画像処理レポート 2

235738B 越後 玲輝

2025 年 5 月 24 日

目次

1　基本課題

1.1　1) 画像の表示

Listing 1　課題 1-1: 画像を表示

```
1 import cv2
2
3 IMG_PATH = "./MyAvatar.png"
4
5 img = cv2.imread(IMG_PATH)
6 cv2.imshow("1) Display Image A", img)
7 cv2.waitKey(0)
8 cv2.destroyAllWindows()
```

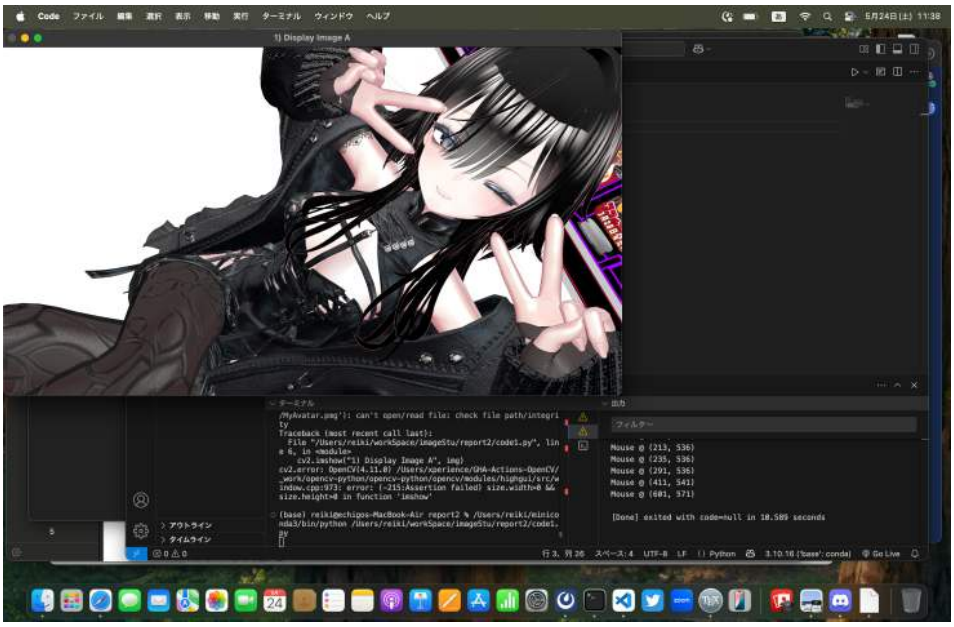

図 1　課題 1-1 実行結果

■実行結果

1.2　2) マウスカーソル位置の出力

Listing 2　課題 1-2: マウスカーソルの位置を出力

```python
import cv2
import numpy as np

def on_mouse_move(event, x, y, flags, param):マウス移動時に座標をコンソール出力
    """"""
    if event == cv2.EVENT_MOUSEMOVE:
        print(f"x={x}, y={y}")

def main():
    # 画面全体に広げるダミー画像
    img = np.zeros((512, 512, 3), dtype=np.uint8)
    winname = "screen"

    # ウィンドウを作成 & フルスクリーン化
    cv2.namedWindow(winname, cv2.WINDOW_NORMAL)
    cv2.setWindowProperty(winname, cv2.WND_PROP_FULLSCREEN, cv2.WINDOW_FULLSCREEN)

    # マウスコールバック登録
    cv2.setMouseCallback(winname, on_mouse_move)

    # 最初に一度だけ表示
    cv2.imshow(winname, img)

    # Esc が押されるまで待機ループ
    while True:
        if cv2.waitKey(20) & 0xFF == 27:  # 27==Esc
            break

    cv2.destroyAllWindows()

if __name__ == "__main__":
    main()
```

