

Outline

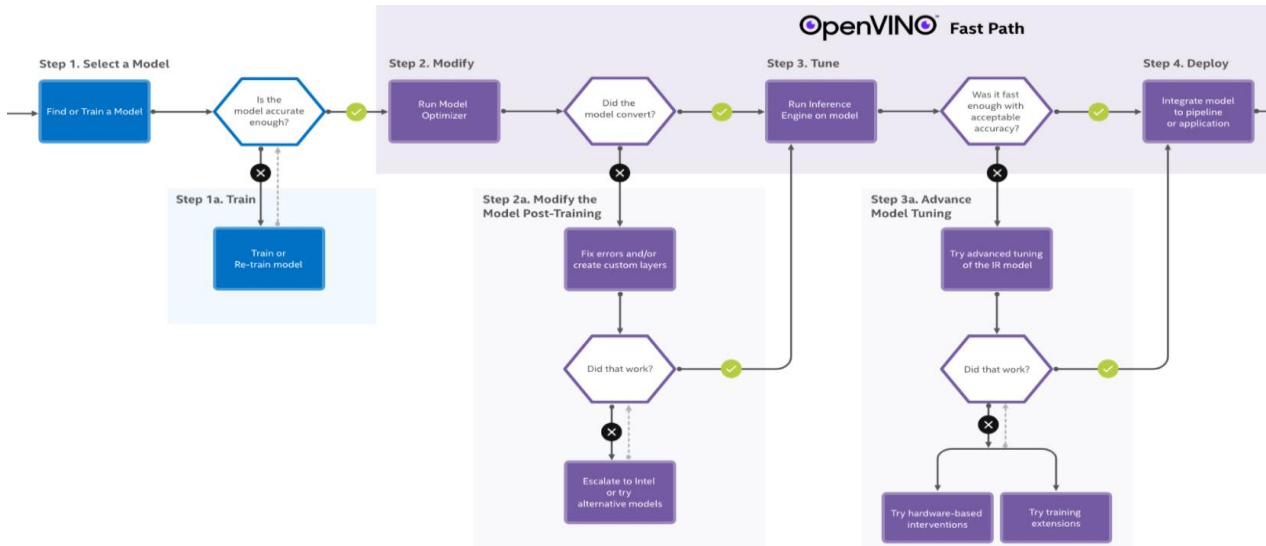
강희근

1. OpenVINO.....	2
1.1. Workflow	2
2. Installation.....	3
3. Setting	6
3.1. Set the Environment Variables.....	6
3.2. Configure the Model Optimizer	6
3.2.1. Model Optimizer Configuration Steps.....	7
4. Get started: Optimization	9
4.1. OpenVINO: Model Conversion	9
4.1.1. .h5 / .hdf5 format ('Keras Model' or 'Tensorflow' version < 2.x)	9
4.1.2. .pb format (Tensorflow version ≥ 2.x)	10
4.2. Applying Model Optimizer	11
5. Inference with IR Model	13
5.1. Inference Engine	13
5.1.1. Supported OS: OpenVINO version (2021.3, latest)	13
5.1.2. Environment Setting.....	13
5.1.3. OpenVINO Inference: using IR files	14

1. OpenVINO

- OpenVINO 는 Intel 에서 개발한 toolkit으로 응용프로그램을 위한 단일 toolkit을 제공한다. 이를 이용해 edge단(CPU, GPU, VPU 및 FPGA 등) 의 여러 플랫폼에서, 가속화하고자 하는 DL network들이 수행 가능하도록 한다

1.1. Workflow



Step 1: Select a Model

- 어떠한 task 에 맞는 Model 을 설계하고 이를 훈련하는 과정이다.

Step 2: Modify

- 훈련된 모델에 OpenVINO toolkit 이 제공하는 **Model Optimizer**를 적용한다.
- **Optimizer**는 가능한 경우, excess layers를 제거하고, group operation을 제거하기 위해, 더 단순하고 빠른 graph 형태로 수행한다. (custom layers를 추가하여, model conversion과 optimization process를 조정할 수 있다.)

Step 3: Tune

- 앞선 과정으로 생성된 **IR(Inference Representation)** format의 model에 **Inference Engine**을 적용하여, optimized된 model의 성능이, step 1에서 설계했던 model의 성능만큼 도달하는지 여부를 확인할 수 있다.

