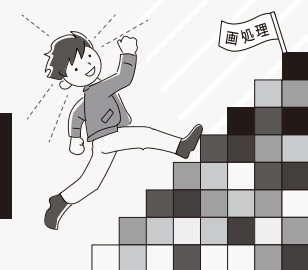


OpenCV

ワンポイント 講座

ダウンロード・データあります



第5回 OpenCV だけで作れる! ボタン付き GUI アプリ

[ご購入はこちら](#)

安川 章

OpenCV は、画像処理や画像解析の機能が搭載されたオープンソースのライブラリで、Windows、macOS、Linux 環境で動作します。本連載では OpenCV を扱う上で、知っておくと便利な小技やライブラリを紹介していきます。

ボタン付き GUI

プログラム名: cvbutton.py,
opencv_button_sample.py

Python 版の OpenCV で使用できるウィジェット (コントロール) は、トラック・バーのみです。そのため GUI アプリケーションを作るには、PyQt や Tkinter などと組み合わせる必要があります。しかし、「せめてボタンだけでも使えれば…」という思いから、OpenCV だけでボタン付き GUI を作る方法を紹介합니다。

● シンプルなボタン・クラスを作成

ボタン付きの GUI を使いまわしやすくするため、

リスト1 マウスがクリックされた位置が領域内であれば True を返す is_clicked 関数

```
def is_clicked(self, x, y):
    """座標がボタンの範囲内にあるか確認"""
    return self.x < x < self.x + self.width and self.y < y < self.y + self.height
```

リスト2 ボタンがクリックされるとコンソールにメッセージが表示されるサンプル

```
import cv2
import numpy as np
from cvbutton import CvButton # CvButton クラスをインポート

def mouse_callback(event, x, y, flags, param):
    """マウスイベントのコールバック関数"""
    global buttons

    if event == cv2.EVENT_LBUTTONDOWN:
        # マウスの左ボタンが押されたとき
        for button in buttons:
            if button.is_clicked(x, y):
                button.button_click(event, x, y, flags)
                break
    elif event == cv2.EVENT_LBUTTONUP:
        # マウスの左ボタンが離されたとき
        for button in buttons:
            button.button_release()

# ウィンドウの作成
window_name = "Button GUI"
cv2.namedWindow(window_name)
# コールバック関数の設定
cv2.setMouseCallback(window_name, mouse_callback)

# コールバック関数 (それぞれのボタンがクリックされたときの処理を定義)
def button1_click(event, x, y, flags):
    print("ボタン1がクリックされました。")
def button2_click(event, x, y, flags):
    print("ボタン2がクリックされました。")
def button3_click(event, x, y, flags):
    print("ボタン3がクリックされました。")

# ボタンの位置とサイズ、文字色、背景色、テキスト、
# コールバック関数を指定してボタンを作成
buttons = [
    CvButton(50, 30, 200, 50, (0, 0, 0),
             (180, 180, 180), "Button 1", button1_click),
    CvButton(50, 90, 200, 50, (255, 255, 255),
             (130, 130, 130), "Button 2", button2_click),
    CvButton(50, 150, 200, 50, (255, 255, 255),
             (80, 80, 80), "Button 3", button3_click)
]

while True:
    # 画像を作成
    image = np.zeros((250, 300, 3), np.uint8)
    for button in buttons:
        # 画像 (image) にボタンを描画
        button.draw(image)

    cv2.imshow(window_name, image)

    # キー入力で終了
    if cv2.waitKey(1) > 0:
        break

cv2.destroyAllWindows()
```