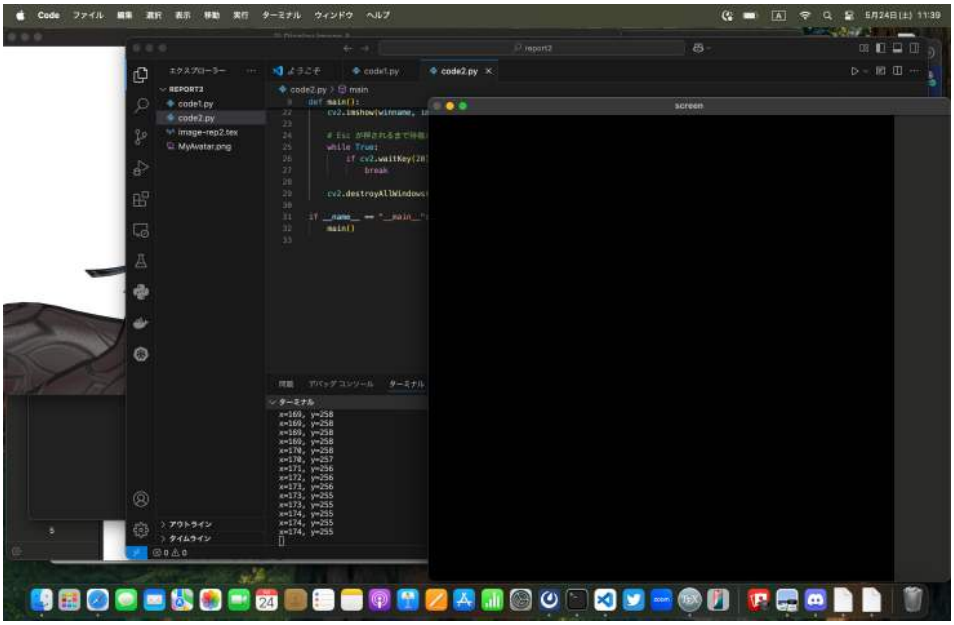

図 2　課題 1-2 実行結果

■実行結果

1.3　3) 画素 P の位置を出力

Listing 3　課題 1-3: 指定画素の位置を出力

```
1  import cv2
2
3  # クリック時のコールバック関数
4  def on_mouse_click(event, x, y, flags, param):
5      if event == cv2.EVENT_LBUTTONDOWN:
6          print(f"Clicked position → x={x}, y={y}")
7
8  def main():
9      IMG_PATH = "MyAvatar.png"
10     # 画像読み込み
11     img = cv2.imread(IMG_PATH)
12     if img is None:
13         print(f"[Error] Failed to load image: {IMG_PATH}")
14         return
15
```

```
16      # ウィンドウ作成
17      winname = 課題"3 - Click Pixel Position"
18      cv2.namedWindow(winname)
19      # コールバック登録
20      cv2.setMouseCallback(winname, on_mouse_click)
21
22      # メインループ
23      while True:
24          cv2.imshow(winname, img)
25          # キーで終了Esc
26          if cv2.waitKey(20) & 0xFF == 27:
27              break
28
29      cv2.destroyAllWindows()
30
31  if __name__ == "__main__":
32      main()
```

