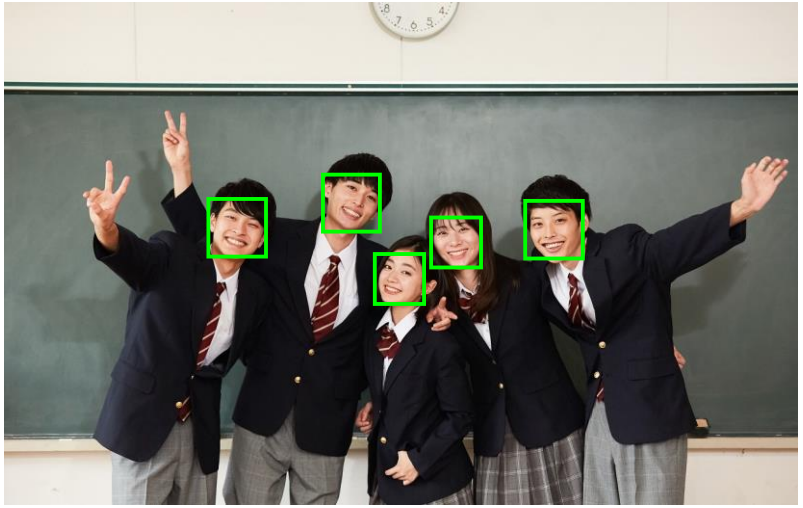
情報とデータサイエンス

モデルを用いた画像認識

自動で顔にぼかしを入れよう！

「顔認識AI」

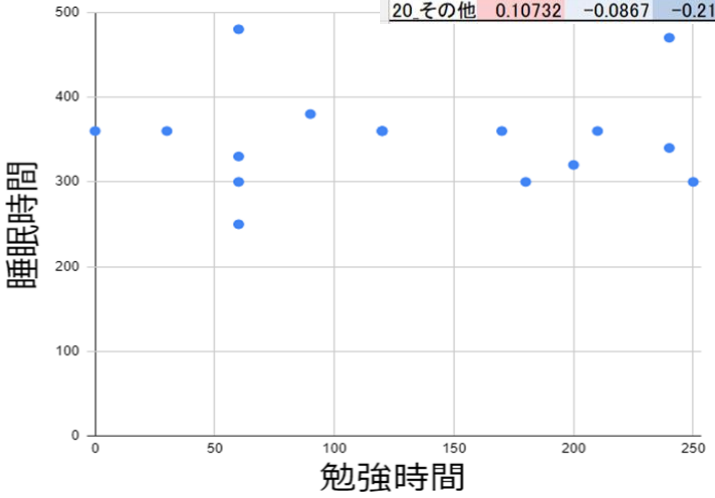


実は「データサイエンス」単元と深い関わりを持つ

データサイエンスというと…？

01_睡眠	06_学業
467	355
469	368
458	453
448	373
467	373
477	377
436	438
456	387
472	304

	01_睡眠	02_身の回りの	03_食事	04_通勤・通	05_仕事	06_学業	07_家事	10_買い物(通勤・通	
01_睡眠									
02_身の回	−0.2308								
03_食事	−0.129	0.04927							
04_通勤・	−0.3563	0.19467	0.28387						
05_仕事	−0.3274	0.08675	−0.1459	0.02845					
06_学業	−0.3835	−0.1301	−0.2055	−0.0192	−0.0992				
07_家事	0.03418	−0.2337	0.04911	0.06168	−0.1948	0.05732			
10_買い物	**0.26421**	0.02212	0.0845	0.12755	−0.4664	−0.239	0.40575		
11_移動(通	−0.1549	0.14999	0.02546	−0.107	−0.1786	−0.2163	−0.0613	0.07928	
12_テレビ・	**0.35748**	−0.1084	0.11338	−0.0222	−0.2739	−0.0922	0.00931	0.13164	−0.0849
13_休養・く	0.09791	−0.1746	0.04584	−0.1438	0.2955	−0.4069	−0.0013	−0.2121	0.1344
14_学習・	−0.1175	0.37528	−0.0702	0.03141	0.02206	−0.2363	−0.2223	−0.0377	0.10067
15_趣味・娯	0.12724	−0.1379	0.11829	−0.052	−0.3413	−0.2828	−0.0363	0.32197	0.02185
16_スポー	**0.0383**	−0.0809	−0.0632	−0.2982	0.01111	−0.1535	−0.0966	−0.0007	0.06199
20_その他	0.10732	−0.0867	−0.2132	−0.3353	−0.1576	0.01509	0.19406	0.22306	0.18288



