

2019/11/27 update

ゼロから始めるインストール



どうも、クラゲです。

彼は、IoTに関する技術コンテンツを発信する
電子工作が大好きなクラゲです。

どうも、ディープなクラゲです。

「[OpenVINO™ でゼロから学ぶディープラーニング推論](#)」シリーズの3
回目記事です。

このシリーズは、ディープラーニング概要、OpenVINO™ツールキッ
ト、Neural Compute Stick、RaspberryPiの使い方、Pythonプログラミ
ングをゼロから徹底的に学び、成果としてディープラーニング推論アプ
リケーションが理解して作れるようになることを目指します。



第3回目は、RaspberryPiのOS、OpenVINO™ツールキットのインスト
ールについてゼロから徹底的に説明します。最後には、正しくインスト
ールされているかをサンプルプログラムを実行して確かめます。

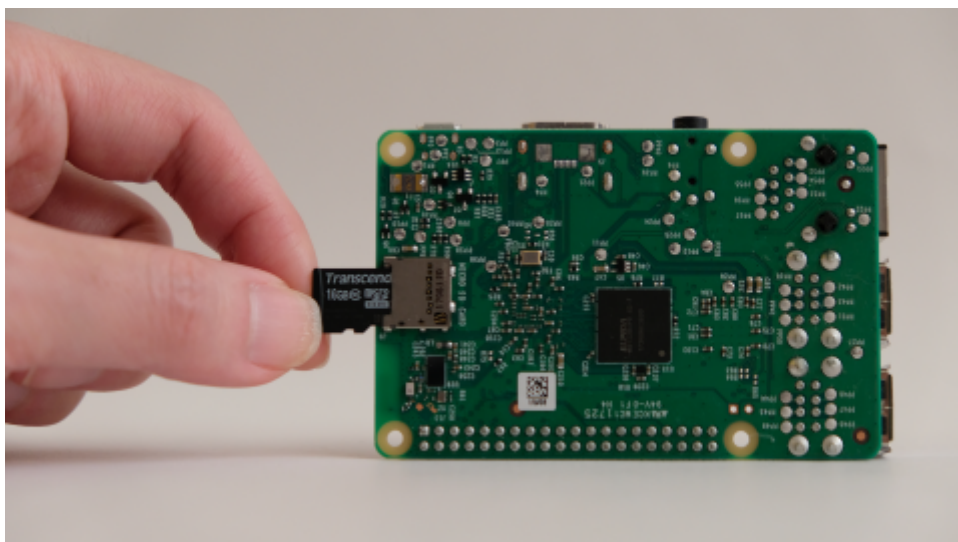
ちなみに、RaspberryPiのOSのダウンロードは、状況にもよりますが1
～2時間くらいかかりますので、事前に実行すると良いと思います。

【 目次 】

- RaspberryPiのOSインストール
- VNCの活用
- OpenVINO™ツールキットインストール
- サンプルのビルドと実行

RaspberryPiのOSインストール

まず始めにラズパイにOSをインストールを行います。



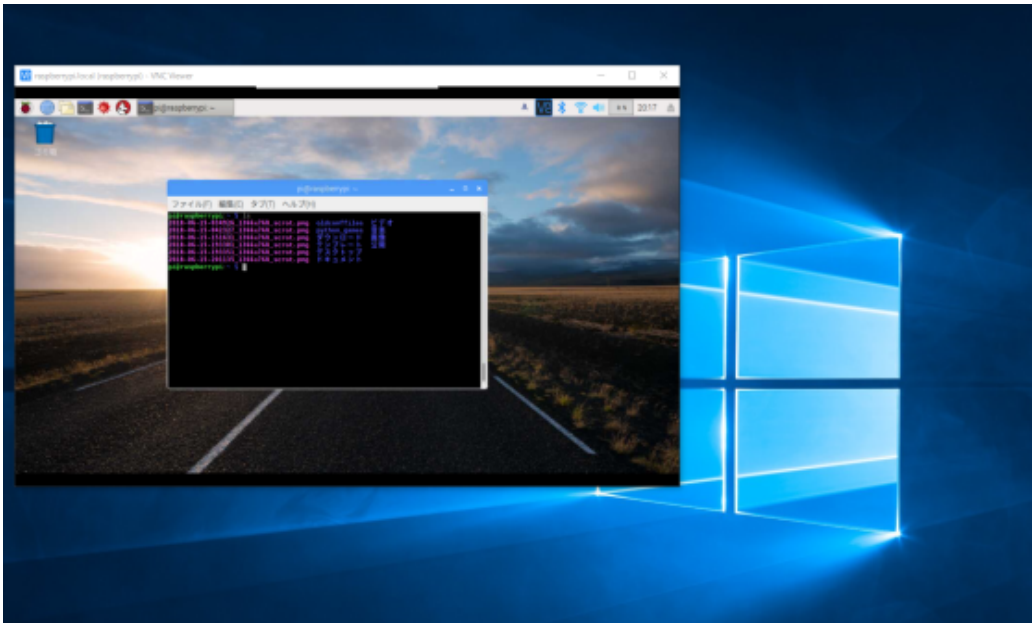
[こちらのページ](#)に詳しく書いてありますので、内容に従ってインストールをお願いします。