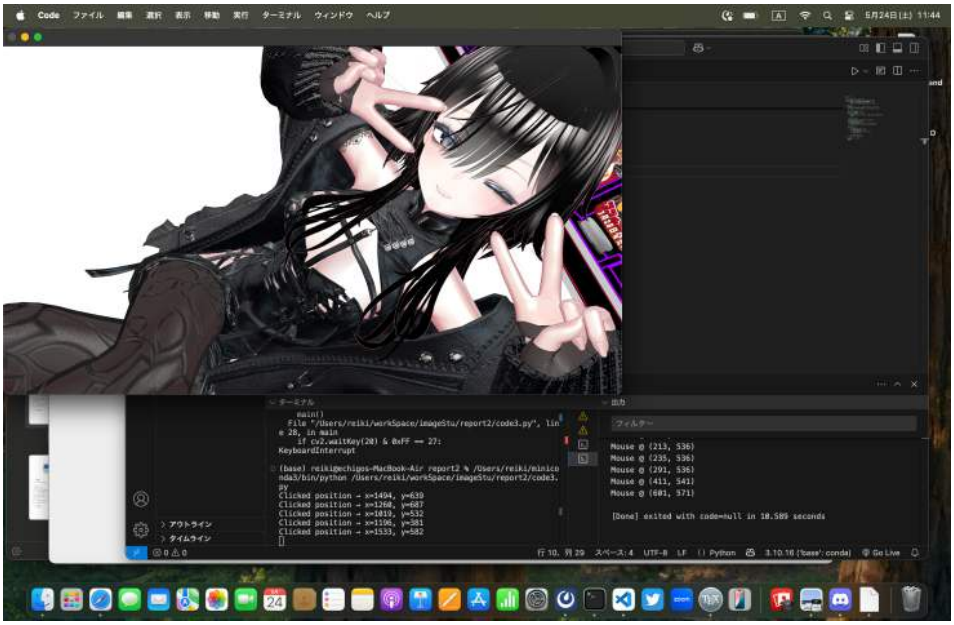

図 3　課題 1-3 実行結果

■実行結果

1.4 4) 画素 P の画素値を出力

Listing 4 課題 1-4: 指定画素の画素値を出力

```
import cv2

# 画像を先に読み込んでグローバル変数にしておく
IMG_PATH = "MyAvatar.png"
img = cv2.imread(IMG_PATH)
if img is None:
    print(f"[Error] 画像が読み込めませんでした：{IMG_PATH}")
    exit(1)

# マウスコールバック関数
def on_mouse_click(event, x, y, flags, param):
    # 左ボタンシングルクリックで座標と画素値(BGR)を表示
    if event == cv2.EVENT_LBUTTONDOWN:
        b, g, r = img[y, x]
        print(f"Clicked at ({x}, {y}) → B={b}, G={g}, R={r}")

# ウィンドウ作成とコールバック登録
winname = 課題"4 - Pixel Value"
cv2.namedWindow(winname)
cv2.setMouseCallback(winname, on_mouse_click)

# メインループ
while True:
    cv2.imshow(winname, img)
    # キーで終了Esc
    if cv2.waitKey(20) & 0xFF == 27:
        break

cv2.destroyAllWindows()
```

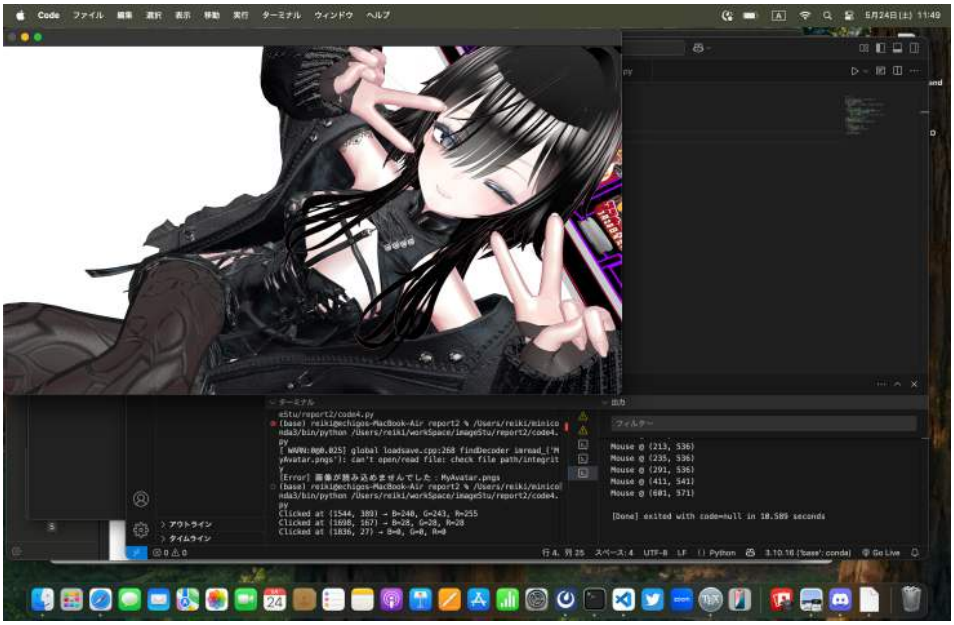

図 4　課題 1-4 実行結果

■実行結果

1.5　5) 画像サイズの出力

Listing 5　課題 1-5: 画像サイズを出力

```
import cv2
import sys

def main():
    IMG_PATH = "MyAvatar.png"

    # 画像読み込み
    img = cv2.imread(IMG_PATH)
    if img is None:
        print(f"[Error] 画像が読み込めませんでした: {IMG_PATH}")
        sys.exit(1)

    # 形状を取得 (height, width, ) channels
    height, width = img.shape[:2]
    print(f"Image size → width: {width}, height: {height}")
```

```
16
17      # 確認用にウィンドウ表示（キーで閉じる）Esc
18      win = 課題"5 - Image Size"
19      cv2.namedWindow(win)
20      cv2.imshow(win, img)
21      while True:
22          if cv2.waitKey(20) & 0xFF == 27:
23              break
24      cv2.destroyAllWindows()
25
26  if __name__ == "__main__":
27      main()
```