Step 4: Deploy

- The Intel Distribution of OpenVINO™ toolkit outputs optimized inference runtimes for the following devices:

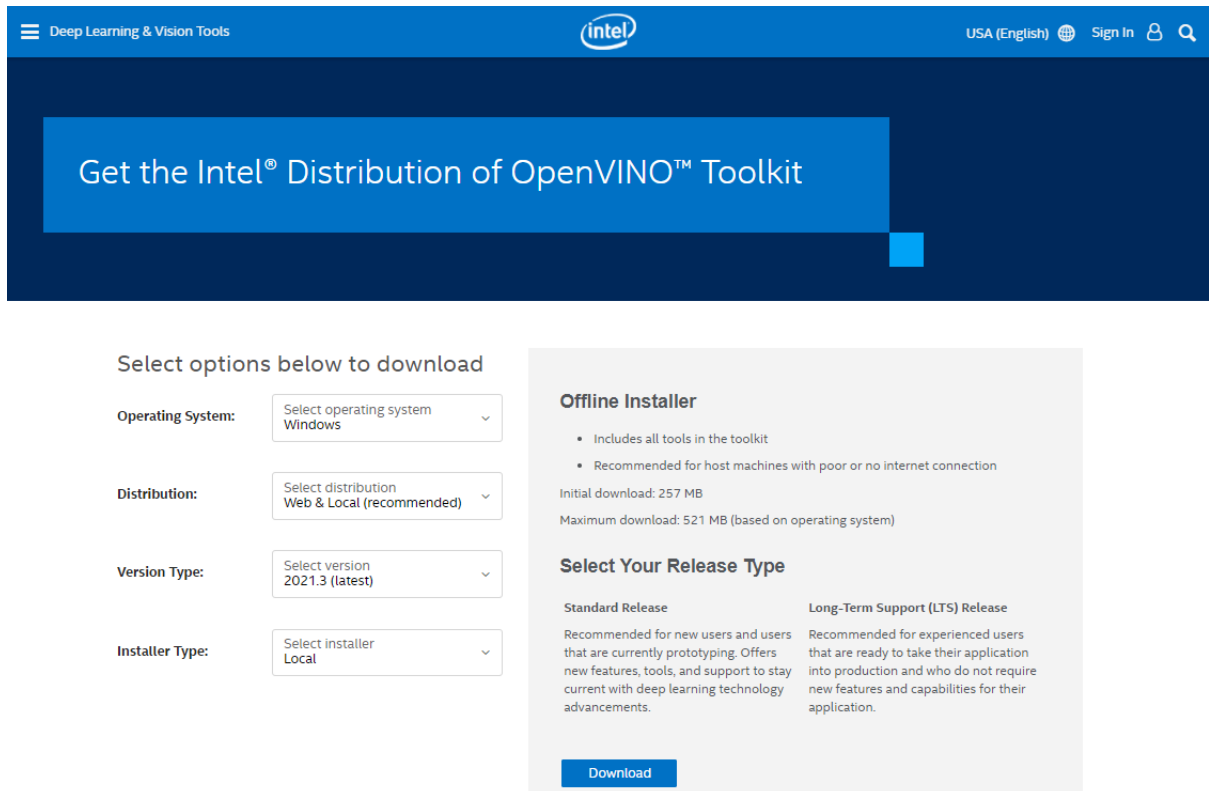
Intel® CPUs / Intel® Processor Graphics / Intel® Neural Compute Stick 2 / Intel® Vision Accelerator
Design with Intel® Movidius™ VPUs /

2. Installation

- Download Link: version (2021.3, latest)

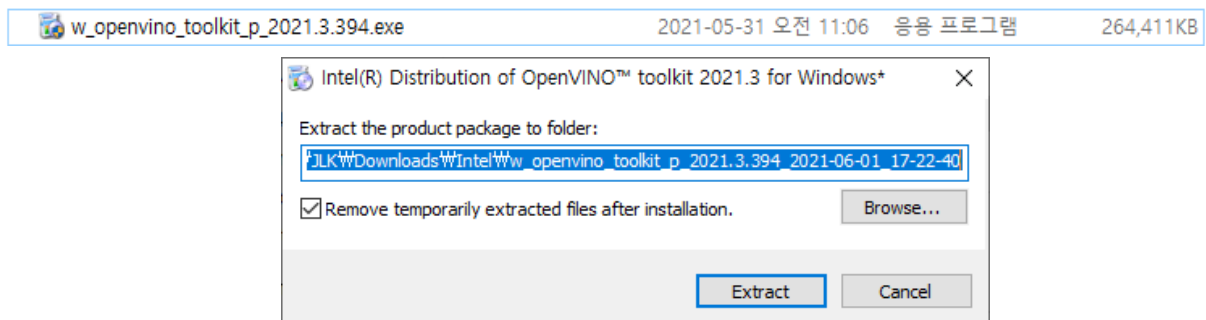
<https://software.intel.com/content/www/us/en/develop/tools/openvino-toolkit/download.html>

