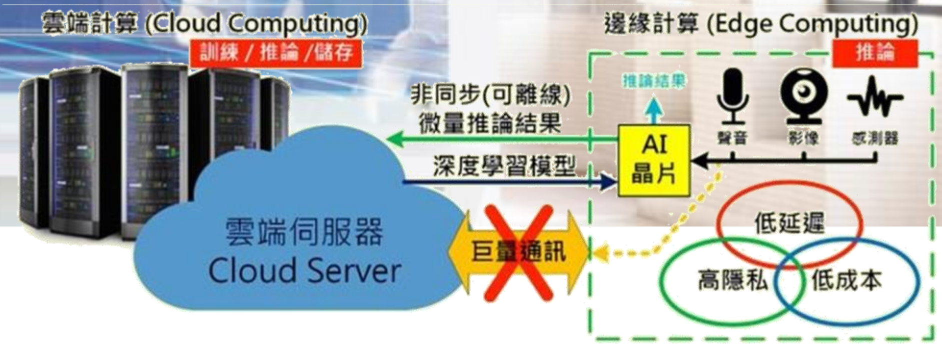
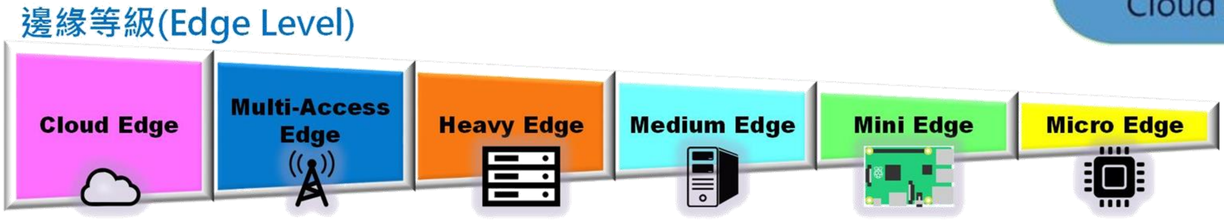
國立臺灣科技大學
NATIONAL TAIWAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Department of Computer Science
& Information Engineering
資訊工程系

人工智慧與邊緣運算實務

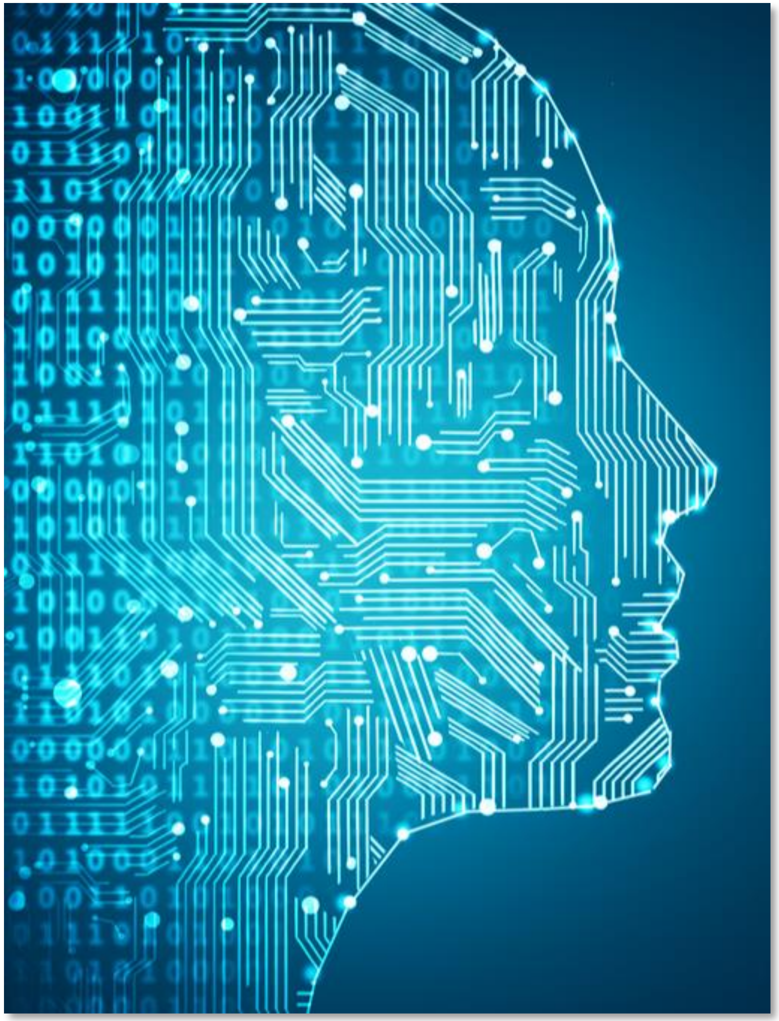
7.1 邊緣智慧案例實作【影像分類】





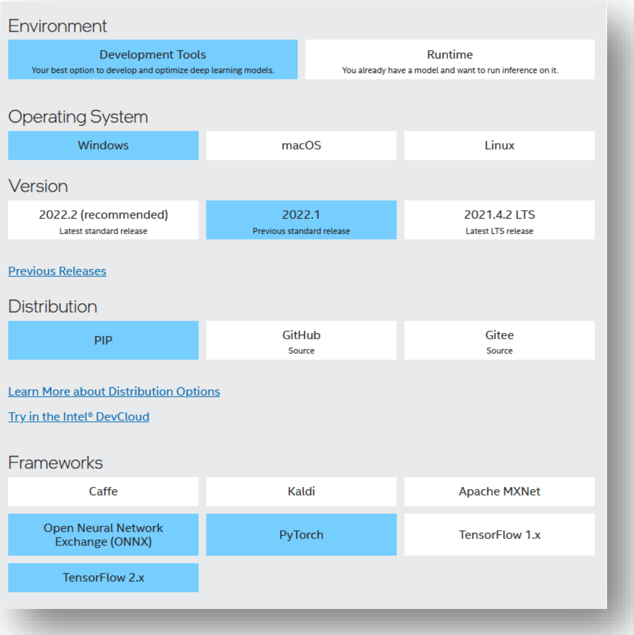
資訊工程系 許哲豪 助理教授

7.1 影像分類



- OpenVINO安裝
- 基本工作流程
- 影像分類範例說明

安裝OpenVINO及相關套件



Environment

[✓ Development Tools — Your best option to develop and optimize deep learning models.] | Runtime — You already have a model and want to run inference on it.

