2. 画像とカメラ

https://www.kkaneko.jp/cc/ai2/index.html

金子邦彦

BY NC SA

アウトライン

2-1 グループワーク

2-2 Rasperry Pi で OpenCV をインストール

2-3 カラー画像と配列

2-4 カメラを扱うプログラム例

2-5 Rasperry Pi の Python 3 でカメラを使う設定

2-6 Python の拡張機能

事前準備

機材

- Raspberry Pi とカメラのセット
- Windows ノートパソコン
- イーサネットケーブル (Windows ノートパソコンとカメラを接続)

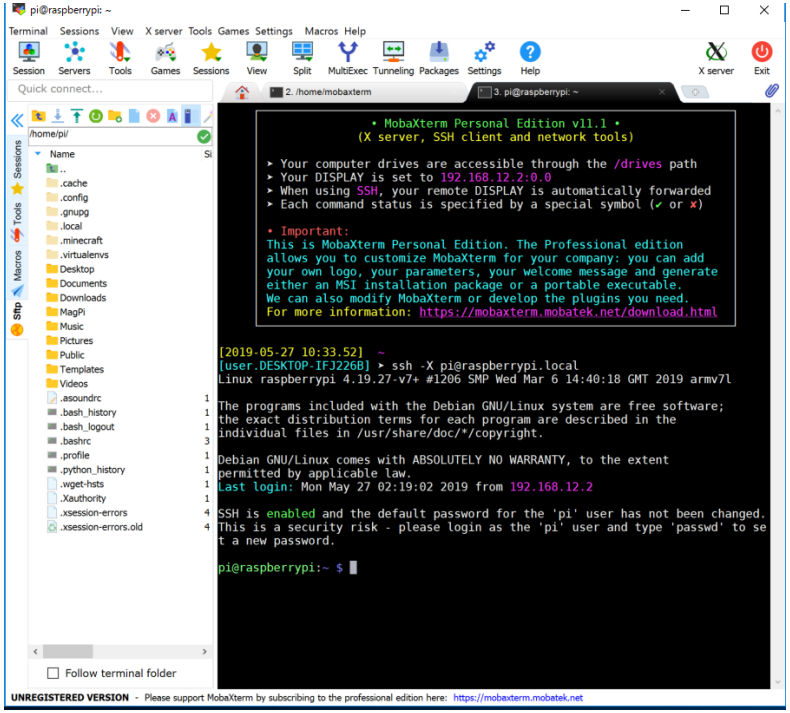
設定

- リモート接続ソフト MobaXterm Personal 版のインストール（Windows 上）

以上，https://www.kkaneko.jp/a/ai.html

事前準備のチェックポイント (1)

Windows パソコンから RaspberryPi にリモート接続

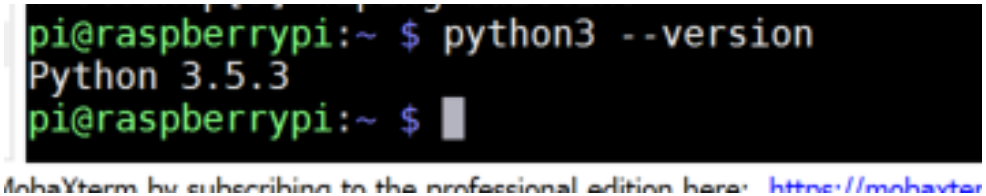


MobaXTerm で「**ssh -X pi@raspberrypi.local**」を実行し，Raspbian にリモートログインしてみる

https://www.kkaneko.jp/tools/raspbian/raspbian.html

事前準備のチェックポイント (2)

システムに **python3 がインストール済み**



次のコマンドを実行し，エラーメッセージが出ないこと

python3 --version

以上，

https://www.kkaneko.jp/tools/raspbian/pikeras.html

（動かない人は宿題とする）　（6/10 まで）

2-1 グループワーク

１．グループ分け

２．７月末までに実現したいことは何ですか

テーマ「　　」

３．その成功を，実験やデータで確かめるには，どうしたらよいですか

「　　」

４．３のための既存の技術を使います．

github https://github.com/ の右上の検索窓

使えそうな技術の候補を探してください．検索キーワードも考えること．複数可能．

キーワード「　　」

見つかったもの「　　」

2-2 Rasperry Pi で OpenCV をインストール

OpenCV とは

コンピュータビジョンと機械学習のライブラリ.

