

実世界情報実験1

画像処理テーマ

D1:中村 文彦/ D2:島田 伸敬

	第1回	第2回～第8回	第9回～第15回
D1クラス	ガイダンス	心理学実験	画像処理実験
D2クラス		画像処理実験	心理学実験

画像処理テーマ

- 画像データの取り扱い方や概念を解説し、簡単な画像処理実験を行う。
- Google Colaboratory用のノートブックをmanaba+Rからダウンロードし、課題に取り組む。Google アカウント必須



実験は**エクセル棟2階の
計算機演習室A/B**で実施
(ノートPC持参推奨)



Google Map:

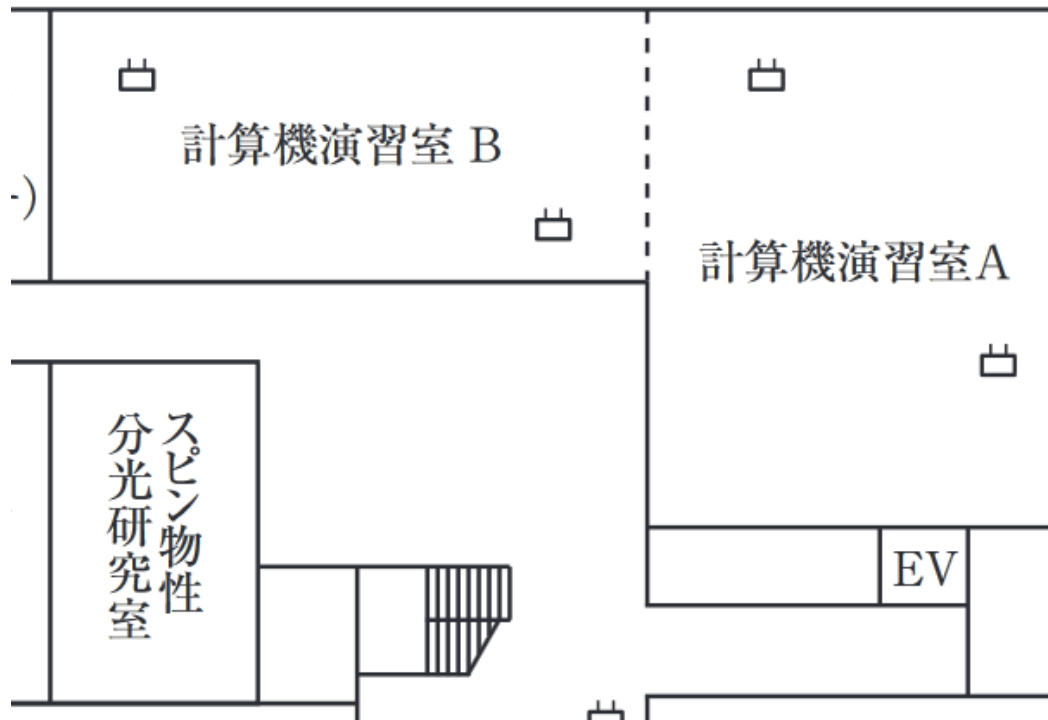
<https://goo.gl/maps/2AcK29UwLMKtWnvZ6>
plus code: 8Q6QXXJ7+CR 草津市、滋賀県

画像処理テーマ

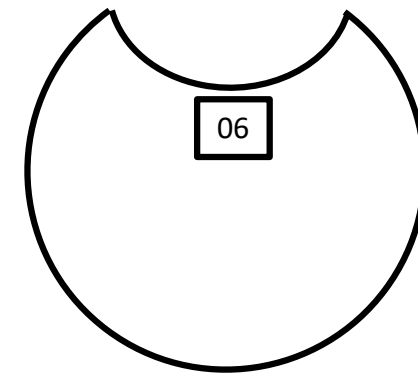
エクセル棟2階の計算機演習室A/Bで実施
各自のノートPC持参のこと
RAINBOW Wi-Fiへの接続を事前準備！



