AIPro Inc. Technical Documentation (Confidential)

AIPro iNet Solution Demo Guide (API Version)

Part I. Demo Program Installation

Part II. iNet Solution Guide

System Requirement & Dependency

Category	Content
Models	Object Detection, Object Tracking, Person Attribute Recognition, Pose Estimation, Action Recognition
Program Language	C/C++
OS	Windows 11
Environment	Visual Studio 2022 CUDA 11.6.2 cuDNN 8.4.0
Demo Dependency	OpenCV-4.5.5(included), TensorRT-8.4.2.4 (included)
GPU	RTX 2070 or newer

Date: Sept. 2022

Author: Chun-Su Park

E-mail: cspk@skku.edu

Part I. Demo Program Installation

1. Visual Studio 2022 Installation

A. Install Community Version(free)

B. (Important!!) You must install Visual Studio before installing CUDA and cuDNN

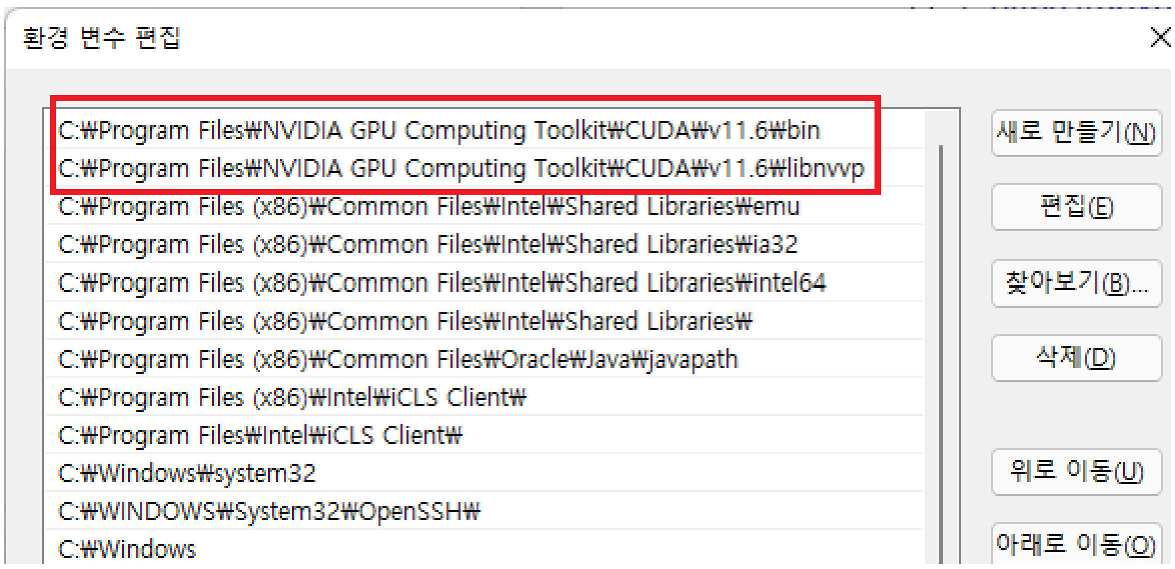
2. CUDA Installation

A. Install CUDA Toolkit 11.6.2

- Link: https://developer.nvidia.com/cuda-toolkit-archive
- File name: cuda_11.6.2_511.65_windows.exe
- Use default path

