

02 FukuMeet の開発

～新しい服との出会いをもっと簡単に～

3 番 及川結加、9 番 齋藤龍聖、24 番 米澤悠貴

指導教員 高橋 強

1. 研究の背景と目的

オンラインショップで洋服を購入する際に、イメージと違う商品を選んでしまう失敗を防ぐため、カメラの前に立ったユーザーが好きな服やアクセサリを画面の中でイメージできる「FukuMeet」を開発したいと考えた。

本研究では授業で学んだ知識の応用と新たな技術に加えて、システム開発のプロセスや問題解決力、メンバーや指導教員との協力を通じてコミュニケーション力やプレゼンテーション力も高めていくことを目的とする。

2. システムの概要

「FukuMeet」の開発には、HTML(Hyper Text Markup Language)[1]やCSS(Cascading Style Sheets)[2]、JavaScript[3]python[4]といったプログラミング言語や、Python のライブラリ(便利な機能をまとめたコード集)の OpenCV (Open Source Computer Vision Library) [5]や Mediapipe[8]、Flask[10]、rembg[14]などを活用している。

2.1 操作手順

操作手順を図1に示す。

①「スタートと新規カタログの追加」

「START」をクリックするとカタログページへ移動する。さらに、「画像を追加」をクリックすると入力フォーム(add.html)に移動する。入力フォームではパソコンに保存している画像を選択して、カテゴリ(Mens、Ladies、Kids、Accessories)の選択と画像の名前の入力を行い、アップロードすることで好きな服やアクセサリを選択することができる。

②「服やアクセサリを選択」

カタログページで服やアクセサリを選択する

と、選択リストに追加される。(カタログには追加した服やアクセサリが格納されている)「決定」をクリックするとアクセサリを選択している場合は page1.html、それ以外の場合は page2.html に移動し、ページが移動してカメラが起動する。

③「カメラ表示」

カメラが起動すると、選んだ服やアクセサリがリアルタイムでリサイズされ、画面上に重ねて表示される。音声入力または、スライドバーを調整することで、服やアクセサリのサイズと位置を変更することができる。

④「撮影」

「写真を撮る」をクリックすると、3秒後に撮影が行われる。写真は画面下部に表示される。

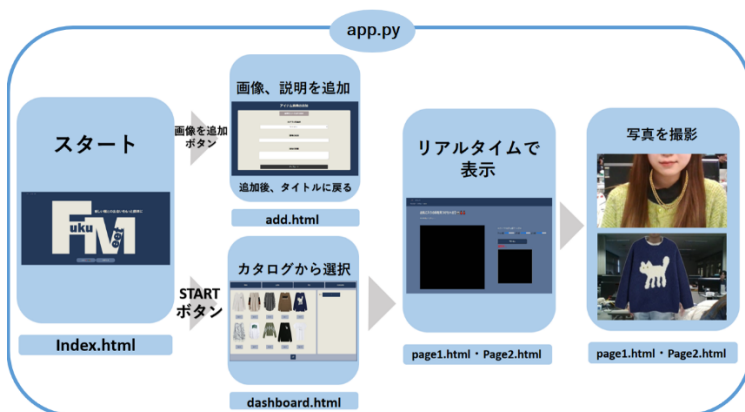


図1 システムの流れ

2.2 機能の詳細

【新規カタログの追加】

入力した画像や商品名、カテゴリなどは、Flask サーバーに送信[11-13]され、projectData.json に追加される。画像は背景透過[14]を行い、カテゴリごと(Mens、Ladies、Kids、Accessories)に保存され、保存先の画像パスが projectData.json

に追加される。

【カタログページでの服やアクセサリーの選択・表示】

ユーザーがカタログページで服やアクセサリーを選択し決定ボタンを押すと、サーバーにデータが送信「13」され対応する画像のパスが返される。返された画像のパスを使用して、カメラ映像に服やアクセサリーの画像をオーバーレイ表示する。服やアクセサリーは写真 1 のように projectData.json に格納されている。projectData.json の中には Mens、Ladies、kids、Accessories ごとの配列に画像パスと名前などを格納している。

【リアルタイムでの動画表示】

Flask サーバーがカメラから取得した動画のフレーム[6]の左右反転を行い、JPEG 形式に変換する。そして、変換した JPEG データを画面上のウィンドウに送信する。HTML では img タグを使って、送られてくる画像を表示し、ブラウザが次々と画像を読み込むことで映像が動いているように見せる仕組みである。

【カメラの起動】

「決定」ボタンをクリックすると、カタログの画像のデータ(ファイル名やカテゴリなど)が Flask サーバーに送信される。Flask サーバーは受け取ったデータをもとにカテゴリの判定を行い、アクセサリーは page1.html、それ以外の場合は page2.html に移動する。各ページに移動後、カメラが起動される。ページを分けている理由は、アクセサリーとそれ以外で異なる処理を行うためである。

