

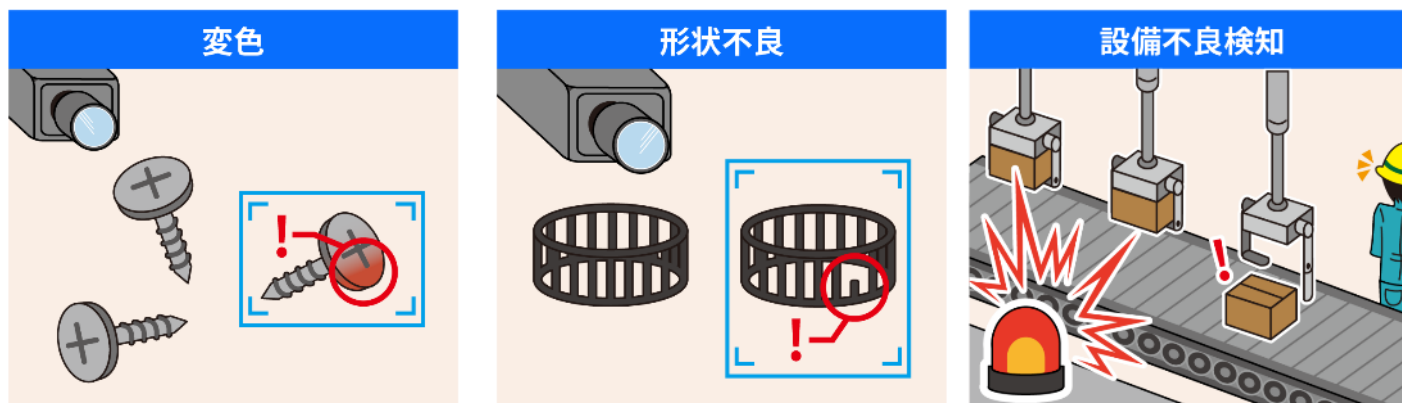
ボードコンピュータ（NVIDIA Jetson）上で動く 学習可能なAIプログラム（画像認識） 利用ガイド

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.08
6. AIプログラムのダウンロードと実行準備	p.09
7. AIプログラムの機能とファイル構成	p.21
8. Jetsonの起動方法	p.24
9. 画像認識における3機能のパラメータファイル	p.25
10. 画像データ収集	p.26
11. 画像認識（学習）	p.31
12. 画像認識（推論）	p.38
13. 参考：ソースコードを利用する場合の実行方法	p.43

1 このAIプログラムでできること

- **認識させたい画像で学習させることで、任意の画像を認識**することができます。
- 例えば、自社の製品等の画像を学習させることで、**製品の良品/不良品の判別**や**設備不良検知**などに活用できます。
- WEBカメラと連動して、画像データの収集、学習、推論という**画像認識の一連の流れをJetson上で行えます**。



利用シーン例

！ 注意点

✓ 画像認識を行うためには、認識させたい画像データを収集し、AIに学習させる必要があります。

2 対象者

- ソースコードを変更しなくても学習回数や学習率などのパラメータを変更できるようにしているため、**プログラミングの知識がない方でもAIの学習についての知識があれば、御利用いただけます。**
- **学習のためのデータセットの準備や、学習回数や学習率などのパラメータの調整が必要**になりますので、それらの知識がない方は、公益財団法人埼玉県産業振興公社が実施している研修の受講や書籍等での自己学習をお勧めします。

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。

(1) 利用環境

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

GPU	128-core Maxwell
CPU	Quad-core ARM A57 @ 1.43 GHz
Memory	2 GB 64-bit LPDDR4 25.6 GB/s
Storage	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.43参照）。
- ▶ 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）。

OS	Windows , Mac , Linux
GPU	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。
CPU	Core(TM) i5-7300U CPU @ 2.60GHz 2.70 GHz
Memory	実装RAM 8.00 GB

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。

(2) 学習データ

- 学習データの条件は以下の通りです。

データ	画像データのみを対象とします。
ファイル形式	ファイル形式は、JPG、PNGを推奨します。
サイズ	学習の際、指定したサイズに自動リサイズするためサイズ規定はありません。 ※画像サイズが大きくなると負荷が高くなり、処理速度が遅くなるためご注意ください。

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.10-14)

(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.15)

(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.16)

(4) microSDカードのフォーマット(詳細はp.17)

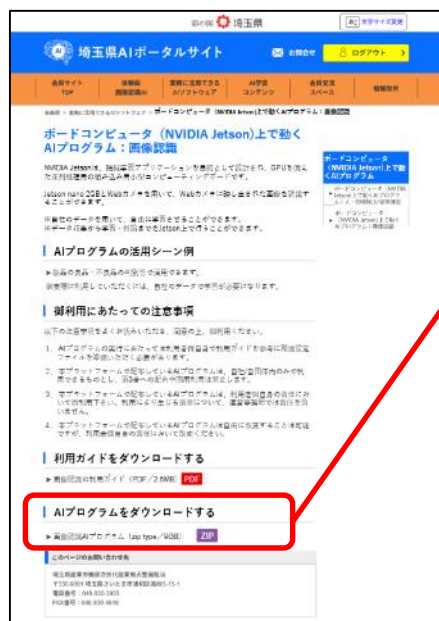
(5) microSDカードへの書き込み(詳細はp.18-19)

(6) Jetsonと周辺機器の接続(詳細はp.20)

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（**画像認識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

