

iANRCR C++

Введение

Назначение: распознавание железнодорожных номеров вагонов.
Версия: 1.01
Платформа: Windows 10, 11

Внешние зависимости

Необходимые пакеты и ссылки на них (ссылки с течением времени могут меняться)
1. Visual Studio 2019 redist (распространяемый пакет):
https://aka.ms/vs/17/release/vc_redist.x64.exe
2. OpenCV 4.7:
https://github.com/opencv/opencv/releases/download/4.7.0/opencv-4.7.0-windows.exe
3. CUDA 11.7:
https://developer.download.nvidia.com/compute/cuda/11.7.0/local_installers/cuda_11.7.0_516.01_windows.exe
4. cuDNN 8.0(Вы должны согласиться с лицензией):
https://developer.nvidia.com/compute/machine-learning/cudnn/secure/8.0.5/11.1_20201106/cudnn-11.1-windows-x64-v8.0.5.39.zip
5. TensorRT 8.5.3.1 (Вы должны согласиться с лицензией):
https://developer.nvidia.com/downloads/compute/machine-learning/tensorrt/secure/8.5.3/zip/TensorRT-8.5.3.1.Windows10.x86_64.cuda-11.8.cudnn8.6.zip

Доступные структуры

Описаны в заголовочном файле iANRCR.h

iANRCRSetting

Структура описывает настройки распознавания. Значения по умолчанию определены заранее и их может быть достаточно.

```
struct iANRCRSetting
{
	float maxDistanceBetweenCharactersW = 2.0f;
	float maxDistanceBetweenCharactersH = 0.8f;
	int symbolsInNumber = 8;
	bool correctNumber = true;
	int memoryNumberFrames = 4;
	int memoryNumberRepeat = 2;
	bool debug = 0;
};
```

Параметры:
maxDistanceBetweenCharactersW – максимальное расстояние между символами по горизонтали в высотах, здесь и далее единицей высоты считается высота текущего символа, по умолчанию значение равно 2.0;
maxDistanceBetweenCharactersH – максимальное расстояние между символами по вертикали в высотах, по умолчанию значение равно 0.8;
symbolsInNumber – количество символов в номере, равно 8, но если это не РЖЖ номер, то можно попробовать и другое;

correctNumber – использовать корректировочный символ РЖД для проверки правильности номера, если не правилен, то возвращаться не будет, по умолчанию используется;
memoryNumberFrames – количество запоминаемых в памяти кадров (при batchSize > 1, считается что каждое изображение в пакете – это отдельный канал, и запоминается для каждого канала), по умолчанию равно 4;
memoryNumberRepeat – количество повторений одного и того же номера в кадре, при котором считается, что номер распознан правильно, по умолчанию равно 2;
debug – использовать отладочный режим, который выводит отладочные файлы detection0.jpg, где 0 – номер канала (Рис 1), по умолчанию false.

