OpenVino範例與應用

designed by freepik

Estimated time:
45 min.

學習目標

- 30-1:基本使用方式
- 30-2:影像分類實作
- 30-3:物件偵測實作

30-1:基本使用方式

- 模型下載器
- 模型轉換器

模型下載器的使用

- Intel提供了相當多各種深度學習框架預先訓練好的模型，供開發者下載使用
- 這個網站為Intel公開模型的網站
  - https://download.01.org/opencv/
- 模型檔案通常都不小且有各種版本，透過下載器可有效率的針對想要的模型做下載

模型下載器的使用

- 模型下載器預設的位置在這個目錄

Local Disk (C:) › Program Files (x86) › IntelSWTools › openvino_2020.1.033 › deployment_tools › tools › model_downloader

- 檔名為downloader.py
- 使用方式為:
  - python downloader.py --name [模型名稱] --output_dir [儲存目錄]

模型下載器的使用

- 以下指令可條列出目前可供下載的模型
  - **python downloader.py --print_all**
- 若有需要，可藉由這個指令下載所有的模型
  - **python downloader.py --all**

模型轉換器的使用

- 下載回來的模型均為該深度學習套件的原始檔案，必須經過轉換才可讓OpenVino使用
- 若為自己訓練的模型，也可透過這個程式轉換成OpenVino用的格式
- 模型轉換器預設的位置在這個目錄

模型轉換器的使用

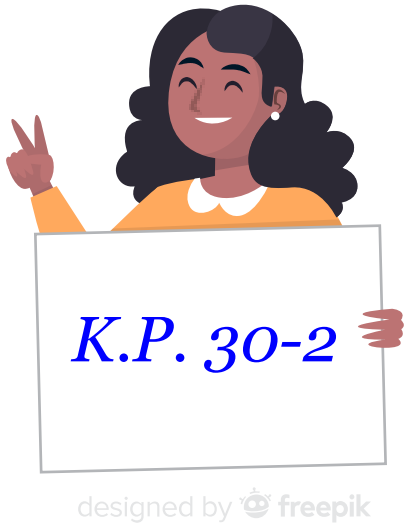
- 根據框架的不同，轉換器的參數也不同
- Caffe模型的轉換方式
  - python mo_caffe.py --input_model <INPUT_MODEL>.caffemodel
- TensorFlow模型的轉換方式
  - python mo_tf.py --input_model <INPUT_MODEL>.pb
- 建議搭配--output_dir參數指定輸出位置
  - 如: python mo.py --input_model alexnet.caffemodel --output_dir C:\models

模型轉換器的使用

- **Non-Frozen TensorFlow模型的轉換方式**
  - **Checkpoint格式**
    - .ckpt + .pb檔案:
      - python mo_tf.py --input_model <INFERENCE_GRAPH>.pb --input_checkpoint <INPUT_CHECKPOINT>
    - .ckpt + .pbtxt檔案:
      - python mo_tf.py --input_model <INFERENCE_GRAPH>.pbtxt --input_checkpoint <INPUT_CHECKPOINT> --input_model_is_text
  - **MetaGraph檔案**
    - python mo_tf.py --input_meta_graph <INPUT_META_GRAPH>.meta
  - **SavedModel(包含.pb檔案與子資料夾variables, assets, and assets.extra等)**
    - python mo_tf.py --saved_model_dir <SAVED_MODEL_DIRECTORY>

30-2:影像分類實作

- 下載必要的模型
- 轉換模型
- 重點程式碼剖析
- 執行程式



下載必要的模型

- 本範例會用到的模型:
  - AlexNet
  - GoogLeNet_v2
- 執行以下指令下載模型(二選一即可)
  - python downloader.py --name alexnet --output_dir C:\models
  - python downloader.py --name googlenet-v2 --output_dir C:\models
- 完成後可從C:\models(或其他指定的目錄)檢查檔案
- 以上兩種模型則一下載使用即可

轉換模型

- 下載回來的模型為TensorFlow(GoogLeNet)或Caffe(AlexNet)的格式
- 透過以下指令轉換模型
  - AlexNet
  - python mo_caffe.py --input_model C:\models\public\alexnet\alexnet.caffemodel --output_dir C:\models
  - GoogLeNet_v2
  - python mo_caffe.py --input_model C:\models\public\googlenet-v2\googlenet-v2.caffemodel --output_dir C:\models
- 轉換過的檔案原則上是以模型名稱作為檔名的.bin與.xml檔案

重點程式碼剖析

- 範例程式碼預設位於

Program Files (x86) › IntelSWTools › openvino_2020.1.033 › inference_engine › samples › python › classification_sample

- 檔案名稱為
  - classification_sample.py

重點程式碼剖析

- 由import的內容可知載入了兩個核心模組
  - IENetwork與IECore
  - IECore: 與Inference Engine溝通的模組
  - IENetwork: 訓練模型的network
- 另外這邊也載入了OpenCV的模組進行辨識圖片的讀取與處理

```
20 from argparse import ArgumentParser, SUPPRESS
21 import cv2
22 import numpy as np
23 import logging as log
24 from time import time
25 from openvino.inference_engine import IENetwork, IECore
```

重點程式碼剖析

- 從這段程式碼可以看出執行的流程分為2大步驟
  - IECore讀入模型>執行模型辨識輸入資料
  - 變數ie為IECore實體
  - 變數net為IENetwork載入模型後的實體
  - 變數res為執行後的輸出資料

```
97      # Loading model to the plugin
98      log.info("Loading model to the plugin")
99      exec_net = ie.load_network(network=net, device_name=args.device)
100
101     # Start sync inference
102     log.info("Starting inference in synchronous mode")
103     res = exec_net.infer(inputs={input_blob: images})
```