▶ 画像認識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。

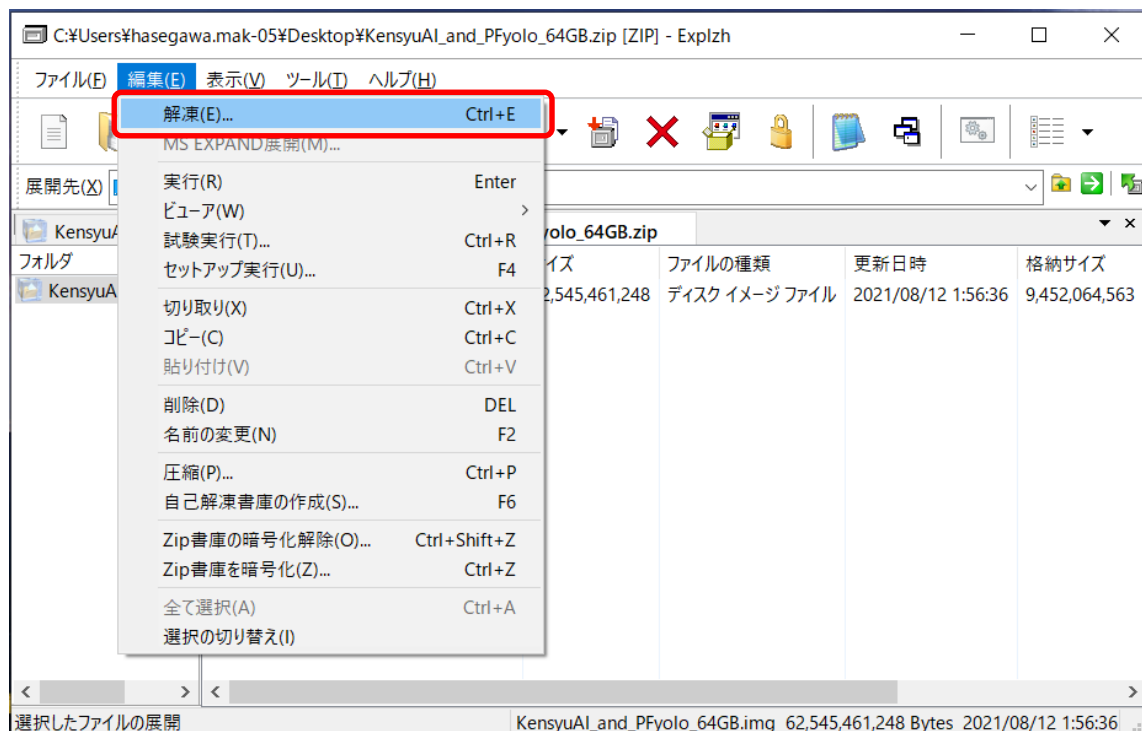


- ④ ダウンロードしたExplzhのexeファイルをダブルクリックして開き、インストーラーにしたがってインストールします。

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

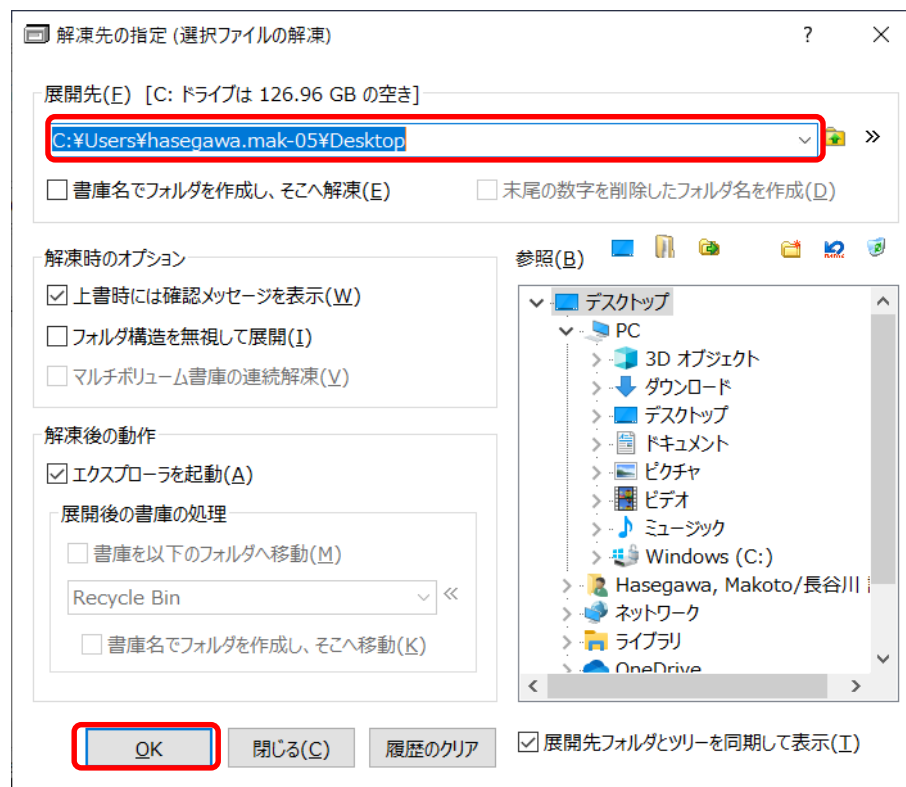
- ⑤ 手順①でダウンロードしたAIプログラムのZIPファイルをダブルクリックすると、ウィンドウが立ち上がりますので、メニューの「編集 (E)」から「解凍 (E)」をクリックします。



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

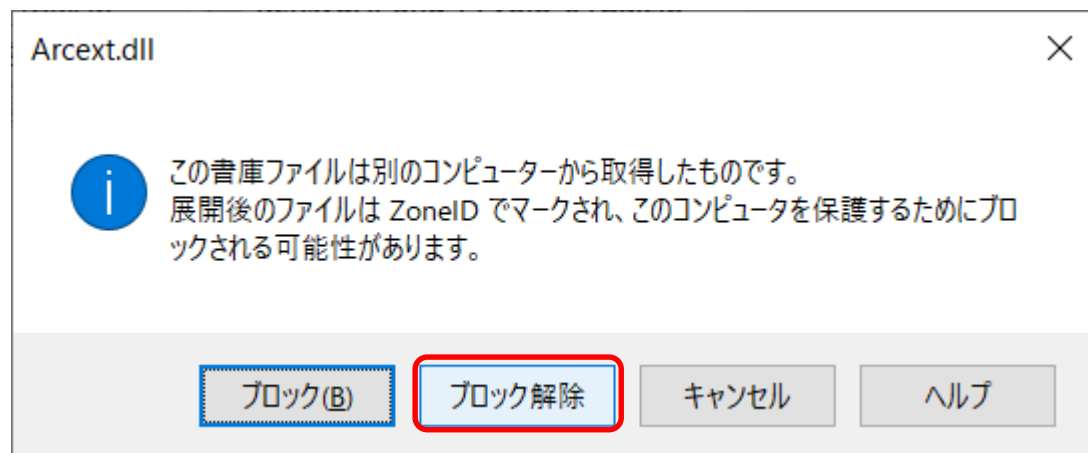
- ⑥ 解凍先を指定するウィンドウが立ち上がりますので、「展開先」でファイルを解凍したい場所を選び、「OK」を押します。



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/>



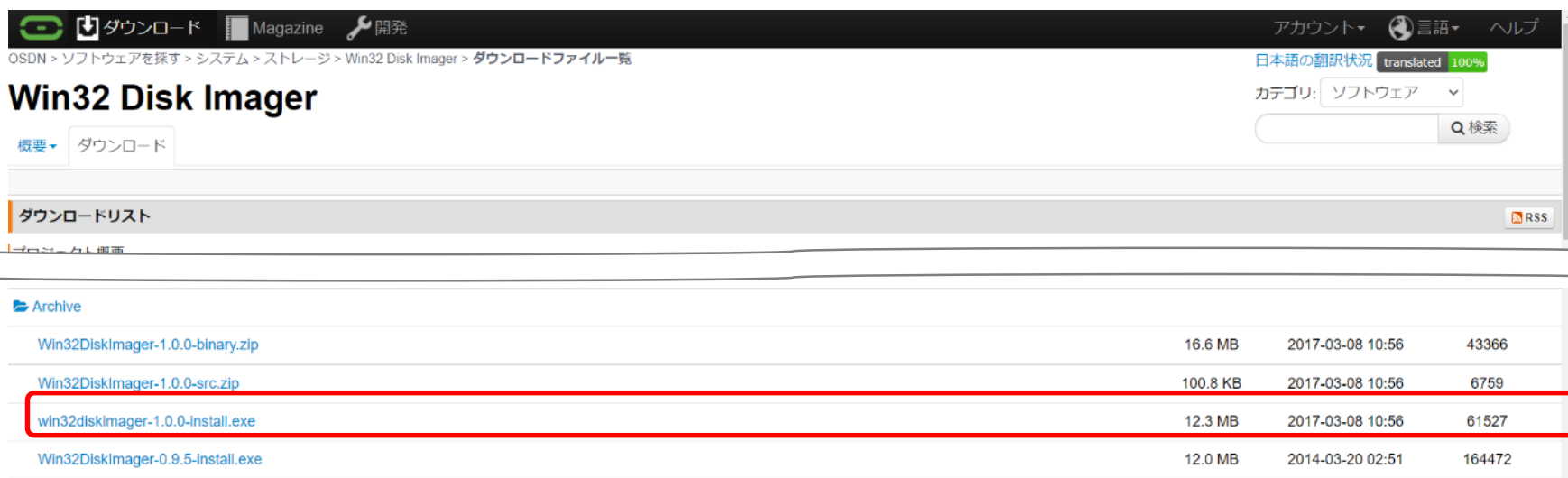
- ② ダウンロードしたZIPファイルを解凍し、中のsetup.exeをダブルクリックして開いて、シインストーラーに従ってインストールします。

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/
※Windowsのみ対応となります。



The screenshot shows the OSDN website for Win32 Disk Imager. The page title is "Win32 Disk Imager". Below the title, there are tabs for "概要" (Overview) and "ダウンロード" (Download). The "ダウンロードリスト" (Download List) section is active, showing a table of download links. The link "win32diskimager-1.0.0-install.exe" is highlighted with a red box.

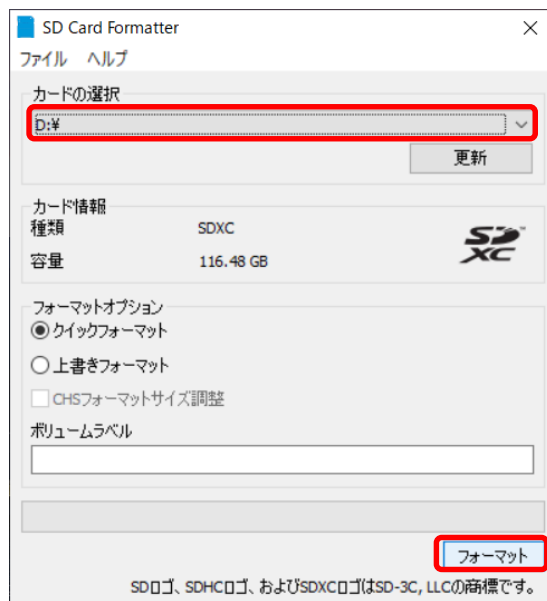
Download Link	File Size	Date	Downloads
Win32DiskImager-1.0.0-binary.zip	16.6 MB	2017-03-08 10:56	43366
Win32DiskImager-1.0.0-src.zip	100.8 KB	2017-03-08 10:56	6759
win32diskimager-1.0.0-install.exe	12.3 MB	2017-03-08 10:56	61527
Win32DiskImager-0.9.5-Install.exe	12.0 MB	2014-03-20 02:51	164472