以下の全ての項目を実施してください

- OSダウンロード
- SDカードのフォーマット
- OSインストール
- 各種初期設定

VNCの活用

先程のインストールでは、ラズパイに繋いだディスプレイ、キーボード、マウスが必須でしたが、インストール後は"VNC"というものを活用することにより、ラズパイのデスクトップ画面を無線でPCにて表示・操作することが可能になります。



VNCは必須の内容ではありませんが、非常に便利ですので[こちらのページ](#)を参考に設定してみてください。以下3項目です。

- RaspberryPi側の設定
- PC側にVNC Viewerソフトをインストール
- PC側でVNCを立ち上げる

OpenVINO™ツールキットインストール

2019年11月27日現在、OpenVINO™ツールキットの最新バージョンは `2019_R3.1` です。

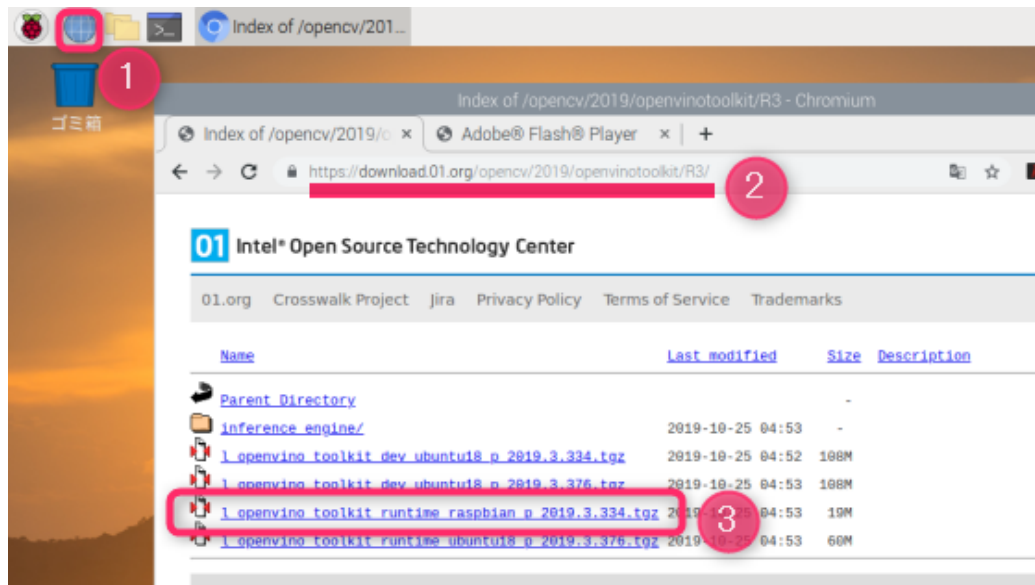
ラズパイへのインストール方法は[こちらの公式ページ](#)に書いてある通りですが、ここではクラゲ視点の解説を交えながらインストール方法を説明します。少しコマンドを使いますが、初心者の方でも分かるようにご説明します。

※ここでは「PC + Raspberry Pi + Neural Compute Stick」環境の方を対象としております。それ以外の「PCのみ」環境の方は[こちら](#)、「PC + Neural Compute Stick」環境の方は[こちら](#)を参照してください