「データ」から学習し、作ったモデルで検出

検出フェーズ

入力画像

「データ」から学習し、作ったモデルで検出

検出フェーズ

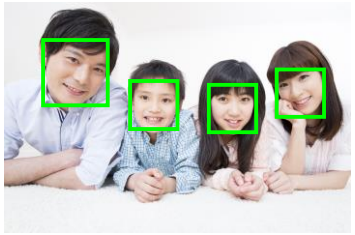
入力画像

特徴量抽出

顔っぽいのは、
ここだ！

検出

認識結果



「データ」から学習し、作ったモデルで検出

顔って、
こんなものだ！

学習結果データ

検出フェーズ

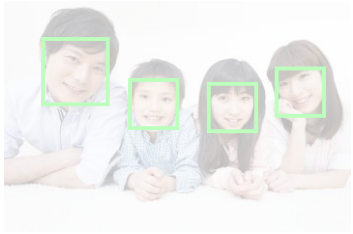
入力画像

特徴量抽出

検出

認識結果

顔っぽいのは、
ここだ！



「データ」から学習し、作ったモデルで検出

顔って、
こんなものだ！

学習フェーズ

学習画像

特徴量抽出

学習

学習結果データ

検出フェーズ

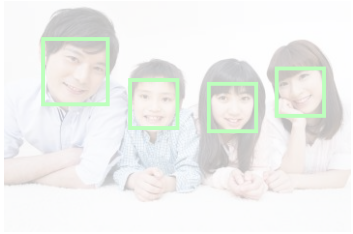
入力画像

特徴量抽出

検出

認識結果

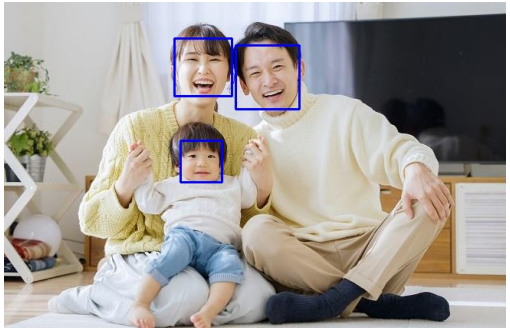
顔っぽいのは、
ここだ！



この動画でやってみること

学習については別動画で！

学習済みのモデルを使って、
顔の検出・ぼかしに挑戦してみよう！


OpenCVとは？

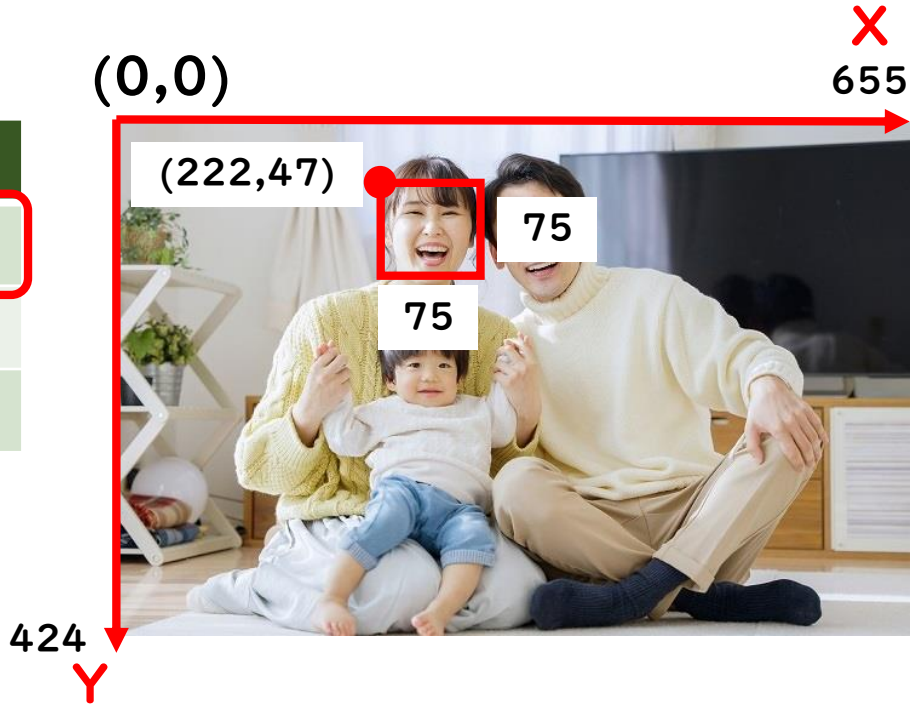
- OpenCV (Open Source Computer Vision Library)
- インテルが開発・公開したオープンソースのコンピュータビジョン向けライブラリ
- Google Colaboratoryですぐに使うことができる！