執行程式

- 首先要做環境變數初始化
  - cd C:\Program Files (x86)\IntelSWTools\openvino\bin\
  - setupvars.bat
- 本範例程式有幾個主要的參數:
  - --model: 模型xml檔案的路徑
  - --input: 輸入圖檔的路徑
  - --labels: 對應輸出類別的標籤檔路徑
- 由於output只會輸出類別ID，我們需要額外準備對照用的標籤檔
  - 可由此下載: https://gist.github.com/yrevar/942d3a0ac09ec9e5eb3a

執行程式

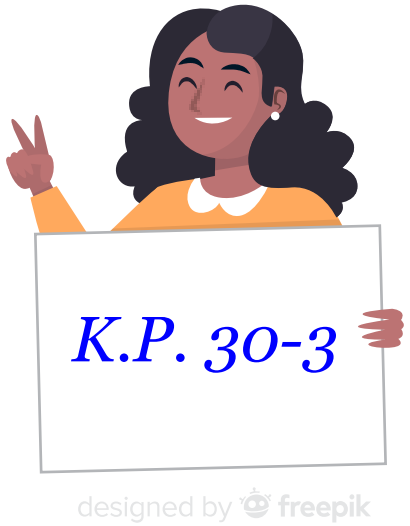
- 執行程式
  - Python classification_sample.py --model C:\models\googlenet-v2.xml --input C:\dog.jpg --labels C:\models\labels.txt
- 輸出內容會類似這樣

```
Image C:\dog.jpg

classid probability
------- -----------
172: 'whippet',0.6220890
44: 'alligator lizard',0.0778558
88: 'macaw',0.0533013
186: 'Norwich terrier',0.0437963
169: 'borzoi, Russian wolfhound',0.0387363
188: 'wire-haired fox terrier',0.0357464
83: 'prairie chicken, prairie grouse, prairie fowl',0.0272263
19: 'chickadee',0.0107987
45: 'Gila monster, Heloderma suspectum',0.0086716
183: 'Kerry blue terrier',0.0082630
```

30-3:物件偵測實作

- 下載必要的模型
- 轉換模型
- 重點程式碼剖析
- 執行程式



下載必要的模型

- 本範例會用到的模型:
  - face-detection-adas-0001
- 執行以下指令下載模型
  - python downloader.py --name face-detection-adas-0001 --output_dir C:\models
- 完成後可從C:\models(或其他指定的目錄)檢查檔案

轉換模型

- 下載回來的模型應該會位於
  - [下載目錄]\intel\face-detection-adas-0001\FP32
- 模型檔案為以下兩個檔案
  - face-detection-adas-0001.bin
  - face-detection-adas-0001.xml
- 此模型是intel自行訓練的人臉偵測模型，本身就為OpenVino用的格式，故不須轉換

重點程式碼剖析

- 範例程式碼預設位於

Program Files (x86) › IntelSWTools › openvino_2020.1.033 › inference_engine › samples › python › object_detection_sample_ssd

- 檔案名稱為
  - object_detection_sample_ssd.py

重點程式碼剖析

- 於以下的行數，可以看到預設是使用opencv的圖片寫入功能將結果存成out.bmp的圖檔

```
176     for imid in classes:
177         tmp_image = cv2.imread(args.input[imid])
178         for box in boxes[imid]:
179             cv2.rectangle(tmp_image, (box[0], box
                2)
180         cv2.imwrite("out.bmp", tmp_image)
181         log.info("Image out.bmp created!")
```

重點程式碼剖析

- 我們先將以下兩行註解

```
176     for imid in classes:
177         tmp_image = cv2.imread(args.input[i
178         for box in boxes[imid]:
179             cv2.rectangle(tmp_image, (box[0
                2)
180         #cv2.imwrite("out.bmp", tmp_image)
181         #log.info("Image out.bmp created!")
```

- 接著新增以下3行的程式碼

```
180         #cv2.imwrite("out.bmp", tmp_image)
181         #log.info("Image out.bmp created!")
182         cv2.imshow("Image", tmp_image)
183         cv2.waitKey (0)
184         cv2.destroyAllWindows()
```

執行程式

- 本範例程式有幾個主要的參數:
  - --model: 模型xml檔案的路徑
  - --input: 輸入圖檔的路徑
- 請自行準備一張有人臉的圖檔

執行程式

- 執行程式
  - **python object_detection_sample_ssd.py --model C:\models\intel\face-detection-adas-0001\FP32\face-detection-adas-0001.xml --input C:\faces.jpg**
- 輸出結果

Demo 30-3

- 模型下載器與模型轉換器
- 實作影像分類
- 人臉偵測

線上Corelab

- 題目1：人臉識別範例
  - 下載預訓練模型face-detection-retail-0004到資料夾C:\models
  - 下載預訓練模型landmarks-regression-retail-0009到資料夾C:\models
- 題目2：人臉識別範例
  - 下載預訓練模型face-reidentification-retail-0095.xml到資料夾C:\models
  - 創建C:/face_gallery資料夾並放入兩張自拍照，照片名稱範例: Hank-0.jpg, Hank-1.jpg
- 題目3：人臉辨識demo
  - 開啟cmd並切換資料夾
  - 執行人臉辨識

本章重點精華回顧

- 基本使用方式
- 影像分類實作
- 人臉偵測實作

Lab:Python 簡介

- **Lab01:模型下載器與模型轉換器**
- **Lab02:實作影像分類**
- **Lab03:人臉偵測**

Estimated time:
20 minutes


THANK YOU