ダウンロード

ラズパイ画面左上の「地球儀アイコン」をクリックしてブラウザを立ち上げて以下のURLをコピー＆ペーストしてアクセスします

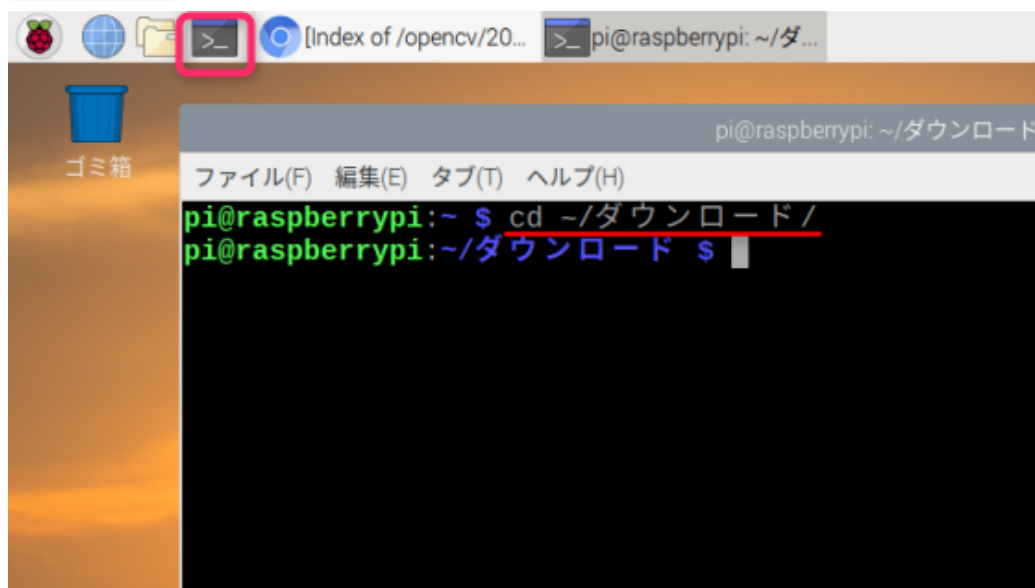
<https://download.01.org/opencv/2019/openvinotoolkit/R3/>



`l_openvino_toolkit_runtime_raspbian_p_2019.3.334.tgz` というファイルをクリックするとダウンロードが自動的に始まります。

ここからは `LXTerminal` を使ってコマンド入力を行います。ラズパイ画面左上の「泣き顔のような黒いアイコン」をクリックすると

`LXTerminal` が立ち上がります。

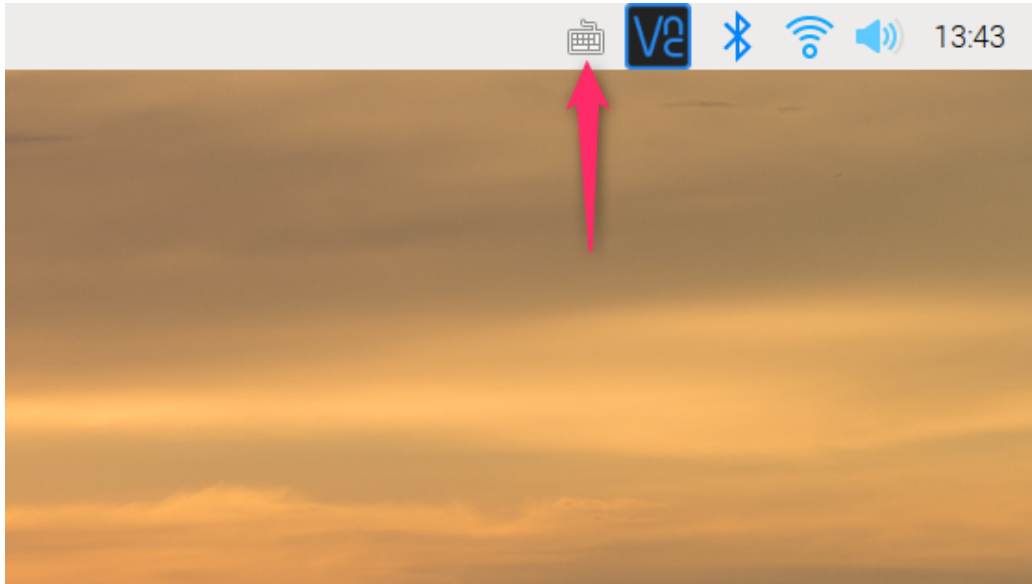


文字が小さいと感じた方は 編集 から Zoon In を選択すると大きくなります。

以下のコマンドを打ち込んでください。 `cd` の後には半角スペースが必要で、`~` は「チルダ」という記号です。 `ダウンロード` という日本語以外は全て半角です。最後にEnterキーを押すことで実行されます。

```
cd ~/ダウンロード/
```