2500 以上のアルゴリズム.

顔認識、物体認識、人間の動きの分類、カメラの動きの追跡、オブジェクトの動きの追跡、３次元モデルの抽出、ステレオカメラからの３次元点群の生成、イメージスティッチング、類似画像の検索、赤目の除去、眼球運動の追跡、ARの機能など

- ライセンス: BSD ライセンス
- インタフェース: C++, Python, Java, MATLAB
- マシン: Windows, Linux, Mac OS, iOS, Android

https://opencv.org/about.html

OpenCV のインストール

```
pi@raspberrypi:~ $ sudo apt install libopencv-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  libopencv-dev
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
```

sudo apt install **libopencv-dev**

```
pi@raspberrypi:~ $ sudo apt install libjasper-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libjasper1
Suggested packages:
```

sudo apt install **libjasper-dev**

```
pi@raspberrypi:~ $ sudo apt install libqtgui4
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libjpeg8 libmng1 libqt4-dbus libqt4-xml libqtcore4 libqtdbus4 c
  qt-at-spi qtchooser qtcore4-l10n
```

sudo apt install **libqtgui4**

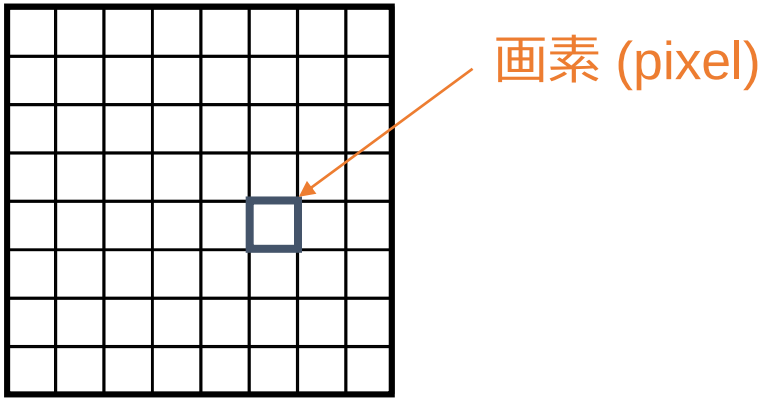
```
pi@raspberrypi:~ $ sudo apt install libqttest4-perl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libqt4-network libqt4-test libqtcore4-perl libsmokebase3v5
  libsmokeqtcore4-3 libsmokeqtgui4-3 libsmokeqtnetwork4-3 libsmok
```

