

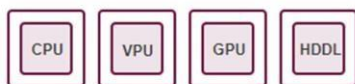
# 【Intel OpenVINO™教学】不用写程序也能轻松上手 AI 模型 分析、优化、部署—DL Workbench 图形化接口工具简介

欧尼克斯实境互动工作室 许哲豪 2021/03/8

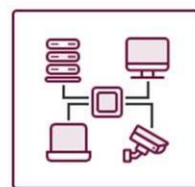
## 不用寫程式也能玩轉深度學習模型 Intel OpenVINO DL Workbench



Analyze



Optimize



Deploy

不用写程序也能玩转深度学习模型

不用写程序也能玩转深度学习模型

看过或玩过「神奇宝贝（宝可梦）」的朋友都知道，训练家们要根据怪兽的属性（火系、草系、电系等）给予适当的培训，再挑选合适的对象及不断的对战磨练才能有好的战斗表现。同样地，AI 模型训练师也是如此，依据不同的应用场景（图像分类、对象检测、图像分割等），挑选适合在指定的硬件（CPU, GPU, VPU, FPGA 等）上推理的模型，并反复调整合适的超参数，如此才能有好的成果。

由于这样的调参工作需要耗费大量的时间及精力才能完成，传统上只靠一堆数字报表来分析调整方向，所以非常仰赖专业工程师的能力才能完成。因此 **Intel OpenVINO Toolkit**（以下简称 **OpenVINO**）在 2019 R3 版之后提供了一项非常方便的网页式图形化接口工具「**DL Workbench**」(Deep Learning Works Benchmark)，让不会写程序、非专业的工程师也能轻易上手进行模型选用、下载、校正、参数调整、效能比较、瓶颈分析、优化及打包部署到目标硬件平台上。

接下来就从“如何安装 **DL Workbench**”、“**DL Workbench** 工作流程”进行简单说明，最后再以“实际案例操作：**SSD 对象检测**”完整说明 AI 模型的分析、优化及部署的步骤。

### \* 如何安装 DL Workbench

目前 OpenVINO 提供多种安装方式，其中最简单的方式就是直接到 Docker Hub 下载(Pull)映像(Images)。如果还不熟悉如何使用 Docker 安装步骤的朋友，可以参考“‘Intel OpenVINO 教学’如何利用 Docker 快速构建 OpenVINO 开发环境” [1]。当然如果想直接在命令行下安装 DL Workbench 的朋友也可参考官网说明[2]。不过这里提醒一下，DL Workbench 只有在 Linux(Ubuntu 18.04/20.04) Docker 环境下才能完整使用 Intel 系列的 CPU, iGPU, Myriad(NCS2), HDDL(VPU)等硬件，而在 Windows 和 macOS 使用 Docker 的朋友则只能使用 CPU。以下说明皆以 Linux 环境做为说明。

如果使用 Docker 安装，只需按照下列指令操作即可完成安装。接着启动 DL Workbench 工作环境，启动时可根据不同需求加上 `-ENABLE_GPU` 或 `-ENABLE_MYRIAD` 或 `-ENABLE_HDDL` 指令来增加硬件推理能力，如 Fig. 1 所示。

推理

```
# 从 Docker Hub 下载 DL Workbench 映像
docker pull openvino/workbench:latest

# 下载完成后，检查可工作映像
docker images

# 第一次启动须先下载批处理文件 start_workbench.sh，并增加批处理文件
可执行权限，第二次启动时就不须执这两条命令

wget
https://raw.githubusercontent.com/openvinotoolkit/workbench_aux/master/start_workbench.sh
sudo chmod +x start_workbench.sh

# 启动 DL Workbench CPU 工作环境，如果已经安装好对应的驱动程序，则
另可选择性加上-ENABLE_XXX 参数增加硬件推理能力（XXX 分别对应 Intel
GPU, Neural Compute Stick 2 单颗 VPU, 多颗 Movidius VPU）
./start_workbench.sh -IMAGE_NAME openvino/workbench -TAG latest [-
ENABLE_GPU] [-ENABLE_MYRIAD] [-ENABLE_HDDL]
```

启动后，会出现 Login token（一长串数字可先复制起来备用）及一个本地端网址 (<http://127.0.0.1:5665/?token=xxxxxx>)，接着将鼠标移至网址处按下左键，点选打开链接 (Open Link)，或者直接复制这一长串网址贴到浏览器上，即可进入 DL Workbench 网页式操作画面，如 Fig. 2 步骤 4 所示。若想结束 DL Workbench，除了关闭网页外，还要回到终端按下 [Ctrl + C] 回到普通命令行状态。

```

jackhsu@jackhsu-All-Series: /home/home
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
jackhsu@jackhsu-All-Series:/home/home$ docker images 檢視已安裝映像檔
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
openvino/workbench  latest      3ecda0d86f11  2 months ago  7.61GB
openvino/ubuntu18_dev  latest      7362311fbea9  2 months ago  7.22GB
openvino/ubuntu18_data_dev  latest      f9a08162207c  2 months ago  8.28GB
hello-world          latest      bf756fb1ae65  13 months ago  13.3kB
jackhsu@jackhsu-All-Series:/home/home$ ./start_workbench.sh -IMAGE_NAME openvino
/workbench -TAG latest 啟動DL Wrokbench映像檔
Stopping the old DL Workbench Docker container.
workbench
Removing the old DL Workbench Docker container.
workbench

Starting the DL Workbench with the following arguments:
Image Name: openvino/workbench
Tag: latest
Container Name: workbench
IP: 0.0.0.0
Port: 5665
Detached: False
GPU Enabled: False
MYRIAD Enabled: False
HDDL Enabled: False 系統檢查是否有對應硬體