出力結果の解釈

faces→検出した顔のリスト

顔番号	x	y	w	h
⓪	222	47	75	75
①	302	56	83	83
②	230	180	54	54

x ：左上x座標
y ：左上y座標
w：幅（width）
h：高さ（height）



コードの説明

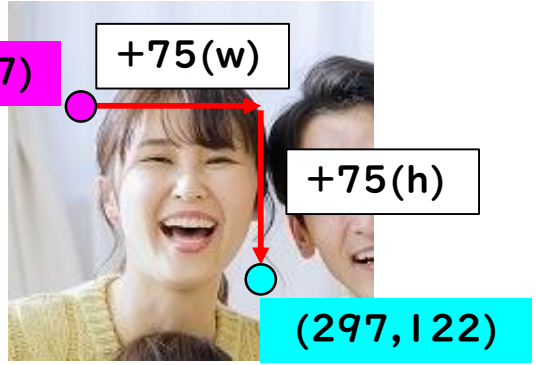
```
for(x,y,w,h) in faces:
    cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),3)
```

矩形(≒四角形)

(x, y): 矩形の左上の頂点の座標
(x+w, y+h): 矩形の右下の頂点の座標
(255, 0, 0): 描画する矩形の色。BGR形式。
3: 矩形の線の太さ。

facesリストから、
順番に顔データを
取り出す。

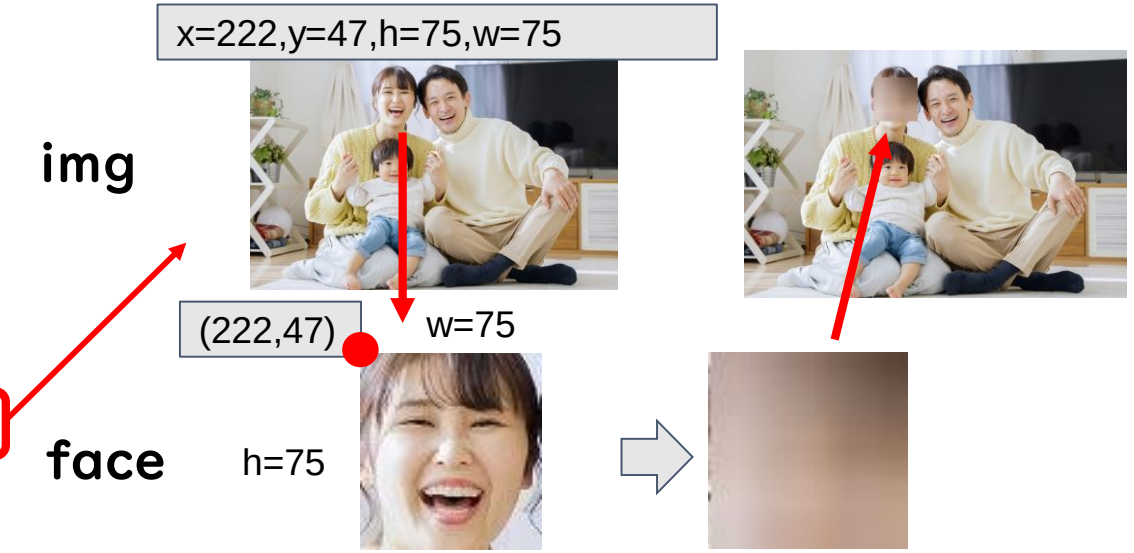
顔番号	x	y	w	h
0	222	47	75	75
1	302	56	83	83
2	230	180	54	54



コードの説明

```
for(x,y,w,h) in faces:
    face = img[y:y+h,x:x+w] #顔部分を切り出す
    face = cv2.blur(face,(55,55)) #切り出しにぼかしを入れる
    img[y:y+h,x:x+w]=face #元の画像データに戻す
```

顔番号	x	y	w	h
0	222	47	75	75
1	302	56	83	83
2	230	180	54	54



発展

- 検出モデルを変えてみる
  haarcascade_frontalface_alt.xml → haarcascade_eye.xml
- 精度を高めるためには？
  - 「誤検出」や「検出漏れ」もある →パラメータを調整してみる
  - カスケード型分類器以外を試してみる etc..
- 顔に間違われるものには何がある？
- 様々なシステムを作ってみよう！