ae-10.ディープラーニングによる顔情報処理技術の基礎と応用

（ディープラーニング，Python を使用）

（全１５回）

https://www.kkaneko.jp/ai/ae/index.html

金子邦彦

人工知能（AI）の学習のための指針

1. **実践重視**： AIツールを実際に使用し、機能に慣れる

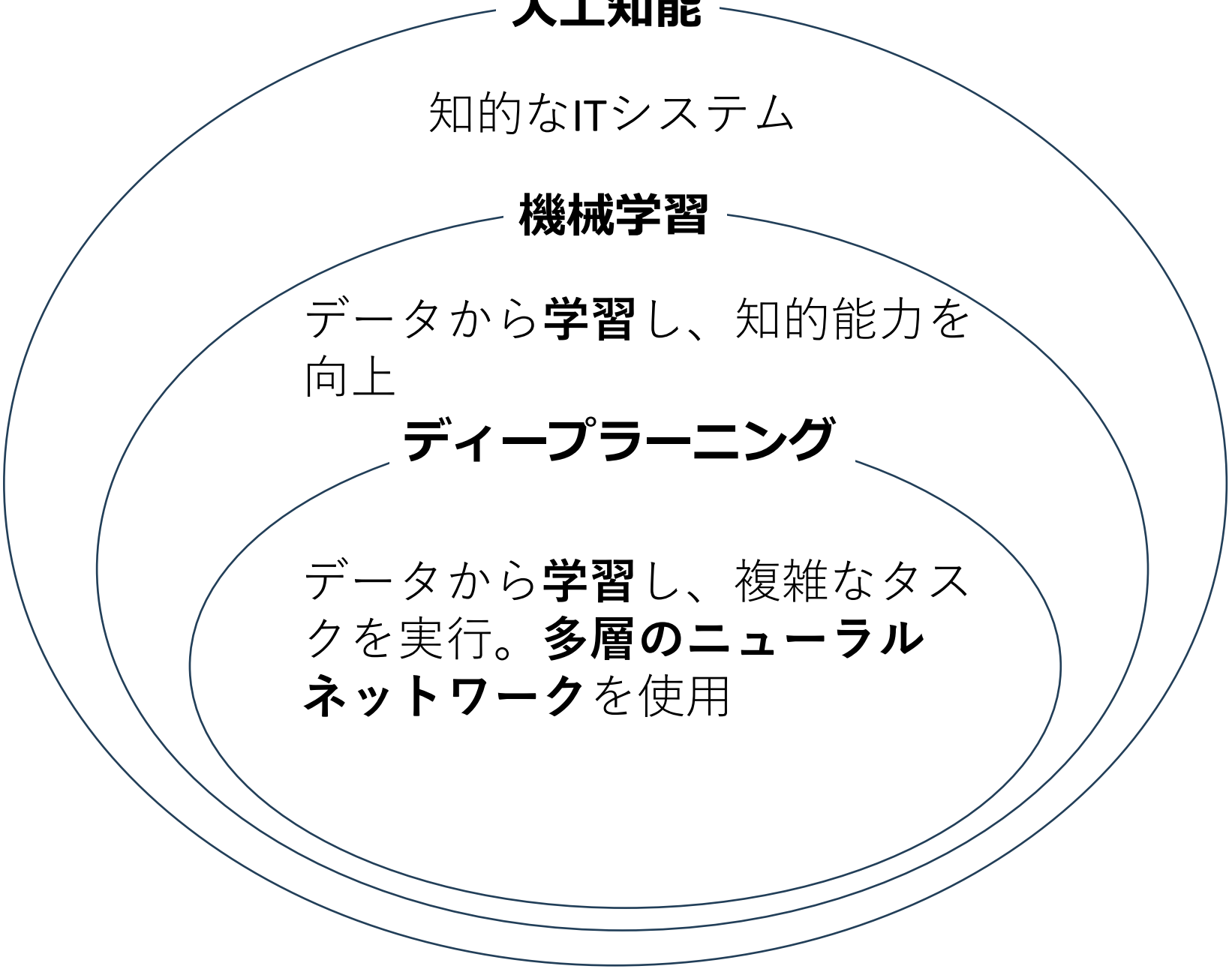
2. **エラーを恐れない**： 実行においては、エラーの発生の可能性がある．エラーを恐れず，むしろ学習の一部として捉えるポジティブさが大切．

3. **段階的学習**：基礎から応用へと段階的に学習を進め，AIの可能性を前向きに捉える

アウトライン

1. イントロダクション
2. 顔情報処理の概要
3. ディープラーニングによる顔検出
4. 顔ランドマーク
5. ディープラーニングによる顔認識
6. ディープラーニングによる感情認識その他

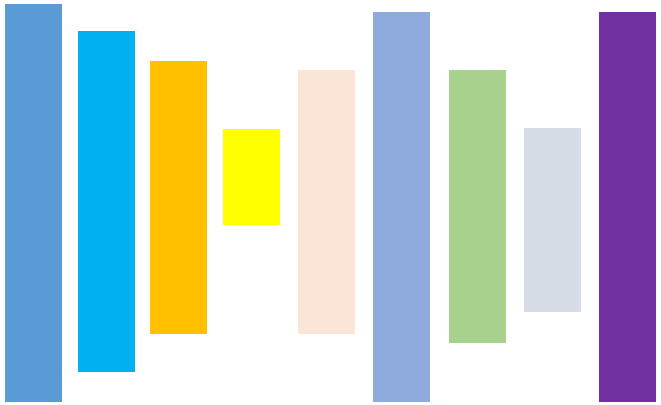
10-1. イントロダクション


人工知能
知的なITシステム
機械学習
データから学習し、知的能力を向上
ディープラーニング
データから学習し、複雑なタスクを実行。多層のニューラルネットワークを使用

ディープニューラルネットワーク

ディープラーニングは**多層のニューラルネットワーク**を用いた機械学習

層の数が少ない（浅い）



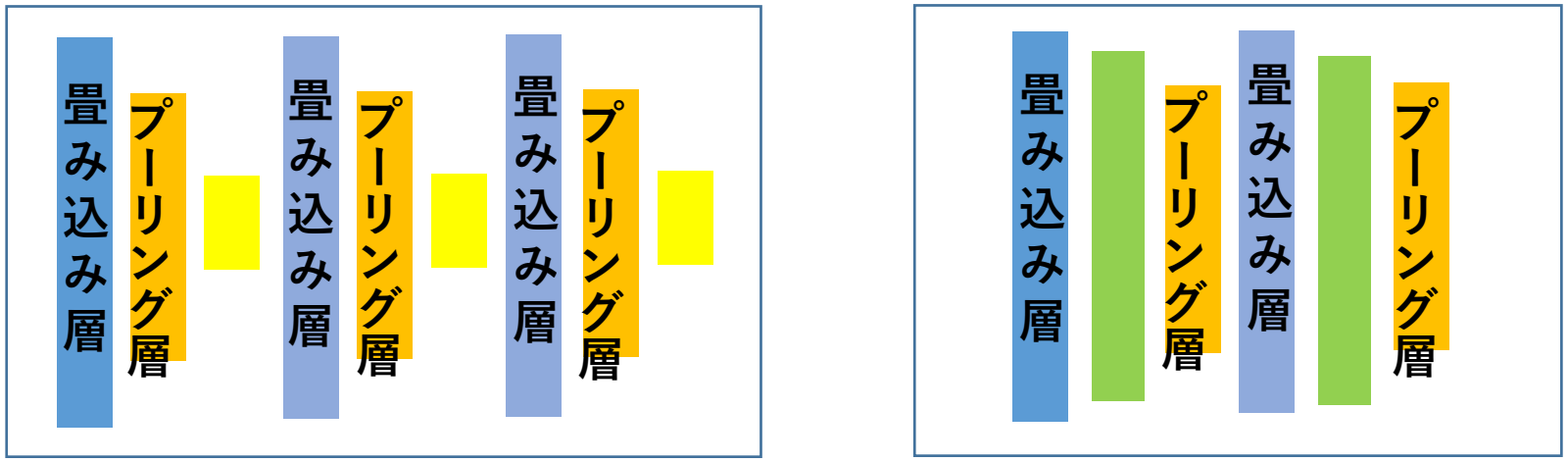
層の数が多い（深い）

畳み込みニューラルネットワーク（CNN）の基本構造

- **画像理解**や**画像の分析**に特化した**ディープラーニング**の一種.
- 主に**畳み込み層**、**プーリング層**、**全結合層**の３種類で構成
  - **畳み込み層**：画像の局所的な**特徴**（エッジ、テクスチャなど）をとらえる
  - **プーリング層**：**特徴マップ**の**サイズを縮小**し，**過学習を防止.**
  - **全結合層**：**特徴マップ**をもとに**画像**を処理する

