14. 組み込み

組み込み（１）

目的と目標

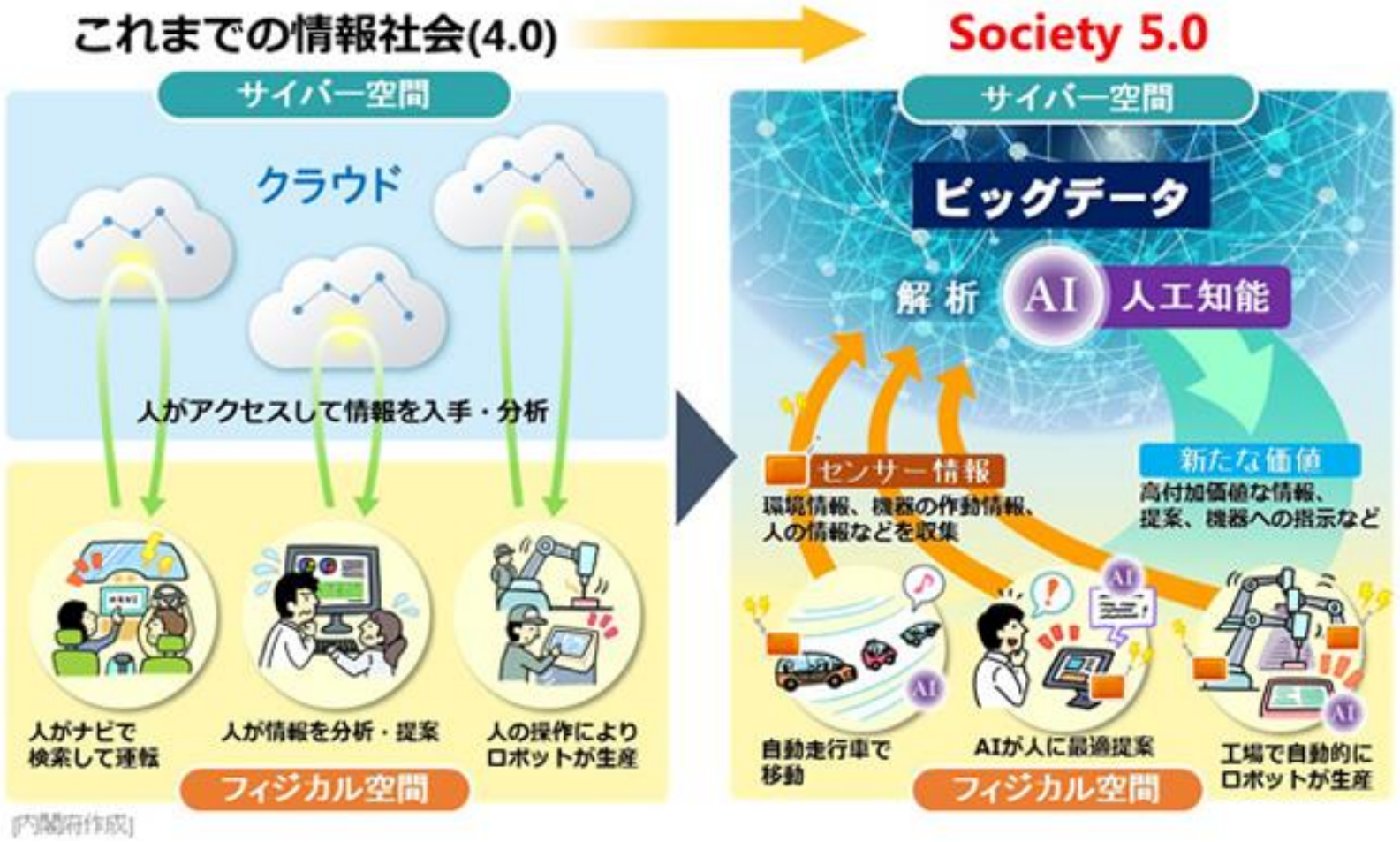
目的：機械学習を搭載するエッジコンピューティングへの理解

- Raspberry Piがセットアップできる
- Linux系OS上にTensorFlow環境を構築できる
- Google colabで作成した学習結果（重みファイル）を別のデバイスで利用できる