2階の角にあるL字型の教室
飲食厳禁の部屋です。



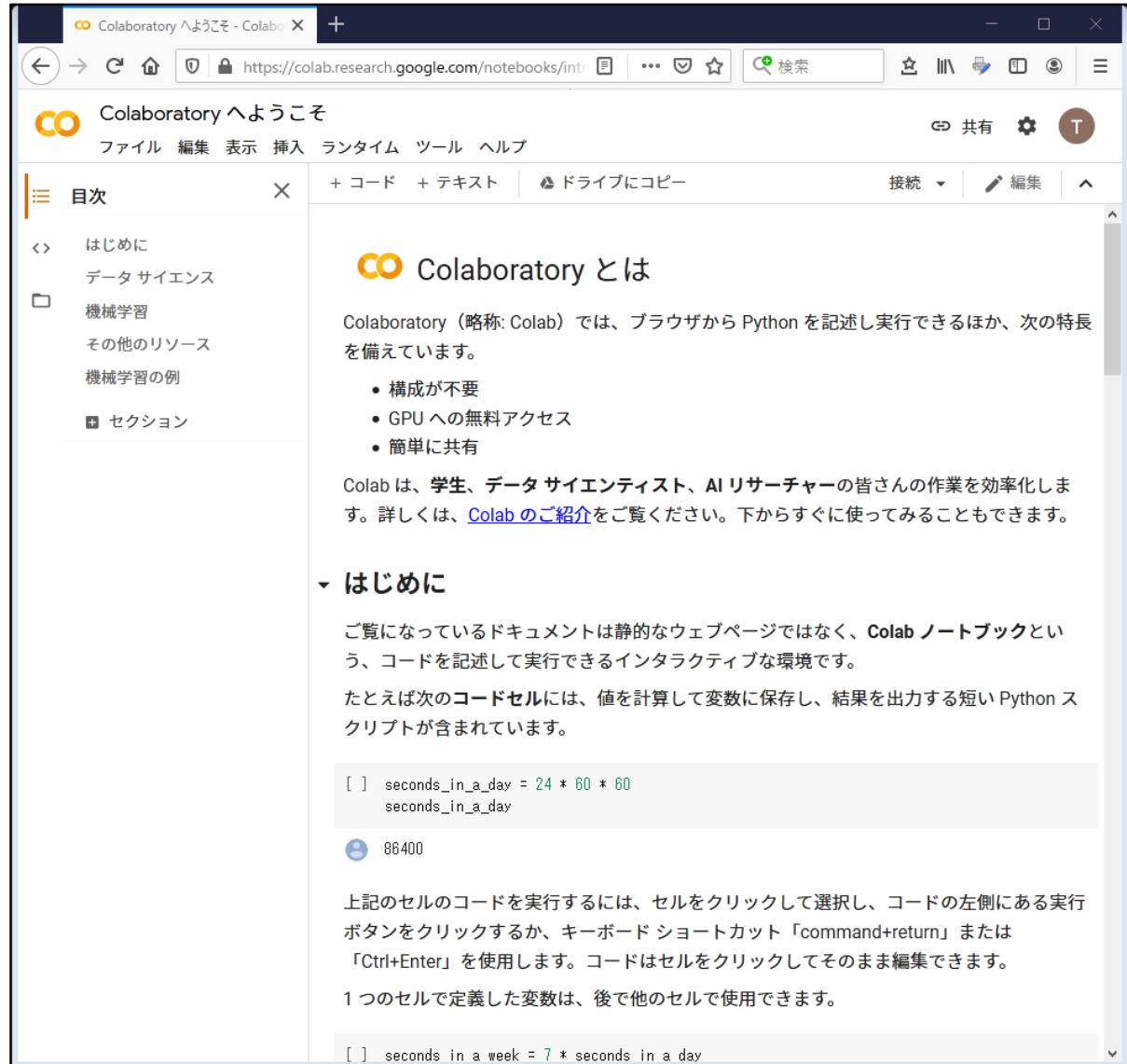
出席はICカードリーダーでとる
ので学生証必携のこと



各テーブルに2, 3人
ずつ着席

Google Colaboratory

- URL:
<https://colab.research.google.com/>
- 利用にはGoogleアカウントが必要
- ブラウザのみでPython環境での実験ができる。
- プログラムコードや実行結果を「ノートブック」としてダウンロードできる。



