

# ボードコンピュータ（NVIDIA Jetson）上で動く 学習不要なAIプログラム（人・物体検出） 利用ガイド

---

# 目次

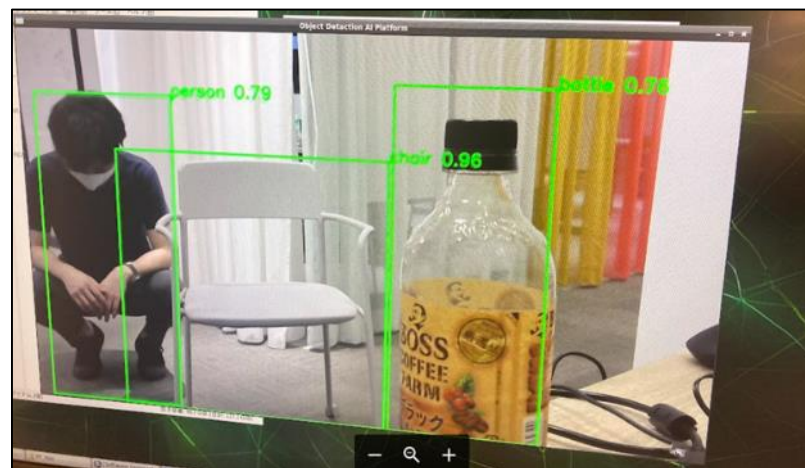
1. このAIプログラムでできること . . . . . p.03
2. 対象者 . . . . . p.04
3. 利用に必要な機器 . . . . . p.05
4. 推奨する利用環境 . . . . . p.06
5. 利用にあたっての注意事項 . . . . . p.07
6. AIプログラムのダウンロードと実行準備 . . . . . p.08
7. 人・物体検出の実行 . . . . . p.20
8. 参考：ソースコードを利用する場合の実行方法 . . . . . p.28

# 1 このAIプログラムでできること

- WEBカメラと連動してリアルタイムで指定したラベル（人・動物・モノ）を検出できます。
- 工場内等において、立入禁止エリアへの侵入検知による安全管理などに御利用いただけます。
- 通常、AIを利用するには学習が必要ですが、オープンソースの学習済みモデルを使用しているため、学習なしで利用できます。



利用シーン例



検出結果イメージ

## ！ 注意点

- ✓ 検出に伴い警告音の発出やアラームの通知を行う場合は、別途、警報器など他の装置と組み合わせ、連携のためのプログラムを組む必要があります。

## 2 対象者

- AIの学習が必要ないため、**AIやプログラミングの知識がない方でも御利用いただけます。**
- 精度の向上や、検出する物の種類を増やす等の**カスタマイズを行う場合は、新たにAI学習を実施する必要**があるため、AI学習を行う機能を開発するための**AIやプログラミングに関する知識が必要**になります。

### 3 利用に必要な機器

利用にあたっては以下の機器が必要になりますので、御準備ください。

- Jetson Nano 2GB Developer Kit
- microSDカード (128GB)
- 電源アダプター
- ディスプレイ (HDMI接続可能なもの)
- HDMIケーブル
- キーボード
- マウス
- パソコン (イメージファイルを書き込む際に使用)
- Webカメラ

#### ! Jetson Nano 2GBに関する注意点

- ✓ microSDカードなしで起動した際にディスプレイにNVIDIAのロゴが出ないマシンでは、microSDカードありで起動できなくなる場合があります。  
参考：Jetson Nano 2GB failing to bootup or show anything  
<https://forums.developer.nvidia.com/t/jetson-nano-2gb-failing-to-bootup-or-show-anything/167053>
- ✓ AIプログラム掲載ページでは、ご利用者の状況や使用したいOSに合わせて利用できるよう、イメージファイルに加えて、ソースコードと必要なライブラリ情報も掲載しています。

#### ! パソコンに関する注意点

- ✓ パソコンは、解凍後のイメージファイルのサイズ (約62GB) 以上の空き容量を確保してください。
- ✓ イメージファイルを解凍するソフトとmicroSDカードへの書き込みを行うソフトはWindowsのみ対応となります。
- ✓ 対応OSのバージョンは、「6.AIプログラムのダウンロードと実行準備」で紹介する各ソフトの提供元ページを確認ください。

## 4 推奨する利用環境

\* 実行に際して、必ずしも以下の環境が必要ではありません。あくまで推奨環境になります。

### ■ Jetsonの推奨スペック（開発に使用したJetson nano 2GBのスペック）

<b>GPU</b>	128-core Maxwell
<b>CPU</b>	Quad-core ARM A57 @ 1.43 GHz
<b>Memory</b>	2 GB 64-bit LPDDR4 25.6 GB/s
<b>Storage</b>	microSD ※64GB以上が必須になります。128GBを推奨します。

### ■ 参考：ソースコードを利用する場合

- ▶ 提供するAIプログラムは、イメージファイルをJetsonで実行することを想定していますが、ソースコードをパソコンで実行することも可能です（実行方法はp.28参照）。
- ▶ 実行するためには、python実行環境を構築する必要があります。
- ▶ パソコンの推奨スペックは以下の通りです（これ以上のスペックでも利用可能です）。

<b>OS</b>	Windows , Mac , Linux
<b>GPU</b>	必須ではありませんが、GPUを使用することで実行／学習速度向上が見込めるため、推奨します。
<b>CPU</b>	Core(TM) i5-7300U CPU @ 2.60GHz 2.70 GHz
<b>Memory</b>	実装RAM 8.00 GB

## 5 利用にあたっての注意事項



以下の注意事項をよくお読みいただき、同意の上、御利用ください。

- 1 AIプログラムの実行にあたっては利用者御自身で環境設定ファイル及び利用ガイドを参考に準備いただく必要があります。
- 2 本プラットフォームで配布しているAIプログラムは、自社/自団体内のみで利用できるものとし、第3者への配布や商用利用は禁止します。
- 3 本プラットフォームで配布しているAIプログラムは、利用者御自身の責任において御利用ください。利用により生じる損害について、運営事務局では責任を負いません。
- 4 本プラットフォームで配布しているAIプログラムは自由に改変することは可能ですが、利用者御自身の責任において改変ください。