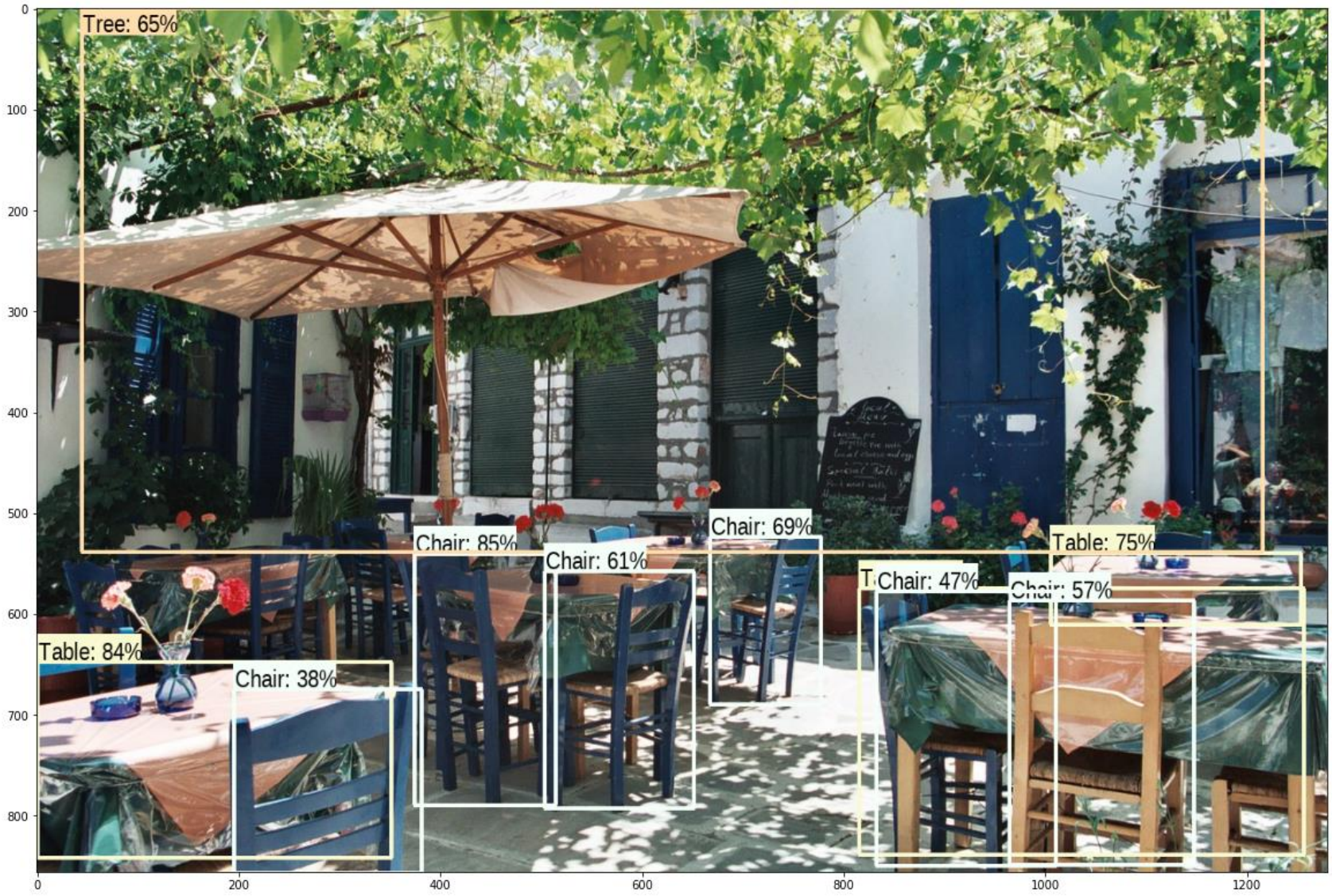
畳み込みニューラルネットワーク（CNN）の例

- CNNの主要な層構成：
  - **プーリング層**
  - **畳み込み層**
  - **全結合層**
- これらの層を組み合わせることで，さまざまなバリエーションのCNNを構築できる.

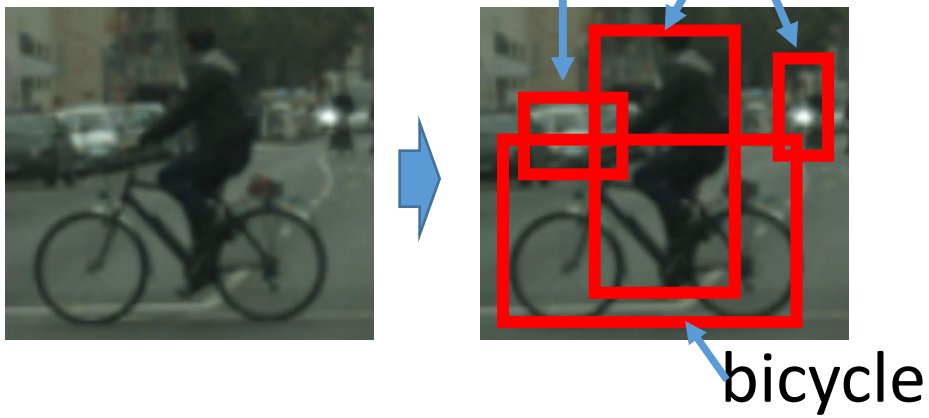


さまざまなバリエーション

物体検出の例


Tree: 65%
Chair: 85%
Chair: 61%
Chair: 69%
Table: 75%
Chair: 47%
Chair: 57%
Table: 84%
Chair: 38%
0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000
1200

物体検出

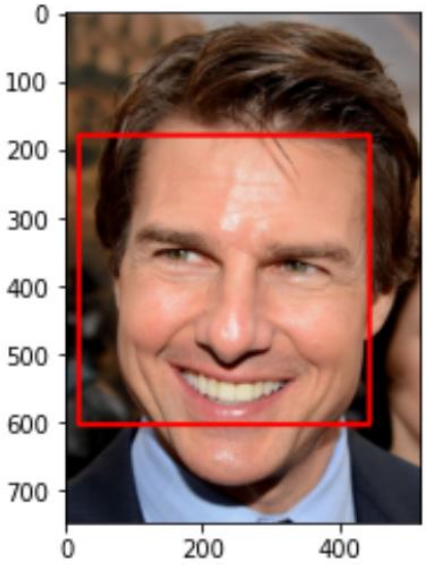


バウンディングボックス，
ラベルを得る

バウンディングボックスは，
物体を囲む最小のボックス（四角形）

10-2. 顔情報処理の概要

顔情報処理のバリエーション①

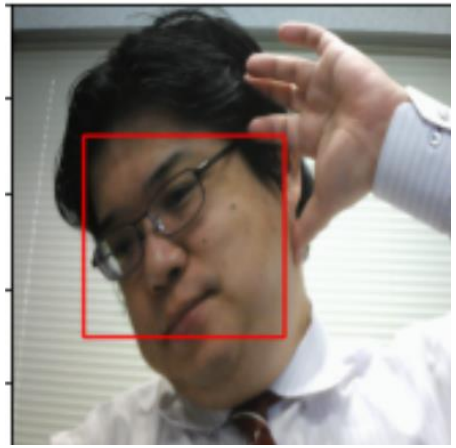


顔検出結果

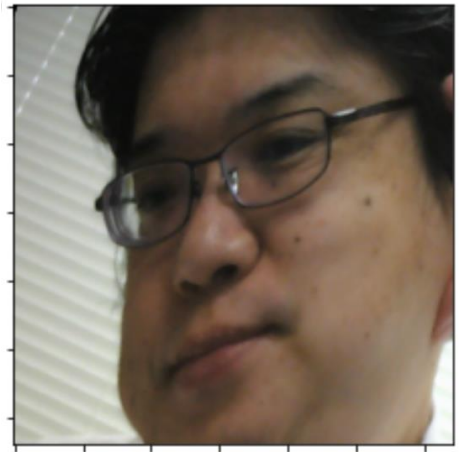
顔検出結果

- **顔検出**

物体検出と同様に顔を検出



アラインメント前

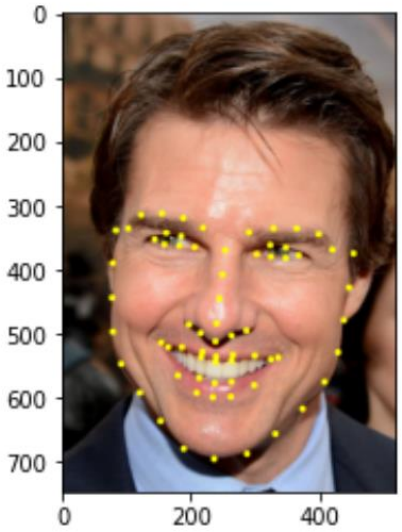


アラインメント後

- **顔のアラインメント**

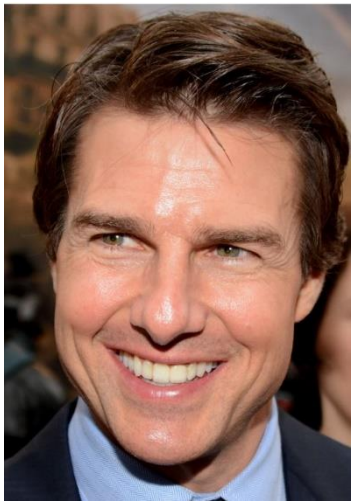
顔の向き，大きさをそろえる

顔情報処理のバリエーション②



- **顔ランドマーク**

顔の重要な特徴点（目、鼻、口など）の位置を示すポイント



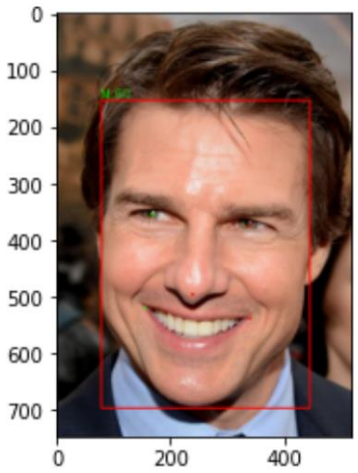
```
[ 0.00455161  0.18412845  0.03993299 -0.00525442 -0.07139583  0.13386713
 -0.00805785 -0.05532961  0.23680274 -0.08285979  0.23483047  0.00531268
 -0.20885959 -0.08143201 -0.07200904  0.08402673 -0.11594398 -0.08790354
 -0.14109443 -0.06426463 -0.01909781  0.04940728 -0.09086579  0.04928832
 -0.21027854 -0.29970455 -0.0172264  -0.11317077  0.01727176 -0.16984728
 -0.09449591  0.13467805 -0.11701771 -0.05477776 -0.00075595  0.02513538
 -0.14347562 -0.03627753  0.16524641 -0.01949969 -0.12205475  0.08029588
  0.0263175   0.27922362  0.26064846  0.00236352 -0.00611193 -0.09485988
  0.21062998 -0.27925995  0.02801728  0.21664803  0.10859946  0.12886725
  0.13740893 -0.2259727  -0.04096841  0.08036234 -0.15912706  0.04860799
 -0.05707375 -0.0537034  -0.0382829  -0.04455713  0.08876641  0.08084331
 -0.08434555 -0.12838244  0.14743124 -0.16956419  0.05809632  0.13801341
 -0.08743402 -0.27705559 -0.17667291  0.12853563  0.40629697  0.25780377
 -0.18405011  0.0308622  -0.04818188 -0.08573647  0.05904602 -0.00438196
 -0.12593931 -0.09314068 -0.07387269  0.06559635  0.18202811  0.02111054
 -0.0110196   0.23951159 -0.00382505 -0.03864232  0.03268222  0.00397856
 -0.16739431  0.05639653 -0.11059792 -0.01466753  0.09382112 -0.13222337
  0.00125152  0.08060078 -0.16358295  0.22464372 -0.00883166  0.03960528
  0.02616193 -0.04999878 -0.14178708  0.06188779  0.25730038 -0.27515596
  0.19043125  0.04089454  0.02481355  0.17113039  0.10915946  0.05673687
 -0.00430411 -0.0710548  -0.14189517 -0.08640159  0.0943445  -0.08253401
  0.04771694 -0.04451045]
```

