

PyTorch ONNX + Raspberry Pi + NCS

최병찬

PyTorch ONNX + Raspberry Pi + NCS Workflow

[관련 Reference]

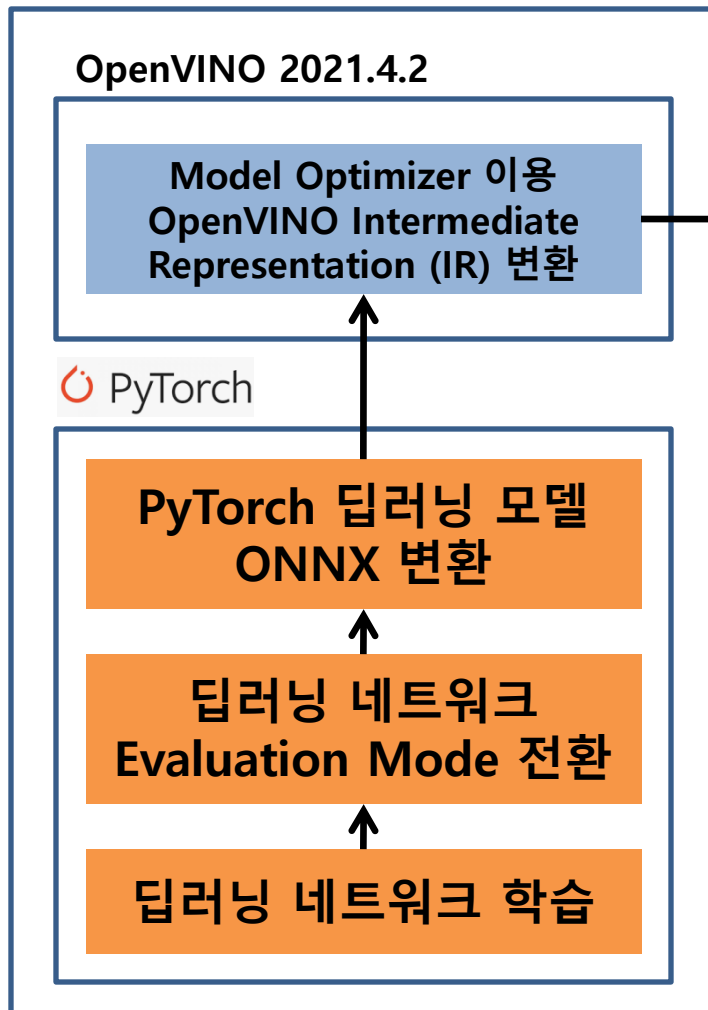
- Conversion of PyTorch Classification Models and Launch with OpenCV Python :
https://docs.opencv.org/4.x/dc/d70/pytorch_cls_tutorial_dnn_conversion.html
- Part 1: Deploying a PyTorch MobileNetV2 Classifier on the Intel Neural Compute Stick 2 :
<https://pemami4911.github.io/blog/2021/07/09/part-1-neural-compute-stick-2.html>
- OpenVINO 2021.4.2 설치 패키지 Repository :
<https://storage.openvinotoolkit.org/repositories/openvino/packages/2021.4.2/>
- OpenVINO 2021.4.2 Toolkit 설치 가이드 :
https://docs.openvino.ai/2021.4/openvino_docs_install_guides_installing_openvino_linux.html
- OpenVINO Model Optimizer ONNX IR 변환 :
https://docs.openvino.ai/latest/openvino_docs_MO_DG_prepare_model_convert_model_Convert_Model_From_ONNX.html
- OpenVINO PyTorch – ONNX – OpenVINO IR 변환 : <https://docs.openvino.ai/2021.4/notebooks/102-pytorch-onnx-to-openvino-with-output.html>
- OpenVINO, OpenCV, and Movidius NCS on the Raspberry Pi :
<https://pyimagesearch.com/2019/04/08/openvino-opencv-and-movidius-ncs-on-the-raspberry-pi/>
- EEMBC – 임베디드 딥러닝 벤치마크 : <https://www.eembc.org/mlmark/scores.php>

