задаче классификации и понятиям, связанным с ней.

Классификация – один из разделов машинного обучения, осуществляемый с помощью обучения с учителем, посвященный решению следующей задачи. Имеется множество объектов (ситуаций), разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется обучающей выборкой. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

Данный раздел содержит в себе два подраздела.

В первом из них приводятся типологии задач классификации: по типу входных данных и по типу классов.

Во втором подразделе производится формальная постановка задачи классификации.

3 Третий раздел

Третий раздел дипломной работы содержит в себе определение компьютерного зрения и пример типичных задачи, решаемых с его помощью.

Компьютерное зрение (англ. Computer Vision, CV), также называемое машинным зрением (англ. Machine Vision, MV) и техническим зрением – совокупность теоретических и практических знаний по созданию систем, способных выполнять различные задачи, связанные с обработкой изображений и извлечением из них данных.

4 Четвёртый раздел

Четвёртый раздел дипломной работы посвящён задаче обнаружения объектов на изображении и способе её решения с помощью специально типа нейронных сетей – свёрточные нейронные сети. Приводятся также понятия, связанные со свёрточными нейронными сетями, основное из которых – операция свёртки.

Задача обнаружения объектов на изображении состоит из двух этапов:

1. поиск положения всех возможных объектов на изображении согласно заранее определённым классам объектов;
2. идентификация всех найденных объектов согласно всё тем же классам.

Операция свёртки (англ. convolution) – операция над двумя матрицами A (размером $n_x \times n_y$) и B (размером $m_x \times m_y$), результатом которой является матрица C размера $(n_x - m_x + 1) \times (n_y - m_y + 1)$. Каждый элемент результирующей матрицы вычисляется как скалярное произведение матрицы B и некоторой подматрицы A такого же размера. Подматрица определяется положением элемента в результате. Таким образом,

$$C_{ij} = \sum_{u=0}^{m_x-1} \sum_{v=0}^{m_y-1} A_{i+u, j+v} B_{u,v}.$$

5 Пятый раздел

Пятый раздел дипломной работы содержит в себе описание модели YOLO, её архитектуры и этапов её работы (рисунок 1).

Модель YOLO (от английского You Only Look Once) – передовая модель для распознавания и обнаружения объектов на изображении, которую отличает высокая скорость и оптимальная точность, что позволяет успешно применять данную модель в реальном времени. Исходя из названия, модель смотрит на изображение только один раз, именно за счёт этого и достигается высокая скорость работы.

На данный момент существует 4 официальных версии модели YOLO: YOLO, YOLOv2, YOLOv3 и YOLOv4. Для YOLOv3 и YOLOv4 также существуют дополнительные реализации с приставкой -tiny (YOLOv3-tiny и YOLOv4-tiny), которые менее точны, но ещё более быстрые с точки зрения вычислений. В работе рассматриваются только 4 последние упомянутые версии, поскольку первые две версии модели уже устарели.