- **顔の数値化**

顔の特徴を数値化したもの．顔による本人確認や、顔認識に使用

数値は複数になる．

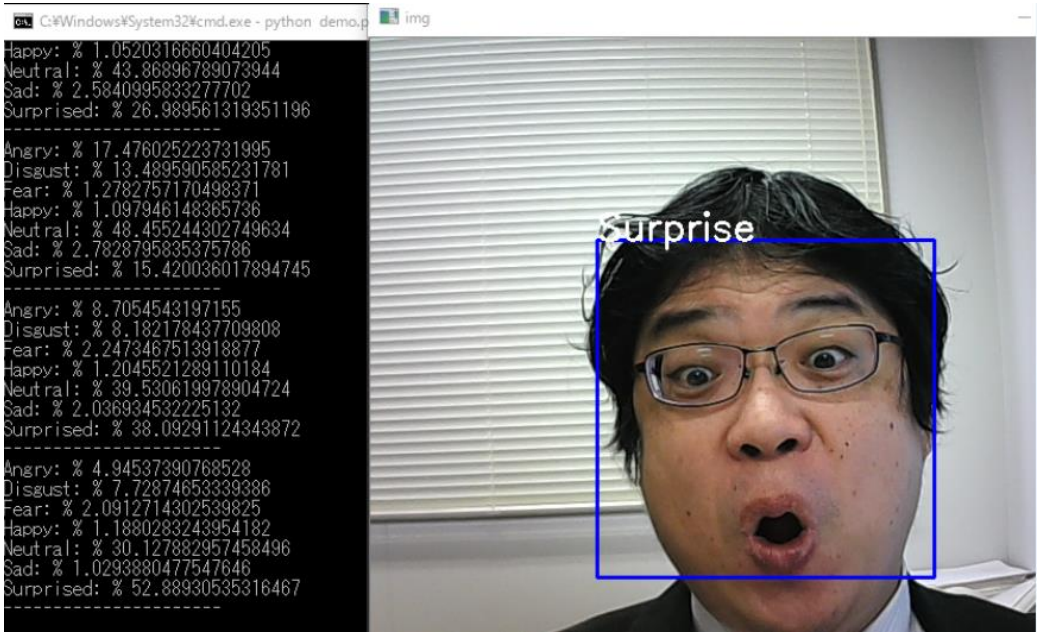
顔情報処理のバリエーション③



gender： 1
age： 60

さまざまな追加機能

- **性別、年齢の推定**



- **表情の推定**

https://github.com/ezgiakcora/Facial-Expression-Keras で公開されている成果物を利用

顔情報処理のバリエーション

- **顔検出**：物体検出の一環として顔を特定
- **顔のアラインメント**：顔の向きや大きさの調整
- **顔ランドマーク**：目，鼻，口などの重要な特徴点の位置
- **顔の数値化**：顔による本人確認や顔認識の基礎
- **性別，年齢の推定**
- **表情の推定**
- その他

顔情報処理の重要性

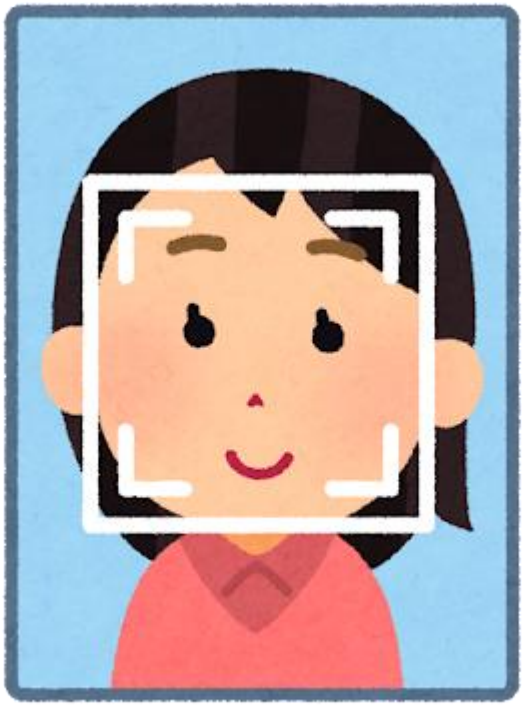
多岐にわたる応用分野

- **セキュリティと監視**：公共の場での監視，身元確認や不審者の追跡
- **顔認証システム**：個人認証技術
- **パーソナライズされたサービス**：個人に適応したサービス提供
- **感情認識**：表情からの感情推定
- その他の応用分野

セキュリティと監視

主な用途：公共の場での監視，身元確認システム，
不審者の追跡，セキュリティ管理

顔認証システム



主な用途：スマートフォンでの顔認証，パソコンでの顔認証，
セキュリティゲート，入退室管理

顔情報処理の歴史

- 誕生（1960年代）

顔認識システムの誕生

- 進化（1980年代）

顔認識アルゴリズム（Eigenfaces など）

- ディープラーニングの登場（2010年代）

精度の向上、

応用の広がり（群衆のカウント、マスク有り顔の処理、顔画像からの３次元顔生成などさまざまな応用）

群衆のカウント技術

画像内の人数を数える．監視等に役立つ．

元画像

FIDTM 法による群衆のカウント

FIDTM法（2021年）の特徴

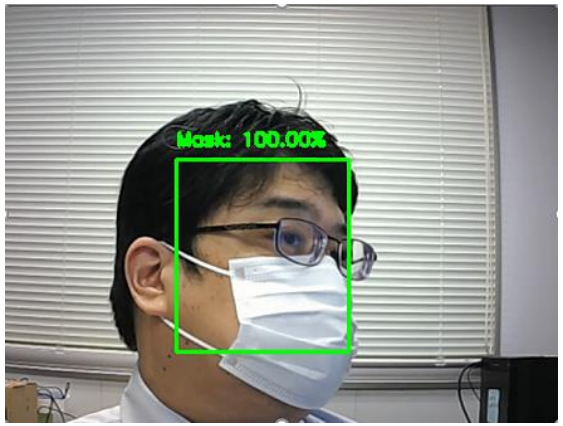
- 群衆のカウントが可能
- 監視等に役立つ
- それ以前の手法より高精度
- 様々な大きさの顔を検出可能

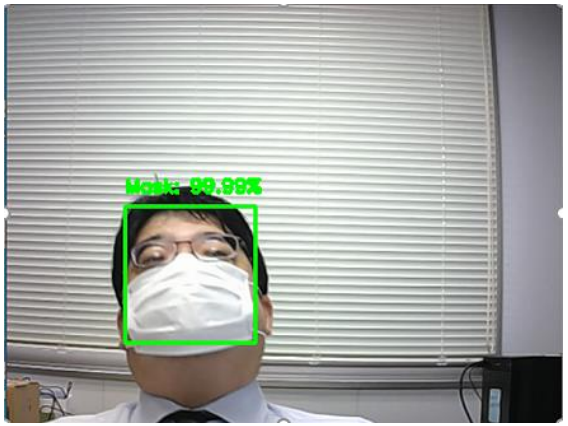
マスク有りの顔，マスク無しの顔検出技術

Chandrika Deb の顔マスク検出法

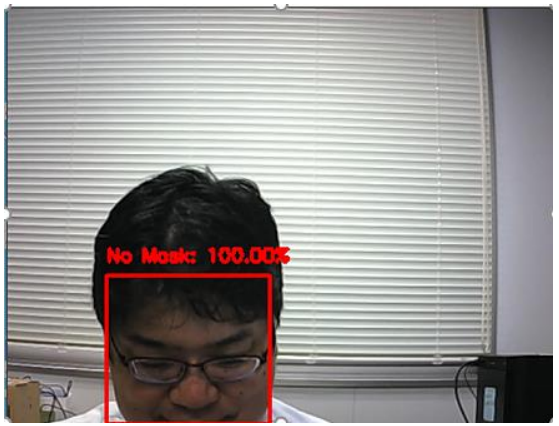
次を同時に実行

- 顔検出
- マスク有りの顔とマスク無しの顔の画像分類





マスク有りの顔検出



マスク無しの顔検出

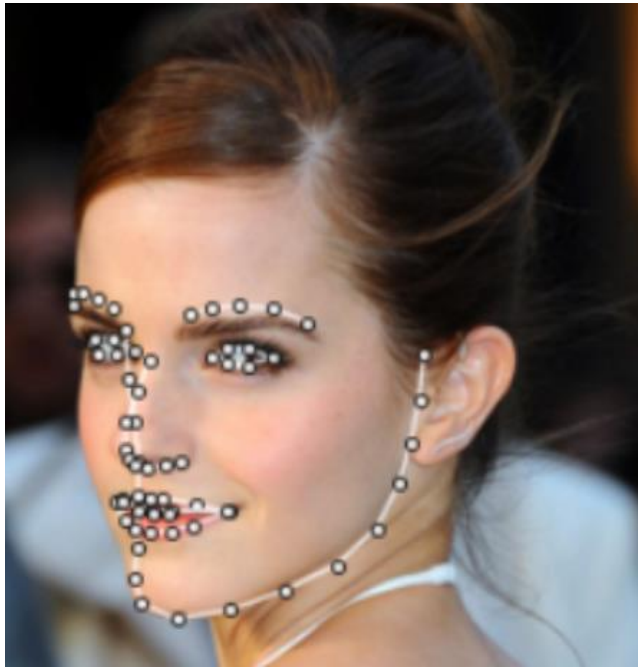
訓練データ：顔のデータセット（マスク有り: 2165 枚，マスク無し 1930 枚）

顔画像からの３次元再構成

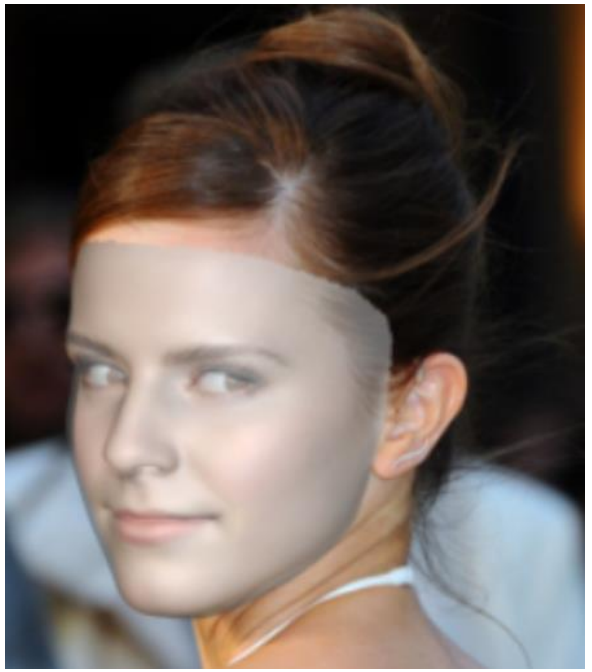
3DFFA 法（2022年発表）の特徴

- 顔画像から，３次元の顔を生成（３次元再構成）
- 顔検出，顔ランドマークの検出の実施
- 顔ランドマークに，顔の３次元モデルをあてはめる

元画像

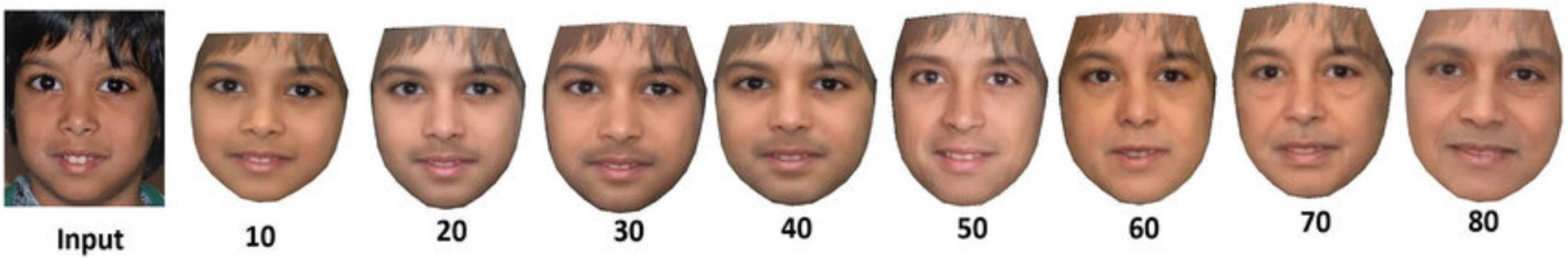


顔ランドマーク



３次元再構成

Face aging 技術



Given the frontal view of a single image of a child, an ageing algorithm can generate his faces with varied ages

