

【Intel OpenVINO™教学】如何利用 Docker 快速创建 OpenVINO 开发环境

欧尼克斯实境互动工作室 许哲豪(OmniXRI Jack Hsu)

2021/1/19



如何利用 Dockers 快速安装 OpenVINO

如何利用 Dockers 快速安装 OpenVINO

2012 年 Alex Krizhevsky 利用卷积神经网络(Convolutional Neural Network, CNN) AlexNet 以 15.3% Top5 错误率远甩第二名好几条街(26.1%)，赢得 ImageNet 大规模视觉识别挑战赛(ILSVRC)首奖，从此开启人工智能、神经网络、深度学习的新世代。自此各个厂商也纷纷推出自家的开发工具，抢夺人工智能市场。

2018 年 5 月 INTEL 有鉴于边缘智能(Edge AI)设备的硬件效率远低于云端机房，且不同硬件、开发工具间存在严重差异，造成工程师学习上极大的困扰，因此推出开放（免费）计算机视觉推理及神经网络（深度学习）优化工具包「OpenVINO」(Open Visual Inference & Neural Network Optimization Toolkit)[1][2]。

OpenVINO 整合了 OpenCV, OpenCL 等开源计算机视觉及加速运算函数库，并支持多种操作系统(Windows, Linux)，轻松搭配自家各种不同硬件加速运算平台(Intel CPU, GPU, FPGA, VPU, GNA)，且支持各种常见 AI 开发框架(TensorFlow, Caffe, MXNet, ONNX, KALDI)训练出的模型及参数，更有提供模型的优化工具使运算效率及模型大小得到更佳输出，最重要的是工程师只需写一次程序就能在不同平台上执行，大大降低了学习及部署的负担。

早先 OpenVINO 的版本弹性虽然较大，但安装程序冗长，一不小心容易造成步骤缺失导致系统无法正确运行。为解决此问题，OpenVINO 还推出了 Docker 安装方式，轻松搞定安装及执行问题，且可有效隔离不同版本及其它开发环境。接着

就简单说明“什么是 Docker? 和虚拟机 VM 有何不同?”、“如何安装 Docker 工作环境”及“如何使用 Docker 安装 OpenVINO”，最后再通过一个范例来说明“如何利用公开模型 mobilenet 执行图像分类”。

* 什么是 Docker? 和虚拟机 VM 有何不同?

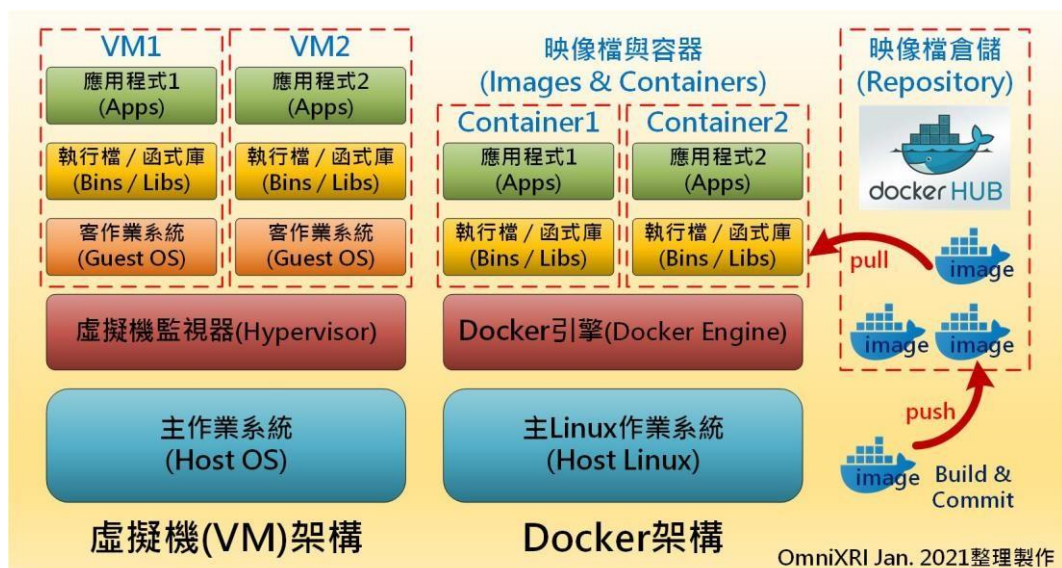
一般大家买来的计算机上大部份都是安装微软的 Windows 操作系统，如果遇到要开发 Linux 相关应用程序时，可能有人会选择安装两个操作系统，在重新启动时选择要执行那一个。虽然这种方式可完全掌控硬件资源（如 CPU, GPU, HDD, RAM）及开发环境（相关套件、函数库），但每次切换操作系统都要重新启动，实在过于麻烦。