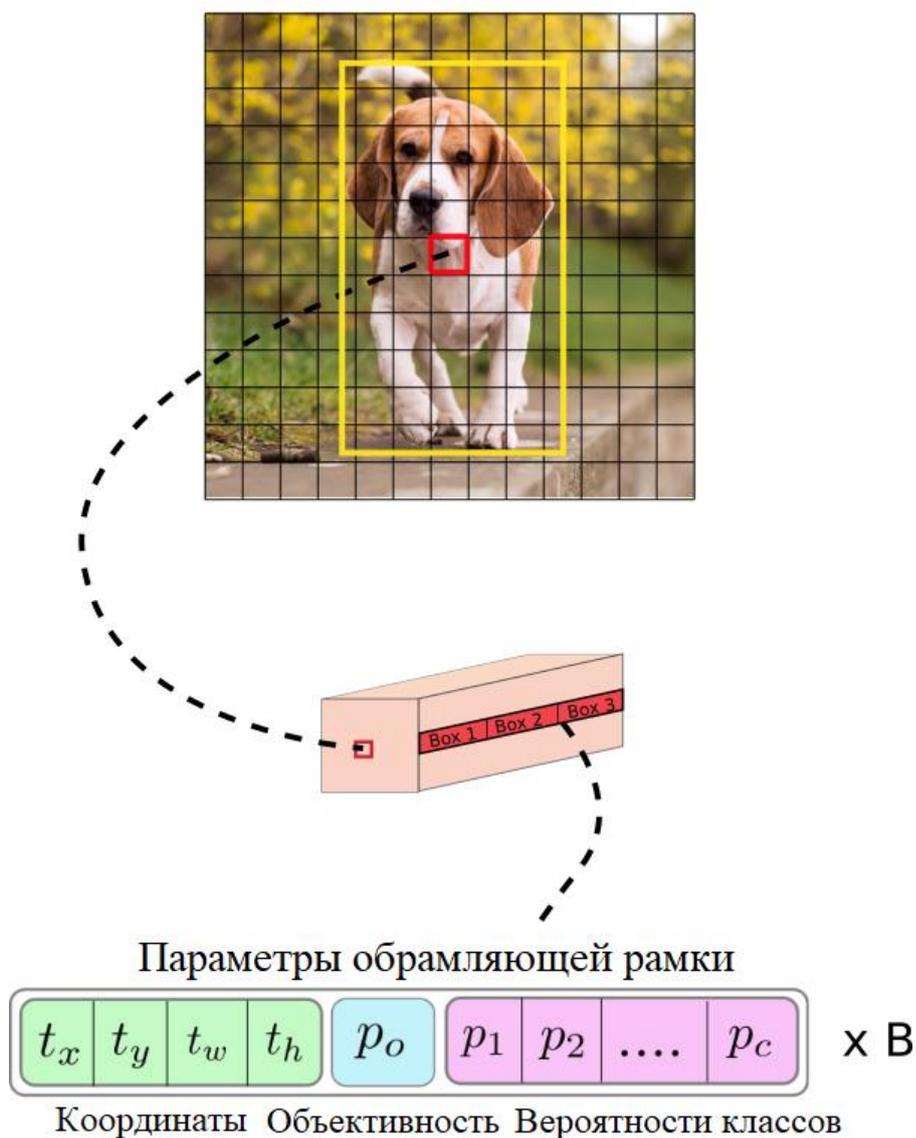


Рисунок 1 – Иллюстрация этапов работы модели YOLO

6 Шестой раздел

Шестой раздел дипломной работы посвящён программной реализации фреймворка (каркаса), позволяющего разрабатывать модели машинного обучения для обнаружения объектов на изображении, и модели YOLO, позволяющей производить обнаружение объектов на изображении и видео на основе фреймворка.

Данный раздел содержит в себе три подраздела.

В первом из них описывается имплементация OpenCV, позволяющая осуществлять обнаружение объектов на изображении с помощью одноимённой библиотеки.

С помощью модуля dnn (deep neural network) и встроенных функций OpenCV легко осуществляется загрузка весов модели и соответствующего конфигурационного файла, после чего на вход подаётся изображение или видеопоток и осуществляется обнаружение объектов.

Во втором подразделе описывается имплементация Tensorflow, позволяющая описывать архитектуру моделей обнаружения объектов на изображении с помощью вспомогательных функций, из которых состоит фреймворк.

В третьем подразделе содержится описание способа запуска программной реализации и результаты её тестирования (рисунок 2 и рисунок 3).