B. Add the followings to PATH

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



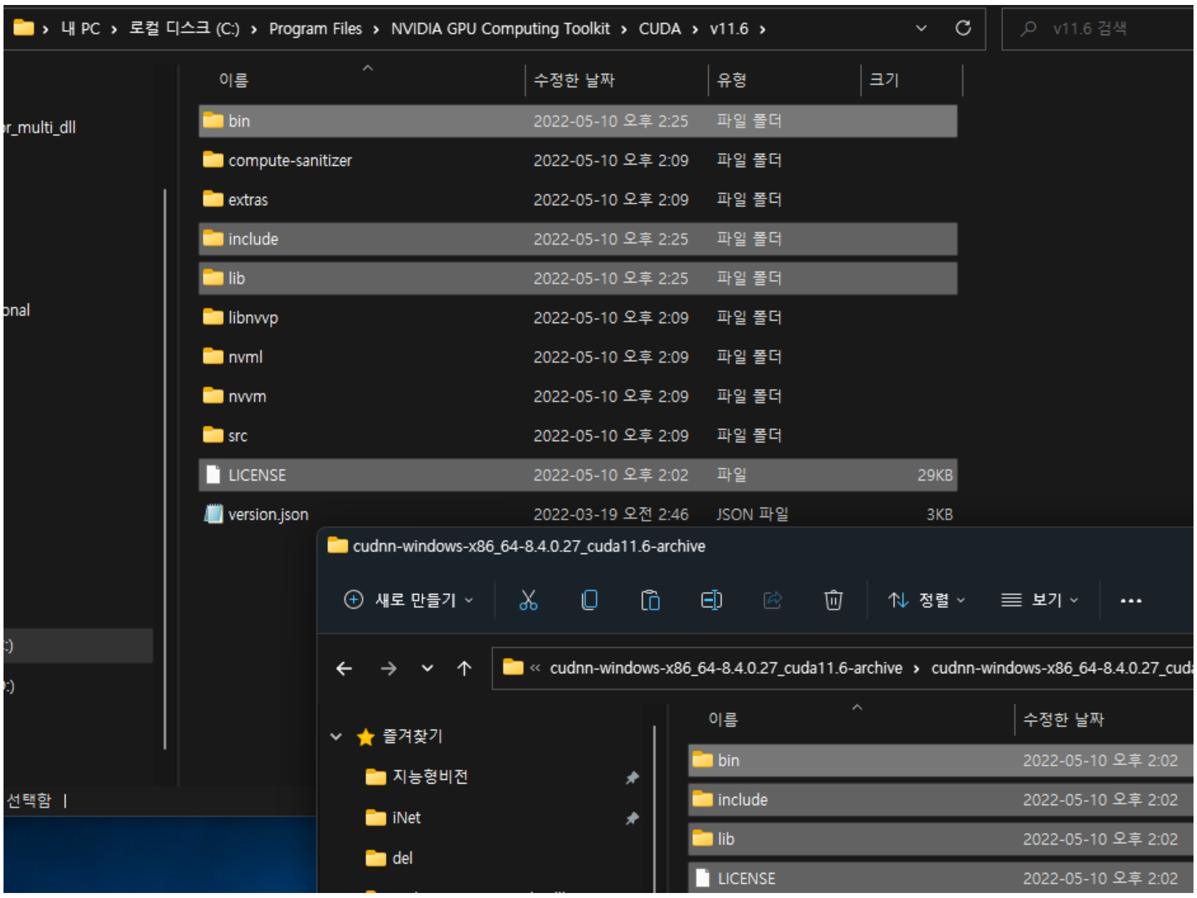
```
C:₩Program Files₩NVIDIA GPU Computing Toolkit₩CUDA₩v11.6₩bin
C:₩Program Files₩NVIDIA GPU Computing Toolkit₩CUDA₩v11.6₩libnvvp
C:₩Program Files (x86)₩Common Files₩Intel₩Shared Libraries₩emu
C:₩Program Files (x86)₩Common Files₩Intel₩Shared Libraries₩ia32
C:₩Program Files (x86)₩Common Files₩Intel₩Shared Libraries₩intel64
C:₩Program Files (x86)₩Common Files₩Intel₩Shared Libraries₩
C:₩Program Files (x86)₩Common Files₩Oracle₩Java₩javapath
C:₩Program Files (x86)₩Intel₩iCLS Client₩
C:₩Program Files₩Intel₩iCLS Client₩
C:₩Windows₩system32
C:₩WINDOWS₩System32₩OpenSSH₩
C:₩Windows
```

3. cuDNN Installation

A. Install cuDNN 8.4.2

- Link: https://developer.nvidia.com/rdp/cudnn-download
- File name: cudnn-windows-x86_64-8.4.0.27_cuda11.6.exe
  - You need to sign up to NVIDIA to download cuDNN

- Unzip the downloaded file. Then, copy and paste bin, include, and lib directories to “C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.6”


내 PC › 로컬 디스크 (C:) › Program Files › NVIDIA GPU Computing Toolkit › CUDA › v11.6 ›
v11.6 검색
r_multi_dll
onal
선택함
새로 만들기
정렬
보기
즐겨찾기
지능형비전
iNet
del

이름	수정한 날짜	유형	크기
bin	2022-05-10 오후 2:25	파일 폴더	
compute-sanitizer	2022-05-10 오후 2:09	파일 폴더	
extras	2022-05-10 오후 2:09	파일 폴더	
include	2022-05-10 오후 2:25	파일 폴더	
lib	2022-05-10 오후 2:25	파일 폴더	
libnvvp	2022-05-10 오후 2:09	파일 폴더	
nvml	2022-05-10 오후 2:09	파일 폴더	
nvvm	2022-05-10 오후 2:09	파일 폴더	
src	2022-05-10 오후 2:09	파일 폴더	
LICENSE	2022-05-10 오후 2:02	파일	29KB
version.json	2022-03-19 오전 2:46	JSON 파일	3KB

cudnn-windows-x86_64-8.4.0.27_cuda11.6-archive

« cudnn-windows-x86_64-8.4.0.27_cuda11.6-archive › cudnn-windows-x86_64-8.4.0.27_cud

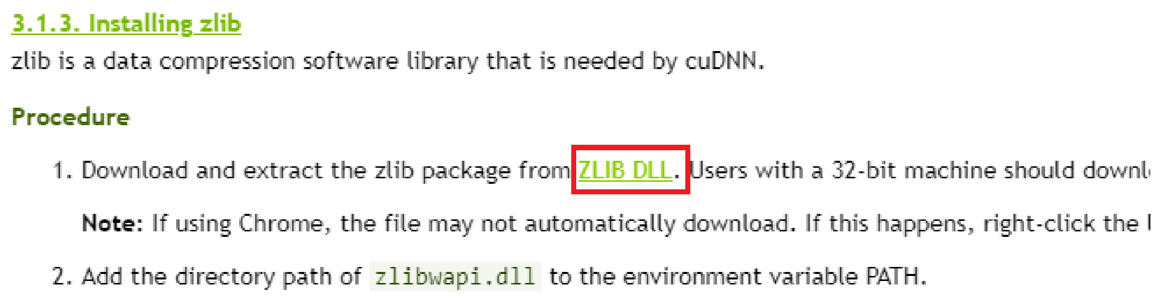
이름	수정한 날짜
bin	2022-05-10 오후 2:02
include	2022-05-10 오후 2:02
lib	2022-05-10 오후 2:02
LICENSE	2022-05-10 오후 2:02

B. Install zlibwapi.dll

- cudnn 8.4 uses zlibwapi.dll of zlib internally
  - If zlibwapi.dll is not installed, you can see the following error message

```
“Could not locate zlibwapi.dll. Please make sure it is in your library path!”
```

- Download zlib123dllx64.zip by using the ZLIB_DLL link in the following page
  - 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.