目標：Raspberry Pi上でのリアルタイム検出の実現

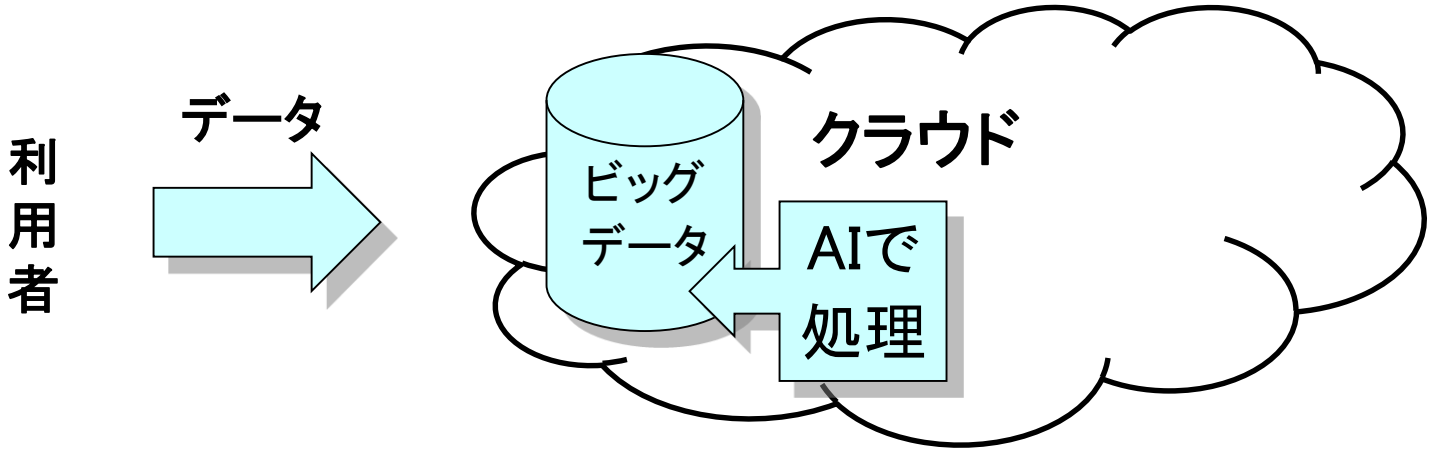
- 演習：10/2に実施したリアルタイム検出をRaspberry Pi上で実現
  ＋MariaDBに保存＋E-mailで送信
- Extra演習