## 6 AIプログラムのダウンロードと実行準備

- 以下の手順でAIプログラムをダウンロードし、microSDカードに書き込んで実行できるようにします。全体で3時間程度かかりますので、ご注意ください。

(1) AIプログラムのダウンロードと解凍(詳細はp.9-13)

(2) microSDカードフォーマット用ソフトのインストール(詳細はp.14)

(3) イメージファイル書き込み用ソフトのインストール(詳細はp.15)

(4) microSDカードのフォーマット(詳細はp.16)

(5) microSDカードへの書き込み(詳細はp.17-18)

(6) Jetsonと周辺機器の接続(詳細はp.19)

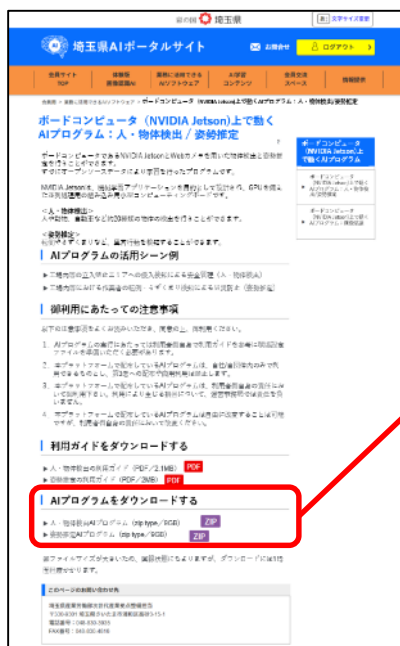


# 6 AIプログラムのダウンロードと実行準備

## (1) AIプログラムのダウンロードと解凍

■ AIプログラムが入った圧縮ファイルをダウンロードし、解凍します。

- ① パソコンで、AI・IoTプラットフォームの掲載ページから、AIプログラムのファイル（**人・物体検出AIプログラム**）をダウンロードします。ファイルサイズが約9GBと大きいため、回線状況によりりますが、1時間程度かかります。



拡大

### AIプログラムをダウンロードする

- ▶ 人・物体検出AIプログラム (zip type/9GB) ZIP
- ▶ 姿勢推定AIプログラム (zip type/9GB) ZIP

- ② ダウンロードした圧縮ファイルを解凍します。解凍すると、約62GBとなるため、パソコンのディスクの空き容量を確保してください。

# 6 AIプログラムのダウンロードと実行準備

## (1) AIプログラムのダウンロードと解凍

- ③ 解凍用ソフト Explzh for Windowsを以下のページからダウンロードします。  
<https://www.ponsoftware.com/archiver/download.htm>  
※Windowsのみ対応となります。  
※32ビット版と64ビット版がありますので、御使用のパソコンにあった方を選んでください。