- Unzip zlib123dllx64.zip. Then, copy zlibwapi.dll in the dll_x64 directory to “C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.6\bin”

4. Visual Studio 2022 Setting

A. Clone or download iNet-API-Demo repository

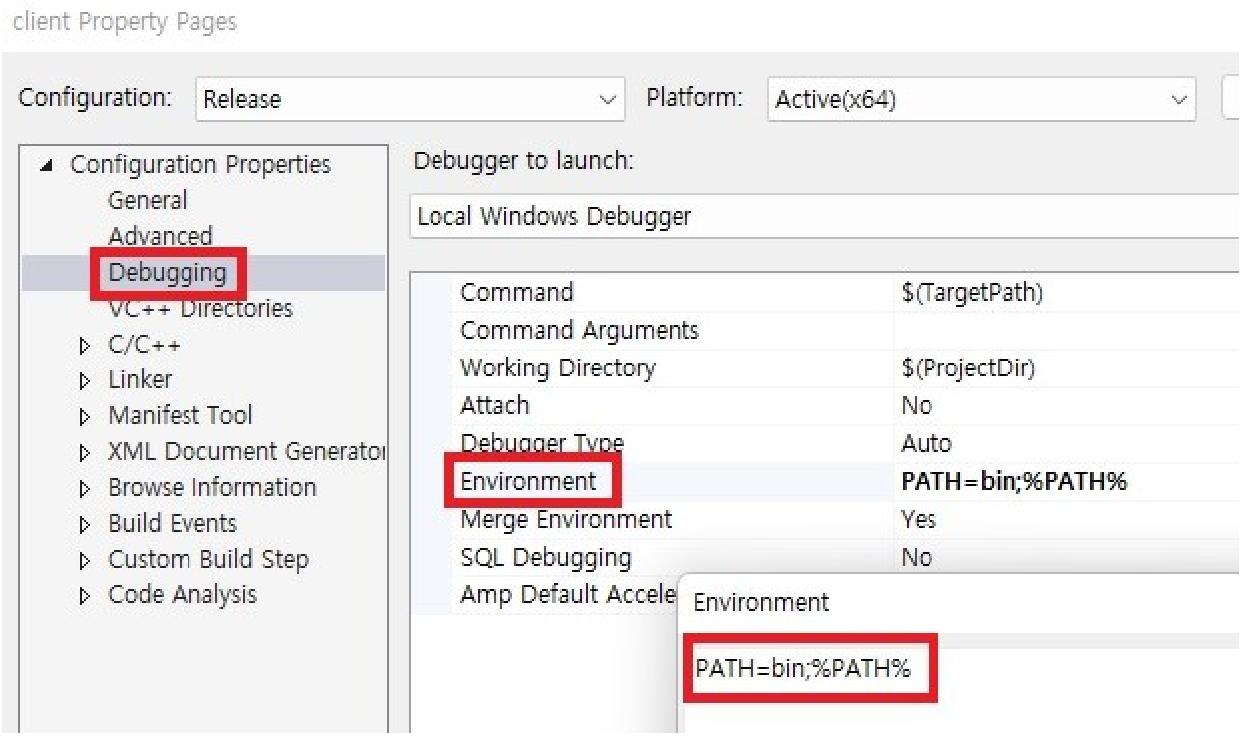
- Repository Address: https://github.com/AIProCo/iNet-API-Demo

B. Open iNet-API-Demo.sln and set environment

- Set Configurations to “Release” and Platforms to “x64”
- Debug configuration is not supported

C. Modify the local PATH variable (the system PATH variable is not affected)

- Path: Properties → Debugging → Environment
- Enter “bin” directory to the PATH variable
  - Example: PATH=bin;%PATH%


client Property Pages

Configuration: Release   Platform: Active(x64)

Debugger to launch: Local Windows Debugger

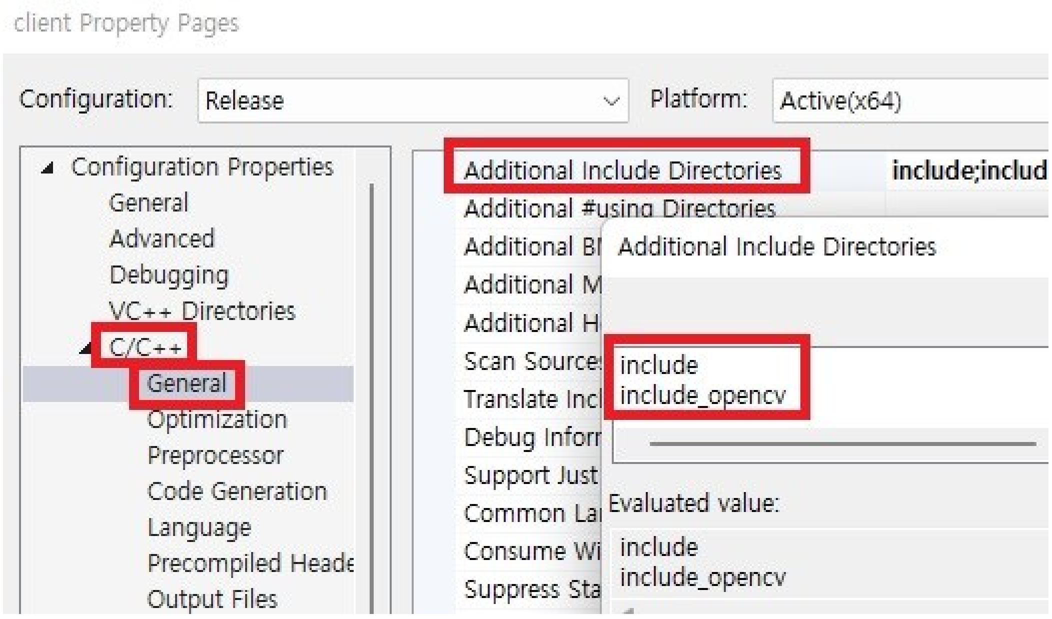
Command	$(TargetPath)
Command Arguments	
Working Directory	$(ProjectDir)
Attach	No
Debugger Type	Auto
Environment	PATH=bin;%PATH%
Merge Environment	Yes
SQL Debugging	No
Amp Default Accele	

