

iANRCR SDK

Версия 2.0.1.0
Документация

СОДЕРЖАНИЕ

iANRCR SDK

1. Функции для работы с библиотекой

2. Инсталляция и использование

2.1. Windows

3. Примеры на C/C++ для Windows

3.1. DetectNumber

iANRCR SDK

iANRCR SDK – это комплект средств разработки для распознавания 8-мизначных номеров железнодорожных вагонов. Основная цель – обеспечить автоматизированное распознавание номеров на основе библиотеки компьютерного зрения OpenCV. Основным языком использования библиотеки – C/C++.

Версия 2.0.1.0 откомпилирована с версиями OpenCV 3.4.

Функции для работы с библиотекой

ProcessANRCR

Функция поиска 8-мизначного номера железнодорожного вагона на изображении формата OpenCV.

```
OUT_ANRCR ProcessANRCR  
(  
    iANRCRObject object,  
    cv::Mat image  
);
```

Параметры:

object – класс служебных данных для распознавания.

image – входное изображение в формате OpenCV (8-битное 3-канальное);

Функция ProcessANRCR возвращает структуру OUT_ANRCR:

```
struct OUT_ANRCR  
{  
    int all_objects;  
    NUMBER_ANRCR objects[max_numbers];  
};
```

где **all_objects** – общее количество найденных номеров (пока не больше 1), **objects** – указатель на массив выделенных структур NUMBER_ANRCR, размер массива взят с запасом max_numbers = 20.

Формат NUMBER_ANRCR:

```
struct NUMBER_ANRCR  
{  
    char number[max_characters + 1];  
    cv::Rect rect_number;  
    float reliability;  
};
```

где **number** – возвращаемый номер (размер массива взят с запасом max_characters = 20;), **rect_number** – ограничивающий прямоугольник номера на изображении, **reliability** – оценочное значение достоверности распознавания номера. reliability – это сумма вероятностей правильного определения символа (принимает значения от 0.3 до 1.0).

ProcessFileANRCR

Функция поиска 8-мизначного номера железнодорожного вагона на изображении не установленного формата, переданного в виде массива char.

```
OUT_ANRCR ProcessFileANRCR  
(  
    iANRCRObject object,  
    char* image_data,  
    int size_image  
);
```

Параметры:

object – класс служебных данных для распознавания.

image_data – указатель на массив данных изображения типа char.

size_image – размер массива image_data.

Функция ProcessFileANRCR возвращает структуру OUT_ANRCR.

InitANRCR

Инициализация класса служебных данных для распознавания.

```
iANRCRObject InitANRCR  
(  
    INIT_ANRCR init_data  
);
```

Параметры:

init_data – входная структура для инициализации класса служебных данных.

Возвращает iANRCRObject объект со служебными данными для распознавания.

Описание структуры INIT_ANRCR:

```
struct INIT_ANRCR  
{  
    bool only_8_symbols,  
    bool memory_sum,  
    unsigned char memory_sum_interval,  
    unsigned char min_frames,  
    bool control_number,  
    char key[101]  
};
```

Параметры:

only_8_symbols – true будут возвращаться номера длиной не менее 8 символов, иначе любой длины.

memory_sum – суммирование результатов из разных кадров. Суммирование позволяет повысить достоверность определения номера, отображая номер, распознаваемый на нескольких кадрах.

memory_sum_interval – интервал кадров, на котором осуществляется суммирование.

min_frames – минимальное количество кадров, где распознан номер.

control_number – нужно ли контролировать номер контрольной цифрой. Контрольная цифра(восьмая) – это цифра, дополняющая под разрядную сумму до ближайшего целого десятка. Программа проверяет таким образом правильность формирования номера.

key – лицензионный ключ.

