

## 4. 画像認識による非接触操作できるアプリの作成

扇田 楊 , 濱岡 伯

指導教員 ソソラバラム バドゥジャルガル

### 1. はじめに

一昨年から COVID-19 が猛威を振るっており、その影響で人や物に直接接触する行為が懸念されている。

そのため、パソコンに直接触れずに手の動きだけで簡単な操作ができれば感染対策になるだけでなく操作性をもっと快適にできるのではないかと考え、調べたところ Python のライブラリを用いてパソコンの備え付けカメラで手の動きをトラッキング(追跡)するコンピュータビジョン(CV)のプロジェクトを紹介した記事を見つけた。

そこで、この記事を参考にハンドトラッキングによる音量調節、マウス操作、文字入力といったプロジェクトを実装して、実際に利用したいと考え本研究テーマを選定した。

### 2. 研究概要

本研究では、OpenCV, MediaPipe などの Python ライブラリについて学習しながら、開発環境を構築し、ハンドトラッキングによるプロジェクトを実装し、動作確認を行った。開発環境は表 1. の通りである。

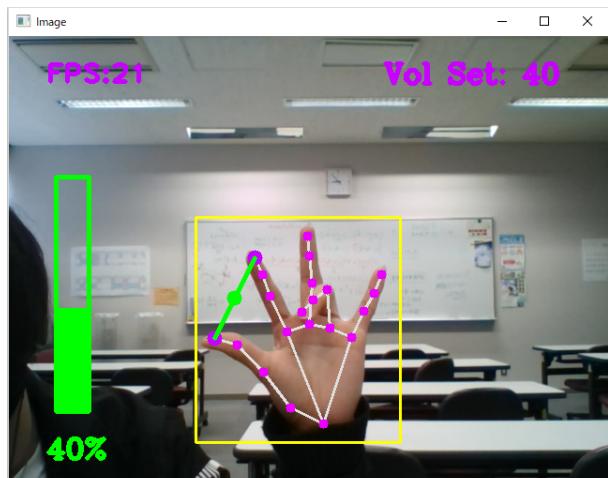


図 1. 音量調節している様子

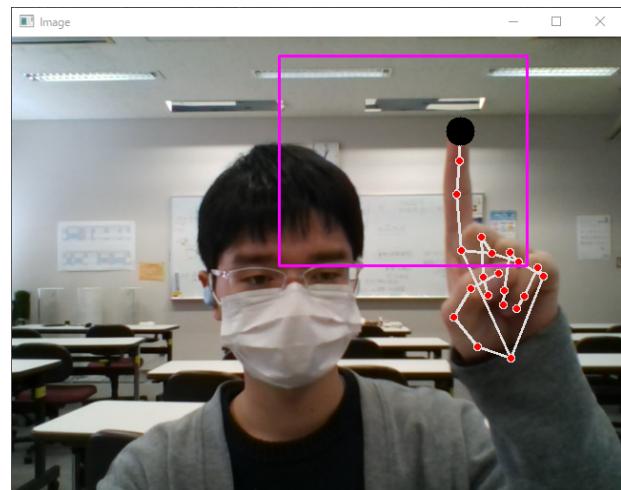


図 2. マウスカーソルを操作している様子



図 3. 文字入力している様子

OS	Windows10 Pro
使用言語	Python3.8
ライブラリ	OpenCV, MediaPipe, PyCaw PyAutoGUI, AutoPy

表 1. 開発環境

## 2.2 ライブライについて

- OpenCV とは画像や動画の処理に特化したコンピュータービジョンのライブラリ。
- MediaPipe とは顔や手の状態を推論する ML(機械学習)ソリューションを Python でカスタマイズ可能にしたパッケージ。
- PyCaw とはオーディオユーティリティに接続したり、音量を変更したりするためのライブラリ。
- PyAutoGUI ,AutoPy とは GUI 操作を簡単に行えるモジュール。

## 3. 手の検出について

- Hand Land Mark について

MediaPipe では、手の状態を取得するため、以下の図のように指先やその間接などに 21 個のキーポイントを設定する。この手のキーポイントのことを Hand Land Mark という。

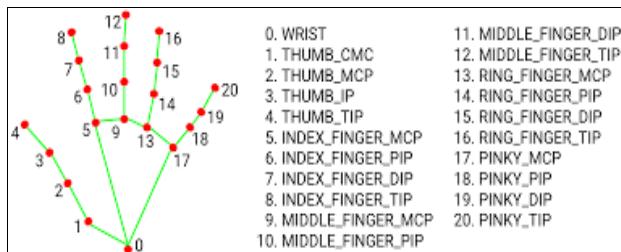


図 4. Hand Land Mark

- Hand Tracking Module について

このモジュールは主にカメラに映った手を認識する findHands(図 5.) 認識した手に合わせて Hand Land Mark を描画する findPosition 関数(図 6.) からなる。

```
def findHands(self, img, draw=True):
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    self.results = self.hands.process(imgRGB)
    #print(results.multi_hand_landmarks)

    if self.results.multi_hand_landmarks:
        for handLms in self.results.multi_hand_landmarks:
            if draw:
                self.mpDraw.draw_landmarks(img, handLms,
                                           self.mpHands.HAND_CONNECTIONS)

    return img
```

図 5. 手を認識する findHands 関数

```
def findPosition(self, img, handNo=0, draw=True):
    lmList=[]
    if self.results.multi_hand_landmarks:
        myHand = self.results.multi_hand_landmarks[handNo]
        for id, lm in enumerate(myHand.landmark):
            # print(id, lm)
            h, w, c = int(lm.x * w), int(lm.y * h)
            cx, cy = int(lm.x * w), int(lm.y * h)
            # print(id, cx, cy)
            lmList.append([id, cx, cy])
            if draw:
                cv2.circle(img, (cx, cy), 15, (255, 0, 235), cv2.FILLED)

    return lmList
```

図 6. Land Mark を描画する findPosition 関数

## 4. おわりに

本研究を通じて OpenCV や MediaPipe の知識を身につけ、ハンドトラッキングによるプロジェクトを実装し、実際に自分たちで利用してみた。

MediaPipe を用いて、手を認識させるモジュールについての理解と、開発環境の構築に時間がかかった。また、私たちにとってコンピュータービジョンを用いたプロジェクトは新らしいチャレンジであったため、常に手さぐりの研究で大変だったがやりがいはとてもあった。

今後は、産技短展までにアプリを完成させたいと考えている。

## 5. 参考文献

- [1] OpenCV ( Python ) と MediaPipe を使用して作成されたコンピュータビジョンプロジェクト

<https://www.wenyanet.com/opensource/ja/6185392c247897743e284a96.html>

- [2] Python パッケージ版の MediaPipe が超お手軽 + 簡易な MLP で指ジェスチャー推定

<https://qiita.com/Kazuhito/items/222999f134b3b27418cd>

- [3] MediaPipe の説明

<https://google.github.io/mediapipe/solutions/hands.html>

- [4] Python でマウスやキーボードを操作できる PyAutoGUI による自動操作マニュアル

<https://hashikake.com/PyAutoGUI>

- [5] Python で GUI 操作をしてみよう! AutoPy 編 - 白猫学生のブログ

<https://whitecat-student.hatenablog.com/entry/2016/10/24/125000>