

2019/07/15

OpenCVプログラミングを学ぶ



どうも、クラゲです。

彼は、IoTに関する技術コンテンツを発信する
電子工作が大好きなクラゲです。

どうも、ディープなクラゲです。

「[OpenVINO™ でゼロから学ぶディープラーニング推論](#)」シリーズの5
回目記事です。

このシリーズは、ディープラーニング概要、OpenVINO™ツールキッ
ト、Neural Compute Stick、RaspberryPiの使い方、Pythonプログラミ
ングをゼロから徹底的に学び、成果としてディープラーニング推論アプ
リケーションが理解して作れるようになることを目指します。



第5回目はOpenCVについて学びます。

本シリーズでは、主にUSBカメラからの映像を入力画像として取り扱う
ためにOpenCVを使います。

【 目次 】

- OpenCVとは
- 画像ファイルの読み込みと表示
- 画像サイズ変更
- 文字や図形の描画
- カメラ映像の読み込みと表示

OpenCVとは？

CVとはComputer Visionの略です。

OpenCVはインテルのComputer Vision向けオープンソースライブラリで、具体的には、画像処理、画像解析、機械学習などが行えます。

OpenCVをPythonで使うためには `cv2` モジュールを `import` する必要があります。 `cv2` の `2` はバージョンとは関係ないのでご注意ください。OpenCV2でも3でも4でも、 `import cv2` です。

画像ファイルの読み込みと表示

まず表示したい画像を実行プログラムと同じディレクトリに用意してください。ファイル名は `cat.png` とします



こちらがコードです

```
import cv2

image_file = 'cat.png'
img = cv2.imread(image_file)
cv2.imshow('image', img)

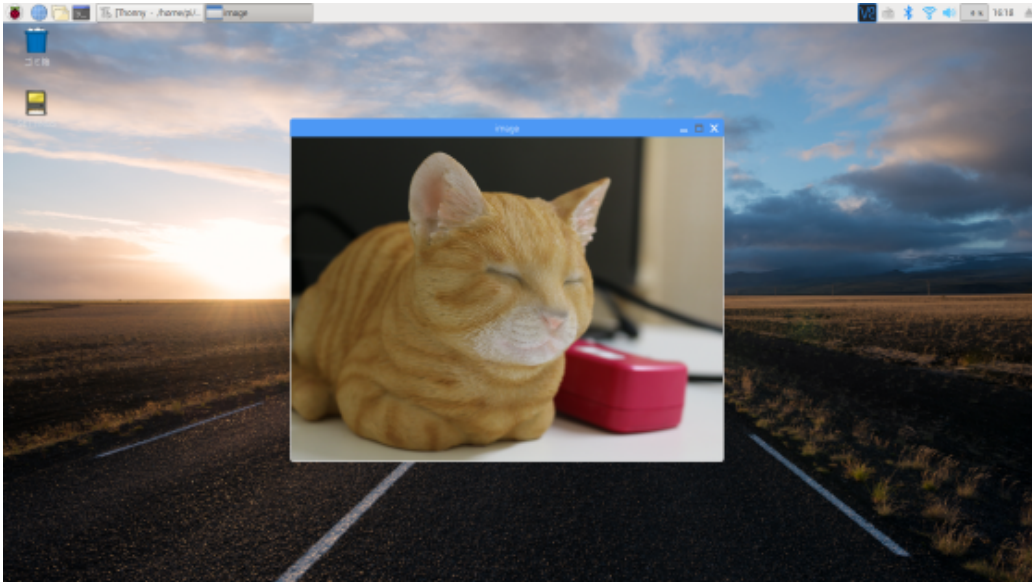
cv2.waitKey(0)
cv2.destroyAllWindows()
```

`cv2.imread` で画像ファイルを読み込みます。また、`cv2.imshow` で画像を表示します。第一引数は画像表示ウィンドウの名前で、任意の文字列を付けることができます。第二引数に`imread`で読み込んだ画像データを指定します。

