

AIPro sNet Solution

Demo Guide

(API Version)

Part I. Demo 프로그램 설치 가이드

Part II. sNet Solution 사용 가이드

System Requirement & Dependency

항목	내용
포함 모델	Super Resolution
사용 언어	C/C++
OS	Windows 11
설치환경	Visual Studio 2022 CUDA 11.6.2 cuDNN 8.4.0
Demo Dependency	OpenCV-4.5.5(자체 포함), TensorRT-8.4.2.4 (자체 포함)
GPU 최소 사양	RTX 2070 이상 필수

작성일: 2022 년 9 월

작성자: 박 천 수

이메일: cspek@skku.edu

Part I. Demo 프로그램 설치 가이드

1. Visual Studio 2022 설치

- A. Community 버전(무료) 설치 가능
- B. (중요)CUDA 와 cuDNN 설치전에 Visual Studio 를 먼저 설치해야 함

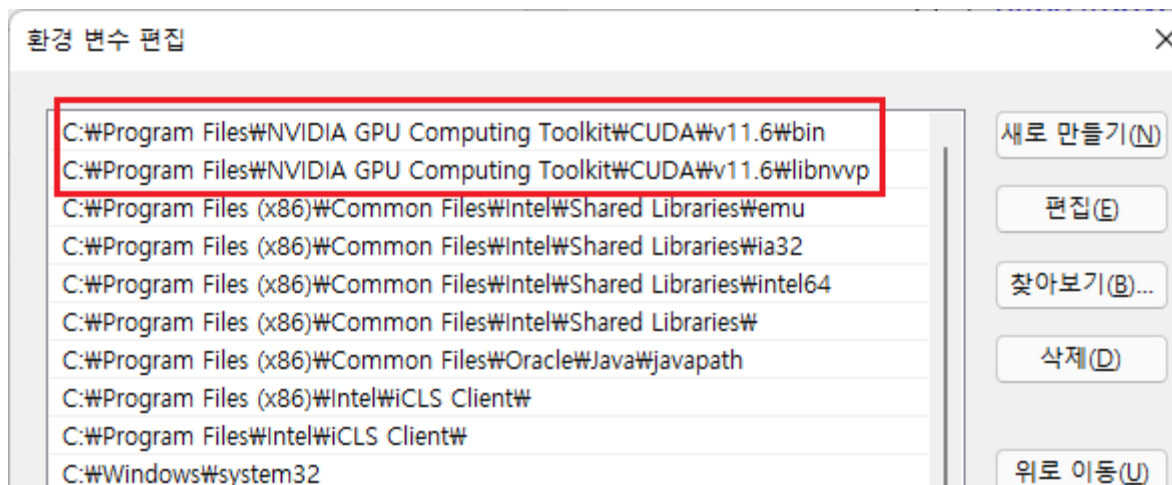
2. CUDA 설치

A. CUDA Toolkit 11.6.2 설치

- 링크: <https://developer.nvidia.com/cuda-toolkit-archive>
- 설치 파일: cuda_11.6.2_511.65_windows.exe
- 설치 시 기본 경로 사용 권장