Environment

PATH=bin;%PATH%

D. Modify Additional include Directories

- Path: Properties → C/C++ → General → Additional include Directories
- Enter “include” and “include_opencv”


client Property Pages

Configuration: Release Platform: Active(x64)

Configuration Properties
- General
- Advanced
- Debugging
- VC++ Directories
- C/C++
  - General
  - Optimization
  - Preprocessor
  - Code Generation
  - Language
  - Precompiled Heade
  - Output Files

Additional Include Directories | include;includ
Additional #using Directories
Additional Bl
Additional M
Additional H
Scan Source
Translate Incl
Debug Inforr
Support Just
Common La
Consume Wi
Suppress Sta

Additional Include Directories

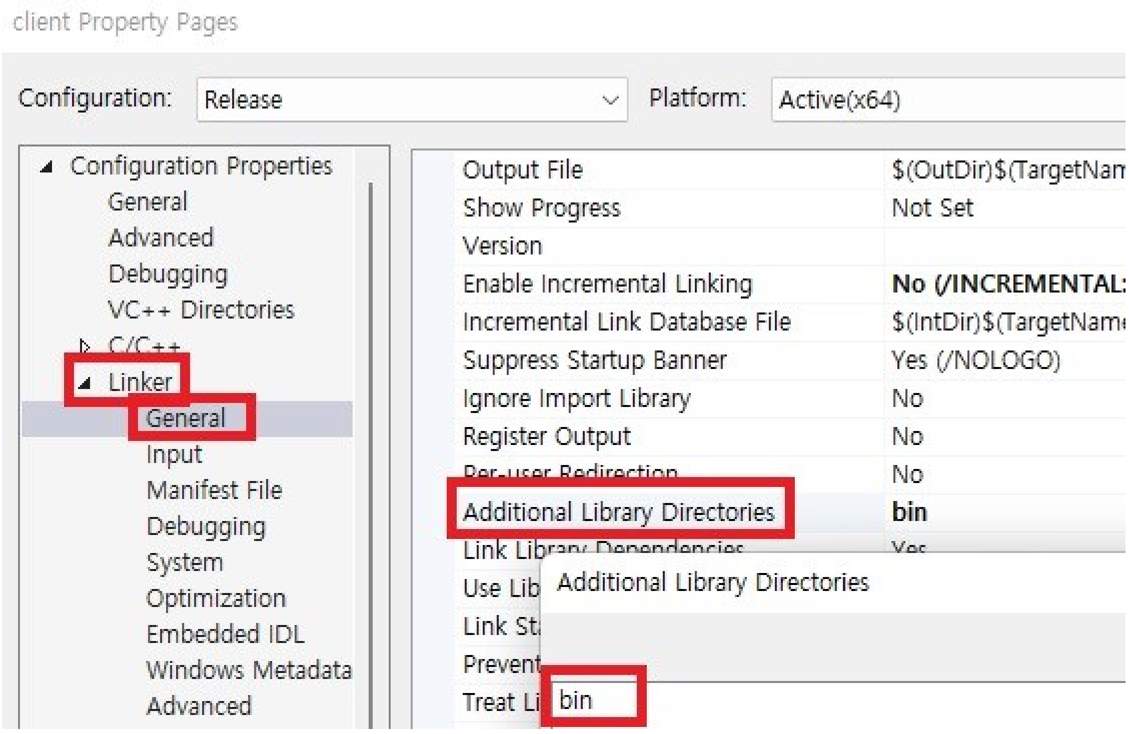
```
include
include_opencv
```

Evaluated value:

```
include
include_opencv
```

E. Modify Additional Library Directories

- Path: Properties → Linker → General → Additional Library Directories
- Enter “bin”


client Property Pages

Configuration: Release Platform: Active(x64)

Configuration Properties
- General
- Advanced
- Debugging
- VC++ Directories
- C/C++
- Linker
  - General
  - Input
  - Manifest File
  - Debugging
  - System
  - Optimization
  - Embedded IDL
  - Windows Metadata
  - Advanced