- 元の顔画像から様々な年齢の顔画像を生成
- 自然な年齢変化を表現
- Ali Elmahmudi, H. Ugailによる2020年の研究成果

Ali Elmahmudi, H. Ugail,
A framework for facial age progression and regression using exemplar face templates, 2020.

ここまでのまとめ

顔情報処理のバリエーション

- 顔検出：物体検出の一環として顔を特定
- 顔のアラインメント：顔の向きや大きさの調整
- 顔ランドマーク：目、鼻、口などの重要な特徴点の位置
- 顔の数値化：本人確認、顔認識の基礎
- 性別、年齢の推定
- 表情の推定

顔情報処理の重要性

- セキュリティと監視
- 個人認証システム
- パーソナライズされたサービス
- 感情認識

演習１．顔検出

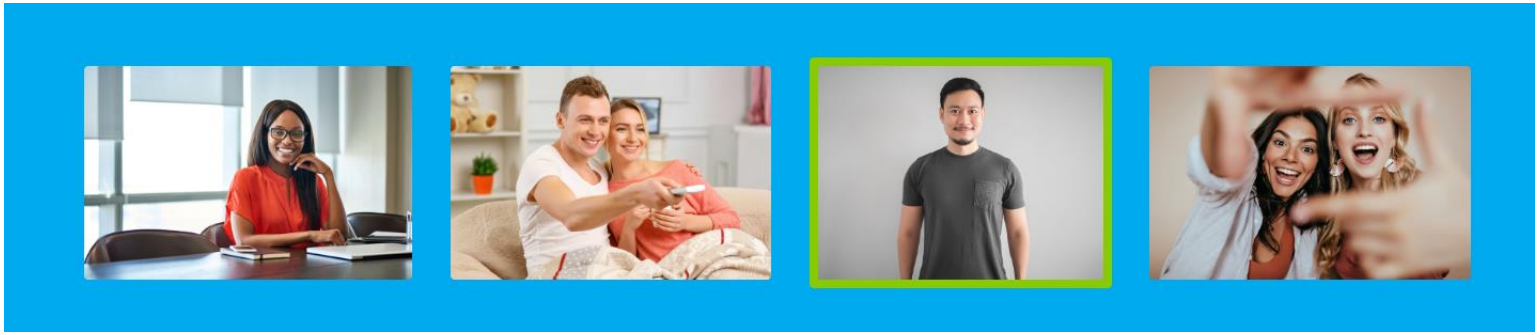
オンラインデモによる

顔検出の体験

① sky biometry の顔検出デモページ

https://skybiometry.com/demo/face-detect/

② 顔画像を選択．結果を確認



顔検出アプリ

各自、ソースコードと実行結果を確認しよう

https://colab.research.google.com/drive/18bN3_pLhAnGAJnvQljZoUUjVhVqI_MLe?usp=sharing

```
!pip install mediapipe opencv-python-headless

import cv2
import mediapipe as mp
import numpy as np
from typing import Tuple
from google.colab import files
from google.colab.patches import cv2_imshow

class FaceDetector:
    def __init__(self, min_detection_confidence: float = 0.5):
        self.mp_face_detection = mp.solutions.face_detection
        self.mp_drawing = mp.solutions.drawing_utils
        self.face_detection = self.mp_face_detection.FaceDetection(
            min_detection_confidence=min_detection_confidence
        )

    def detect_faces(self, image: np.ndarray) -> Tuple[np.ndarray, list]:
        image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        results = self.face_detection.process(image_rgb)

        annotated_image = image.copy()
        face_locations = []

        if results.detections:
            for detection in results.detections:
                bboxC = detection.location_data.relative_bounding_box
                ih, iw, _ = image.shape
                x, y = int(bboxC.xmin * iw), int(bboxC.ymin * ih)
                w, h = int(bboxC.width * iw), int(bboxC.height * ih)

                face_locations.append((x, y, w, h))