- 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

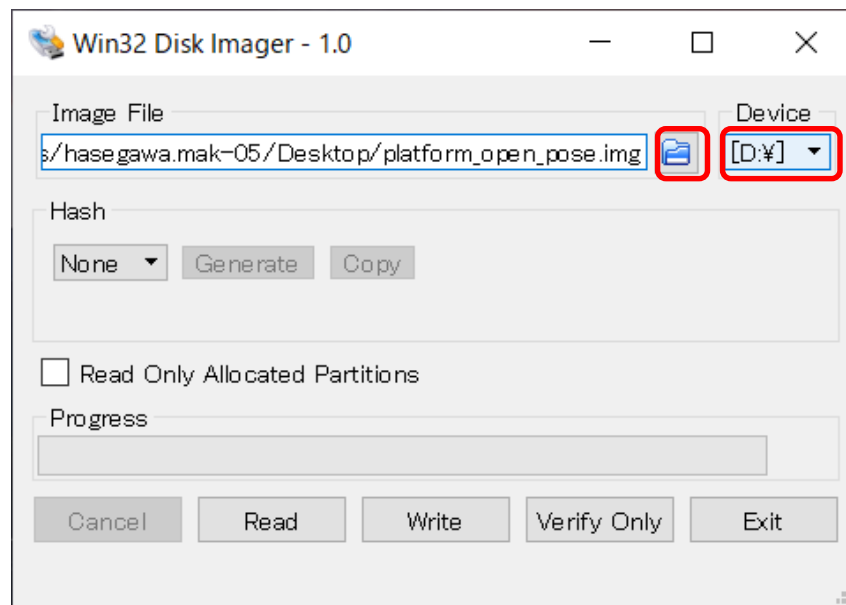
- microSDカードにAIプログラムのイメージファイルを書き込める状態にします。
- ① microSDカードをパソコンに差し込みます。
 - ② (2) ①でインストールして作成されたSD Card Formatterのショートカットをダブルクリックして起動します。
 - ③ microSDカードのドライブ名を確認して、「カードの選択」で同じドライブを選び、「フォーマット」を押します。いくつか確認のポップアップが出ますので、全て「はい」を選択してください。



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

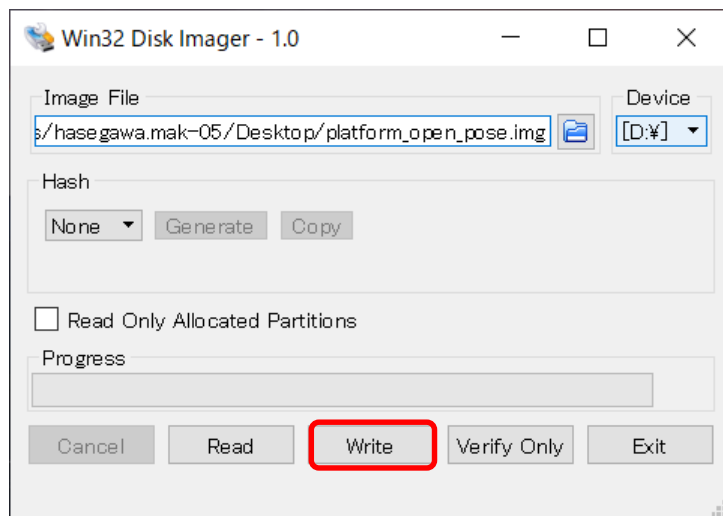
- AIプログラムのイメージファイルをmicroSDカードに書き込み、AIプログラムを実行できる状態にします。
- ① Win32 Disk Imagerのショートカットをダブルクリックして起動します。
 - ② 「Image File」のフォルダマークのボタンを押して、(1)⑧で解凍したimgファイルを選択し、その右側の「Device」でmicroSDカードのドライブを選びます。



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。

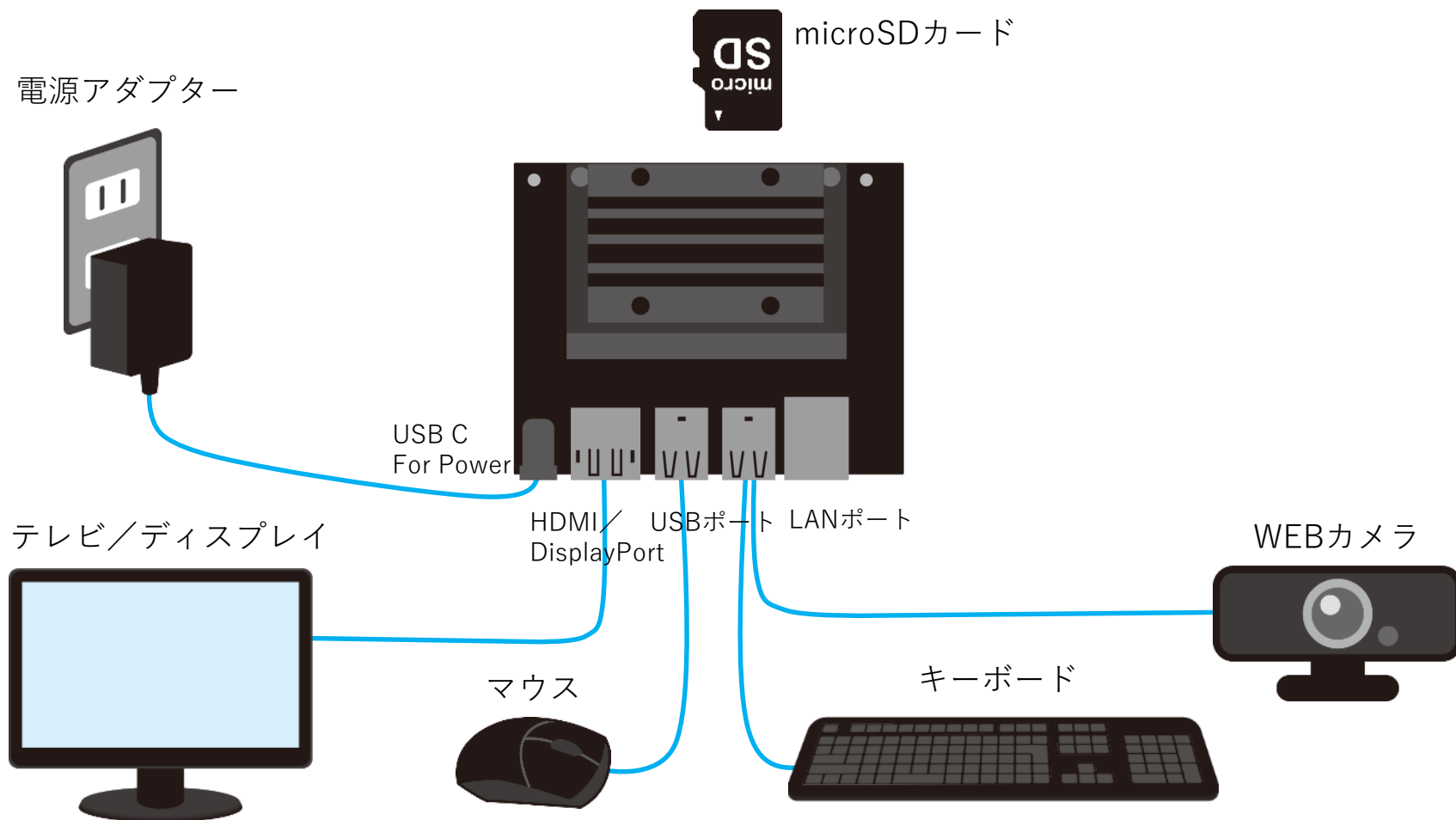


- ④ 書き込んで良いかを確認するポップアップが出てくるので「Yes」を押すと、書き込みが始まります。PCの使用状況によりますが、書き込みに40分程度かかります。
- ⑤ imgファイルの書き込みが終わったらポップアップが出てきますので、「OK」を押して、Win32 Disk Imagerを終了します。
- ⑥ imgファイルを書き込んだmicroSDカードをパソコンから取り出し、完了です。

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 AIプログラムの機能とファイル構成

ファイルに実装されているAIモデルとその機能

モデル	機能	機能詳細
画像認識	画像データ収集	<ul style="list-style-type: none">カメラと連動したウィンドウ上で、指定されたキーボードのキーを押すことによりキャプチャ画像を保存
	学習	<ul style="list-style-type: none">上記で収集した画像データによる画像認識AIモデルの学習を行う上記の学習により作成したモデル (.hd5f) を保存する
	推論	<ul style="list-style-type: none">上記で構築したAIモデルによる画像認識を実行するWEBカメラと連動した認識が可能