Operating System

[✓ Windows] | macOS | Linux

Version

2022.2 (recommended) — Latest standard release | [✓ 2022.1 — Previous standard release] | 2021.4.2 LTS — Latest LTS release

Previous Releases

Distribution

[✓ PIP] | GitHub — Source | Gitee — Source

Learn More about Distribution Options

Try in the Intel® DevCloud

Frameworks

Caffe | Kaldi | Apache MXNet

[✓ Open Neural Network Exchange (ONNX)] | [✓ PyTorch] | TensorFlow 1.x

[✓ TensorFlow 2.x]

- 進入命令列模式，切換到指定磁碟（假設為D槽）

```
cd d:\
```

- Step 1: 建立和啟動Python虛擬環境

```
python -m venv openvino_env
openvino_env\Scripts\activate
```

- Step 2: 更新 pip 到最新版本

```
python -m pip install --upgrade pip
```

不含openvino/samples和open_model_zoo/demos

- Step 3: 下載及安裝OpenVINO開發者版本（含指定框架）

```
pip install openvino-dev[ONNX,tensorflow2,pytorch]==2022.1.0
```

- Step 4: 檢查OpenVINO安裝及執行MO（若無錯誤則表示安裝OK）

```
python -c "from openvino.runtime import Core"
mo -h
```

https://www.intel.com/content/www/us/en/developer/tools/openvino-toolkit/download.html

安裝Tutorials Notebooks (1/2)

1. 安裝Python

➢至Python官網下載並安裝對應版本原則上支援3.7~3.9版Python 64bit版本

注意：Notebooks目前不支援3.10及以上版本

2. 安裝Git

➢至Git官網下載並安裝最新版本

3. 安裝C++ Redistributable (For Python 3.8)

➢下載Microsoft Visual C++ Redistributable並安裝。

步驟1~3為執行Tutorials Notebooks前必要動作，若已安裝過則可略過。

4. 建立虛擬環境

➢**python -m venv openvino_env**

5. 啟動虛擬環境

➢**openvino_env\Scripts\activate**

若已執行過安裝OpenVINO步驟，則步驟4可略過。直接切換到工作磁碟路徑，執行步驟5即可。

參考資料：https://github.com/openvinotoolkit/openvino_notebooks/wiki/Windows

安裝Tutorials Notebooks (2/2)

6. 複製程式庫並進入工作路徑

- **git clone --depth=1 https://github.com/openvinotoolkit/openvino_notebooks.git**
- **cd openvino_notebooks**

7. 安裝相關套件

- **python -m pip install --upgrade pip wheel setuptools**
- **pip install -r requirements.txt**

若沒安裝過OpenVINO 2022.1會自動安裝。若已有安裝過OpenVINO則會自動移除後再自動重新安裝。

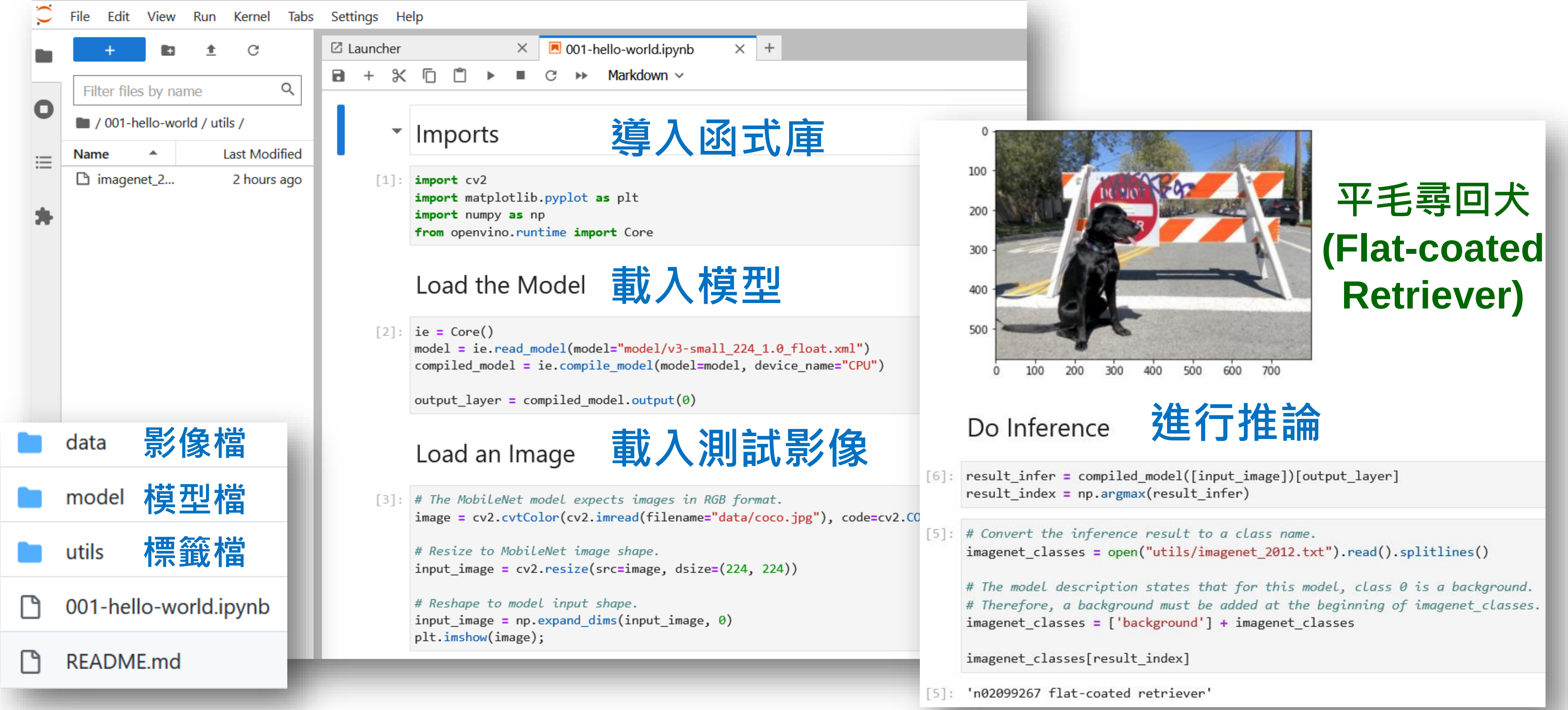
8. 執行範例

- 若欲執行所有範例則在命令列執行 「**jupyter lab notebooks**」，進入後再選擇欲運行的範例。
- 若想運行特定範例則指定對應程式路徑及名稱(*.ipynb)即可。
- **jupyter notebook notebooks/001-hello-world/001-hello-world.ipynb**