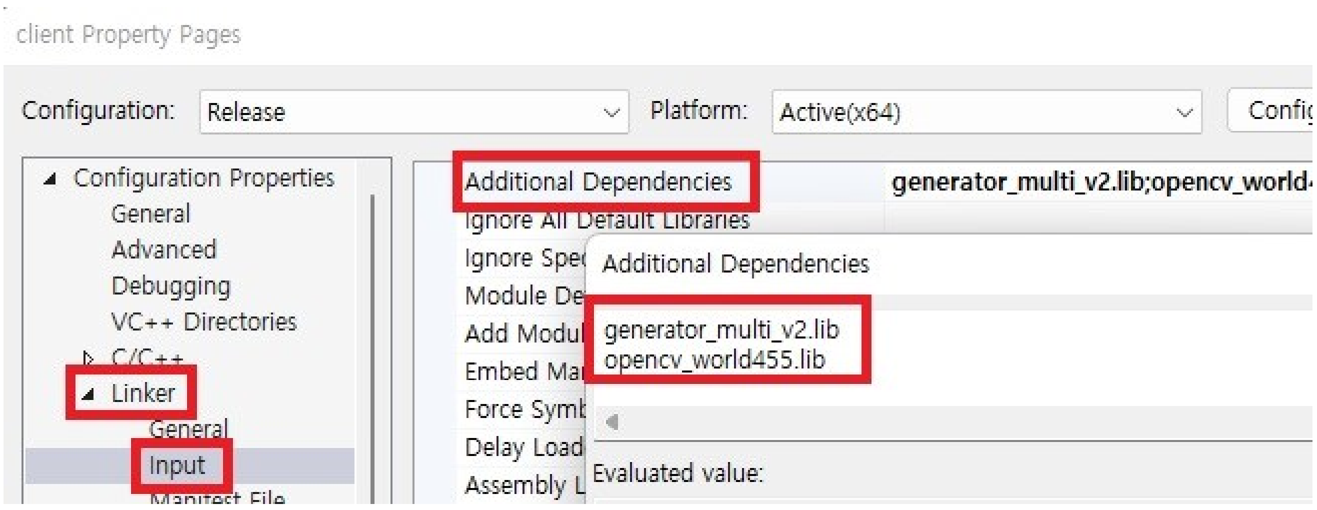
Output File	$(OutDir)$(TargetNam
Show Progress	Not Set
Version	
Enable Incremental Linking	No (/INCREMENTAL:
Incremental Link Database File	$(IntDir)$(TargetName
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	bin
Link Library Dependencies	Yes
Use Lib	
Link St	
Prevent	
Treat Li	

Additional Library Directories

```
bin
```

F. Modify Additional Dependencies

- Path: Properties → Linker → General → Additional Dependencies
- Enter "generator_multi_b.lib" and "opencv_world455.lib"


client Property Pages

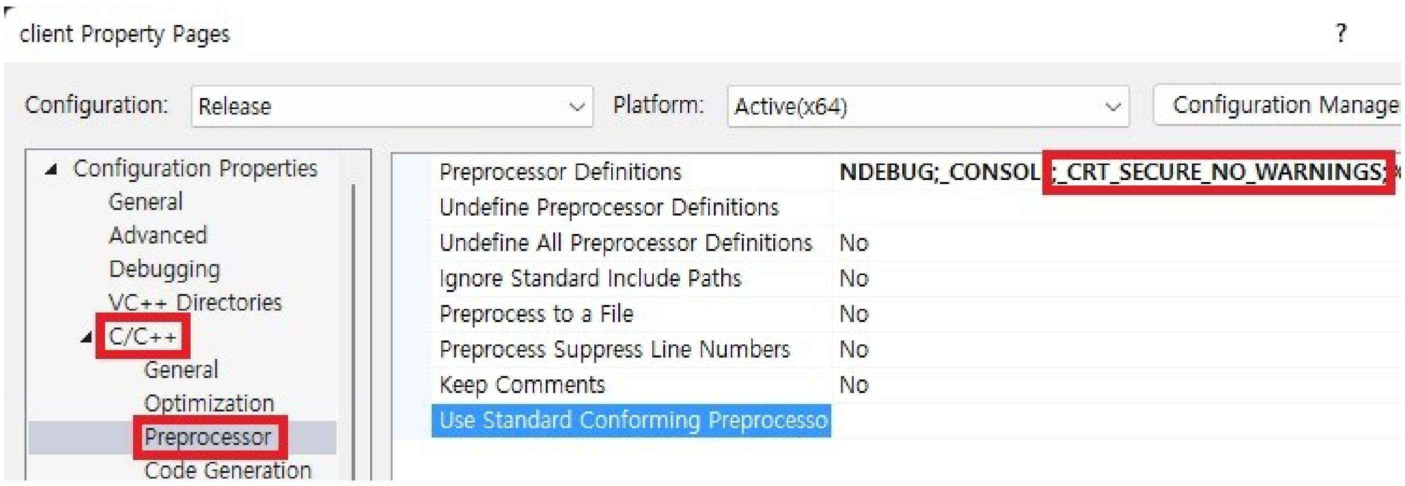
Configuration: Release   Platform: Active(x64)

Additional Dependencies: generator_multi_v2.lib;opencv_world

```
generator_multi_v2.lib
opencv_world455.lib
```

G. Disable security error

- Disable forced MS security functions usage
  - Error Message: error C4996: 'localtime': This function or variable may be unsafe.
- Path: Properties → C/C++ → preprocessor → Preprocessor Definitions
- Enter "_CRT_SECURE_NO_WARNINGS" to Preprocessor Definitions


client Property Pages

Configuration: Release   Platform: Active(x64)

Preprocessor Definitions	NDEBUG;_CONSOL;_CRT_SECURE_NO_WARNINGS;
Undefine Preprocessor Definitions	
Undefine All Preprocessor Definitions	No
Ignore Standard Include Paths	No
Preprocess to a File	No
Preprocess Suppress Line Numbers	No
Keep Comments	No
Use Standard Conforming Preprocesso	

5. Install required directories

A. Download and upzip the followings zip file. Then, copy and paste bin, inputs, and videos directories to the solution directory (the directory including the .sln file):

- Link: https://drive.google.com/file/d/1pDwwVHiY48qUBAHbY52iDCrz_ftb7ugO/view?usp=sharing
- If the link is broken, please refer to the link of the github repository.