```

(中間省略)

```

DL Workbench version: 1.0.3357.4646b582
Login token: ce12096ef83d6a6bf70ebf0549c21401757ef5a1ed86 254c
DL Workbench is available at: http://127.0.0.1:5665/?token=5e386c26a9f0c583f1c4d0daed05c62da3cc494059a868a467
Note: authentication via this link is available only once.
After login, it will expire and new link will be generated.

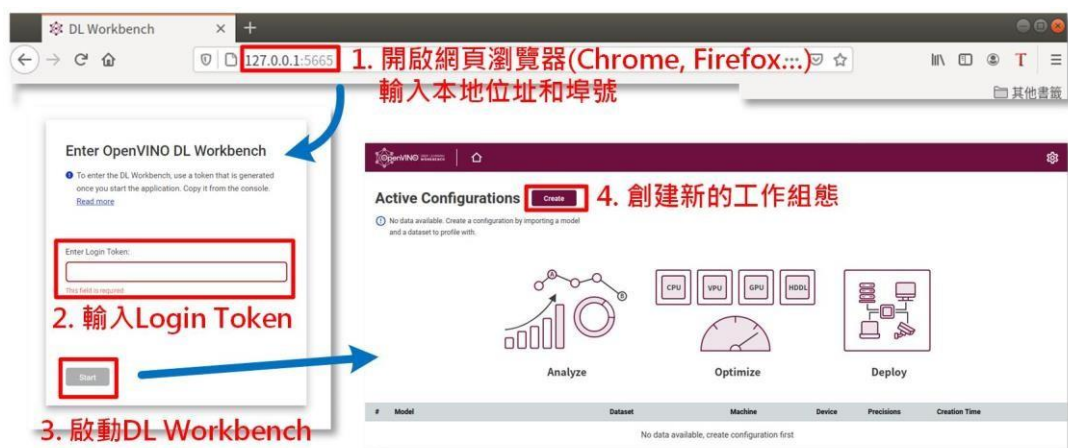
```

OmniXRI Mar. 2021整理繪製

检视已安装映像檔	检查已安装映像
启动 DL Workbench 映像檔	启动 DL Workbench 映像
系统检查是否有对应硬件	系统检查是否有对应硬件
(中間省略)	(中間省略)
登入 Token	登录 Token
鼠标点击左键开启连结(Open Link)	鼠标点击左键打开链接(Open Link)
OmniXRI Mar. 2021 整理绘制	OmniXRI Mar. 2021 整理绘制

Fig. 1 终端命令行成功启动 DL Workbench 界面。(OmniXRI Mar. 2021 整理制作)

如果下次想快速启动 DL Workbench，可不用执行上面的操作，直接开启网页，输入网址 <http://127.0.0.1:5665>，界面会出现要求输入 Login Token，此时再将刚才的 token 贴上，按下“Start”便可直接启动。最后按下网页左上角“Create”创建新的工作配置就可以开始测试了，如 Fig. 2 所示。



OmniXRI Mar. 2021整理繪製

1. 开启网页浏览器(Chrome, Firefox...) 输入本地地址和端口号	1.启动网页浏览器(Chrome, Firefox...) 输入本地地址和端口号
2. 输入 Login Token	2.输入 Login Token
3. 启动 DL Workbench	3.启动 DL Workbench
4. 创建新的工作组态	4.创建新的工作配置
OmniXRI Mar. 2021 整理绘制	OmniXRI Mar. 2021 整理绘制

Fig. 2 快速启动 DL Workbench 网页操作步骤。(OmniXRI Mar. 2021 整理制作)

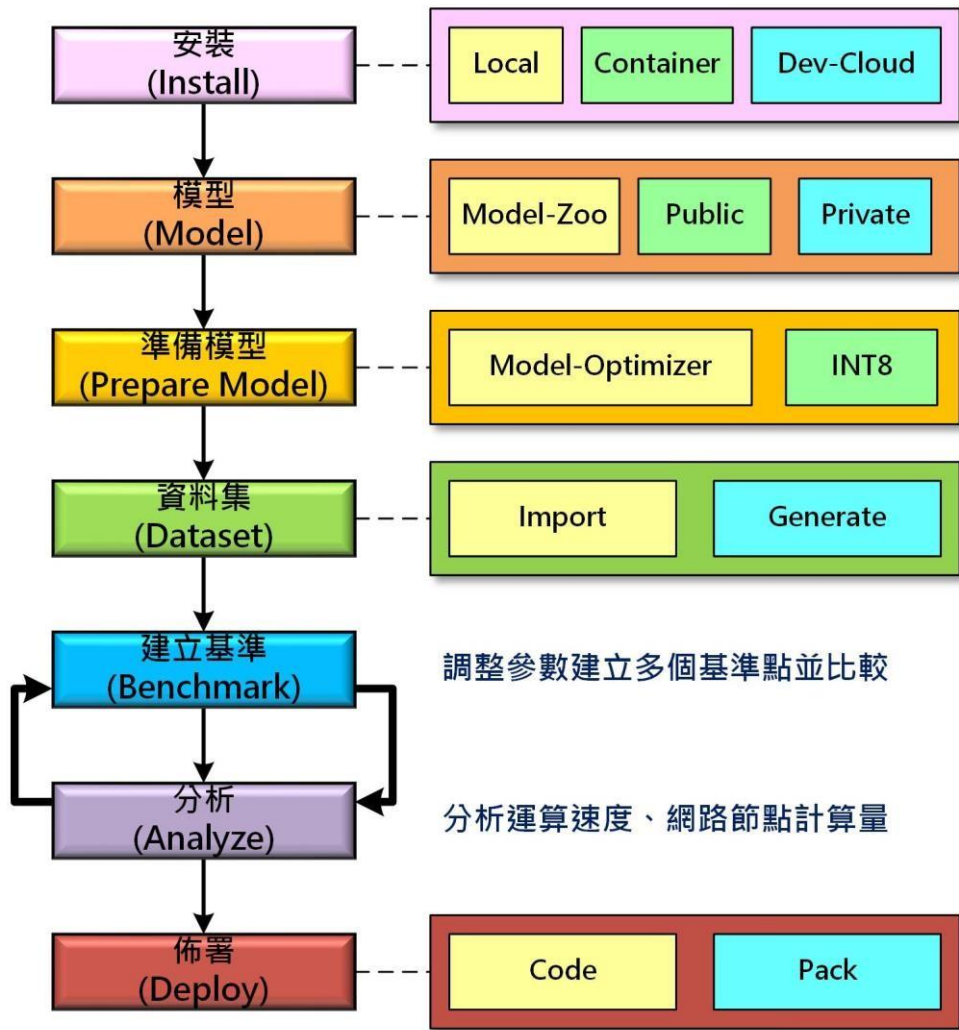
### \* DL Workbench 工作流程

在开始实际操作前，先大致说明一下整个 DL Workbench 的工作流程，方便后面理解实际操作项目，如 Fig. 3 所示。

- 安装(Install):** 前面已经介绍过，除了可直接一步一步安装在计算机上，也可支持使用 Docker 映像方式安装 DL Workbench。另外 OpenVINO 也支持以 Dev-Could 方式安装，方便没有采购硬件的人，先利用 Intel 提供的云端测试平台来选择合适的硬件。
- 模型(Model):** 不会设计 AI 模型的人可通过 OpenVINO 内建的模型下载器(Model Downloader)选用 Intel 已预训练好参数的模型(Pre-Trained Open Model Zoo)，或者使用公开模型(Public)。当然也可支持自己开发并训练好的模型。不过这里只支持 Caffe, MXNet, ONNX, TensorFlow 和 PyTorch 格式的模型转换，最后会都转换成 OpenVINO 支持的中间表示(Intermediate Representation, IR)格式模型网络描述及权重文件(\*.xml, \*.bin)。
- 准备模型(Prepare Model):** 由于导入的模型有可能过大，因此可通过 OpenVINO 内建的优化工具(Optimizer)协助处理网络架构调整以减少计算量。另外，一般训练好的模型参数（网络权重）多半都是使用 32 位浮点数(FP32)来表示，为实现更好的推理效果，这里也可协助校正(Calibration)，将 32 位浮点数(FP32)转换到 16 位浮点数(FP16)，甚至是 8 位整数(INT8)。虽然在不同场景下可能会损失几个百分点的推理精度，但好处是参数所需内存可缩小至 1/2 到 1/4 间，而指令周期则可增加 2 到 3 倍。