ちなみに日本語入力を行う場合はラズパイ画面右上の「キーボードアイコン」をクリックすると可能になります。



今打ち込んだコマンドのおおまかな意味を説明します。 `cd` はchange directoryのことで、ディレクト（フォルダ）を `~/ダウンロード/` という場所に移動せよ という意味です。 `~` はユーザーのホームディレクトリを指します。

次に以下のコマンドを打ち込んでみて下さい。そのディレクトリにあるフォルダやファイルの一覧が表示されます。先程ダウンロードしたファイルが表示されていればOKです。

```
ls
```

これ以降はコマンドの詳細説明は割愛して、簡単な解説のみ行います。詳細が気になる方は[こちらのページ](#)を参照したり、ネットで検索してみてください。なお、各コマンドを覚える必要は全くありません。コマンドは、打ち間違い防止のためにも基本的にはコピー＆ペーストすることをお勧めします。

本体のインストール

OpenVINO™ツールキットを展開するフォルダを作成します。 `/opt/intel/opencvino` というフォルダを作成します。

```
sudo mkdir -p /opt/intel/opencvino
```

先程ダウンロードしたファイルを `/opt/intel/opencvino` に展開します。コマンドを実行しても何も表示されませんが、裏では動いています。

```
sudo tar -xf l_opencvino_toolkit_runtime_raspbian_p_2019.3.334.tgz --strip
```

フォルダを作成して、ZIPファイルを展開しただけですが、これで本体のインストールは完了です

CMakeのインストール

サンプルコードをビルドするためのCMakeというソフトウェアをインストールします。

続行しますか？ [Y/n] という表示が出てきたら `y` を入力します。たくさん文字が出てきて1分も経たずに終了すると思います。

```
sudo apt install cmake
```

環境変数の設定

`setupvars.sh` スクリプトを実行することによりOpenVINO™ツールキットが動作するように環境を整えます。実行して `[setupvars.sh] OpenVINO environment initialized` と表示されれば成功です。

```
source /opt/intel/opencvino/bin/setupvars.sh
```

これでOpenVINO™ツールキットを使えるようになりましたが、再起動や別のターミナルを開いた場合にも先程のコマンドが毎回必要で面倒です。毎回先程のコマンドを打たなくても良いように、起動時に自動でコマンド実行するように設定します。

```
echo "source /opt/intel/opencvino/bin/setupvars.sh" >> ~/.bashrc
```

このコマンドが反映されているか確認します。

ラズパイ画面左上の「泣き顔のような黒いアイコン」をクリックして別の `LXTerminal` を開いてみて下さい。 `[setupvars.sh]` `OpenVINO environment initialized` と最初に出てくれば成功です。今開いたターミナルは閉じてOKです。

USBルールの追加

現在のユーザーを `users` グループに追加します

```
sudo usermod -a -G users "$(whoami)"
```

Neural Compute Stickを使ってOpenVINO™を実行できるように以下のスクリプトを実行します。 `[install_NCS_udev_rules.sh]` `udev rules installed` と表示されればOKです。

```
sh /opt/intel/openvino/install_dependencies/install_NCS_udev_rules.sh
```

これで、OpenVINO™ツールキットとNeural Compute StickをRaspberryPiで実行する準備が整いました！

サンプルのビルドと実行

[こちらの公式ページ](#)では、サンプル実行として以下の2つの例が記載されていますが、どちらも同じ顔検出です。ここでは1番目のみ実行することとしましょう。

- Build and Run Object Detection Sample
- Run Inference of Face Detection Model Using OpenCV* API

ユーザーのホームディレクトリに `build` というフォルダを作成して移動します

```
mkdir ~/build && cd ~/build
```


サンプルをビルドするためのコマンド前半です。たくさん文字が出て30秒足らずで終了します

```
cmake -DCMAKE_BUILD_TYPE=Release -DCMAKE_CXX_FLAGS="-march=a
```

