

4-1 準備…拡大/縮小

プログラム名: Resize_default.py (GPU版はResize_default_gpu.py)
 CPU版 https://interface.cqpub.co.jp/4-1Resize_default.py,
 GPU版/4-1Resize_default_gpu-py



CPU版



GPU版



図1 リサイズ前



図2 リサイズ後

● 使用頻度の多いリサイズ機能

カメラ映像や撮影動画を再生した場合は、そのままではサイズが大きくなり、1フレームごとの負荷が高すぎてリアルタイム処理できないことがあります。縮小映像でも十分な解析精度を得られる場合は、処理速度を上げるために縮小して解析する方法があります。

また、AIによる画像処理などの学習データを扱う場合も、同じサイズにそろえる必要があります。このようにリサイズは使用場面の多い機能となります。

● CPU版プログラム…

Resize_default.py

▶ リスト1: 036行…リサイズ

リサイズは図1から図2へ拡大したり、逆に図2から図1のように縮小したりする変換のことです。

リスト1は実行全体を記載しているため、分かりにくいと思いますが、実際にリサイズ変換しているのは036行だけです。

```
dst = cv2.resize(src, (int(width *
resizeRate), int(height *
resizeRate)), interpolation=None)
```

第1引数: 画像

第2引数: 縮小サイズ [横幅 (Pixel), 縦幅 (Pixel)]

第3引数: 補間アルゴリズム

なお、補間アルゴリズムは4-2項以降で紹介します。第3引数は初期値のまま使っており、Noneにしていますが、省略しても問題なく動作します。

● GPU版プログラム…

Resize_default_gpu.py

▶ Resize_default_gpu.py: 042行…リサイズ

```
gpuDst = cv2.cuda.resize(gpuSrc,
(int(width * resizeRate),
int(height * resizeRate)))
```

第1引数: 画像

第2引数: 縮小サイズ [横幅 (Pixel), 縦幅 (Pixel)]

第3引数: 省略

使用方法はCPU版と同じですが、CUDA処理の前処理/後処理も確認してください。

リスト1 リサイズ変換のプログラムCPU版(Resize_default.py)

```
000: import cv2
001:
002: def __main__():
003:     cap = cv2.VideoCapture(0, cv2.CAP_V4L)
004:     cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
005:     cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)
006:
007:     if not cap.isOpened():
008:         # ビデオ・キャプチャ可能か判断
009:         print("Not Opened Video Camera")
010:         exit()
011:
012:     while True:
013:         ret, img = cap.read()
014:         if ret == False:
015:             # キャプチャ画像取得に失敗したら終了
016:             print("Video Capture Err")
017:             break
018:
019:         # ここで処理を実行する
020:         img = getResize(img)
021:
022:         # 画面表示
023:         if cv2.waitKey(10) > -1:
024:             break
025:
026:
027: def getResize(src):
028:     """CPUを使用"""
029:     basePixSize = 720
030:     # 縦横で大きい辺の変更したいサイズ
031:     height = src.shape[0]
032:     width = src.shape[1]
033:
034:     largeSize = max(height, width)
035:     # 大きい方の辺のサイズ
036:     resizeRate = basePixSize / largeSize
037:     # 変更比率を計算
038:
039:     dst = cv2.resize(src, (int(width
040: * resizeRate), int(height
041: * resizeRate)), interpolation=None)
042:
043:     return dst
044:
045: if __name__ == '__main__':
046:     print(cv2.__version__)
047:
048: __main__()
```