为解决这个问题，于是就有人提出在主机操作系统(Host OS)上安装一个虚拟机(Virtual Machine, VM)监视器(Hypervisor)，如 VMWare, VirtualBox 这类工具，方便安装其它操作系统或者多种不同的开发环境，如 Fig. 1 左图所示。虽然这样很方便同时使用两种以上的操作环境，但硬件资源要和主机系统共享且额外多了一些虚拟化转换工作，所以执行效率较差，甚至有些硬件（如 GPU）无法直接使用，导致部份应用程序无法运作。

很多在 Linux 环境下开发程序的人都知道，它的应用程序不像在 Windows 下一个执行文件(*.exe)或搭配几个动态函数库(*.dll)就能顺利执行，各种开发工具及相关套件包常会有版本不兼容问题。本来在 Linux 上也可以执行虚拟机(VM)来区隔不同开发环境，但效率实在太差，所以便有人开发出以一类类似大船载送货柜，货柜装载各种不同大小、形状货物的执行环境，称为 Docker。

如 Fig. 1 中图所示，容器(Containers)就像货柜，是用来装载映像(Images)这样的货物，再由 Docker 引擎（大船承载）执行的概念。表面上看起来和 VM 没什么不同，但最大差别就是它很轻量，不通过虚拟化直接驱动硬件资源，共享操作系统资源，所以启用一个容器和映像仅须数秒，这和加载一个 VM 客操作系统核心要数分钟差异颇大。以往 Docker 只能在 Linux 上运行，近年来已有支持 Window 及 Mac OS 的版本出现了。

一般 Docker 使用上极为方便，供货商仅需将要提供的映像(Images)推上(Push)公开映像文件仓储(Repository / Docker Hub)，用户再去拉下(Pull)想要的映像即可执行，如 Fig. 1 右图所示。



虚拟机(VM)架构	虚拟机(VM)架构
Docker 架构	Docker 架构
OmniXRI, Jan. 2021 整理制作	OmniXRI, Jan. 2021 整理制作

Fig. 1 VM 和 Docker 架构差异比较图。(OmniXRI, Jan. 2021 整理制作)

* 如何安装 Docker 工作环境

为了后面能使用 Docker 快速安装 Intel OpenVINO 工具包及相关环境，这里先介绍如何在 Ubuntu 18.04 / 20.04 (Linux)环境下安装 Docker。

首先启动终端(Terminal)进入命令行操作模式，然后要确定是否有安装旧版的 Docker(docker, docker-engine, docker.io 等)，若有，则执行下列命令进行移除，因为最新版的 Docker 名称已经变更了。

```
sudo apt-get remove docker docker-engine docker.io containerd runc
```