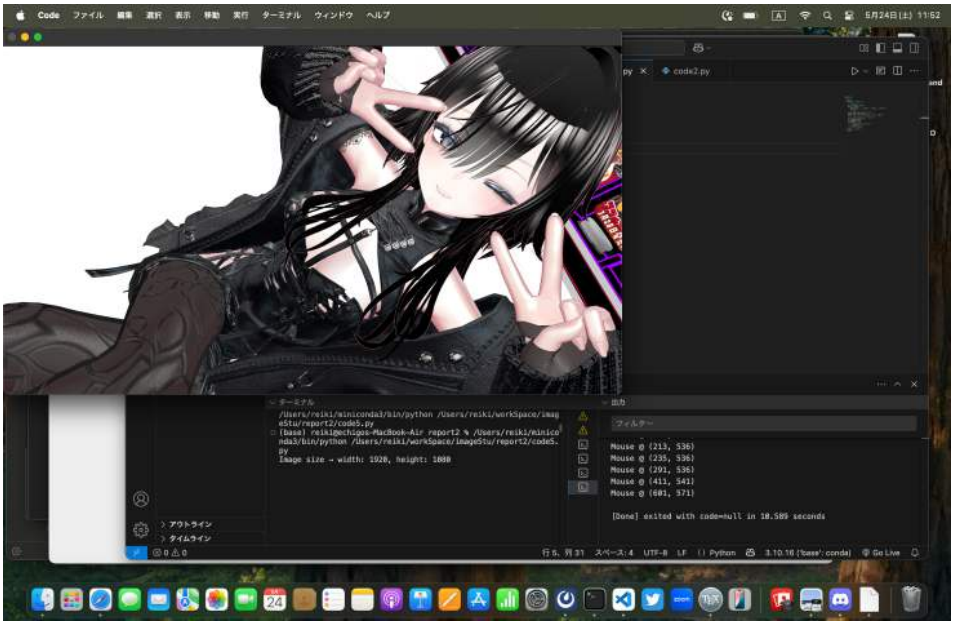

図 5　課題 1-5 実行結果

■実行結果

1.6　6) 矩形領域の指定と描画

Listing 6　課題 1-6: 矩形領域を指定して描画

```
1  import cv2
2  import numpy as np
```

```
3
4  # 状態保持用のグローバル変数
5  drawing = False    # マウス押下中フラグ
6  ix, iy = -1, -1    # ドラッグ開始位置
7
8  # コールバック関数
9  def draw_rectangle(event, x, y, flags, param):
10     global drawing, ix, iy, img
11     if event == cv2.EVENT_LBUTTONDOWN:
12         drawing = True
13         ix, iy = x, y
14
15     elif event == cv2.EVENT_LBUTTONUP:
16         drawing = False
17         # マウスを離した位置(x,y)まで矩形を描画
18         cv2.rectangle(img, (ix, iy), (x, y), (0, 255, 255), 2)
19
20 # 画像読み込み（JPEG/）BMP
21 img = cv2.imread("MyAvatar.png")
22 if img is None:
23     raise SystemExit画像が読み込めませんでした("")
24
25 # ウィンドウ作成＆コールバック登録
26 cv2.namedWindow課題("6 - 矩形指定")
27 cv2.setMouseCallback課題("6 - 矩形指定", draw_rectangle)
28
29 # メインループ
30 while True:
31     cv2.imshow課題("6 - 矩形指定", img)
32     # Esc キーで終了
33     if cv2.waitKey(20) & 0xFF == 27:
34         break
35
36 cv2.destroyAllWindows()
```