參考資料：https://github.com/openvinotoolkit/openvino_notebooks/wiki/Windows

Notebooks範例 001-hello-world

Jupyter Notebook 操作介面



data	影像檔
model	模型檔
utils	標籤檔
001-hello-world.ipynb	
README.md	

Imports 導入函式庫

```
[1]: import cv2
     import matplotlib.pyplot as plt
     import numpy as np
     from openvino.runtime import Core
```

Load the Model 載入模型

```
[2]: ie = Core()
     model = ie.read_model(model="model/v3-small_224_1.0_float.xml")
     compiled_model = ie.compile_model(model=model, device_name="CPU")

     output_layer = compiled_model.output(0)
```

Load an Image 載入測試影像

```
[3]: # The MobileNet model expects images in RGB format.
     image = cv2.cvtColor(cv2.imread(filename="data/coco.jpg"), code=cv2.CO

     # Resize to MobileNet image shape.
     input_image = cv2.resize(src=image, dsize=(224, 224))

     # Reshape to model input shape.
     input_image = np.expand_dims(input_image, 0)
     plt.imshow(image);
```

平毛尋回犬
(Flat-coated
Retriever)

Do Inference 進行推論

```
[6]: result_infer = compiled_model([input_image])[output_layer]
     result_index = np.argmax(result_infer)

[5]: # Convert the inference result to a class name.
     imagenet_classes = open("utils/imagenet_2012.txt").read().splitlines()

     # The model description states that for this model, class 0 is a background.
     # Therefore, a background must be added at the beginning of imagenet_classes.
     imagenet_classes = ['background'] + imagenet_classes

     imagenet_classes[result_index]

[5]: 'n02099267 flat-coated retriever'
```

在命令視窗按**Ctrl+C**結束Jupyter Notebook

安裝Open Model Zoo Demos

1. 複製程式庫

➢git clone --recurse-submodules https://github.com/openvinotoolkit/open_model_zoo.git

2. 測試單一影像產生深度圖範例 MonoDepth

支援模型名稱
fcrn-dp-nyu-depth-v2-tf
midasnet

➢cd \open_model_zoo\demos\monodepth_demo\python

```
omz_downloader --list models.lst
omz_converter --list models.lst
```

下載清單中所有模型並轉換

二選一

```
➢omz_downloader --name midasnet
omz_converter --name midasnet
```

下載指定模型並轉換

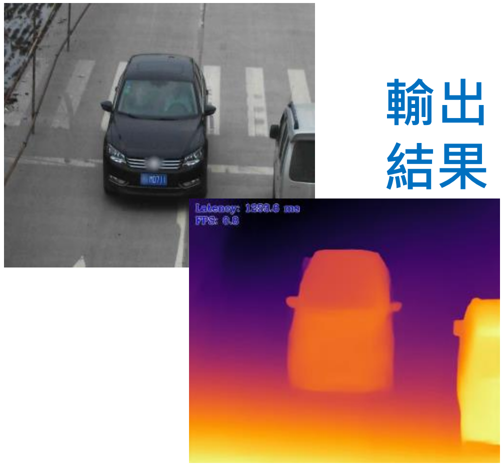
➢**下載圖檔並放入此目錄**
https://storage.openvinotoolkit.org/data/test_data/images/car.bmp

➢**執行MonoDepth範例程式**

```
python monodepth_demo.py -d GPU -i car.bmp \
-m ./public/midasnet/FP16/midasnet.xml -o car_depth.bmp
```

輸入影像

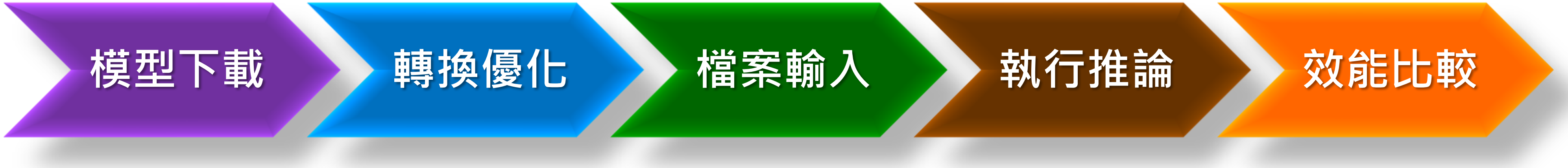
輸出結果



OpenVINO基本工作流程

omz_convert
(Model Optimizer)

***.exe, *.py**
(CPU, GPU, MYRIAD)



模型下載 → 轉換優化 → 檔案輸入 → 執行推論 → 效能比較

omz_downloader
(Open Model Zoo Model Downloader)

Images / videos
(*.jpg, *.png… / *.mp4)

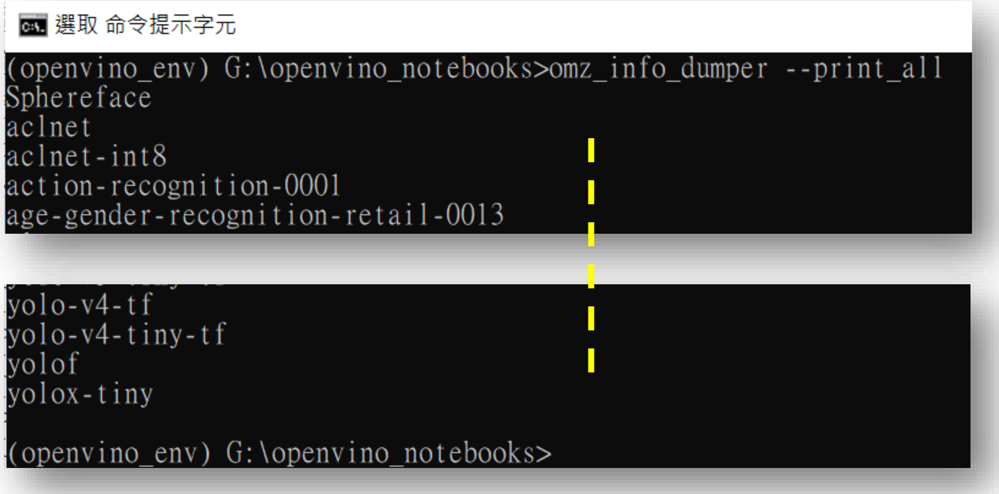
benchmark_app
(Latency, Throughput)

Notebooks - Working with Open Model Zoo Models
https://docs.openvino.ai/2022.1/notebooks/104-model-tools-with-output.html