`imread(画像ファイル)`

`imshow(ウィンドウ名, 画像)`

ラズパイ上では、このような感じで表示されます。画像が表示ウィンドウにフォーカスがあるとき（タイトルバーが青くなっているとき）、何かキーを押すとプログラムが終了します。



`cv2.waitKey(0)` はキーボードが押されるまで処理を待つ関数です。この関数がないと、プログラムが終了してしまうため、画像表示を見ることができません。

`cv2.waiKey` は引数の数値だけ処理を待つという関数です。数値の単位はms(ミリセカンド)です。例えば `cv2.waitKey(1000)` であれば1秒待ちま

す。ただし、引数が `0` の場合は特別で、0秒待つという意味ではなく、キーボードが押されるまでずっと処理を待つという意味になります。

最後の `cv2.destroyAllWindows` は、全てのウィンドウを閉じるという関数です。

画像サイズ変更

`cv2.resize` を使うと画像サイズを変更できます。

```
resize(画像, 画像サイズ(width, height))
```

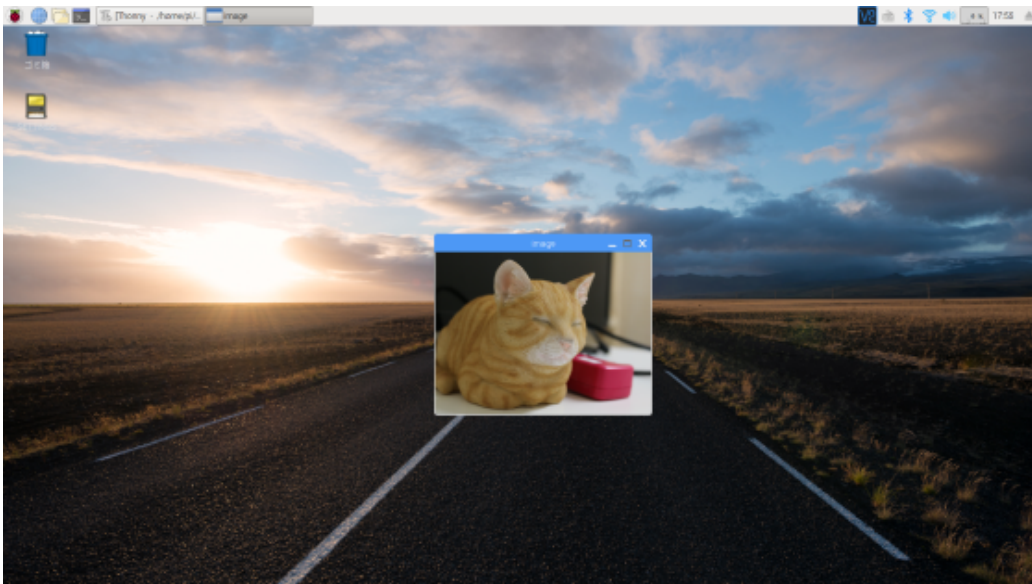
第一引数に入力画像、第二引数に タプル形式で画像サイズの幅と高さを指定します。

以下のコードは、先程のコードに1行追加しただけです。実行すると画像ウィンドウが400×300のサイズに小さくなって表示されることが確認できます。

```
import cv2

image_file = 'cat.png'
img = cv2.imread(image_file)