接着下载官方建议的快速安装脚本并进行安装。

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

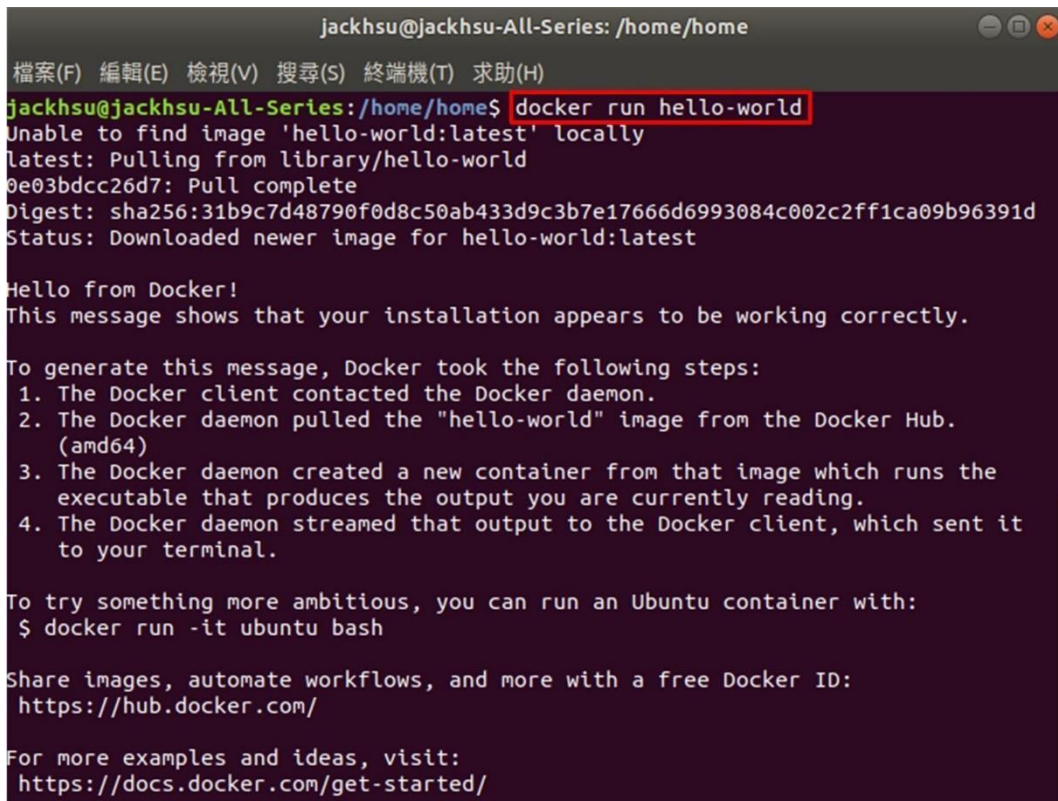
由于 Docker 预设只能在 root(sudo)的权限下执行，为了方便一般用户也能直接呼叫使用，可使用下列命令创建一个 Docker 群组，并将当前用户添加至该群组中，注销系统后再重新登录，后续执行 Docker 相关命令就不用再加上 sudo 了。

```
sudo groupadd docker
sudo usermod -aG docker $USER
```

重新登录后，便可执行下列命令测试 Docker 是否正常。

```
docker run hello-world
```

由于第一次执行，本机端并没 `hello-world` 这个映像，所以会自动到 Docker Hub 寻找并下载。如果成功，会看到下面画面。



```
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 run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:31b9c7d48790f0d8c50ab433d9c3b7e17666d6993084c002c2ff1ca09b96391d
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Fig. 2 Docker 执行 `hello-world` 映像执行结果。(OmniXRI Jan. 2021 整理制作) 后续若有需要除移 Docker Engine 及手动删除相关路径时，则可执行下列命令。

```
sudo apt-get purge docker-ce docker-ce-cli containerd.io
sudo rm -rf /var/lib/docker
```

另外这里补充几个 Docker 常用指令，想了解更多 Docker 操作指令，可参考官网 [3]。