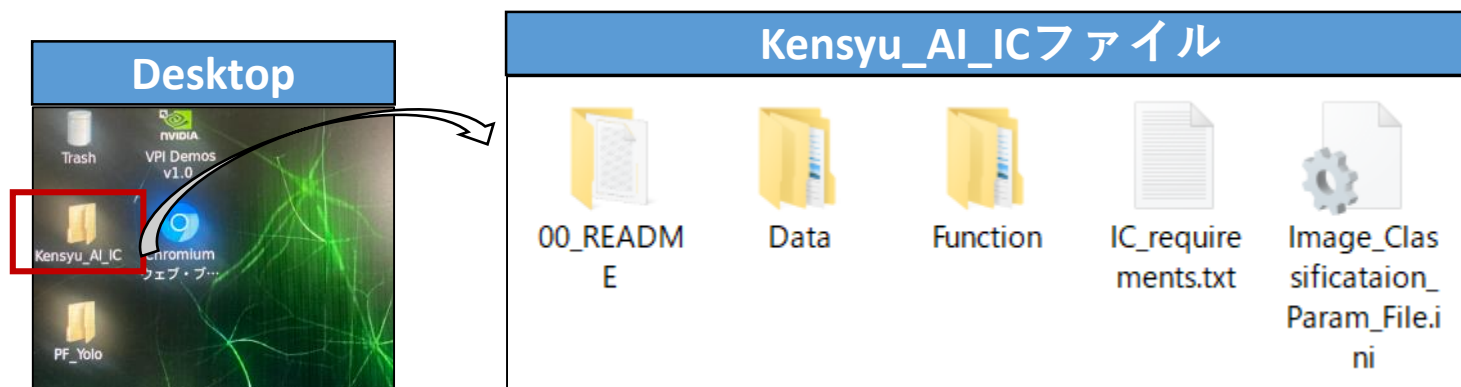
※フォルダ内の以下の2つの画像認識モデルは、学習を実施しなくても使用できます。

- じゃんけんの分類モデル
- ねじの故障判定モデル

7 AIプログラムの機能とファイル構成

ファイル構成 概要

Desktop上に「Kensyu_AI_IC」ファイルを格納しています。

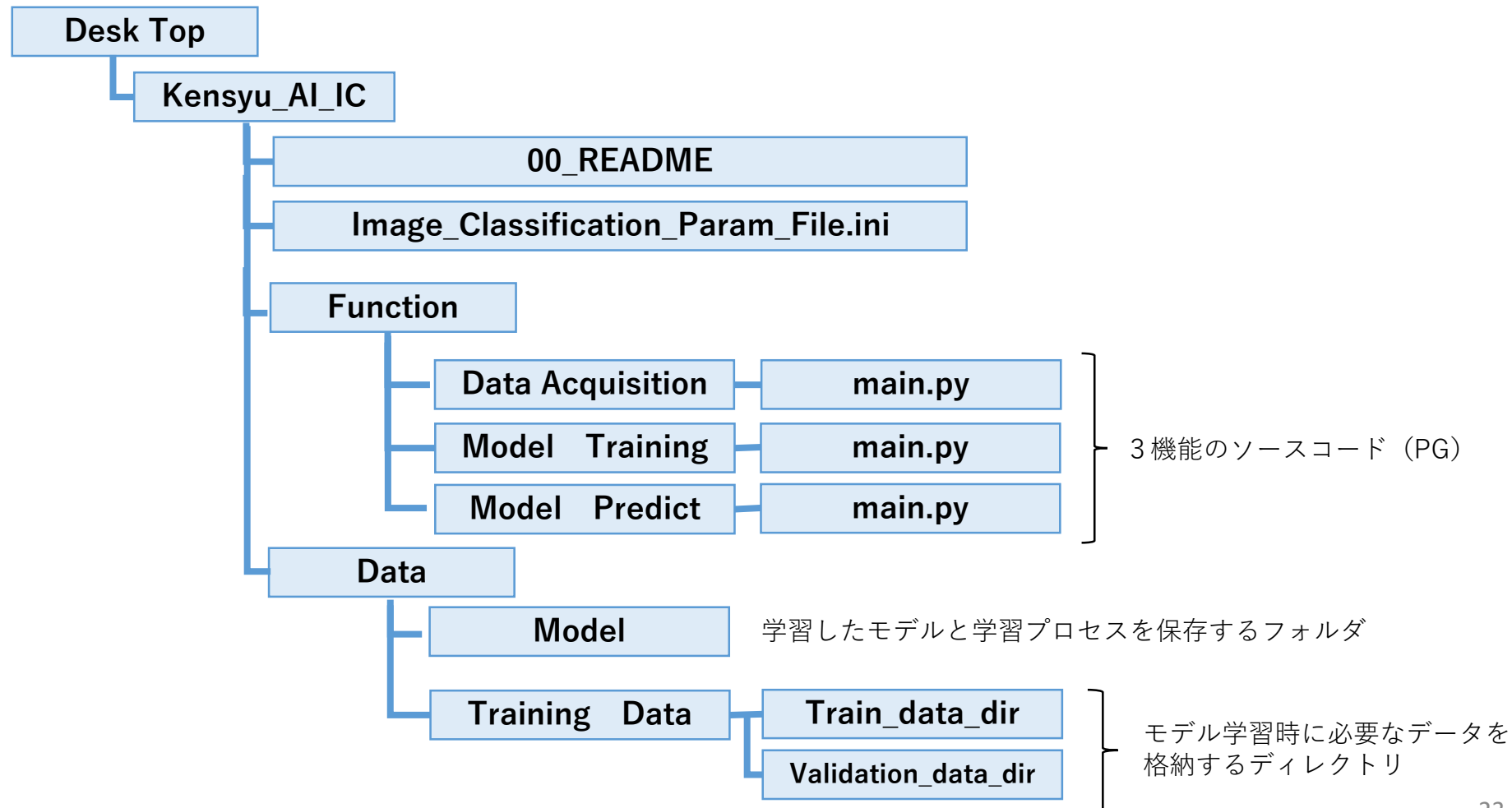


ファイル	ファイル内容
00_README	画像認識プログラムの各機能の実行方法を記載したメモファイル
Data	データ収集／画像認識モデル学習を実行するデータフォルダ
Function	データ収集／画像認識モデル学習／画像分類（推論）を実行するプログラムが格納されているファイル
Image_Classificataion_Param_File.ini	各機能のパラメータ設定ファイル。こちらのメモファイル内容を変更することでパラメータ調整可能
IC_requirement.txt	画像認識を実行する際に必要なpythonライブラリを記載したメモファイル

7 AIプログラムの機能とファイル構成

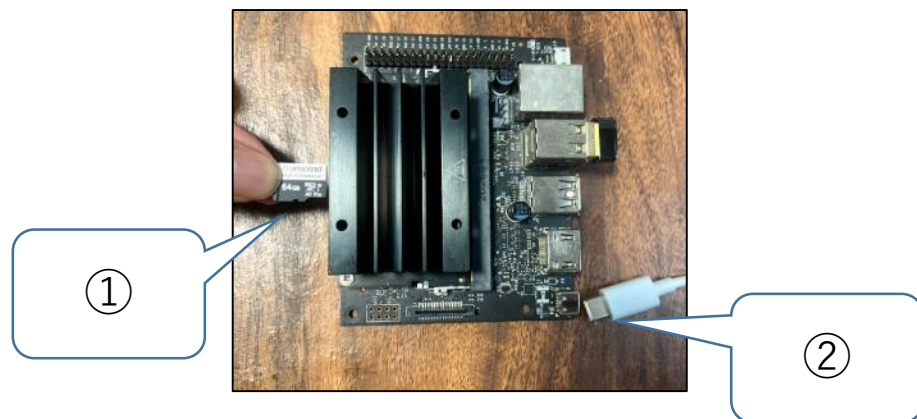
Kensyu_AI_ICファイルのディレクトリ詳細

Desktop上に格納してある「Kensyu_AI_IC」ファイルのディレクトリ構成は以下になります。



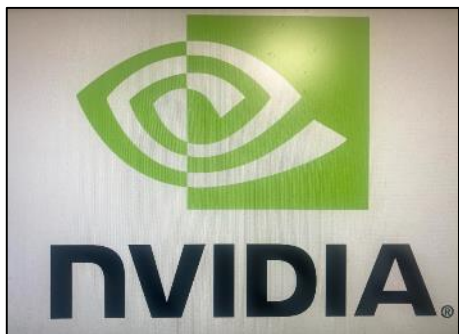
8 Jetsonの起動方法

- 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

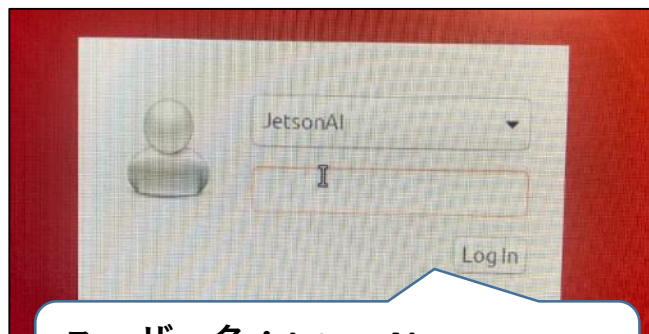


- ③電源を入れるとJetsonが起動し、パスワードを入力するとデスクトップが表示される

Jetson起動



パスワード入力



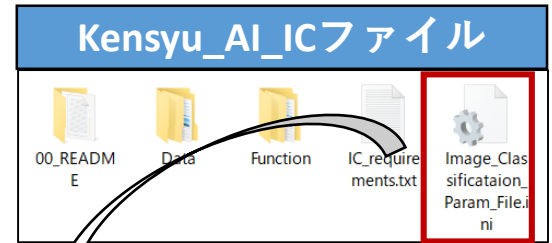
ユーザー名：JetsonAI
パスワード：password01

デスクトップ画面表示



9 画像認識における3機能のパラメータファイル

画像認識3機能を実行するためのパラメータ（設定）ファイル
ファイル名：Image_Classification_Param_File.ini（textファイル）
の中の各パラメータを修正することで、**ソースコードを変更することなくパラメータを変更できます**