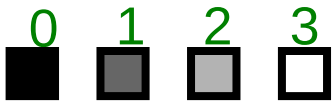
sudo apt install **libqttest4-perl**

sudo apt install <**パッケージ名**>
は **Raspbian システムのパッケージ**をダウンロード,
インストールするコマンド

2-3 カラー画像と配列

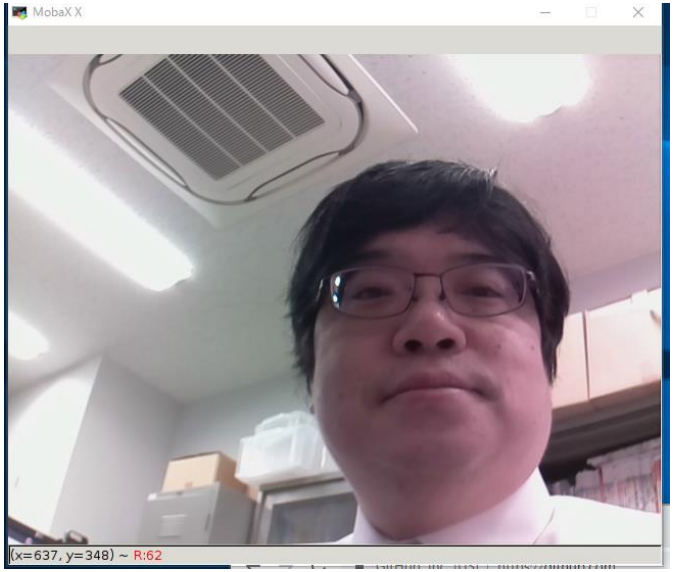
画像と画素





黒　＝　0，
暗い灰色　＝　1，
明るい灰色　＝　2，
白　＝　3
のようにコード化できる

画像の B, G, R の 3成分



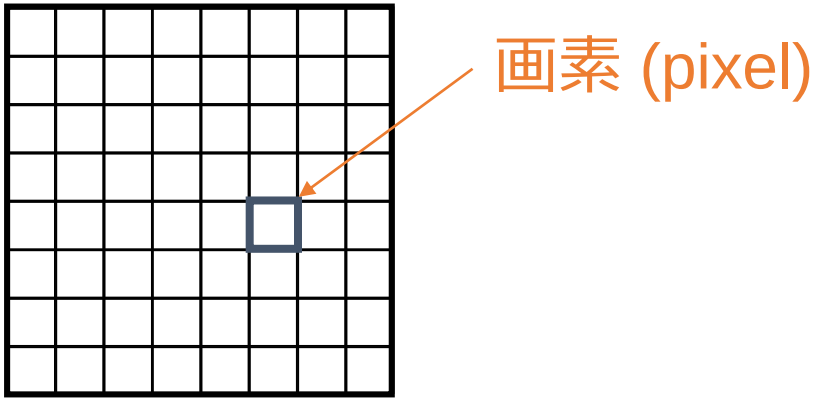
カラー画像 **f**

f[:,:,0] B成分

f[:,:,1] G成分

f[:,:,2] R成分

カラー画像画像と画素



画素が，B, G, R の **3つの値を持つ**

配列

要素の並び．要素には**添字**がある．

0	1	2	3	4
8	5	4	1	3

１次元の配列 [8 5 4 1 3] の**添字**は、

０１２３４

配列の次元

配列は Python では次のように表示される.

1 次元： [要素の並び]

2 次元：[[要素の並び] … [要素の並び]]

3 次元：[[[要素の並び] … [要素の並び]]]

```
[8  5  4  1  3]
```

1 次元

```
[[   1    2    3    4]
 [  10   20   30   40]
 [100  200  300  400]]
```

2 次元

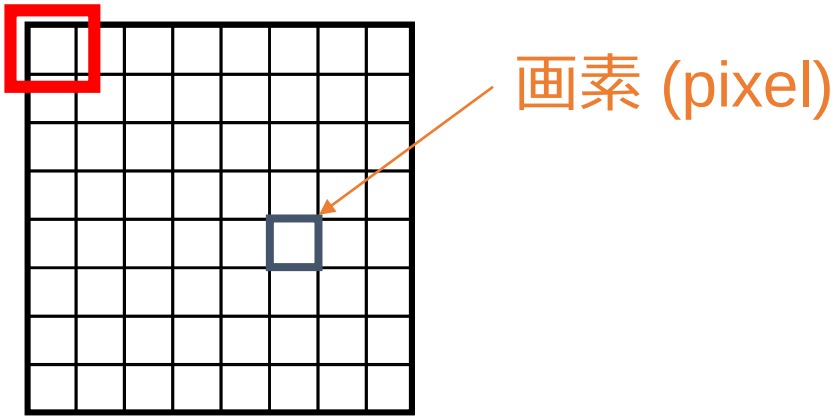
```
>>> print(f)
[[[197 191 194]
  [195 189 192]
  [194 188 191]
  ...,
  [199 192 214]
  [173 169 189]
  [236 232 252]]

 [[196 190 193]
  [194 188 191]
```

