

FukuMeetの開発

情報技術科2年 及川結加 齋藤龍聖 米澤悠貴

指導教員 高橋強

研究の背景と目的

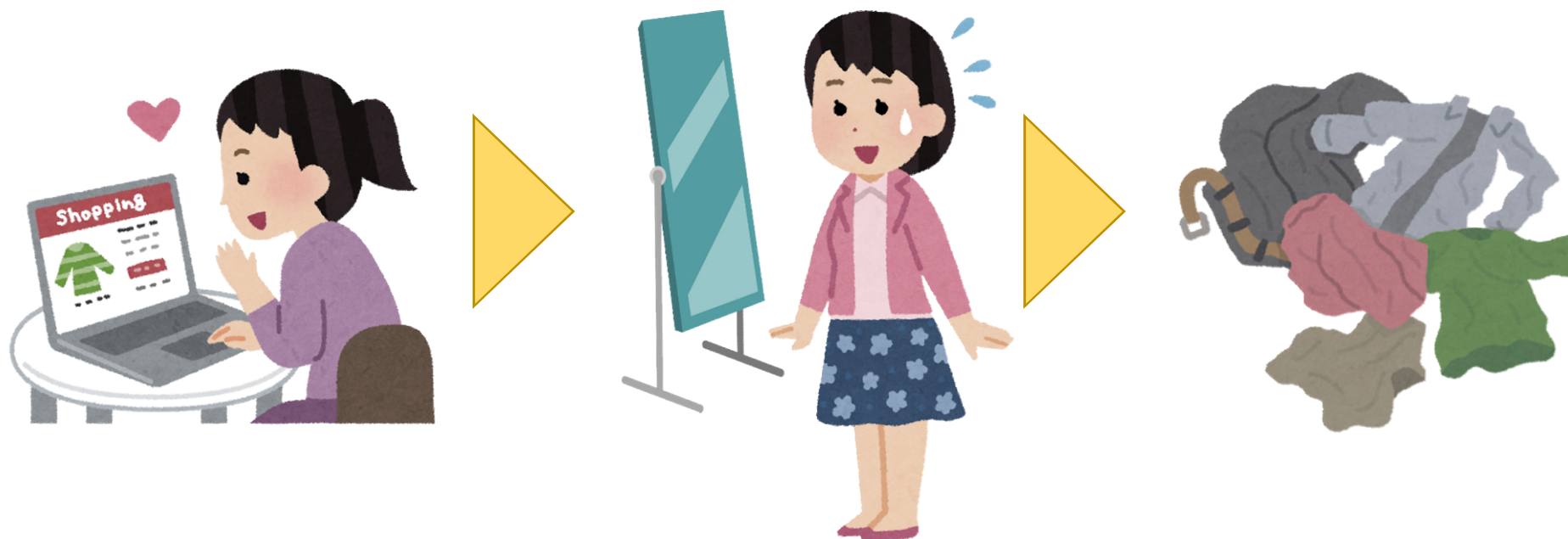
背景：オンラインショップで洋服購入時のイメージ違いを防ぐ
ユーザが画面の中で試着イメージを確認できるシステム
「FukuMeet」を開発したいと考えた。

目的：授業で学んだ知識を応用しながら、新たな技術を学び、
開発力や**問題解決力**を高める。メンバーや指導教員と
協力し、コミュニケーション力やプレゼンテーション
能力を向上させる。