參考資料：https://docs.openvino.ai/2022.1/openvino_docs_get_started_get_started_demos.html

模型下載

- ➢進入命令列模式。
- ➢**列印所有可下載模型清單：**
- ➢**omz_info_dumper --print_all**

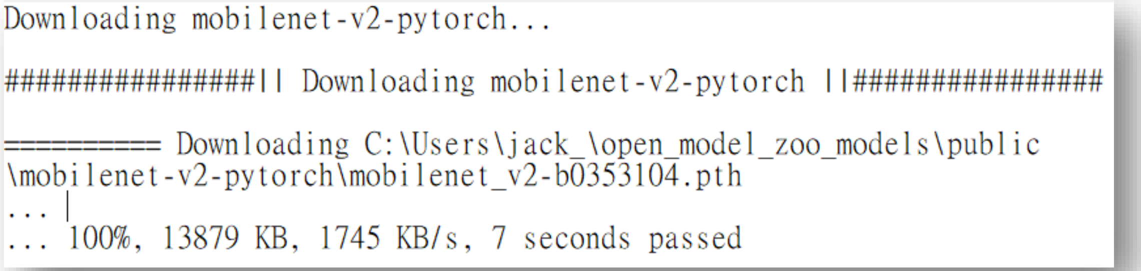


```
(openvino_env) G:\openvino_notebooks>omz_info_dumper --print_all
Sphereface
aclnet
aclnet-int8
action-recognition-0001
age-gender-recognition-retail-0013
```

```
yolo-v4-tf
yolo-v4-tiny-tf
yolof
yolox-tiny

(openvino_env) G:\openvino_notebooks>
```

- ➢**下載特定模型範例：**
- ➢**omz_downloader --name 模型名稱 --output_dir 自定義路徑名稱**



```
Downloading mobilenet-v2-pytorch...

################|| Downloading mobilenet-v2-pytorch ||################

========== Downloading C:\Users\jack_\open_model_zoo_models\public
\mobilenet-v2-pytorch\mobilenet_v2-b0353104.pth
... |
... 100%, 13879 KB, 1745 KB/s, 7 seconds passed
```

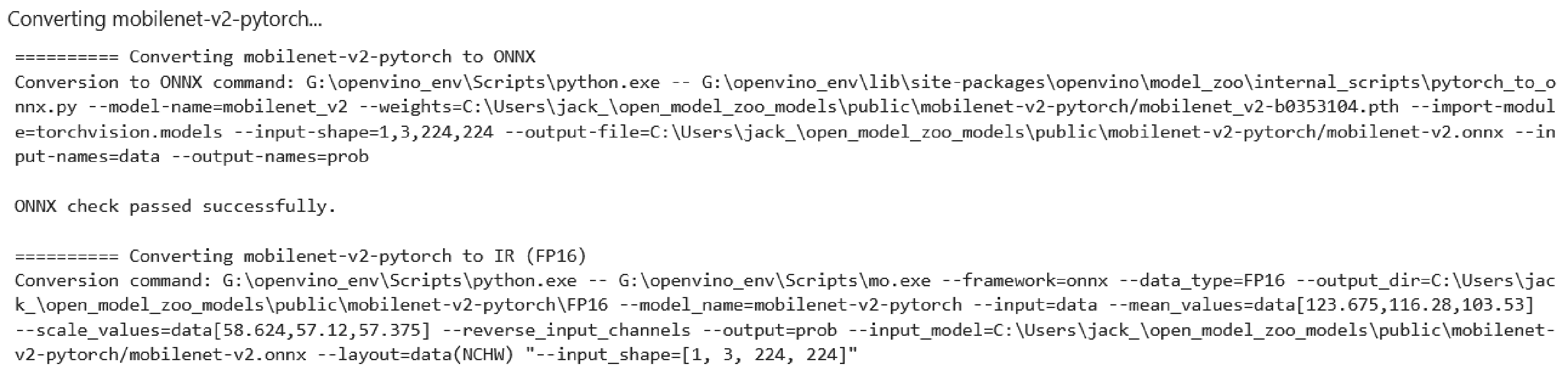
貼心提醒：
下載的模型會依Intel’s或Pulic Pre-Trained，分別在自定義路徑下建立 **/Intel** 或 **/Public** 後再建立模型名稱的檔案夾存放模型。

轉換優化

- 先建立一個存放轉換後IR檔的路徑

IR檔包含**模型結構(.xml)**及**權重(.bin)**另外有時會產生.json及.mapping檔案

- **mkdir IR檔輸出路徑名稱**
- 執行轉換優化程式，指定模型名稱、精度、模型路徑及輸出路徑
- **omz_converter --name 模型名稱 --precisions FP16 --download_dir 模型下載路徑 --output_dir IR檔輸出路徑**



```
Converting mobilenet-v2-pytorch...

========== Converting mobilenet-v2-pytorch to ONNX
Conversion to ONNX command: G:\openvino_env\Scripts\python.exe -- G:\openvino_env\lib\site-packages\openvino\model_zoo\internal_scripts\pytorch_to_onnx.py --model-name=mobilenet_v2 --weights=C:\Users\jack_\open_model_zoo_models\public\mobilenet-v2-pytorch/mobilenet_v2-b0353104.pth --import-module=torchvision.models --input-shape=1,3,224,224 --output-file=C:\Users\jack_\open_model_zoo_models\public\mobilenet-v2-pytorch/mobilenet-v2.onnx --input-names=data --output-names=prob

ONNX check passed successfully.

========== Converting mobilenet-v2-pytorch to IR (FP16)
Conversion command: G:\openvino_env\Scripts\python.exe -- G:\openvino_env\Scripts\mo.exe --framework=onnx --data_type=FP16 --output_dir=C:\Users\jack_\open_model_zoo_models\public\mobilenet-v2-pytorch\FP16 --model_name=mobilenet-v2-pytorch --input=data --mean_values=data[123.675,116.28,103.53] --scale_values=data[58.624,57.12,57.375] --reverse_input_channels --output=prob --input_model=C:\Users\jack_\open_model_zoo_models\public\mobilenet-v2-pytorch/mobilenet-v2.onnx --layout=data(NCHW) "--input_shape=[1, 3, 224, 224]"
```

檔案輸入

- 根據不同模型需求，輸入檔案格式可能為下列類型：
- 聲音(*.wav, *mp3 …)
- 影像(*.jpg, *.png …)
- 影片(*.mp4 …)
- 數據集(*.csv, *.txt …)
- 其它
- OpenVINO提供一些常用的測試像和影片方便測試。

https://storage.openvinotoolkit.org/data/test_data
https://github.com/intel-iot-devkit/sample-videos