PyTorch ONNX + Raspberry Pi + NCS

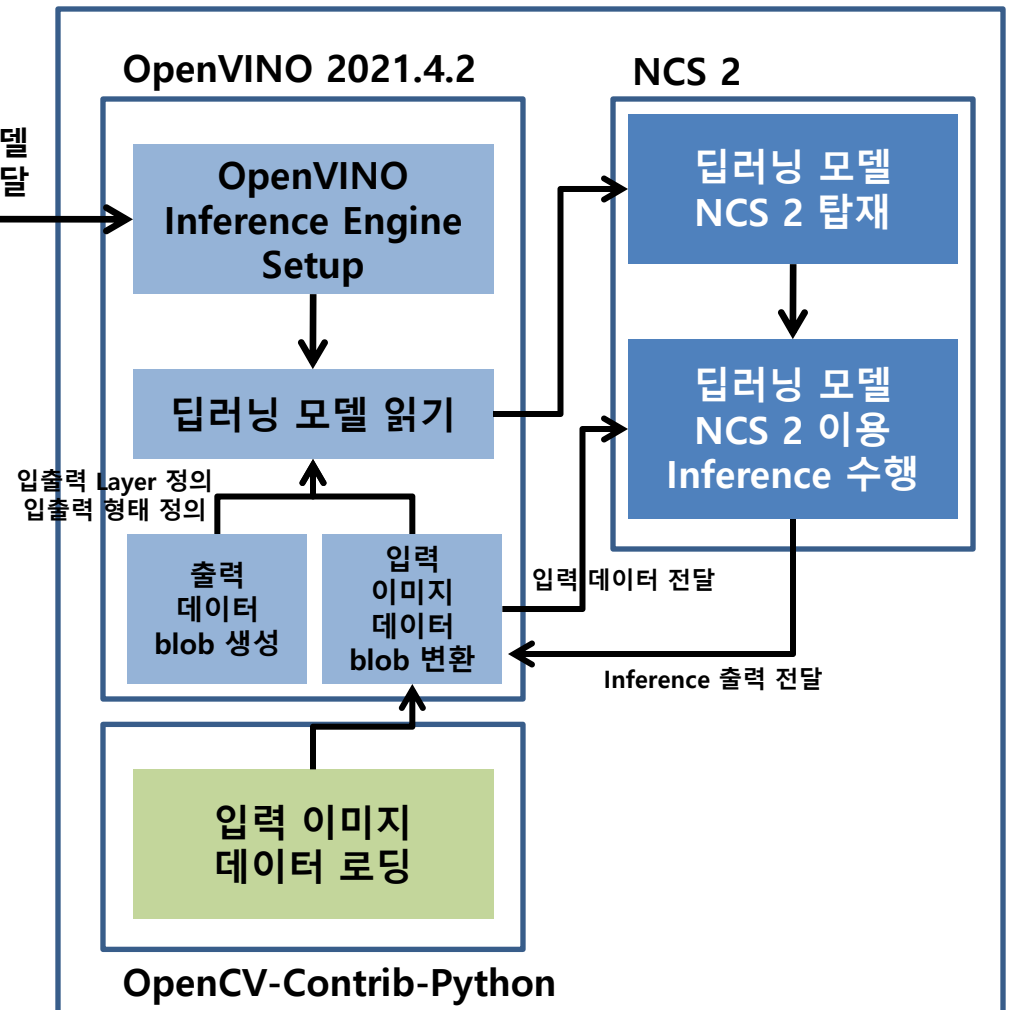
Workflow Setup

PyTorch ONNX + Raspberry Pi + NCS Workflow

Host PC (GPU)