B. PATH 환경 변수에 아래 내용 추가

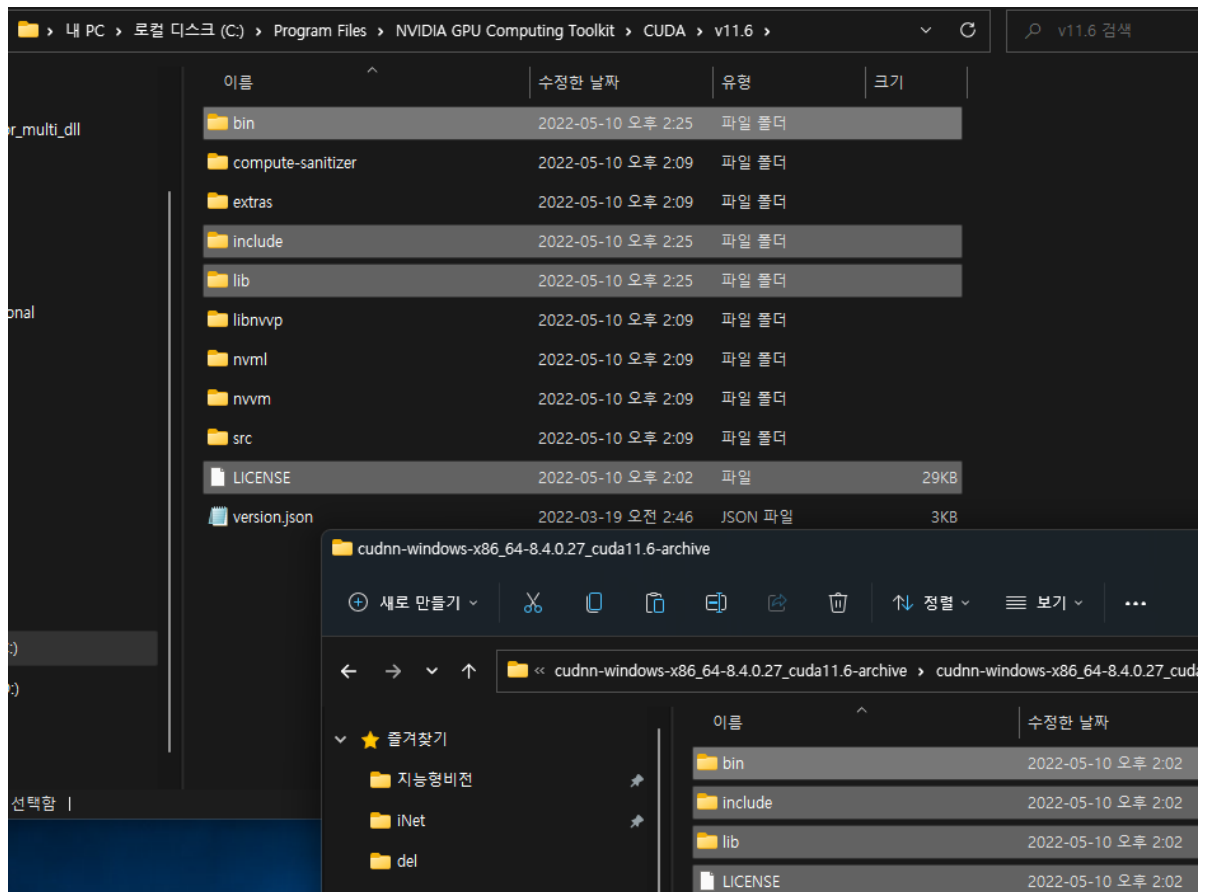
- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.6\bin
- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.6\libnvvp



3. cuDNN 설치

A. cuDNN 8.4.2 설치

- 설치링크: <https://developer.nvidia.com/rdp/cudnn-download>
- 설치 파일: cudnn-windows-x86_64-8.4.0.27_cuda11.6.exe
 - 다운 받기위해 NVIDIA 회원 가입이 필요
- 압축을 풀고 bin, include, lib 폴더를 “C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.6” 위치에 덮어쓰는 방식으로 복사



B. zlibwapi.dll 설치

- cudnn 8.4 는 zlib 의 zlibwapi.dll 을 내부적으로 이용
 - zlibwapi.dll 이 설치가 안된 경우 “Could not locate zlibwapi.dll. Please make sure it is in your library path!” 에러 메시지가 출력됨
- 아래 페이지에서 ZLIB_DLL 링크를 통해 zlib123dllx64.zip 을 다운
 - <https://docs.nvidia.com/deeplearning/cudnn/install-guide/index.html#install-zlib-windows>

3.1.3. Installing zlib

zlib is a data compression software library that is needed by cuDNN.