続けてサンプルをビルドするためのコマンド後半です。今度は5分程度かかります。[100%] Built target object_detection_sample_ssd と出ればビルド完了です。

```
make -j2 object_detection_sample_ssd
```

これで物体検出を行う実行ファイルはできましたが、学習済みモデルと重みが別途必要になります。顔検出が学習済みであるモデルと重みをそれぞれダウンロードします。

まずは学習済み重みファイルです。保存完了 などと出ればOKです

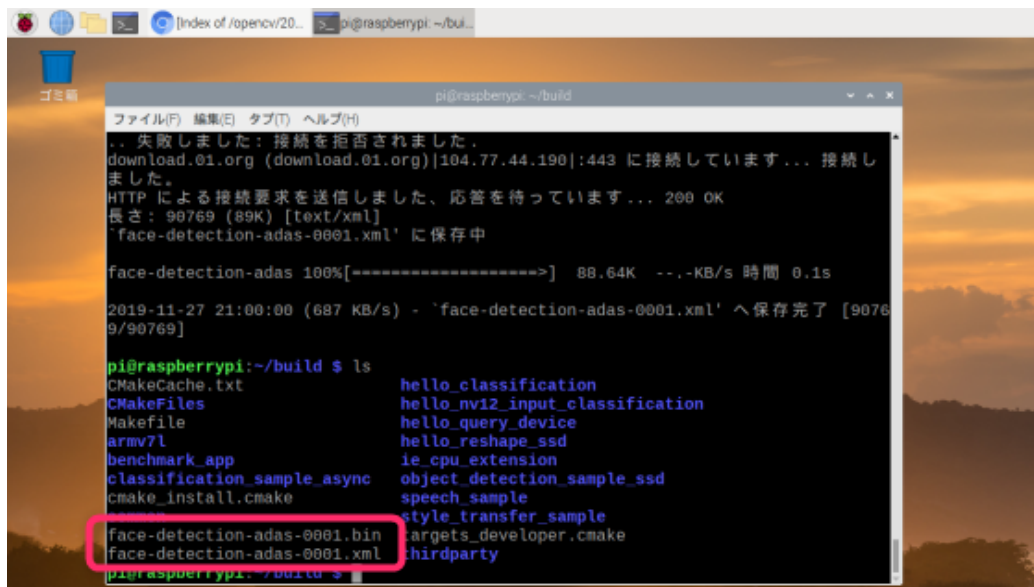
```
wget --no-check-certificate https://download.01.org/opencv/2019/open_m
```

次に学習済みモデルファイルをダウンロードします。こちらも 保存完了 などと出ればOKです

```
wget --no-check-certificate https://download.01.org/opencv/2019/open_m
```