```
# 从 Docker Hub 拉下（下载）映像 [映像名称 xxxxx]
docker pull xxxxx
# 检查目前已下载之映像
```

```
docker images
# 检查目前正在执行的映像
docker ps
# 开始执行映像 [映像名称 xxxxx]
docker start xxxxx
# 停止执行映像 [映像名称 xxxxx]
docker stop xxxxx
```

如果想要知道更多 Docker 安装信息，可参考官网[4]。另外想在 Windows 上安装 Docker 的人，可参考官网[5]。

* 如何使用 Docker 安装 OpenVINO

Intel OpenVINO 对应不同操作系统(Windows, Linux, macOS, RaspbianOS)提供很多种安装方式。近年来随着各种容器技术兴起，也开始支持像 Docker 这类直接安装映像方式，充份解决用户不熟悉安装步骤及满足开发者需要不同版本或开发环境隔离需求。

Intel OpenVINO 目前在 Docker Hub 上提供了多个映像[6]让使用者下载(pull)，其中有两组使用率较高，[openvino/workbench](#) 用户不需编程，直接操作图型化接口就可测试各种模型的效率，[openvino/ubuntu18_dev](#) 则提供一般 Ubuntu 18.04 (Linux) 开发环境，本次测试将以后者为例。

首先执行下列指令将映像文件 `openvino/ubuntu18_dev` 从 Docker Hub 拉下（下载）至本机端。由于文件数量较多且较大（解压缩后约 7.22GByte），所以根据网络速度可能需要等待数分钟到数十分钟不等。

```
docker pull openvino/ubuntu18_dev
```

若想启动映像文件，根据不同硬件执行下列命令。

```
# Intel CPU
docker run -it --rm openvino/ubuntu18_dev
# Intel GPU
docker run -it --rm --device /dev/dri openvino/ubuntu18_dev
# NCS2 (单个 VPU)
docker run -it --rm --device-cgroup-rule='c 189:* rmw' -v
```

```
/dev/bus/usb:/dev/bus/usb openvino/ubuntu18_dev
# HDDL (多个 VPU)
docker run -it --rm --device=/dev/ion:/dev/ion -v /var/tmp:/var/tmp
openvino/ubuntu18_dev
```

启动 Docker 并指定执行的映像文件后命令行提示字符会显示目前 OpenVINO 版本，默认启动时就会执行环境变量设定(setupvars.sh)，且安装好 Python 3.6.9 及 OpenCV 4.5.1_OpenVINO (OpenVINO 删减版)。如果想离开 Docker，则输入 exit 再按 Enter 键即可回到正常命令行状态，如 Fig. 3 所示。若想更完整了解 Docker 在 Linux 上安装方式，可参考官网说明[7]。而 Docker 在 Windows 上安装方式则参考另一份官网说明[8]。

```

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 检查映像檔
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
openvino/workbench  latest      3ecda0d86f11  3 weeks ago   7.61GB
openvino/ubuntu18_dev latest      7362311fbea9  4 weeks ago   7.22GB
hello-world         latest      bf756fb1ae65  12 months ago 13.3kB
jackhsu@jackhsu-All-Series:/home/home$ docker run -it --rm openvino/ubuntu18_dev 执行映像檔於CPU
[setupvars.sh] OpenVINO environment initialized
openvino@f5b127a452d8:/opt/intel/openvino_2021.2.185$ ls OpenVINO版本存放路徑
bin          documentation  install_dependencies  opencv
deployment_tools  inference_engine  licensing              python
openvino@f5b127a452d8:/opt/intel/openvino_2021.2.185$ exit 離開Docker環境
exit
jackhsu@jackhsu-All-Series:/home/home$

```

检查映射档	检查映像
执行映射档于 CPU	执行映像于 CPU
OpenVINO 版本存放路径	OpenVINO 版本存放路径
离开 Docker 环境	离开 Docker 环境

Fig. 3 检查 Docker 本地端映像，启动及离开 Docker 环境。(OmniXRI Jan. 2021 整理制作)

若不想使用 Docker 安装，而想直接在 Linux 上安装 OpenVINO，则可参考另一篇官网说明[9]。

另外提醒一下，如果使用 Docker 进行推理前要注意不同操作系统有不同硬件的最低基本要求[10]，可以参考 Fig. 4 所示。

Minimum Requirements			
PREREQUISITE	LINUX*	WINDOWS*	MACOS*
Operating system	Ubuntu* 18.04. Other Linux distributions, such as Ubuntu* 16.04 and CentOS* 7, are not validated.	Windows* 10	macOS* 10.15 Catalina
CPU	Intel® Core™ i5	Intel® Core™ i5	Intel® Core™ i5
GPU	Intel® Pentium® processor N4200/5 with Intel® HD Graphics	Not supported	Not supported
HDDL, Myriad	Intel® Neural Compute Stick 2 Intel® Vision Accelerator Design with Intel® Movidius™ VPUs	Not supported	Not supported
Available RAM space	4 GB	4 GB	4 GB
Available storage space	8 GB + space for imported artifacts	8 GB + space for imported artifacts	8 GB + space for imported artifacts
Docker*	Docker CE 18.06.1	Docker Desktop 2.1.0.1	Docker CE 18.06.1

Fig. 4 OpenVINO 建议最低推理要求[10]。

* 如何利用公开模型 **MobileNet** 执行图像分类

利用 Docker 安装好 OpenVINO 工作环境后就可以开始进行推理(Inference)工作了。OpenVINO 除了提供多种预训练好的模型外[11]，另外也支持很多不同框架(Caffe/Caffe2, PyTorch, TensorFlow, MXNet, ONNX 等)产生的公开模型[12]，也称为「Open Model Zoo」，其中包含图像分类、对象侦测、图像分割、人脸识别、人体姿态、深度估测、图像填补、风格转移、动作识别、图像着色、声音分类、语音识别、图像翻译等公开模型。

使用这些模型前须先下载并执行优化，最后产生 OpenVINO 推理时所需的中间表示文件(Intermediate Representation, IR)，即模型描述文件(*.xml)及参数权重文件(*.bin)，存放在/opt/intel/opencvino/public/模型名称/FP16(或 FP32, INT8)路径下。接下来就举一个 mobilenet 图像分类的例子来说明。