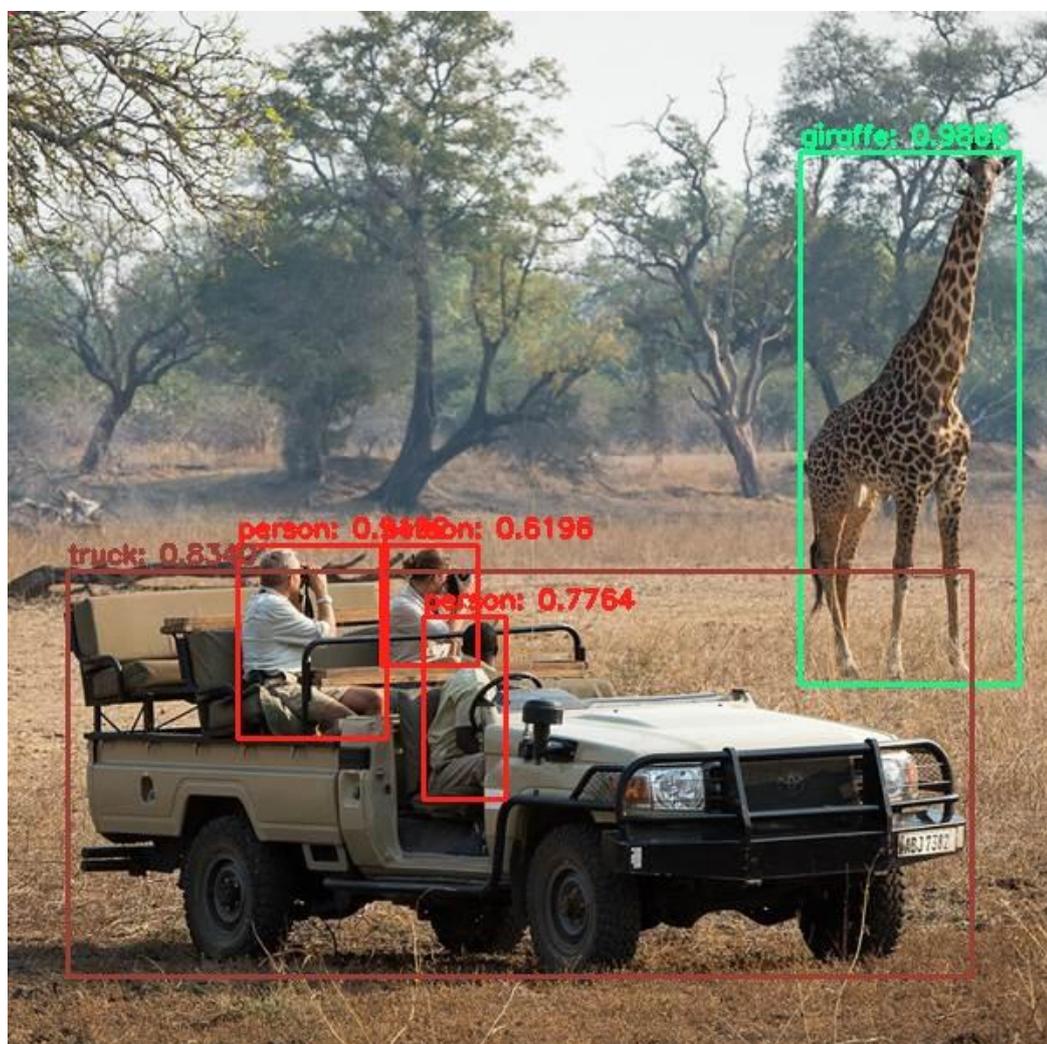


Рисунок 2 – Изображение после осуществлённого обнаружения объектов



Рисунок 3 – Фрагмент видео (скриншот) после осуществления обнаружения объектов

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы были рассмотрены следующие теоретические моменты: понятие машинного обучения и искусственной нейронной сети, общая постановка задачи обучения по прецедентам, типология задач обучения по прецедентам, задача классификации и типология этой задачи, формальная постановка задачи классифицирования объектов, понятие компьютерного зрения, задач, решаемых с помощью компьютерного зрения, задача обнаружения объектов на изображении, весь необходимый аппарат для решения этой задачи (свёрточные нейронные сети и вспомогательные функции) и различные модели, в том числе и наиболее оптимальная на данный момент – модель YOLO.

В качестве практической части был спроектирован и создан фреймворк и программная реализация модели YOLO на его основе, позволяющая производить обнаружение объектов на изображении и видео, на языке Python версии 3.8.6 в среде PyCharm с использованием дополнительных инструментов в виде библиотек CUDA, cuDNN, OpenCV, TensorFlow, NumPy для языка Python.

Проведённое в работе тестирование показывает, что YOLO – оптимальная модель, которая позволяет решать задачу обнаружения объектов на изображении. В дальнейшем, пользуясь созданным фреймворком, можно реализовывать и другие модели, решающую поставленную задачу.