**Субтитры ИИ\_26 «Поиск цветных объектов на видео с помощью OpenCV-Python»**

Привет! Ваня, помнишь, мы с тобой искали цветные объекты  на фотографиях?

Да, помню. Было интересно!

А ты знаешь, что поиск цветовых объектов на изображении используется при разработке автономных систем управления беспилотными автомобилями? Например, чтобы транспортное средство могло распознавать дорожные знаки или сигналы светофора.

Ого, круто!

Но я заметил, что распознаются не только нужные, но и лишние объекты.

Почему так происходит?

Потому что мы искали по всему кадру.

Но искомый объект может быть только в определенной его части, и искать нужно именно там. Это позволит не захватывать лишнее.

Это как искать конфету в коробке, а не по всему столу!

Точно! И тогда вероятность найти то, что нужно, гораздо выше.

Но сначала нужно определить область поиска.

Мы можем подобрать размеры области опытным путем.

А что это значит?

Это значит, что мы будем пробовать разные размеры и смотреть, какой результат получается.

И выберем тот, который лучше всего подходит для нашей задачи.

Звучит сложновато. Но я готов попробовать!

Порядок действий для поиска цветного объекта в видеопотоке будет состоять из нескольких этапов:

Давай напишем небольшую программу, которая реализует этот алгоритм.

Для программы нам пригодится библиотека NumPy.

Она может быть полезна для выполнения математических расчетов.

Например, если тебе нужно решить систему линейных уравнений или найти корни многочлена, то NumPy может значительно ускорить этот процесс.

А какое отношение NumPy имеет к OpenCV?

OpenCV использует NumPy для работы с массивами данных,

например, пиксельными изображениями.

Инструкция по установке NumPy находится в материалах к занятию.

Давай начнем писать программу.

Cначала выбери видео, которое нужно обработать.

Ой, а у меня нет видео.

Ничего страшного, ты можешь попросить нейросеть сгенерировать небольшое видео.

Нейросети уже умеют создавать реалистичные видео-анимации, которые используются в игровой индустрии или кино.

Так, например, российская нейронная сеть Kandinsky Video умеет создавать небольшие видео.

Воспользоваться ей можно двумя способами:

Подробная инструкция находится в материалах к занятию.

Я хочу создать видео с птицами на фоне неба.

Переходи на сайт https://fusionbrain.ai/t2v/.

Выбери вкладку «Видео» на панели слева.

В поле с запросом введи свой текст, установи необходимый размер

и выбери модель. Это может быть анимация

или видео.

Хорошо.

Ввожу текст:

Нажимаю Создать видео.

Время генерации 4 минуты. Подождем.

После генерации видео нужно нажать на кнопку «Скачать», расположенную в правом верхнем углу.

Думаю, что сначала надо подключить библиотеку CV и открыть видеофайл?

Еще не забудь подключить библиотеку NumPy.

Далее нужно воспользоваться циклом while для непрерывного считывания кадров  из видеопотока до тех пор, пока не будет достигнут конец видео.

Внутри цикла выполняются следующие шаги:

Цикл завершится когда будет достигнут конец видеопотока.

А какой код для этого нужно написать?

Необходим вот такой фрагмент:

После цикла тебе нужно будет прописать две строки.

Первая — освобождает ресурсы, занятые видеопотоком, что позволяет программе корректно закрыть видеофайл и освободить память.

И вторая — закрывает все окна созданные OpenCV.

Сложно. Надо попросить Гигачат рассказать, что означает каждая строка в цикле.

Запуск бесконечного цикла, пока открыто видеоизображение capImg.

Чтение кадра из видео изображения capImg и присвоение его переменной frame.

Проверка на успешность чтения кадра (ret) и на наличие кадра в переменной frame. Если чтение не удалось или кадр отсутствует, то происходит выход из цикла.

Преобразование цветового пространства кадра из BGR в HSV и сохранение результата в переменную frame\_hsv.

Получение высоты и ширины кадра и присвоение их переменным height и width соответственно.

Обрезка исходного кадра и сохранение результата в переменную crop\_frame.

Обрезка кадра в цветовом пространстве HSV по тем же границам, что и для crop\_frame, и сохранение результата в переменную crop\_frame\_hsv.

Определение нижней и верхней границ диапазона желтого цвета в цветовом пространстве HSV.

Создание маски, которая будет выделять пиксели с желтой компонентой цвета в crop\_frame\_hsv, и сохранение результата в переменную yellow\_mask.

Отображение полученной маски на экране с названием "video\_mask".

Отображение обрезанного кадра на экране с названием "video\_frame".

Ожидание нажатия клавиши и присвоение кода нажатой клавиши переменной key\_press.

Проверка нажатой клавиши, если это 'q', то происходит выход из цикла (выход из программы).

Здорово, Гигачат построчно все объяснил.  Стало понятнее.

Запускаю программу. Работает!

Основываясь на предыдущих промптах, выполните задания.

Отлично поработали! До встречи на следующем занятии!