
KEIm-CVSoC
開発キット Rev.B
OpenVINO™ ツールキット
導入ガイド

Ver.1.0



株式会社近藤電子工業

はじめに

この度は、KEIm 製品をお買い上げいただき誠にありがとうございます。

本製品をご使用になる前に、本マニュアル及び関連資料を十分ご確認ください、使用上の注意を守って正しくご使用ください。



取扱い上の注意

- 本書に記載されている内容は、将来予告なく変更されることがあります。本製品のご使用にあたっては、弊社窓口または弊社ホームページなどで最新の情報をご確認ください。
- 本製品には一般電子機器用部品が使用されています。極めて高い信頼性を要求する装置（航空、宇宙機器、原子力制御機器、生命維持のための医療機器等）には使用しないでください。
- 本製品は国内使用を前提として開発及び製造を行っています。本製品または本製品を組み込んだ製品を輸出される場合は、お客様の責任において「外国為替及び外国貿易法」及びその他輸出関連法令等を順守し、必要な手続きを行ってください。
- LAN、USB 以外のコネクタへのケーブルの抜き差しは、必ず電源を OFF にした状態で行ってください。
- 水、湿気、ほこり、油煙等の多い場所では使用しないでください。
- 本製品の関連資料の全部または一部を弊社に無断で使用または複製することを禁止します。
- 本書及び関連資料で取り上げる会社名及び製品名等は、各メーカーの商標または登録商標です。

お問い合わせ先

- 製品に関するお問い合わせは、下記のメールアドレスよりお願いいたします。

keim-support@kd-group.co.jp

目次

1. 概要.....	4
1.1. 関連文書.....	4
2. 使用機材	4
3. 本書の流れ.....	5
4. AI アクセラレータの組み込み	5
5. Linux の準備	6
5.1. SD イメージのダウンロード.....	6
5.2. SD イメージ書き込み手順	6
5.3. microSD カードの取り出し方と差し込み方.....	7
5.4. 接続構成.....	8
5.5. ターミナル接続.....	9
5.5.1. VCPドライバのインストール	9
5.5.2. 通信フォーマット.....	9
5.6. Linux 起動	10
6. AI デモ環境の準備.....	11
6.1. OpenVINO ツールキットのセットアップ	11
6.1.1. OpenVINO ツールキットのダウンロード及びインストール.....	11
6.1.2. セットアップスクリプトの編集.....	11
6.1.3. udev ルールの設定.....	12
6.2. Open Model Zoo のセットアップ.....	12
6.2.1. Open Model Zoo のインストール.....	12
7. 物体検出デモサンプルのビルド及び実行(C++)	13
7.1. サンプルプログラムのダウンロード	13
7.2. サンプルプログラムのビルド	13
7.3. モデル及びラベルファイルの準備	13
7.4. デモ実行	14
8. 顔認証デモサンプルのビルド及び実行(Python)	16
8.1. サンプルプログラムのダウンロード	16
8.2. face_gallery の準備	16
8.3. デモ実行	17
9. 更新履歴	18

1. 概要

本書では KEIm-CVSoC 開発キットにインテル® Movidius™ Myriad™ X VPU を搭載した IEI 製 Mustang-M2BM-MX2 を組み込み、OpenVINO ツールキットのデモンストレーションを実行する手順を紹介します。

1.1. 関連文書

項目	備考
KEIm-CVSoC SoM Rev.B ハードウェアマニュアル	KD-KEIM1061
KEIm-CVSoC 開発キット Rev.B スタートアップガイド	KD-KEIM1062
KEIm-CVSoC 開発キット Rev.B Mustang-M2BM-MX2 インストールマニュアル	KD-KEIM1064

2. 使用機材

プログラムの動作のため、本製品の他にご準備いただくものを表 2-1 に記載します。また作業に必要な PC ツールを表 2-2 に記載します。

表 2-1 使用機材一覧

項目	備考
KEIm-CVSoC 開発キット	本装置。AC アダプタと USB マイクロ Type-B ケーブルは同梱。
AI アクセラレータ	IEI 製 Mustang-M2BM-MX2
HDMI ミニケーブル	
USB マイクロ Type-AB ケーブル	
microSD カード	16GByte 以上、Class10 を推奨
USB キーボード	
USB マウス	
USB ハブ	
Ethernet	有線での Internet 接続環境が必要
ディスプレイ	HDMI 入力がついているもの
PC	microSD カードへのデータの書き込みや UART ターミナルに使用します。パフォーマンスは高い必要はありません。ただし、PC に SD カードスロットがない場合は別途 USB 等の SD カードリーダーが必要です。

