**Субтитры ИИ\_25 «Как обрабатывается видео в OpenCV-Python»**

Источником видеоданных может быть видеокамера или видеофайл…

Ваня, с тобой все хорошо? Ты уже сколько времени сидишь за книжкой?

Да я только начал читать, но уже совсем нет сил, хочется спать.

Похоже, тебе нужно немного расслабиться. У Гигачата есть функция медитации. Давай попробуем вернуть твою энергию!

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

Сначала вы можете ознакомиться с гайдом, а потом создать медитацию.

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

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

Куда вы отправитесь медитировать, например, в горы.

Затем выберите «Сгенерировать текст» и нажмите «Продолжить». Выберите звуковое сопровождение и немного подождите – медитация будет готова.

Ух ты, это очень классно! Расслабляет и снимает усталость! Чувствую себя замечательно!

Отлично, рада, что ты взбодрился! Теперь давай начнем изучение темы обработки видео в OpenCV.

Да, я прочитал, что можно использовать OpenCV для отображения видеоданных с веб-камеры. Там используется какая-то детекция, но дальше я запутался.

Давай разбираться по порядку.

Для поиска и выделения объектов применяются различные специализированные алгоритмы и методы. Рассмотрим один из таких методов.

Он был назван в честь венгерского математика Альфреда Хаара.

Я ничего про него не слышал… А расскажи, как происходит обучение классификатора Хаара на видео.

Классификатор Хаара обучается на большом наборе положительных и отрицательных изображений.

Положительные изображения содержат нужный объект, например, лицо, а отрицательные – его не содержат.

Во время обучения классификатор применяет несколько простых проверок классификаторов последовательно, чтобы быстро и точно распознать объект.

После обучения классификатор используется для анализа каждого кадра видео.

Ну да, точно!

Видео — это всего лишь набор кадров, являющихся изображениями.

Теперь понятно, как это работает, не терпится применить метод самому!

Отлично, но начнем с анализа изображения и подсчета количества лиц на фото.

Для этого создадим новую папку «Детекция»

и файл с расширением .py под названием «Count\_detection» внутри этой папки.

Выполняйте те же шаги, что делали в предыдущем занятии.

Подробное описание этапов работы и картинку можно найти в материалах  к занятию.

Открываем visual studio code.

Для начала импортируем модуль OpenCV.

Затем в переменной img загружаем изображение и изменяем его в оттенки серого.

Далее в переменной face\_cascade запишем каскад Хаара, необходимый для детекции лиц.

А что за конструкция в кавычках? Раньше не встречал такое расширение.

Это название файла, который содержит предобученный каскад Хаара для обнаружения лиц в изображениях и видео.

Продолжим.

Создаем переменную “faces”, в которой будут храниться координаты обнаруженных лиц на изображении.

Затем подсчитываем количество этих найденных лиц и выводим результат.

В консоли вы увидите надпись «Количество человек: 3», которая отображает количество обнаруженных лиц на изображении.

Получилось, но только странно! На фотографии 4 человека, а обнаружилось только 3. Может, это связано с тем, что девочка и ее мама сидят в одинаковой красной одежде?

Возможно. Алгоритм детекции лиц иногда путает людей, которые близко расположены или имеют похожий цвет одежды. Для улучшения точности можно настроить параметры алгоритма или использовать более сложные модели обучения.

Понял, спасибо! Теперь я готов работать с видео!

Отлично! Давай создадим новый файл с расширением .py «Detection» и начнем с похожих строк кода, как в предыдущем примере.

Ваня, теперь выбираем источник видеоданных.

Если ты все так же хочешь использовать веб-камеру, мы сможем протестировать детекцию лиц прямо на тебе!

Если перехотел, то можем использовать видеофайл, который я заранее тебе подготовила.

Хочу все-таки попробовать через веб-камеру!

Помните, что одновременно можно использовать только одну из строчек, начинающихся с “cap = ...”

Тогда первый вариант беру к себе в код!

А мне вот интересно, почему при вызове функции для использования веб-камеры стоит в скобках 0?

Отличный вопрос! Давай отправим этот фрагмент Гигачату и спросим его.

Понятно, ну у меня одна камера, поэтому оставлю 0.

Хорошо, что разобрались!

Давай теперь попробуем написать цикл, который будет считывать по очереди каждый кадр из видеопотока и анализировать его.

Нейросеточка, а ты случайно не ошиблась в третье строке?

Нет, я не ошиблась.

Метод cap.read() возвращает два значения: логическое True/False, которое указывает, успешно ли считался кадр, и сам кадр.

Нам нужен только кадр, поэтому мы используем \_ для игнорирования первого значения.

Если бы мы не написали \_, это бы не сработало, так как метод возвращает два значения, а не одно.

Продолжим писать внутри цикла. Следующая часть потребует особого внимания, потому что в этом цикле мы рисуем bounding box для лица.

О, а я знаю, каким цветом будет этот прямоугольник!

Он будет синим в цветовой модели BGR!

Еще предполагаю, что x, y, w и h — это координаты, как в математике: h — это высота, правильно?

Верно, Ваня!

x и y — это координаты верхнего левого угла прямоугольника.

w — это ширина, а h — высота.

Координаты нижнего правого угла вычисляются как x + w, y + h.

Функция cv2.rectangle() рисует прямоугольник, для этого ей нужны: изображение, координаты верхнего левого угла,

нижнего правого, цвет обводки и толщина линии.

Эти параметры можно настроить по собственному желанию, но пока давай оставим все так, как есть.

А теперь осталось показать изображение, то есть видео, в отдельном окне.

Это окно будет открыто, пока вы не нажмете 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 дописываем последнюю команду, останавливающую прием видеопотока.

Подробную инструкцию и весь код вы найдете в материалах к занятию.

Классно получилось! Теперь я могу анализировать различные видео с помощью детекции лиц!

Увидимся на следующем занятии! До скорой встречи!