4. **数据集(Dataset):** 为了测试选用的模型，这里可以导入(Import)自行准备符合常见的公开数据集（如 ImageNet, Pascal VOC, MS COCO 等）格式的数据集。或者使用生成(Generate)高斯噪声变相产生小型数据集来辅助简易性测试，但请注意这种方式仅适用图像分类模型测试，且因不含标注数据，所以不适用精确度测量及校准工作。如果手上没有现成的数据集，亦可参考官网 [3] 说明，下载常见的 ImageNet, Pascal VOC, Microsoft COCO 等公开数据集。
5. **建立基准(Benchmark):** 准备好模型、数据集及选用欲执行的硬件环境后，便可根据并行流(Parallel Streams)及批次数量(Batch Size)设置值开始进行推理并建立一个基准点(如 Fig. 4a)，其内容包含产出速度(Throughput, Frame Per Second, FPS)、延迟(Latency, ms)。若不满意推理结果，则可重新调整上述参数，建立多个基准点，方便后续比较哪种组合的效果更好。
6. **分析(Analyze):** 为了更容易理解模型每个环节的工作负担，这里提供了许多可视化表格及图形来帮助理解，包含模型主要执行耗时分布(Execution Time by Layer)圆饼图(如 Fig. 4b)、每一层(Layers)的结构、名称、推理耗时、输出精确度(如 Fig. 4c)，模型整体网络节点连接关系图，模型优化前后(IR 及 nGraph) 的网络结构(如 Fig. 4d)，另外还会搭配颜色表示各节点耗时，方便后续模型建立者调整结构及进一步的参数。
7. **部署(Deploy):** 待反复步骤 5 和 6 找到最佳效能点后，便可打包部署到实际硬件上执行，如此就能轻松完成模型分析、优化、部署工作。不过要注意的是这项功能目前仅提供 Ubuntu(Linux)使用。

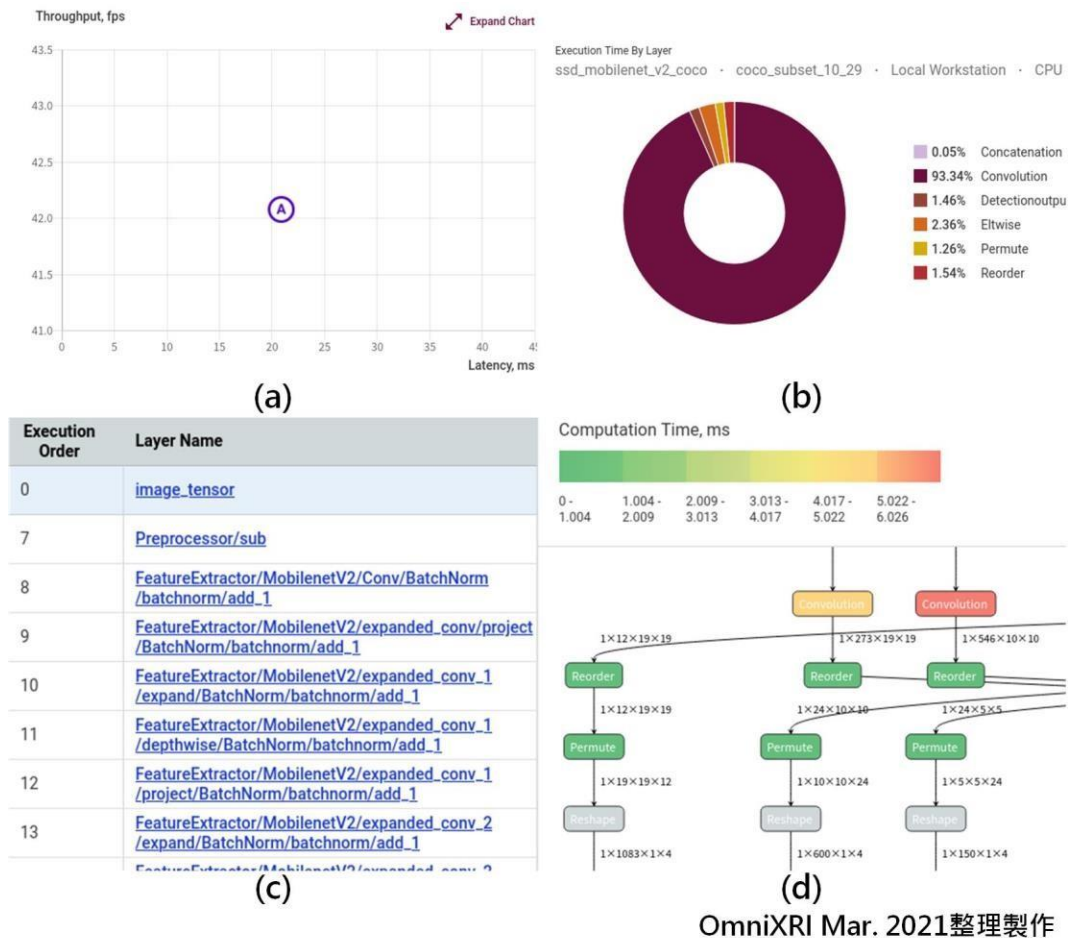
若想更完整了解以上说明步骤，也可参考官方提供的基础教学视频[4]及高级教学视频[5]。



OmniXRI Mar. 2021整理製作

調整參數建立多個基準點並比較	調整參數建立多個基準點並比較
分析指令周期、網路節點計算量	分析指令周期、網路節點計算量
OmniXRI Mar. 2021 整理繪制	OmniXRI Mar. 2021 整理繪制

Fig. 3 DL Workbench 工作流程圖。(OmniXRI Mar. 2021 整理制作)