                cv2.rectangle(
                    annotated_image,
                    (x, y),
                    (x + w, y + h),
                    (0, 255, 0),
                    2
                )

                score = detection.score[0]
                cv2.putText(
                    annotated_image,
                    f'Score: {score:.2f}',
                    (x, y - 40),  # 位置を上に調整
                    cv2.FONT_HERSHEY_SIMPLEX,
                    2.0,          # フォントサイズを4倍に
                    (0, 255, 0),
                    4             # 線の太さを調整
                )

                for keypoint in detection.location_data.relative_keypoints:
                    kp_x = int(keypoint.x * iw)
                    kp_y = int(keypoint.y * ih)
                    cv2.circle(
                        annotated_image,
                        (kp_x, kp_y),
                        2,
                        (255, 0, 0),
                        2
                    )

        return annotated_image, face_locations

def process_image():
    try:
        print("画像ファイルをアップロードしてください")
        uploaded = files.upload()

        for filename in uploaded.keys():
            print(f"処理中の画像: {filename}")

            file_bytes = np.asarray(bytearray(uploaded[filename]), dtype=np.uint8)
            image = cv2.imdecode(file_bytes, cv2.IMREAD_COLOR)

            if image is None:
                print(f"エラー: 画像の読み込みに失敗しました - {filename}")
                continue

            detector = FaceDetector(min_detection_confidence=0.7)
            result_image, faces = detector.detect_faces(image)

            print(f"検出された顔の数: {len(faces)}")

            cv2_imshow(result_image)

    except Exception as e:
        print(f"エラーが発生しました: {str(e)}")

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