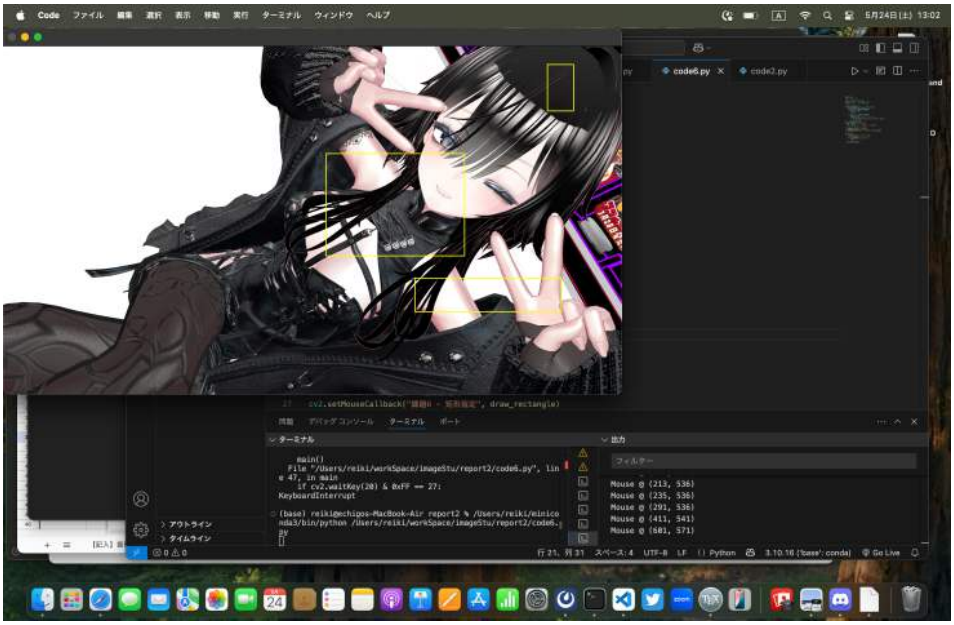

図 6　課題 1-6 実行結果

■実行結果

2　発展課題

2.1　A7) 12 枚の画像を 6 秒ずつ順番に表示

Listing 7　課題 A7: 6 秒ずつ表示

```
1  import cv2
2  import time
3  import glob
4
5  def main():
6      paths = sorted(glob.glob("images/*.jpg"))[:12]
7      if len(paths) < 12:
8          print("Error: images フォルダに枚の12 JPEG 画像を用意してください
               ")
9          return
10
11     for p in paths:
12         img = cv2.imread(p)
13         if img is None:
```

```
14                print(f"Failed to load: {p}")
15                continue
16            cv2.imshow("A7) Slideshow", img)
17            # を分割しないとキー入力が効かないので先に一瞬呼び出すwaitKey
18            cv2.waitKey(1)
19            time.sleep(6)
20        cv2.destroyAllWindows()
21
22    if __name__ == "__main__":
23        main()
```