OmniXRI Mar. 2021整理製作

OmniXRI Mar. 2021 整理绘制 | OmniXRI Mar. 2021 整理绘制

Fig. 4 DL Workbench 可视化工具, (a)输出速度/延迟时间基准点图, (b)各层运行时间圆饼图, (c)各层信息表, (d)网络结构图及耗时表示。(OmniXRI Mar. 2021 整理制作)

### \* 实际案例操作: SSD 对象检测

接下来就用一个实际的例子来说明如何操作 DL Workbench。

#### 1. 数据集预处理

如果手上没有现成的数据集可供测试,可以参考官网[6]的说明下载需要的公开数据集及标注档(Annotations)。这里以 COCO 2014 数据集举例说明。点选网页[6]上 Download COCO Dataset 下的[2014 Val images](约 6.2G Byte)和[2014 Train/Val annotations](约 241M Byte)下载图像集 (val2014.zip) 和标注文档 (annotations\_trainval2014.zip)。

接着建立一个文件夹(/home/<用户名称>/Work)来存放这两个文件。

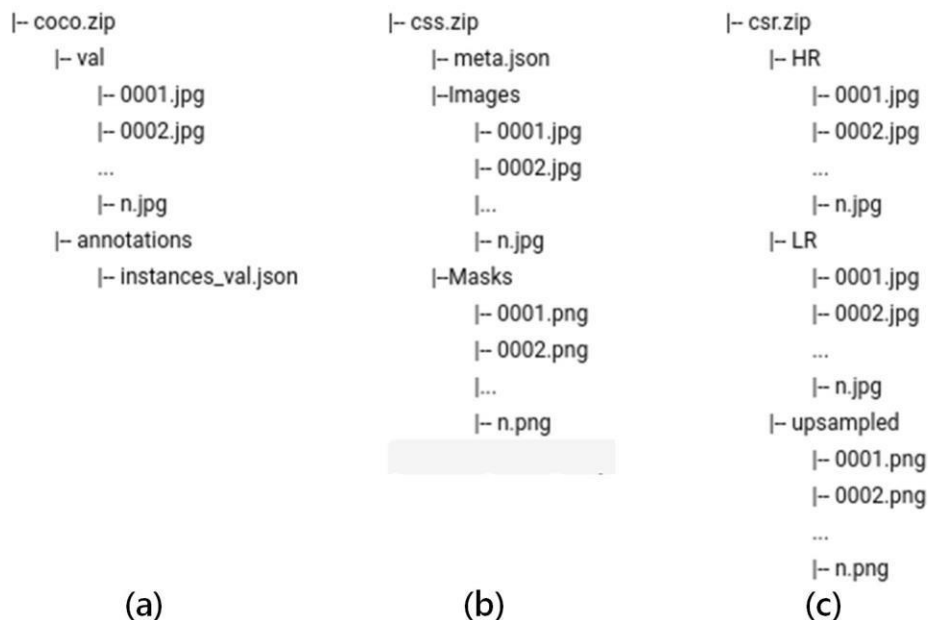
选择网页[6] Cut COCO Dataset 下的 [the script to cut datasets](cut\_dataset.py)另存新文件到刚才建立的文件夹中。最后执行数据集减量(cut)操作,只保留少部份文件,方便 DL

Workbench 导入(import)测试，完成后会自动产生新的压缩文件并存放在 /home/<使用名称>/Work/subsets/coco\_subset\_10\_29.tar.gz。完整执行命令如下所示。

```
# 先将数据集和标注文件及剪切程序复制到/home/用户名称/Work 路径下
# 再执行 cut_dataset.py，根据指定路径 (*_dir) 获得源文件，设置指定
# 数量 (output_size)的图像，最后输出到 output_archive_dir 路径下。
# 这里的 python 是指 3.x 不是 2.x，若计算机中同时有两种版本请改用
python3

python /home/<User Name>/Work/cut_dataset.py \
    --source_images_archive_dir=/home/<User Name>/Work/val2014.zip \
    --source_annotations_archive_dir=/home/<User
Name>/Work/annotations_trainval2014.zip \
    --output_size=20 \
    --output_archive_dir=/home/<User Name>/Work/subsets \
    --dataset_type=coco \
    --first_image=10
```

如果想自行准备数据集，可参考官网[7]，其中 Common Objects in Context (COCO)就有说明 COCO 文件安排格式，如 Fig. 5 所示。



OmniXRI Mar. 2021整理製作

OmniXRI Mar. 2021 整理绘制	OmniXRI Mar. 2021 整理绘制
------------------------	------------------------

Fig.5 DL Workbench 自备 COCO 数据集格式，(a)一般分类/对象检测，(b)图像语义分割，(c)超分辨率。(OmniXRI Mar. 2021 整理制作)



## 2. 导入模型及转换参数精度

为了方便后续同时测试 CPU 及 Neural Compute Stick 2 (MYRIAD), 可使用下列命令启动 DLWorkbench。

```
./start_workbench.sh -IMAGE_NAME openvino/workbench -TAG latest -  
ENABLE_MYRIAD
```

不过这里经测试后发现, 如果你的计算机上没安装标准版的 Distribution of OpenVINO toolkit for Linux[8], 而只有安装 Linux Docker 环境时, 将无法使用 iGPU 和 NCS2(MYRIAD)。如果使用 Windows Docker 环境 (Windows 10 + Windows Subsystem for Linux 2 [WSL2] + Ubuntu 18.04) 时则只能使用 CPU, 因为目前微软在 WSL2 上 GPU 和 USB 驱动程序支持性仍不完整。

接着就可以进入 DL Workbench 网页操作界面, 开始设置主动配置(Active Configurations)。这里需要设置四个项目, 包括模型(Model)、目标(Target)、环境 (Environment)及数据集 (Dataset)。使用前要确保网络畅通, 因为接下来的工作需要从网络上下载许多数据。

首先按下“Create”键创建一组新的配置, 导入(Import)现成的公开模型及预训练好的模型, 由于在线预训练好的模型项目太多, 可以直接输入模型名称快速检索想要的模型。由于这个范例要测试对象检测功能, 所以选用「ssd\_mobilenet\_v2\_coco」, 再来选择模型的参数精度 (FP16/FP32) 进行转换(Convert), 完成后就会在模型区出现模型名称、产生日期、参数精度、模型大小, 状态区会显示是否还在传输中, 完成后会产生一个绿色的对勾。完整流程如 Fig. 6 所示。