表 2-2 PC ツール一覧

項目	備考
Win32 Disk Imager	SD イメージを SD カードに書き込むために使用します。
TeraTerm	ターミナルソフト
デモソフト SD イメージ	弊社のホームページよりダウンロードして使用します。

3. 本書の流れ

本書は次の流れで構成され、最終的に2種類のデモンストレーションの実行結果をご紹介します。

- ① AI アクセラレータの組み込み
- ② Linux の準備: microSD カードへの SD カードイメージの書き込み及び Linux の起動
- ③ AI デモ環境の準備: OpenVINO ツールキット及び Open Model Zoo のインストール及びセットアップ
- ④ 物体検出デモサンプルのビルド及び実行(C++)
- ⑤ 顔認証デモサンプルのビルド及び実行(Python)

4. AI アクセラレータの組み込み

本デモンストレーションでは推論エンジンとしてインテル® Movidius™ Myriad™ X VPU を使用します。

KEIm-CVSoC 開発キット Rev.B Mustang-M2BM-MX2 インストールマニュアル(KD-KEIM1064)に沿って Mustang-M2BM-MX2 を組み込んでください。

5. Linux の準備

5.1. SD イメージのダウンロード

KEIm-CVSoC カメラユニットに挿入する microSD カードに Linux のイメージを書き込みます。本書では書き込みツールとして、Win32 Disk Imager を使用します。2020 年 9 月現在、下記のサイトよりバージョン 1.0 がダウンロード可能ですので、ダウンロードしたのち、インストーラーの指示に従ってインストールしてください。

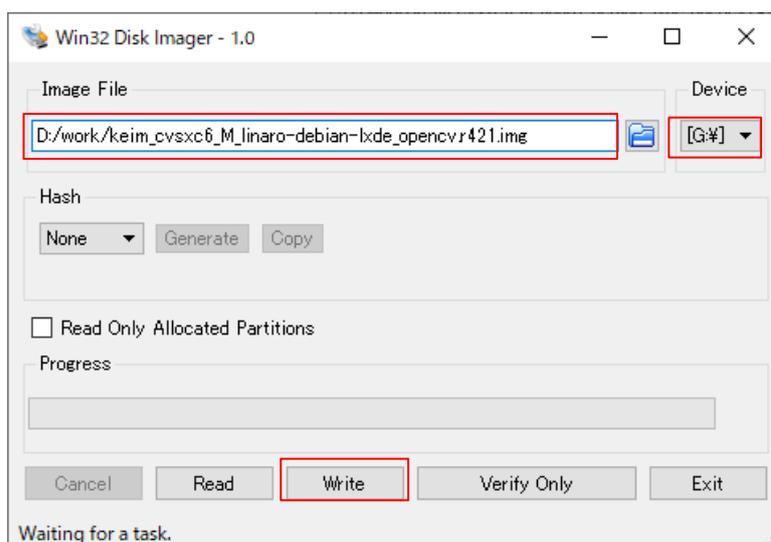
<https://sourceforge.net/projects/win32diskimager/>

また、SD イメージファイル keim-cvsoc-b_devkit_gsrd.7z は弊社のホームページの下記 URL からダウンロード可能ですので、最新版を入手してください。ファイルは 7z 形式で圧縮されていますので、7zip で解凍すると.img ファイルが生成されます。

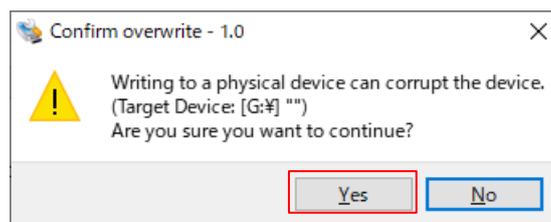
<https://kd-group.co.jp/download/>

5.2. SD イメージ書き込み手順

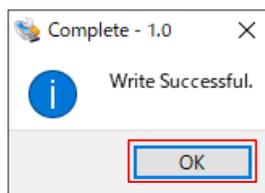
- ① microSD カードを PC に挿入します。PC に SD カードスロットがない場合は、USB カードリーダー等をご使用ください。
- ② Win32 Disk Imager を立ち上げます。Image File に用意した.img ファイルのパスを指定します。また、Device に SD カードのドライブを指定してください。