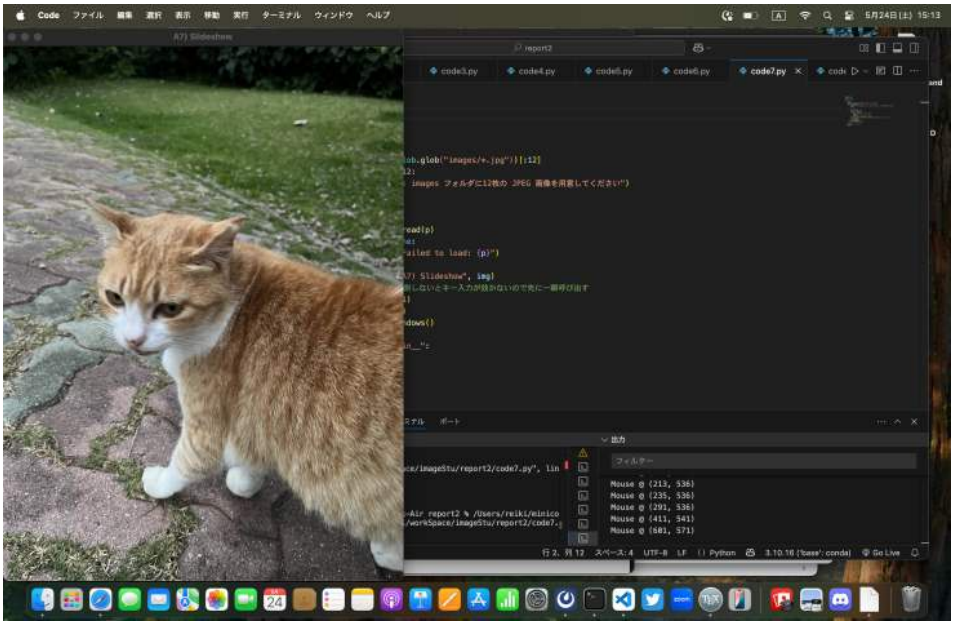

図 7　課題 A7 実行結果 1

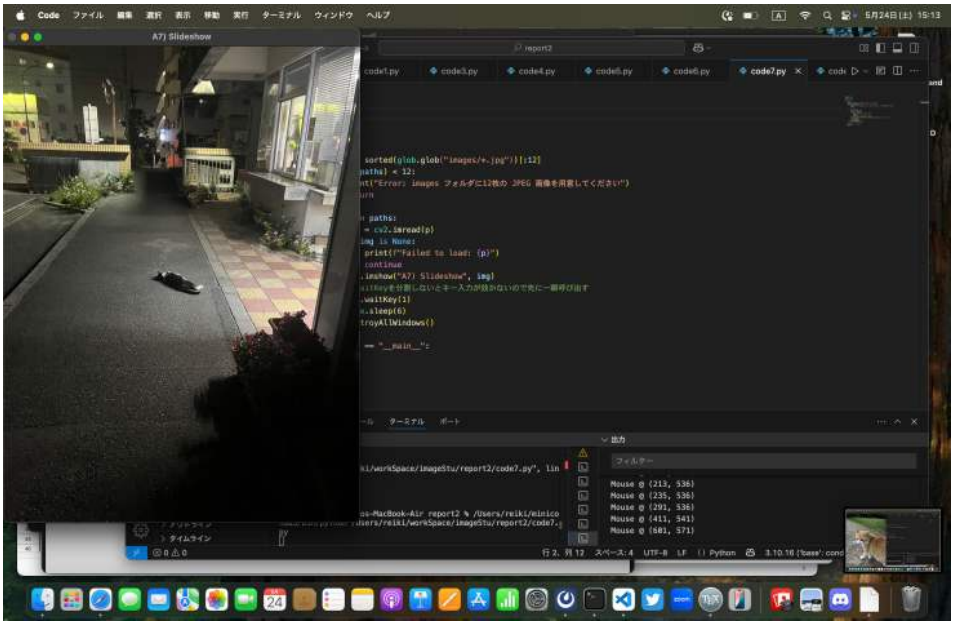

図 8　課題 A7 実行結果 2

2.2　A8) 12 枚の画像を 4 × 3 に並べて同時表示

Listing 8　課題 A8: 12 枚表示

```
import cv2
import numpy as np
import glob

def main():
    paths = sorted(glob.glob("images/*.jpg"))[:12]
    imgs = []
    for p in paths:
        img = cv2.imread(p)
        if img is None:
            print(f"Failed to load: {p}")
            return
        imgs.append(img)

    # 全画像を同じサイズにリサイズ
    h, w = imgs[0].shape[:2]
    imgs = [cv2.resize(im, (w, h)) for im in imgs]
```

```
18
19      # 枚ずつ横に連結して4 3 行作り、さらに縦に連結
20      rows = [np.hstack(imgs[i*4:(i+1)*4]) for i in range(3)]
21      grid = np.vstack(rows)
22
23      cv2.imshow("A8) 4x3 Grid", grid)
24      cv2.waitKey(0)
25      cv2.destroyAllWindows()
26
27  if __name__ == "__main__":
28      main()
```

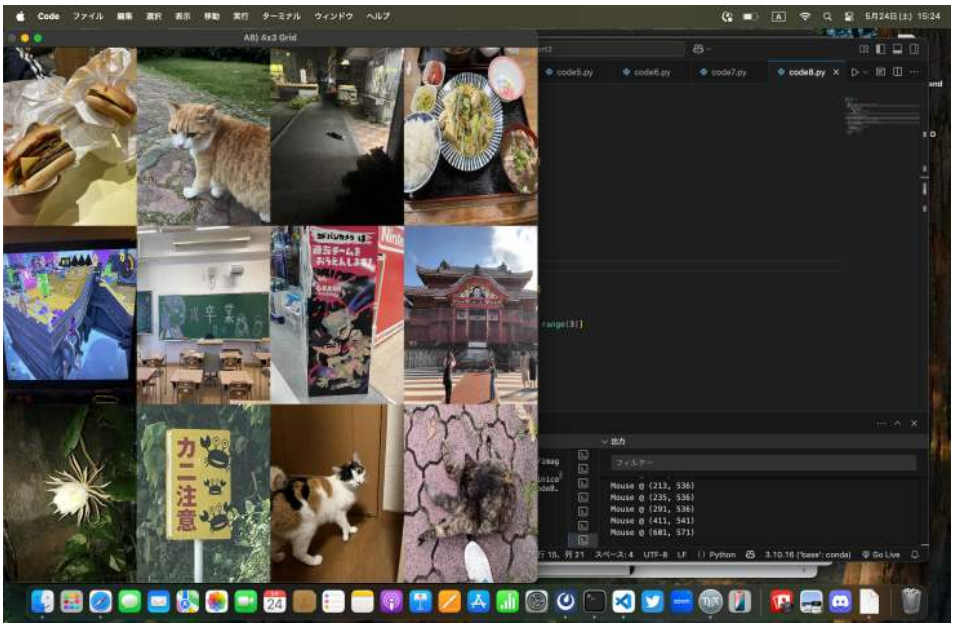

図 9　課題 A8 実行結果

2.3　A9) 任意の 1 つをクリックで消去

Listing 9　課題 A9: 12 枚表示、クリックで削除

```
1  import cv2
2  import numpy as np
3  import glob読み込み
4
5  #
```

```
paths = sorted(glob.glob("images/*.jpg"))[:12]
imgs = []


RESIZE_TO = (300, 300)

for p in paths:
    img = cv2.imread(p)
    if img is None:
        print(f"Failed to load: {p}")
        continue
    img = cv2.resize(img, RESIZE_TO)
    imgs.append(img)

positions = [
    (j * RESIZE_TO[0], i * RESIZE_TO[1], (j+1) * RESIZE_TO[0], (
       i+1) * RESIZE_TO[1])
    for i in range(3) for j in range(4)
]

def update_grid():
    rows = [np.hstack(imgs[i*4:(i+1)*4]) for i in range(3)]
    grid = np.vstack(rows)
    cv2.imshow("A9) Remove One", grid)

def on_click(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        for idx, (x0, y0, x1, y1) in enumerate(positions):
            if x0 <= x < x1 and y0 <= y < y1:
                # 選択した画像を白で塗り潰す
                imgs[idx][:] = 255
                update_grid()
                break

def main():
    if len(imgs) != 12:
        print画像が枚揃っていません("12")
```

```
42            return
43
44        cv2.namedWindow("A9) Remove One")
45        cv2.setMouseCallback("A9) Remove One", on_click)
46        update_grid()
47        cv2.waitKey(0)
48        cv2.destroyAllWindows()
49
50    if __name__ == "__main__":
51        main()
```