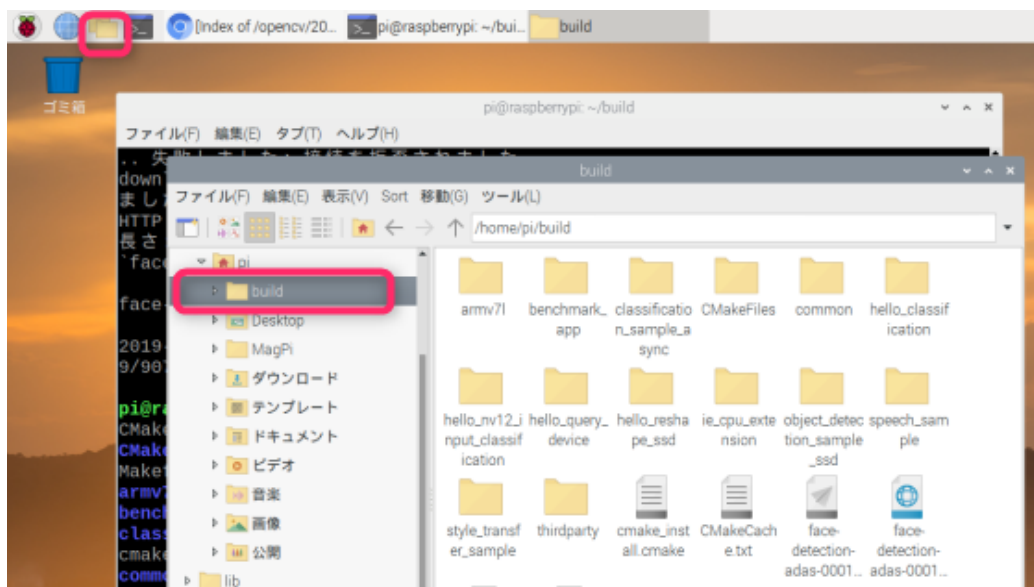
ここでlsコマンドを実行してみてください

```
ls
```

このように binファイルとxmlファイルがあることが確認できると思います

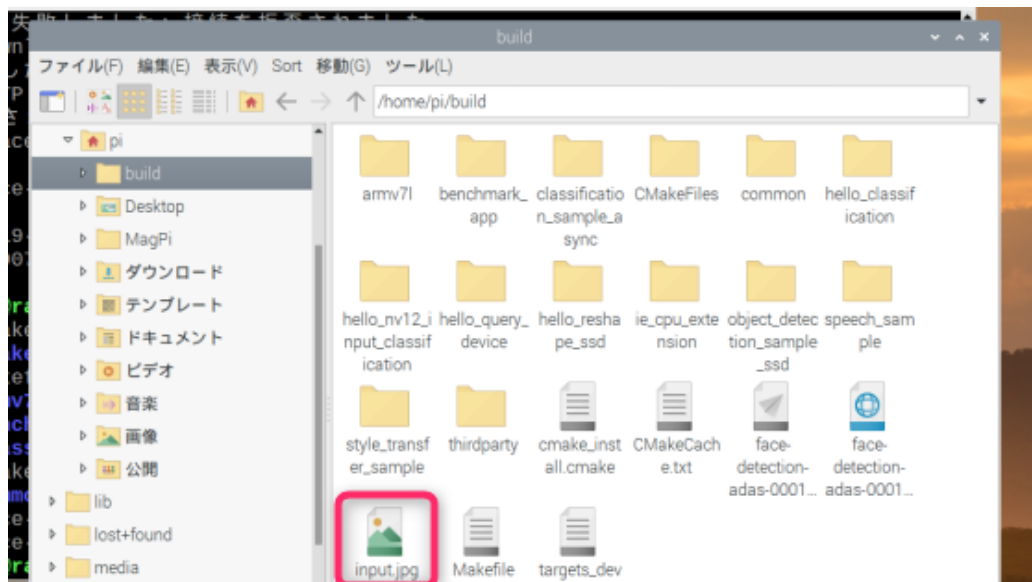
今までコマンドで見てきましたが、普通にマウスを使ってもファイルを確認することが可能です。



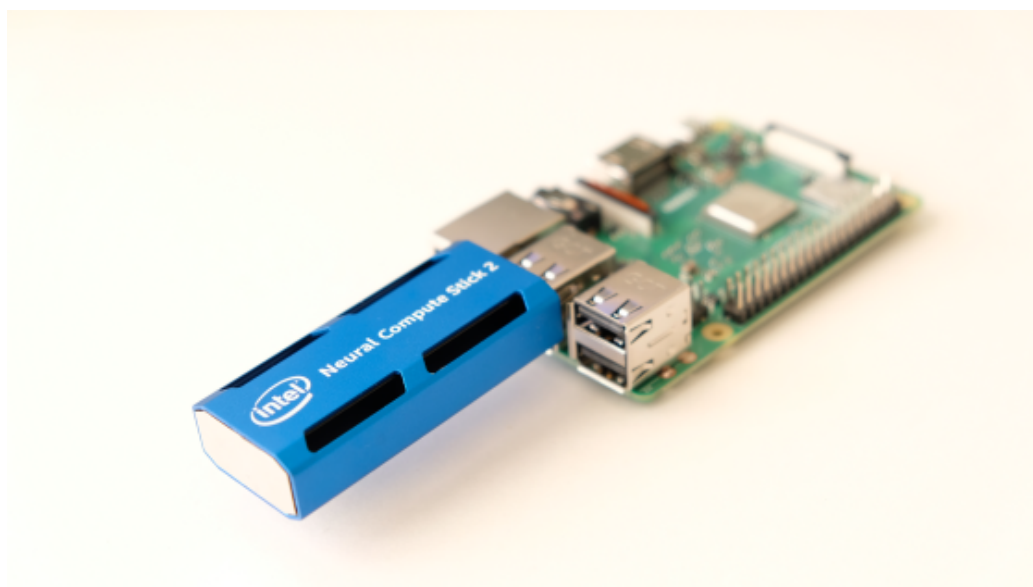
ラズパイ画面左上の「フォルダアイコン」をクリックし、 build フォルダを選択することで中身が見えます。