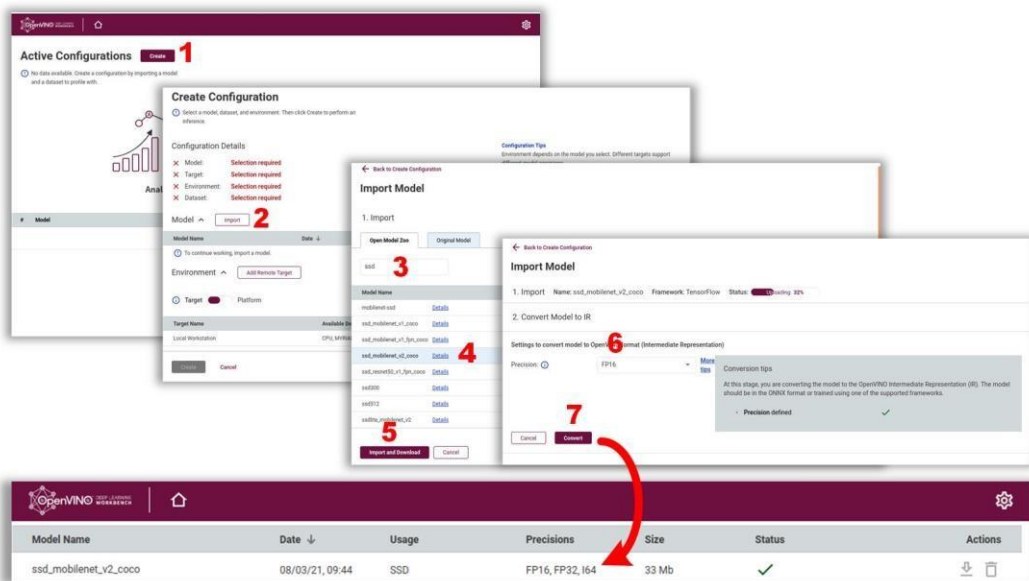
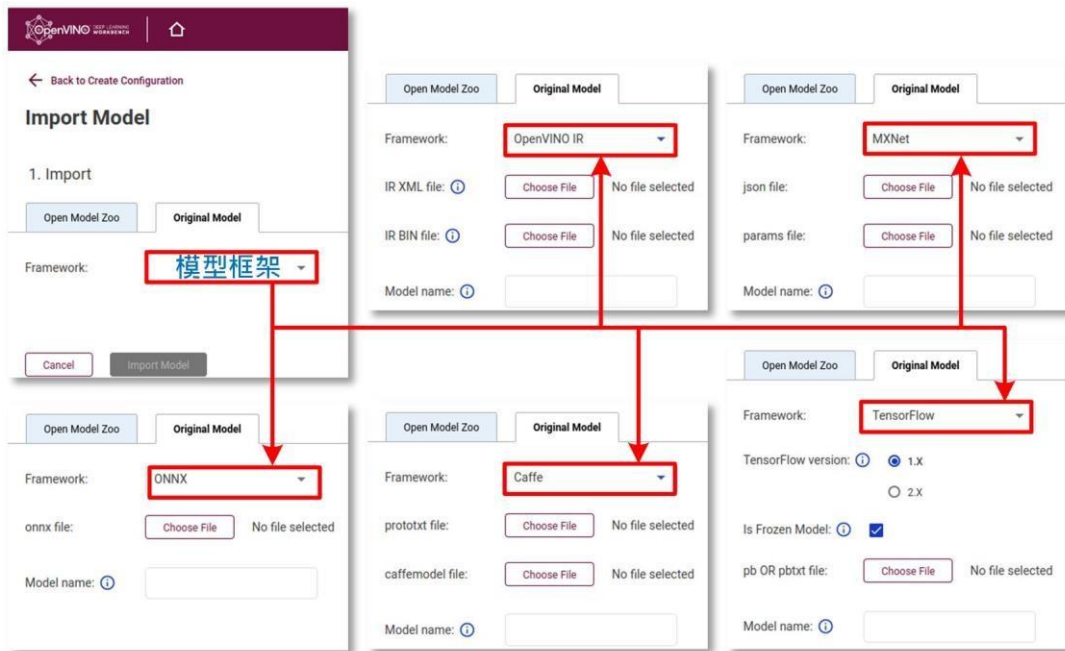


Fig. 6 设置 DL Workbench 推理模型流程。(OmniXRI Feb. 2021 整理制作)

如果不想使用现成的模型，也可切换到原始模型(Original Model)页面，根据系统要求自行提供对应格式的模型（网络）及参数（权重值），再按“Import Model”即可导入自定义模型。

目前可支持 OpenVINO IR, Caffe, MXNet, ONNX, TensorFlow 等格式。



OmniXRI Mar. 2021整理製作

模型框架

Fig. 7 DL Workbench 导入自定义模型。(OmniXRI Mar. 2021 整理制作)

### 3. 导入数据集

點選 Validation Dataset 旁的“Import”键，进入导入验证数据集页面，按“Choose File”键，选择步骤 1 产生的 coco\_subset\_10\_29.tar.gz，按下“Import”，即可完成导入数据集。如 Fig. 8 所示。

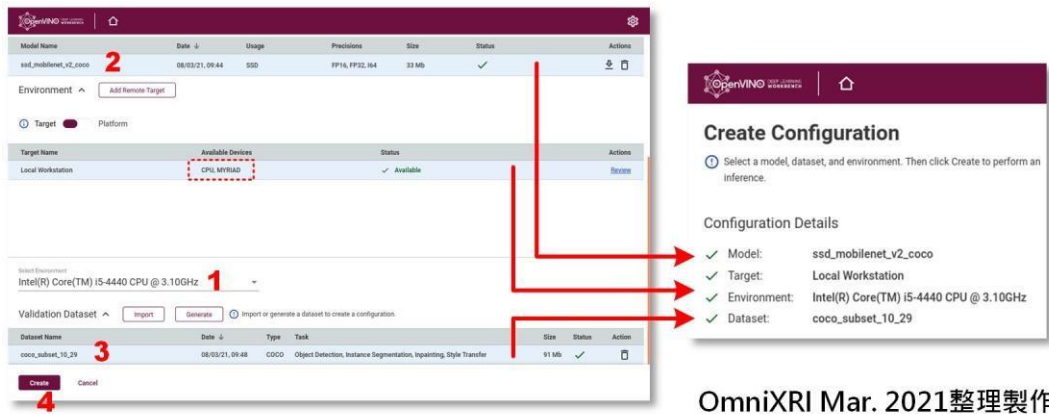


OmniXRI Mar. 2021整理製作

Fig. 8 DL Workbench 导入数据集。(OmniXRI Mar. 2021 整理制作)

## 4. 设置目标/环境

这里可看到目前有 CPU 和 MYRIAD (NCS2)两种装置可供选择，可先选择其中一种 (CPU)，点击模型和数据集后即可按下页面最下方的“Create”建立第一种配置。如 Fig. 9 所示。这里要注意的是，模型及数据集没有准备好之前（尚在传输或转换中）是无法设置目标及环境。