- ③ Write をクリックすると下記のダイアログ(上書きの確認)が表示されますが、Yes をクリックして書き込みを開始します。



- ④ 書き込みが完了すると Complete のダイアログが表示されますので、OK をクリックしてください。

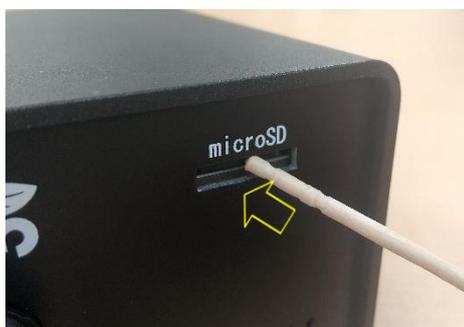


- ⑤ 書き込み完了後、Windows のタスクトレイから SD カードの取り外しを実行した後、SD カードスロットから microSD カードを取り出してください。

※Windows のアップデート(20H2)以降、SD カードのパーティション認識に対して変化があり、一度リファレンスデザインを書き込んだ SD カードに対して、上記の手順に沿ってイメージを上書きしようとする、失敗することがあります。その場合は、Windows の管理ツールや Diskpart ツールを使用して SD カードのパーティションを削除してから再度イメージを書き込んでください。

5.3. microSD カードの取り出し方と差し込み方

microSD カードをカメラユニット本体から取り出すときは、電源を OFF にし細い棒などを利用して microSD カードを軽く押し込み、少し飛び出して来たら引き抜いてください。また microSD カードを差し込むときは、「カチッ」と音が鳴るまで差し込んでください。このとき、上面側に microSD カードのロゴが見えるように挿入してください。