Raspberry Pi 4 + NCS 2



Host PC Anaconda Setup

Host PC Anaconda Setup

Host PC – Anaconda (pytorch) : Anaconda Install

- 설치 파일 전송 방법

- <https://www.anaconda.com/products/distribution#Downloads> 이동
- 64-Bit (x86) Installer (581 MB) 다운로드
- WinSCP 또는 MobaXterm을 통해 설치파일을 WSL 세션으로 전송함
- bash 명령어를 사용하여 Anaconda 설치 Bash 파일 실행

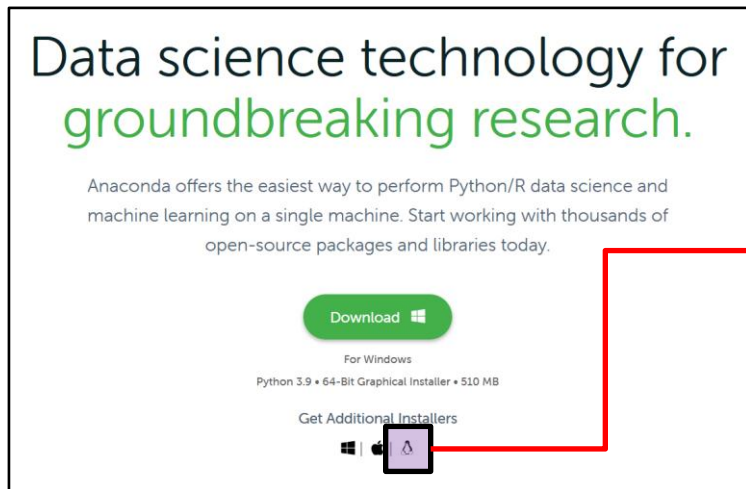
- Wget을 이용한 Terminal에서 설치 방법

- <https://www.anaconda.com/products/distribution#Downloads> 이동
- 64-Bit (x86) Installer (581 MB) 다운로드 우클릭 → 새로운 탭에서 실행
- 새로운 탭에 나오는 경로를 WSL Terminal 세션에서 Wget <새로운 탭 주소> 명령어 실행
(ex : `wget https://repo.anaconda.com/archive/Anaconda3-2021.11-Linux-x86_64.sh`)
- Wget을 통해 다운로드된 Anaconda 설치 Bash 파일을 실행

Host PC Anaconda Setup

Host PC – Anaconda (pytorch) : Anaconda Install

<https://www.anaconda.com/products/distribution#Downloads> 으로 이동



Anaconda 공식 사이트에서 리눅스 아이콘 클릭



(1) 64-Bit (x86) Installer (581 MB) 다운로드
→ 설치파일 WSL 세션으로 전송 후 설치

(2) 64-Bit (x86) Installer (581 MB) 우클릭 후 주소 복사
→ WSL 세션에서 Wget으로 다운로드 후 설치

Host PC Anaconda Setup

Host PC – Anaconda (pytorch) : Anaconda Install

- Anaconda 환경 만들기

- Anaconda 설치 후 Terminal 종료 후 재실행
- 실행 시 앞에 <base>가 붙는지 확인 (예 : `(base) luwis@DESKTOP-MDNHLT3:~$`)
- Python 3 이상의 환경을 갖춘 Conda 가상환경 생성 명령어 사용하여 가상 환경 만들기
 - `conda create -n <가상환경 이름> python=3.xx`
(※ OpenVINO는 Python 3.7 이상을 요구함)
(ex : 'Python 3.7'을 이용하는 'pytorch'라는 conda 가상환경 만들기 : `conda create -n pytorch python=3.7`)
- Anaconda 환경이 만들어진 후 가상환경 접속하기
 - `conda activate <가상환경 이름>`
(ex : 'pytorch'라는 conda 환경 접속하기 : `conda activate pytorch / (pytorch) luwis@DESKTOP-MDNHLT3:~$`)
- 현존하는 모든 Anaconda 환경 리스트 확인하기 : `conda env list`
- 현재 접속한 Anaconda 환경 로그아웃 하기 : `conda deactivate`