発展演習１

①顔画像を準備

②演習１と同じページ（sky biometry の顔検出デモページ）

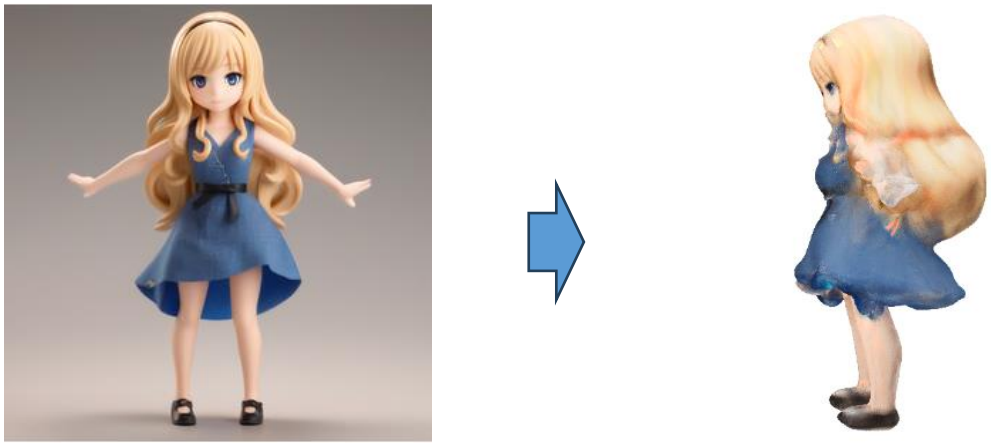
https://skybiometry.com/demo/face-detect/

③「Custom Upload」をクリックしてアップロード

発展演習２

次の Google Colaboratory で、**画像の３次元化**を体験してみる。
１つの画像を、ディープラーニングにより３次元化している。

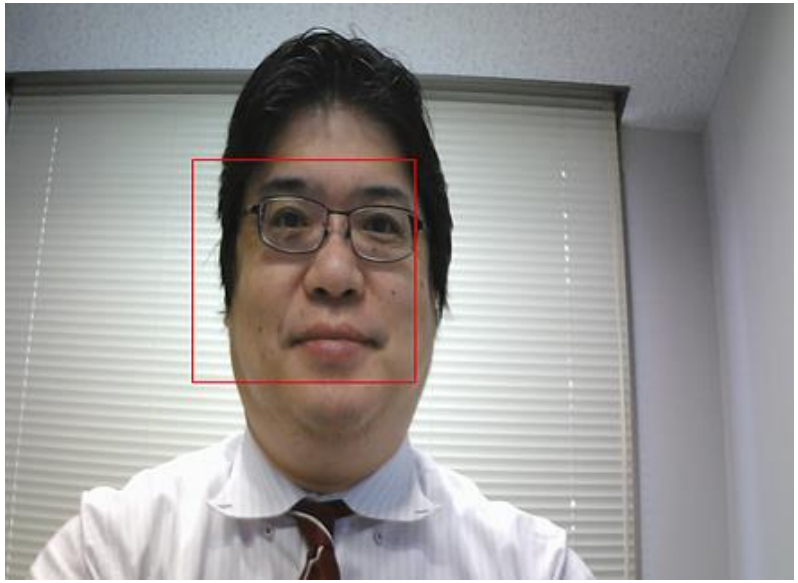
https://colab.research.google.com/drive/1sLpYmmLS209-e5eHgcuqdryFRRO6ZhFS?usp=sharing



自分で画像を準備し、アップロードして実行してみることもできる

10-3. ディープラーニングによる顔検出

顔検出の基本



- 顔検出は、画像内で**顔と思われる領域を特定**
- **顔の特徴**（目、鼻、口など）を**識別**
- **それらが集まる領域を顔**と判断

顔検出の仕組み

ディープラーニングによる顔検出

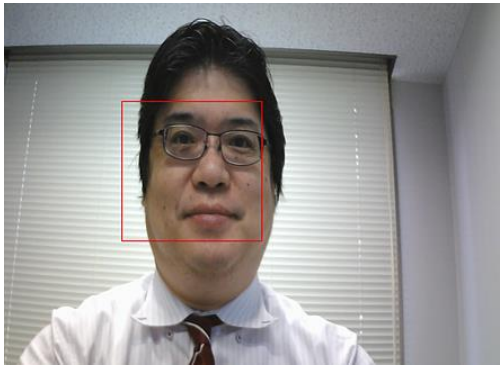
- **ニューラルネットワーク**を使用
- 畳み込みニューラルネットワーク（CNN）が広く用いられている
- **大量の画像データからの学習**
- 学習後，新しい画像に適用して顔検出を実行

顔検出での訓練データ

データ
（入力）

AI

顔検出結果



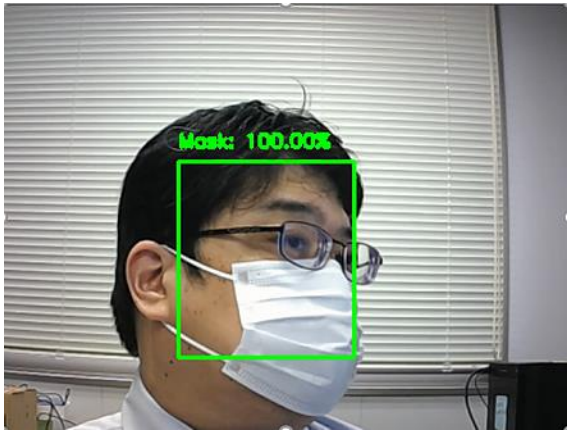
訓練データ

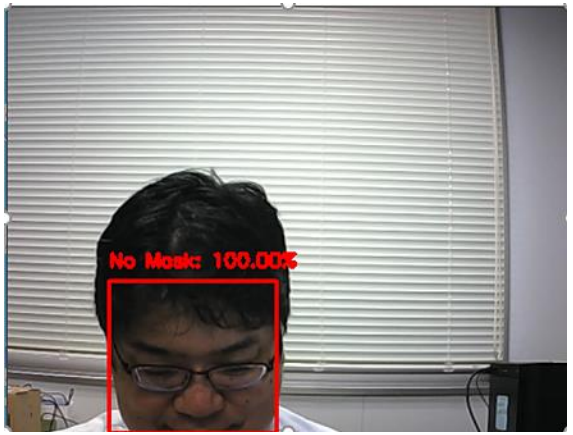
顔画像の例

訓練データの構成
顔画像と
バウンディングボックス

顔検出の課題

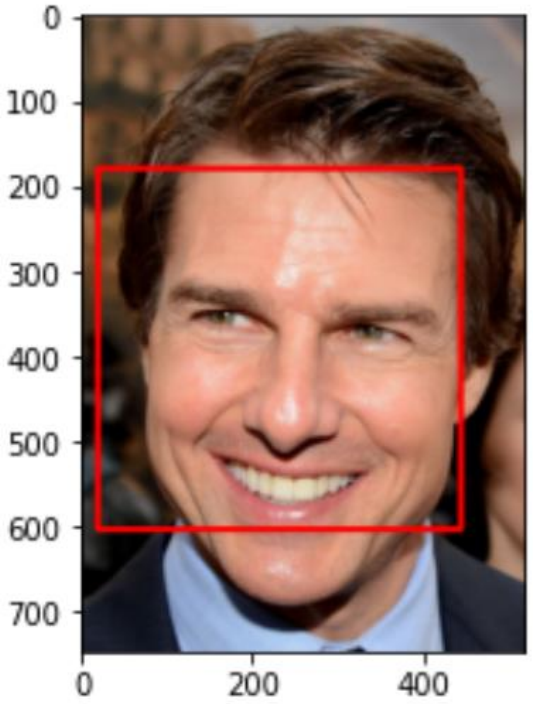
- 照明のバリエーション
- さまざまな顔の向き
- 顔の個性
- 表情
- 付属物（例：メガネ、帽子）



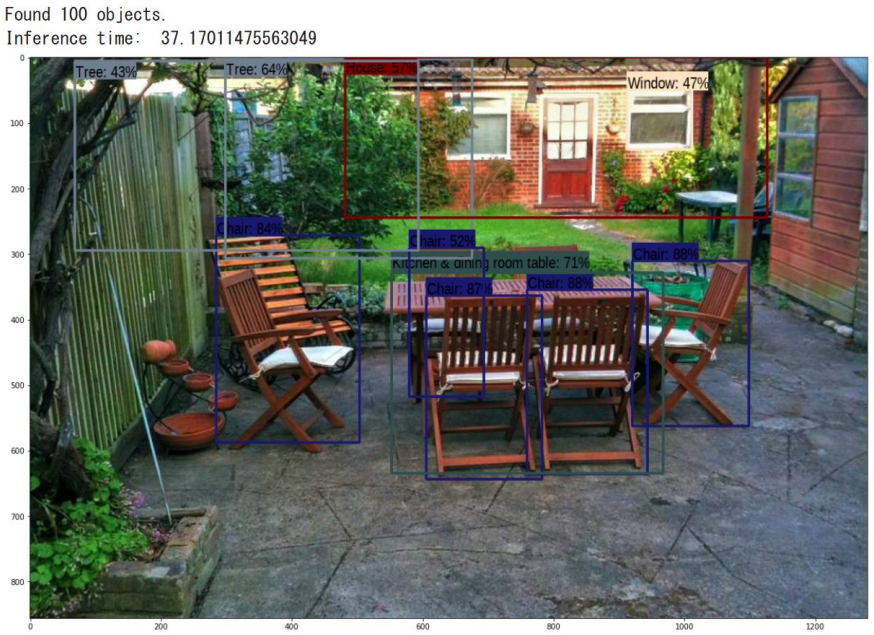


進展：ディープラーニングにより，従来困難な場合（例：マスクあり顔）でも高い精度で検出可能

顔検出と物体検出の比較



顔検出



物体検出

視覚的な違い

- 物体検出：様々な物体の検出
- 顔検出：顔に特化した検出

顔検出と物体検出

顔検出	物体検出
顔のみの検出	さまざまな種類の物体を検出
顔の位置とサイズを識別	物体の種類、位置、サイズを識別
特徴抽出の対象は、顔ランドマーク	特徴抽出の対象は、形状、色、テクスチャなど
代表手法: RetinaFace, MTCNN	代表手法：YOLO

顔検出まとめ

- 顔検出
  画像内から顔と思われる領域を特定し，目・鼻・口などの顔の特徴を識別して，それらが集まる領域を顔として判断する技術
- 顔検出の課題
  照明条件の変化，顔の向き，表情の変化，メガネや帽子などの付属物への対応が必要である．
  ディープラーニングの活用により，これらの課題に対する解決策が進展している．

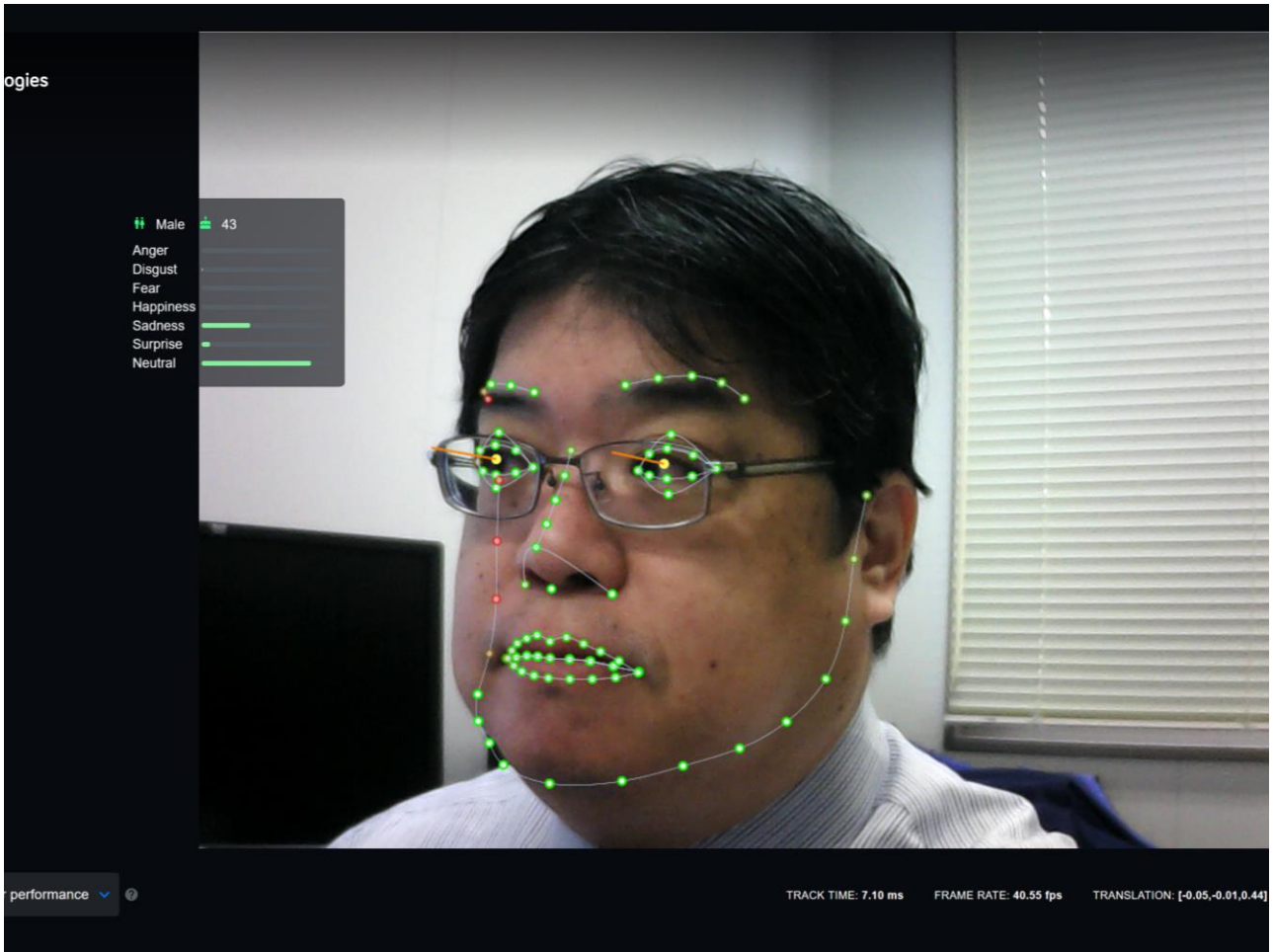
10-4. 顔ランドマーク

顔ランドマークのデモ

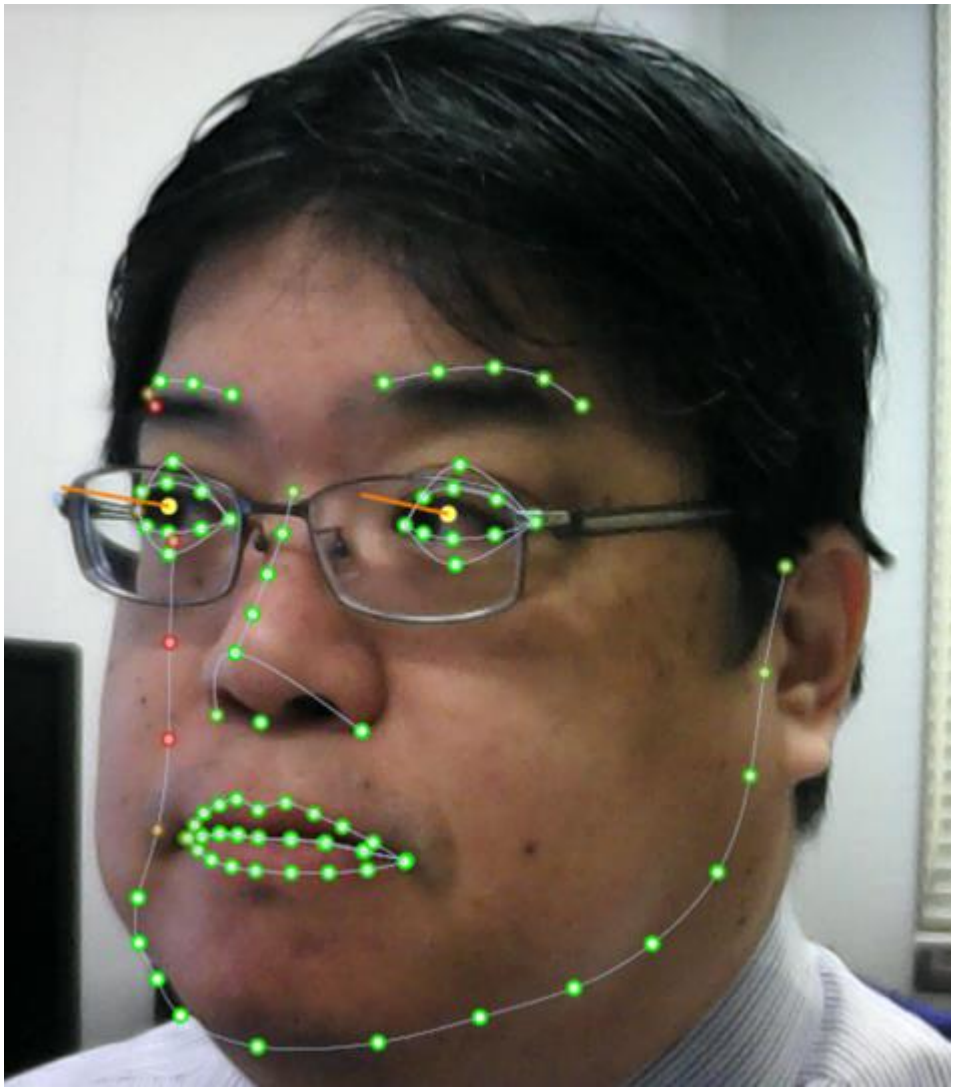
パソコン（カメラ付き）で試すことが可能

URL: https://visagetechnologies.com/demo/

機能：顔ランドマーク検出，表情，リアルタイム処理



顔ランドマークの基本



顔ランドマーク

顔の重要な特徴点（目，鼻，口など）の位置を示すポイント

【用途】

- 顔の表情解析
- 顔の向き推定
- 顔識別
- その他の分析

顔ランドマークの検出の歴史

重要な進展：

- 2013年に畳み込みニューラルネットワークによる手法が登場
- Yi Sun, Xiaogang Wang, Xiaoou Tangによる研究
- 高精度な検出を実現