OmniXRI Mar. 2021 整理绘制

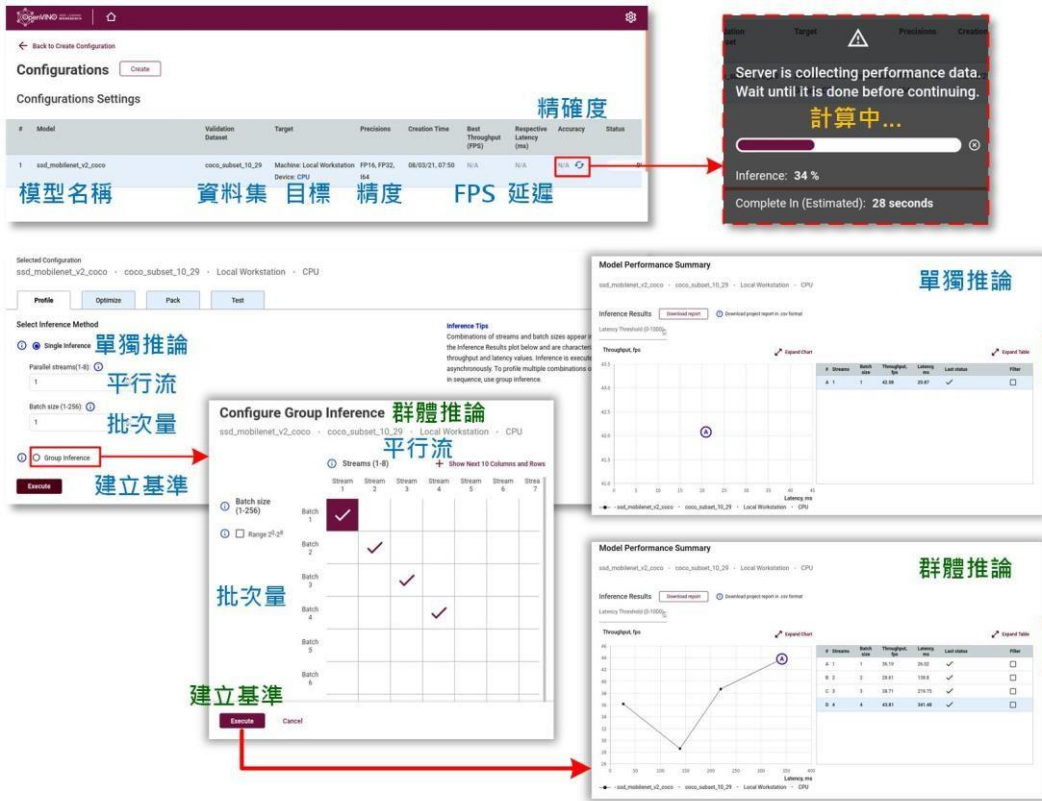
OmniXRI Mar. 2021 整理绘制

Fig. 9 DL Workbench 设置目标环境并建立配置。(OmniXRI Mar. 2021 整理制作)

## 5. 建立基准点

按下“Create”建立第一组配置后，就会进入配置页面，并进行第一次推理，自动产生第一个基准点。推理时根据执行硬件的不同，可能需要等待一小段时间，完成后会得到最佳的输出速度(Best Throughout, FPS)、延迟时间(Respective Latency, ms)及推理精确度(Accuracy)，并在效率综合图表上绘出一个参考点。

通常预设为单独推理(Single Inference)，用户可自定义要测试的并行推理 (Parallel Infers) 及批次大小(Batch Size)，再按下“Execute”键来测试不同条件的推理效果。如果觉得一直手动调整太麻烦，也可使用组推理(Group Inference)，直接点击要测试的组合，按下“Execute”键，然后泡杯咖啡慢慢等结果出来。从产出的图表就可一眼看出哪种组合的效果最好，做为后续打包输出的依据，如 Fig. 10 所示。



OmniXRI Mar. 2021整理製作

精确度	精确度
模型名称	模型名称
数据集	数据集
目标	目标
精度	精度
FPS 延迟	FPS 延迟
计算中	计算中
单独推理	单独推理
平行流	并行推理
批次量	批次大小
建立基准	建立基准
群体推理	组推理
群体推理	组推理
OmniXRI Mar. 2021 整理绘制	OmniXRI Mar. 2021 整理绘制

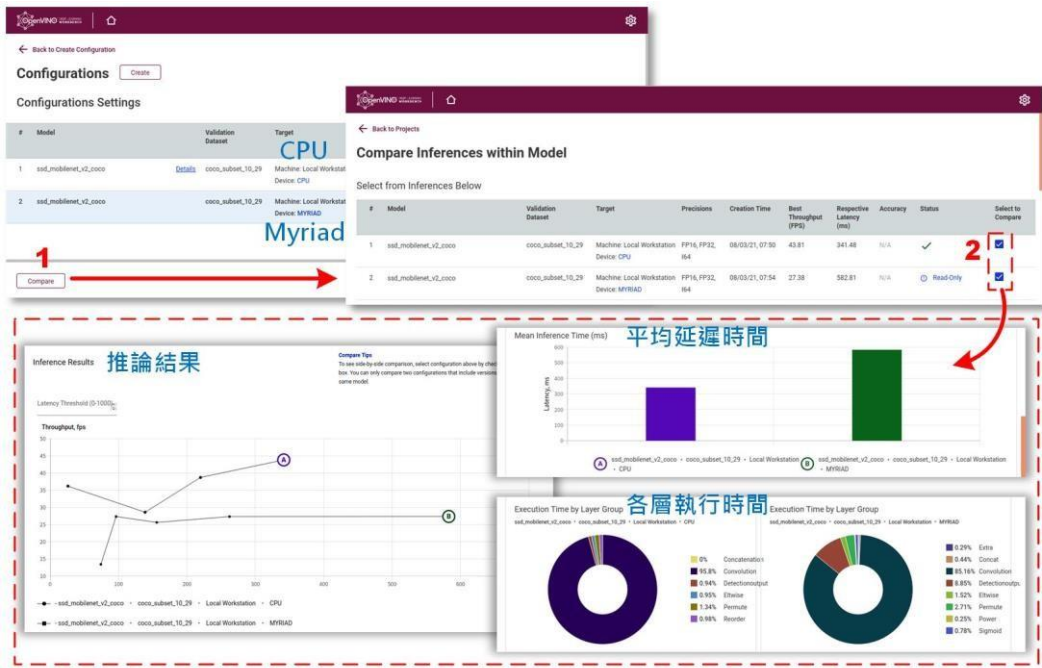
Fig. 10 DL Workbench 建立基准点及结果图。(OmniXRI Mar. 2021 整理制作)