Host PC Anaconda Setup

Host PC – Anaconda (pytorch) : Anaconda Install

• Anaconda 환경에 PyTorch 설치하기

➤ Anaconda 환경에서 PyTorch 가상환경 접속하기

➤ conda activate <가상환경 이름>

(ex : 'pytorch'라는 conda 환경 접속하기 : conda activate pytorch / (pytorch) luwis@DESKTOP-MDNHLT3:~\$)

➤ PyTorch 공식 사이트(<https://pytorch.org/>)에서 제공하는 Anaconda 설치 명령어 실행

PyTorch Build	Stable (1.11.0)	Preview (Nightly)	LTS (1.8.2)
Your OS	Linux	Mac	Windows
Package	Conda	Pip	LibTorch
Language	Python	C++ / Java	Source
Compute Platform	CUDA 10.2	CUDA 11.3	ROCm 4.5.2 (beta)
Run this Command:	conda install pytorch torchvision torchaudio cudatoolkit=10.2 -c pytorch		

① 설치하려는 PyTorch 버전과 실행환경을 설정함

※ Anaconda에 설치하기에 Conda 선택

② 설치 명령어를 WSL 터미널의 Anaconda 환경에서 실행함

➤ Anaconda 환경에서 PyTorch 설치 확인

```
(base) luwis@DESKTOP-T793Q6K:~$ conda activate pytorch
(pytorch) luwis@DESKTOP-T793Q6K:~$ python3
Python 3.7.15 (default, Nov 24 2022, 21:12:53)
[GCC 11.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.__version__
'1.13.1'
```

Host PC OpenVINO Setup

Host PC OpenVINO Setup

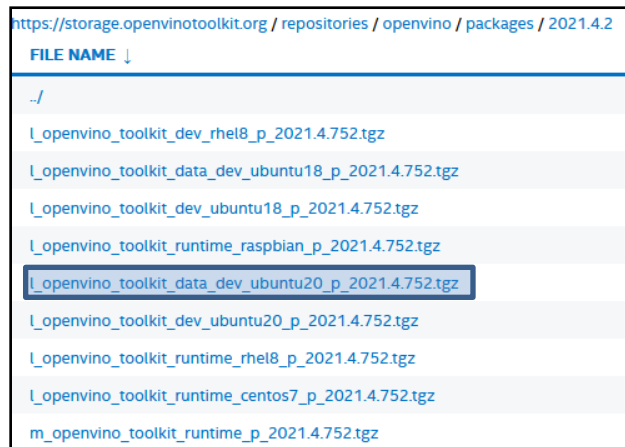
Host PC - OpenVINO 2021.4.2

1) `lsb_release -a` : Host PC 리눅스 버전 확인

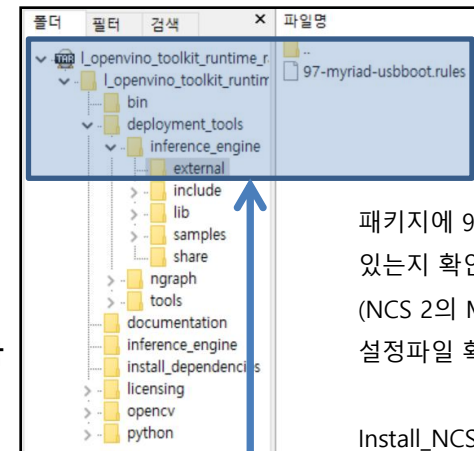
2) `wget`

`https://storage.openvinotoolkit.org/repositories/openvino/packages/2021.4.2/l_openvino_toolkit_data_dev_ubuntu20_p_2021.4.752.tgz -O openvino_2021.4.2_dev.tgz`