使用技術：Python、HTML、JavaScriptを活用し、OpenCV
Mediapipe、Flaskなどのライブラリを使用する。

FukuMeet

イメージと違う…



FukuMeet



システムの操作方法

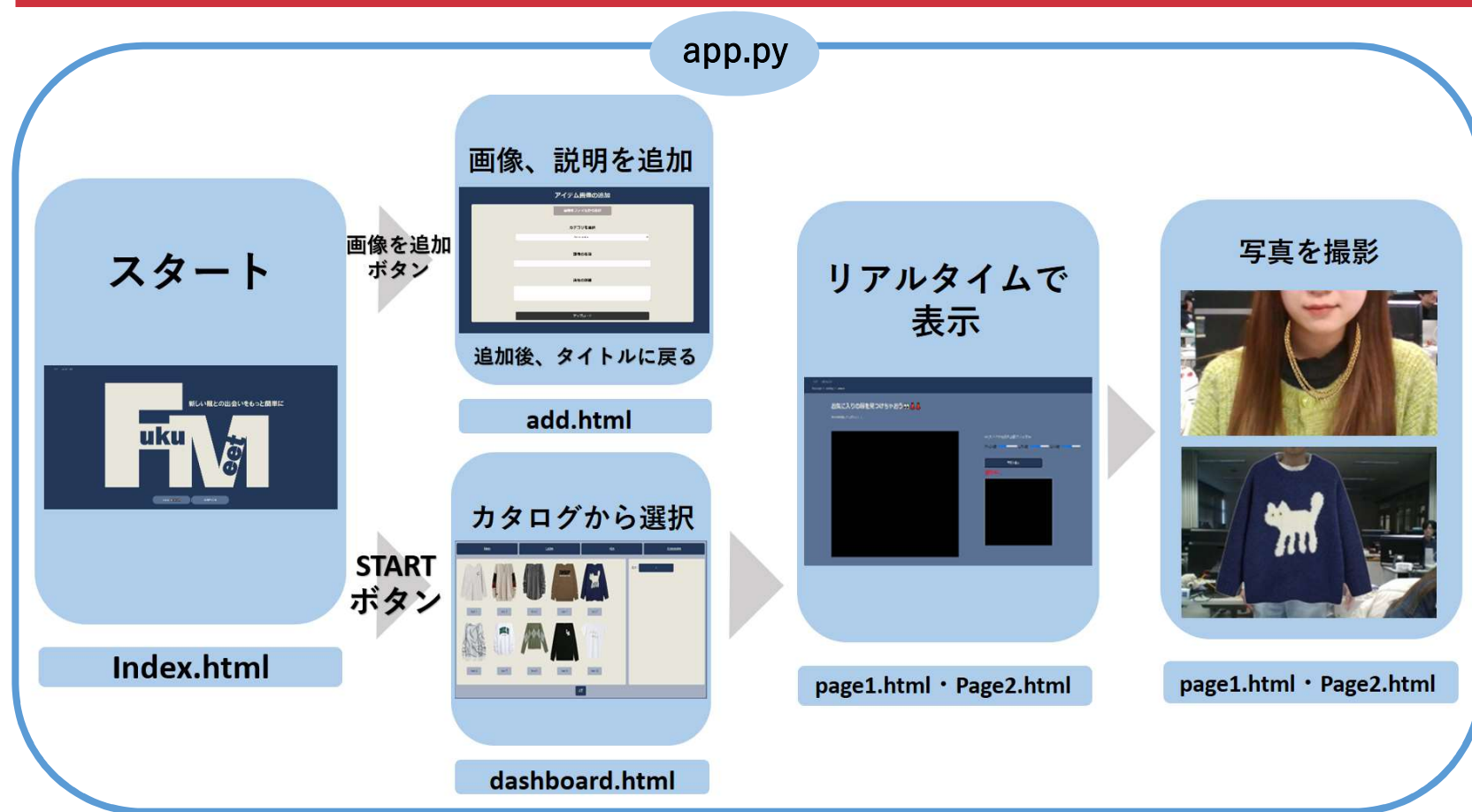


図1：システムの流れ

システムの操作方法



「画像を追加」を押した場合

システムの操作方法



「START」を押した場合

新規カタログの追加

```
"Mens": [  
  {  
    "image": "static/images/Mens/mens1.png",  
    "name": "Item 1",  
    "details": "サイズ: M, 素材: 綿, 価格: 3000円, カラー:  
  },  
  {  
    "image": "static/images/Mens/mens2.png",  
    "name": "Item 2",  
    "details": "サイズ: L, 素材: ポリエステル, 価格: 350  
  },  
  {  
    "image": "static/images/Mens/T_transparent.png",  
    "name": "shiro",  
    "details": "白"  
  }  
]
```