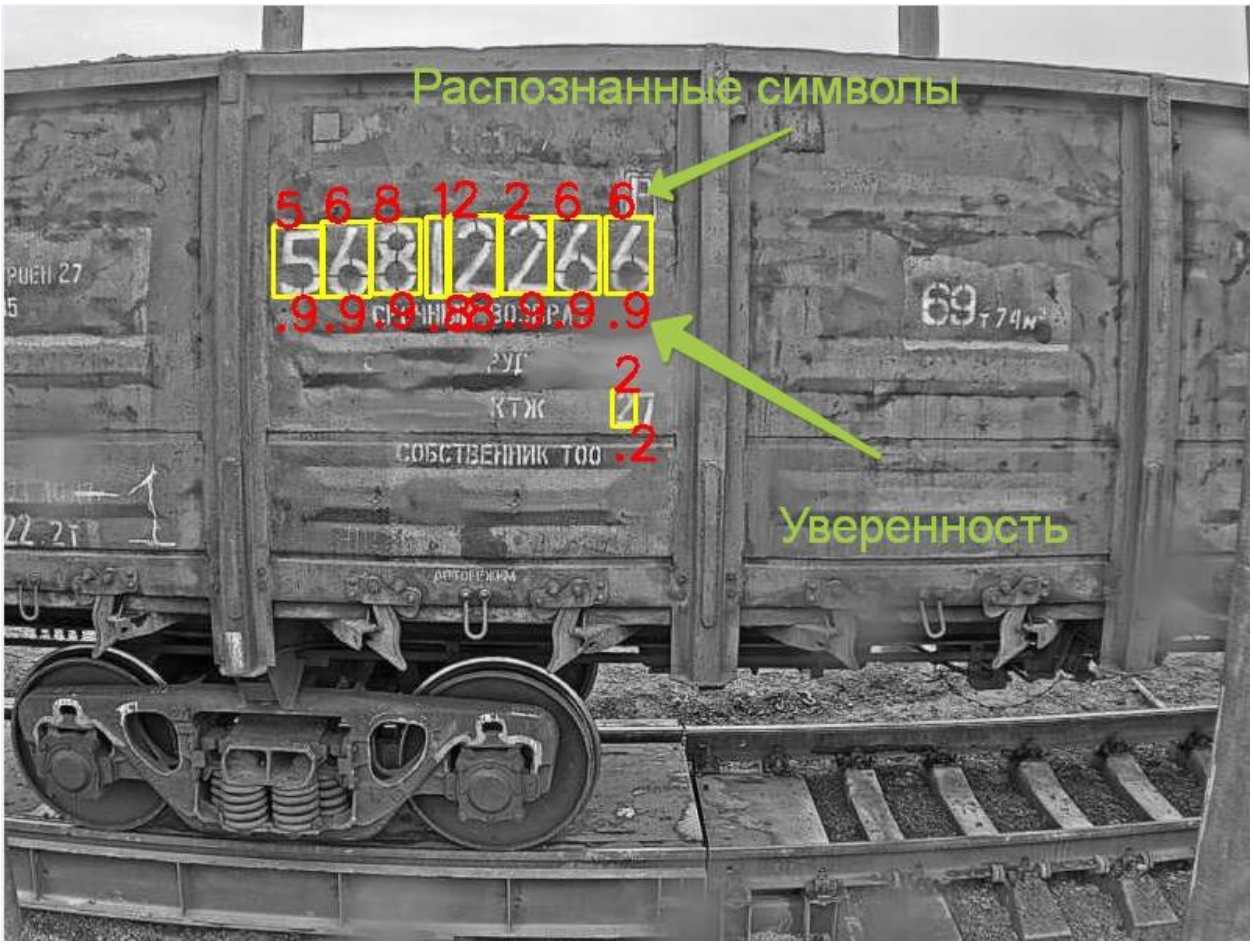


Рис. 1. Пример отладочного изображения с разъяснением

NumberiANRCR
Структура, описывающая возвращаемый номер.

```
struct NumberiANRCR
{
	cv::Rect rect;
	char stringResult[40];
	float confidence[40];
	float score;
};
```

Параметры:
rect – область детектированного номера;
stringResult – результат (цифры);
confidence – вероятность (уверенность) каждой цифры;
score – вероятность всего номера.

NumberiANRCRMem

Структура, описывающая возвращаемый номер из памяти.

```
struct NumberiANRCRMem
{
	char stringResult[40];
};
```

На настоящий момент содержит в себе только параметр номера.

Доступные функции

Описаны в заголовочном файле iANRCR.h

iANRCRInit

Инициализация объекта iANRCRObject, включает в себя загрузку и инициализацию TensorRT движка.

```
iANRCRObject iANRCRInit(
	char* engineFile,
	int batchSize,
	int* error = 0
);
```

Аргументы:

engineFile – путь до движка TensorRT;
batchSize – размер пакетов (движок TensorRT должен соответствовать размеру);
error – информация об ошибке, если нужна.

Функция возвращает объект iANRCRObject или NULL в случае ошибки. Для просмотра кода ошибки необходимо передать указатель на целое число error.

iANRCRRelease

Удаление объекта iANRCRObject, включая удаление движка TensorRT из памяти.

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

Аргументы:

object – указатель на объект iANRCRObject.

iANRCRReleaseP

Удаление объекта iANRCRObject, включая удаление движка TensorRT из памяти. Без двойного указателя для удобства других языков.

```
void iANRCRRelease(
	iANRCRObject object
);
```

Аргументы:

object – объект iANRCRObject.

iANRCRSettings

Установить настройки распознавания.

```
bool iANRCRSettings(
	iANRCRObject object,
	iANRCRSetting settings
```

```
);
```

Аргументы:

object – объект iANRCRObject;
settings – настройки, заданные структурой iANRCRSetting.

Функция возвращает true если настройки удалось применить, и false, если не удалось (object ==NULL).

iANRCRSettingsJSON

Установить настройки распознавания через буфер, содержащий JSON.

```
bool iANRCRSettingsJSON(
	iANRCRObject object,
	char* json
);
```

Аргументы:

object – объект iANRCRObject;
json – настройки, заданные в JSON формате. Пример задания:

```
{
	"maxDistanceBetweenCharactersW":2.0,
	"maxDistanceBetweenCharactersH": 0.8,
	"symbolsInNumber": 8,
	"correctNumber":1,
	"memoryNumberFrames":4,
	"memoryNumberRepeat":2,
	"debug":0,
}.
```

Этот текст JSON представлен в строке. correctNumber и debug могут принимать значения 0 или 1.