Procedure

1. Download and extract the zlib package from **ZLIB DLL**. Users with a 32-bit machine should downl

Note: If using Chrome, the file may not automatically download. If this happens, right-click the l

2. Add the directory path of **zlibwapi.dll** to the environment variable PATH.

- zlib123dllx64.zip 파일 압축을 풀고 dll_x64 위치의 zlibwapi.dll 파일을 “C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.6\bin” 위치에 복사

4. Visual Studio 2022 설정

A. sNet-API-Demo Repository 클론 또는 다운로드

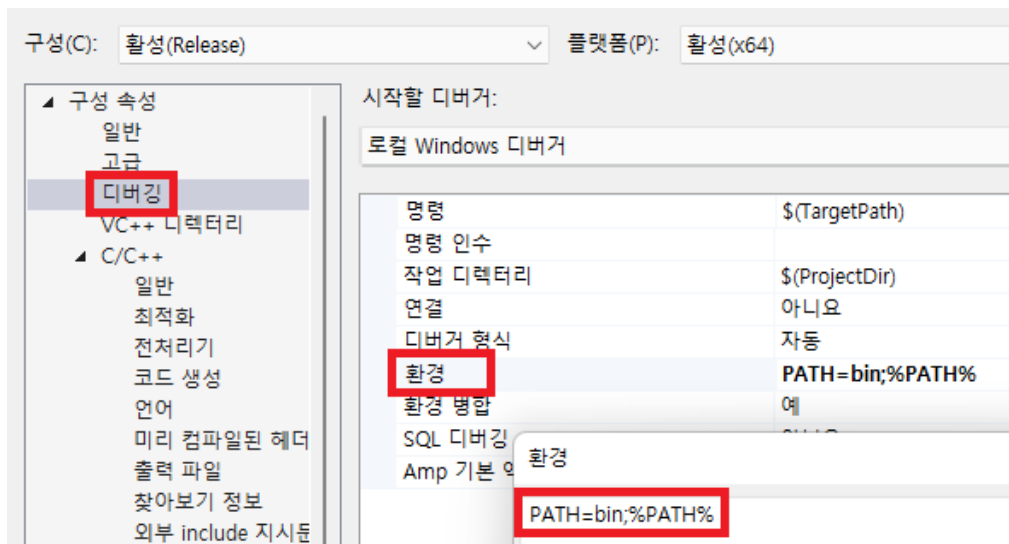
- Address: <https://github.com/AIProCo/sNet-API-Demo>

B. sNet-API-Demo.sln 솔루션을 열고 프로젝트 실행 환경 설정

- 프로젝트 속성 페이지에서 구성을 "Release", 플랫폼을 "x64"로 선택
- 실행할 때(디버깅 포함)도 "Release", "x64"로 선택해야 함(Debug 모드 호환 안됨)

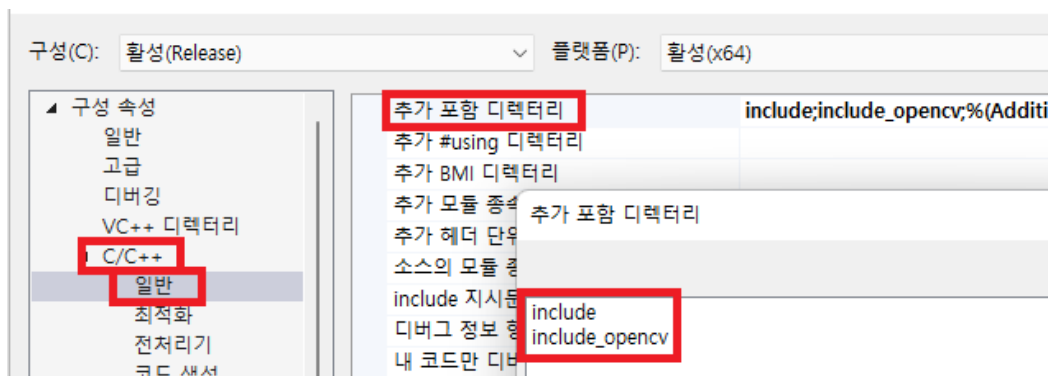
C. 솔루션 실행을 위한 로컬 PATH 변수 수정(시스템 PATH 변수에 영향 없음)

- 이동 경로: 디버깅 → 환경
- PATH 변수에 "bin" 폴더와 %PATH% 입력(해당 위치 dll 를 읽을 수 있게 됨)
 - 입력 예시: PATH=bin;%PATH%