2.1 OpenVINO를 사용할 PC에 따라, Operating System > Distribution > Version Type > Installer Type 을 선택. (example. Windows > Web & Local > latest > Local)

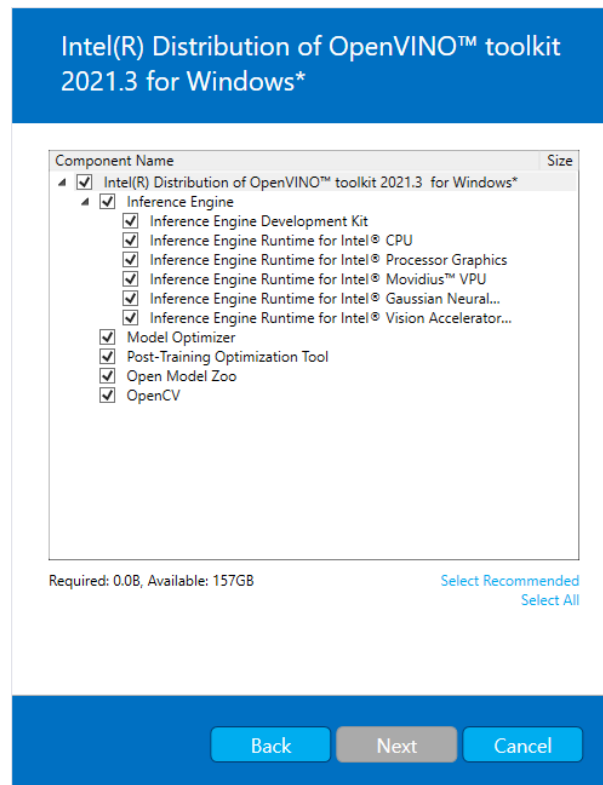


2.2 .exe 파일 실행 및 추출

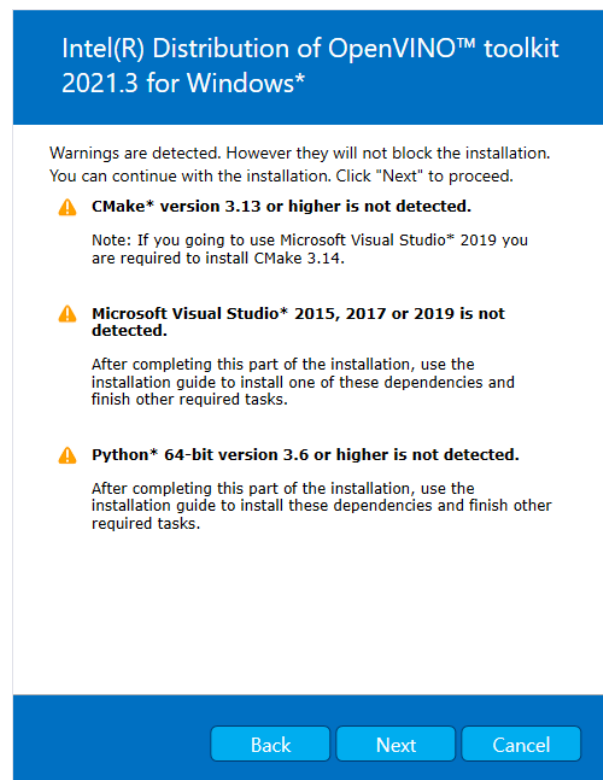
(installation path는 default setting으로 하여야 추후 환경 변수 세팅이 어렵지 않음.)