img = cv2.resize(img, (400, 300))
cv2.imshow('image', img)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



文字や図形の描画

OpenCVでは、画像を表示しているウィンドウに文字や図形を重ね合わせて描画することができます。

文字列描画

`cv2.putText` は文字列を描画する関数です。引数は以下のように指定します。

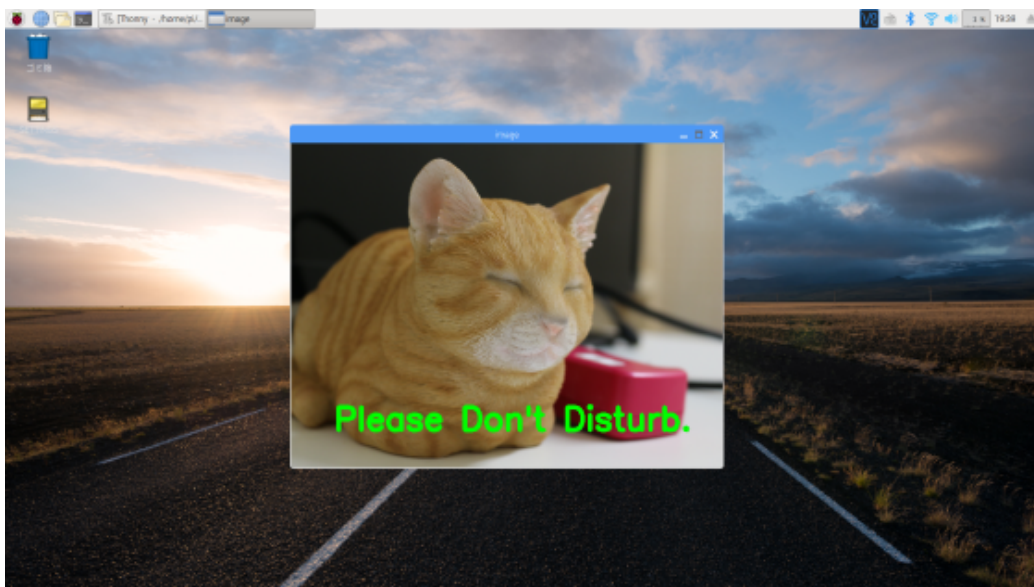
位置(X, Y)は、文字列の左下角の座標で指示します。

```
putText(画像, 文字列, 位置(X, Y), フォント, 倍率, 色(B, G, R),  
        線の太さ)
```

色の順番はOpenCVの場合、RGBではなくBGRであることに注意してください。また、倍率とはフォントの倍率のことで小数で設定可能です。一般的なフォントのサイズとは単位が異なります。

以下は、最初のサンプルコードに `cv2.putText` を 1 行追加した例です

```
import cv2  
  
image_file = 'cat.png'  
img = cv2.imread(image_file)  
cv2.putText(img, "Please Don't Disturb.", (80, 530), cv2.FONT_HERSHEY_S  
cv2.imshow('image', img)  
  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```