: Host PC OpenVINO 2021.4.2 다운로드



`https://storage.openvinotoolkit.org/repositories/openvino/packages`에서
Host PC의 리눅스 버전에 적합한
OpenVINO Dev Toolkit을 다운로드함



패키지에 97-myriad-usbboot.rules가
있는지 확인함
(NCS 2의 Myriad 장치 인식을 위한
설정파일 확인)

`Install_NCS_udev_rules.sh`가 해당
.rules를 제대로 호출하는지 확인

```
if [ -f "$INTEL_OPENVINO_DIR/deployment_tools/inference_engine/external/97-myriad-usbboot.rules" ]; then
    sudo usermod -a -G users "$(whoami)"
    sudo cp "$INTEL_OPENVINO_DIR/deployment_tools/inference_engine/external/97-myriad-usbboot.rules" /etc/udev/rules.d/
```

3) `tar -xf openvino_2021.4.2.tgz` : 압축 해제

Host PC OpenVINO Setup

Host PC - OpenVINO 2021.4.2

4) `mv l_openvino_toolkit_runtime_raspbian_p_2021.4.752.tgz /home/<USER>/intel/opencvino_2021.4.2`
: intel 디렉토리로 압축물 이동

5) `cd /home/<USER>/intel/opencvino_2021.4.2` : opencvino 디렉토리로 이동

6) `sudo apt install cmake` : cmake 다운로드

Optional : Dependency 문제로 실행 안될 경우 적용함

7) (base) `sudo -E ./install_dependencies/install_opencvino_dependencies.sh` : OpenVINO 종속성 프로그램 설치

8) (base) `sudo -E ./development_tools/model_optimizer/install_prerequisites/install_prerequisites.sh`

: Model Optimizer Dependency 설치 (※ Sytem-wide로 Dependency 설치 / ∴ apt 사용 설치)

```
(base) luwis@DESKTOP-T793Q6K:~/intel/opencvino_2021.4.2_dev/deployment_tools/model_optimizer/install_prerequisites$ sudo -E ./install_prerequisites.sh
```

9) (base) `source ./bin/setupvars.h`

: 환경변수 적용

→ 시스템 및 터미널 실행 시 자동 적용하기 위해서는 .bashrc에 “source /home/<USER>/intel/bin/setupvars.sh” 추가

10) [setupvars.sh] OpenVINO environment initialized 확인

Model Conversion

Model Conversion (PyTorch → ONNX)

Host PC – Anaconda (pytorch) : PyTorch – ONNX Conversion

```
import torch
import torch.onnx

from torchvision import models

import os

onnx_model_path = "models"
onnx_model_name = "resnet50.onnx"
full_model_path = os.path.join(onnx_model_path, onnx_model_name)

batch_size = 1

original_model = models.resnet50(weights=models.ResNet50_Weights.IMAGENET1K_V1)
original_model.eval()

dummy_input = torch.randn(batch_size, 3, 224, 224)
dummy_output = original_model(dummy_input)

torch.onnx.export(
    original_model,
    dummy_input,
    full_model_path,
    verbose=True,
    input_names=["input"],
    output_names=["output"],
    opset_version=13
)
```

Pretrained ResNet 50 로딩

PyTorch Model을 ONNX로 변환하는
과정에 사용할 Dummy Input/Output

ONNX 변환

변환할 ONNX
버전 명시

opset_version=13

Model Conversion (ONNX → OpenVINO IR)

Host PC – Anaconda (base) : Model Optimizer Conversion Success

```
(base) luwis@DESKTOP-T793Q6K:~/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer$ python3 mo.py --input_model /home/luwis/tmp/torch_onnx_conversion/models/resnet50.onnx --output_dir /home/luwis/tmp/torch_onnx_conversion/models
```