6. **Set Release mode and x64 platform. Then, run the solution.**

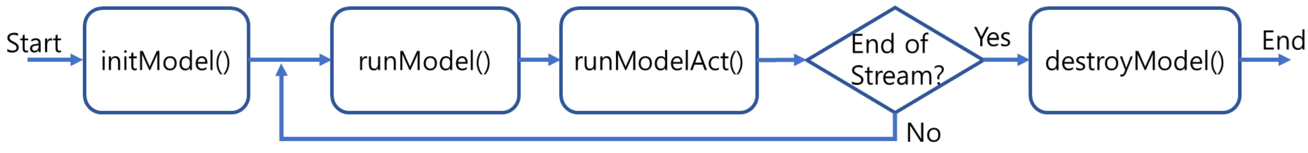
Part II. iNet Solution Developer Guide

1. Solution Introduction

■ AIPro iNet Solution proceeds in four major steps, Initialization, Execution-1, Execution-2, and Destruction. The functions and details corresponding to each step are as follows:

Step	API Function	Content
Initialization	initModel()	− Initialize models and internal memory required to run the solution
Execution-1	runModel()	− Receive a batch of frames and perform inference − Responsible for object detection, tracking, counting (Line & Zone), and PAR (Person attribute recognition) − Fill out detected object boxes, tracking IDs, counting results, and PAR info to the DetBox object
Execution-2	runModelAct()	− Perform inference of Pose and Action models for the detected object boxes − Responsible for detecting pose (Skeleton) and recognizing behavior − Fill out detected Skeleton information and action IDs to the DetBox object
Destruction	destroyModel()	− Destroy models and free memory

■ The initialization and destruction functions are called once at the start and end of the program, respectively. Execution proceeds by repeatedly calling runModel() and runModelAct() for each batch of frames



<Figure> Flowchart of iNet Solution

2. Program development using the iNet solution

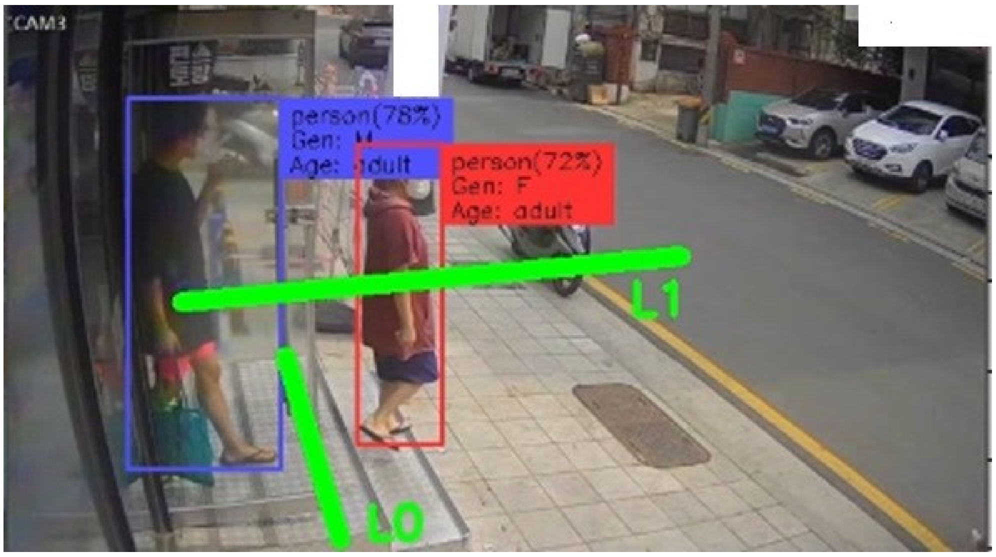
- Basically, the iNet solution parses the config.json file to create a Config object (cfg) and uses it to operate the entire solution. In order to develop a program using the solution, the developer should modify the parseConfigAPI() function depending on each application.
  - It is recommended not to modify constant values in parseConfigAPI()
- After creating a cfg object that fits the application using both the json file and data extracted during application operation, initialization, execution, and destruction steps should be performed in the same way as the example code