執行推論

- 如果是採用標準安裝(installer)或一步一步安裝(step by step)者，在執行OpenVINO任何範例前要先執行環境變數設定。
- **<INSTALL_DIR>\setupvars.bat**
- 若欲使用C/C**++**範例程式，要先執行编譯工作。
- **<INSTALL_DIR>\samples\cpp\build_samples_msvc.bat**
- 完成後會在下列路徑產生可執行檔(*.exe)
- C**++** samples:
  C:\Users\<user>\Documents\Intel\OpenVINO\inference_engine_cpp_samples_build\intel64\Release
- 接著輸入各執行檔所需參數即可執行。
- ***.exe [option 1] [option 2] …**

若欲執行Python相關Samples，則可略過後面兩個步驟，直接運行*.py即可。

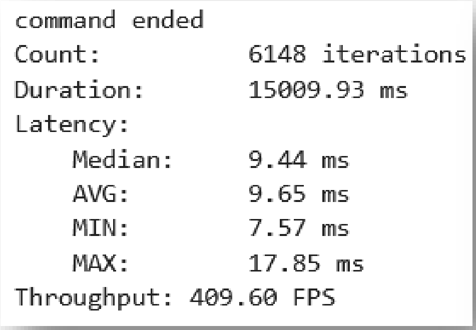
效能比較

- **Benchmark_app**可指定模型自行運行一段時間後再統計相關**延遲**(Latency, ms)及**吞吐率**(Throughput, FPS)。
- **benchmark_app -m IR模型路徑及名稱 [依需求加入下面項目統計]**
- **-t 運行時間** ，單位為秒數，預設60秒。
- **-d 裝置名稱** ，可為CPU, GPU, MULTI，預設為CPU。
- **-api 同步模式** ，可為async, sync，預設為異步async。
- **-b 批次數量** ，指定批次大小，預設為1。
- 例：
- **benchmark_app -m .\open_model_zoo_models\public\mobilenet-v2-pytorch\FP16\mobilenet-v2-pytorch.xml -d CPU -t 15 -api async -b 1**

mobilenet-v2-pytorch benchmark

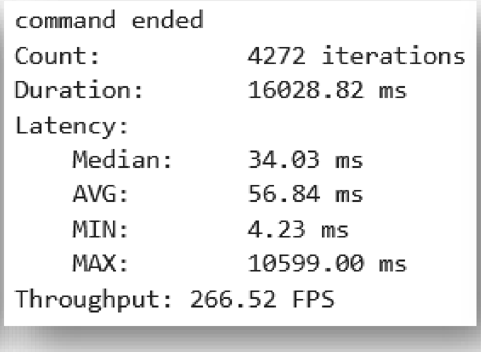
- 測試條件：
- CPU: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
- GPU: Intel(R) UHD Graphics 630 (iGPU)

-d CPU
-t 15
-api async
-b 1



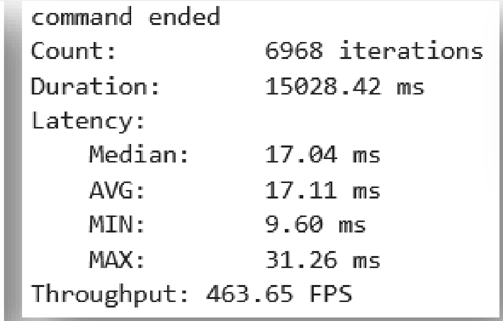
```
command ended
Count:          6148 iterations
Duration:       15009.93 ms
Latency:
    Median:     9.44 ms
    AVG:        9.65 ms
    MIN:        7.57 ms
    MAX:        17.85 ms
Throughput: 409.60 FPS
```

-d AUTO
-t 15
-api async
-b 1



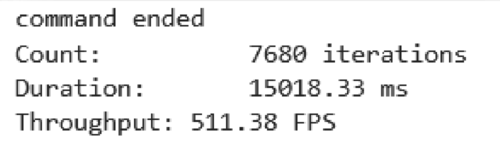
```
command ended
Count:          4272 iterations
Duration:       16028.82 ms
Latency:
    Median:     34.03 ms
    AVG:        56.84 ms
    MIN:        4.23 ms
    MAX:        10599.00 ms
Throughput: 266.52 FPS
```

-d GPU
-t 15
-api async
-b 1



```
command ended
Count:          6968 iterations
Duration:       15028.42 ms
Latency:
    Median:     17.04 ms
    AVG:        17.11 ms
    MIN:        9.60 ms
    MAX:        31.26 ms
Throughput: 463.65 FPS
```

-d MULTI:CPU,GPU
-t 15
-api async
-b 1



```
command ended
Count:          7680 iterations
Duration:       15018.33 ms
Throughput: 511.38 FPS
```

表現最佳
較純CPU提升24.8%

範例來源

Samples （C/C++/Python)	https://docs.openvino.ai/2022.1/openvino_docs_OV_UG_Samples_Overview.html
Tutorials （Python)	https://docs.openvino.ai/2022.1/tutorials.html
demos （C++/Python)	https://docs.openvino.ai/2022.1/omz_demos.html
Intel’s Pre-Trained	https://docs.openvino.ai/2022.1/omz_models_group_intel.html
Public Pre-Trained	https://docs.openvino.ai/2022.1/omz_models_group_public.html