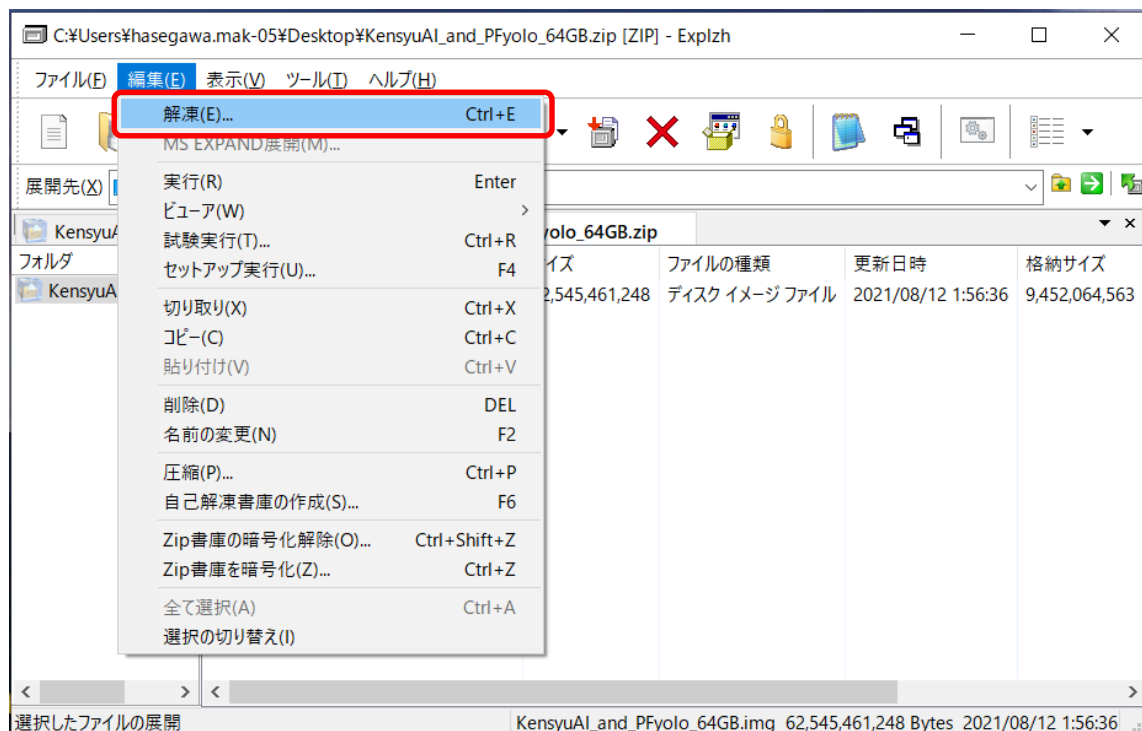


- ④ ダウンロードしたExplzhのexeファイルをダブルクリックして開き、インストーラーにしたがってインストールします。

# 6 AIプログラムのダウンロードと実行準備

## (1) AIプログラムのダウンロードと解凍

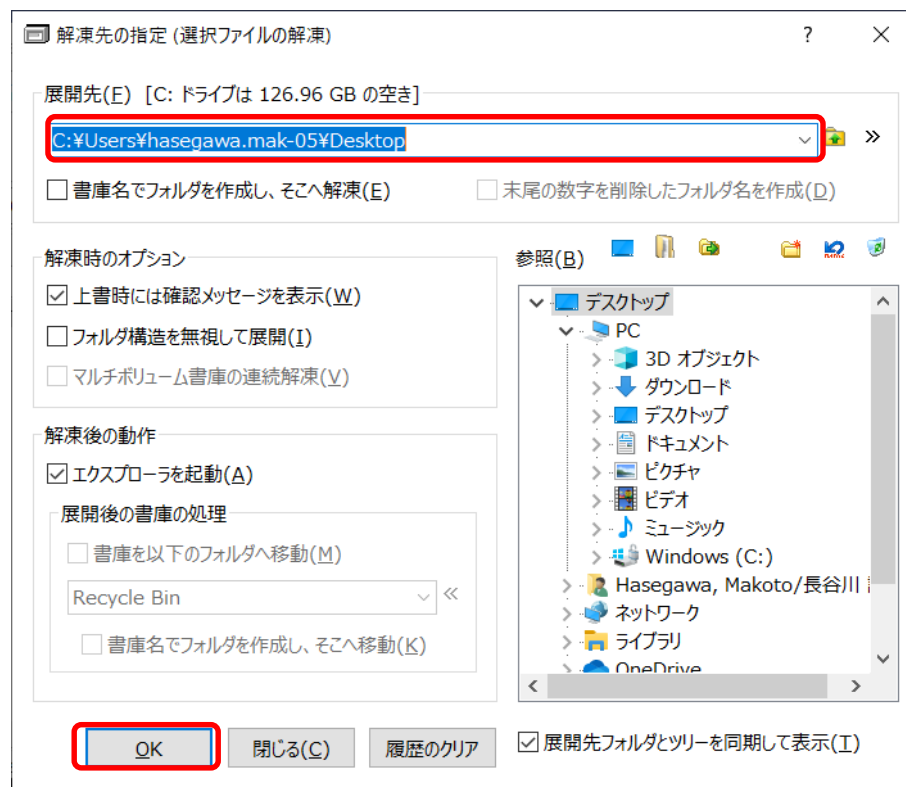
- ⑤ 手順①でダウンロードしたAIプログラムのZIPファイルをダブルクリックすると、ウィンドウが立ち上がりますので、メニューの「編集 (E)」から「解凍 (E)」をクリックします。



## 6 AIプログラムのダウンロードと実行準備

### (1) AIプログラムのダウンロードと解凍

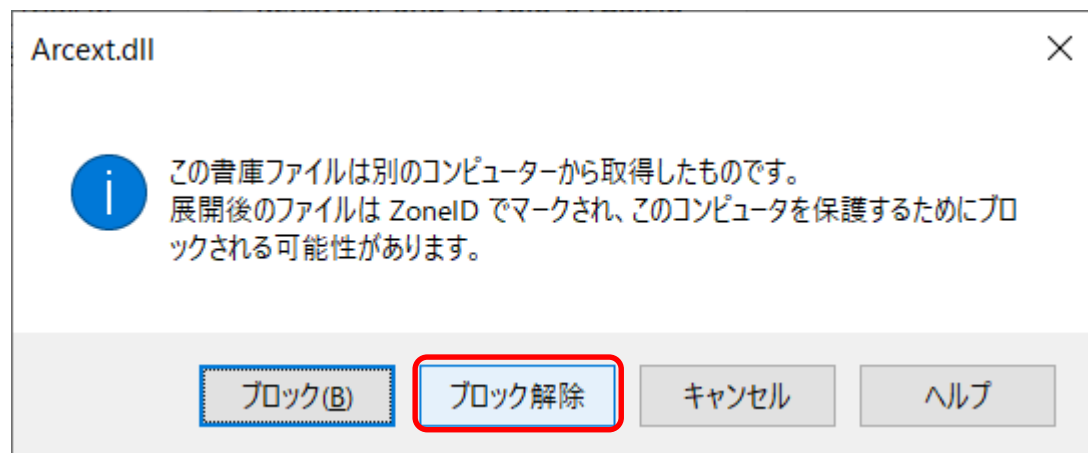
- ⑥ 解凍先を指定するウィンドウが立ち上がりますので、「展開先」でファイルを解凍したい場所を選び、「OK」を押します。



