

Обнаружение объектов в реальном времени с помощью библиотеки TensorFlow

Семченко Регина Викторовна

*Приамурский государственный университет имени Шолом-Алейхема
студент*

Еровлев Павел Андреевич

*Приамурский государственный университет имени Шолом-Алейхема
студент*

Научный руководитель: Баженов Руслан Иванович

*Приамурский государственный университет им. Шолом-Алейхема
к.п.н., доцент, зав. кафедрой информационных систем, математики и
правовой информатики*

Аннотация

В данной статье рассмотрена библиотека TensorFlow и возможность реализации с ее помощью обнаружение объектов. Практическим результатом является рабочее web-приложение с возможностью обнаружения предметов в реальном времени, через web-камеру

Ключевые слова: обнаружение, web-приложение, искусственный интеллект

Real-time Object Detection with TensorFlow Library

Semchenko Regina Viktorovna

*Sholom-Aleichem Priamursky State University
student*

Erovlev Pavel Andreevich

*Sholom-Aleichem Priamursky State University
student*

Scientific adviser: Bazhenov Ruslan Ivanovich

*Sholom-Aleichem Priamursky State University
Candidate of pedagogical sciences, associate professor, Head of the Department
of Information Systems, Mathematics and Legal Informatics*

Abstract

This article discusses the TensorFlow library and the possibility of implementing object detection using it. The practical result is a working web-application with the possibility of setting items in real time, through a web-camera

Keywords: discovery, web application, artificial intelligence

На сегодня охват стран использующих искусственный интеллект становится все больше. Так, например, в некоторых странах имеется возможность найти человека, который пропал, по камерам, при чем это делает даже не человек, а нейросеть подключенная к серверу с камерами. Она ищет из всех возможных лиц похожего на данный ей образец, и если совпадение будет найдено, то люди уже реагируют и отправляются в то место, где его увидела камера.

Классификация изображений может быть очень полезным инструментом, она может дать представление о том, что на изображении. Тем самым упростив обнаружение объектов, которых много, либо скрыты.

В своей статье Хаякава Ясухиса и Фуката Осаму рассмотрели возможность обнаружения трехмерных моделей с помощью нейросети, а так же реализовали модель позволяющую определять объемные модели [1]. Векслер Б.А., Ковалева А.В. и Максимова И.Л. рассмотрели возможность использования нейронных сетей для диагностики биотканей по пространственным характеристикам упругого светорассеяния[2]. Так же Допира Р.В и Маслов Ф.Д описали метод акустической диагностики вращающихся частей антенно-фидерных устройств с применением самообучающиеся нейросети [3]. В статье Горбачева С.В. и Сырямкина В.И был рассмотрен метод местоопределения движущегося объекта с помощью высокопроизводительной нейросетевого алгоритма навигационного счисления [4]. Хамухин А.В. описал устройство, а так же устройство каскадной обработки потока изображений с помощью сверточных нейронных сетей [5].

Обнаружение объекта открывает возможность подсчета количества объектов в сцене, отслеживания движения и просто определения местоположения объекта.

Чтобы делать прогнозы обнаружения объектов, все, что нужно, это импортировать библиотеку «TensorFlow coco-ssd», которую можно установить с помощью менеджера пакетов, или просто импортировать в `<script>` тег. Затем можно загрузить модель и сделать прогноз (рис.1).

```
import React from
import ReactDOM from

import * as cocoSsd from "
import
import
```

Рисунок 1 – добавление библиотеки

Изображение, которое мы передаем в функцию обнаружения, является просто ссылкой на HTML- ``тег.