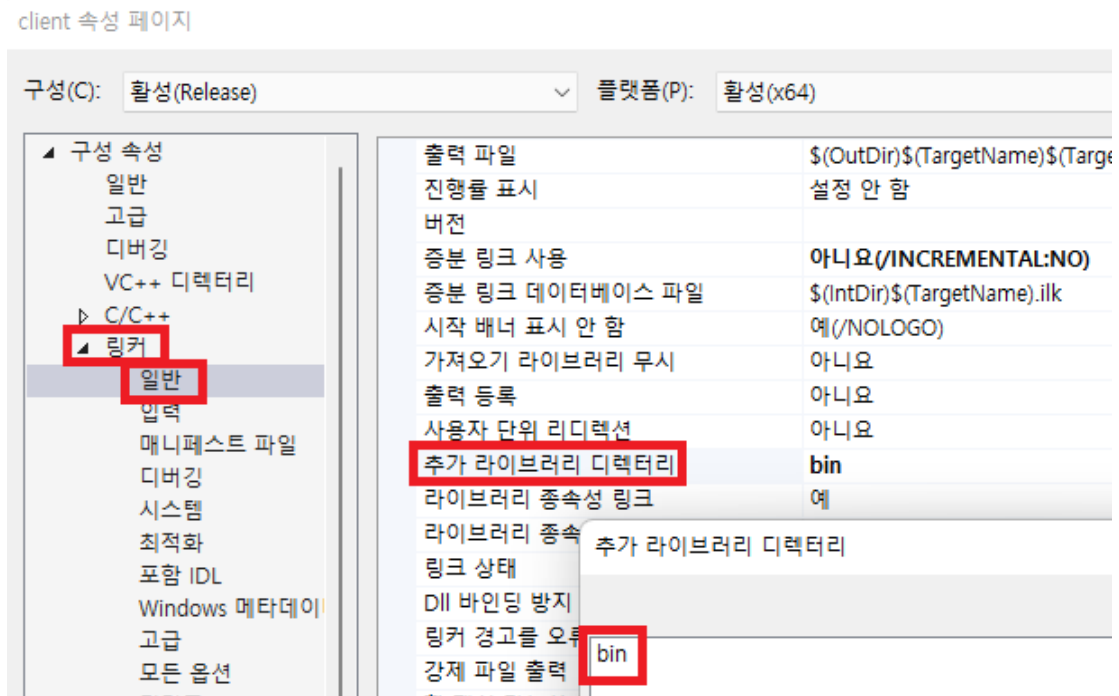
D. 추가 포함 디렉터리 입력

- 이동 경로: C/C++ → 일반 → 추가 포함 디렉터리
- 추가 포함 디렉터리에 "include"와 "include_opencv" 입력



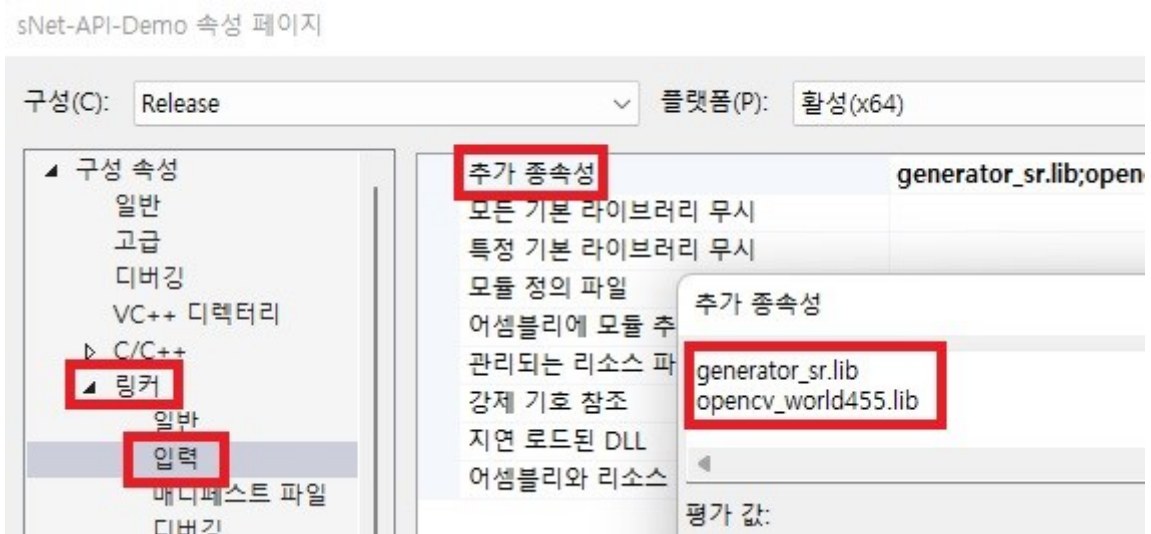
E. 추가 라이브러리 디렉터리 입력

- 이동 경로: 링커 → 일반 → 추가 라이브러리 디렉터리
- 추가 라이브러리 디렉터리에 “bin” 입력



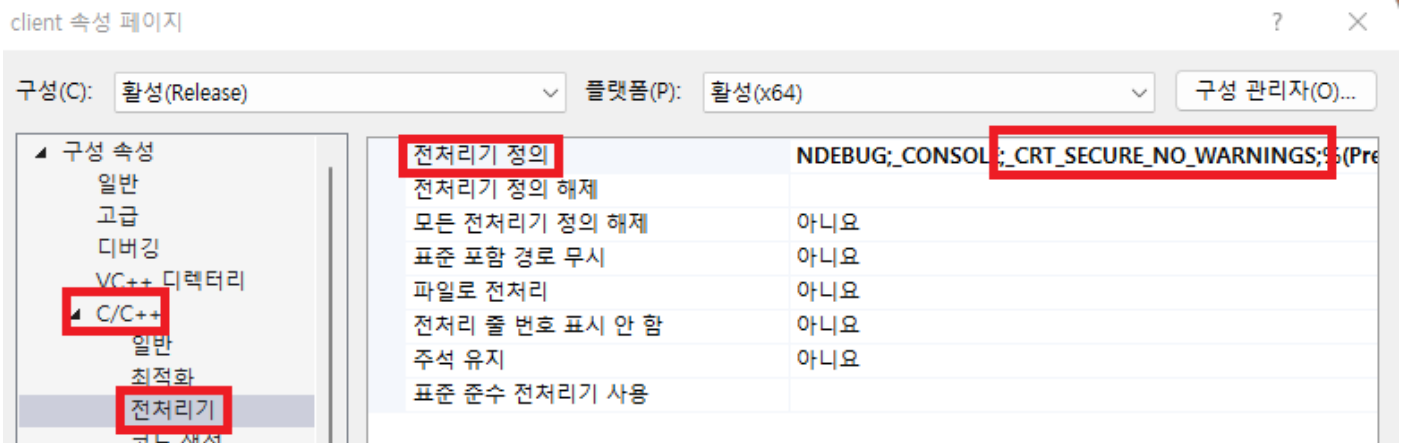
F. 추가 종속성 입력

- 이동 경로: 링커 → 입력 → 추가 종속성
- 추가 종속성에 “generator_sr.lib”와 “opencv_world455.lib” 입력



G. 보안 에러 Disable

- MS 보안 함수 강제 사용 Disable
 - (참고!) 에러 메시지: error C4996: 'localtime': This function or variable may be unsafe.
- 이동 경로: C/C++ → 전처리기 → 전처리기 정의
- 전처리기 정의에 “_CRT_SECURE_NO_WARNINGS” 입력



5. 필요 파일 설치

A. “bin, inputs, videos(sNet).zip” 파일 다운로드 및 압축 해제. bin, inputs, videos 디렉토리를 솔루션 디렉토리(*.sln 파일과 같은 위치)로 복사

- 링크: https://drive.google.com/file/d/1RVNGcMH7q_dwcqnZNK6a1niKbUfCJp9i/view?usp=sharing
- 링크에 문제가 있는 경우 github repository 의 readme 문서 참고

B. 각 디렉토리의 파일 확인

- bin
 - generator_sr.dll, generator_sr.lib, nvinfer.dll, nvinfer_plugin.dll, opencv_world455.dll, opencv_world455.lib
- inputs
 - aipro_sr_x1_5_1_2.net, aipro_sr_x2_1_2.net, config.json
- videos
 - hd.mp4, qhd0.mp4, qhd1.mp4