最後にディープラーニング推論する画像を用意して下さい。顔が写っている画像であれば何でも良いです。ブラウザ等を使ってインターネットから適当にダウンロードしてもOKです。画像を build フォルダに入れて下さい。この後のコマンドでファイル名を記述するので、ファイル名

を `input.jpg` に変更して下さい。右クリックすれば変更できます。



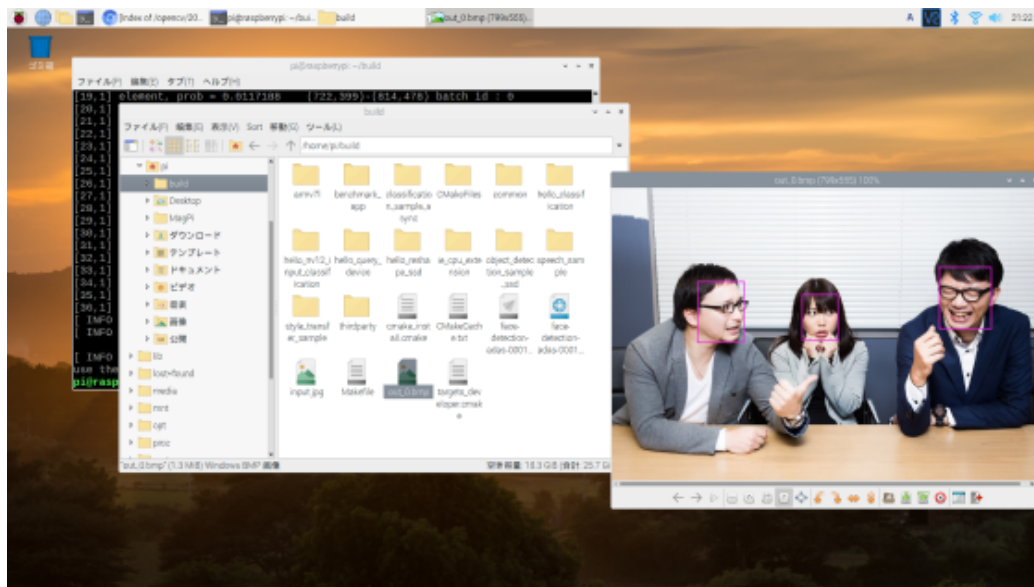
実行の前にNeural Compute Stickをラズパイに挿入します。



これで全て準備が揃いましたので以下のコマンドで実行します

```
./armv7l/Release/object_detection_sample_ssd -m face-detection-adas-000
```

最後の方に `[INFO] Execution successful` という表示が出れば成功です。 `build` フォルダに `out_0.bmp` というファイルが出来ていますので、ダブルクリックしてみてください



元々用意した画像の顔部分に紫色の矩形が描かれていると思います。別の入力画像を用意すれば、最後のコマンドだけですぐに実行可能です。

さて、ここまでがRaspberrPiへのOpenVINO™インストールと実行確認でした。後半のコマンドが難しいと感じた方やビルド時間が長いと感じた方もいらっしゃると思います。本シリーズではPythonを使ってアプリを作りますので、今のような長いコマンドやビルドなどは今後一切出てきませんので安心してください。モデルと重みと入力画像は同様に必要ですが、Pythonで行う場合はプログラムを書いて即座に実行できます！次回はPythonの基礎から学んでゆきます。

以上、「ゼロから始めるインストール」でした。



Newsletter

ご登録いただくと、新商品やイベント情報をお知らせします。

メールアドレスを入力してください

登録

個人情報について

VISION ABOUT WORKS BLOG  

JELLYWARE

〒160-0004 東京都新宿区四谷2-3-6 パルム四谷702号室

03-6273-0758

info@jellyware.jp

Copyright 2016-2017 JellyWare Inc.

個人情報について

CONTACT

お名前

メールアドレス

メールアドレス確認

お電話番号（任意）

ご件名