資料來源：https://docs.openvino.ai/2022.1/index.html

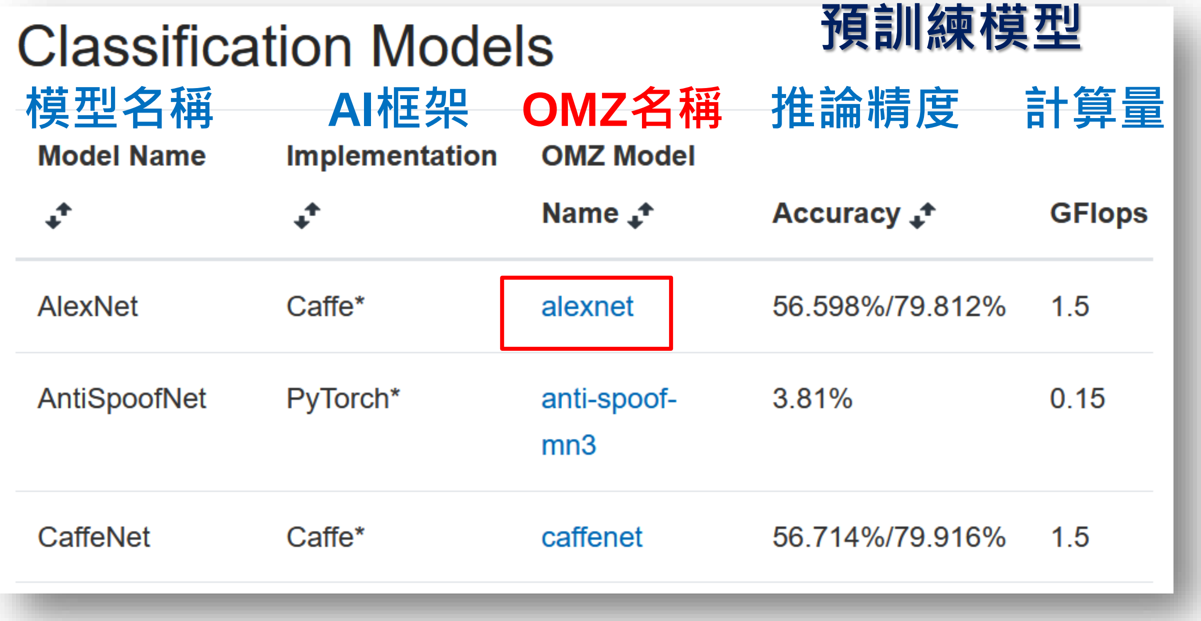
挑選模型

Public Pre-Trained Models

54項

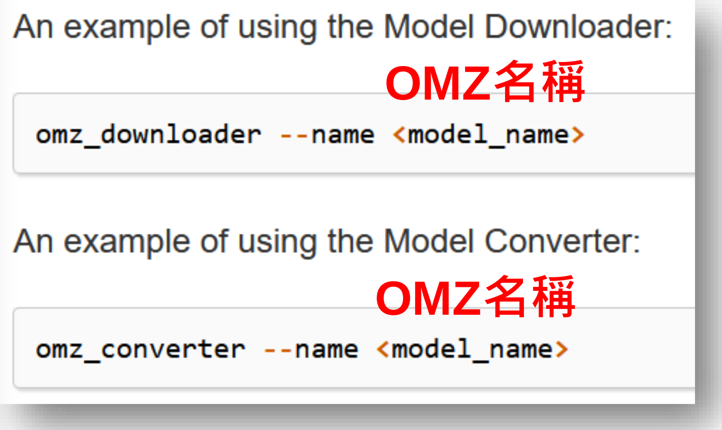
預訓練模型

下載和轉換模型到IR檔(xml, bin)



Classification Models

模型名稱	AI框架	OMZ名稱	推論精度	計算量
Model Name	Implementation	OMZ Model Name	Accuracy	GFlops
AlexNet	Caffe*	alexnet	56.598%/79.812%	1.5
AntiSpoofNet	PyTorch*	anti-spoof-mn3	3.81%	0.15
CaffeNet	Caffe*	caffenet	56.714%/79.916%	1.5



An example of using the Model Downloader:

OMZ名稱

```
omz_downloader --name <model_name>
```

An example of using the Model Converter:

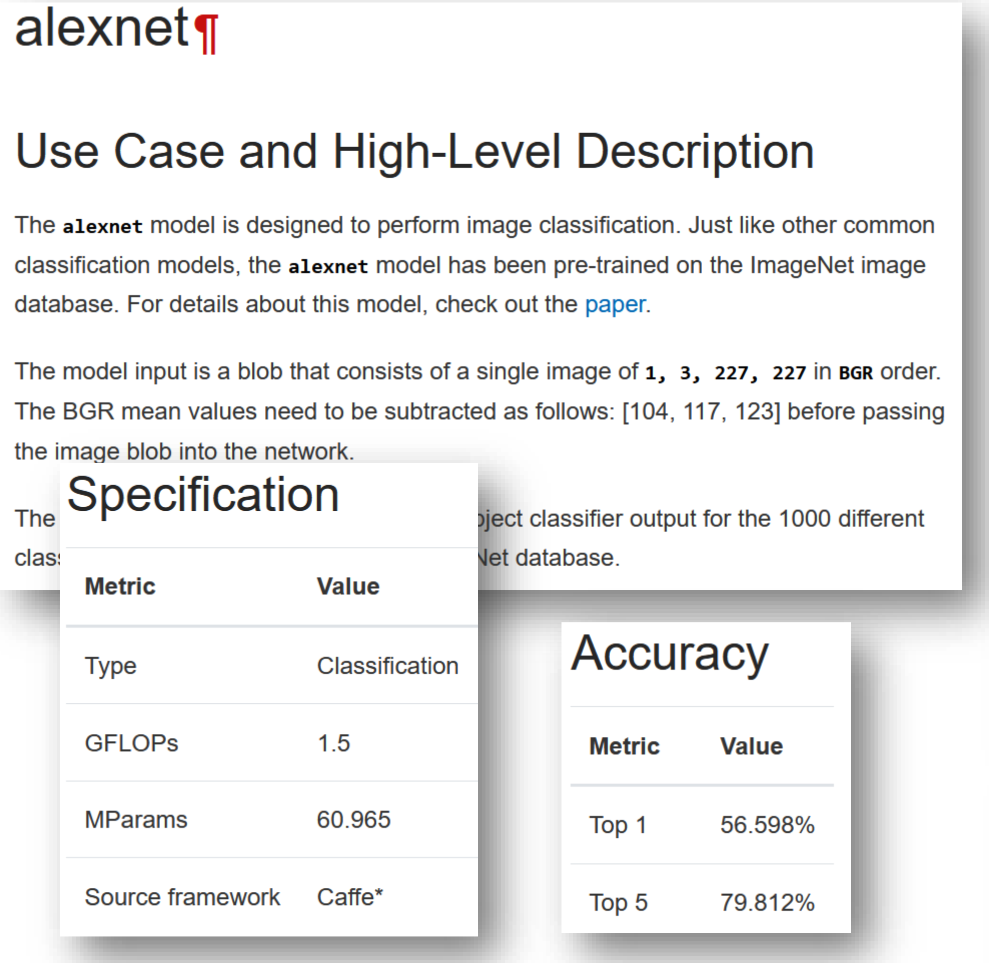
OMZ名稱

```
omz_converter --name <model_name>
```

點擊OMZ名稱，查看模型完整說明。

資料來源：https://docs.openvino.ai/2022.1/omz_models_group_public.html

alexnet 模型完整說明



alexnet¶

Use Case and High-Level Description

The `alexnet` model is designed to perform image classification. Just like other common classification models, the `alexnet` model has been pre-trained on the ImageNet image database. For details about this model, check out the paper.