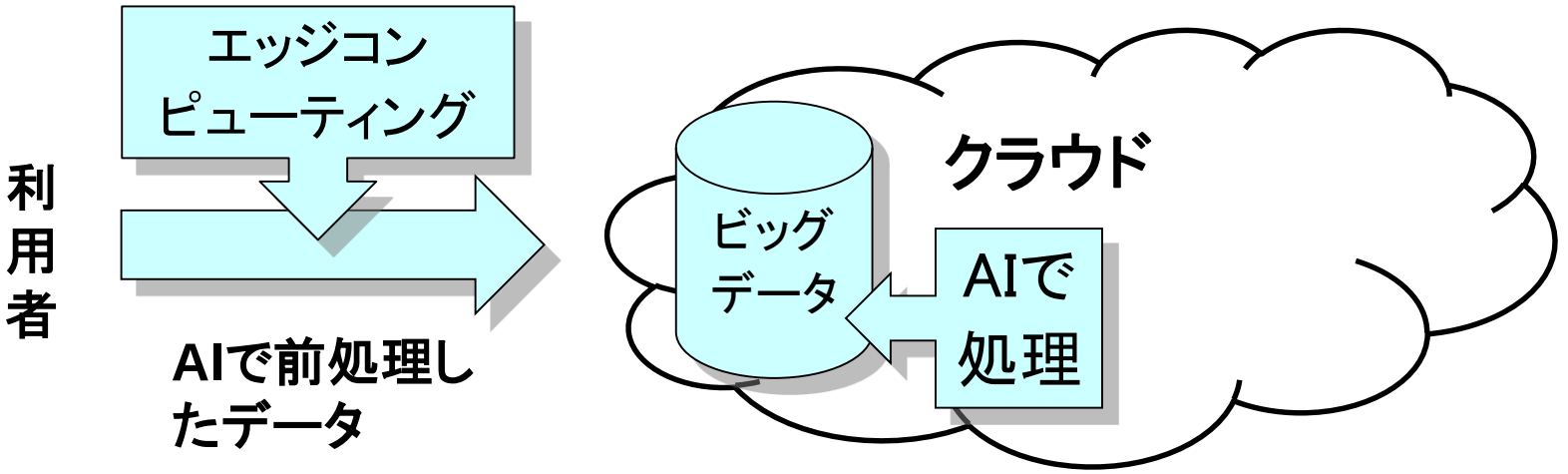
概要



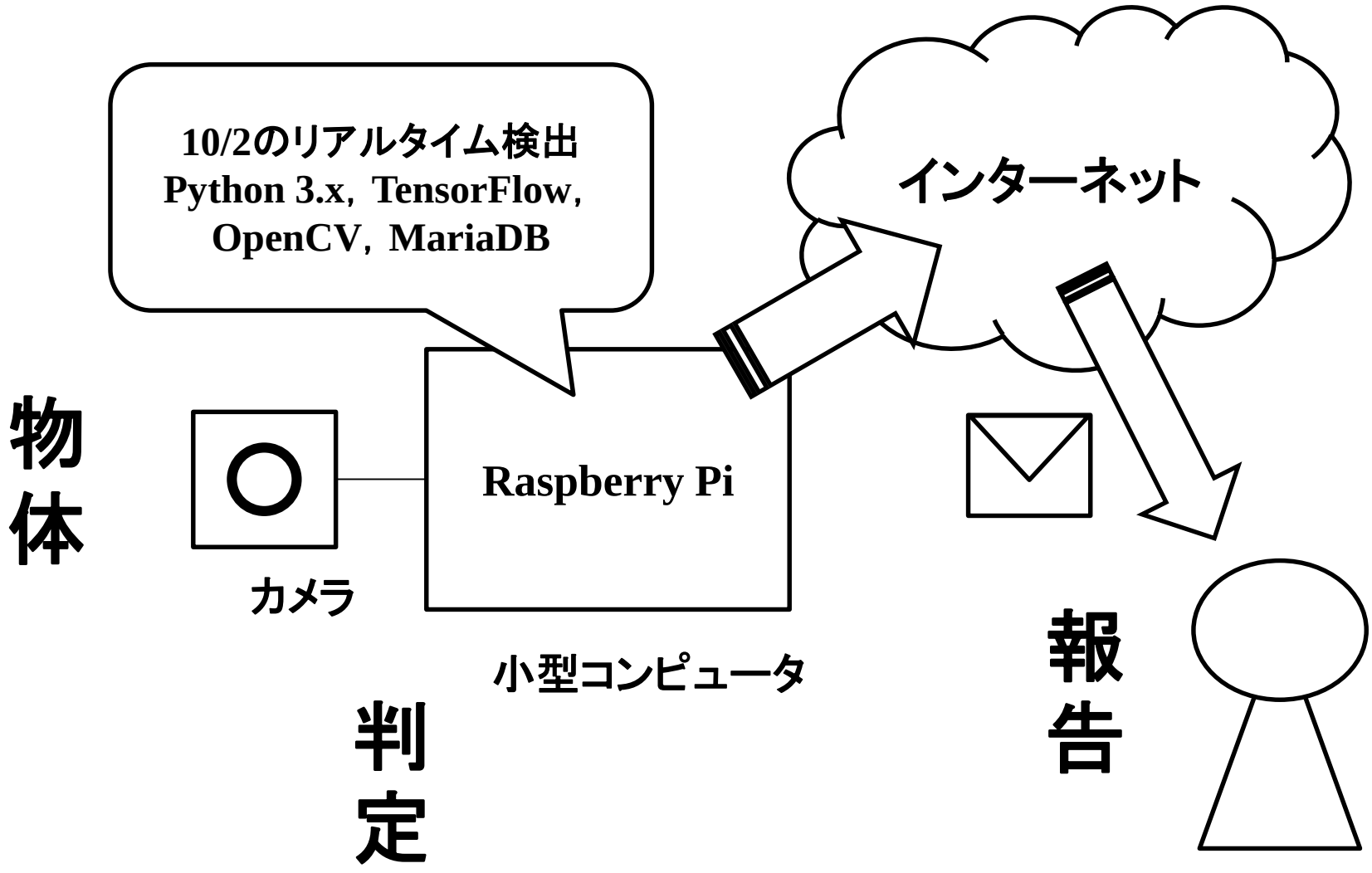
内閣府　Society 5.0　https://www8.cao.go.jp/cstp/society5_0/index.html