画像,

る

カメラの起動

「決定」ボタンをクリック

アクセサリの場合



Page1.html

服の場合



Page2.html

各ページに移動後、カメラが起動される

服とアクセサリーのモニター表示



服は肩と腰の中心を計算して肩から腰の範囲で表示

服とアクセサリーのモニター表示



アクセサリーは両肩から**肩の中心値**を求めて首元に表示

服とアクセサリーのモニター表示



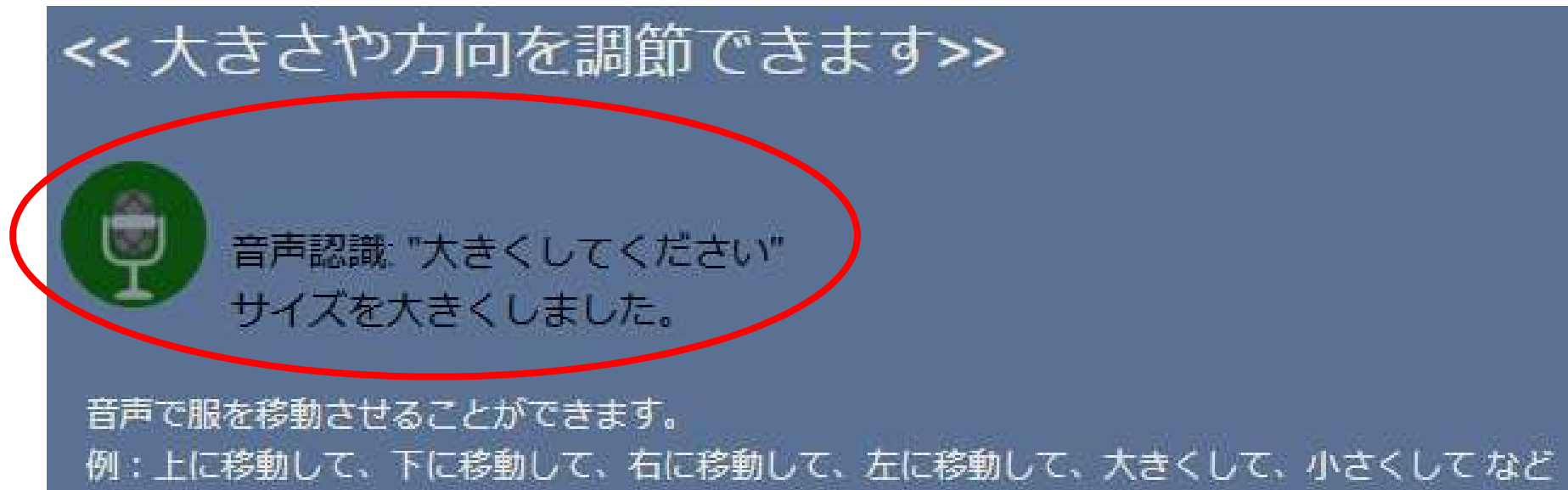
横を向くと**画像の横幅**が自動で調整される

画像の位置や大きさを調整するスライダー



ブラウザが**スライダーの値**を送信し、
サーバーが画像を調整後、値を返して反映する

音声入力による位置と大きさの調整



マイクの画像をクリックすると**10秒間**音声認識が可能になり、
「**大きく、小さく、上、下、右、左**」を**テキスト**として認識

開発環境

表1 開発環境

OS	Windows10
使用言語	Python3.12.4
ライブラリ	Flask3.0.3関連ライブラリ OpenCV(CV2) MediaPipe Math rembg
ソフトウェア	Visual Studio Code