The model input is a blob that consists of a single image of `1, 3, 227, 227` in `BGR` order. The BGR mean values need to be subtracted as follows: [104, 117, 123] before passing the image blob into the network.

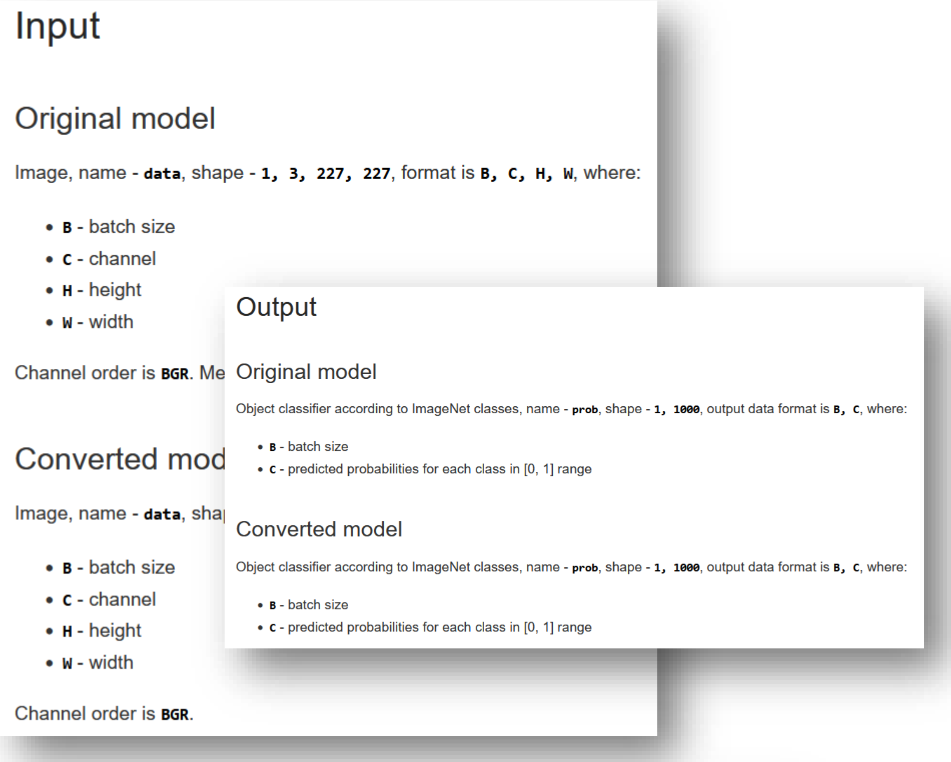
The bject classifier output for the 1000 different clas Net database.

Specification

Metric	Value
Type	Classification
GFLOPs	1.5
MParams	60.965
Source framework	Caffe*

Accuracy

Metric	Value
Top 1	56.598%
Top 5	79.812%



Input

Original model

Image, name - `data`, shape - `1, 3, 227, 227`, format is `B, C, H, W`, where:

- `B` - batch size
- `C` - channel
- `H` - height
- `W` - width

Channel order is `BGR`. Me

Converted mod

Image, name - `data`, sha

- `B` - batch size
- `C` - channel
- `H` - height
- `W` - width

Channel order is `BGR`.

Output

Original model

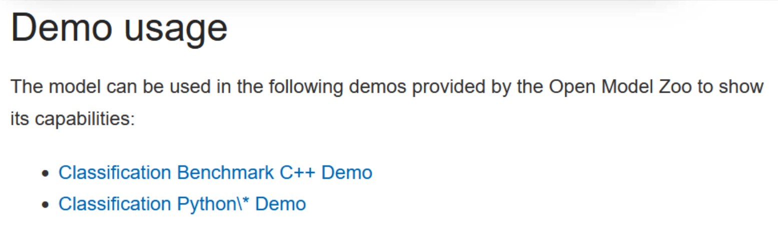
Object classifier according to ImageNet classes, name - `prob`, shape - `1, 1000`, output data format is `B, C`, where:

- `B` - batch size
- `C` - predicted probabilities for each class in [0, 1] range

Converted model

Object classifier according to ImageNet classes, name - `prob`, shape - `1, 1000`, output data format is `B, C`, where:

- `B` - batch size
- `C` - predicted probabilities for each class in [0, 1] range



Demo usage

The model can be used in the following demos provided by the Open Model Zoo to show its capabilities:

- Classification Benchmark C++ Demo
- Classification Python* Demo

資料來源： https://docs.openvino.ai/2022.1/omz_models_model_alexnet.html

影像分類實作 resnet-50-pytorch (1/2)

1. 下載測試用影片
   https://github.com/intel-iot-devkit/sample-videos/raw/master/fruit-and-vegetable-detection.mp4

2. 複製到工作路徑 \open_model_zoo\demos\classification_demo\python

3. 下載及轉換模型

```
➢omz_downloader --name resnet-50-pytorch
 omz_converter --name resnet-50-pytorch
```

可支援模型太多，
不要使用 --list models.lst
一次全部下載

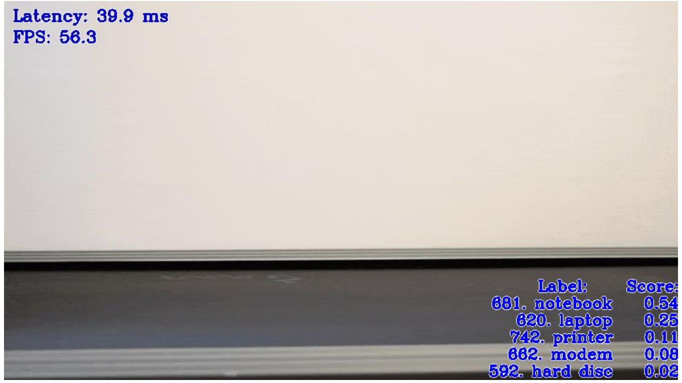
資料來源：https://docs.openvino.ai/2022.1/omz_demos_classification_demo_python.html

影像分類實作 resnet-50-pytorch (2/2)

4. 執行推論

```
python classification_demo.py ^
-m ./public/resnet-50-pytorch/FP16/resnet-50-pytorch.xml ^
-i fruit-and-vegetable-detection.mp4 ^
--labels ../../../data/dataset_classes/imagenet_2012.txt ^
-d GPU ^
-o result.avi
```