## 6 AIプログラムのダウンロードと実行準備

### (1) AIプログラムのダウンロードと解凍

- ⑦ 確認のためのポップアップが出てくるので、「ブロック解除」を押すと、解凍が始まります。PCの使用状況によりますが、解凍に1時間程度かかります。



- ⑧ 指定した展開先にimgファイルができれば、完了です。

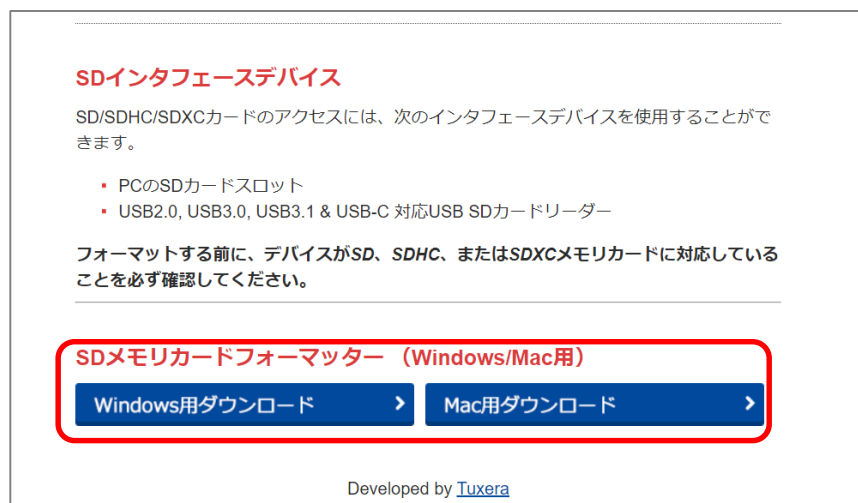
# 6 AIプログラムのダウンロードと実行準備

## (2) microSDカードフォーマット用ソフトのインストール

- microSDカードにAIプログラムのイメージファイルを書き込める状態にすることをフォーマットと言い、そのためのソフトをインストールします。

- ① SD Formatterのダウンロードページから、ページの下方にある「SDメモリカードフォーマッター」をダウンロードします。

<https://www.sdcard.org/ja/downloads-2/formatter-2/>



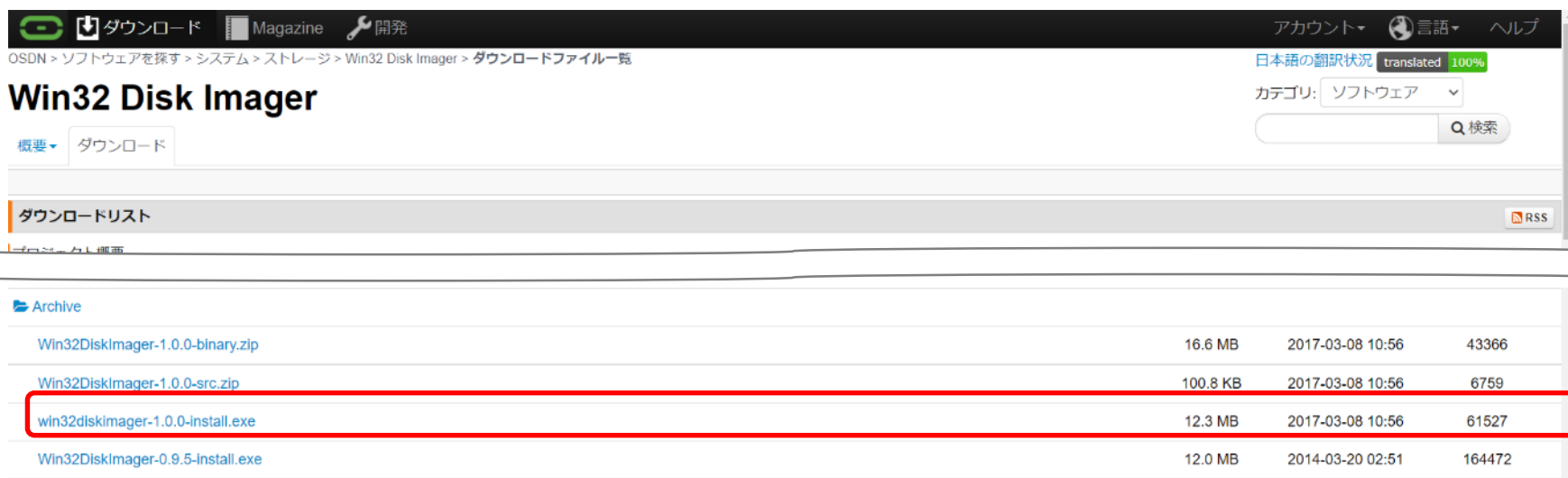
- ② ダウンロードしたZIPファイルを解凍し、中のsetup.exeをダブルクリックして開いて、シインストーラーに従ってインストールします。

## 6 AIプログラムのダウンロードと実行準備

### (3) イメージファイル書き込み用ソフトのインストール

- AIプログラムのイメージファイルをmicroSDカードに書き込み、AIプログラムを実行できる状態にするためのソフトをインストールします。

- ① Win32 Disk Imagerのダウンロードページから、下方の「Archive」にある「win32diskimager-1.0.0-install.exe」をダウンロードします。  
[https://ja.osdn.net/projects/sfnet\\_win32diskimager/releases/](https://ja.osdn.net/projects/sfnet_win32diskimager/releases/)  
※Windowsのみ対応となります。



The screenshot shows the OSN website for Win32 Disk Imager. The page title is "Win32 Disk Imager" and it includes a "ダウンロード" (Download) button. Below the title, there is a "ダウンロードリスト" (Download List) section. The list contains several entries, with the following data extracted from the table:

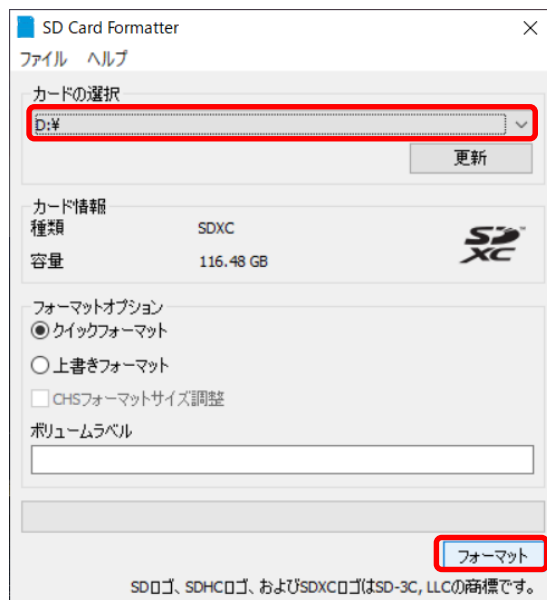
File Name	Size	Date	Downloads
Win32DiskImager-1.0.0-binary.zip	16.6 MB	2017-03-08 10:56	43366
Win32DiskImager-1.0.0-src.zip	100.8 KB	2017-03-08 10:56	6759
<b>win32diskimager-1.0.0-install.exe</b>	12.3 MB	2017-03-08 10:56	61527
Win32DiskImager-0.9.5-Install.exe	12.0 MB	2014-03-20 02:51	164472

- ② exeファイルをダブルクリックして開き、インストーラーにしたがってインストールします。

## 6 AIプログラムのダウンロードと実行準備

### (4) microSDカードのフォーマット

- microSDカードにAIプログラムのイメージファイルを書き込める状態にします。
- ① microSDカードをパソコンに差し込みます。
  - ② (2) ①でインストールして作成されたSD Card Formatterのショートカットをダブルクリックして起動します。
  - ③ microSDカードのドライブ名を確認して、「カードの選択」で同じドライブを選び、「フォーマット」を押します。いくつか確認のポップアップが出ますので、全て「はい」を選択してください。