Image_Classification_Param_File.iniファイル

```
#training data directory param
[IC_Data]
Train_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/sample/Train_data_dir
Validation_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/sample/Validation_data_dir
Test_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/sample/Test_data_dir
#Image Classification model directory and name
[IC_Model]
Model_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/Model
Model_name = model.hdf5
Predict_Model_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/Model/sample_model/neji_model
Predict_Model_name = neji_model.hdf5
#Training Param
[IC_Param]
#image size
Img_w = 224
Img_h = 224
#Number of epoch
Epoch = 2
#Learning rate
Lr = 1e-6
```

収集したデータの保存／
学習データの場所（ディ
レクトリ）を表す

学習したモデルを保存するディレクトリ
学習したモデルの名前を表すパラメータ ※)モデル名は必ず.hdf5形式

推論で使用する画像認識モデルの保存場所（ディレクトリ）と
使用するモデルの名前を表すパラメータ

データ収集／モデル学習の際に行う「前処理：画像サイズ変更」の際の幅と高さを表すパラメータ
※) 整数値で100-1000までの値推奨

学習回数を表すパラメータ ※) 学習回数を増やすと学習時間が長くなる／整数値入力

学習率を表すパラメータ ※) 通常 1e-8/ 1e-7/1e-6/ 1e-5/ 1e-3/ 1e-2（指数オーダー）を使用

！ 注意点

✓ 青字がユーザーで変更できる箇所です。青字以外の部分を変更すると、エラーが発生し、各機能が実行できなくなりますので、変更しないでください。

10 画像データ収集

画像データ収集 実行方法サマリー

(1) パラメータファイル準備 (詳細はp.27)

データ収集機能により実行する「画像の前処理」と「画像の保存場所」を設定するパラメータを調整する

(2) コマンド入力 (詳細はp.28)

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/Kensyu_AI_IC/Function/DataAcquisition`

③mainファイルの実行

コマンド：`python3 main.py`

④ラベル名の入力（撮影した画像をどのディレクトリに保存するか指定）

※ Jetsonの負荷状況によりですが、実行されるまで30-60秒程度かかります。

※ ラベル名は半角ローマ字入力のみになります。例：choki

※ 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います。

(3) 画像データ収集の実行 (詳細はp.29)

「main.py」ファイルを実行し、カメラ起動～撮影／保存までを実行

(4) 画像データ収集結果 (詳細はp.30)

カメラから入手した映像から写真を撮影し、指定したディレクトリに保存する

10 画像データ収集

(1) パラメータファイル準備

- ① デスクトップ上の「Kensyu_AI_IC」フォルダの中から、「Image_Classification_Param_File.ini」ファイルを選択して開きます。
- ② 「Image_Classification_Param_File.ini」ファイル内の、データ収集機能により実行する「画像の前処理」と「画像の保存場所」を設定するパラメータを確認/変更します。

Image_Classification_Param_File.iniファイル

```
#training data directory param
[IC_Data]
Train_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/sample/Train_data_dir
Validation_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/sample/Validation_data_dir
Test_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/sample/...
#Image Classification model directory and name
[IC_Model]
Model_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/Model
Model_name = model.hdf5
Predict_Model_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/Model/sample_model/neji_model
Predict_Model_name = neji_model.hdf5
#Training Param
[IC_Param]
#image size
Img_w = 224
Img_h = 224
#Number of epoch
Epoch = 2
#Learning rate
Lr = 1e-6
```

撮影した画像の
保存場所（ディレクトリ）を入力

画像を保存する際に行う前処理（画像サイズの調整）の画像幅/広さを入力
※整数値入力,大きすぎるとモデル学習の際に時間がかかる

10 画像データ収集

(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/Kensyu_AI_IC/Function/DataAcquisition`

③ mainファイルの実行

コマンド: `python3 main.py`

③ ラベル名の入力 (撮影した画像をどのディレクトリに保存するか指定)

※ 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

※ ラベル名は半角ローマ字入力のみになります 例: choki

⇒ 入力したラベルのフォルダがない場合は新しくフォルダを作成し、該当フォルダに保存

入力したラベルがある場合は、該当の既存フォルダに保存

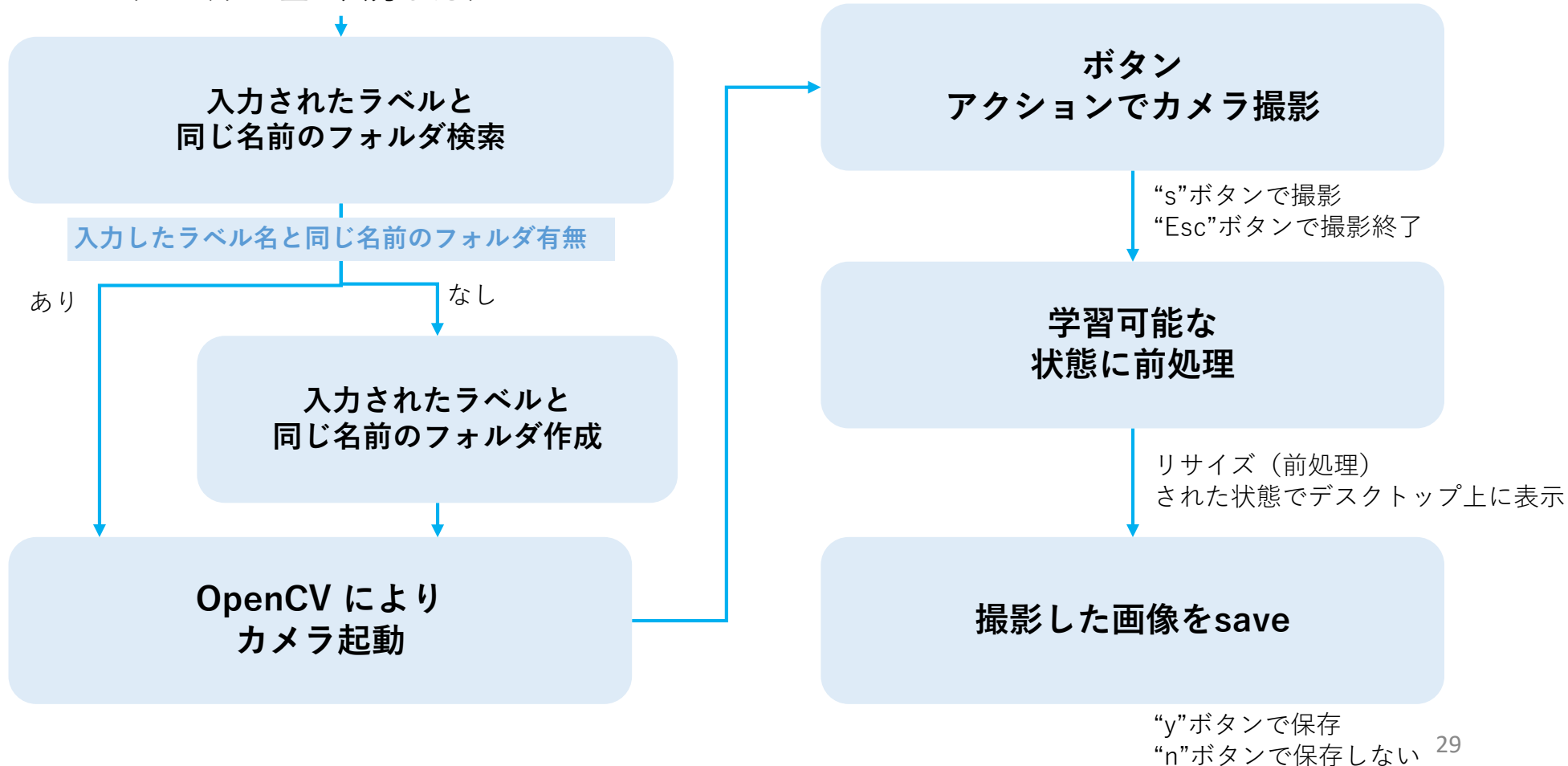
```
jetsonai@jetsonai-desktop: ~/Desktop/Kensyu_AI_IC/Function/DataAcquisition
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
jetsonai@jetsonai-desktop: $ cd /home/jetsonai/Desktop/Kensyu_AI_IC/Function/DataAcquisition
jetsonai@jetsonai-desktop: ~/Desktop/Kensyu_AI_IC/Function/DataAcquisition$ python3 main.py
Label Name is   aaa
```