5.4. 接続構成

接続構成図を図 5-1 に示します。接続に際しては電源を OFF にした状態で行ってください。

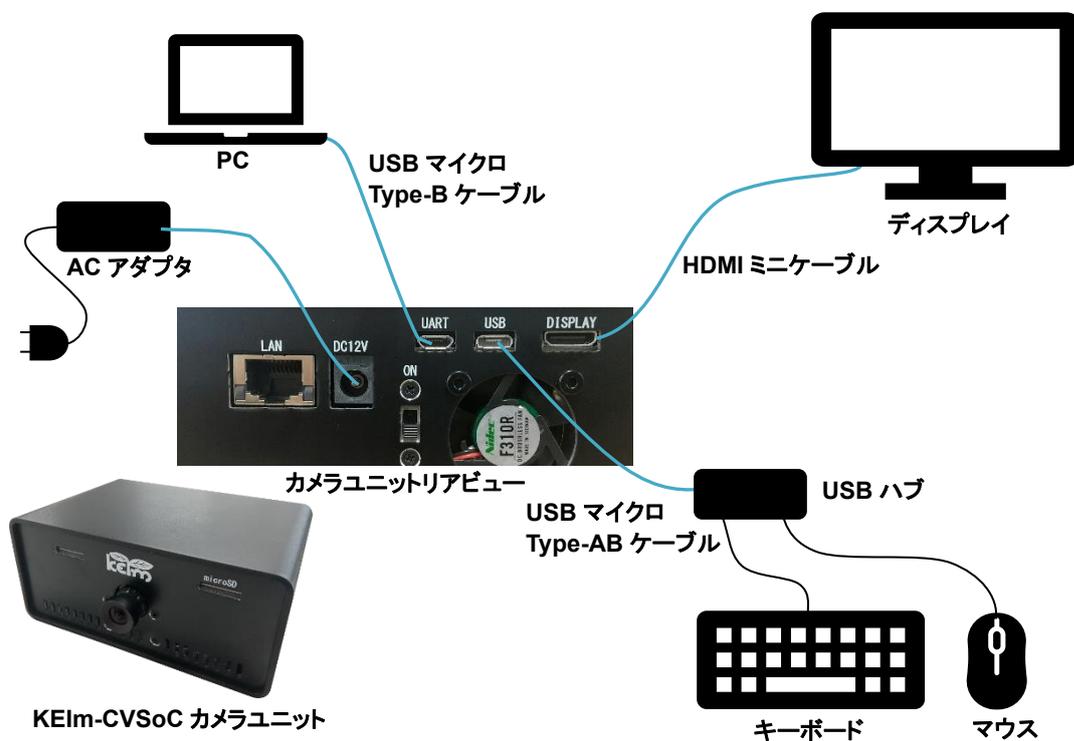


図 5-1 接続構成図

5.5. ターミナル接続

KEIm-CVSoC カメラユニットとのターミナル接続は USB シリアル CP2102N-A02-GQFN24 (Silicon Labs) を介して行われます。これを使用するためには Silicon Labs の Virtual COM Port ドライバ(以降 VCP ドライバと称す)のインストールが必要です。

5.5.1. VCP ドライバのインストール

既にインストールしたことがある PC であれば、KEIm-CVSoC カメラユニットと PC を USB マイクロ Type-B ケーブルで接続すると USB シリアルドライバのインストールが始まります。もし自動的にインストールされない場合は、Silicon Labs 社サイトの下記 URL より Windows 用の VCP ドライバをインストールしてください。

<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>

正常にインストールされると、図 5-2 のように Windows のデバイスマネージャー上に CP210x の COM ポートが表示されます。

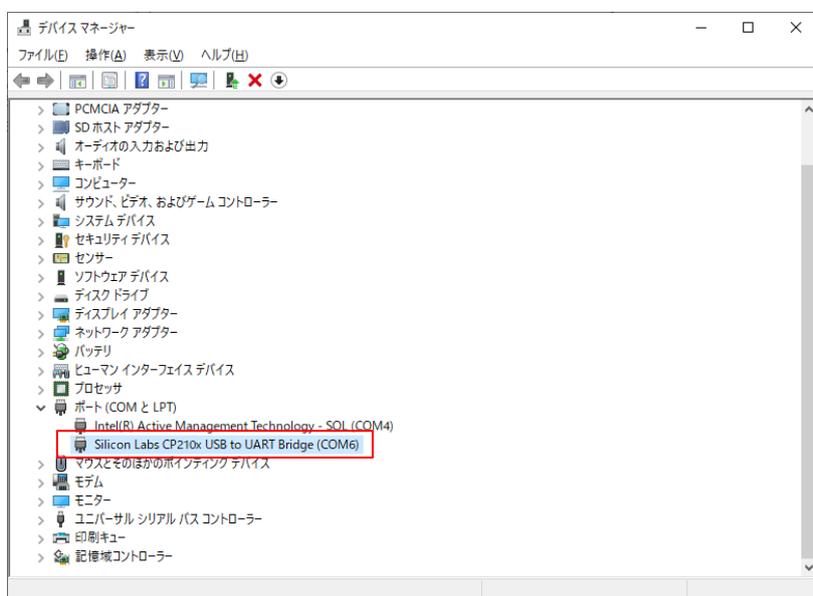


図 5-2 デバイスマネージャー

5.5.2. 通信フォーマット

リファレンスデザインの UART 通信フォーマットは表 5-1 のとおりです。TeraTerm などのターミナルソフトの通信設定を同様に設定してください。

表 5-1 通信フォーマット

項目	設定
ポート	デバイスマネージャーで確認した COM 番号を設定
スピード	115200bps
データ	8bit
パリティ	none
ストップビット	1bit
フロー制御	None

5.6. Linux 起動

電源投入すると、Linux が起動しディスプレイに下図のようなログイン画面が表示されます。username, password とともに「linaro」と入力し、Log In ボタンをクリックしてください。