## 6. 基准点效能比较

测试效果除了可以使用单独或组测试外，也可进行多种硬件的推理效果比较。首先点击页面上方的“Create”重新回到步骤 4，此时只需改选装置为[Intel Movidius Myriad X VPU] (NCS2)，其它模型和数据集不变，移到页面最下方按下“Create”键就能新增一组配置，再次回到步骤 5。此时可参考步骤 5，将所有操作重复一遍，得出相同参数结果方便进行比较。

接着按下“Compare”就能进入比较页面进行两组配置的测试结果比较，若有两组以上配置，

则一次只能勾选两组进行比较。这里除了基本效果图表外，还有平均延迟时间、各层运行时间及更多图表信息，可根据需求自行参考，如 Fig. 11 所示。



OmniXRI Mar. 2021整理製作

推理结果	推理结果
平均延迟时间	平均延迟时间
各层运行时间	各层运行时间
OmniXRI Mar. 2021 整理绘制	OmniXRI Mar. 2021 整理绘制

Fig. 11 DL Workbench 多配置基准点效能比较。(OmniXRI Mar. 2021 整理制作)

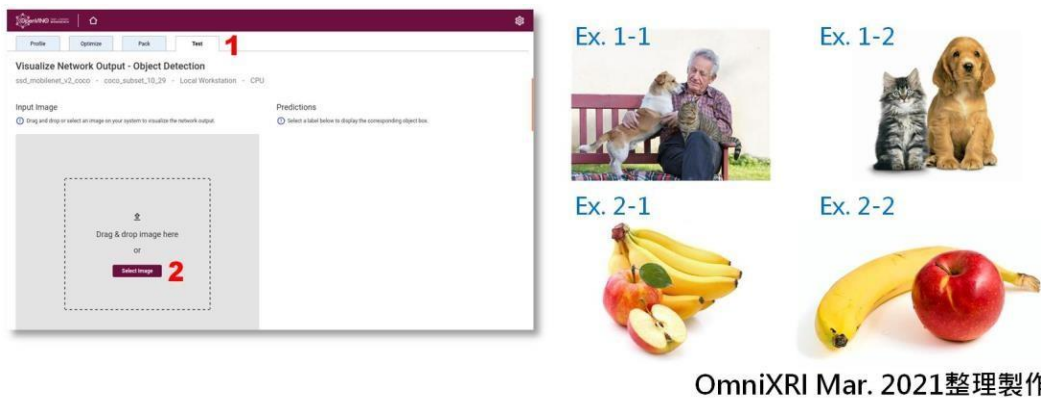
## 7. 上传图片测试

经过上面一连串设置及找到最优配套后就可以来测试一下模型推理能力，这里支持直接上传单张测试图像来实验。首先切换到“Test”页面，按下“Select Images”上传待测试的图像，或者直接把图像拉到这个框里面。接着按下“Test”进行测试，预测(Predictions)结果就会出现在右方。显示结果内容包括一组可调的临界值 (Threshold)和高于临界的对象，同时会显示对象的分类编号(Class ID #)和置信度 (Confidence)。点击对象编号时，左侧还会绘制出对应的内容。若觉得检测出的对象太少，可将临界值调低，再确认一下。

另外目前 COCO 检测结果只有用数字编号(Class #00)表示，并没有直接卷标(Label)文字标示，不容易直接确认结果。如果想更确认编号对应内容，可参考[9]说明。一般 COCO 有 91 分类和 80 分类（从 91 类删除部份），而现在对应的是 91 分类的标签。

接着就用几张图来实验一下结果，第一组猫和狗，一张对象重迭(Ex. 1-1)、一张物件分开(Ex. 1-2)。第二组苹果和香蕉，一张对象重迭(Ex. 2-1)、一张物件分开(Ex. 2-2)。初步实验结果可

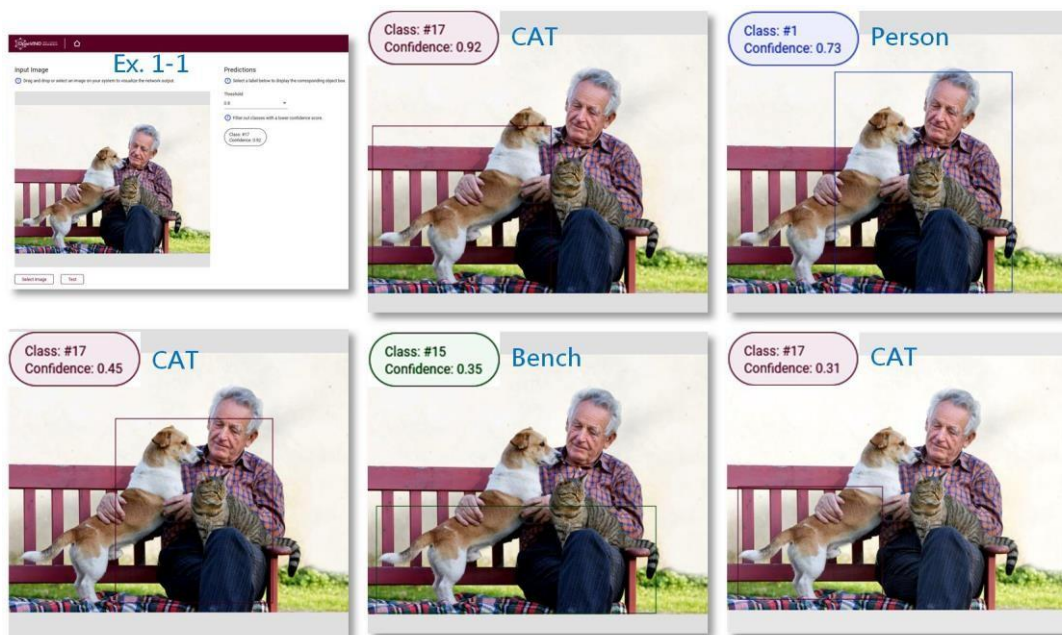
看出 SSD\_MobileNet\_v2\_COCO 这组预训练模型对于重迭的对象似乎分辨能力较弱，而较大的对象则辨识能力尚可接受。不过这样的实验数据太少，所以结论可能不完全正确，仅供参考。测试内容如 Fig. 12 所示，而测试结果如 Fig. 13~16 所示。



OmniXRI Mar. 2021整理製作

OmniXRI Mar. 2021 整理绘制	OmniXRI Mar. 2021 整理绘制
------------------------	------------------------

Fig. 12 DL Workbench 进行对象检测流程及测试样本图像。(OmniXRI Mar. 2021 整理制作)



