



C++ Developer. Professional

Меня хорошо видно & слышно?



Защита проекта

Тема: Обработка потоковой передачи видео с использованием библиотеки *OpenCV*



Осипов Дмитрий

Разработчик C++



План защиты



Цели проекта

Что планировалось

Используемые
технологии

Что получилось

Выводы

Краткое описание проекта:

Разработать клиентское приложение на базе библиотеки OpenCV :

- ✓ • с возможностью подключения к камере (серверу) или нескольким камерам,
- ✓ • с возможностью распознавания объектов/людей/лиц,
- ✗ • с возможностью регистрации события (появления объекта на изображении),
- ✗ • с возможностью внесения информации о событии (источник регистрации/дата и время регистрации/тип объекта).

Цели проекта

1. Изучить использование пакетных менеджеров и взаимодействие их с CMake
2. Более подробно изучить работу CMake и формирование структуры проекта
3. Ознакомиться с работой библиотеки OpenCV и основными принципами обработки видео/изображений
4. Ознакомиться с языком запросов SQL на примере работы с СУБД SQLite
- 5.

Что планировалось

1. Научить камеру на базе микроконтроллера ESP32 (ESP32-CAM) передавать потоковое видео по протоколу [RTSP](#)
2. Сформировать качественную структуру проекта
3. Разработать класс клиента для подключения к камере
4. Разработать класс очереди для передачи изображения в очередь на обработку
5. Разработать класс анализа изображения для детектирования объектов
6. Разработать класс для формирования структуры данных каждого распознанного объекта и записи его в БД
- 7.

Используемые технологии

1. Cmake

2. Conan

3. OpenCV

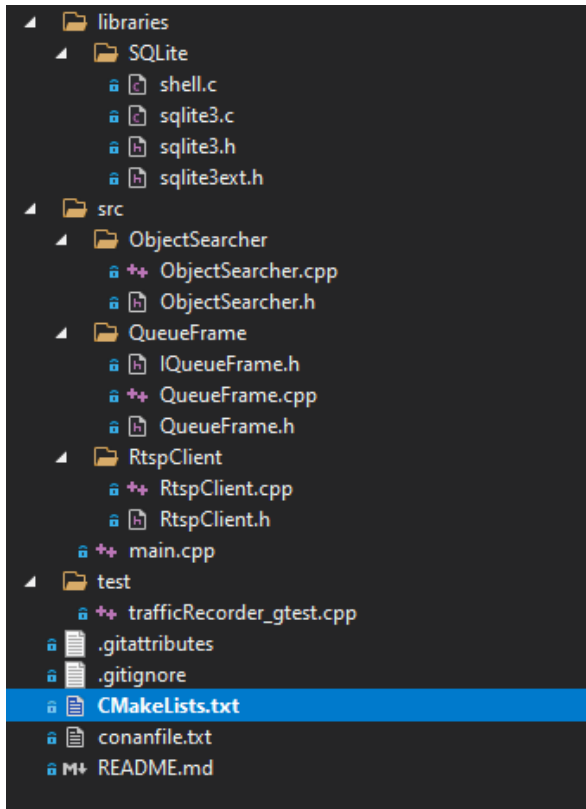
4. SQL (SQLite)

5. GoogleTest

6. Boost

7.

CMake



Conan

Build Windows

Если используется пакетный менеджер Conan, то перед сборкой проекта необходимо установить флаг CONAN_PKG опции (по умолчанию OFF)

```
cmake . -DCONAN_PKG=ON
```



Установка пакетов с использованием версии Conan выше 2 с использованием preset

```
cd "./project_dir"  
conan install conanfile.txt --build=missing
```



- Создается файл "CMakeUserPresets" в корневой папке
- Создается папка `./build` из-за опции в "conanfile"

```
[layout]  
cmake_layout
```

- Создается папка `./build/generators` с нужной фигней для CMake
Далее из папки проекта вызываем

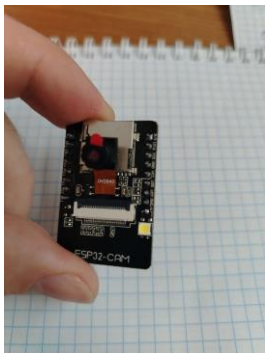
```
cmake --preset conan-default  
cmake --build . --preset conan-release
```



Ак

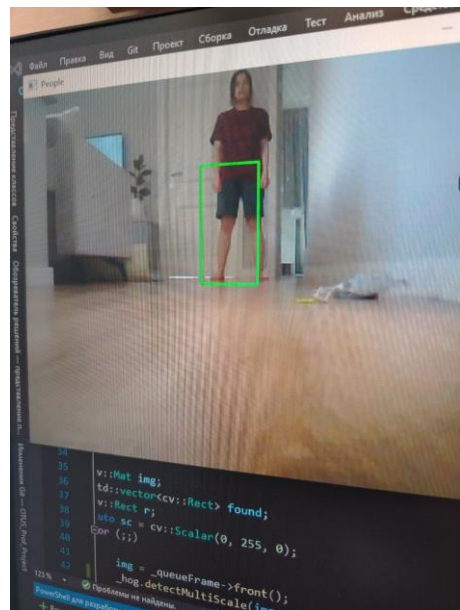


Что получилось



Это камера на базе микроконтроллера ESP32. Думаю многие с ней знакомы. Она умеет передавать видео по протоколу RTSP. Спасибо оптимизированной библиотеке Micro-RTSP и ее создателю. Примеры использования и исходный код можно посмотреть [тут](#).

Вот как выглядит распознавание на потоковом видео с использованием [OpenCV](#)



Выводы и планы по развитию

1. Доделать основной функционал, который задумывался
2. Доработать структуру проекта и оптимизировать CMakeLists
3. Оптимизировать обработку изображения (распознавание) используя различные алгоритмы. Попробовать использовать [YOLO](#) (интересная статья на [хабр](#))
4. Разработать клиентскую часть приложения как серверную.
5. Разработать механизмы позволяющие отслеживать выбранный объект
6. Многое-многое другое
- 7.

Спасибо за внимание!