開発環境の構成イメージ

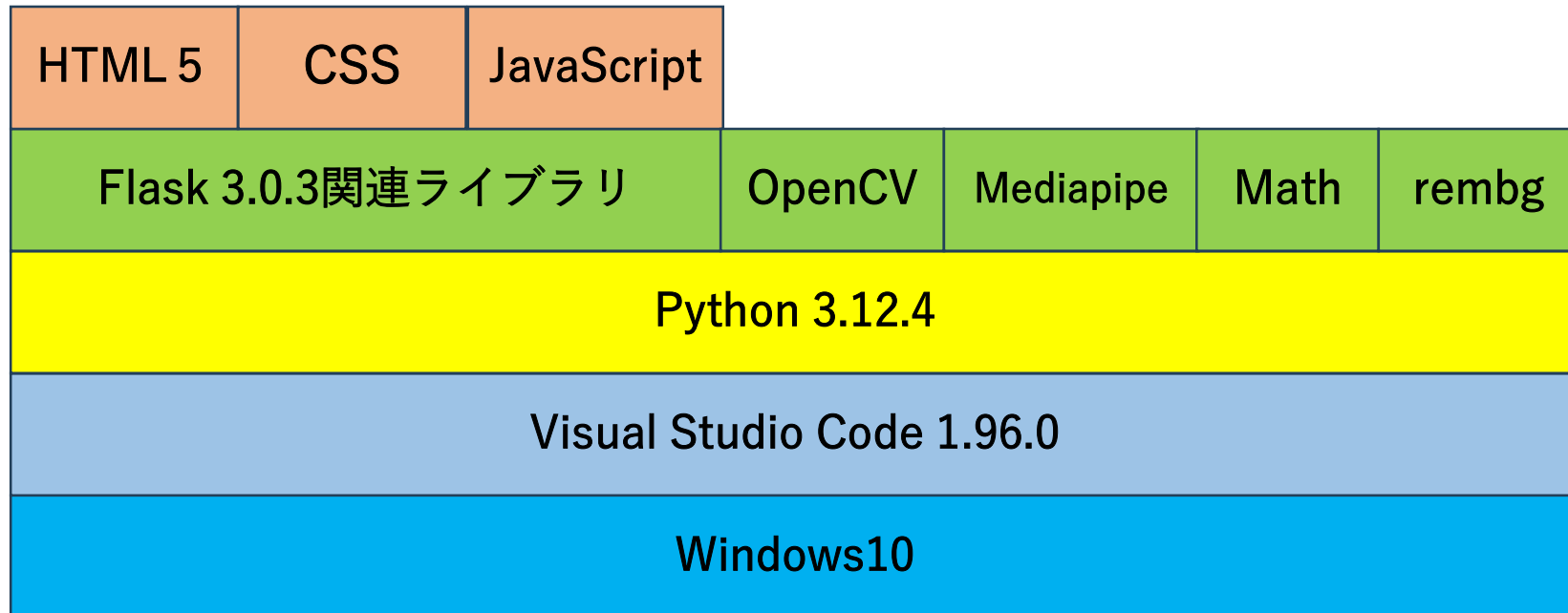


図2：開発環境イメージ



言語・ライブラリ



1 *Python*

画像認識やポーズ推定、リアルタイム処理に最適な選択肢であり、多くのフレームワークが利用可能

2 *OpenCV (cv2)*

画像やビデオの解析、認識、変換を行うオープンソースライブラリ

< 本研究の活用例 >

画像の左右反転、変換、画像の重ね合わせ、
動画の撮影・保存

3 *MediaPipe* MediaPipe

リアルタイムで**ポーズ推定**、**顔認識**、**動きの追跡**、**物体検出**が可能な機械学習ライブラリ

<本研究の活用例>

ポーズ推定を活用し、両肩や両腰の位置を特定

4 *Rembg*

高い精度で画像の**背景を除去**するライブラリ

<本研究の活用例>

追加された画像の**背景除去**を行う

6 *Flask*



アプリケーションを作成する軽量なWebフレームワークであり、下記の関連ライブラリによって様々な処理を行っている。

<関連ライブラリ>

Response, render_template, request, jsonify, send_from_directory

<本研究の活用例>

Responseは動的な映像配信、render_templateはHTMLテンプレートをブラウザに表示、requestで画像追加フォームのデータの取得、send_from_directoryではファイルの送信を行っている。