【服とアクセサリーのモニター表示】

服とアクセサリーを人の体に重ねるには、Mediapipe のポーズ推定[9]を利用して体の関節の位置を取得し、関節の位置をもとに表示することが重要である。

服は、両肩と両腰の位置を取得し、肩と腰の中心値を求め首から腰の範囲で画像を重ねている。アクセサリーは両肩の関節の位置を取得し、両肩の中心値を求めて首元に画像を重ねている。表示中は正面

ではなく横を向くと画像の横幅も自動的に調整する。

【画像の位置や大きさを調整するスライダー】

ユーザーがスライダーを動かすとその値が Flask サーバーに送信される。Flask サーバーでは受け取った値を処理し、ブラウザに結果を返して画面を更新することで調整を行っている。

【音声入力による大きさや位置の調整】

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」と音声入力を行うと音声をテキストとして認識する。[14-17]そして、認識したテキストに応じて変更したスライダーの値を Flask サーバーに送信し、カメラ映像や画像処理に反映する。

【カメラ動画の撮影・保存】

カメラデバイスから取得した1フレームを撮影し、[7]左右反転を行い、写真は、photo というファイルに保存する。ファイル名には日付と時間が記載されている。

```
{
  "Mens": [
    {
      "image": "static/images/Mens/mens1.png",
      "name": "Item 1",
      "details": "サイズ: M, 素材: 綿, 価格: 3000円, カ"
    }
  ],
  "Ladies": [
    {
      "image": "static/images/Ladies/ladies1.png",
      "name": "Item 1",
      "details": "サイズ: S, 素材: シルク, 価格: 5000円,"
    }
  ]
}
```

写真1 服やアクセサリーの格納例
(projectData.json)

3. 開発環境

OS、ソフトウェア、プログラミング言語、ライブラリを表 1 に示す。開発環境の構成は図 2 に示す。

python と OpenCV(cv2)や Mediapipe、Rembg、Flask などの主なライブラリの説明と使用用途を以下に示す。

表1 開発環境

OS	Windows
ソフトウェア	Visual Studio Code
使用言語	Python,HTML,CSS,JavaScript
ライブラリ	Flask(Response,render_template, request,Jsonify,send_from_directory), OpenCV(cv2),Mediapipe,rembg

HTML5	CSS	JS			
Flask 3.0.3関連ライブラリ	OpenCV	Mediapipe	Math	rembg	
Python 3.12.4					
Visual Studio Code 1.96.0					
Windows10					

図2 開発環境イメージ

Python は画像認識やポーズ推定、リアルタイム処理に特化したライブラリやフレームワークが豊富である。今回の開発では、レイアウト以外の機能の詳細で示した全体の制御で python を使用した。

OpenCV(cv2)は、画像やビデオの解析、認識、変換などを行うオープンソースライブラリである。cv2 とは python で OpenCV を使用するためのモジュール名である。「FukuMeet」では、画像の反転や動画の撮影、保存、画像の重ね合わせなどで活用している。

Mediapipe は、リアルタイムでポーズ推定や顔認識、動きの追跡、物体検出などが可能である。「FukuMeet」ではポーズ推定を利用して、リアルタイムで両肩や左右の股関節の位置を特定している。

Rembgは画像の背景を高い精度で透過することが可能である。「FukuMeet」では好きな画像を追加する際に、背景透過を行っている。

Flask は web アプリケーションの作成やフォーム処理、HTML テンプレートのレンダリングなどが可能な軽量フレームワークである。「FukuMeet」では、フォーム処理や HTML テンプレートのレンダリング、OpenCV や mediapipe などで行った処理を web アプリケーションとして表示させている。

4. 今後追加したい機能

開発途中で追加したいと考えた機能を以下に示す。

- ①服とアクセサリなど違うカテゴリを同時に表示させる。(page3.html の追加)
- ②メガネなどのカテゴリを増やす。
- ③好きな画像を追加して背景をその画像に変更することができる。

5. 終わりに

本研究では OpenCV(cv2)や Mediapipe などのライブラリを活用し、「FukuMeet」の基本的な実装を行った。開発は個人ではなく、メンバーや指導教員と協力しながら進め、意見を共有することで、より良いアイデアを生み出し、課題に対する理解を深めることができた。開発中に発生した課題に対しては、インターネットで関連する技術や解決策を調べ、試行錯誤を繰り返すことで、問題解決力を高めることができた。一方、リアルタイム試着の精度向上や直感的な操作性の実現には、今後さらなる改善が求められる。今後も改良を重ね、より実用的なシステムを目指していきたい。また、産技短展ではより改善された「FukuMeet」を展示することができるよう、機能の向上やユーザー体験の最適化に取り組んでいきたいと思う。

6. 参考文献

[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/4b77a86030ed0071f5>

48?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>