```
# 启动 Docker 并指定映像 Intel CPU 上执行，以 root 身份执行，同时指令显示器和系统共享，方便 OpenCV 执行 imshow() 能正常工作
docker run -itu root:root --rm -v /tmp/.X11-unix:/tmp/.X11-unix -e
```

```
DISPLAY=$DISPLAY openvino/ubuntu18_dev
```

```
# 下载公开模型 mobilenet-v1-1.0-224
```

```
python3 \
```

```
/opt/intel/openvino/deployment_tools/tools/model_downloader/downloader.py \
```

```
\ --name mobilenet-v1-1.0-224
```

```
# 优化模型 mobilenet-v1-1.0-224 并产出中间表示(*.xml, *.bin)文件,  
FP16/FP32 格式于 /opt/intel/openvino/deployment_tools/public/模型名称  
路径下
```

```
python3 \
```

```
/opt/intel/openvino/deployment_tools/tools/model_downloader/converter.py \
```

```
-name mobilenet-v1-1.0-224
```



Fig. 5 推理测试用图像。

接着输入一张汽车图像（如 Fig. 5 所示），经 mobilenet-v1 PF16 格式推理输出结果，Top 1 为 Car, Sport Car 0.4242045 (42.42%置信度)，成功完成范例。

```
# 以 classification_sample_async.py 进行“图像分类”异步推论
```

```
# -m 指定 mobilenet-v1-1.0-224 模型为 FP16 格式进行推论
```

```
# -i 指定待测试之输入影像
```

```
# --labels 指定输出 ImageNet 1000 分类对应的标签文件
```

```
python3 \
```

```
/opt/intel/openvino/inference_engine/samples/python/classification_sample
```

```
_async/classification_sample_async.py \
```

```
-m /opt/intel/openvino/public/mobilenet-v1-1.0-224/FP16/mobilenet-v1-1.0-
```

```
224.xml \
```

```
-i /opt/intel/openvino/deployment_tools/demo/car.png \
```



```
--labels /opt/intel/openvino/deployment_tools/demo/squeezenet1.1.labels

# 输出结果（部份省略，仅显示 Top 10 结果卷标及置信度 0.0~1.0）
[ INFO ] Top 10 results:
Image /opt/intel/openvino/deployment_tools/demo/car.png