今後追加したい機能

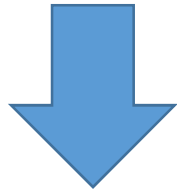


服とアクセサリーを重ねて表示

今後追加したい機能

眼鏡カテゴリ

追加



dashboard.html

表示



眼鏡などのカテゴリの増加

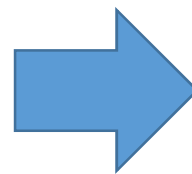
今後追加したい機能



背景画像



人物



好きな背景画像の挿入

終わり

- メンバーや指導教員と協力して進め、意見交換を行うことがとても大切だと学んだ。
- 開発中に発生した課題をインターネットで調査し、試行錯誤を繰り返すことで問題解決力を高めた。
- リアルタイム試着の精度向上や直感的な操作性を改善し、産技短展での展示に向けてシステムやユーザー体験の向上を目指す。

参考リンク

- [1] HTML要素リファレンス
<https://developer.mozilla.org/ja/docs/Web/HTML/Element>
- [2] CSS要素リファレンス
<https://developer.mozilla.org/ja/docs/Web/CSS/Reference>
- [3] JavaScript リファレンス
<https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference>
- [4] python ドキュメント
<https://docs.python.org/ja/3/>
- [5] OpenCV Python Tutorials
https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
- [6] cv2.VideoCaptureで動画ファイル・Webカメラ映像の読み込みと再生
<https://python.joho.info/opencv/opencv-videocapture-mp4-movie-py/#toc1>
- [7] Python、OpenCVで画像ファイルの読み込み、保存
<https://note.nkmk.me/python-opencv-imread-imwrite/>
- [8] MediaPipe ソリューション ガイド
<https://ai.google.dev/edge/mediapipe/solutions/guide?hl=ja>
- [9] 姿勢ランドマーク、検出ガイド
<https://medium.com/@tayyabjavedbrw789/pose-detection-using-mediapipe-solutions-dabmove-detection-4c4e39080142>
- [10] Flask Tutorials
<https://realpython.com/tutorials/flask/>
- [11] Flaskで画像とファイルを効率的にアップロードする方法を徹底解説
https://ittrip.xyz/python/flask-file-image-upload#index_id11
- [12] Flaskで画像ファイルをアップロード
<https://qiita.com/keimoriyama/items/7c935c91e95d857714fb>
- [13] PythonでいろいろPOSTしてFlaskで受け取る
<https://qiita.com/tamyu/items/54db82b1c30a3966d8a1>
- [14] 背景除去する Background-remove
<https://qiita.com/kotai2003/items/2cddf1b3e17c728439b0>
- [15] ウェブ音声APIの使用-Web API |MDN
https://developer.mozilla.org/ja/docs/Web/API/Web_Speech_API/Using_the_Web_Speech_API
- [16] Webページでブラウザの音声認識機能を使おう
https://qiita.com/hmmrjn/items/4b77a86030ed0071f548?utm_source=chatgpt.com
- [17] WebSpeechAPIを利用してWebブラウザで音声認識を行う
<https://zenn.dev/micronn/articles/b654ceca1bdf13>