各週での課題と進め方

第1回: OpenCVによるデジタル画像の取扱い: 課題1

- ・ 離散化(標本化、量子化)、濃淡画像カラー画像などの理解と取り扱い、画像読み込み／画面表示などの基本的な操作

第2回: 画像処理の基礎(1): 課題2

- ・ 二値化、濃淡化等、ヒストグラム

第3回: 画像処理の基礎(2): 課題3

- ・ エッジ抽出、モルフォロジー変換、二値図形の輪郭追跡

第4回: 画像処理の基礎(3): 課題4

- ・ 特徴抽出(Harris、Haar)、SIFTなどのモジュールの利用法

第5回: 画像認識の基礎: 課題5

- ・ ハフ変換, GrabCut, 特徴点マッチング, 対応付けに基づく射影行列計算

第6回: 画像認識応用(1): 課題6

- ・ 顔検出, パターン認識

第7回: 画像認識応用(2): 課題6つづき

- ・ 画像処理サービスの提案と実装、ドキュメント整備

各週での課題と進め方

- 課題ファイルは毎週1つ(複数のプログラミング課題が含まれる)
 - 課題6は6, 7回の2週にわたって取り組む
- 課題ファイル(ipynb)と入力・結果画像などを
 - **<毎回>** manaba+Rに提出
 - 次の週までに提出されればフィードバックを受けられる
 - 再提出は何回でも可能
 - 最終の課題6の締め切り時点で提出されている内容で成績評価する

画像処理テーマ

画像データの取り扱い方や概念を解説し、PythonとOpenCVを用いた画像処理実験を行います。




```
[ ] min_height = min(height1, height2)
    result_img = np.zeros((min_height, min_width, 3), np.uint8)

alpha = 0.5
weight1 = alpha
weight2 = 1.0 - alpha
for y in range(min_height):
    for x in range(min_width):
        for c in range(3):
            v = weight1 * img1[y, x, c] + weight2 * img2[y, x, c]
            result_img[y, x, c] = v

plt.figure(figsize=(10, 4))
plt.subplot(1, 3, 1)
plt.xticks([], plt.yticks([]))
plt.imshow(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB))
plt.subplot(1, 3, 2)
plt.xticks([], plt.yticks([]))
plt.imshow(cv2.cvtColor(img2, cv2.COLOR_BGR2RGB))
plt.subplot(1, 3, 3)
plt.xticks([], plt.yticks([]))
plt.imshow(cv2.cvtColor(result_img, cv2.COLOR_BGR2RGB))
```

<matplotlib.image.AxesImage at 0x7f177b354f98>



昨年度のプログラミング演習1と同じくGoogle Colaboratoryでプログラムを作成し実行結果を確認しながら取り組みます（PCにインストールも可）

画像処理ライブラリOpenCV

- OpenCVライブラリチュートリアルページ

- <http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/index.html>

若干古いが基本的な内容については十分.

