Этот код считывает видеофайл (3.mp4) и применяет методы обработки изображения для обнаружения сигналов трафика в видеокадрах.

Во-первых, он изменяет размер каждого видеокадра до более низкого разрешения на основе указанного процентного значения (процентов). Это сделано для ускорения обработки каждого кадра.

Затем он применяет ряд методов обработки изображений, чтобы выделить цвета сигналов светофора. Функция cv2.cvtColor используется для преобразования каждого кадра с измененным размером из цветового пространства BGR по умолчанию в цветовое пространство HSV. Для трех цветов сигналов светофора определены три цветовых диапазона: красный, желтый и зеленый.

Далее создается маска для каждого цветового диапазона с помощью cv2.inRange. Маски объединяются с помощью cv2.bitwise\_and для создания единой маски, включающей все три цвета.

К маске применяется размытие по Гауссу, чтобы сгладить любой шум, а затем с помощью детектора краев Канни обнаруживаются края. Затем края расширяются с помощью ядра 3x3, чтобы сделать их более заметными.

Контуры обнаруживаются с помощью cv2.findContours на расширенных краях. Для каждого контура код вычисляет минимальную охватывающую окружность и проверяет ее положение на изображении. Если круг находится в пределах определенной области верхнего левого угла кадра, контур рисуется фиолетовым цветом, а стоп-переменная устанавливается на 1. Если круг слишком мал или выходит за пределы обозначенной области, стоп-переменная устанавливается на 0 и контур не рисуется.

Наконец, обработанный кадр отображается на экране вместе с обнаруженными контурами. Цикл продолжается до тех пор, пока пользователь не нажмет клавишу «q», после чего видео отключается и все окна закрываются.

В целом код использует комбинацию методов пороговой обработки цвета, обнаружения границ и обнаружения контуров для идентификации сигналов трафика в видеопотоке.

Этот раздел кода находит контуры в кадре, а затем проверяет, соответствует ли контур сигналу светофора.

Первая строка этого раздела для i in range(len(contours)): создает цикл, который перебирает все контуры, обнаруженные в текущем кадре.

Следующая строка (x, y), radius = cv2.minEnclosingCircle(contours[i]) вычисляет минимальную охватывающую окружность для i-го контура в текущем кадре и извлекает центр окружности как (x,y) и радиус окружности.

Строка if x < 400 и y < 500: проверяет, находится ли центр круга в пределах заданного диапазона координат, который можно рассматривать как интересующую область в кадре, где ожидаются сигналы светофора.

Затем строка if cv2.contourArea(contours[i]) > 100: проверяет, больше ли площадь контура, чем 100, что используется для фильтрации небольших шумовых контуров.

Если контур удовлетворяет условиям, функция cv2.drawContours() используется для рисования контура на рамке с определенным цветом и толщиной. Переменная остановки установлена на 1, указывая на то, что сигнал светофора был обнаружен.

Если контур не удовлетворяет условиям, переменная остановки устанавливается в 0 и кадр отображается с другим именем окна.

Наконец, строка if cv2.waitKey(1) & 0xFF == ord('q'): проверяет, была ли нажата клавиша 'q' для выхода из цикла и завершения программы.

В коде используется функция cv2.findContours() для извлечения контуров из бинарного изображения, созданного путем пороговой обработки входного видеокадра. Функция возвращает список контуров и иерархию для заданного порогового изображения. Функция findContours() принимает три аргумента: входное изображение, режим поиска контура и метод аппроксимации контура.

Флаг cv2.RETR\_EXTERNAL используется как режим поиска контуров, который извлекает только внешние контуры и отбрасывает все внутренние контуры. Флаг cv2.CHAIN\_APPROX\_SIMPLE используется как метод аппроксимации контура, который сжимает горизонтальные, вертикальные и диагональные сегменты и оставляет только их конечные точки.

Как только контуры получены, код выполняет итерацию по каждому контуру и находит минимальный окружающий круг, используя функцию cv2.minEnclosingCircle(). Если центр круга лежит в пределах заданной области (x < 400 и y < 500) и если площадь контура больше определенного порога (100), контур считается сигналом светофора и рисуется на кадр с помощью функции cv2.drawContours(). Если площадь контура меньше порога, то он не считается сигналом светофора и не рисуется на кадре. Переменная остановки используется для отслеживания того, был ли обнаружен сигнал светофора или нет.

Как можно решить задачу ?

Распознавание светофоров под разными углами и в разных погодных условиях может быть сложной задачей. Вот несколько подходов, которые вы можете рассмотреть для повышения производительности вашей системы распознавания сигналов светофора:

Обучите модель глубокого обучения: вы можете обучить сверточную нейронную сеть (CNN) распознавать сигналы светофора. Этот подход может быть более надежным и точным, чем традиционные методы компьютерного зрения. Вы можете использовать методы увеличения данных, чтобы генерировать больше обучающих данных с разными углами и погодными условиями.

Используйте несколько цветовых порогов: светофоры имеют разные цвета, и вы можете использовать несколько цветовых порогов для их обнаружения. Например, вы можете использовать пороговые значения красного, желтого и зеленого цветов для обнаружения соответствующих источников света. Вы можете настроить пороги в зависимости от условий освещения.

Используйте методы на основе признаков: вы можете использовать методы на основе признаков для обнаружения светофоров на основе их формы, размера и положения. Например, вы можете использовать детектор круга для определения круглой формы огней. Вы также можете использовать алгоритм сопоставления шаблонов, чтобы сопоставить шаблон светофора с предварительно определенным шаблоном.

Используйте отслеживание объектов: как только вы обнаружите светофор, вы можете использовать отслеживание объектов, чтобы отслеживать его с течением времени. Это может помочь вам распознать светофор, даже когда он движется или меняет свое положение.

Используйте ансамблевые методы: вы можете комбинировать несколько подходов для повышения производительности вашей системы. Например, вы можете использовать модель глубокого обучения для обнаружения светофоров и использовать методы на основе признаков для уточнения результатов.

Стоит отметить, что распознавание сигналов светофора под разными углами и в разных погодных условиях — сложная задача, и универсального решения не существует. Возможно, вам придется поэкспериментировать с различными подходами и параметрами, чтобы найти лучшее решение для вашего конкретного приложения.

Решение с помощью ROI

Код считывается из видеофайла и определяет область интереса (ROI), где ожидается сигнал светофора. Затем он перебирает каждый кадр видео, извлекает область интереса и выполняет некоторую обработку изображения, чтобы попытаться изолировать светофор. Конкретно:

Область интереса преобразуется в оттенки серого и размывается для уменьшения шума.

Операция пороговой обработки выполняется для отделения светофора от фона.

Контуры находятся на пороговом изображении.

Контур с наибольшей площадью принимается за светофор, вокруг него рисуется окружность.

Затем ROI и пороговое изображение отображаются в отдельных окнах. Если нажата клавиша «q», цикл прерывается и программа завершается.

Обратите внимание, что в этом коде предполагается, что светофор всегда находится в одном и том же месте на видео (т. е. в области интереса) и что условия освещения относительно одинаковы во всех кадрах. На практике эти допущения могут не выполняться, и может потребоваться дополнительная обработка (например, сегментация на основе цвета) для повышения точности обнаружения сигнала светофора.