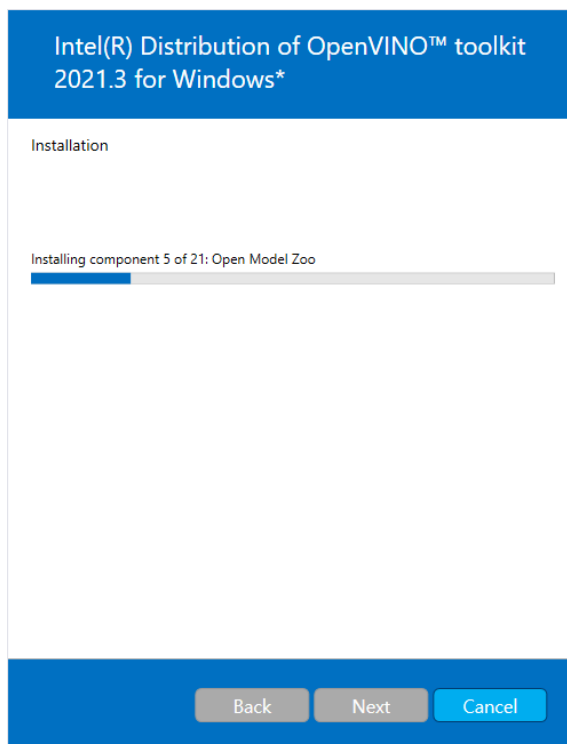
2.3 추출 이후 자동으로 설치 페이지로 전환.



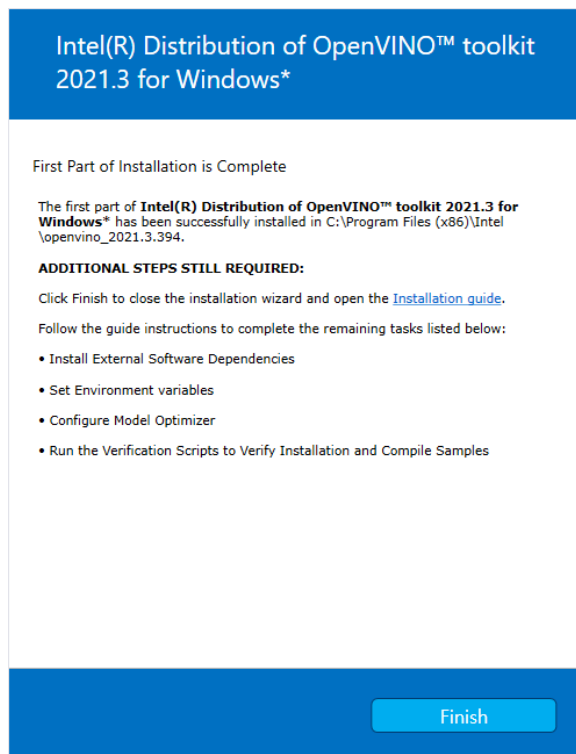
2.4 이후, 아래와 같이 OpenVINO 실행에 필요한 요구사항들이 체크된다. (CMake ≥ 3.13, MS Visual Studio 2015, 2017, or 2019 needed, Python 64bit ≥ 3.6) 설치 시, 자동으로 요구사항들이 설치되므로 넘어간다.



2.5 Installation



2.6 설치 이후 추가적으로 setting이 필요하다. (다음에 section에 정리)



3. Setting

Documents Link:

https://docs.openvinotoolkit.org/latest/openvino_docs_install_guides_installing_openvino_windows.html#set-the-environment-variables

3.1. Set the Environment Variables

OpenVINO app. 을 compile하고 실행하기 위해서는 몇 가지 환경 변수들을 업데이트해주어야 한다. 우선 Command Prompt 창을 열고, OpenVINO가 설치된 경로에 있는 `setupvars.bat` batch file을 실행하여 일시적으로 환경변수들을 지정해준다. (Command Prompt 창을 닫을 경우, OpenVINO toolkit 환경 변수들은 삭제된다.)

```
C:\Program Files (x86)\Intel\openvino_2021\bin\setupvars.bat
```

IMPORTANT: Windows PowerShell* is not recommended to run the configuration commands, please use the Command Prompt instead.

3.2. Configure the Model Optimizer

Model Optimizer가 CPU를 사용하여 훈련된 모델에 대해 작업하는데 필요한 모든 installation, configuration, build을 하는 과정이다.

IMPORTANT: These steps are required. You must configure the Model Optimizer for at least one framework. The Model Optimizer will fail if you do not complete the steps in this section.

