

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

по дисциплине «Прикладные интеллектуальные системы и экспертные
системы»

Компьютерное зрение

Студент

Бахмутский М.В.

Группа М-ИАП-22

Руководитель

Кургасов В.В.

Доцент, к.п.н.

Липецк 2022 г.

Ход работы

Краткая теоретическая справка

Компьютерное зрение — это область компьютерных наук, которая стремится расширить возможности компьютеров по идентификации и определению объектов и людей на изображениях и видео. Как и другие типы ИИ, компьютерное зрение ориентируется на выполнение и автоматизацию задач, имитирующих человеческие возможности. В этом случае компьютерное зрение старается имитировать зрение и восприятие человека.

Спектр практического применения технологий компьютерного зрения обуславливает то, что оно стало центральным компонентом множества современных инноваций и решений. Рабочие нагрузки компьютерного зрения можно выполнять в облаке или локально.

Принципы работы компьютерного зрения

Приложения компьютерного зрения используют входные данные с сенсорных устройств, возможности ИИ, машинного обучения и глубокого обучения для имитации того, как работает человеческое зрение. Такие приложения работают на основе алгоритмов, обученных на огромных объемах визуальных данных или изображений в облаке. Они распознают образы в визуальных данных и с их помощью определяют содержимое, присутствующее на изображениях.

Анализ изображения с помощью технологий компьютерного зрения

- Сенсорное устройство создает изображение. Сенсорное устройство часто представлено просто фотокамерой, но может быть видекамерой, устройством медицинской визуализации или любым другим устройством, создающим изображение для анализа.
- Изображение затем отправляется на устройство для анализа. Устройство для анализа использует возможности распознавания изображений для разбиения изображения на отдельные части, сравнения найденных образов с библиотекой известных образов и их сопоставления. Образом может быть что-то общее, например внешний

вид объекта определенного типа, или же образ может быть основан на уникальных признаках, таких как черты лица.

- Пользователь запрашивает определенную информацию об изображении, а анализирующее устройство предоставляет такие сведения по результатам анализа изображения.

Глубокое обучение и компьютерное зрение

Современные приложения компьютерного зрения все чаще используют для анализа изображений возможности глубокого обучения, а не статистические методы, как раньше. Глубокое обучение позволяет такому приложению выполнять определенный алгоритм, который называется нейронной сетью, позволяющий более точно выполнять анализ. Кроме того, при использовании глубокого обучения программа компьютерного зрения может сохранять информацию о каждом проанализированном изображении и с течением времени еще больше повышать точность.

Возможности компьютерного зрения

- Классификация объектов. Система классифицирует объекты на изображении по определенной категории. Например, при классификации объектов компьютер может отличать людей от предметов, а также определять их количество.
- Индексация объекта. Система идентифицирует определенный объект на фото, видео или изображении. Например, при идентификации объектов система может не только отличать людей на фото, но также анализировать их внешний вид для установления личности или признаков таких людей.
- Отслеживание объекта. Система анализирует видео для обработки данных о расположении движущегося объекта со временем. Например, при отслеживании объектов камера на парковке может идентифицировать автомобили и предоставлять данные об их расположении и передвижении.

- Оптическое распознавание символов. Система распознает буквы и цифры на изображениях и преобразует их в компьютерный текст, который может быть считан другими компьютерными приложениями или изменен пользователями.

Описание работы

Обнаружение объектов с использованием каскадных классификаторов на основе признаков Хаара - это эффективный метод обнаружения объектов, предложенный Полом Виолой и Майклом Джонсом в их статье "Быстрое обнаружение объектов с использованием усиленного каскада простых признаков" в 2001 году. Это подход, основанный на машинном обучении, при котором каскадная функция обучается на основе множества положительных и отрицательных изображений. Затем он используется для обнаружения объектов на других изображениях.

В данной работе мы будем работать с распознаванием лиц. Изначально алгоритму требуется много положительных изображений (изображений лиц) и отрицательных изображений (изображений без лиц) для обучения классификатора. Затем нам нужно извлечь из него функции. Для этого используются функции haar, показанные на изображении ниже. Они точно такие же, как наше сверточное ядро. Каждый объект представляет собой одно значение, полученное путем вычитания суммы пикселей под белым прямоугольником из суммы пикселей под черным прямоугольником