OmniXRI Mar. 2021整理製作

OmniXRI Mar. 2021 整理绘制	OmniXRI Mar. 2021 整理绘制
------------------------	------------------------

Fig. 13 Ex. 1-1 对象检测实验结果。(OmniXRI Mar. 2021 整理制作)



OmniXRI Mar. 2021整理製作

OmniXRI Mar. 2021 整理绘制	OmniXRI Mar. 2021 整理绘制
------------------------	------------------------

Fig. 14 Ex. 1-2 对象检测实验结果。(OmniXRI Mar. 2021 整理制作)



OmniXRI Mar. 2021整理製作

OmniXRI Mar. 2021 整理绘制	OmniXRI Mar. 2021 整理绘制
------------------------	------------------------

Fig. 15 Ex. 2-1 对象检测实验结果。(OmniXRI Mar. 2021 整理制作)

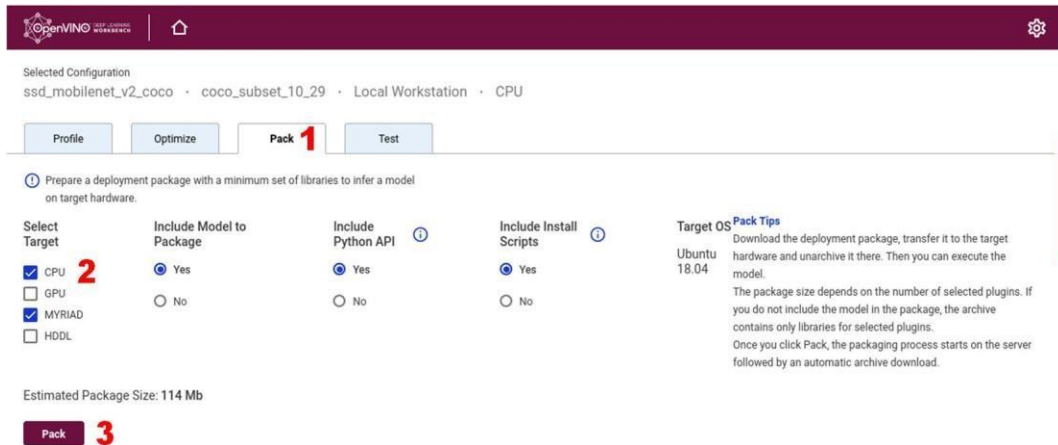


OmniXRI Mar. 2021整理製作

Fig. 16 Ex. 2-2 对象检测实验结果。(OmniXRI Mar. 2021 整理制作)

## 8. 打包输出

为了后续更方便地部署到对应目标工作环境，并且让编程人员能直接编写及呼叫对应函数，这里还提供打包输出，不过目前只支持 Linux。打包内容包含所选择 执行目标（硬件）相关函数、模块、Python API 及安装脚本。首先切换到“Pack”页面，勾选所需项目，再按下“Pack”就会开始打包成一个压缩文件(\*.tar.gz)并询问要存放的路径，如此即完成所有程序。如 Fig. 17 所示。



OmniXRI Mar. 2021整理製作

OmniXRI Mar. 2021 整理绘制	OmniXRI Mar. 2021 整理绘制
------------------------	------------------------

Fig. 17 DL Workbench 打包选项及程序。(OmniXRI Mar. 2021 整理制作)

## \* 小结

通过以上的说明及实例操作，相信大家应该对 Intel OpenVINO Toolkit 中的 DL Workbench 有了一些初步的认识，图形化的接口更是让用户可以轻松上手，一目了然。对于那些不会或不想编程就想要快速实验想法的人，及需要耗费大量心力调参的工程师来说，该工具确实缩短了开发及测试时间。后续还有更多优化及提升推理效果的工作，敬请期待。

## \* 参考文献

[1] 许哲豪，”【Intel OpenVINO 教学】如何利用 Docker 快速创建 OpenVINO 开发环境”  
<http://omnixri.blogspot.com/2021/01/intel-openvinodockeropenvino.html>

[2] Intel OpenVINO Toolkit, "Install the DL Workbench"  
[https://docs.openvino toolkit.org/latest/workbench\\_docs\\_Workbench\\_DG\\_Install\\_Workbench.html](https://docs.openvino toolkit.org/latest/workbench_docs_Workbench_DG_Install_Workbench.html)

[3] Intel OpenVINO Toolkit, "Download and Cut Datasets"  
[https://docs.openvino toolkit.org/latest/workbench\\_docs\\_Workbench\\_DG\\_Download\\_and\\_Cut\\_Datasets.html](https://docs.openvino toolkit.org/latest/workbench_docs_Workbench_DG_Download_and_Cut_Datasets.html)

[4] Intel, "DL (Deep Learning) Workbench | OpenVINO™ toolkit | Ep. 42 | Intel Software"  
<https://youtu.be/20ROqz5j1y8>

[5] Intel, "DL (Deep Learning) Workbench - The Full Flow | OpenVINO™ toolkit | Ep. 43 | Intel Software"  
<https://youtu.be/DaR49bs8qwk>

[6] Intel OpenVINO Toolkit, "Download and Cut Datasets"



[https://docs.openvinotoolkit.org/latest/workbench\\_docs/Workbench\\_DG\\_Download\\_and\\_Cut\\_Datasets.html](https://docs.openvinotoolkit.org/latest/workbench_docs/Workbench_DG_Download_and_Cut_Datasets.html)

[7] Intel OpenVINO Toolkit, "Dataset Types"

[https://docs.openvinotoolkit.org/2021.2/workbench\\_docs/Workbench\\_DG\\_Dataset\\_Types.html](https://docs.openvinotoolkit.org/2021.2/workbench_docs/Workbench_DG_Dataset_Types.html)

[8] Intel OpenVINO Toolkit, "Install Intel® Distribution of OpenVINO™ toolkit for Linux"

[https://docs.openvinotoolkit.org/2021.2/openvino\\_docs/install\\_guides/installing\\_openvino\\_linux.html](https://docs.openvinotoolkit.org/2021.2/openvino_docs/install_guides/installing_openvino_linux.html)

[9] Amikelve Technology Blog, "What Object Categories / Labels Are In COCO Dataset?"

<https://tech.amikelive.com/node-718/what-object-categories-labels-are-in-coco-dataset/>

### \* 延伸阅读

[A] 许哲豪, "【Intel OpenVINO™教学】GStreamer 串流视频智能分析不再慢吞吞—了解 Intel OpenVINO DL Stream 如何加速视频推理" <http://omnixri.blogspot.com/2021/02/intel-openvino-gstreamer-intel-openvino.html>