

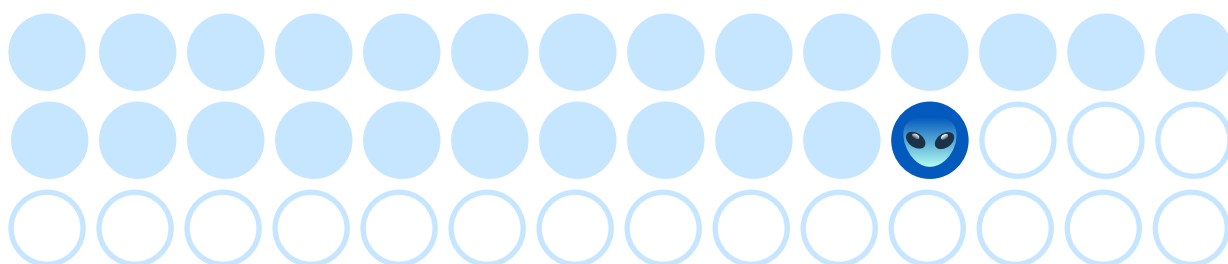
Инструменты ИИ: применяем и программируем

2 модуль

Занятие 2.3.3

Отображение видео с помощью OpenCV на Python

О занятии	1
Практическая работа	2
Итоги занятия	4



О занятии

Рабочая тетрадь предназначена для слушателей курса «Инструменты ИИ: применяем и программируем».

Данное пособие поможет слушателям лучше усвоить материал, представленный в видеоролике. Содержит практические задания для закрепления навыка работы с нейросетями, а также инструкции по регистрации и авторизации на сайтах нейросетей.

В рабочей тетради предложены термины, которые помогут лучше понять темы и закрепить знания об искусственном интеллекте.

Цель практического занятия:

На этом занятии мы создадим медитацию с использованием функции Гигачата. Изучим детекцию в компьютерном зрении, познакомимся с каскадами Хаара и принципом их работы. Повторим шаги по настройке рабочей среды, напишем код для анализа изображения и подсчёта количества лиц на фото, а также код для обнаружения лиц с использованием веб-камеры. В заключение мы научимся анализировать различные видео с помощью детекции лиц.

Задачи:

- Создать медитацию с помощью Гигачата.
- Узнать о детекции и каскадах Хаара в компьютерном зрении.
- Составить код для подсчёта количества лиц на фото.
- Создать код для обнаружения лиц с использованием веб-камеры.

Необходимые инструменты:

Gigachat, Visual Studio Code.

Теория и термины

Детекция — обнаружение объектов определённого типа на изображении или видео, например, лиц, жестов, движений.

Каскады Хаара — метод, который используется для обнаружения объектов на изображениях.

Bounding box — ограничивающий прямоугольник, который обводит обнаруженный объект на изображении.

Практическая работа

Задание 1

В главном меню Гигачата выберите раздел «Полезные AI-функции» и нажмите «Устройте себе сеанс медитации».

Опишите, что сейчас чувствуете, например, усталость, упадок сил.

Опишите желаемый результат медитации: вернуть энергию и снова стать продуктивным.

Опишите куда вы отправитесь медитировать, например, в горы.

Затем выберите «Сгенерировать текст» и нажмите «Продолжить».

Выберите звуковое сопровождение и немного подождите — медитация будет готова.

Задание для самостоятельной работы:

При помощи Гигачата создайте медитацию, которая учтёт ваши пожелания.

Задание 2

Создайте новую папку «Детекция» и файл с расширением .py под названием «Count_detection» внутри этой папки.

Выполняйте те же шаги по настройке рабочей среды, что делали в предыдущем занятии.

Задание 3

Откройте Visual Studio Code, импортируйте модуль OpenCV.

Затем в переменной `img` загрузите изображение и преобразуйте его в оттенки серого.

Далее в переменной `face_cascade` запишите каскад Хаара, необходимый для детекции лиц.

Создайте переменную «`faces`», в которой будут храниться координаты обнаруженных лиц на изображении.

Затем программа подсчитывает количество этих найденных лиц и выводит результат.

Задание для самостоятельной работы:

Выберите фото с семьёй или с друзьями и используйте предложенный код для подсчёта количества лиц. Код находится в материалах к занятию в документе: «Код для подсчёта лиц на фотографии».

Задание 4

Создайте новый файл с расширением .py «Detection» и начните с похожих строк кода, как в предыдущем примере.

Выберите источник видеоданных — веб-камеру, раскомментировав строчку для чтения видео с веб-камеры.

Задание 5**Промпт:**

Почему в функции `cv2.VideoCapture()` используется аргумент 0 для захвата видео с веб-камеры?

Возможный результат (Критерии оценивания и формат оформления):

Аргумент 0 в функции `cv2.VideoCapture()` используется для указания источника видео.

В данном случае, когда мы хотим получить видео с веб-камеры, аргумент 0 указывает на то, что источником является первая доступная веб-камера.

Если у вас есть несколько веб-камер, вы можете использовать аргументы 1, 2 и т.д. для выбора конкретной камеры.

Задание для самостоятельной работы:

При помощи Гигачата узнайте, какие форматы видео поддерживает OpenCV при использовании функции `cv2.VideoCapture()`.

Задание 6

Продолжайте работать в файле «Detection.py».

Переносится в код цикл, который будет считывать по очереди каждый кадр из видеопотока и анализировать его.

Затем следующая часть кода рисует bounding box для лица.

Далее показывается изображение, то есть видео, в отдельном окне. Это окно будет открыто, пока вы не нажмёте клавишу ESC на клавиатуре.

Каждые 30 миллисекунд проверяется, была ли нажата клавиша с кодом 27, то есть ESC и если да, то цикл прерывается.

И наконец, после цикла дописывается в код команда `cap.release()`, останавливающая приём видеопотока.

Код находится в материалах к занятию в документе: «Код для обнаружения лиц в видео с помощью каскада Хаара».

Задание для самостоятельной работы:

Выполните детекцию лица в видеопотоке, используя любое видео.

Итоги занятия

На этом занятии вы:

- Познакомились с медитацией от Гигачата.
- Узнали, что такое детекция компьютерного зрения.
- Познакомились с каскадом Хаара для обнаружения лиц.
- Научились подсчитывать количество лиц на фотографии с помощью каскада Хаара.
- Узнали, как обрабатывать видеопотоки с веб-камеры и видеофайла при помощи OpenCV.

Дальше интереснее. До встречи на следующих занятиях!

Молодцы!

Переходите на платформу для решения задач.

Увидимся на следующем занятии!