Model Optimizer 는 OpenVINO toolkit 의 주요 구성요소이다. Model Optimizer를 통해 처리하지 않는다면 훈련시킨 모델을 inference 할 수 없다. Model optimizer를 통해 사전 훈련된 모델을 처리할 경우, 네트워크의 **Intermediate Representation(IR)** 을 출력으로 받는다. 이때, **IR** 은 전체 모델을 나타내는 파일 쌍이다.

- `.xml` : 네트워크 topology 에 대해 나타냄.
- `.bin` : weights, biases binary data 를 포함한다.

이후에, Inference Engine은 CPU, GPU, VPU등 하드웨어의 common API를 사용하여, 해당 IR 파일들을 reads, loads, and infer 한다.

Model Optimizer는 Python-based command line tool 이고, 아래의 경로에 있으며, 해당 경로에는 6 종류의 Model Optimizer들이 존재한다.

```
"C:\Program Files (x86)\Intel\openvino_2021\deployment_tools\model_optimizer\"
```

```
"mo.py" , "mo_caffe.py" , "mo_kaldi.py" , "mo_mxnet.py" , "mo_onnx.py" , "mo_tf.py"
```

이들은 mo{framework}.py 형태의 이름을 갖는데, 해당 framework로 훈련된 모델에 맞는 Model Optimizer를 사용하여, Inference Engine이 사용할 수 있는 최적화된 IR format으로 변환해줄 수 있다.

3.2.1. Model Optimizer Configuration Steps

Model Optimizer는 지원하는 모든 framework에 대해 한번에 configure 하거나, 한 번에 하나의 framework에 대해 configure 할 수 있다. 사용하는 자의 필요에 따라 사용하되, error messages가 출력 될 경우 모든 dependency들을 설치했는지 다시 확인해야한다. (설치 유무 및 버전 정보)
아래의 Model Optimizer configuration step에서는 가능하면 command prompt를 사용하였다.

IMPORTANT: The Internet access is required to execute the following steps successfully. If you have access to the Internet through the proxy server only, please make sure that it is configured in your environment.

Option 1 : Configure the Model Optimizer for all supported frameworks :

1. Command prompt를 실행해 주고, Model Optimizer의 prerequisites 디렉토리로 이동한다.

```
cd C:\Program Files (x86)\Intel\openvino_2021\deployment_tools\model_optimizer\install_prerequisites
```

2. 이후 아래의 batch file을 실행해준다. 해당 batch file은 Caffe*, Tensorflow* 1.x, MXNet*, Kaldi*, ONNX*를 위한 Model optimizer 의 prerequisites를 포함하고있다. (Tensorflow version이 1.x 까지만 지원하므로 2.x 이상의 버전을 사용하여 생성한 모델의 경우 유의할 것.)

```
install_prerequisites.bat
```

Option 2 : Configure the Model Optimizer for each framework separately :

1. Command prompt를 실행해 주고, Model Optimizer의 prerequisites 디렉토리로 이동한다.

```
cd C:\Program Files (x86)\Intel\openvino_2021\deployment_tools\model_optimizer\install_prerequisites
```

2. For **Caffe** :

```
install_prerequisites_caffe.bat
```

3. For **Tensorflow 2.x** :

```
install_prerequisites_tf2.bat
```

"C:\Program Files (x86)\Intel\wopenvino_2021.3.394\deployment_tools\model_optimizer\" 경로에 framework 마다 존재하는 requirements.txt 를 통해, 어떠한 것들이 어떤 버전으로 설치되는지 미리 알 수 있다.

IMPORTANT: python* 혹은 CMake*를 Windows* 환경 변수에 추가해주는 작업을 해주어야한다.

4. Get started: Optimization

4.1. OpenVINO: Model Conversion

훈련된 모델을 생성한 framework에 따라 확장자가 다르다. 대부분은 Keras 혹은 Tensorflow 를 사용하므로 이에 대한 경우를 다룰 것이다.