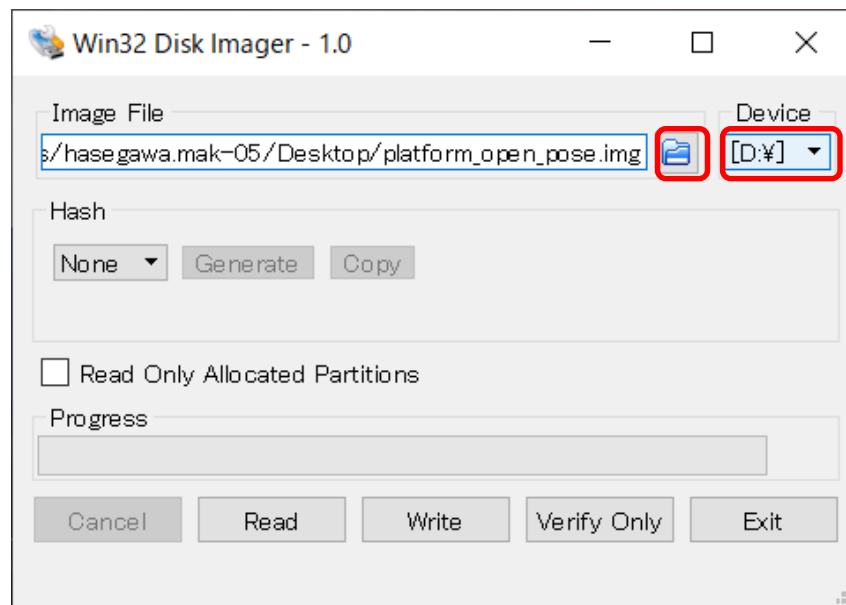




## 6 AIプログラムのダウンロードと実行準備

### (5) microSDカードへの書き込み

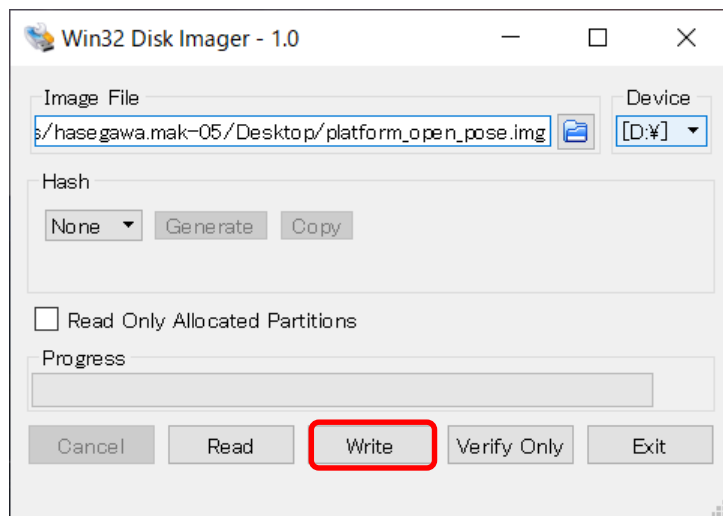
- AIプログラムのイメージファイルをmicroSDカードに書き込み、AIプログラムを実行できる状態にします。
- ① Win32 Disk Imagerのショートカットをダブルクリックして起動します。
  - ② 「Image File」のフォルダマークのボタンを押して、(1)⑧で解凍したimgファイルを選択し、その右側の「Device」でmicroSDカードのドライブを選びます。



## 6 AIプログラムのダウンロードと実行準備

### (5) microSDカードへの書き込み

- ③ imgファイルと書き込み先ドライブに間違いが無ければ「Write」を押します。

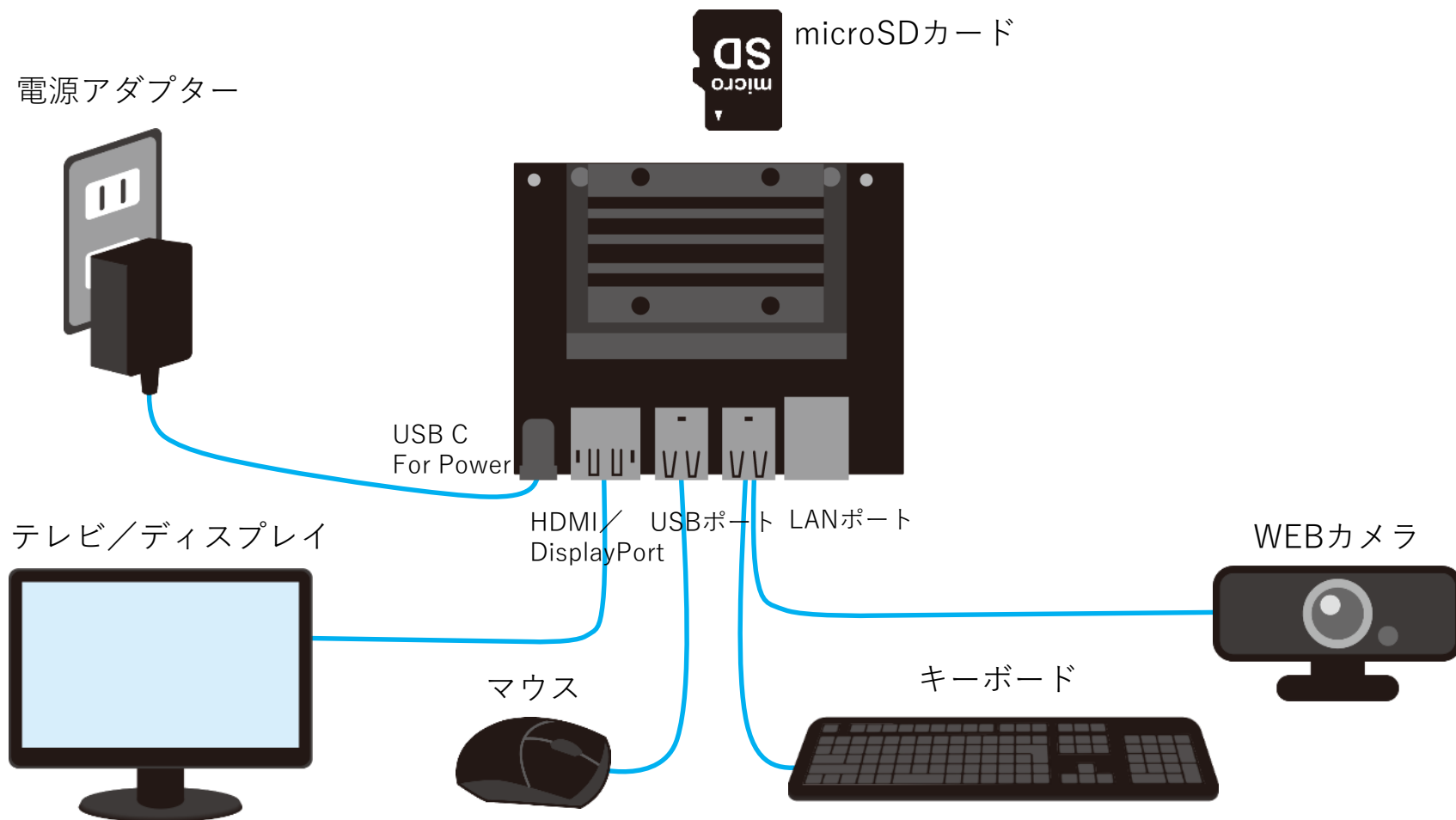


- ④ 書き込んで良いかを確認するポップアップが出てくるので「Yes」を押すと、書き込みが始まります。PCの使用状況によりますが、書き込みに40分程度かかります。
- ⑤ imgファイルの書き込みが終わったらポップアップが出てきますので、「OK」を押して、Win32 Disk Imagerを終了します。
- ⑥ imgファイルを書き込んだmicroSDカードをパソコンから取り出し、完了です。