エッジコンピューティング





カメラ＋小型コンピュータ



機器

Raspberry Pi 4 Model B セット内容

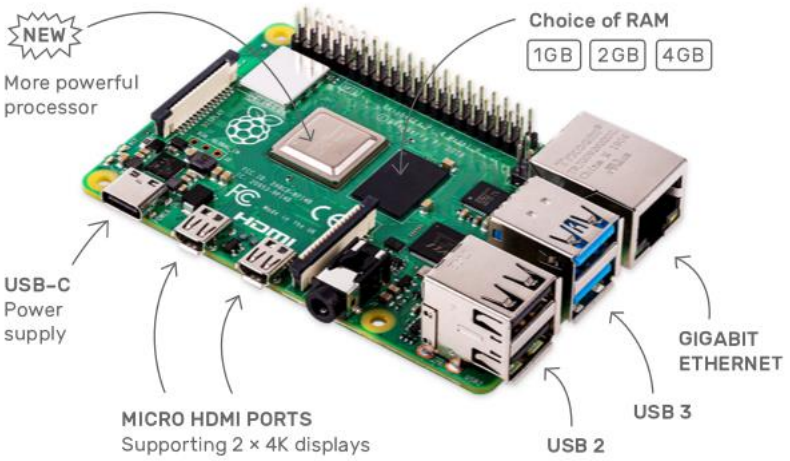
- ラズベリーパイ4B（4GB RAM）
  - セットによってはラズベリーパイ4B用ケース付き
- microSDカード（32GB）
  - 必要があればBackup用に別のカードを準備
- カードリーダー
- 放熱対策のためのヒートシンク、ファン
- 5.1V3.0A USB Type-C電源アダプタ
  - セットによってはスイッチ付き電源ケーブル付き
- MicroHDMI-to-HDMIケーブル
- その他（LANケーブル、プラスドライバなど）