Функция возвращает true если настройки удалось применить, и false, если не удалось (object ==NULL).

iANRCRProcess

Передать пакет изображений на распознавание и распознать номера.

```
int iANRCRProcess(
	iANRCRObject object,
	std::vector<cv::Mat> images
);
```

Аргументы:

object – объект iANRCRObject;
images – вектор 24 битных изображений (8UC3).

Возвращает 0 (iANRCRErrors::IA_OK) при удачной выполненной операции, в противном случае – код ошибки.

iANRCRAddImage

Альтернатива функции **iANRCRProcess**. Добавляет изображение в формате байт-кода в буфер. После этого нужно вызвать функцию **iANRCRinference**.

```
int iANRCRAddImage (
	iANRCRObject object,
	const char* image_bytes,
	long size_image
);
```

Аргументы:

object – объект iANRCRObject;
image_bytes – изображение в байтах в формате BMP, JPEG, PNG или TIFF;
size_image – размер этого изображения.

Возвращает 0 (iANRCRErrors::IA_OK) при удачной выполненной операции, в противном случае – код ошибки.

iANRCRinference

Альтернатива функции **iANRCRProcess**. Запускает распознавание изображения(-ий) добавленных ранее функцией **iANRCRAddImage**.

```
int iANRCRinference(
        iANRCRObject object
);
```

Аргументы:

object – объект iANRCRObject;

Возвращает 0 (iANRCRErrors::IA_OK) при удачной выполненной операции, в противном случае – код ошибки.

iANRCRGetResult

Получить результат распознавания текущего кадра.

```
std::vector<std::vector<NumberiANRCR>> iANRCRGetResult(
        iANRCRObject object
);
```

Аргументы:

object – объект iANRCRObject;

Возвращает вектор векторов номеров (может быть несколько номеров в кадре).

iANRCRGetResultJSON

Получить результат распознавания текущего кадра в формате JSON.

```
const char* iANRCRGetResultJSON(
        iANRCRObject object
);
```

Аргументы:

object – объект iANRCRObject;

Возвращает пустую строку, если результатов распознавания нет. В противном случае возвращает JSON в следующем формате:

```
{
    "images":[
        [
            {
                "x":172,
                "y":134,
                "width":243,
                "height":53,
                "string":56812266,
                "score":0.908945
            }
        ]
    ]
}
```

images – ключ к последовательности результатов из нескольких изображений (ее длина 1 при batchSize == 1, если используется пакетное распознавание, например, batchSize == 4, то длина равна 4). Для каждого изображения может быть формально несколько номеров (обычно 1), и если несколько, то секции {} перечисляются через запятую. Внутри каждой секции информации о местоположении номера, строка и достоверность.

iANRCRGetResultMem

Получить результат распознавания при объединении результатов в памяти.

```
std::vector<std::vector<NumberiANRCRMem>> iANRCRGetResultMem(
        iANRCRObject object
);
```

Аргументы:

object – объект iANRCRObject;

Возвращает вектор векторов номеров (может быть несколько номеров в кадре).

iANRCRGetResultMemJSON

Получить результат распознавания при объединении результатов в памяти в формате JSON.

```
const char* iANRCRGetResultMemJSON(
        iANRCRObject object
);
```

Аргументы:

object – объект iANRCRObject;

Возвращает пустую строку, если результатов распознавания нет. В противном случае возвращает JSON в следующем формате:

```
{
    "numbers":[
        [
            {
                "string":"64295462"
             }
        ]
     ]
}
```

Возвращаемые ошибки

Описаны в заголовочном файле **iANRCRErrors.h**.

Табл.1. Ошибки

IA_NULL = -1	Пустой объект iANRCRObject
IA_OK = 0	Операция успешна
IA_TRTENGINELOADERROR = 1	Невозможно загрузить движок TensorRT
IA_COPYTOGPUMEMERROR = 2	Ошибка копирования в память GPU
IA_TRTINFERENCEFAIL = 3	Ошибка при распознавании (инференс) изображений
IA_TRTGETFROMGPUERROR = 4	Ошибка получения результатов распознавания с GPU на CPU
IA_NOIMAGES = 5	Нет изображений для распознавания
IA_IMAGENOT8U =6	Изображения не соответствуют типу 24 бита.
IA_ADDIMGBUFFERFULL = 7	Возвращается для **iANRCRAddImage**, если буфер уже внутри равен batchSize
IA_ADDIMGCANNTDECODE = 8	Возвращается для **iANRCRAddImage**, если изображение не удалось декодировать
IA_ADDIMGNOTEQUALBATCH = 9	Возвращается для **iANRCRinference**, если изображений в буфере меньше batchSize.