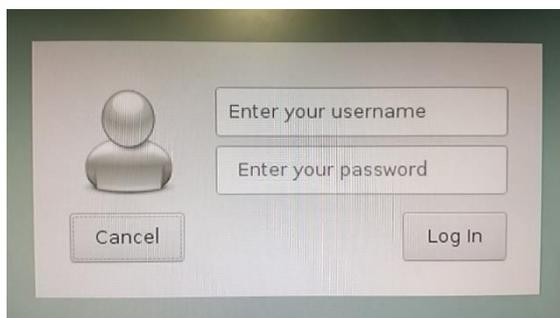


図 5-3 Log In 画面

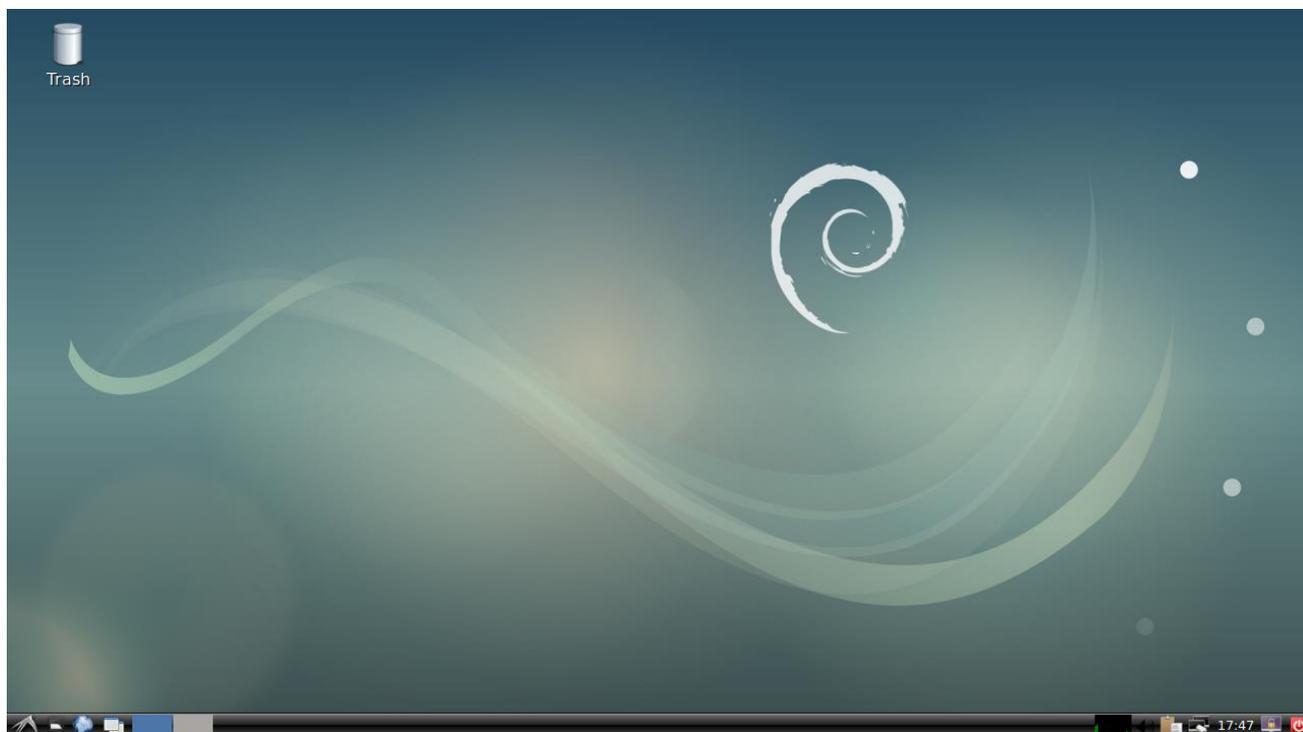


図 5-4 Linux デスクトップ

6. AI デモ環境の準備

本項では、各デモサンプルのビルド環境を構築するために OpenVINO ツールキット、及び Open Model Zoo のインストール及びセットアップの手順を記載します。本書では RaspberryPi 用の OpenVINO ツールキットをカスタマイズして使用します。

6.1. OpenVINO ツールキットのセットアップ

6.1.1. OpenVINO ツールキットのダウンロード及びインストール

インテルのオープンソーステクノロジーセンターである 01.org より RaspberryPi 用 OpenVINO ツールキットをダウンロードします。2020.4 以降のバージョンは古いモデルが使用できなくなっているため、本書では 2020.3.341 を使用します。

```
$ cd ~/Downloads/  
$ wget https://download.01.org/opencv/2020/openvinotoolkit/2020.3/_openvino_toolkit_runtime_raspbian_p_2020.3.341.tgz  
$ sudo mkdir -p /opt/intel/openvino  
$ sudo tar -xf _openvino_toolkit_runtime_raspbian_p_2020.3.341.tgz --strip 1 -C /opt/intel/openvino/
```