セット以外

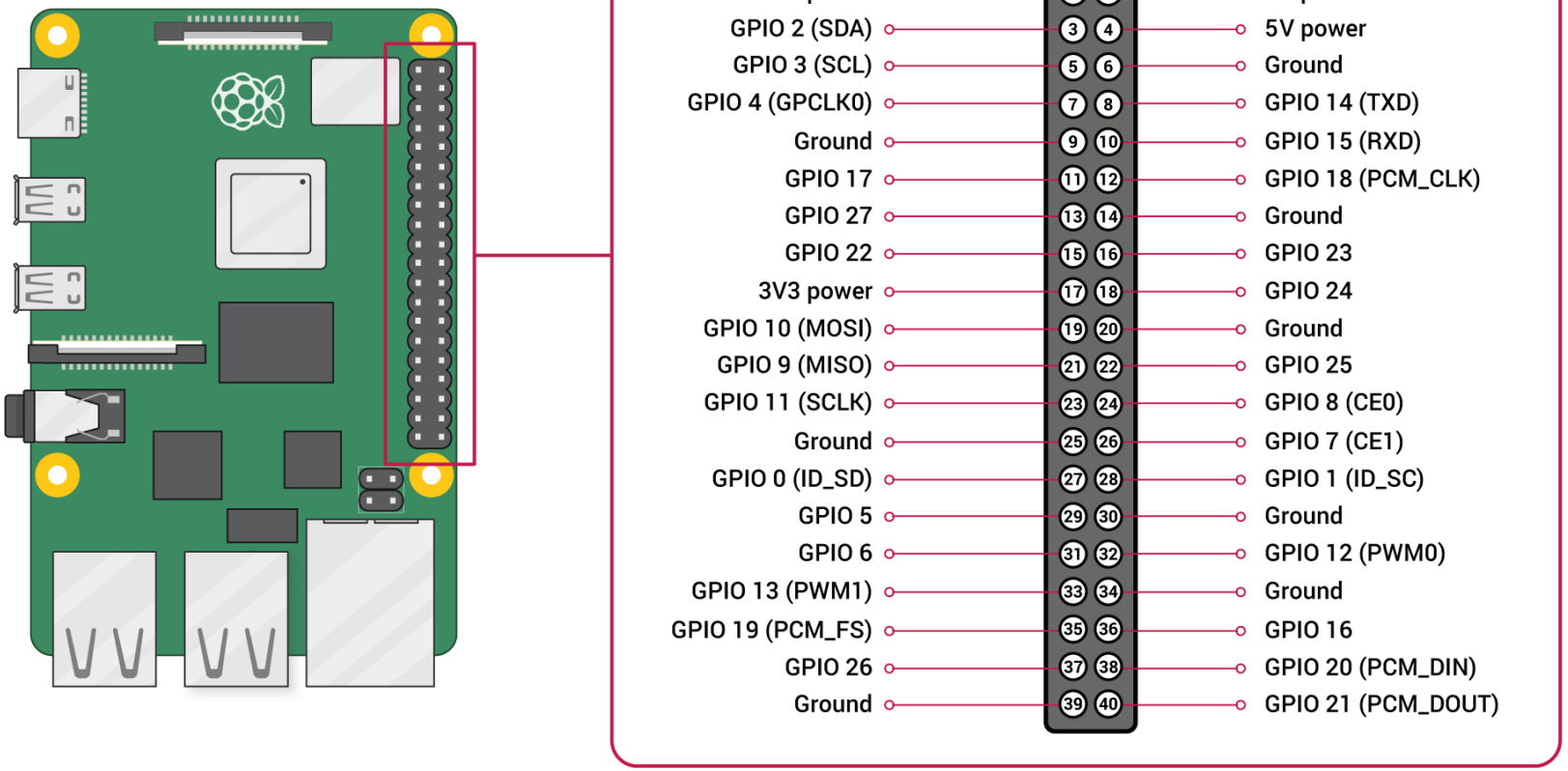
モニタ（HDMI）Camera Module V2、キーボード、マウス

Raspberry Pi 4 Model B

通常版：Model Aシリーズ（正方形）
　　Model Bシリーズ（長方形，高スペック，標準）
小型版：Zeroシリーズ
一般向け：Compute Moduleシリーズ
産業向け：第1世代～第4世代
Raspberry Pi 4は第4世代



PINOUT



https://www.raspberrypi.org/documentation/usage/gpio/

Raspberry Pi 4 Model B

発売日：2019年6月24日（2019年9月電波法認証）
メモリ（GPUと共用）：1GB($35), 2GB($45), 4GB ($55)
CPU：ARM Cortex-A72（1.5GHz）
GPU：Broadcom VideoCore VI（Dual Core 500MHz, OpenGL ES 3.0）
USBポート：2.0×2，3.0×2
映像入力：15ピンMIPIカメラインターフェース
映像出力：コンポジット RCA, micro-HDMI (up to 4kp60) x 2 2.0 , MIPI DSI
音声出力：3.5 mm ジャック, micro-HDMI, I²S
ストレージ：microSDカード
有線ネットワーク：Gigabit Ethernet
無線ネットワーク：2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 5.0, BLE
拡張コネクタ：GPIO 40 ピン
電源：3A(15W), USB Type-C, GPIO, Power over Ethernet

演習：10/2に実施したリアルタイム検出をRaspberry Pi上で実現＋MariaDBに保存＋E-mailで送信

詳細は別紙の演習資料

10/2の振り返り

リアルタイム検出

Raspberry Piのカメラを使用して、リアルタイムであるターゲットを認識させる

必要なもの：

- 対象画像（学習用、評価用）
- 背景画像（学習用、評価用）
- データセットを記述したcsvファイル

かなりの部分はこちらで用意してありますが（料理番組方式）、本当はこれらを全部自分で用意します。ぜひ1度自分でチャレンジしてください

10/2の振り返り

リアルタイム検出の手順

次の順番で行う

1. 対象物を撮影する
2. 背景画像を撮影する
3. 対象物画像を水増しして300枚用意する
4. 背景画像を分割して625枚用意する
5. 対象物画像と背景画像をcsvファイルに記述する
6. 用意したデータで学習を行う
7. 学習されたデータを用いて、リアルタイムで判定させる

対象物の撮影

対象物をカメラアプリで撮影し、128x128のサイズで切り出す
正方形で切り出すため、なるべく縦横比が極端でないものを選ぶ
顔は意外と認識されにくい（頑張って下さい）
手のひらは割と簡単。スマホケース、ペットボトルの模様、社員証など
ファイル名はtarget.jpgとする。（詳しい手順は、この後の対象画像作成手順を参照）

今度は、同じ位置で背景のみで撮影し、ファイル名をother.jpgとする。サイズ等は変更しない

10/2の振り返り

撮影

対象物が入った写真を撮影 →　対象物だけ切り抜いてtarget.jpg
次に対象物がない写真を撮影　→　other.jpg

認識させたい対象物

自分が映り込まないようにして撮影

指定サイズでの切り出し(1)

正方形での切り出しが可能なXnViewを使用する
https://forest.watch.impress.co.jp/library/software/xnview/ などからダウンロード（XnViewで検索すると窓の杜が最初にヒット）