GetSumNumberANRCR

Суммирование результатов распознавания. Суммирование позволяет повысить достоверность определения номера, отображая номер, распознаваемый на нескольких кадрах.

```
int GetSumNumberANRCR
(
    iANRCRObject object,
    char* buf
);
```

Параметры:

object – объект со служебными данными, получаемый функцией InitANRCR.

buf – возвращаемое значение - указатель на массив символов, для записи номера по итогам суммирования (один номер).

Возвращаемое функцией значение – целое число. Если ноль, значит функция выполнена успешно и в buf есть результат, отличное от нуля значение означает, что в buf результата нет.

DeleteANRCR

Очистка памяти от объекта, созданного функцией InitANRCR.

```
void DeleteANRCR
(
    iANRCRObject* object
);
```

Параметры:

object – указатель на объект.

DrawResultANRCR

Функция отрисовки результатов распознавания на изображении.

```
void DrawResultANRCR  
(  
    cv::Mat image,  
    OUT_ANRCR result,  
    char* sum_text = 0  
);
```

Параметры:

image – изображение.

result – структура возвращаемая функциями ProcessANRCR и ProcessFileANRCR.

sum_text – указатель на массив с номером, который возвращает функция GetSumNumberANRCR.

2. Инсталляция и использование

Перед инсталляцией необходимо убедиться, что Ваша видеокарта поддерживает технологию CUDA.

Рекомендуется видеокарта от nvidia, модель не ниже 1060GTX с 6 ГБ оперативной памяти.

Во время использования библиотечных функций распознавания рекомендуется соблюдать отношение сторон передаваемых на распознавание изображений в пределах 16:9. При этом рекомендуется передавать на распознавание изображение, в котором отношение ширины целевых символов к ширине всей картинке как можно ближе к 0,24.

Для того, чтобы SDK заработало, на компьютер необходимо установить:

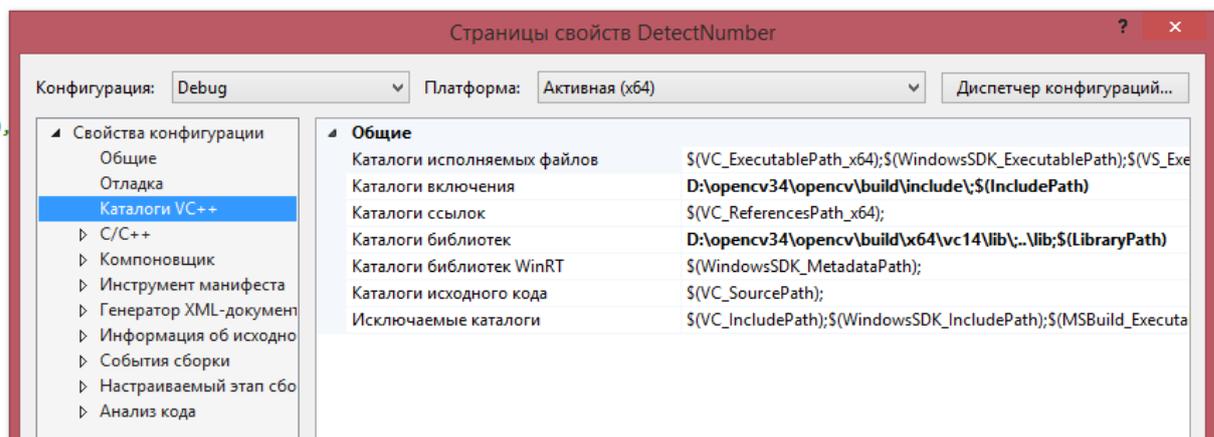
Для Microsoft Visual C++ 2015 Redistributable Package
<https://www.microsoft.com/ru-RU/download/details.aspx?id=52685>