Figure 1: Examples of facial point detection. First row: ini-

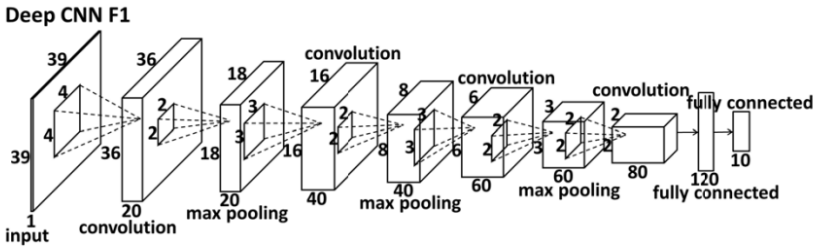


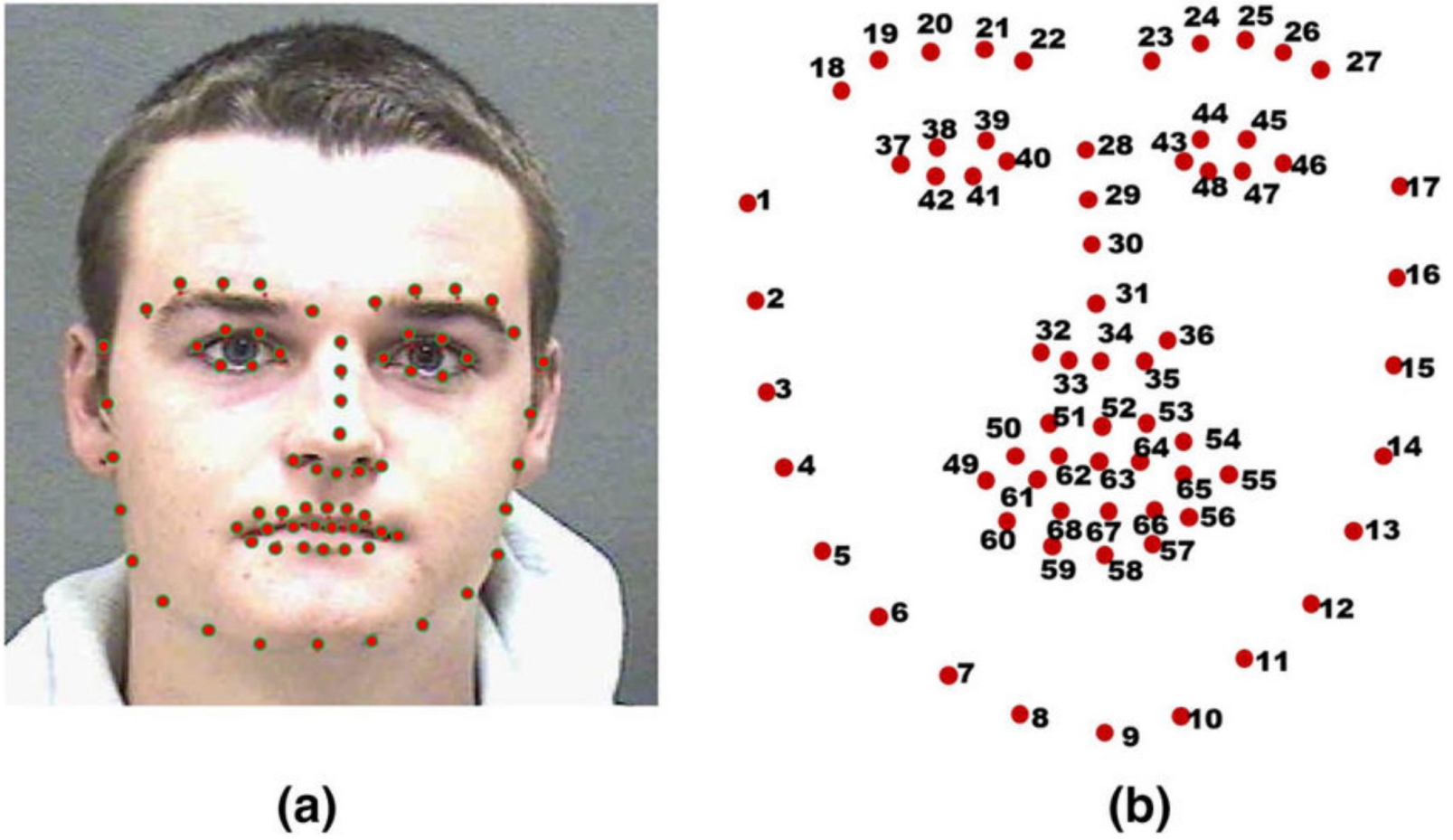
Figure 3: The structure of deep convolutional network F1.

畳み込みニューラルネットワーク
を使用

Yi Sun, Xiaogang Wang, Xiaoou Tang
Deep Convolutional Network Cascade for Facial Point Detection, 2013

６８ランドマーク方式

顔の 68個のランドマーク．目，眉毛，鼻，口，顔の輪郭など．
詳細な顔分析が可能．



Identification of facial landmarks using Dlib. a Facial landmarks. b The position and order of 68 points on the face

出典 https://www.researchgate.net/figure/dentification-of-facial-landmarks-using-Dlib-a-Facial-landmarks-b-The-position-and_fig2_343699139

顔ランドマークまとめ

- 顔ランドマーク
  顔検出における特徴点のことで，目・鼻・口などの位置を示す点群である．これらの特徴点の配置パターンを分析することで，顔の向きや表情の変化も検出できる．

10-5. ディープラーニングによる顔認識

顔認識

顔認識は、画像または動画から人の顔を認識

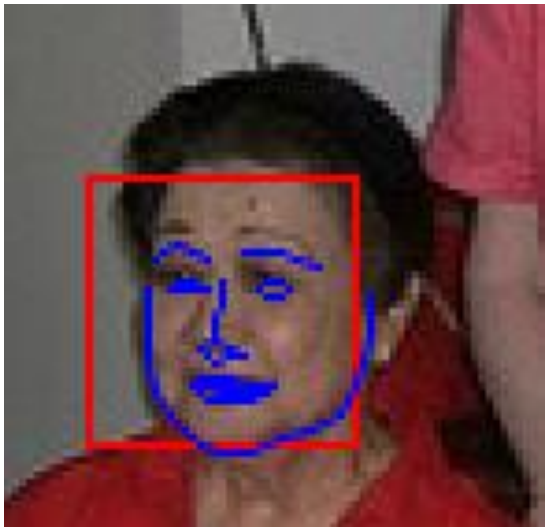
比較

データベース

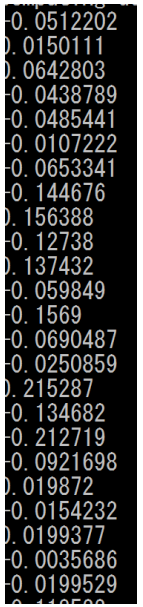
データベース内の顔との比較により、人物を特定

顔の数値化プロセス

- **顔の数値化**は，**複数の数値**（ふつう１００以上）の組み合わせ
- **顔画像**から，**顔ランドマーク**を検出
- 顔ランドマークから，顔を数値化する処理を実行



顔検出，
顔ランドマーク



数値化

同一人物判定プロセス

手順：

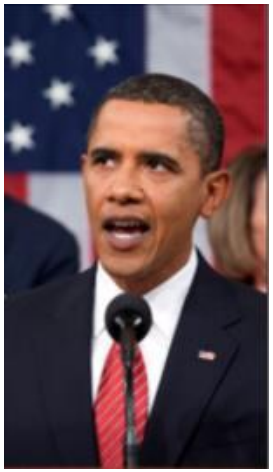
1.2つの別の写真あるいは動画を入力

2.顔の数値化を実行

3.比較（距離計算）を実施

4.判定結果を出力： 同一人物である，同一人物でない

顔の数値 **比較（距離計算）** **顔の数値**



同一人物である

または

同一人物でない

顔認識システムの構造

処理の流れ

- 精度向上のため，同一人物の多数の写真と比較
- 顔の数値化処理
- 距離計算による比較
- 本人判定の実行

顔の数値

比較
（距離計算）

顔の数値

顔の数値

顔の数値

本人である
または
本人でない

距離学習の基本

- **距離学習**は，ディープラーニングでの学習で，**距離に関する学習を行うこと.**
- 顔認識で使用される重要技術

目的：同一人物の顔を近く、異なる人物の顔を遠くなるように学習

仕組み：畳み込みニューラルネットワーク

顔認識まとめ

- 顔認識技術
  画像や動画から人の顔を検出し，データベース内の顔情報と照合して人物を特定する技術である.
- 顔の数値化処理
  顔画像から特徴点（顔ランドマーク）を検出し，それらの位置関係や形状を100以上の数値の組み合わせとして表現する処理である.
- 同一人物判定プロセス
  2つの異なる写真や動画から抽出した顔の数値データを比較し，その類似度から同一人物であるかを判定する処理である.
- 距離学習：ディープラーニングを用いて，同一人物の顔は近く，異なる人物の顔は遠くなるように特徴空間を学習する技術である.

演習２．顔認識

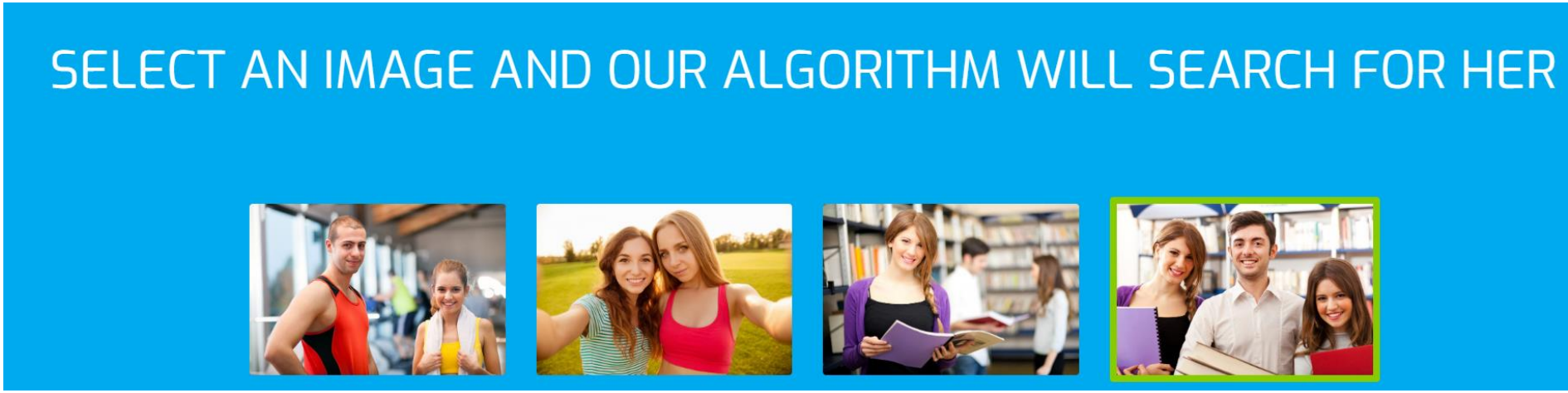
オンラインデモによる

顔認識の体験

① sky biometry の顔認識デモページ