# 6 AIプログラムのダウンロードと実行準備

## (6) Jetsonと周辺機器の接続

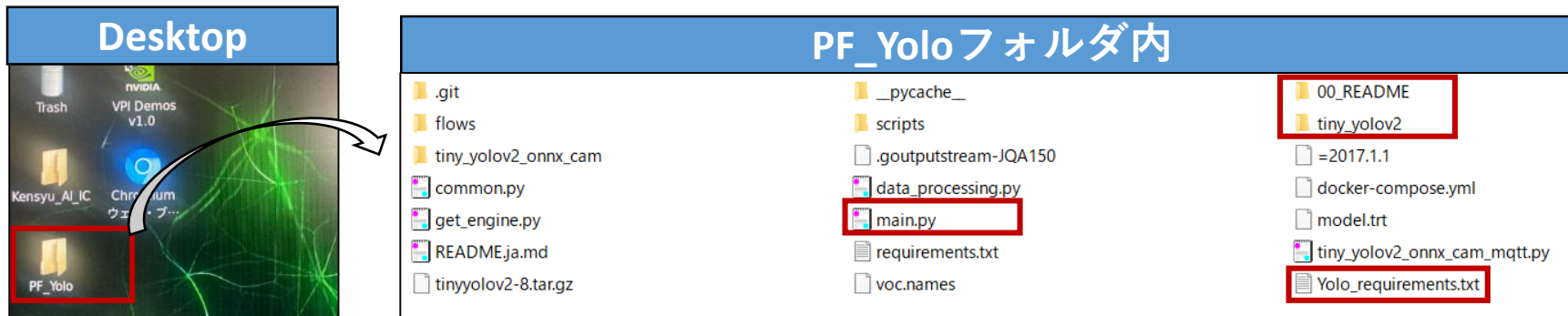
- Jetsonと周辺機器を接続します。



# 7 人・物体検出の実行

## ファイル構成

Desktop上に「PF\_Yolo」ファイルを格納しております。



※) 人・モノ検出では学習済みモデルつまりOSS(オープンソース)を使用しているため、参考にした元ファイル等は残しております。

参考文献/ソース: 「Jetson Nano 超入門」

そのためファイル数は多くなっておりますが、ユーザーが参照する主要ファイルは赤枠で囲った4ファイルになります。

主要ファイル	ファイル内容
00_README	人・モノ検出プログラムの実行方法/注意点/検出ラベルを記載したメモファイル。こちらを参考に実行してください
main.py	人・モノ検出を実行するメインソースコード
Yolo_requirement.txt	人・モノ検出を実行する際に必要なpythonライブラリを記載したメモファイル
tiny_yolo_v2	今回使用する学習済みモデル (tiny_yolo_v2)

# 7 人・物体検出の実行

## 実行方法サマリー

### (1) Jetson起動 (詳細はp.22)

microSDカードを挿入&電源を付け、パスワードを入力するとJetsonが起動 ※) 起動まで10秒程かかります

### (2) コマンド入力 (詳細はp.23)

①デスクトップのターミナルを開く

②mainファイル (実行ファイル) が置いてあるディレクトリに移動

コマンド: `cd /home/jetsonai/Desktop/PF_Yolo`

③GPU(CUDA)指定

コマンド: `export CUDA_VISIBLE_DEVICES="0"`

④mainファイルの実行 ※) Jetsonの負荷状況によりますが、実行されるまで30-60秒程かかります

コマンド: `python3 main.py`

※) 上記実行方法につきましては、「00\_README」ファイル内に記載しております

### (3) 人・物体検出の実行 (詳細はp.24)

「main.py」ファイルを実行し、カメラ起動～物体検出までを実行

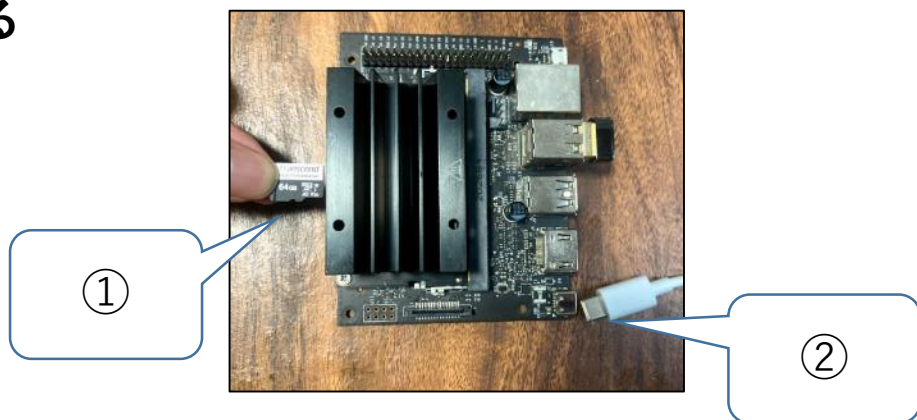
### (4) 推論結果の表示 (詳細はp.25)

カメラで撮影された映像から指定したラベルを検出し、「バウンディングボックス (検出した物体の囲い)」、左上に「予測ラベルと予測確率」を書いた結果をデスクトップにリアルタイムで表示