6.1.2. セットアップスクリプトの編集

セットアップスクリプト setupvars.sh を今回の環境に合わせて編集します。

```
$ cd /opt/intel/openvino/  
$ sudo sed -i "s/Raspbian/Debian/" bin/setupvars.sh
```

環境変数を設定するためにセットアップスクリプトを実行します。

```
$ cd  
$ source /opt/intel/openvino/bin/setupvars.sh
```

環境変数の設定は、ターミナルを立ち上げるたびに実行する必要がありますが、下記のように.bashrc に登録すればスクリプトを実行する手間が省けます。

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

6.1.3. udev ルールの設定

4 項で組み込んだ Mustang-M2BM-MX2 のデバイス管理のための udev ルールをインストールします。

```
$ sudo usermod -a -G users "$(whoami)"  
$ sh /opt/intel/opencvino/install_dependencies/install_NCS_udev_rules.sh
```

インストールした udev ルールは root でしか動かないようになっていますので、下記の通りパーミッションの設定部分を編集します。

```
$ sudo sed -i "s/660/666/" /etc/udev/rules.d/97-myriad-usbboot.rules
```

一旦再起動します。

```
$ sudo reboot
```

6.2. Open Model Zoo のセットアップ

Open Model Zoo には OpenVINO ツールキットの開発に有用なデモアプリケーションや複数のトレーニング済みモデルなどが入っています。本書では、Open Model Zoo のビルド環境を使用して後述のサンプルプログラムをビルドします。

6.2.1. Open Model Zoo のインストール

GitHub からリポジトリをクローンし、2020.3 のタグをチェックアウトします。

```
$ git clone https://github.com/opencv/open_model_zoo.git  
$ cd open_model_zoo  
$ git tag  
$ git checkout 2020.3
```

7. 物体検出デモサンプルのビルド及び実行(C++)

本項では Open Model Zoo に用意されている複数のサンプルのうち、Object Detection SSD C++ Demo を流した物体検出のデモンストレーションを実行する手順を記載します。

このソフトウェアには、[Apache 2.0 ライセンス](#)で配布されている製作物が含まれています。

7.1. サンプルプログラムのダウンロード

サンプルプログラムを弊社のホームページのダウンロードサイトよりダウンロードし、Open Model Zoo の demos ディレクトリに展開します。

```
$ cd ~/Downloads/  
$ wget https://kd-group.co.jp/download-data/keim-cvsoc/ai_app/keim-cvsoc-b_devkit_openvino_object-detection-demo.tgz  
$ tar -xzvf keim-cvsoc-b_devkit_openvino_object-detection-demo.tgz -C ~/open_model_zoo/demos/
```

7.2. サンプルプログラムのビルド

```
$ cd ~/open_model_zoo/demos/  
$ mkdir build && cd build  
$ cmake -DCMAKE_BUILD_TYPE=Release ..  
$ make -j2 object_detection_demo_ssd_async_cam_scaler_multi
```

ビルドが成功すると、./armv7l/Release に実行ファイルが生成されます。

7.3. モデル及びラベルファイルの準備

7.1 で展開したサンプルプログラムのディレクトリにモデルファイル(Caffe の MobileNet-SSD)とラベルファイルが入っていますので、実行ディレクトリにコピーします。

```
$ cd ~/open_model_zoo/demos/build/armv7l/Release  
$ cp ../../object_detection_demo_ssd_async_cam_scaler_multi/mobilenet-ssd.* .
```

7.4. デモ実行

下記の通り引数を設定し実行すると、図 7-1 のようにカメラから入力した画像に映った物体にバウンディングボックスが出現し、ボックス上部に検出した物体名が表示されます。同時にレンダリング、処理時間(処理速度)、対象物検知時間も表示されます。この結果では 17.66fps の処理速度を記録しました。

```
$ cd ~/open_model_zoo/demos/build/armv7l/Release  
$ ./object_detection_demo_ssd_async_cam_scaler_multi -i FPD -d MYRIAD -m mobilenet-ssd.xml
```