3 次元

カラー画像画像と画素

```
>>> print(f)
[[[197 191 194]
  [195 189 192]
  [194 188 191]
  ...,
  [199 192 214]
  [173 169 189]
```



画素が，B, G, R の **3つの値を持つ**

2-4 カメラを扱うプログラム例

OpenCV を用いたカメラ表示プログラム

```
import cv2
import numpy as np

v = cv2.VideoCapture(0)
while(v.isOpened()):
    r, f = v.read()
    if ( r == False ):
        break
    cv2.imshow("", f)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

v.release()
cv2.destroyAllWindows()
```

ビデオ表示ではループを行う

read : フレーム読み出し

imshow : 表示

「q」キーで，ループ終了

release: 使用終了

OpenCV を用いたカメラ表示プログラム

```
import cv2
import numpy as np

v = cv2.VideoCapture(0)
while(v.isOpened()):
    r, f = v.read()
    if ( r == False ):
        break
    cv2.imshow("", f)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

v.release()
cv2.destroyAllWindows()
```

f は３次元の配列

```
>>> print(f)
[[[197 191 194]
  [195 189 192]
  [194 188 191]
  ...,
  [199 192 214]
  [173 169 189]
```

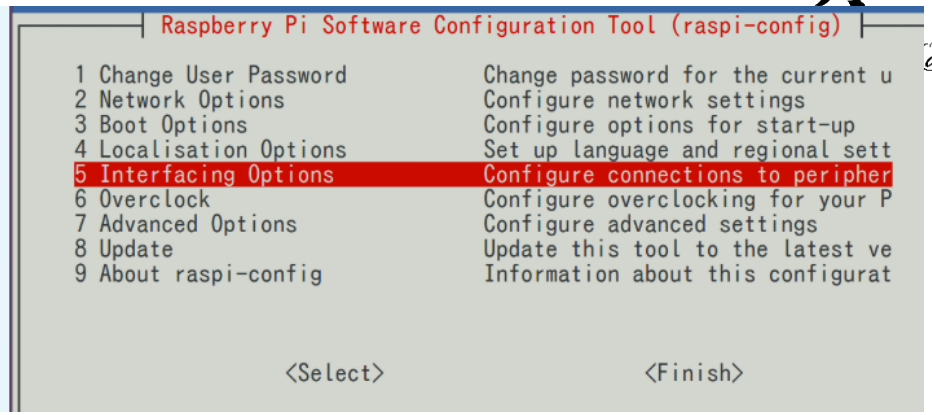
画素の B 値，G値，R値の並び

2-5 Rasperry Pi の Python 3 でカメラを使う設定

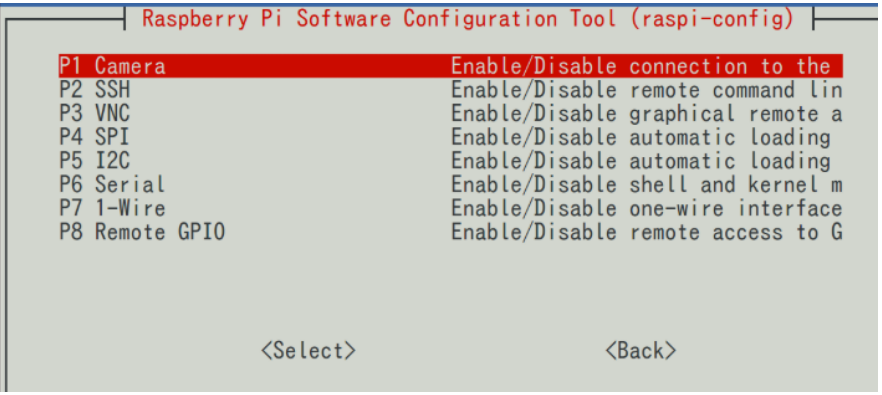
Rasperry Pi でカメラを使う設定

```
sudo raspi-config
```