# 7 人・物体検出の実行

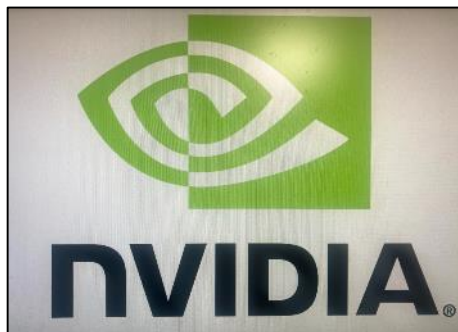
## (1) Jetsonの起動

- ①imgファイルを書き込んだmicroSDカードをJetsonに差し込む
- ②電源を入れる

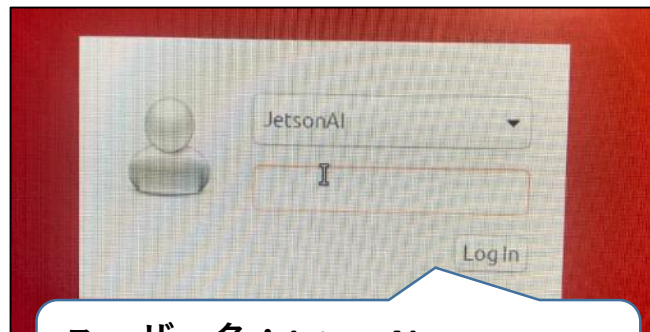


- ③電源を入れるとJetsonが起動し、パスワードを入力するとデスクトップが表示される

Jetson起動



パスワード入力



デスクトップ画面表示

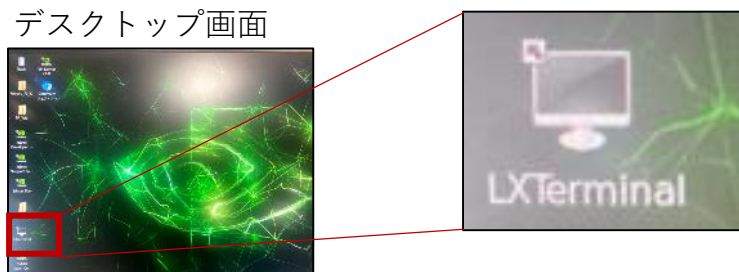


# 7 人・物体検出の実行

## (2) コマンド入力

- ① デスクトップのターミナルを開く  
(デスクトップ画面で「LXTerminal」をダブルクリックし、ターミナルを起動)

デスクトップ画面



—————↓ターミナル上で入力—————

- ② mainファイル (実行ファイル) が置いてあるディレクトリに移動

コマンド : `cd /home/jetsonai/Desktop/PF_Yolo`

- ③ GPU(CUDA)指定

コマンド : `export CUDA_VISIBLE_DEVICES="0"`

- ④ mainファイルの実行 ※) Jetsonの負荷状況によりますが、画面が起動されるまで30-60秒程かかります

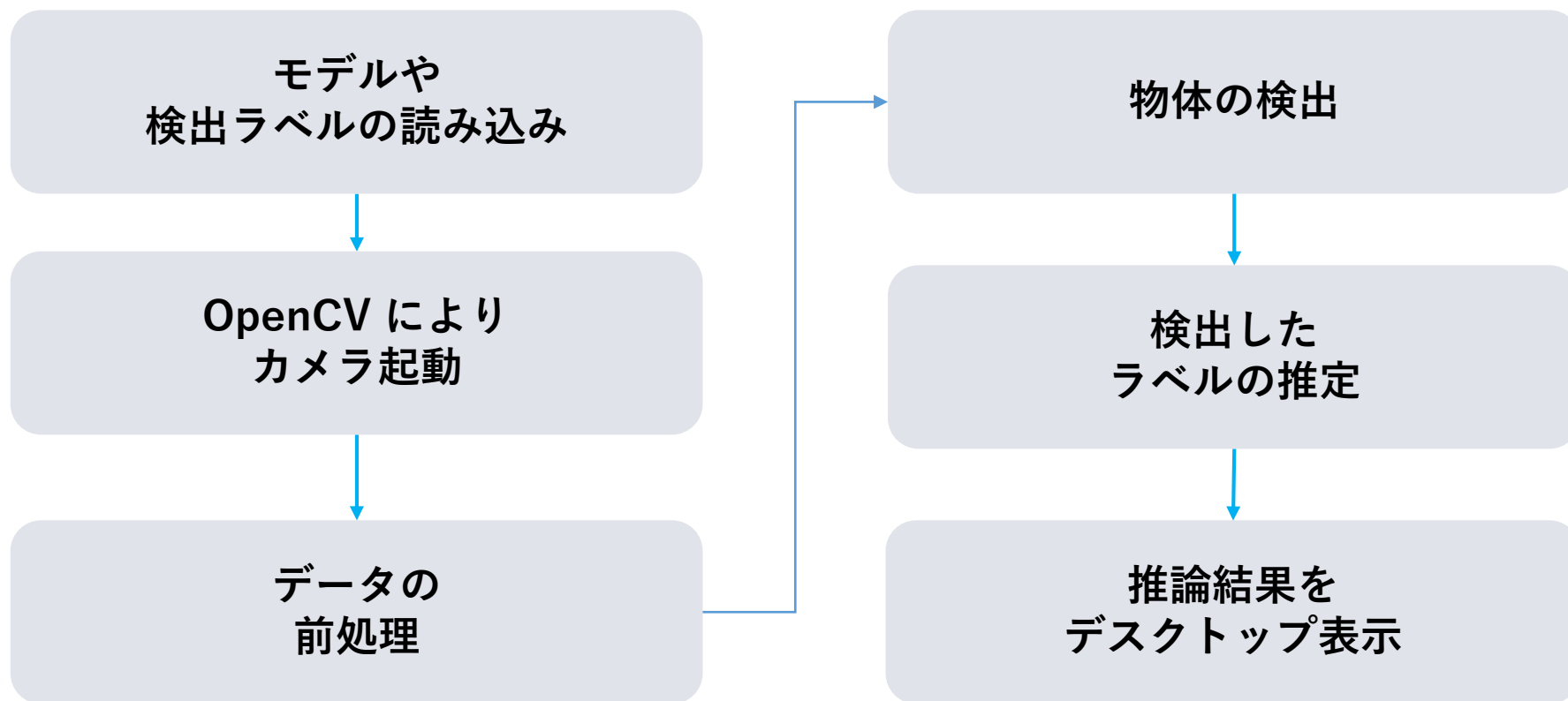
コマンド : `python3 main.py`

```
jetsonai@jetsonai-desktop: ~/Desktop/PF_Yolo
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
jetsonai@jetsonai-desktop:~$ cd /home/jetsonai/Desktop/PF_Yolo
jetsonai@jetsonai-desktop:~/Desktop/PF_Yolo$ export CUDA_VISIBLE_DEVICES="0"
jetsonai@jetsonai-desktop:~/Desktop/PF_Yolo$ python3 main.py
```