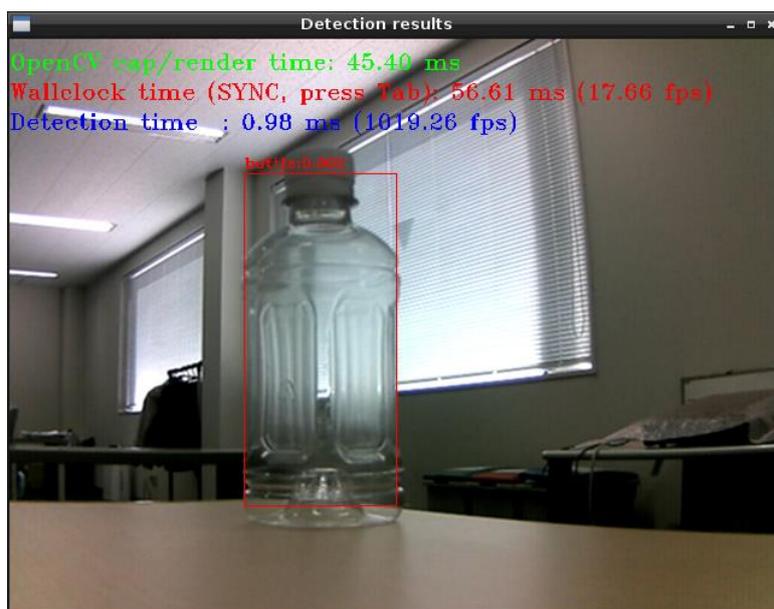


図 7-1 物体検出実行結果

先のコマンドのように何も指定しない場合、入力画像サイズは 640×480 に設定されますが、下記のように指定することも可能です。結果を図 7-2 の左に示します。画像サイズを絞ることにより、処理速度が 34.61fps に向上しているのが分かります。

```
$ cd ~/open_model_zoo/demos/build/armv7l/Release
$ ./object_detection_demo_ssd_async_cam_scaler_multi -i FPD -d MYRIAD -m mobilenet-ssd.xml -i
w 300 -ih 300
```

実行中に TAB キーを押すことによりタスク処理の同期／非同期を切り替えることが可能です。非同期処理では処理速度は 48.41fps に達しています。(図 7-2 の右)

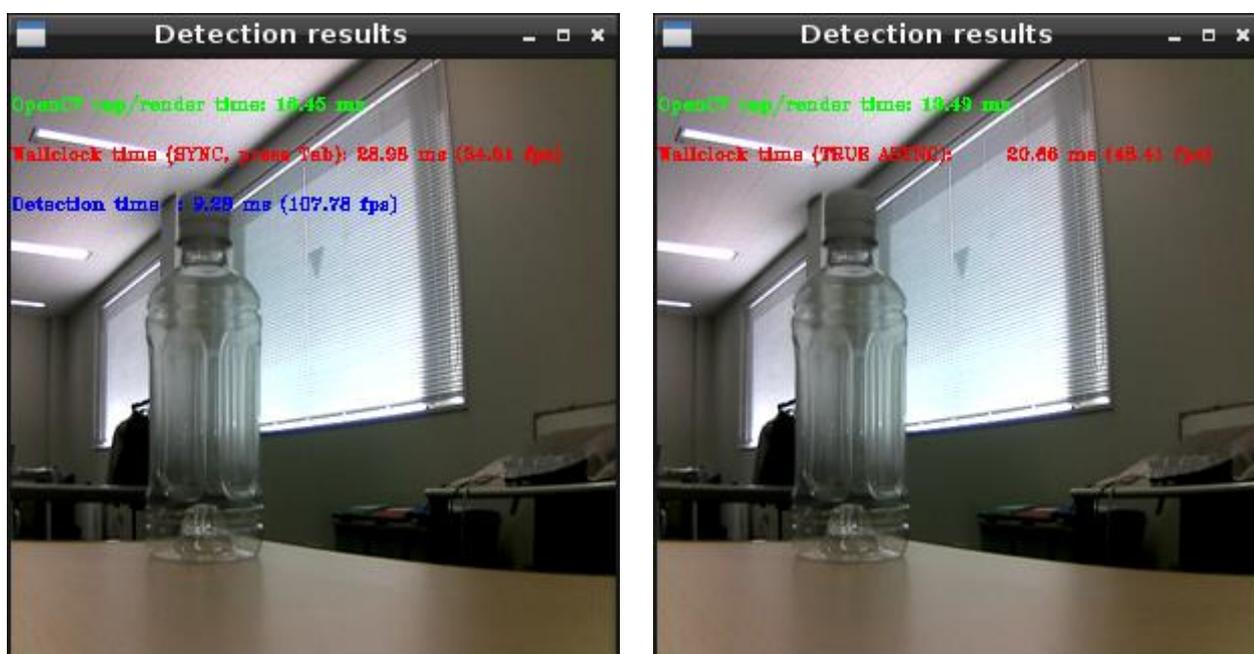


図 7-2 物体検出実行結果 300×300

8. 顔認証デモサンプルのビルド及び実行(Python)