6. Release 모드와 x64 플랫폼을 설정 후 실행

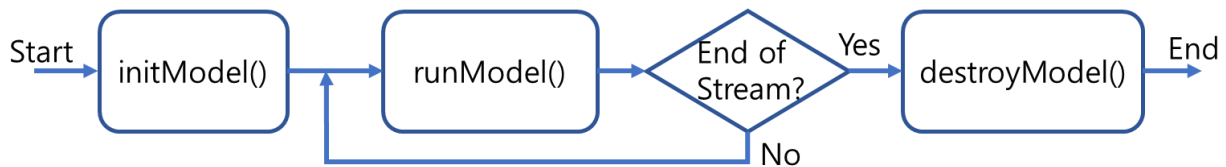
Part II. sNet Solution 사용 가이드

1. Solution 동작

- AIPro sNet Solution 실행은 크게 초기화, 실행, 소멸 3 단계로 이루어짐. 각 단계에 해당하는 함수와 세부 내용은 아래와 같음

단계	함수	내용
초기화	initModel()	• 솔루션 실행에 필요한 모델과 내부 저장소를 초기화
실행	runModel()	• frame 배치를 입력(frames) 받고 SR 추론 동작 수행 • 고해상도 변환 영상 배치(srFrames)를 출력
소멸	destroyModel()	• 생성된 모델과 저장소 공간 해제

- 초기화와 소멸 단계는 프로그램 시작시와 종료시에 각 1 번씩 호출되고, 실행 단계는 frame 배치를 구성 후 runModel()을 반복적으로 호출하며 실행



<그림> sNet 솔루션 동작 플로우

2. 솔루션을 이용한 프로그램 개발 작업 내용

- 기본적으로 sNet 솔루션은 config.json 파일을 Parsing 해 Config 객체(cfg)를 생성하고 이를 이용해 전체 솔루션을 동작 시킴. 솔루션을 이용한 프로그램을 개발하기 위해서는 cfg 객체를 초기화하는 parseConfigAPI() 함수 내용을 각 응용 프로그램에 맞게 수정해야 함
 - parseConfigAPI() 함수 내부의 상수 값은 변경하지 않는 것을 권장

- 미리 설정된 json 파일과 응용 프로그램 동작시 추출되는 데이터를 이용해 응용 프로그램에 맞는 cfg 객체를 생성 후 예제 코드와 같은 방식으로 초기화, 실행, 소멸 동작을 수행해야 함

3. API 함수 설명

bool initModel(**Config** &cfg)

Initialize model

- param cfg configuration struct
- return initialization result(true: success, false: fail)

bool runModel(**vector**<**Mat**> &frames, **vector**<**Mat**> &srFrames, **vector**<**int**> &vchIDs)

Run the SR model for a frame batch

- param frames batch of frames
- param srFrames batch of output SR frames
- param vchIDs vchIDs of batched frames
- return run result(true: success, false: fail)

bool destroyModel()

Destroy model

- param None
 - return flag for destruction result(true: success, false: fail)
-

4. config.json 파일 설정

구분	Name	Value
global	apikey	솔루션 사용 키(aiprotest 고정)
	frame_limit	처리할 최대 프레임 수
	input_files	비디오 입력(경로 포함)
	output_files	SR 비디오 출력(경로 포함)
	filter_enable	filter 기반 출력 Enable - Bilinear filter 방식을 이용해 입력 frame 을 upsample 하고 결과 영상을 저장 - 입력파일 이름에 “_filter” 가 추가된 파일 생성
sr	job	SR 변환 동작. - “SR_X2”: 가로 세로 2 배 확대 - “SR_X1_5”: 가로 세로 1.5 배 확대

- Demo 프로그램의 SR 변환 출력 영상 해상도는 FHD(1920x1080)로 고정. 입력 영상의 해상도와 SR 변환 동작 종류를 출력 영상 해상도를 고려해 설정해야 함
 - SR_X2 동작의 경우 입력 영상 해상도가 960x540, SR_X1_5 동작의 경우 입력 영상 해상도가 1280x720 이 되어 최적임

5. 평균 추론 동작 시간

- runModel() 함수 평균 시간 복잡도 측정
 - GPU: 2080Ti, CPU: i9-10900X@3.70GHz, Batch Size: 4 frames
 - SR_X2: 38ms/frame
 - SR_X1_5: 55ms/frame