- オリジナル(英語)(以下は2023/4/3時点最新)

- https://docs.opencv.org/4.7.0/d6/d00/tutorial_py_root.html

OpenCV3-Python Tutorials ▶



OpenCV

Previous topic
OpenCV3-Python チュートリアル

Next topic
OpenCVの導入

This Page
[Show Source](#)

Quick search

[Go](#)

OpenCV3-Pythonチュートリアル

- OpenCVの導入
OpenCV-Pythonを使う環境を自分のPC上に構築する方法を学ぶ

- OpenCVのGUI機能
画像と動画の表示方法と保存方法、およびGUIの機能であるマウスとトラックボールの操作

- 基本の処理
画像に対する基本の処理を学ぶ: 画素値の編集, 幾何変換, コードの最適化(code opt)

00	01	02	03
04	05	06	07
08	09	10	11
12	13	14	15
- OpenCVを使った画像処理
OpenCVが提供する様々な画像処理の関数について学ぶ

- 特徴量検出と特徴量記述
特徴検出器, 特徴量記述子について学ぶ

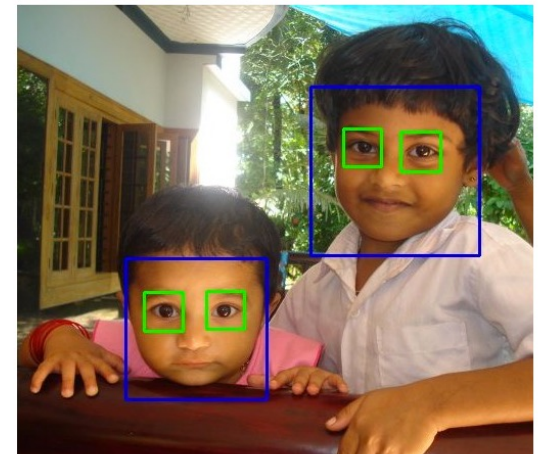
- 動画解析

次に入力画像に対し顔検出を行う。顔が検出された時は、顔の位置がRect(x,y,w,h)として返される。この位置情報から、顔に対する注ができる(なぜなら目は常に顔の中に位置するからである!!).

```
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
for (x,y,w,h) in faces:
    img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

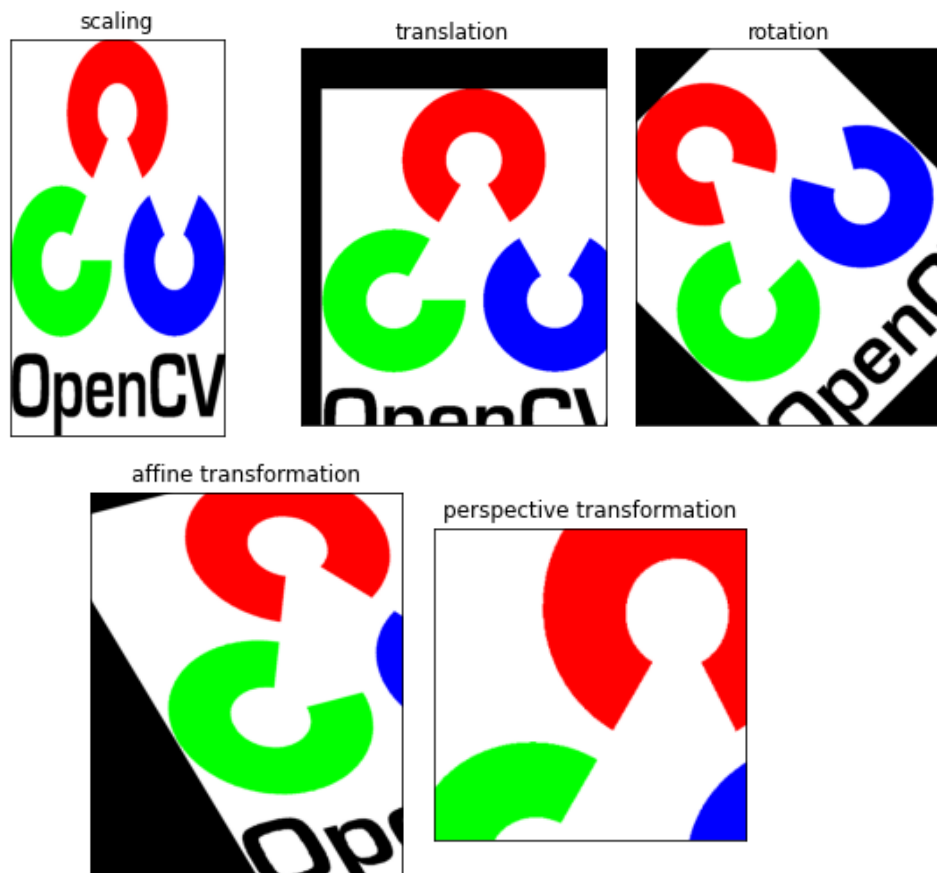