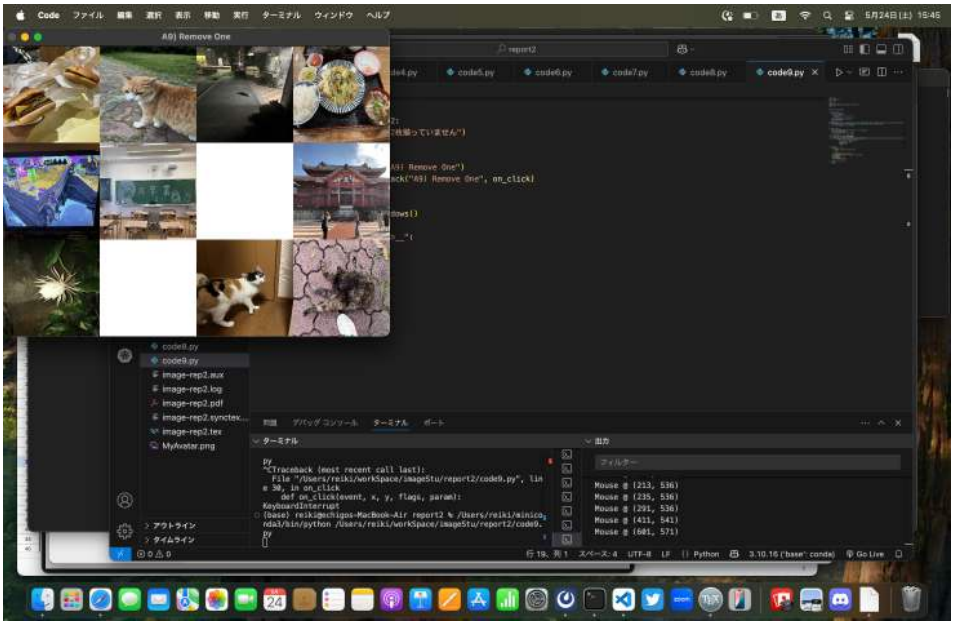

図 10　課題 A9 実行結果

2.4　A10) 任意の 2 つをクリックで入れ替え

Listing 10　課題 A10: 12 枚表示、クリックで入れ替え

```
1  import cv2
2  import numpy as np
3  import glob
4
5  # 定数
6  RESIZE_TO = (300, 300)
```

```
7
8  # 画像読み込みとリサイズ
9  paths = sorted(glob.glob("images/*.jpg"))[:12]
10 imgs = []
11
12 for p in paths:
13     img = cv2.imread(p)
14     if img is None:
15         print(f"Failed to load: {p}")
16         continue
17     img = cv2.resize(img, RESIZE_TO)
18     imgs.append(img)
19
20 # 配置情報 (x0, y0, x1, y1)
21 positions = [
22     (j * RESIZE_TO[0], i * RESIZE_TO[1], (j + 1) * RESIZE_TO[0],
           (i + 1) * RESIZE_TO[1])
23     for i in range(3) for j in range(4)
24 ]
25
26 clicks = []  # 選択された画像のインデックス（最大）2
27
28 # 描画関数（赤枠表示付き）
29 def update_grid():
30     # グリッド画像を一時的に結合
31     rows = [np.hstack(imgs[i*4:(i+1)*4]) for i in range(3)]
32     grid = np.vstack(rows)
33
34     # 赤枠を描画（選択中の画像）
35     for idx in clicks:
36         x0, y0, x1, y1 = positions[idx]
37         cv2.rectangle(grid, (x0, y0), (x1, y1), (0, 0, 255), 5)
38
39     cv2.imshow("A10) Swap Two (with highlight)", grid)
40
41 def on_click(event, x, y, flags, param):
42     global clicks
```

```
43      if event == cv2.EVENT_LBUTTONDOWN:
44          for idx, (x0, y0, x1, y1) in enumerate(positions):
45              if x0 <= x < x1 and y0 <= y < y1:
46                  if idx not in clicks:
47                      clicks.append(idx)
48                      print(f"Selected: {idx}")
49
50                  if len(clicks) == 2:
51                      i, j = clicks
52                      imgs[i], imgs[j] = imgs[j], imgs[i]
53                      clicks = []  # 入れ替え後、選択をクリア
54                  update_grid()
55                  break
56
57  def main():
58      if len(imgs) != 12:
59          print画像が枚揃っていません("12")
60          return
61
62      cv2.namedWindow("A10) Swap Two (with highlight)")
63      cv2.setMouseCallback("A10) Swap Two (with highlight)",
            on_click)
64      update_grid()
65      cv2.waitKey(0)
66      cv2.destroyAllWindows()
67
68  if __name__ == "__main__":
69      main()
```