https://skybiometry.com/demo/face-recognition-demo/

② 顔画像を選択．結果を確認



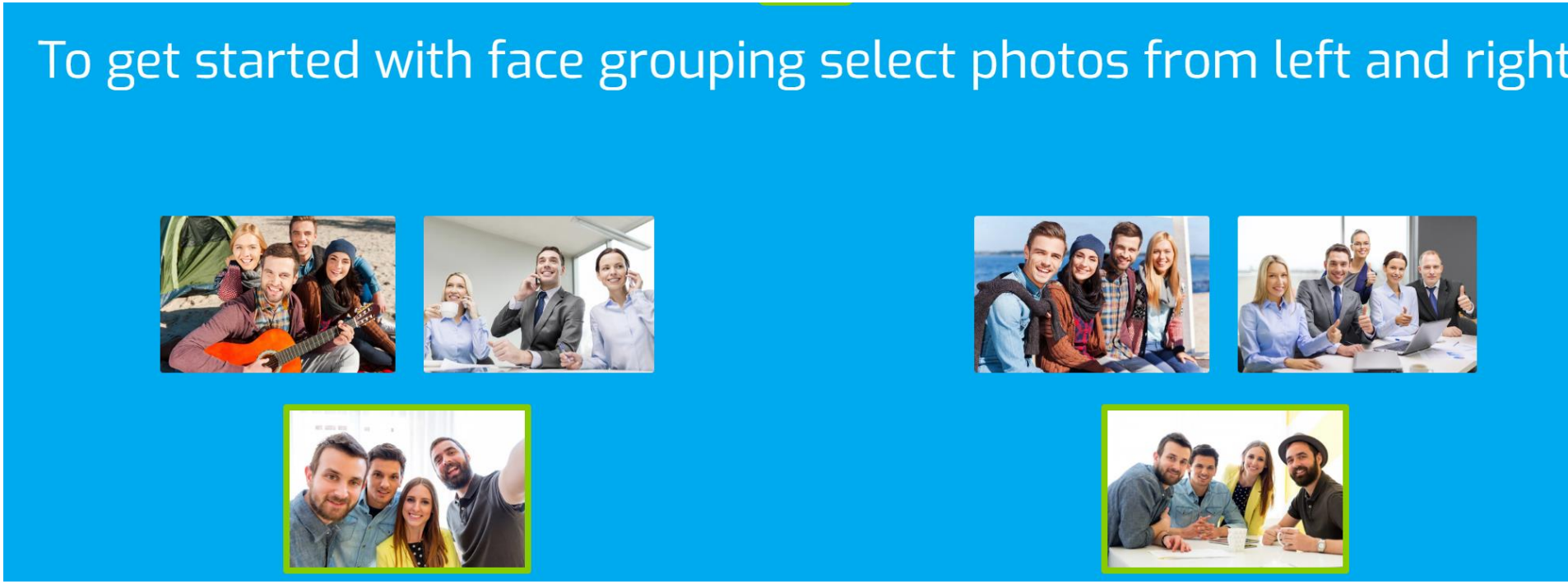
演習 3

顔を手掛かりに，同一グループ化の判定

① sky biometry の face grouping デモページ

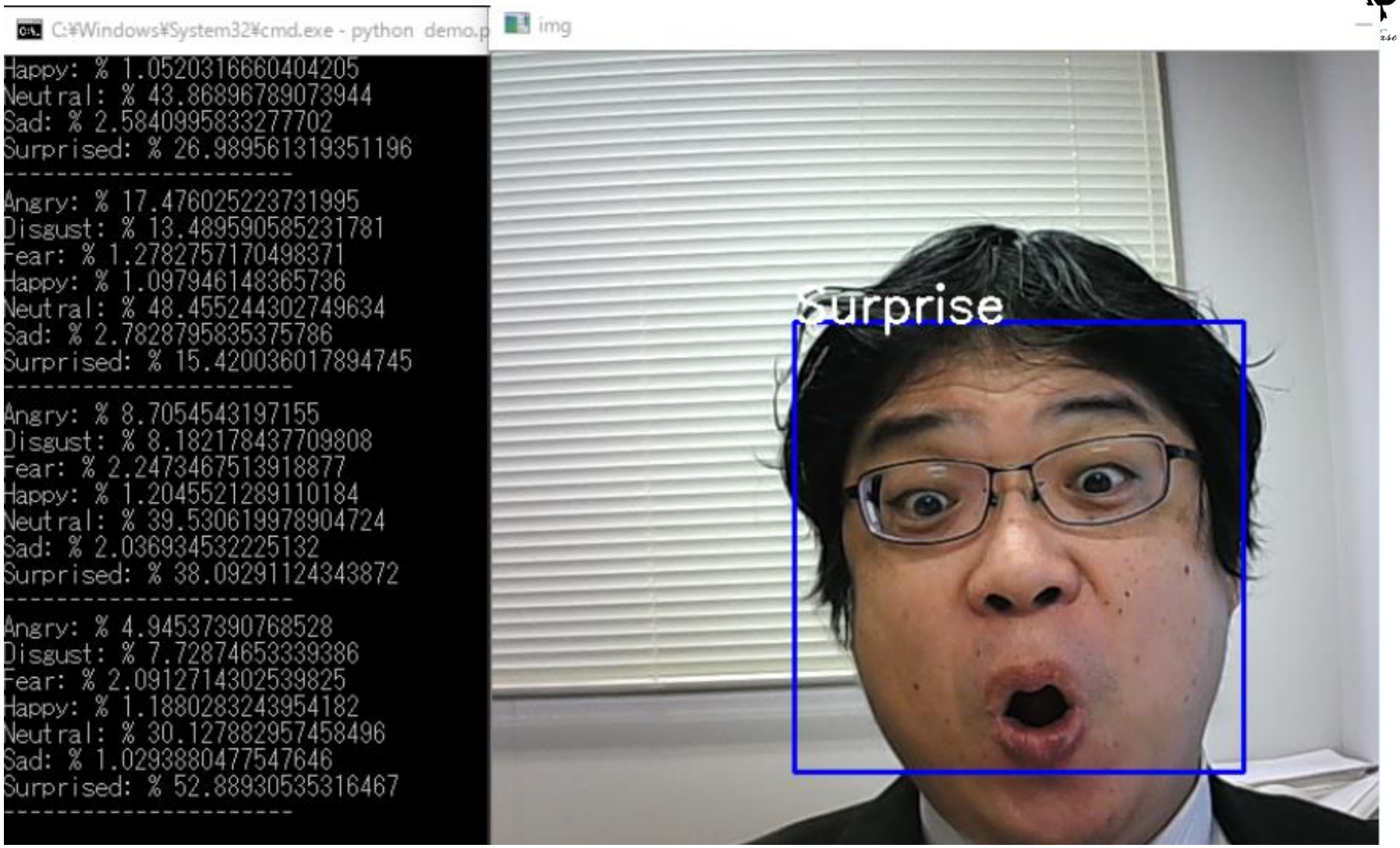
https://skybiometry.com/demo/face-grouping/

② 右から１つ，左から１つ画像を選択．結果を確認



10-5. ディープラーニングによる感情認識その他

感情認識システム

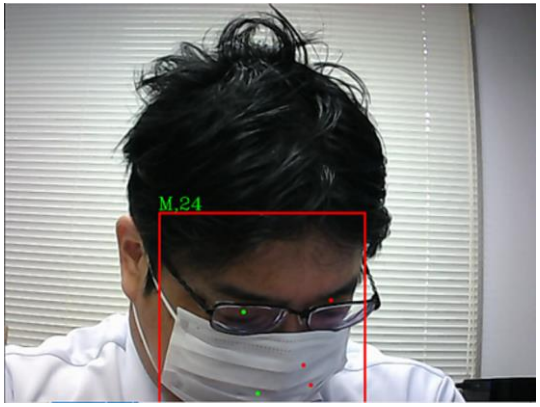


- ここでは，**７種の表情** Angry, Disgust, Fear, Happy, Neutral, Sad, Surprised の**それぞれの確率**を推定
- https://github.com/ezgiakcora/Facial-Expression-Keras で公開されている成果物

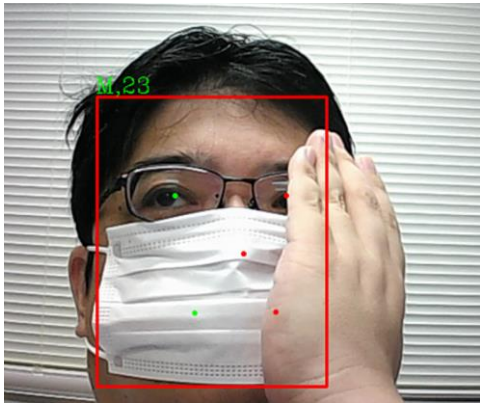
感情認識システム

- 7種の表情認識
  - Angry（怒り）
  - Disgust（嫌悪）
  - Fear（恐怖）
  - Happy（幸福）
  - Neutral（中立）
  - Sad（悲しみ）
  - Surprised（驚き）
- ディープラーニングの活用
- 顔ランドマークの使用

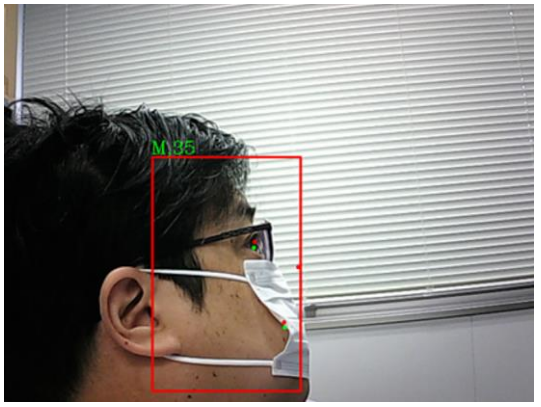
年齢推定，性別推定の実例



推定結果：24歳男性



推定結果：23歳男性



推定結果：35歳男性

InsightFace による処理結果

【関連情報】

https://www.kkaneko.jp/ai/win/insightface.html
https://colab.research.google.com/drive/1S55yEFiQpdIRdjWbdH0zzEYD5VAfklHd?usp=sharing

表情推定アプリ

各自、ソースコードと実行結果を確認しよう

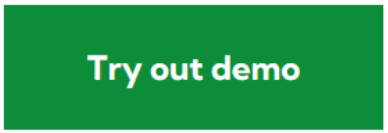
https://colab.research.google.com/drive/1t8YlRamyYij40wND--k1NHTDR5fMXvHq?usp=sharing

発展演習３

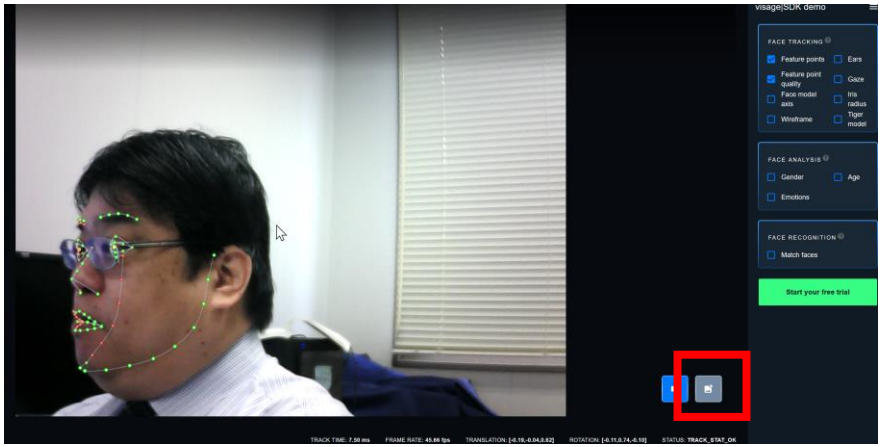
- まず、カメラ付きのパソコンを準備する
- 次のページを開く。Visage Technologies 社のデモページ

https://visagetechnologies.com/demo/

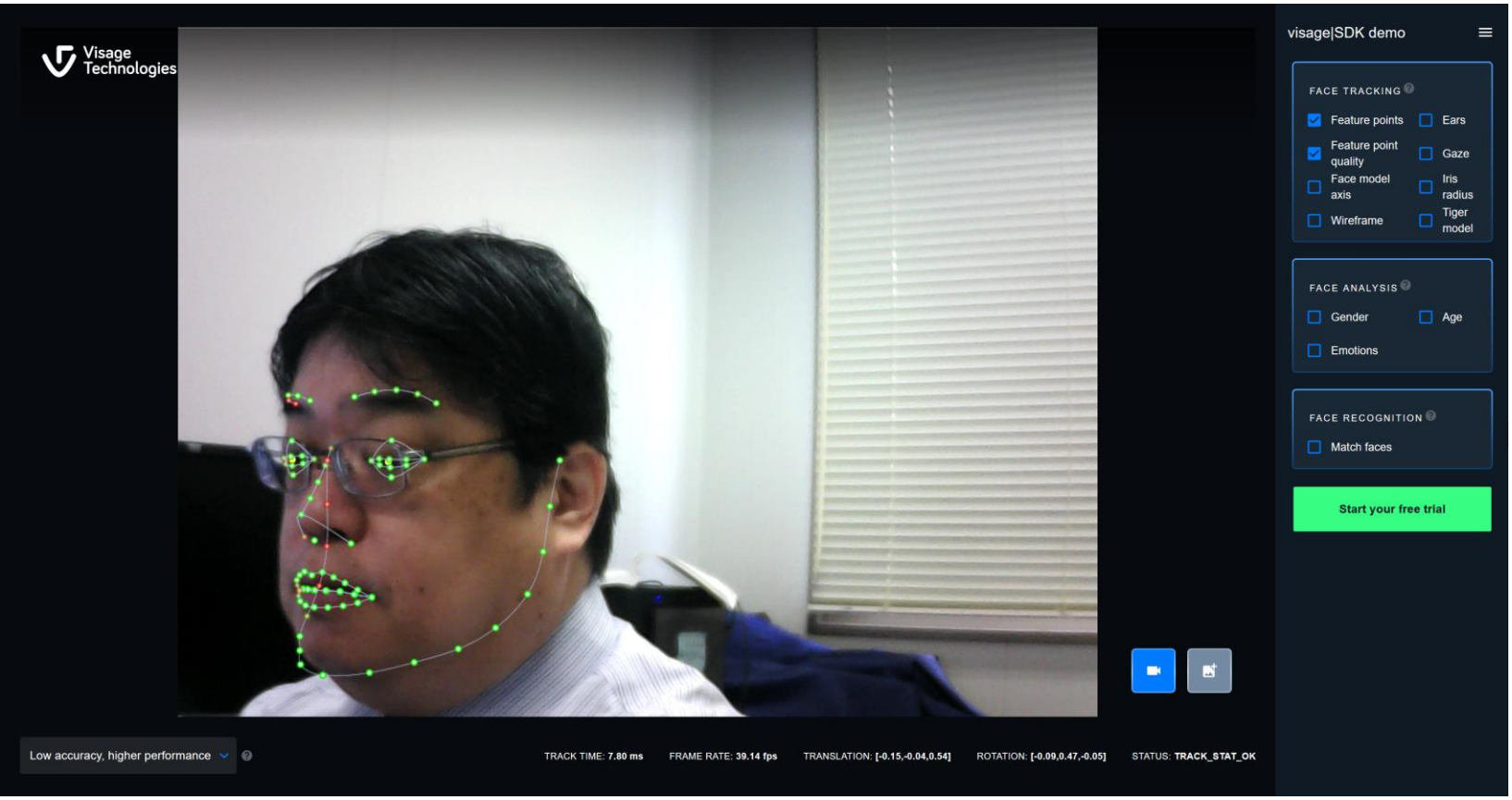
- 「**Try out demo**」をクリック



- **画像をアップロード**するので、**右下のボタン**をクリックして、画像をアップロード。（パソコンのカメラにも対応）



さまざまな設定



Gaze
視線

Gender
性別

Emotions
感情

Age
年齢

全体まとめ

- **顔検出**
  画像内から顔と思われる領域を特定し，目・鼻・口などの顔の特徴を識別して，それらが集まる領域を顔として判断する技術である（物体検出の一環として顔を特定する基本技術）
- **顔ランドマーク**
  顔の重要な特徴点（目，鼻，口など）の位置を示すポイントで，顔の表情解析や向き推定，顔識別などの分析に活用される
- **感情認識**
  顔表情から感情状態を推定する技術で，Angry（怒り），Disgust（嫌悪），Fear（恐怖），Happy（幸福），Neutral（中立），Sad（悲しみ），Surprised（驚き）の7種類の感情を分類する
- **顔認識**
  画像や動画から人の顔を検出し，データベース内の顔情報と照合して人物を特定する技術である.