classid probability
-----
car, sport car 0.4242045
convertible0.2728052
wheel      0.1330036
wagon, station wagon, wagon, estate car, beach waggon, station waggon,
wagon 0.0699500
race car, racing car0.0339605
radiator grille0.0263465
pickup truck0.0155464
hack, taxi, taxicab0.0127090
minivan    0.0051457
T          0.0018786
```

* 小结

Intel OpenVINO 目前大约每季度就会有一版更新，官方也有专门的论坛在解决大家的疑难问题，支持的公开模型也越来越多，如果再配合像 Docker 这类容器方式的安装，新手们就更容易上手，使用上也更加方便了。所以，如果您不知道如何发挥自己的 AI 应用，还不赶快来试试。

更多 OpenVINO 相关应用可参考下方“延伸阅读”项目。

提醒：目前 OpenVINO 最新版本为 2021.2，自 2020.4 版本后就不再支持神经运算棒(Movidius Neural Computing Stick, NCS)第一代产品，而 NCS2 第二代产品则不受影响。

参考文献

[1] Get the Intel® Distribution of OpenVINO™ Toolkit
<https://software.intel.com/content/www/us/en/develop/tools/openvino-toolkit/download.html>

[2] OpenVINO™ Toolkit Overview
<https://docs.openvino toolkit.org/latest/index.html>

[3] Docker docs <https://docs.docker.com/>

[4] Install Docker Engine on Ubuntu
<https://docs.docker.com/engine/install/ubuntu/>

[5] Install Docker Desktop on Windows <https://docs.docker.com/docker-for-windows/install/>

[6] Docker Hub – openvino <https://hub.docker.com/u/openvino>

[7] Install Intel® Distribution of OpenVINO™ toolkit for Linux* from a Docker* Image
https://docs.openvino toolkit.org/latest/openvino_docs_install_guides_installing_openvino_docker_linux.html

[8] Install Intel® Distribution of OpenVINO™ toolkit for Windows* from Docker* Image
https://docs.openvino toolkit.org/latest/openvino_docs_install_guides_installing_openvino_docker_windows.html

[9] Install Intel® Distribution of OpenVINO™ toolkit for Linux*
https://docs.openvino toolkit.org/latest/openvino_docs_install_guides_installing_openvino_linux.html

[10] Install the DL Workbench
https://docs.openvino toolkit.org/latest/workbench_docs_Workbench_DG_Install_Workbench.html

[11] Overview of OpenVINO™ Toolkit Intel's Pre-Trained Models
https://docs.openvino toolkit.org/latest/omz_models_intel_index.html

[12] Overview of OpenVINO™ Toolkit Public Models
https://docs.openvino toolkit.org/latest/omz_models_public_index.htm

延伸阅读

[A] 许哲豪，【AI_Column】运用 Intel OpenVINO 土炮自驾车视觉系统
<https://omnixri.blogspot.com/2018/10/aicolumnintel-openvino.html>

[B] 许哲豪，【OpenVINO™教学】土炮图像实例分割型智能监控系统
<http://omnixri.blogspot.com/2019/09/openvino.html>

[C] 许哲豪, 【OpenVINO™教学】土炮体感控制系统
http://omnixri.blogspot.com/2019/09/opencvino_20.html

[D] 许哲豪, 【课程简报分享】20191019_2019 Intel OpenVINO™ x Edge AI 创意应用自造松_OpenVINO 平台运用要领
<https://omnixri.blogspot.com/2019/10/201910192019-intel-openvino-x-edge.html>

[E] 许哲豪, 【课程简报分享】20191026_2019 Intel OpenVINO™ x Edge AI 创意应用自造松_OpenVINO 预训模型使用及应用实例开发
<http://omnixri.blogspot.com/2019/10/201910262019-intel-openvino-x-edge.html>

[F] 许哲豪, OpenVINO 与 OpenCV 搭配—干计算机视觉的活一点都不累
<http://omnixri.blogspot.com/2019/11/opencvinoopencv.html>

[G] 许哲豪, 【OpenVINO™+RealSense™教学】土炮智能机器手臂之视觉系统
<https://omnixri.blogspot.com/2019/12/opencvinorealsense.html>

[H] 许哲豪, 如何在 Google Colab 上安装执行 Intel OpenVINO 的范例
<http://omnixri.blogspot.com/2020/09/google-colabintel-openvino.html>