本項では Open Model Zoo に用意されている複数のサンプルのうち、Interactive Face Recognition Demo を流用した顔認証のデモンストレーションを実行する手順を記載します。

このソフトウェアには、[Apache 2.0 ライセンス](#)で配布されている製作物が含まれています。

8.1. サンプルプログラムのダウンロード

サンプルプログラムを弊社のホームページのダウンロードサイトよりダウンロードし、Open Model Zoo の demos ディレクトリに展開します。

```
$ cd ~/Downloads/  
$ wget https://kd-group.co.jp/download-data/keim-cvsoc/ai_app/keim-cvsoc-b_devkit_openvino_face-recognition-demo.tgz  
$ tar -xzvf keim-cvsoc-b_devkit_openvino_face-recognition-demo.tgz -C ~/open_model_zoo/demos/python_demos/
```

8.2. face_gallery の準備

認証に使うデータ(顔写真)は/home/face_galleryから読み出されますので、ディレクトリを作成し、データを保存します。データの形式は PNG(.png)または JPEG(.jpg)で、ファイル名に記載した文字列が検出時の結果となって画面に表示されます。

```
$ mkdir ~/face_gallery
```

8.3. デモ実行

下記の通り引数を設定し実行すると、図 8-1 のようにカメラから入力した画像に映った人物にバウンディングボックスが出現し、ボックス上部に検出した名称(person_1_name)が表示されます。

```
$ cd ~/open_model_zoo/demos/python_demos/face_recognition_demo_cam
$ python3 ./face_recognition_demo.py ¥
  -m_fd face-detection-retail-0004.xml ¥
  -m_lm landmarks-regression-retail-0009.xml ¥
  -m_reid face-reidentification-retail-0095.xml ¥
  --verbose ¥
  -fd_iw 640 ¥
  -fd_ih 480 ¥
  -d_fd MYRIAD ¥
  -d_lm MYRIAD ¥
  -d_reid MYRIAD ¥
  -fg "/home/linaro/face_gallery"
```

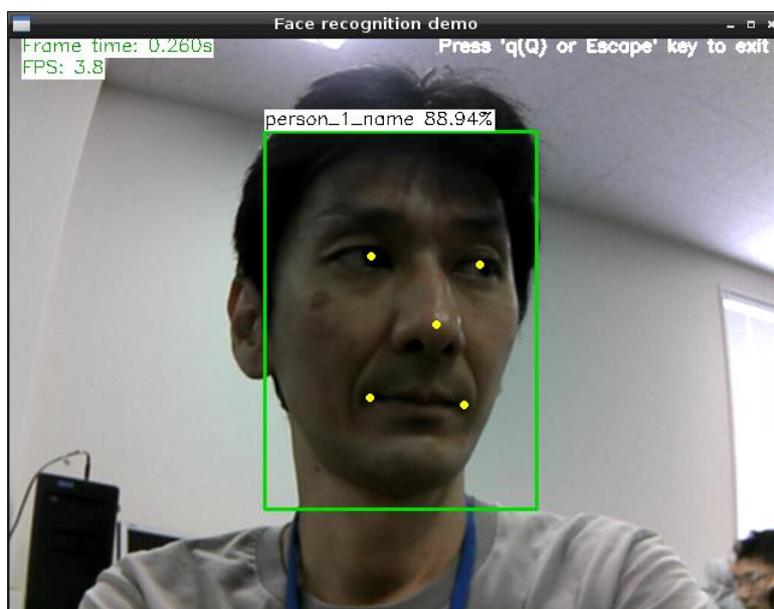


図 8-1 顔認証実行結果

9. 更新履歴

Ver.	更新日付	内容
1.0	2021/8/31	新規作成