10 画像データ収集

(3) 画像データ収集の実行

「main.py」ファイルを実行すると、以下のフローで画像データ収集が実行されます。

ターミナル上で入力したラベル



10 画像データ収集

(4) 画像データ収集結果

カメラから入手した映像から写真を撮影し、指定したディレクトリに保存します。

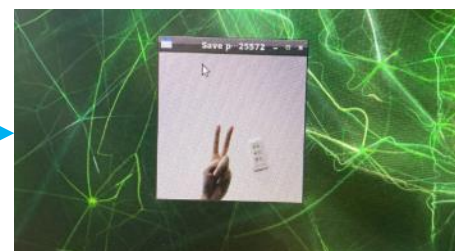
カメラ映像のデスクトップ表示



“Esc”ボタンで撮影終了

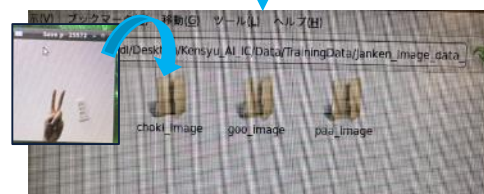
データ収集終了

“s”ボタンで撮影



“y”ボタンで保存

“n”ボタンで保存しない



撮影画像を
保存しない

! 注意点

- ✓ タブを多く開いた状態や他のアプリケーション実行時では処理速度が遅くなるため、複数の処理（WEB検索、長時間の使用等）を避けることを推奨します。
- ✓ Jetsonの負荷を軽減するために、冷却ファンの使用を推奨します。

11 画像認識（学習）

画像認識（学習）実行方法サマリー

（1）学習データ準備（詳細はp.32）

画像認識モデルで学習するデータの準備を行う

（2）パラメータファイル準備（詳細はp.33）

画像認識モデルで学習するための、学習回数等のパラメータを確認／変更

（3）コマンド入力（詳細はp.34）

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/Kensyu_AI_IC/Function/DataAcquisition`

③学習リソース確保

コマンド：`export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1`

④GPU(CUDA)指定

コマンド：`export CUDA_VISIBLE_DEVICES="1"`

⑤mainファイルの実行 ※) 学習条件によって学習時間は異なるが、学習開始までは30-60秒程で起動

コマンド：`python3 main.py`

（4）画像認識（学習）の実行（詳細はp.35）

「main.py」ファイルを実行し、モデル学習～モデル保存までを実行

（5）画像認識（学習）結果（詳細はp.36）

収集した画像を使用して、画像認識モデルを学習し、モデルを保存

11 画像認識（学習）

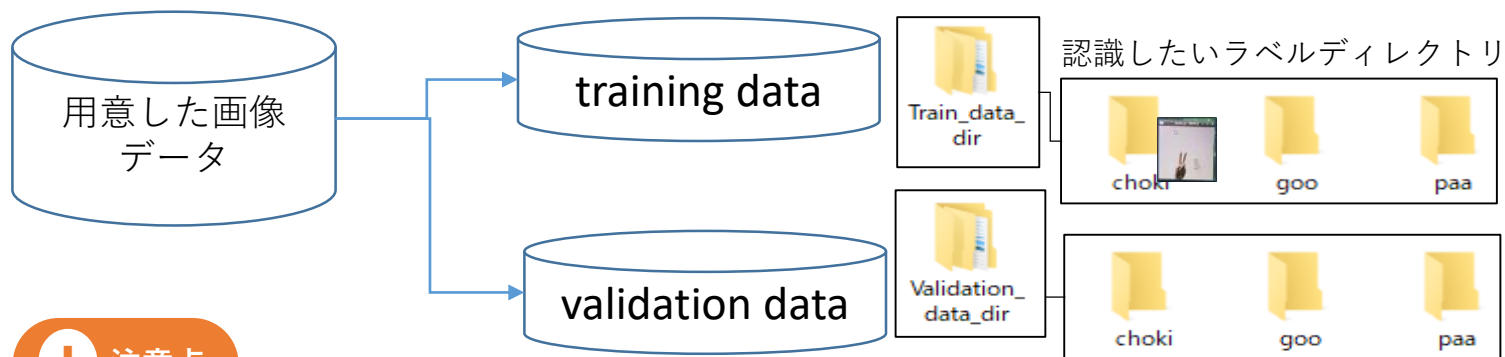
（1）学習データ準備

AIモデルを作成する際には、通常「training data」と「validation data」を準備する必要があります。ここでは、最初に画像認識モデルで学習するため、この2つのデータセット準備を行います。

train data（訓練データ）	AIモデルのパラメータを更新するための学習用データ。
validation data（検証データ）	AI学習が上手く出来ているかを確認するための検証用データ。 このデータは検証のために使用するので、学習は行わない。

参考URL <https://algorithm.joho.info/programming/python/keras-train-validation-test-dataset/>

AI学習で使用するデータから、「AI学習に使用したいデータ（training data）」と「学習時のモデル評価に使用したいデータ（validation data）」に**ディレクトリを分けてください**。



！ 注意点

- ✓ 通常、training dataとvalidation dataの割合は、**training dataのデータ量を多く**学習に使用します（7:3 or 8:2の割合が良く使われる）。**各ラベルフォルダが0枚にならないようデータを準備してください**。
- ✓ 学習の際、training dataとvalidation dataの**ラベルが一致しているか確認してください**。
- ✓ training dataとvalidation dataで**どちらにも同じ画像データを使用しないでください（両方のフォルダに同じ画像データが存在しないようにしてください）**。

11 画像認識（学習）

（2）パラメータファイル準備

- ① デスクトップ上の「Kensyu_AI_IC」フォルダの中から、「Image_Classification_Param_File.ini」ファイルを選択して開きます。
- ② 「Image_Classification_Param_File.ini」ファイル内の、画像認識モデル学習で設定するパラメータを確認／変更します。

Image_Classification_Param_File.iniファイル

```
#training data directory param
[IC_Data]
Train_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/sample/Train_data_dir
Validation_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/sample/Validation_data_dir
Test_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/sample/Test_data_dir
#Image Classification model directory and name
[IC_Model]
Model_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/Model
Model_name = model.hdf5
Predict_Model_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/Model/sample_model/neji_model
Predict_Model_name = neji_model.hdf5
#Training Param
[IC_Param]
#image size
Img_w = 224
Img_h = 224
#Number of epoch
Epoch = 2
#Learning rate
Lr = 1e-6
```

学習に使用するtraining dataと validation dataのディレクトリを指定

学習したモデル／学習プロセスを保存する場所、モデル名を設定する
※ モデル名は必ず.hdf5形式

モデルを学習する際に行う前処理（画像サイズの調整）の画像幅／広さを入力
※ 整数値入力,大きすぎるとモデル学習の際に時間がかかる

学習回数を設定する
※ 学習回数を増やすと学習時間が長くなる／整数値入力

学習率（重みを更新する幅）を設定する
※ 通常 1e-8/ 1e-7/1e-6/ 1e-5/ 1e-3/ 1e-2等の指数オーダーの値を使用

11 画像認識（学習）

（3）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/Kensyu_AI_IC/Function/DataAcquisition`

③ 学習リソース確保

コマンド：`export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1`

④ GPU(CUDA)指定

コマンド：`export CUDA_VISIBLE_DEVICES="1"`

⑤ mainファイルの実行 ※) 学習条件によって学習時間は異なるが、学習開始までは30-60秒程で起動

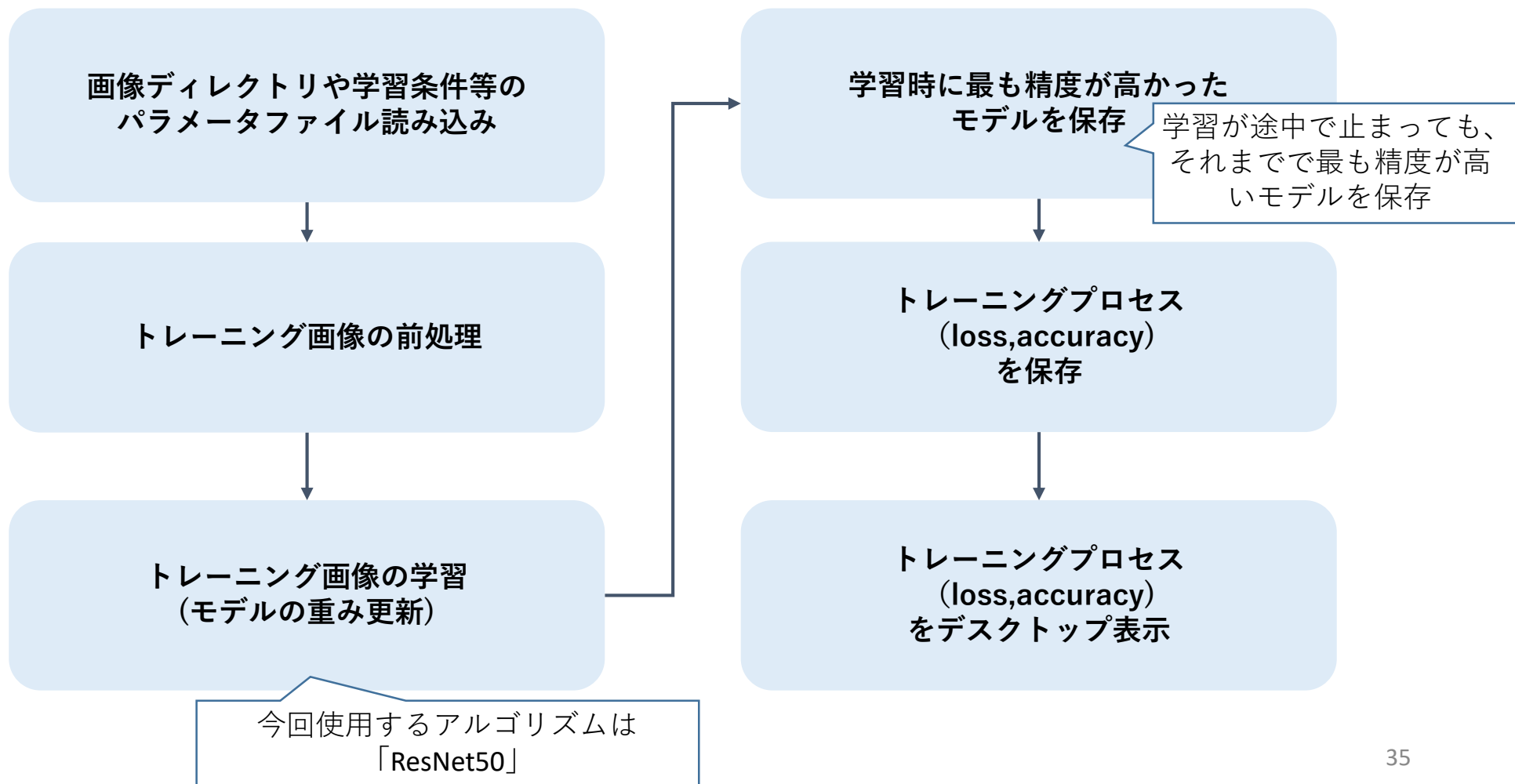
コマンド：`python3 main.py`

```
jetsonai@jetsonai-desktop: ~/Desktop/Kensyu_AI_IC/Function/ModelTrain
jetsonai@jetsonai-desktop: $ cd /home/jetsonai/Desktop/Kensyu_AI_IC/Function/ModelTrain
jetsonai@jetsonai-desktop: ~/Desktop/Kensyu_AI_IC/Function/ModelTrain$ export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1
jetsonai@jetsonai-desktop: ~/Desktop/Kensyu_AI_IC/Function/ModelTrain$ export CUDA_VISIBLE_DEVICES="1"
jetsonai@jetsonai-desktop: ~/Desktop/Kensyu_AI_IC/Function/ModelTrain$ python3 main.py
```