もちろん、オフィシャルサイトからDLしてもよい。
https://www.xnview.com/

XnViewを起動し、カメラアプリで撮影した対象物のファイルを読み込む
もし画像が鏡映しになっていたら、画像 → 反転 → 左右反転

10/2の振り返り

指定サイズでの切り出し(2)

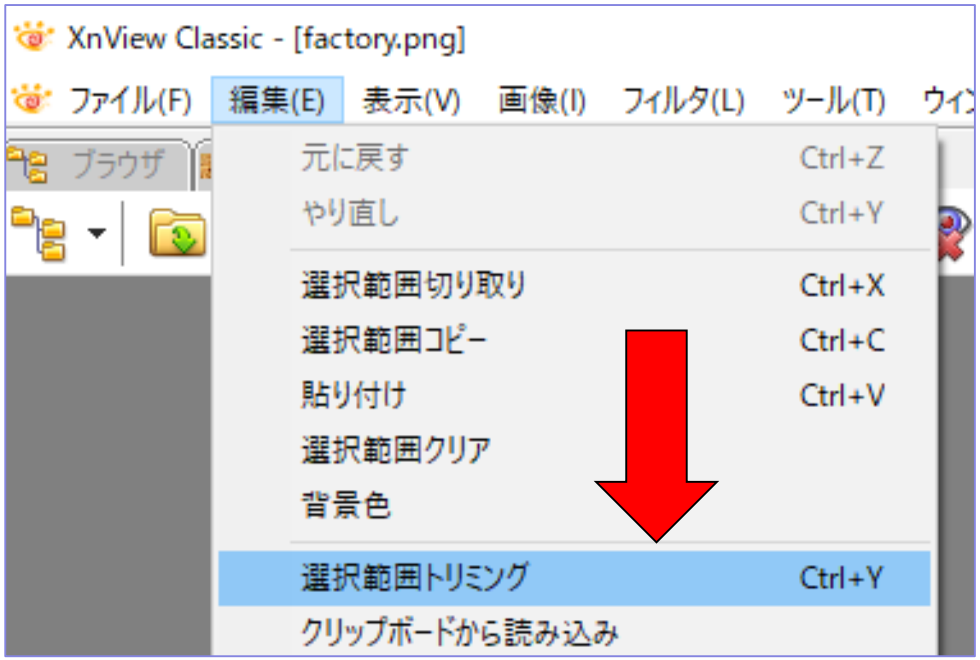
編集 → 選択範囲縦横率設定 → 1:1 (1.00) を選択
これで正方形で範囲選択が可能

10/2の振り返り

指定サイズでの切り出し(3)

任意の範囲を選択後、編集 → 選択範囲トリミング を選択し、切り抜く

切り抜いた後、画像 → リサイズ、でサイズ変更画面が開くので、縦横128ピクセルを指定してサイズ変更し、target.jpg という名前で保存



CSVファイル

こちらで準備済み

- target_or_other_train.csv　：学習（訓練用）ファイルパスとラベル
- target_or_other_test.csv　：検証用ファイルパスとラベル

それぞれのファイル内に、各画像のパスとラベルがカンマ区切りで書かれている

エディタやExcelなどで1度中味をチェックすること

水増しと学習

Colab: 08-Augmentation_Learning.ipynb を開く

VM内にtarget.jpgとother.jpg, csvファイルをコピーし、実行する

先程のcsvファイルと、フォルダtrain_target, test_targetとの対応、実際のファイルの存在をチェックしておく

学習後、重みファイル(detection_weight.h5)をダウンロードし、後は実機で

ネットワークはある程度しか書いていないので、認識できるように、拡張して下さい

重みファイルはネットワークごとに名前を変えておいて、比較するとよい

結果

緑の枠内にターゲットを持っていく
無事に認識されると赤枠になる
どうにもダメなら、ネットワークやエポック数など変えてみる

Extra演習

extraA）TensorFlowを使った物体認証の復習1
各自でリアルタイム検出を行うことができるか、新しい物体を選んで各自で実施してみましょう。

extraB）TensorFlowを使った物体認証の復習2
microSDカードにraspberry pi OSのイメージを書き込んで、各自でインストールの最初からセットアップができるか実施してみましょう。

extraC）TensorFlowを使った物体認証（VGG-16版1）
10/02に実施した「VGG-16演習」を各自でラズベリーパイ上に再現してみましょう。

extraD）TensorFlowを使った物体認証（VGG-16版2）
10/02に実施した「VGG-16演習」の機能を今回実施したプログラムに組み込んでみましょう。※リアルタイム検出の中身をVGG-16に設定する