4.1.1. .h5 / .hdf5 format ('Keras Model' or 'Tensorflow' version < 2.x)

기본적으로 OpenVINO는 Tensorflow-based model일 경우, `.pb` format의 훈련된 모델을 입력으로 받는다. Tensorflow (version \geq 2.x) 일 경우, `.pb` format 으로 model과 weights를 저장하지만, Keras-based model or Tensorflow (version < 2.x) 일 경우는 `.h5 / .hdf5` format으로 저장한다. 이와 같은 경우, 아래의 코드를 통해, `.pb` format으로 변환이 필요하다. (해당 코드의 python=3.6, Tensorflow<2.x))

```
import tensorflow as tf
import tensorflow as tf
from tensorflow.keras.models import load_model

save_dir = "./models/" # save path
model_name = "./models/MobileNetV2.h5" # 변환할 파일의 path

tf.keras.backend.set_learning_phase(0) # test phase
model = load_model(model_name)
session = tf.keras.backend.get_session()

input_node = [t.op.name for t in model.inputs]
output_node = [t.op.name for t in model.outputs]

print(input_node, output_node)

from tensorflow.python.framework import graph_io

def freeze_graph(session, output, save_dir, save_fname='frozen_model.pb', save_pb_as_text=False):
    with session.graph.as_default():
        graph_def_inf = tf.graph_util.remove_training_nodes(session.graph.as_graph_def())
        graph_def_frozen = tf.graph_util.convert_variables_to_constants(session, graph_def_inf, output)
        graph_io.write_graph(graph_def_frozen, save_dir, save_fname, as_text = save_pb_as_text)
        return graph_def_frozen

frozen_graph = freeze_graph(session, [t.op.name for t in model.outputs], save_dir= save_dir)
```

Notes

해당 코드를 통해 변환된, frozen_model.pb 가 생성됨을 확인할 수 있다.

4.1.2. .pb format (Tensorflow version $\geq 2.x$)

Tensorflow (version $\geq 2.x$) 에서 model.save() 를 통해 모델을 저장할 경우, /assets/, /variables/, saved_model.pb 등이 생성된다. 아래의 코드를 통해, 모델을 load할때는, model.save() 와 같은 path를 지정해주어야 한다.

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.python.framework.convert_to_constants import convert_variables_to_constants_v2
import numpy as np
from tensorflow.keras.models import load_model

#path of the directory where you want to save your model
frozen_out_path = './models/MobileNetV2/'

# name of the .pb file
frozen_graph_filename = 'frozen_graph'
model = load_model('./models/MobileNetV2/') # tf >= 2.x 에서 모델을 저장한 디렉토리

# Convert Keras model to Concrete Function
full_model = tf.function(lambda x: model(x))
full_model = full_model.get_concrete_function(tf.TensorSpec(model.inputs[0].shape, model.inputs[0].dtype))

# Get frozen graph def
frozen_func = convert_variables_to_constants_v2(full_model)
frozen_func.graph.as_graph_def()

layers = [op.name for op in frozen_func.graph.get_operations()]
print("-" * 60)
print("Frozen model layers: ")
for layer in layers:
    print(layer)
print("-" * 60)
print("Frozen model inputs: ")
```

```

print(frozen_func.inputs)
print("Frozen model outputs: ")
print(frozen_func.outputs)

tf.io.write_graph(graph_or_graph_def=frozen_func.graph,
                  logdir=frozen_out_path,
                  name=f"{frozen_graph_filename}.pb",
                  as_text=False)
tf.io.write_graph(graph_or_graph_def=frozen_func.graph,
                  logdir=frozen_out_path,
                  name=f"{frozen_graph_filename}.pbtxt",
                  as_text=True)

```

Notes

해당 코드를 통해 변환된, frozen_graph.pb 와 frozen_graph.pbtxt 가 생성됨을 확인할 수 있다.

4.2. Applying Model Optimizer

앞선 과정에 의해 생성한 frozen model 들을 Model Optimizer를 통해 최적화 시켜줄 수 있다. 두 경우 모두 같은 방식을 사용한다.

1. Command prompt를 실행해 주고, Model Optimizer 디렉토리로 이동한다.

```
cd C:\Program Files (x86)\Intel\openvino_2021\deployment_tools\model_optimizer\
```

2. 아래와 같은 옵션을 참조하여 mo.py 를 실행하면, model이 optimized IR 파일(.bin , .mapping , .xml 파일들)로 변환된 것을 확인할 수 있다.