図形描画

`cv2.rectangle` は長方形を描画する関数です。引数は以下のように指定します。

`rectangle(画像, 左上座標(X, Y), 右下座標(X, Y), 色(B, G, R), 線の太さ)`

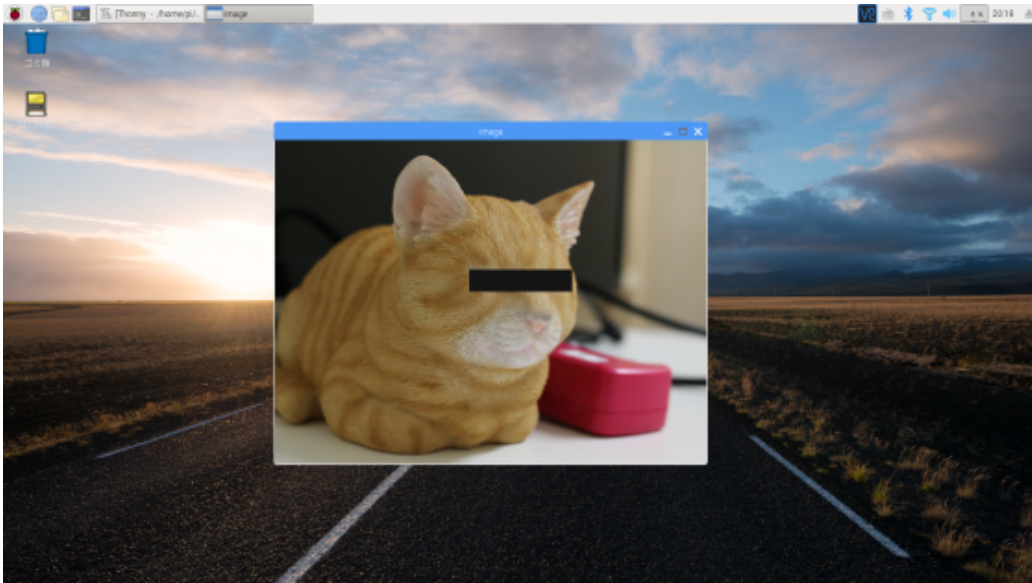
線の太さを `-1` に設定すると長方形の中身を塗りつぶします。

以下は、最初のサンプルコードに `cv2.rectangle` を 1 行追加した例です

```
import cv2

image_file = 'cat.png'
img = cv2.imread(image_file)
cv2.rectangle(img, (360, 240), (550, 280), (20, 20, 20), -1)
cv2.imshow('image', img)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



他にも `cv2.line` で直線描画、`cv2.circle` で円描画、`cv2.ellipse` で楕円描画などありますので、色々調べてみて下さい。なお、サンプルソースは静止画の例でしたが、カメラ映像にも同様に表示させることができます。

カメラ映像の読み込みと表示

今までは静止画でしたが、カメラ映像の動画を読み込んで表示します。ラズパイにUSBカメラを接続してから、以下のコードを実行してみてください。カメラ映像が動画としてリアルタイムに表示されます。

```
import cv2

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    cv2.imshow('image', frame)
    key = cv2.waitKey(1)
    if key != -1:
        break

cap.release()
cv2.destroyAllWindows()
```

OpenCVでカメラ映像表示



コードの解説をします

VideoCapture object取得

`cv2.VideoCapture` は引数にデバイス番号を指定して"VideoCapture object"を取得する関数です。

ラズパイに接続されているカメラは1つだけなので、デバイス番号は `0` を指定します。ただし、プログラム終了後にデバイス番号が変わる場合があります。その場合は、抜き差し前後で以下のコマンドを実行することによりデバイス番号を確かめることもできます。

```
ls /dev/video*
```

```
# 実行結果例  
# /dev/video0
```

While Trueによる無限ループ

OpenCVでカメラ映像を取得する手順は、動画として一括読み込み&表示ではなく、カメラ画像1フレーム毎の読み込みと表示を行う必要があります。1コマ読み込んで1コマ表示するということを繰り返すため、無限ループになっています。

画像の読み込みと表示

取得した"VideoCapture object" の `cap` を使って、`cap.read` とすることで、カメラからの静止画像を読み込みます。静止画像は2番目の戻り値 `frame` へ代入されます。1番目の戻り値 `ret` には正しく静止画像が読み込みできたか否かのブール値 (True/False)が入ります。

画像を表示させる手順は画像ファイル表示のときと同じで、`cv2.imshow` を使います。

キー入力

無限ループのままだとプログラムが終了できないため、キーボード入力があったときに `break` させループから抜け出します。

今までの静止画像表示のときのように `cv.waitKey(0)` にしてしまうと、そこで止まって動画にならないため、引数を最小値の `1` に設定しています。当然数値を大きくするとカクカクした映像になります。

キーが押されていないときの `cv.waitKey` の戻り値は `-1` です。つまり `-1` 以外のときは、何らかのキーが押されたということになりますので、それを判別して `break` しています。

VideoCapture object解放

不要になった"VideoCapture object"は、最後に `cap.release` で解放します。

以上カメラ映像表示について説明しましたが、OpenCVによるカメラ映像の読み込みと表示に関しては、ほぼこのコードのまとまりをテンプレ

ートとして使えます。

以上、「OpenCVプログラミングを学ぶ」でした。



Newsletter

ご登録いただくと、新商品やイベント情報をお知らせします。

メールアドレスを入力してください

登録

個人情報について

VISION ABOUT WORKS BLOG  

JELLYWARE

〒160-0004 東京都新宿区四谷2-3-6 パルム四谷702号室

03-6273-0758

info@jellyware.jp

Copyright 2016-2017 JellyWare Inc.

個人情報について

CONTACT

お名前

メールアドレス

メールアドレス確認

お電話番号（任意）

ご件名