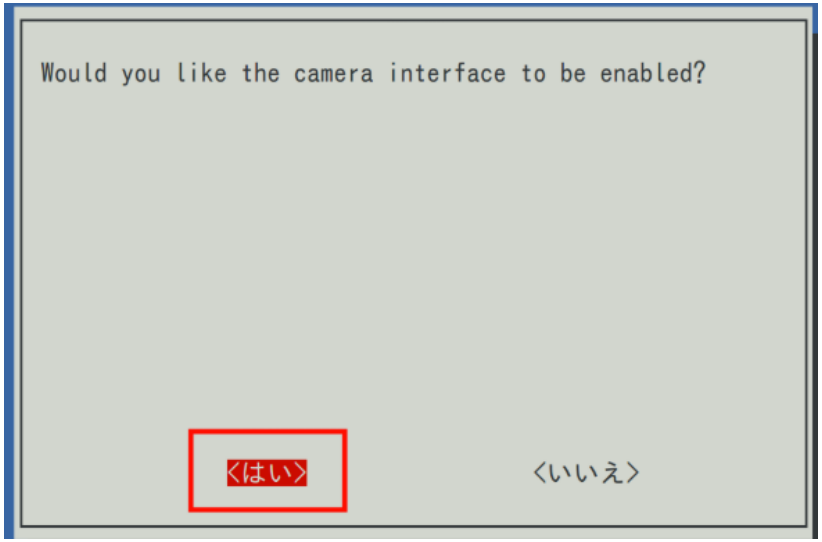
① 端末で、コマンドを実行



② 「**5. Interfacing Options**」を選ぶ

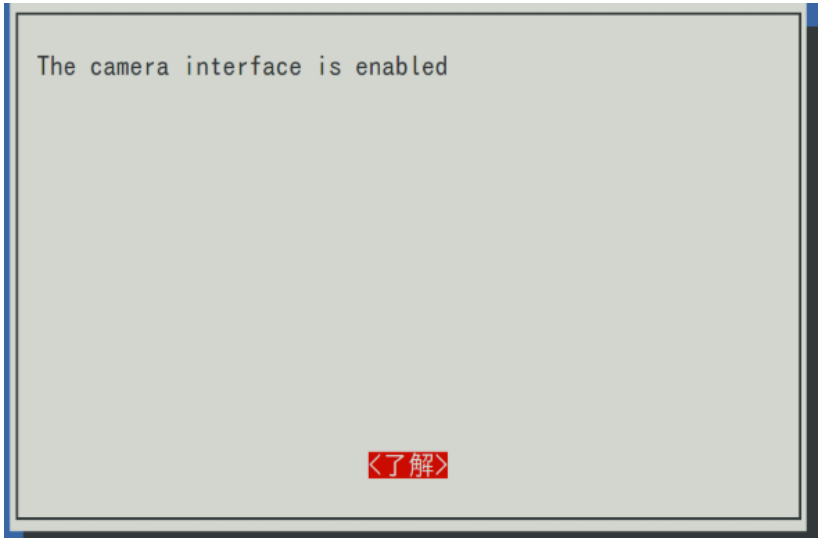


③ 「**P1. Camera**」を選ぶ

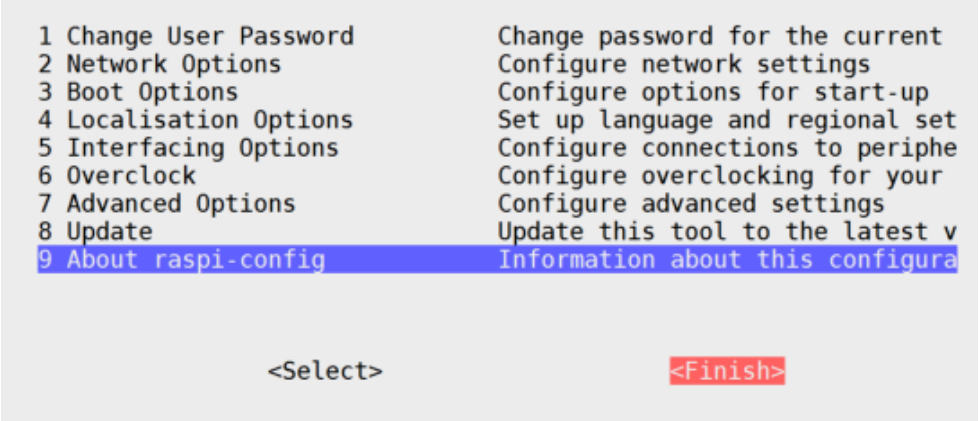


④ 「**はい**」を選ぶ

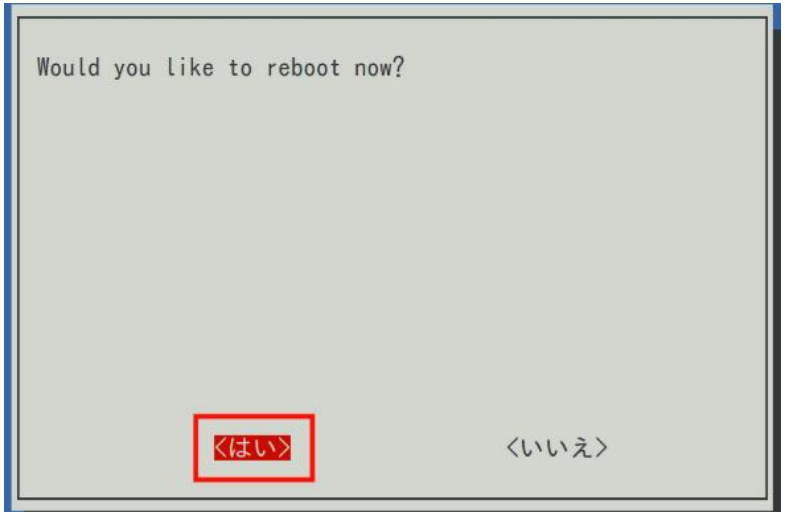
Rasperry Pi でカメラを使う設定



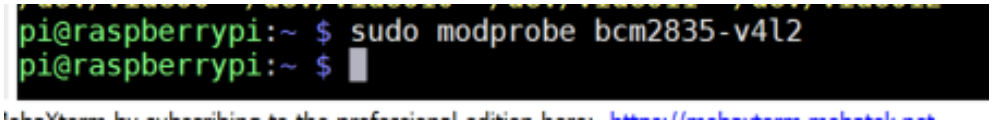
⑤ 「**了解**」を選ぶ



⑥ 元の画面で「**Finish**」を選ぶ



⑦「**はい**」を選び、システムを再起動する



⑧ sudo modprobe bcm2835-**v4l2**
を実行．「**v 4 エル 2**」です

Python パッケージのインストール

```
pi@raspberrypi:~ $ sudo pip3 install opencv-python
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: opencv-python in /usr/local/lib/python3.5/dist-p
ackages (3.4.4.19)
Requirement already satisfied: numpy>=1.12.1 in /usr/lib/python3/dist-packages
(from opencv-python) (1.12.1)
You are using pip version 19.0.3, however version 19.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
pi@raspberrypi:~ $
```

sudo pip3 install **opencv-python**

Python から OpenCV を使う機能

```
pi@raspberrypi:~ $ sudo pip3 install picamera
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting picamera
  Downloading https://www.piwheels.org/simple/picamera/picamera-1.13-py3-none-a
ny.whl (154kB)
    100% |[redacted]| 163kB 363kB/s
Installing collected packages: picamera
Successfully installed picamera-1.13
You are using pip version 19.0.3, however version 19.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
pi@raspberrypi:~ $
```

sudo pip3 install **picamera**

OpenCV なしで手軽にカメラを扱いたいときのため

sudo pip3 install <**パッケージ名**>
は **Python 3 のパッケージ（PyPI形式）** をダウンロード，
インストールするコマンド

OpenCV を用いたカメラ表示プログラム

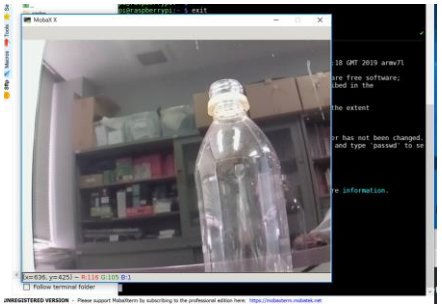
```
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
python3
```

```
>>> import cv2
>>> import numpy as np
>>>
>>> v = cv2.VideoCapture(0)
>>> while(v.isOpened()):
...     r, f = v.read()
...     if ( r == False ):
...         break
...     cv2.imshow("", f)
...     if cv2.waitKey(1) & 0xFF == ord('q'):
...         break
```

```
import cv2
import numpy as np

v = cv2.VideoCapture(0)
while(v.isOpened()):
    r, f = v.read()
    if ( r == False ):
        break
    cv2.imshow("", f)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```



「q」キーで終了

```
>>> v.release()
>>> cv2.destroyAllWindows()
>>> exit()
pi@raspberrypi:~ $
```

```
v.release()
cv2.destroyAllWindows()
exit()
```

exit() で Python を終了

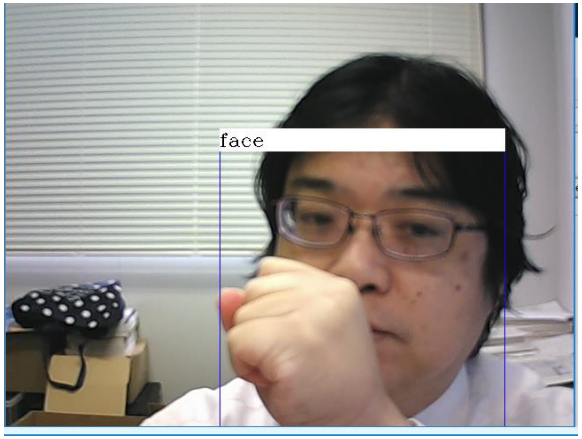
2-6 Rasperry Pi で Python の格納機能を動かす

コンピュータビジョンに関わるライブラリ類

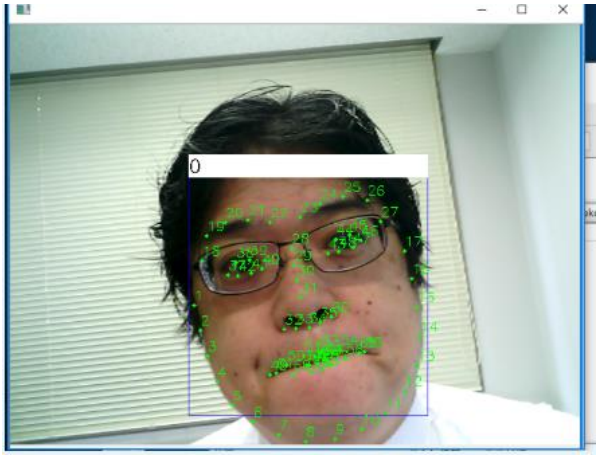
- 顔検知、顔識別：　Dlib など
- 文字認識： Tesseract, OpenALPR など
- 一般物体認識：　種々の手法

  Mask R-CNN, YOLO など

Dlib による顔検知，顔識別の例



顔検知は，画像の中から，顔である領域を抜き出すこと



顔のランドマークは，顔から，右目の端など所定のランドマークを取り出すこと

```
In [5]: import face_recognition
   ...: picture_of_obama = face_recognition.load_image_file("C:/pytools/face_recognition/known_people/obama.jpg")
   ...: obama_face_encoding = face_recognition.face_encodings(picture_of_obama)[0]
   ...:
   ...: biden_picture = face_recognition.load_image_file("C:/pytools/face_recognition/known_people/biden.jpg")
   ...: biden_face_encoding = face_recognition.face_encodings(biden_picture)[0]
   ...:
   ...: results = face_recognition.compare_faces([obama_face_encoding], biden_face_encoding)
   ...: print(results)
[False]

In [6]:
```

顔識別は，２つの顔画像が同一人物かを判定すること
（左図では Python の face_recognition パッケージを追加しての結果）

文字認識

日本国民は、正当に選挙された国会における代表者を通じて行動し、われらとわれらの子孫のために、諸国民との協和による成果と、わが国全土にわたつて自由のもたらす恵沢を確保し、政府の行為によつて再び戦争の惨禍が起ることのないやうにすることを決意し、ここに主権が国民に存することを宣言し、この憲法を確定する。そもそも国政は、国民の厳粛な信託によるものであつて、その権威は国民に由来し、その権力は国民の代表者がこれを行使し、その福利は国民がこれを享受する。これは人類普遍の原理であり、この憲法は、かかる原理に基くものである。われらは、これに反する一切の憲法、法令及び詔勅を排除する。

日本国民は、恒久の平和を念願し、人間相互の関係を支配する崇高な理想を深く自覚するのであつて、平和を愛する諸国民の公正と信義に信頼して、われらの安全と生存を保持しようと決意した。われらは、平和を維持し、専制と隷従、圧迫と偏狭を地上から永遠に除去しようと努めてゐる国際社会において、名誉ある地位を占めたいと思ふ。われらは、全世界の国民が、ひとしく恐怖と欠乏から免かれ、平和のうちに生存する権利を有することを確認する。

われらは、いづれの国家も、自国のことのみに専念して他国を無視してはならないのであつて、政治道徳の法則は、普遍的なものであり、この法則に従ふことは、自国の主権を維持し、他国と対等関係に立たうとする各国の責務であると信ずる。

日本国民は、国家の名誉にかけ、全力をあげてこの崇高な理想と目的を達成することを誓ふ。

画像ファイル

日本国民は、正当に選芋された国会における代表者を通じて行動し、わ
れらとわれらの子孫のために、諸国民との協和による成果こ、わが国人
土にわたつて目由のもたらす吊沢を確保し、政府の行為によつて再び戦
手の惨補が起ることのないやうにすることを決意し、こごここに主権が国民
に存することを宣言し、ごの憲法を確定する。そもそも国政は、国民の
茂蚤な信託によるものであつて、その権威は国民に由来し、その権力は
国民の代表者がこれを行使し、その福利は国民がこれを享受する。こごこれ
は人類普遍の原理であり、ごこの憲法は、かかる原理に碁くものである。
われらは、これに反する一切の憲法、法令及び語勅を排除する。

日本国民は、恒久の平和を念願し、人間相互の関係を支配する崇高な理
想を深く自覚するのであつて、平和を愛する諸国民の公正と信義に信頼
して、われらの安全と生存を保持しようと決意した。われらは、 平和を
維持し、専制と隷従、圧迫と偏狭を地上から永遠に除去しようと努めて

テキストデータ

Tesseract 4.1 での実験結果例.
Python のパッケージ pyocr から Tesseract をコントロールすることも可能. （Raspberry Pi では動かない. Windows で）

物体識別

Python のパッケージ pycocotools の実行結果例
（マイクロソフトのサービスである coco を用いて物体識別を行うもの）