После того, как будет получен прогноз, то понадобится способ отобразить его на экране. Сделаем так, чтобы объект был обведен в рамку и прописывал, что это за предмет (рис.2).

```
renderPredictions = predictions => {
  const ctx = this.canvasRef.current.getContext("2d");
  ctx.clearRect(0, 0, ctx.canvas.width, ctx.canvas.height);
  const font = "16px sans-serif";
  ctx.font = font;
  ctx.textBaseline = "top";
  predictions.forEach(prediction => {
    const x = prediction.bbox[0];
    const y = prediction.bbox[1];
    const width = prediction.bbox[2];
    const height = prediction.bbox[3];
    ctx.strokeStyle = "#00FFFF";
    ctx.lineWidth = 4;
    ctx.strokeRect(x, y, width, height);
    ctx.fillStyle = "#00FFFF";
    const textWidth = ctx.measureText(prediction.class).width;
    const textHeight = parseInt(font, 10);
    ctx.fillRect(x, y, textWidth + 4, textHeight + 4);
  });
};
```

Рисунок 2 – отображение объектов

Чтобы нарисовать ограничивающие рамки, будем использовать `<canvas>` элемент.

Элемент `<canvas>` позволяет использовать функцию `strokeRect`, которая идеально соответствует результатам прогнозирования (рис.3).

```
const x = prediction.bbox[0];
const y = prediction.bbox[1];
const width = prediction.bbox[2];
const height = prediction.bbox[3];

const canvas = document.getElementById("canvas");
const ctx = canvas.getContext("2d");

ctx.strokeRect(x, y, width, height);
```

Рисунок 3 – Отрисовка рамок

Выполнить обнаружение в режиме реального времени в потоке веб-камеры очень легко, за исключением большого кода для запуска веб-камеры (рис.4).

```
const video = document.getElementById("video")

navigator.mediaDevices
  .getUserMedia({
    audio: false,
    video: {
      facingMode: "user",
      width: 600,
      height: 500
    }
  })
  .then(stream => {
    video.srcObject = stream
    video.onloadedmetadata = () => {
      video.play()
    }
  })
```

Рисунок 4 – подключение веб-камеры

Затем элемент видео будет передавать запрос модели для обнаружения. Однако на этот раз вызовем «requestAnimationFrame», который будет вызывать функцию обнаружения снова и снова в бесконечном цикле, пропуская кадры (рис.5).

```
function detectFrame() {
  model.detect(video).then(predictions => {
    renderOurPredictions(predictions)
    requestAnimationFrame(detectFrame)
  })
}
```

Рисунок 5 – Вызов функции

На этом составление кода готово, можно запускать и проверять работоспособность. Для начала пусть он определит человека (рис.6).

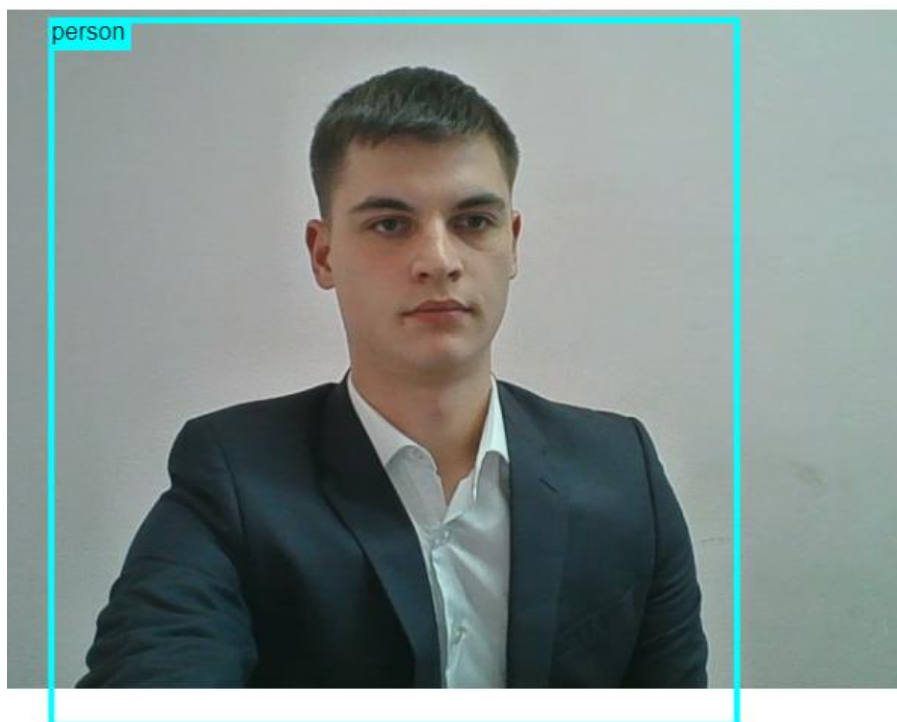


Рисунок 6 – Определение человека

Следующим шагом попробуем определить телефон (рис.7).