11 画像認識（学習）

（４）画像認識（学習）の実行

「main.py」ファイルを実行すると、以下のフローで学習が実行されます。



11 画像認識（学習）

（5）画像認識（学習）結果

学習時

```
-----Library import finish-----
Label is
['paa_image', 'choki_image', 'goo_image', 'Choki']
Num Label
4
-----change jpg finish-----
-----Model train START-----
train sample number
10
validation sample number
6
Found 10 images belonging to 4 classes.
Found 6 images belonging to 4 classes.
2021-07-28 21:42:23.492945: I tensorflow/compiler/jit/xla_cpu_device.cc:41] Not creating XLA devices, tf_xla_enable_xla_devices not set
2021-07-28 21:42:23.494425: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not creating XLA devices, tf_xla_enable_xla_devices not set
/usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/training.py:1844: UserWarning: 'Model.fit_generator' is deprecated and
will be removed in a future version. Please use 'Model.fit', which supports generators.
warnings.warn("'Model.fit_generator' is deprecated and
! (registered 2)
!021-07-28 21:42:31.672303: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes are enable
!
!2021-07-28 21:42:31.673562: W tensorflow/core/platform/profile_utils/cpu_utils.cc:116] Failed to find bogomips or clock in /proc/cpuinfo; c
cannot determine CPU frequency
epoch 1/2
0/10 [=====] - 53s 3s/step - loss: 33.3111 - accuracy: 0.1673 - val_loss: 3259.3770 - val_accuracy: 0.3333
epoch 2/2
0/10 [=====] - 22s 2s/step - loss: 111.4680 - accuracy: 0.5218 - val_loss: 50275820.0000 - val_accuracy: 0.3333
-----Model train Finish-----
```

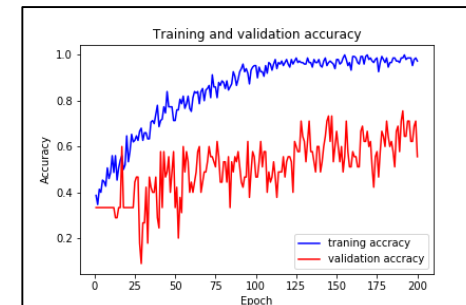
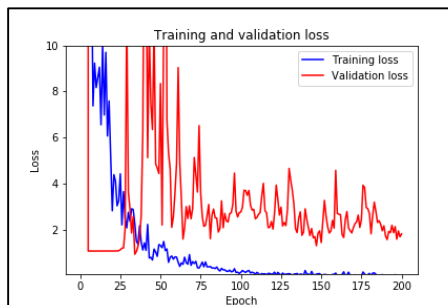
学習中では、ターミナル上にエポック数毎の
トレーニングプロセス（lossとaccuracy）を表示

参考）学習時間について

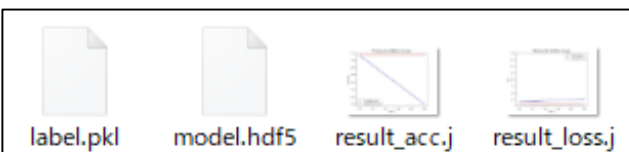
学習条件：画像サイズ224×224 ， 学習率：1e-6

Train data:150枚 Validation data:

学習終了後



epoch（エポック）毎のlossとaccuracy(正解率)の
グラフを保存／デスクトップ表示



設定したモデル保存ディレクトリに以下を保存

- 学習した画像認識モデル(.hdf5)
- 画像認識で分類するラベルファイル(.pkl)
- epoch毎のlossとaccuracy(正解率)のグラフ図

！ 注意点

- ✓ 学習時間は、学習データ数／質（画素値等）、エポック数等によって異なります
- ✓ Jetsonの負荷状況によっても学習時間は異なります
 - ✓ タブを多く開いた状態や他のアプリケーション実行時では処理速度が遅くなるため、複数の処理（WEB検索、長時間の使用等）を避けることを推奨します。
 - ✓ Jetsonの負荷を軽減するために、冷却ファンの使用を推奨します。

11 画像認識（学習）

（参考）学習パラメータ毎の学習時間について

学習パラメータ「学習データ枚数」, 「学習率」, 「画像サイズ」の設定によって学習時間は異なるため、下記を参考に学習パラメータとエポック数（学習回数）を設定して学習時間をコントロールしてください。

学習データ枚数	学習率	画像サイズ	1エポックあたりの学習時間
Train data: 150枚 Validation data: 45枚	1e-6	224 × 224	5分 / 1エポック
Train data: 60枚 Validation data: 15枚	1e-6	224 × 224	2分 / 1エポック
Train data: 60枚 Validation data: 15枚	1e-3	224 × 224	1.5分 / 1エポック
Train data: 60枚 Validation data: 15枚	1e-3	100 × 100	0.75分 / 1エポック

※あくまで目安の時間になります。

Jetsonの負荷状況によっても学習時間は異なりますのでご注意ください

- タブを多く開いた状態や他のアプリケーション実行時では処理速度が遅くなるため、複数の処理（WEB検索、長時間の使用等）を避けることを推奨します。
- Jetsonの負荷を軽減するために、冷却ファンの使用を推奨します。

12 画像認識（推論）

画像認識（推論）実行方法サマリー

（1）パラメータファイル準備（詳細はp.39）

画像認識モデルで推論するための、使用するモデルの保存場所等のパラメータを確認／変更

（2）コマンド入力（詳細はp.40）

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/Kensyu_AI_IC/Function/ModelPredict`

③学習リソース確保

コマンド：`export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1`

④GPU(CUDA)指定

コマンド：`export CUDA_VISIBLE_DEVICES="1"`

⑤mainファイルの実行 ※) 推論起動までは30-60秒程で起動

コマンド：`python3 main.py`

（3）画像認識（推論）の実行（詳細はp.41）

「main.py」ファイルを実行し、カメラ起動～推論結果表示までを実行

（4）画像認識（推論）結果（詳細はp.42）

構築したAIモデルによる画像認識を実行する。WEBカメラと連動した認識が可能。

12 画像認識（推論）

（1）パラメータファイル準備

- ① デスクトップ上の「Kensyu_AI_IC」フォルダの中から、「Image_Classification_Param_File.ini」ファイルを選択して開きます。
- ② 「Image_Classification_Param_File.ini」ファイル内の、画像認識（推論）で設定するパラメータを確認／変更します。

Image_Classification_Param_File.iniファイル

```
#training data directory param
[IC_Data]
Train_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/
Validation_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/
Test_data_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/TrainingData/sample_model/
#Image Classification model directory and name
[IC_Model]
Model_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/Model
Model_name = model.hdf5
Predict_Model_dir = /home/jetsonai/Desktop/Kensyu_AI_IC/Data/Model/sample_model/neji_model
Predict_Model_name = neji_model.hdf5
#Training Param
[IC_Param]
#image size
Img_w = 224
Img_h = 224
#Number of epoch
Epoch = 2
#Learning rate
Lr = 1e-6
```