FukuMeetの開発

情報技術科2年 及川結加 齋藤龍聖 米澤悠貴

指導教員 高橋強

Jsonファイル

Javascriptというプログラミング言語の書き方を参考に作られたデータの保存・送受信するためのフォーマットのこと。

「HukuMeetでの活用」

productData.json

- ✓商品情報の保存・管理
- ✓ FlaskとJavaScript間でデータをやり取り

Jsonファイル

Javascriptというプログラミング言語の書き方を参考に作られたデータの保存・送受信するためのフォーマットのこと。

「HukuMeetでの活用」

productData.json

- ✓商品情報の保存・管理
- ✓ FlaskとJavaScript間でデータをやり取り

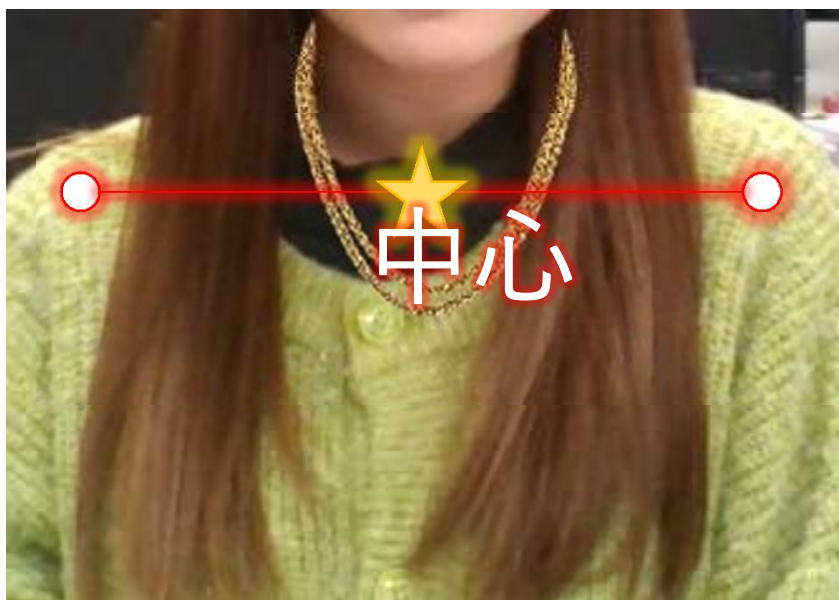
FukuMeet: 将来的なAI活用計画

- 将来的にはAIによる**体型認識技術**を導入し、ユーザーの身長や体格に応じた自動サイズ調整を実現したいと考えている。

体型認識技術とは？

体型認識技術（Body Measurement & Recognition Technology）とは、カメラやセンサーを用いて人の体型・姿勢・寸法を自動で測定・解析する技術です。AIやコンピュータビジョンを活用し、衣服のフィット感をシミュレーションしたり、サイズの推奨を行うことが可能になります。

服やアクセサリーを 正確に配置するためのアルゴリズム



服の場合：両肩と腰の中心点を計算し、そこから適切な範囲内で表示。
アクセサリーの場合：両肩の座標を取り、中心点を基準に配置。

OpenCV (cv2)

✓ 選定理由：高性能な画像処理・リアルタイム映像解析

- カメラ入力の処理 → Webカメラからの映像をリアルタイムで取得し、表示をスムーズに行う。
- 画像の左右反転 → カメラ映像はデフォルトで反転されるため、ユーザー視点と一致させる処理が必要。
- 試着画像の重ね合わせ → 服やアクセサリーを人物の映像に違和感なくオーバーレイ表示する。

✦ 代替案：Pillow, Scikit-Image

→ しかし、PillowやScikit-Imageは静的画像処理向けで、リアルタイム映像処理には不向きだったため、OpenCVを採用。

Web Speech API

✓ 選定理由：ブラウザだけで動作する音声認識

- 「大きく」「小さく」「右」「左」などの音声コマンドを認識し、試着アイテムの位置やサイズを調整。
- 追加のライブラリ不要 → ブラウザ上で簡単に音声入力进行处理可能。
- Webアプリとの親和性が高い → JavaScriptで直接扱え、リアルタイム操作がしやすい。