SDK откомпилировано для OpenCV 3.4.7

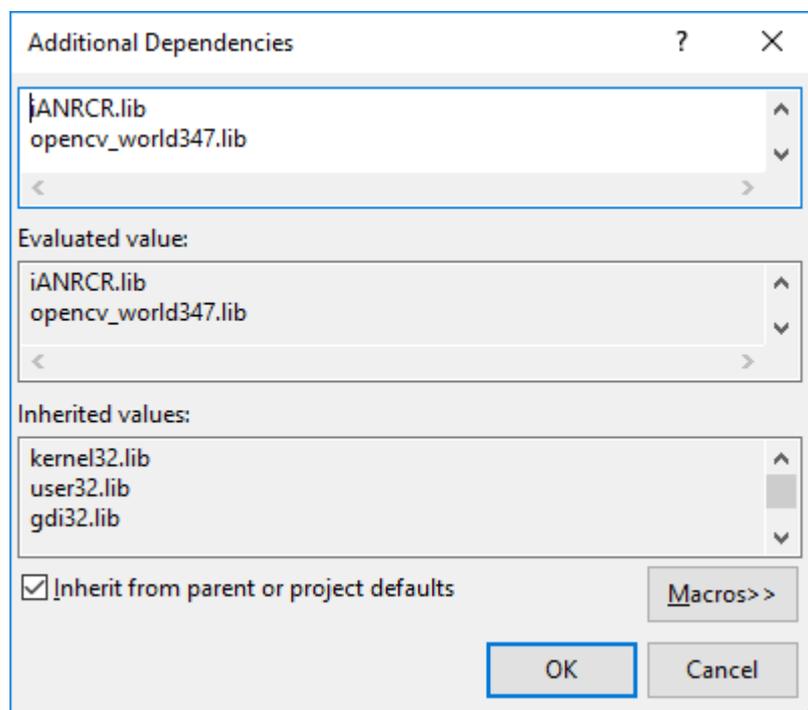
Хотя нужные библиотеки OpenCV идут вместе с iANPR SDK, но для разработки программ вам вероятно понадобится подключение заголовочных файлов. В iANPR использовались библиотеки, откомпилированные vs14 (2015).

Далее осуществляете подключение в виде обычных динамических библиотек.

Если вы реализуете проект на C/C++ на Visual Studio, то пропишите пути до h и lib для OpenCV и места, где находится iANPR:



Добавьте подключаемые библиотеки (в свойствах Компоновщика):



После этого обратите внимание, чтобы все dll из папки x86 или x64 находились или в папке с вашим исполняемым файлом, или в папке, прописанной в переменной PATH.

SDK использует с_api библиотеки tensorflow, для работы SDK tensorflow.dll так же должна быть доступна.

Для правильной работы tensorflow необходима библиотека cudnn 7.5. Файл cudnn64_7.dll будет в сборке SDK.

3. Примеры на C/C++ для Windows

iANRCRImageVideoProcess

Пример показывает работу с изображениями и видео

Параметры запуска:

```
cmd> iANRCRImageVideoProcess.exe [ -image | -image_file  
| -video ] [<path>]
```

Аргументы:

-image – стандартное распознавание изображения.

-image_file – распознавание изображения с помощью функции ProcessFileANRCR.

-video – распознавание видеофайла.

<path> - путь к файлам источникам.

Пример команды запуска:

```
cmd> iANRCRImageVideoProcess.exe -image ../img/src.jpg
```

В режиме распознавания видеофайла появится окно с отображением хода распознавания. По завершении процесса в папке с программой появится видеофайл out.avi с результатами распознавания отрисованными на исходном видеофайле.

После запуска приложения в режиме распознавания изображения, после работы приложения в папке с программой появится файл out.png с результатами распознавания (рис 3.1.1).

Детектированный номер будет обведен рамкой, а результат его распознавания написан в левом нижнем углу изображения.



Рисунок 3.1.1 – Пример результата работы программы в режиме распознавания изображения.

Если в режиме работы с видеофайлом захватываемая картинка превышает 1024 пикселя в ширину, то из такой картинки в центре будет вырезан фрагмент шириной 1024 пикселя с сохранением пропорций соответствующих исходной картинке. И уже вырезанный фрагмент будет передан на распознавание в библиотеку.