推論で使用するモデル／分類ラベルが保存されているディレクトリ、モデル名を設定

※）検証で作成した下記2画像認識モデル

✓ じゃんけんの分類モデル

✓ ねじの故障判定モデル

を格納しておりますので、学習を実施しなくても上記2モデルは使用できます。

推論する際に行う前処理（画像サイズの調整）の画像幅／広さを入力

※学習時に設定したサイズと同じにする

※整数値入力,大きすぎるとモデル学習の際に時間がかかる

12 画像認識（推論）

（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/Kensyu_AI_IC/Function/ModelPredict`

③ 学習リソース確保

コマンド：`export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1`

④ GPU(CUDA)指定

コマンド：`export CUDA_VISIBLE_DEVICES="1"`

⑤ mainファイルの実行 ※) 推論までは30-60秒程度で起動

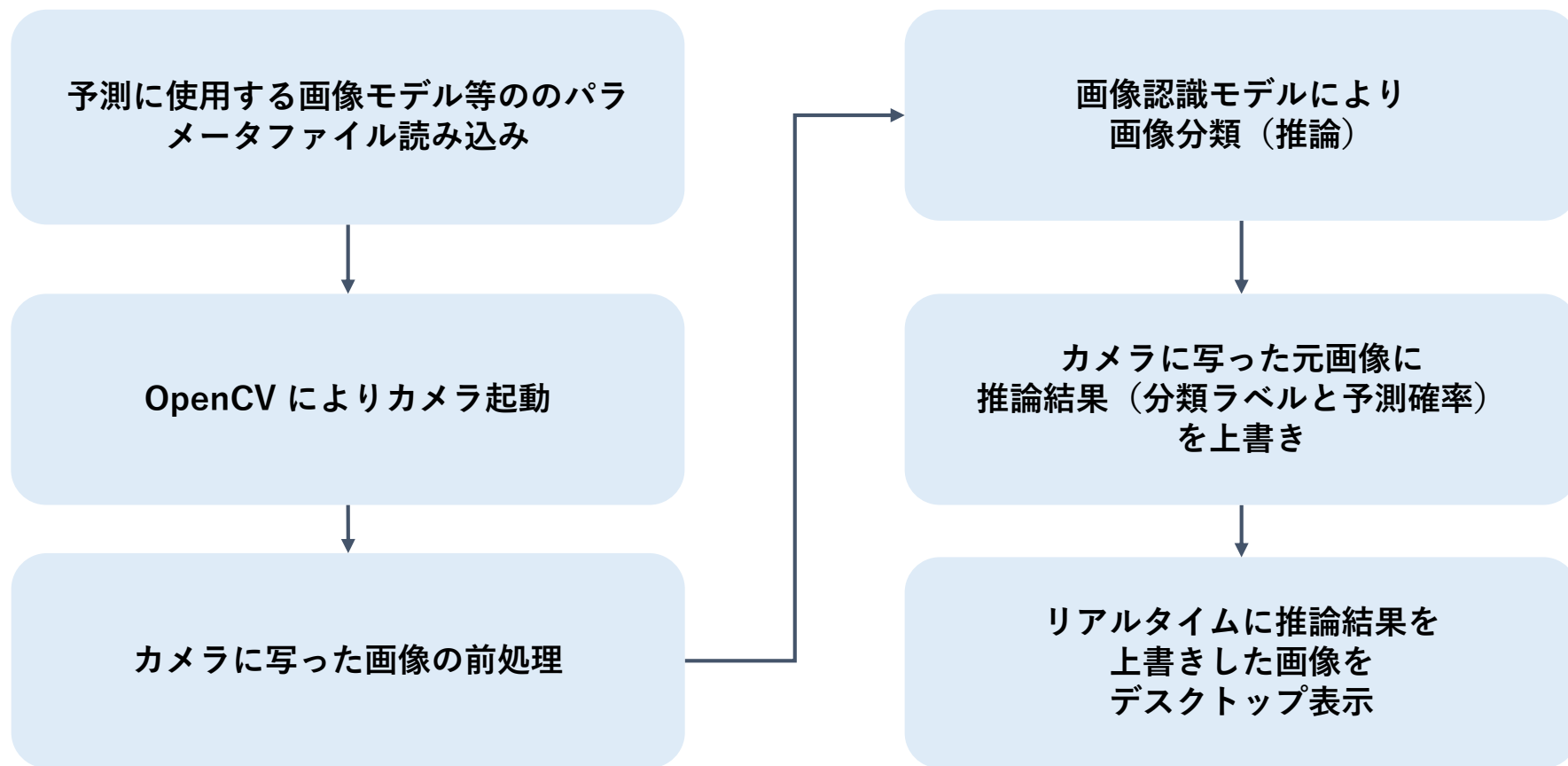
コマンド：`python3 main.py`

```
jetsonai@jetsonai-desktop: ~/Desktop/Kensyu_AI_IC/Function/ModelPredict
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
jetsonai@jetsonai-desktop: $ cd /home/jetsonai/Desktop/Kensyu_AI_IC/Function/ModelPredict
jetsonai@jetsonai-desktop: ~/Desktop/Kensyu_AI_IC/Function/ModelPredict$ export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1
jetsonai@jetsonai-desktop: ~/Desktop/Kensyu_AI_IC/Function/ModelPredict$ export CUDA_VISIBLE_DEVICES="1"
jetsonai@jetsonai-desktop: ~/Desktop/Kensyu_AI_IC/Function/ModelPredict$ python3 main.py
```


12 画像認識（推論）

（3）画像認識（推論）の実行

「main.py」ファイルを実行すると、以下のフローで推論が実行されます。

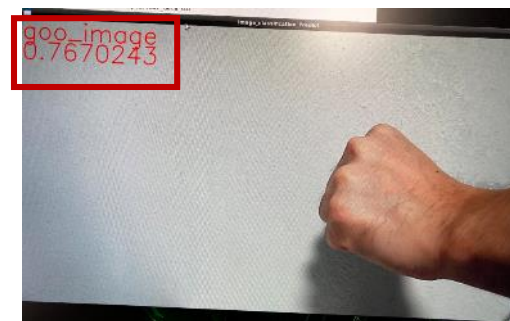
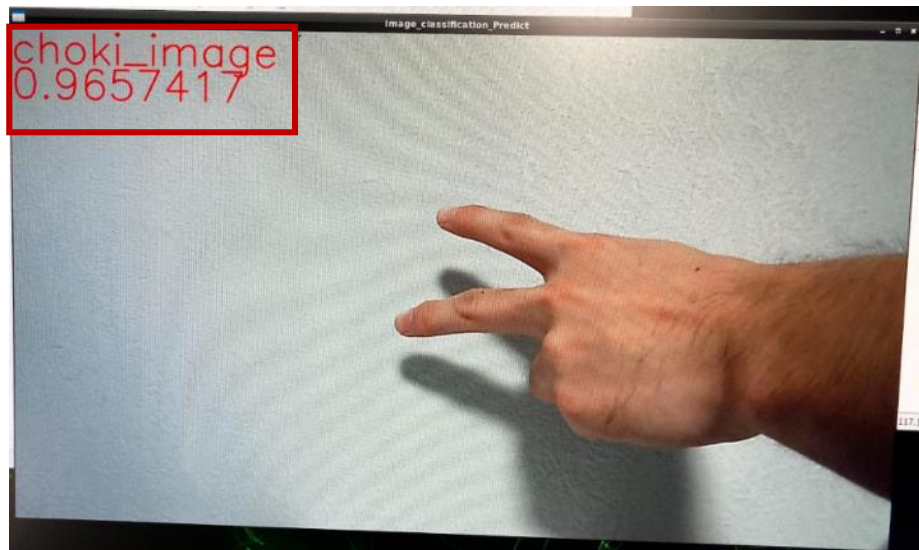


“ESC key”を押すと推論終了

12 画像認識（推論）

（4）画像認識（推論）結果

カメラで写っている映像から画像分類（推論）を行い、左上に「**分類ラベル**」, 「**予測確率（最大値1最小値0）**」を記載した画像を**デスクトップ**にリアルタイムで表示します。



！ 注意点

- ✓ 約1秒程のタイムラグが生じる場合もありますが、ほとんどリアルタイムに近い形で予測可能。
- ✓ タブを多く開いた状態や他のアプリケーション実行時では処理速度が遅くなるため、複数の処理（WEB検索、長時間の使用等）を避けることを推奨します。
- ✓ Jetsonの負荷を軽減するために、冷却ファンの使用を推奨します。

13 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せんが、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8章を参考にしてください。

②起動したJetsonから、デスクトップ上のソースコードを含むファイル「Kensyu_AI_IC」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「Kensyu_AI_IC」ファイル内の環境ファイル「IC_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、「IC_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)	ResNet50 (画像認識アルゴリズム)
ライブラリ群	「Kensyu_AI_IC」ファイル内の環境ファイル「IC_requirements.txt」を参考にしてください

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

実行コマンド：python3 main.py