```
python mo.py --input_model {frozen model .pb file path} --output_dir {save dir} --data_type FP32 --input_shape [1,160,160,3] --model_name {save file name} --framework tf
```

mo.py 의 parameters 의 description은 아래와 같다.

Parameters	
--input_model	Frozen model 의 .pb 파일의 경로
--output_dir	출력 결과를 저장할 폴더 경로
--data_type	Tensor와 weight의 데이터 type 지정, (FP16 or FP32)
--input_shape	Model에 입력할 data shape ex) [1,160,160,3] Batch를 None으로 지정할 경우 error (1이상으로 지정할 것.)
--model_name	저장할 model 이름
--framework	사용한 framework ex) [tf, caffe, mxnet, kald, onnx 등...]

Notes

IR format의 파일들이 제대로 생성되었다고 해도, 생성된 .xml 파일의 metadata 파트를 확인하여 정보들이 맞는지 확인하도록 한다. (ex. OpenVINO version, model layers)

5. Inference with IR Model

5.1. Inference Engine

5.1.1. Supported OS: OpenVINO version (2021.3, latest)

- Ubuntu 18.04 : py 3.6, 3.7
- Ubuntu 20.04 : py 3.6, 3.7, 3.8
- Windows 10 : py 3.6, 3.7, 3.8 (check IMPORTANT section)

IMPORTANT: 21.06.09 현재, Windows 10 x64, Python 3.8 환경에서 OpenVINO의 python library를 찾지 못하는 issue가 있음. OpenVINO 공식 Github에서는 이를 해결 중인 것으로 사료됨. (21.4.29 last comment).

<https://github.com/openvinotoolkit/openvino/issues/5267>

아래의 prerequisites batch file을 실행하여, Python 3.6 과 Tensorflow 1.15.x 버전을 사용하는 경우에는 문제없이 model Inference가 됨을 확인함.

```
<INSTALL_DIR>%deployment_tools%model_optimizer%install_prerequisites%install_prerequisites_tf.bat
```

5.1.2. Environment Setting

Inference Engine Python API 를 위한 환경을 구성하기 위해,

- Ubuntu 18.04 or 20.04 : `source <INSTALL_DIR>/bin/setupvars.sh` .
- Windows 10 : `call <INSTALL_DIR>%bin%setupvars.bat`

IMPORTANT: Command Prompt 창을 닫았을 때는 다시 위의 script를 재실행해주어야 한다.

위의 script들은 최신 Python 버전을 자동으로 감지하여, 해당 버전을 지원하는 경우, 요구되는 환경을 구성한다. 만약에 다른 특정 버전의 Python을 사용하고 싶을 경우에는 해당 script를 실행하기 이전에 아래와 같이 환경변수를 지정해주어야 한다. (사용중인 python 버전에 대한 path는 꼭 추가 할 것.)

```
PYTHONPATH=<INSTALL_DIR>/python/<desired_python_version>
```

5.1.3. OpenVINO Inference: using IR files

```
import sys
sys.path.append("C:\\Program Files (x86)\\Intel\\openvino_2021.3.394\\python\\python3.6") # openvino API 의
path 가 잘 잡히지 않는 경우 run the code

from openvino.inference_engine import IENetwork, IECore, Blob

model_path = "C:/Users/JLK/Desktop/working/openvino_guide/models/MobileNetV2/ir" # IR format 파일이
있는 path (.xml/.bin)
model_xml = model_path + "/frozen_mobilenetv2.xml"
model_bin = model_path + "/frozen_mobilenetv2.bin"

device='CPU' # resource
net = IENetwork(model=model_xml, weights=model_bin)

iec = IECore()
# supported_layers = ie.query_network(net, device) # check available layers
model = iec.load_network(network=net, device_name=device)

## tensorflow 의 predict ~= openVINO 의 infer
# 입력과 출력 구조 확인.
in_blob = next(iter(net.input_info))
out_blob = next(iter(net.outputs))

# infer
result = model.infer(inputs = {in_blob: image}) # image = model 을 optimizing 했던 input_shape 인 (1, 160,
160, 3)을 넣어줘야한다.
clf_result = result[out_blob]
clf_prob = clf_result.tolist()[0]
```

Notes

OpenVINO의 **Model Optimizer**를 사용해 변환한 **IR** (Inference Representation) 파일을 **Inference Engine API** 를 기반으로 작성한 위의 script를 통해 network 로써 load 해주었다. 또한 위의 model.infer() 를 사용하여, Keras or Tensorflow 의 model.predict() 와 동일한 결과를 도출할 수 있다