- deployment_tools → model_optimizer : OpenVINO Model Optimizer 디렉토리 이동
 - mo.py --input_model <ONNX Model 경로> --output_dir <OpenVINO IR 결과물 출력 디렉토리>
- : Model Optimizer 실행 Python Script 사용

(※ Anaconda base에서 실행 / ∴ Dependency가 System-wide로 설치 되었기 때문임)

```
Model Optimizer arguments:
Common parameters:
- Path to the Input Model:      /home/luwis/tmp/torch_onnx_conversion/models/resnet50.onnx
- Path for generated IR:        /home/luwis/tmp/torch_onnx_conversion/models
- IR output name:               resnet50
- Log level:                     ERROR
- Batch:                        Not specified, inherited from the model
- Input layers:                 Not specified, inherited from the model
- Output layers:                Not specified, inherited from the model
- Input shapes:                 Not specified, inherited from the model
- Mean values:                  Not specified
- Scale values:                 Not specified
- Scale factor:                 Not specified
- Precision of IR:              FP32
- Enable fusing:                 True
- Enable grouped convolutions fusing: True
- Move mean values to preprocess section: None
- Reverse input channels:       False
```

**ONNX – IR
변환 셋팅 확인**

```
[ WARNING ]
Detected not satisfied dependencies:
  numpy: installed: 1.21.5, required: < 1.20

Please install required versions of components or use install_prerequisites script
/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/install_prerequisites
Note that install_prerequisites scripts may install additional components.
/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/extensions/front/on
ture_function' instead.
'data_type': TENSOR_TYPE_TO_NP_TYPE[t.type.elem_type]
/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/extensions/analysis
fe. If you specifically wanted the numpy scalar type, use 'np.bool_' here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1
nodes = graph.get_op_nodes(op='Parameter', data_type=np.bool)
/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/extensions/ops/Redu
u specifically wanted the numpy scalar type, use 'np.bool_' here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1
used dims = np.zeros(len(in_shape), dtype=np.bool)
/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/mo/front/common/par
safe. If you specifically wanted the numpy scalar type, use 'np.bool_' here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1
mask = np.zeros_like(shape, dtype=np.bool)
[ SUCCESS ] Generated IR version 10 model.
[ SUCCESS ] XML file: /home/luwis/tmp/torch_onnx_conversion/models/resnet50.xml
[ SUCCESS ] BIN file: /home/luwis/tmp/torch_onnx_conversion/models/resnet50.bin
[ SUCCESS ] Total execution time: 6.62 seconds.
[ SUCCESS ] Memory consumed: 399 MB.
It's been a while, check for a new version of Intel(R) Distribution of OpenVINO(TM) toolkit
the GitHub*
```

모델 변환 성공 확인

Model Conversion (ONNX → OpenVINO IR)

Host PC – Anaconda (base) : Model Optimizer Conversion Fail – Dependency Error

```
(base) luwis@DESKTOP-T793Q6K:~/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer$ python3 mo.py --input_model /home/luwis/tmp/torch_onnx_conversion/models/resnet50.onnx --output_dir /home/luwis/tmp/torch_onnx_conversion/models
```

- deployment_tools → model_optimizer : OpenVINO Model Optimizer 디렉토리 이동
- mo.py --input_model <ONNX Model 경로> --output_dir <OpenVINO IR 결과물 출력 디렉토리>
: Model Optimizer 실행 Python Script 사용