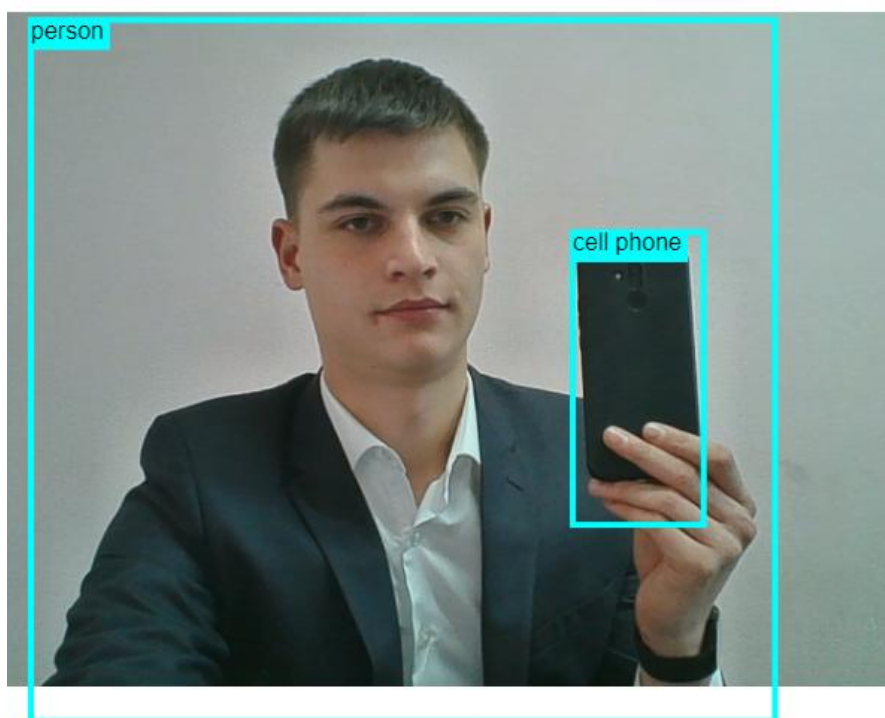


Рисунок 7 – Определение телефона

Как видно из картинок, то определение объектов работает, но бывают моменты, когда она не всегда определяет правильно, но с каждым разом она обучается и представляет вариант более точный.

Практическим результатом была рассмотрена библиотека «TensorFlow» и разработано web-приложение, позволяющее определять предметы, а так же животных и людей.

Библиографический список

1. Хаякава Я. Устройство обнаружения трехмерных объектов и способ обнаружения трехмерных объектов // Научно-практическое исследование. 2019. №7(14). С. 77-86.;
2. Векслер Б.А., Ковалева А.В., Максимова И.Л. Использование нейронных сетей для диагностики биотканей по пространственным характеристикам упругого светорассеяния // Синергия наук. 2015. №15. С. 45-52.;
3. Допира Р.В., Маслов Ф.Д. Метод акустической диагностики вращающихся частей антенно-фидерных устройств с применением самообучающейся нейросети // Социальные коммуникации: наука, образование, профессия. 2017. №8. С. 88-97.;
4. Горбачев С.В., Сырякин В.И. Высокопроизводительные нейросетевые алгоритмы навигационного счисления // ЕО IPSO. 2015. №1. С. 7-22.;
5. Хамурин А.В. Устройство и способ каскадной обработки потока изображений с помощью свёрточных нейронных сетей // Научно-практическое исследование. 2015. №7(13). С. 67-82.