3. Counting Information

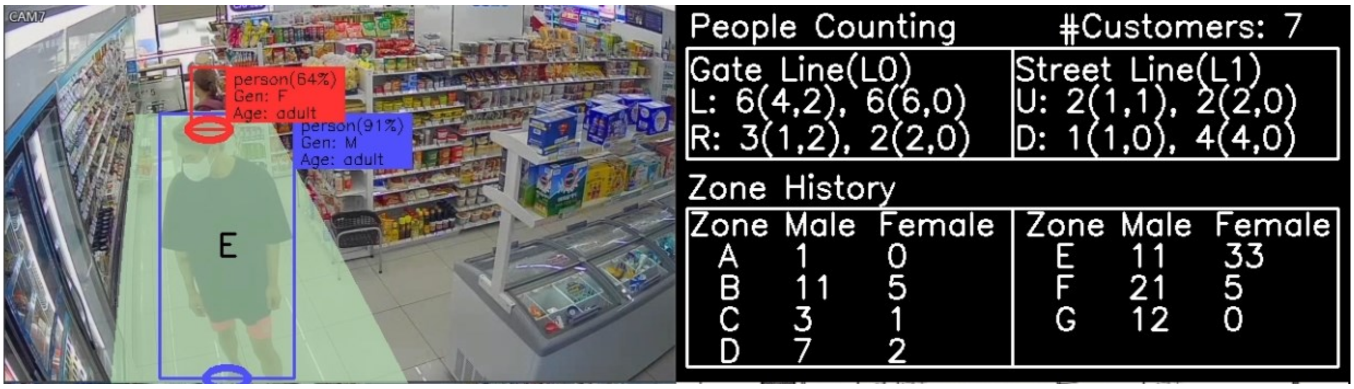
- By default, count the number of people passing through the CntLine and the number of people staying inside the Zone. When counting the number of people, count a total of six cases, considering gender (male and female) and age group (children, adult, and the elder)
  - Each counting information is stored in a 2x3 array in a [male/female][child/adult/elderly] manner and is output in the following format:

```
M_Total(M_Child, M_Adult, M_Elder)   F_Total(F_Child, F_Adult, F_Elder)
```

- Each CntLine object counts the number of people passing in the Up/Down or Left/Right direction, and each Zone stores the number of people currently located inside and the number of people present (counting once a second)
  - CntLine class includes TotalUL[2][3] and TotalDR[2][3] arrays for counting
  - Zone class includes curPeople[2][3] and hitMap[2][3] arrays for hitmap



<Figure> Line Setting Example (L0 and L1)



People Counting #Customers: 7

Gate Line(L0)	Street Line(L1)
L: 6(4,2), 6(6,0)	U: 2(1,1), 2(2,0)
R: 3(1,2), 2(2,0)	D: 1(1,0), 4(4,0)

Zone History

Zone	Male	Female	Zone	Male	Female
A	1	0	E	11	33
B	11	5	F	21	5
C	3	1	G	12	0
D	7	2			

<Figure> Line Counting & Zone Hitmap Example

4. API Functions

bool initModel(Config &cfg)
Initialize model − param cfg configuration struct − return initialization result(true: success, false: fail)

bool runModel(vector<:vector<DetBox>> &dboxesMul, vector<Mat> &frames, vector<int> &vchIDs, vector<int> &frameCnts, float scoreTh, int framesStory, int maxDist)
Run detection and PAR models for a frame batch − param dboxesMul return detected dboxes of all video channels(vchIDs) − param frames batch of frames − param vchIDs vchIDs of batched frames − param scoreTh threshold for filtering low confident detections

- param framesStory the number of False-Negative detections, during which track_id will be kept
- param maxDist max distance in pixels between previous and current detection, to keep the same track_id
- return runModel result(true: success, false: fail)

bool runModelAct(vector<:vector<DetBox>> &dboxesMul, vector<Mat> &frames, vector<int> &vchIDs, vector<int> &frameCnts, vector < vector <DetBox>> dboxesMul)

Run POSE and Action models for the detected dboxesMUL

- param dboxesMul return extracted Skeletons and Actions for the inserted dboxesMul
- param frames batch of frames
- param vchIDs vchIDs of batched frames
- param frameCnts frameCnts of batched frames
- param dboxesMul detected dboxes of batched frames in runModel
- return runModelAct result(true: success, false: fail)

bool destroyModel()

Destroy model

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

bool resetCntLineAndZone(Config &cfg)

Reset CntLine and Zone configuration

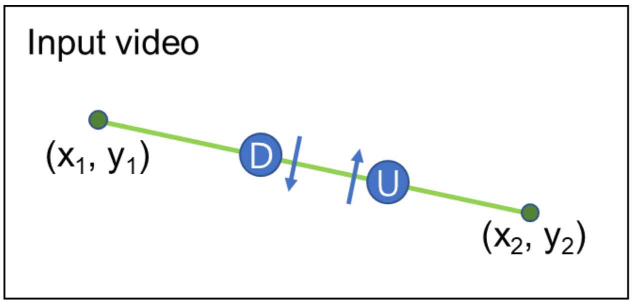
- param None
- return flag for reset(true: success, false: fail)

bool resetRecord()

Reset all records such as zone.curPeople, zone.hitMap, cntLine.totalUL, and cntLine.totalDR.

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

5. Configuration of config.json

Name	Item	Value
global	apikey	Solution key (must use "aiprotest")
	frame_limit	Number of frames to be processed
	input_files	Input video files with path
	output_files	output videos files with path
od	score_th	Score value for object detection
par	enable	PAR inference On/Off
pose	enable	Pose inference On/Off
	score_th	Score value for drawing Skeleton
act	enable	Action inference On/Off
	score_th	Score value for Action recognition
line	param	Enter Counting Line information as follows: [line_id vchID x1 y1 x2 y2] - line_id: unique ID - vchID: Video Channel ID - x1, y1: Point 1 - x2, y2: Point 2  (Counting example) UP: Male 3(1/2/0), Female 2(0/1/1) Down: Male 5(2/2/1), Female 1(0/1/0)
zone	param	Enter Zone information as follows: [zone_id vchID isRestricted x1 y1 x2 y2 x3 y3 x4 y4] - zone_id: unique ID - vchID: Video Channel ID - isRestricted: flag for specifying a restricted area (internally not used) - x1 y1 x2 y2 x3 y3 x4 y4: four vertex coordinates(entered in consecutive directions)

6. Person Attribute Recognition

- In the runModel() function, enter attributes of the PedAtts struct and additional member variables of the DetBox object