cv2.imshow('img',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

結果を以下に示す:(コード, XMLファイルはここから必要なものをダウンロードすること)



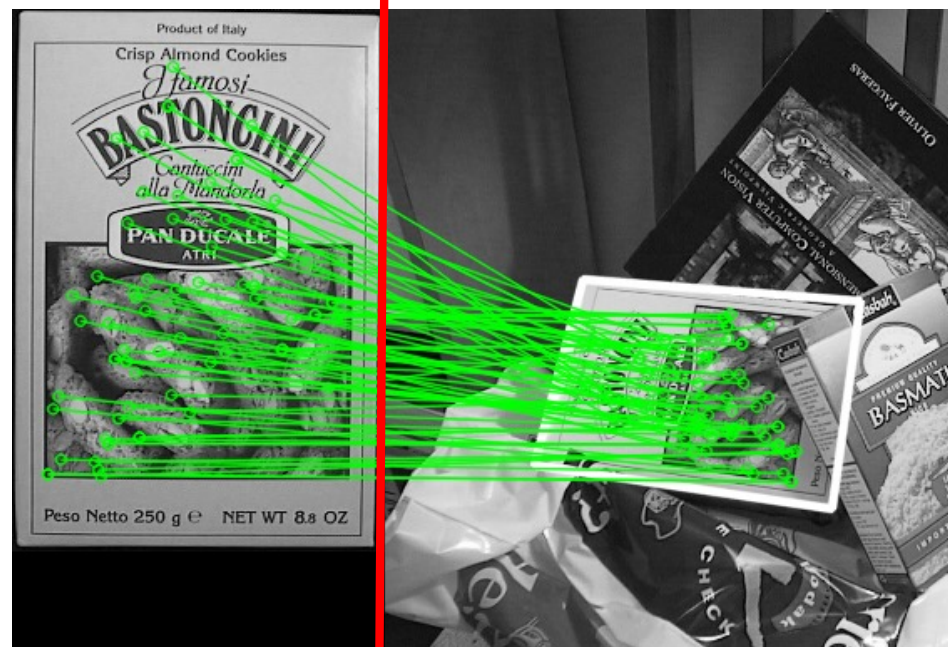
画像処理テーマ

画像の合成や変形などの簡単な操作から物体検出などのより高度な処理までOpenCVを使って実装していきます。



対象物体

対象物体を含む風景

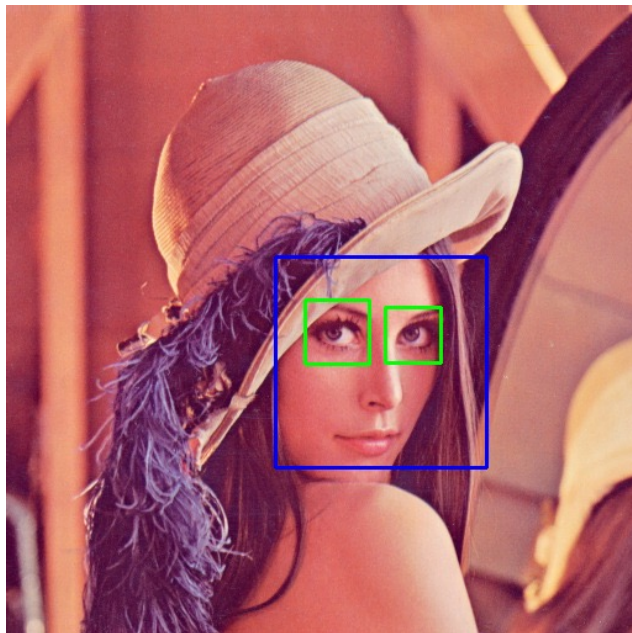


特徴点を用いた物体検出

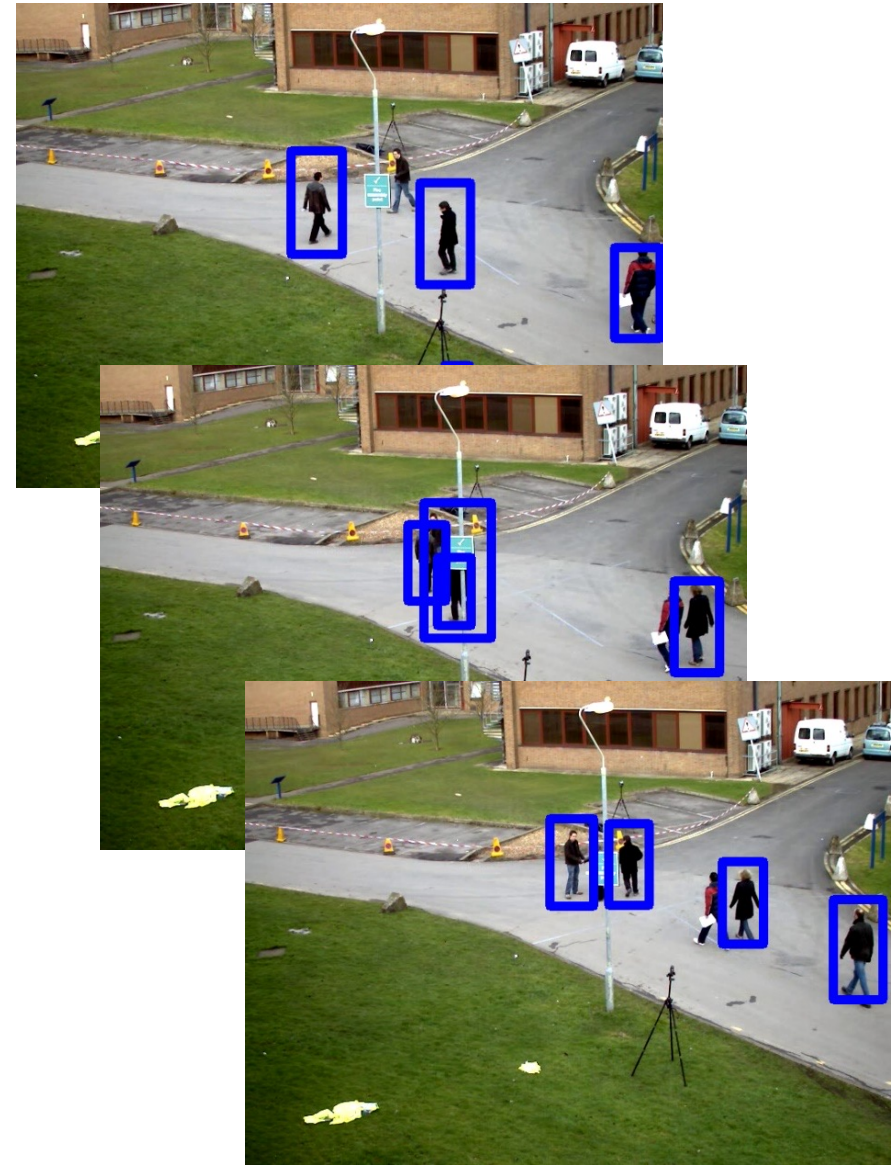
画像処理テーマ

OpenCVには機械学習の技術を用いた歩行者検出や顔検出の機能もあります。

これらの機能を組み合わせて課題のプログラムを作成し、画像処理の基礎やPythonでの外部ライブラリ使用を実践的に学びます。



顔検出+目検出



動画内の歩行者検出