# 7 人・物体検出の実行

## (3) 人・物体検出の実行

「main.py」ファイルを実行すると、以下のフローで人・物体検出が実行されます。

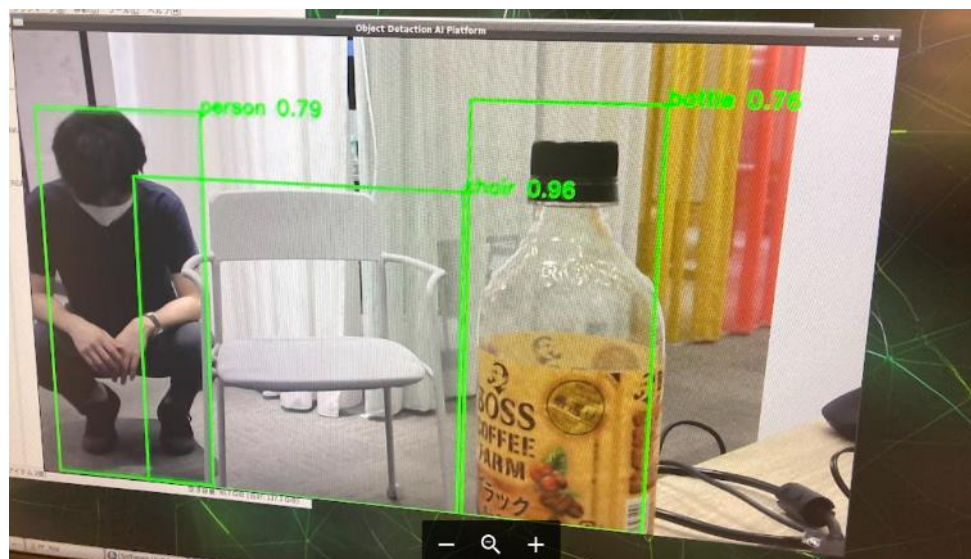




# 7 人・物体検出の実行

## (4) 推論結果の表示

カメラで撮影された映像から指定したラベルを検出し、「**バウンディングボックス（検出した物体の囲い）**」、左上に「**予測ラベルと予測確率**」を書いた結果をデスクトップにリアルタイムで表示します。



### ! 注意点

- ✓ 約20種類のラベルを予測可能（詳細は次ページ参照）
- ✓ 約1秒程のタイムラグがある際もありますが、ほとんどリアルタイムに近い形で予測出来ます。  
一負荷軽減のために、冷却ファンの使用等を推奨します

## 7 人・物体検出の実行

参考：今回のアルゴリズムで検出出来る20種類のラベル

乗り物	aeroplane	飛行機	動物	bird	鳥
	bicycle	自転車		cat	猫
	boat	ボート		cow	牛
	bus	バス		dog	犬
	car	車		horse	馬
	motorbike	モーターバイク		sheep	羊
	train	電車	人	person	人物
モノ	bottle	ペットボトル			
	chair	椅子			
	diningtable	テーブル			
	pottedplant	植木鉢			
	sofa	ソファ			
	tvmonitor	TVモニター			

## 7 人・物体検出の実行

### 参考：今回使用したアルゴリズムtiny\_yolo\_v2について

今回のAIプログラムには、既にAI学習が終わっているオープンソース（OSS）の学習済み物体検出モデルであるtiny\_yolo\_v2を使用しています。

#### tiny\_yolo\_v2アルゴリズムとは：

tiny\_yolo\_v2の基になっているYOLOv2はYOLOと呼ばれる物体検出アルゴリズムの改良モデルで、より多種類を分類可能なモデルです。

今回使用したtiny\_yolo\_v2アルゴリズムは、上記のYOLOv2の軽量版のモデルであり、「**認識精度は落ちるが高速に検出可能（リアルタイム性に長けている）**」であることが特徴です。

※参考URL <https://deepsquare.jp/2020/09/yolo/>

#### tiny\_yolo\_v2アルゴリズム ソースコード：

今回使用したtiny\_yolo\_v2アルゴリズムのような学習済みモデルは一般的に広く公開されており、ソースコードも公開されています。

※今回のソースコードはJetsonに関する書籍で著名な『Jetson Nano 超入門 改訂第2版』（ソーテック社）を参考にしています。

<http://www.sotechsha.co.jp/pc/html/1283.htm>



# 8 参考：ソースコードを利用する場合の実行方法

ソースコードを利用する場合は、パソコンでの実行を推奨いたします。

※提供するAIプログラムはJetsonでの利用を前提にしています。

ソースコードはパソコンで利用出来ますが、実行するためのpython/実行環境は各自で構築する必要があります。

## <実行方法>

### ①microSDにIMGファイルを書き込み後、Jetsonを起動

※IMGファイル書き込み方法は本ガイド6章を、Jetson起動方法は7章（1）を参考にしてください。

### ②起動したJetsonから、デスクトップにあるソースコードを含むファイル「PF\_Yolo」をPCにコピー

### ③python環境を構築

### ④今回ソースコードを使用するために必要なライブラリをインストール

※必要なライブラリは「PF\_Yolo」ファイル内の環境ファイル「Yolo\_requirements.txt」にまとめております。

※各ライブラリのversion/種類は各自の環境によって異なります。その際は、各自で修正&対応を行ってください。

また、「Yolo\_requirements.txt」はJetson用のライブラリになっておりますのでご注意ください。

特に、tensorflowのversionはPCで使用できるversion（tensorflow 2.4.0）をインストールしてください。

#### Python/実行環境

使用言語	Python 3.6.9
フレームワーク	tensorflow2.4.0+nv21.5 , Keras 2.4.0
主要OSS(algorithm)	tiny_yolov2 (物体検出アルゴリズム)
ライブラリ群	「PF_Yolo」ファイル内の環境ファイル「Yolo_requirements.txt」を参考にしてください

### ⑤ファイル「PF\_Yolo」内の「main.py」をpythonで実行してください

実行コマンド：python3 main.py