Attribute	**Content**
Gender	Male/Female recognition (accuracy: about 93%) - DetBox-PedAtts-atts[0]: 0:Male, 1:Female
Age	Child/Adult/Elder recognition (accuracy: about 85%) - DetBox-PedAtts-atts[1]: confidence to be child - DetBox-PedAtts-atts[2]: confidence to be adult - DetBox-PedAtts-atts[3]: confidence to be elder
Movement	Motion activity (for detecting fainting, falling, sleep, etc) - DetBox-distVar: box center variation after temporal pooling
Time to appear	For calculating the time spent after appear - DetBox-inTime: time when this object is detected
Previous position	For estimating movement path - DetBox-(rxP, ryP): reference position in the previous frame

7. Action Recognition

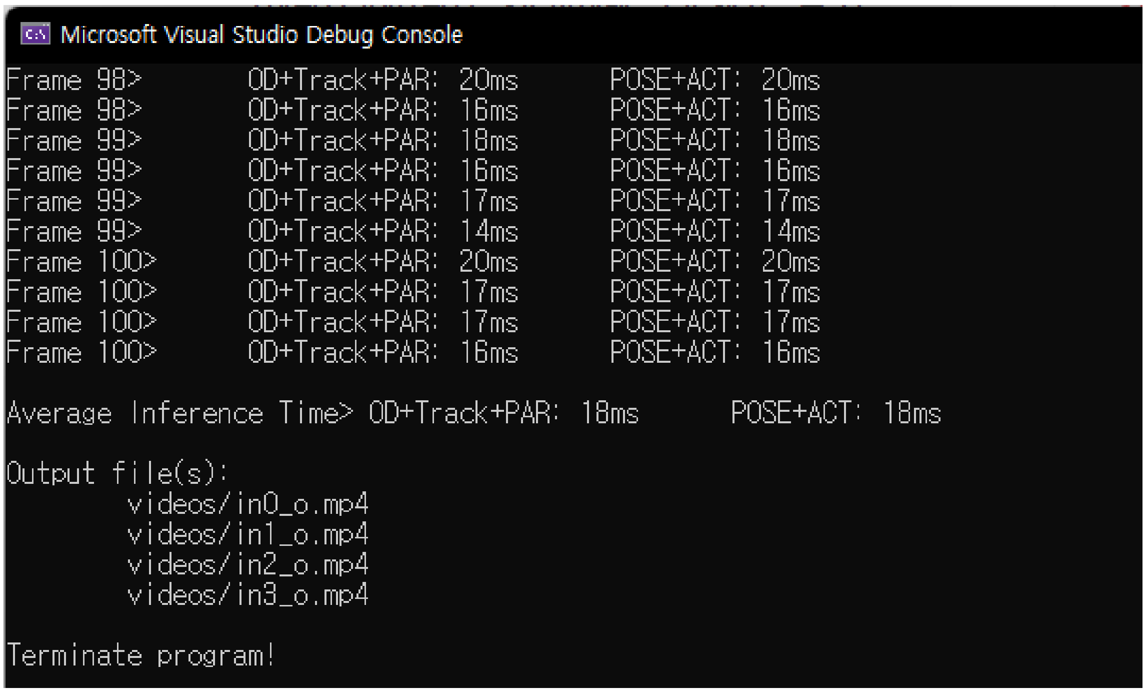
- In the runModelAck() function, enter the action information of the DetBox object
  - DetBox-actID: action ID in actIDMapping
  - DetBox-actConf: confidence of the current act
- Action ID Mapping

17 actions (ID: Label)			
0: Hand on mouth	1: Pick up	2: Throw	3: Sit down
4: Stand up	5: Clapping	6: Reading/writing	7: Hand wave
8: Kick	9: Cross hands	10: Staggering	11: Fall down
12: Punch/Slap	13: Push	14: Walk	15: Squat down
16: Run			

8. Average Inference Time

- Measure time delays of runModel() and runModelAct() functions
  - runModel(): OD + Tracking + PAR
  - runModelAct(): Pose Estimation + Action Recognition
  - Input Video Resolution: FHD, Model Input Resolution: 960x544, GPU: 2080Ti, CPU: i9-10900X@3.70GHz, Batch Size: 1 frame

OD+Track+PAR: 18ms/frame, POSE+ACT: 18ms/frame



```
Microsoft Visual Studio Debug Console
Frame 98>       OD+Track+PAR: 20ms      POSE+ACT: 20ms
Frame 98>       OD+Track+PAR: 16ms      POSE+ACT: 16ms
Frame 99>       OD+Track+PAR: 18ms      POSE+ACT: 18ms
Frame 99>       OD+Track+PAR: 16ms      POSE+ACT: 16ms
Frame 99>       OD+Track+PAR: 17ms      POSE+ACT: 17ms
Frame 99>       OD+Track+PAR: 14ms      POSE+ACT: 14ms
Frame 100>      OD+Track+PAR: 20ms      POSE+ACT: 20ms
Frame 100>      OD+Track+PAR: 17ms      POSE+ACT: 17ms
Frame 100>      OD+Track+PAR: 17ms      POSE+ACT: 17ms
Frame 100>      OD+Track+PAR: 16ms      POSE+ACT: 16ms

Average Inference Time> OD+Track+PAR: 18ms      POSE+ACT: 18ms

Output file(s):
        videos/in0_o.mp4
        videos/in1_o.mp4
        videos/in2_o.mp4
        videos/in3_o.mp4

Terminate program!
```