（選配輸出）

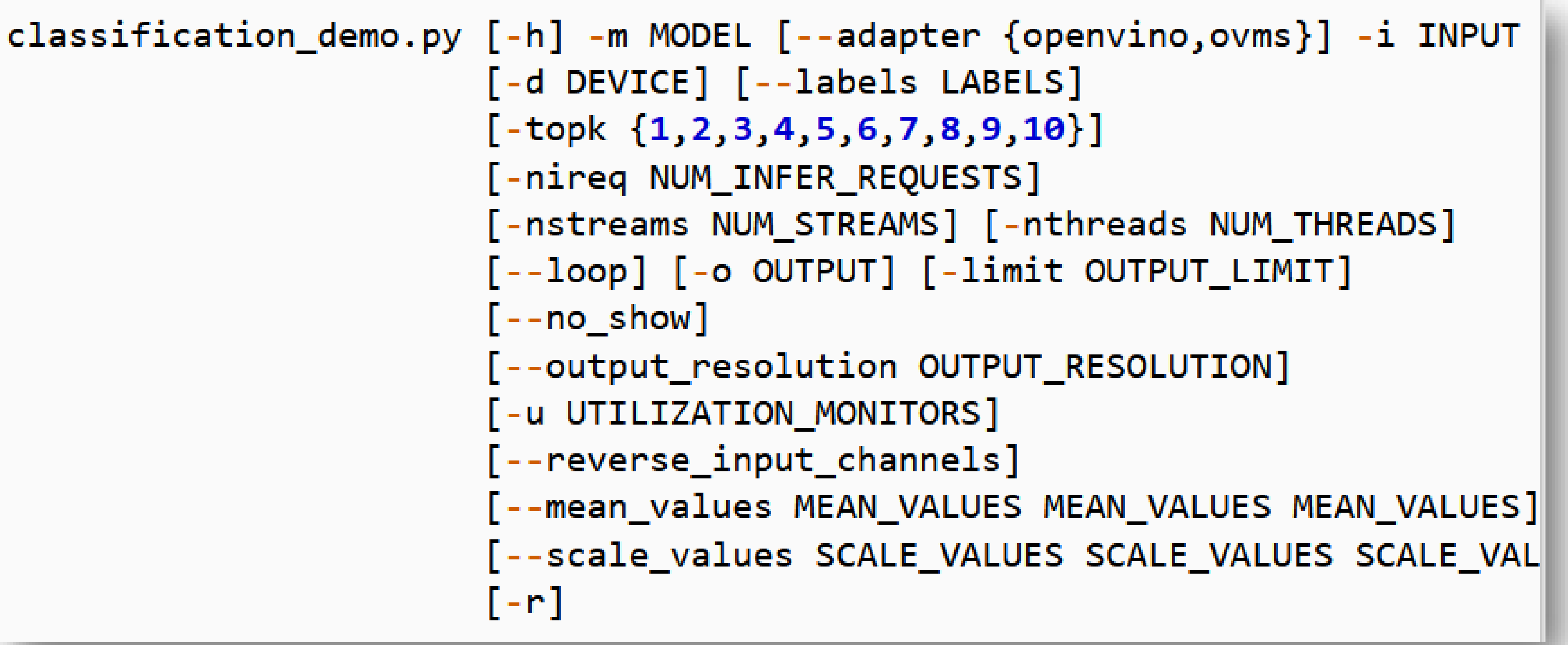


```
Latency: 39.9 ms
FPS: 56.3

        Label:     Score:
  681. notebook     0.54
    620. laptop     0.25
   742. printer     0.11
     662. modem     0.08
592. hard disc      0.02
```

資料來源：https://docs.openvino.ai/2022.1/omz_demos_classification_demo_python.html

classification_demo usage



```
classification_demo.py [-h] -m MODEL [--adapter {openvino,ovms}] -i INPUT
                       [-d DEVICE] [--labels LABELS]
                       [-topk {1,2,3,4,5,6,7,8,9,10}]
                       [-nireq NUM_INFER_REQUESTS]
                       [-nstreams NUM_STREAMS] [-nthreads NUM_THREADS]
                       [--loop] [-o OUTPUT] [-limit OUTPUT_LIMIT]
                       [--no_show]
                       [--output_resolution OUTPUT_RESOLUTION]
                       [-u UTILIZATION_MONITORS]
                       [--reverse_input_channels]
                       [--mean_values MEAN_VALUES MEAN_VALUES MEAN_VALUES]
                       [--scale_values SCALE_VALUES SCALE_VALUES SCALE_VAL
                       [-r]
```

原始碼：
https://github.com/openvinotoolkit/open_model_zoo/blob/master/demos/classification_demo/python/classification_demo.py

資料來源：https://docs.openvino.ai/2022.1/omz_demos_classification_demo_python.html

classification_demo 支援模型清單

- alexnet
- caffenet
- densenet-121
- densenet-121-tf
- dla-34
- efficientnet-b0
- efficientnet-b0-pytorch
- efficientnet-v2-b0
- efficientnet-v2-s
- googlenet-v1
- googlenet-v1-tf
- googlenet-v2
- googlenet-v2-tf
- googlenet-v3
- googlenet-v3-pytorch
- googlenet-v4-tf
- hbonet-0.25
- hbonet-1.0
- inception-resnet-v2-tf
- mixnet-l
- mobilenet-v1-0.25-128
- mobilenet-v1-1.0-224
- mobilenet-v1-1.0-224-tf
- mobilenet-v2
- mobilenet-v2-1.0-224
- mobilenet-v2-1.4-224
- mobilenet-v2-pytorch
- mobilenet-v3-large-1.0-224-tf
- mobilenet-v3-small-1.0-224-tf
- nfnet-f0
- octave-resnet-26-0.25
- regnetx-3.2gf
- repvgg-a0
- repvgg-b1
- repvgg-b3
- resnest-50-pytorch
- resnet-18-pytorch
- resnet-34-pytorch
- resnet-50-pytorch
- resnet-50-tf
- resnet18-xnor-binary-onnx-0001
- resnet50-binary-0001
- rexnet-v1-x1.0
- se-inception
- se-resnet-50
- se-resnext-50
- shufflenet-v2-x0.5
- shufflenet-v2-x1.0
- squeezenet1.0
- squeezenet1.1
- swin-tiny-patch4-window7-224
- t2t-vit-14
- vgg16
- vgg19

參考文獻

- Intel OpenVINO Document

https://docs.openvino.ai/latest/index.html

- Intel Github openvinotoolkit / open_model_zoo

https://github.com/openvinotoolkit/open_model_zoo