Конвертация в движок TensorRT

1. Установить (распаковать) TensorRT-8.5.3.1 для 11x версий CUDA

2. Переместить или скопировать в TensorRT-8.5.3.1\bin\ папку dll из TensorRT-8.5.3.1\lib\

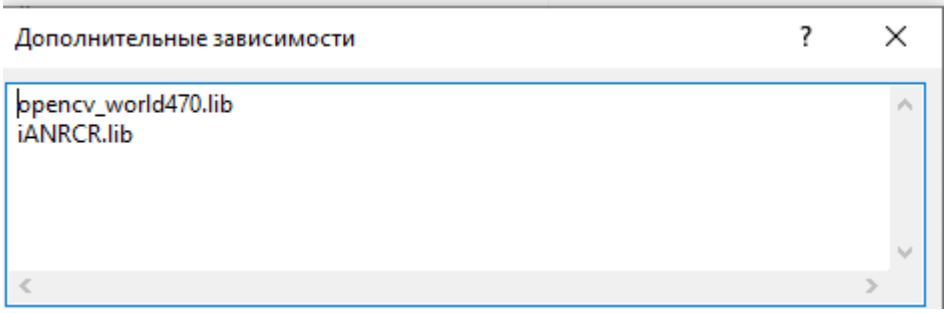
3. Запустить конвертацию

trtexec --onnx=rr_i1024_b32_yxm.onnx --workspace=3000 --fp16 --saveEngine=rr_i1024_b32_yxm.engine

4. Если не будет запускаться, значит не видит соответствующих CUDA и cuDNN dll - нужно или расшарить в PATH или скопировать библиотеки непосредственно в TensorRT-8.5.3.1\bin\

Подключение библиотеки в программе

Работает только в режиме x64. В Visual Studio C++ проекте подключать так:



Все примеры на использование представлены в утилите iartest. Её запуск, примеры.

Одиночное изображение:
iartest.exe -i data\2080\rr_i1024_b32_yxm_20x.engine imagetest1.jpg

Видео как есть:
iartest.exe -v data\2080\rr_i1024_b32_yxm_20x.engine D:\data_rr\rrvagon\20190913_121006_CAM_2.avi 1

Видео левая половина кадра:
iartest.exe -v data\2080\rr_i1024_b32_yxm_20x.engine F:\datasets\vagons\20220726121713.mp4 2

Видео правая половина кадра:
iartest.exe -v data\2080\rr_i1024_b32_yxm_20x.engine D:\iANRCR3\vlc-record-2022-11-25-11h00m59s-rtsp___192.168.1.108_554_cam_realmonitor-.mp4 3

Тест производительности batSize == 1:
iartest.exe -p1 data\2080\rr_i1024_b32_yxm_20x.engine

Тест производительности batSize == 4:
iartest.exe -p4 data\2080\rr_i1024_b32_yxmb4_20x.engine

Тест папки с изображениями:
iartest.exe -d data\2080\rr_i1024_b32_yxm_20x.engine D:\data_rr\rrvagon\1 D:\iANRCR3\iANRCRCPP\bin\outyx

Рекомендации и особенности распознавания

- Модель обучена на изображениях 1024x1024, поэтому на распознавание нужно передавать примерно квадратное изображение. Т.е. оно конечно само меняется к нужному размеру, но если у вас изображение сильно вытянуто по горизонтали, то лучше на распознавание передавать половину кадра – цель снизить искажение цифр.
- По умолчанию стоит параметр коррекции correctNumber = true, поэтому если вы не уверены, что эти номера с коррекцией или вы хотите получать абсолютно все номера и сами потом как-то объединять, то нужно отключать.
- Используя debug, можно посмотреть причины проблем на изображениях и собрать потом дополнительную выборку для решения подобных проблем путем переобучения моделей с нашей стороны.

Возможные внешние ошибки и сообщения

Ошибка CUDA инициализации 35.

```
CUDA initialization failure with error: 35
```

Сведение: старый драйвер для используемой версии CUDA:
Решение: обновить драйвер

Сообщение CUDA LAZY в Windows:

```
CUDA lazy loading is not enabled. Enabling it can significantly reduce device memory usage.
```

Сведения: выбран неоптимальный режим загрузки движков в память устройства
Решение: Можно оставить, как есть. Либо установить переменную окружения CUDA_MODULE_LOADING равным LAZY: