

11-1

HTTPによるビデオ・ストリーミング配信



CPU版

プログラム名: WebStream.py
CPU版 <https://interface.cqpub.co.jp/11-1webstream-py/>

● 概要

Jetson Nanoのようなシングル・ボード・コンピュータを使えば、カメラを使ったストリーミング配信ができます。用途はほぼ無限に考えられるのではないのでしょうか。例えばロボットにカメラを接続して、離れた場所からロボット目線の映像を確認したり、監視カメラとして遠隔地の様子を確認したりできます。ウェブ・ストリーミングの映像は、クライアントPCのブラウザで閲覧できます。このように多くの可能性を感じられる機能も、PythonとOpenCVの組み合わせで簡単に実現できます。

ここでは、ストリーミング機能のみを解説しますが、実際に使用する場合は、Jetson Nano側でさまざまな解析処理を行った結果を映像配信することで、さらに便利に使いそうです。本サンプル・プログラムを実行した後は、ウェブ・ブラウザでhttp://IPアドレス:8080と入力すると、図1のようにストリーミング映像が閲覧できます。例えば、IPアドレスが192.168.1.100の場合は、http://192.168.1.100:8080と入力します。

● 前処理

この機能の実装では、前処理としてJPEG変換を行っています。

● CPU版…プログラム名: WebStream.py

▶リスト1: 004行…ウェブ・サーバを使う宣言

```
web = bottle.Bottle()
```

ウェブ・サーバを使うための宣言をします。

▶リスト1: 044行…ウェブ・サーバを動かす

```
web.run(host='0.0.0.0', port=8080)
```

004行の宣言を使って、ウェブ・サーバを動作させます。



図1 ウェブ・ブラウザで表示したJetson Nanoのビデオ・ストリーミング映像

リスト1 HTTPによるビデオ・ストリーミングを行うプログラム(WebStream.py)

```
000: import cv2
001: import bottle
002: import time
003:
004: web = bottle.Bottle()
005:
006: def __main__():
007:     cap = cv2.VideoCapture(0, cv2.CAP_V4L)
008:     cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
009:     cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)
010:
011:     if not cap.isOpened():
012:         # ビデオキャプチャー可能か判断
013:         print("Not Opened Video Camera")
014:         exit()
015:
016:     while True:
017:         ret, img = cap.read()
018:
019:         if not ret: # キャプチャー画像取得に失敗したら終了
020:             print("Video Capture Err")
021:             break
022:
023:         # ここで処理を実行する
024:         result, jpgImg = cv2.imencode('.jpg',
025:                                     img=img, params=[int(cv2.IMWRITE_JPEG_QUALITY),
026:                                                         80]) # 0 - 100
027:
028:         yield b'--frame\r\n' + b'Content-Type:
029:               image/jpeg\r\n\r\n' + bytearray(jpgImg) +
030:               b'\r\n\r\n'
031:
032:         time.sleep(1 / 60)
033:
034:     cap.release()
035:     cv2.destroyAllWindows()
036:
037:     return 0
038:
039: @web.route('/')
040: def main():
041:     return bottle.static_file('index.html',
042:                               root='./')
043:
044: @web.route('/video_recv')
045: def video_recv():
046:     bottle.response.content_type =
047:         'multipart/x-mixed-replace;boundary=frame'
048:     return __main__()
049:
050: if __name__ == '__main__':
051:     print(cv2.__version__)
052:
053:     web.run(host='0.0.0.0', port=8080,
054:            reloader=True, debug=True)
```