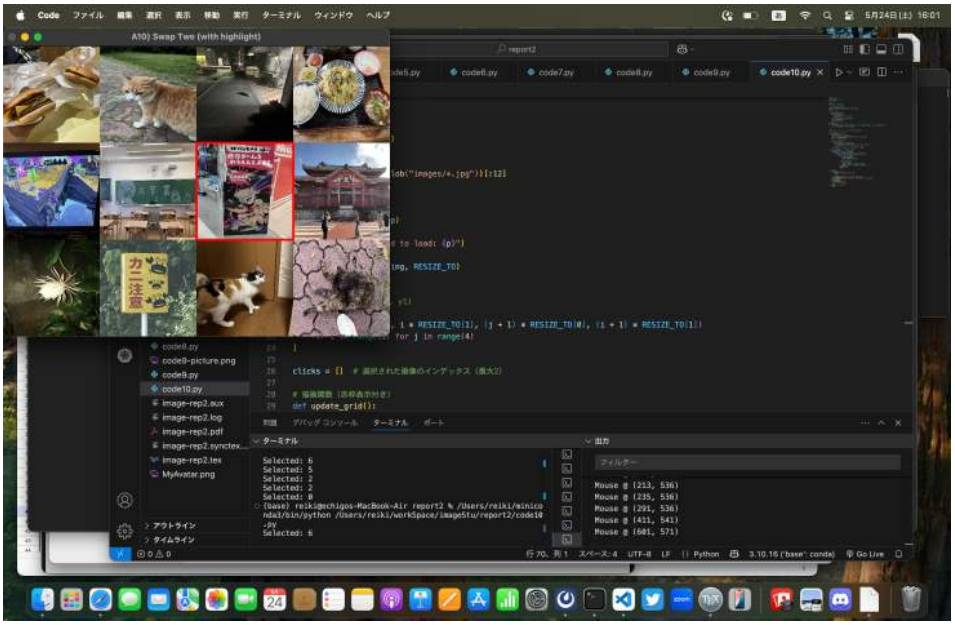

図 11　課題 A10 実行結果 1

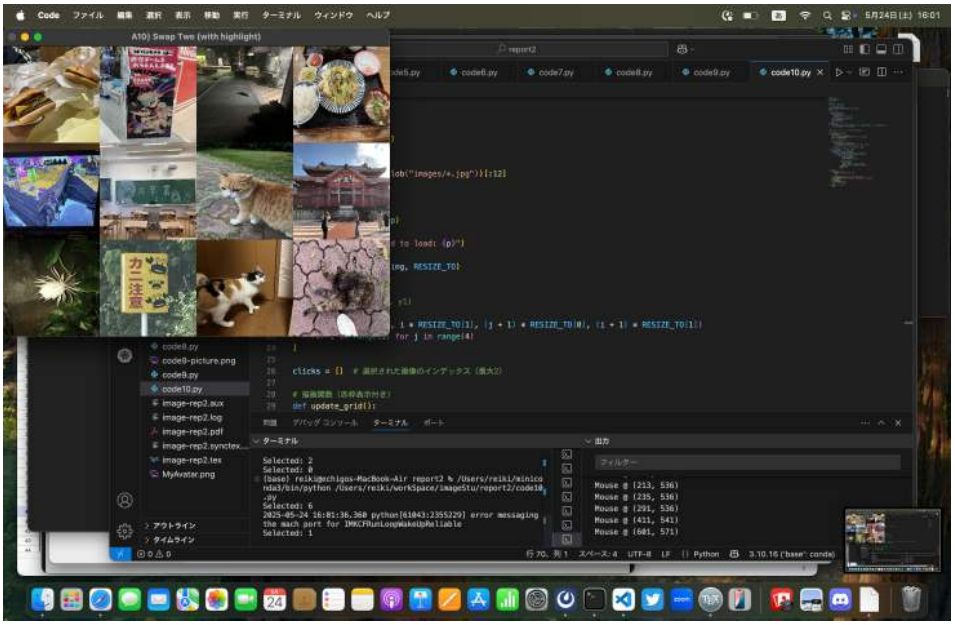

図 12　課題 A10 実行結果 2