✦ 代替案：Google Cloud Speech API, DeepSpeech
→ クラウドAPIはAPIキー管理や通信コストが発生するため、ローカルで動作するWeb Speech APIを選択。

音声認識の比較

技術	動作環境	導入の手軽さ	リアルタイム性	コスト	精度
Web Speech API	Chrome, Edge, Opera	◎ (JavaScriptのみで実装)	◎ (即時応答)	無料	高精度
Google Cloud Speech API	クラウド環境	△ (APIキー管理が必要)	○ (ネット接続が必要)	有料 (課金制)	非常に高精度
Mozilla DeepSpeech	ローカル環境	△ (モデルの事前学習が必要)	○ (オフライン処理可能)	無料	高精度 (事前学習が必要)
SpeechRecognition (Python)	ローカル環境	△ (環境構築が必要)	△ (リアルタイム処理は遅め)	無料	中程度

選定理由

ライブラリ	役割	選定理由	代替案との比較
OpenCV	画像処理・リアルタイム映像解析	軽量でリアルタイム処理が可能	Pillow, Scikit-Image（静的処理向け）
Mediapipe	ポーズ推定（人体ランドマーク検出）	高速・軽量でリアルタイム向け	OpenPose（高精度だが重い）
rembg	背景除去	高精度な背景除去をローカルで実行	OpenCV（精度が低い）、Adobe API（外部依存）
Flask	Webアプリケーション構築	軽量でAPI開発が容易	Django（多機能すぎる）、FastAPI（習熟度考慮）
Web Speech API	音声認識	ブラウザのみで動作し導入が簡単	Google Cloud Speech API（APIキー管理が必要）

1 新規カタログの

入力した商品
Flaskサーバーに
ごとにprojectData

```
{  
  "image": "static/images/Mens",  
  "name": "Item 1",  
  "details": "サイズ: M, 素材: ",  
},  
  "Ladies": [  
    {  
      "image": "static/images/Ladies",  
      "name": "Item 1",  
      "details": "サイズ: S, 素材: ",  
    },  
  ],  
}
```

2 カタログページでの服やアクセサリーの選択・表示

- 選択した画像Flaskサーバーに送信し、返された
画像パスを使用してカメラ映像に表示する。
- 画像データはprojectData.jsonに格納される

5 服とアクセサリーのモニター表示

- ・服はMediapipeで両肩と両腰の位置を取得して、**肩と腰の中心値**を求め、首から腰の範囲で表示する。アクセサリーは両肩から**肩の中心値**を求めて首元に表示をする。
- ・表示中は横を向くと、**画像の横幅**が自動で調整される

6 画像の位置や大きさを調整するスライダー

スライダーの**値**をブラウザが取得してサーバーに送信する。サーバーは画像の調整処理後、ブラウザに値返して反映させる。

7 音声入力による大きさと位置の調整

- ・マイクの画像をクリックすると、**10秒間**音声認識が可能になり、「**大きく、小さく、上、下、右、左**」と話すと**テキストとして認識**する。
- ・認識したテキストに応じて値の変更を行い、サーバーに送信後カメラ映像に反映される。

8 動画の撮影・保存

カメラデバイスから取得したフレームを撮影し、**左右反転**を行う。写真は**日付と時間**を含むファイル名でphotoファイルに保存される。

3 リアルタイムでの動画表示

- ・カメラから取得したフレームをFlaskサーバーが**左右反転**の処理を行い、JPEG形式に変換する
- ・処理されたフレームを**HTMLのimgタグ**で連続的に送信することで実現している

4 カメラの起動

- ・選択した**画像**と**ファイル名**や**カテゴリ**などがFlaskサーバーに送信される。
- ・受信したデータからカテゴリの判定を行う。アクセサリーは**page1.html**、それ以外は**page2.html**に移動してカメラが起動する。