(※ Anaconda base에서 실행 / ∴ Dependency가 System-wide로 설치 되었기 때문임)

```
[ WARNING ]
Detected not satisfied dependencies:
onnx: not installed, required: >= 1.8.1
numpy: installed: 1.21.5, required: < 1.20

Please install required versions of components or use install_prerequisites script
/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/install_prerequisites/install_prerequisites_onnx.sh
Note that install_prerequisites scripts may install additional components.
[ ERROR ] -----
[ ERROR ] ----- INTERNAL ERROR -----
[ ERROR ] Unexpected exception happened.
[ ERROR ] Please contact Model Optimizer developers and forward the following information:
[ ERROR ] No module named 'onnx'
[ ERROR ] Traceback (most recent call last):
  File "/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/mo/main.py", line 394, in main
    ret_code = driver(argv)
  File "/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/mo/main.py", line 356, in driver
    ret_res = emit_ir(prepare_ir(argv), argv)
  File "/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/mo/main.py", line 251, in prepare_ir
    import_extensions.load_dirs(argv.framework, extensions, get_front_classes)
  File "/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/mo/utils/import_extensions.py", line 91, in load_dirs
    load_dir(framework, path, get_front_classes)
  File "/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/mo/utils/import_extensions.py", line 76, in load_dir
    import_by_path(os.path.join(path, *p), [ext, *p])
  File "/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/mo/utils/import_extensions.py", line 20, in import_by_path
    importlib.import_module('{:}.format('.join(middle_names), name))
  File "/home/luwis/anaconda3/lib/python3.9/importlib/_init_.py", line 127, in import_module
    return bootstrap.gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1030, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1007, in _find_and_load
  File "<frozen importlib._bootstrap>", line 986, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 680, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 850, in exec_module
  File "<frozen importlib._bootstrap>", line 228, in _call_with_frames_removed
  File "/home/luwis/intel/openvino_2021.4.2_dev/deployment_tools/model_optimizer/mo/front/onnx/loader.py", line 11, in <module>
    import onnx
ModuleNotFoundError: No module named 'onnx'
```

Prerequisites : onnx, networkx, numpy

- pip install onnx
- pip install networkx
- pip install numpy

OpenVINO – NCS Inference Run

OpenVINO – NCS Inference Run

Raspberry Pi 4 + NCS 2 : OpenVINO Inference Run

```
import cv2 as cv
import numpy as np
import sys
import os
import time

from openvino.inference_engine import IECore

# Model Setup
model_path = "./resnet50.xml"
device = "MYRIAD"

# OpenVINO Inference Engine Setup
ie = IECore()

# Intermediate Representation Reading
print("Reading network : {}".format(model_path))
net = ie.read_network(model=model_path)

# Input Image Loading
input_img = cv.imread("./test_input.jpg")
input_img = cv.resize(input_img, (224, 224))
input_img = np.transpose(input_img, (2, 0, 1))
input_img = np.expand_dims(input_img, axis=0)
input_img = input_img / 255.

# Input Image Blob Preparation
input_blob = next(iter(net.input_info))
out_blob = next(iter(net.outputs))

net.input_info[input_blob].precision = 'U8'
net.outputs[out_blob].precision = 'FP32'

# Model Loading to Device
print("Loading network on {}".format(device))
exec_net = ie.load_network(network=net, device_name=device)

# Model Inference Forward Pass
print("Starting inference in synchronous mode")

start_time = time.time()
res = exec_net.infer(inputs={input_blob : input_img})
end_time = time.time()

print("Time Taken : {:.3f} sec".format(end_time - start_time))
print("Estimated FPS : {:.3f} fps".format(1 / (end_time - start_time)))

print((res[out_blob]).shape)
```

변환된 OpenVINO IR 모델 경로 명시

모형을 실행할 NCS2의 아키텍처 명시

OpenVINO Inference Engine 인스턴스 생성

모형 Weight 로딩

입력 이미지 로딩

OpenVINO IR 모델에 사용할 입력 Layer 선언

입출력 Numerical Precision 설정

OpenVINO IR 모델을 NCS2에 로딩함

Input Layer에 입력 데이터를 넣고 Inference 수행

```
pi@raspberrypi:~/intel/cv_vino_test/onnx_test $ python3 cv_vino_test.py
Reading network : ./resnet50.xml
Loading network on MYRIAD
Starting inference in synchronous mode
Time Taken : 0.060 sec
Estimated FPS : 16.566 fps
(1, 1000)
```