Результат выполнения

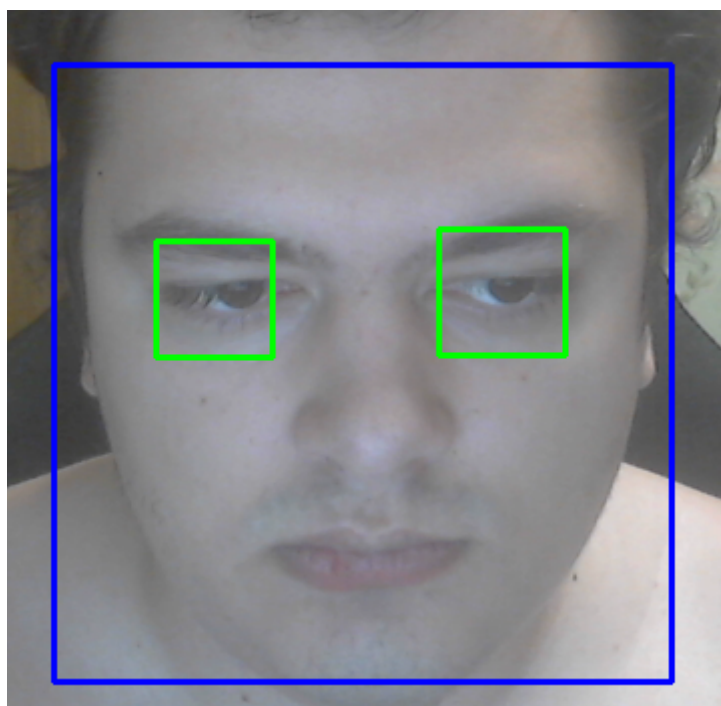


Рисунок 1 - Пример выполнения программы

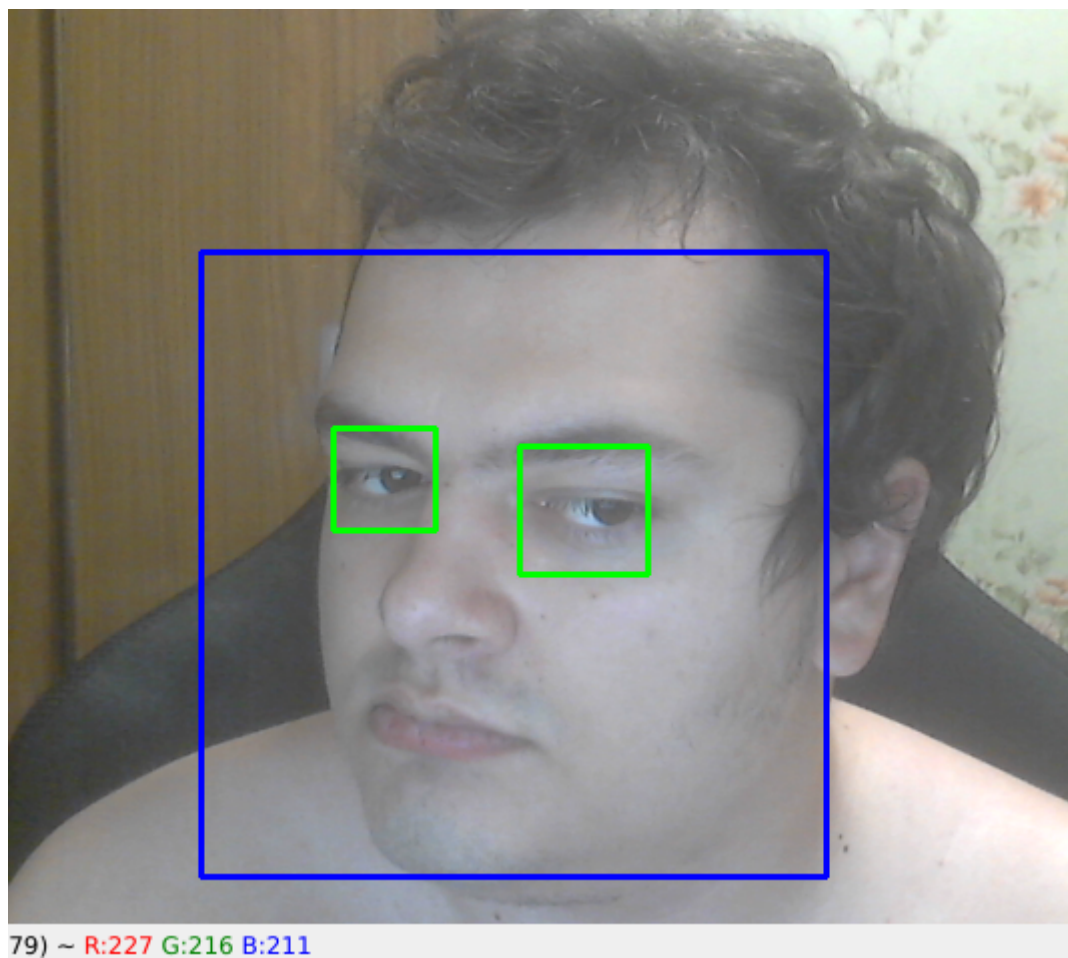


Рисунок 2 - Пример выполнения программы

Вывод

В ходе выполнения индивидуального домашнего задания были изучены теоретические основы компьютерного зрения и реализована программа поиска лица и глаз на языке Python.

Для этого использовалась библиотека OpenCV — это open source библиотека компьютерного зрения, которая предназначена для анализа, классификации и обработки изображений.

Код программы

```
import numpy as np
import cv2

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
# smile_cascade=cv2.CascadeClassifier('haarcascade_smile.xml')
#body_cascade=cv2.CascadeClassifier('haarcascade_fullbody.xml')
cap=cv2.VideoCapture(0)
while True:
    ret, img=cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray)
    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        # font=cv2.FONT_HERSHEY_SIMPLEX

#cv2.putText(img,'Jeevy',(x+w/2,y+h/2),font,0.5,(0,0,255),1,cv2.LINE_AA)
    print(x,y)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]
    eyes=eye_cascade.detectMultiScale(roi_gray)
    # smiles=smile_cascade.detectMultiScale(roi_gray)
    for(ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex, ey),(ex+ew, ey+eh),(0,255,0),2)
        print(ex,ey)
    # for(sx,sy,sw,sh) in smiles:
        # cv2.rectangle(roi_color,(sx, sy),(sx+sw, sy+sh),(0,0,255),2)
        # print(sx,sy)
cv2.imshow('img',img)
```



```
k=cv2.waitKey(30) & 0xFF
if k==32:
    break

cap.release()
cv2.destroyAllWindows()
```