IJARSCT
ISSN: 2581-9429
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Volume 5, Issue 7, June 2025
Impact Factor: 7.67

Generative Fashion Design Web Platform

Pragati Chandekar[1], Khushi Dorlikar[2], Awani Karkade[3], Ashlesha Patil[4], Khushi Chunarkar[5] Dr. Dhananjay Dumbere[6]
Department of Computer Science Engineering[1,2,3,4,5,6]
Department of Computer Science Engineering College
Rajiv Gandhi College of Engineering, Research and Technology, Chandrapur, India
pragatichandekar629@gmail.com, dorlikarkhushi6@gmail.com, avanikarkade65@gmail.com, patilashlesha875@gmail.com, arryachunarkarharrya@gmail.com, dddumbere@gmail.com

Abstract: *This paper presents a web-based real-time virtual try-on system that uses computer vision and machine learning techniques to enhance the online shopping experience. Leveraging MediaPipe for pose estimation, remove.bg for background removal, and a custom fit score metric, the system overlays 3D images onto users via webcam input. Shirt images are dynamically fetched using SerpAPI based on user prompts. The system is implemented using Flask and WebSockets for efficient real-time video streaming and feedback. Experimental results demonstrate the practicality and accuracy of the proposed try-on approach in delivering a realistic apparel fitting experience.*

Keywords: *web-based real-time virtual try-on system*

I. INTRODUCTION

Online shopping for clothing lacks the physical trial experience, often leading to dissatisfaction due to sizing and fit issues. Virtual try-on systems aim to bridge this gap by allowing users to visualize clothing items on themselves. This paper proposes a lightweight, real-time virtual try-on solution for shirts using a webcam and web technologies.

We utilize MediaPipe Pose for landmark detection, remove.bg for shirt background removal, and an intuitive fit scoring mechanism based on user pose and shirt dimensions. Our system allows shirt generation based on user prompts using SerpAPI and offers dynamic interaction through WebSocket-based updates.

II. LITERATURE SURVEY

Virtual try-on systems combine computer vision, image processing, and machine learning to simulate the experience of trying clothes digitally. The main components include pose estimation, background removal, image retrieval, and real-time garment overlay.

[1] This paper introduced ClothFlow, which uses optical flow to model how clothes deform around a person's pose. Unlike static overlays, this method focuses on preserving natural cloth deformations when a user changes position. Although your system doesn't use neural networks to model cloth deformation, it benefits from the idea of aligning clothing with dynamic human poses, guiding how your project places and scales shirts using shoulder landmarks.

[2] CP-VTON introduced a warping module and a refinement network to maintain fine garment textures and structural characteristics during virtual try-on. This is crucial when transforming 2D clothing images to fit different poses and body types. Your system adapts this principle by ensuring that shirt images are resized and rotated while preserving their original appearance (e.g., logos, sleeves).[3] MediaPipe is a real-time perception framework developed by Google. It supports human pose tracking, which is central to your project. Using pose landmarks like shoulders, neck, and hips, you detect the position and orientation of the user's upper body. This enables accurate placement of the shirt image in real-time using webcam input.[4] OpenPose is one of the first real-time pose estimators using part affinity fields (PAFs). It allows detection of body parts in video frames. Although your system uses MediaPipe, OpenPose shaped the general field of 2D human pose estimation. Understanding its mechanics helped you select a lighter alternative (MediaPipe) for better performance in web applications.[5] Remove.bg is a commercial API that uses AI to automatically remove backgrounds from images. In your system, this is crucial for preprocessing shirt images obtained

via SerpAPI. Removing the background ensures that only the shirt remains, ready to be overlaid transparently on the live webcam feed.[6] SerpAPI is used to retrieve shirt images dynamically based on user input (e.g., “red checked shirt”). This allows your system to adapt to different user preferences and makes the virtual try-on experience more personalized. It expands your project beyond static image libraries and into live shirt fetching.[7] PIFu reconstructs 3D models of clothed humans from a single image. While your system is 2D, this research is useful because it shows how important it is to consider body depth, contour, and orientation when overlaying clothes. You use MediaPipe’s 2D keypoints to approximate this, such as shoulder width and angle.[8] Spatial Transformer Networks (STN) allow networks to learn spatial manipulation (rotate, scale, translate) to align inputs. Although your project uses OpenCV for manual transformation, this work provides theoretical background on learnable transformation modules, which could be incorporated later for adaptive fit tuning.[9] This paper presents techniques for modeling human body shapes and poses efficiently. It shows how geometry and pose data can be used to tailor clothing. Your system simplifies this by focusing on shoulder width and frame size from MediaPipe landmarks to judge shirt fit.[10] This work focuses on the classification of apparel by type, pattern, and style. Your project doesn't classify clothing, but this line of work supports the idea of using visual features to enhance recommendations. In future versions, your system could analyze shirt features and match them to body types or events.

III. SYSTEM ARCHITECTURE

3.1 Overview

The system comprises:

A Flask-based web backend with SocketIO support. OpenCV for video capture and frame processing.

MediaPipe Pose for landmark tracking. Dynamic shirt image generation via SerpAPI. remove.bg integration for shirt isolation.

3.2 Components

Pose Detection: MediaPipe Pose detects 33 landmarks in real-time.

Shirt Generation: Based on prompts, SerpAPI fetches shirt images from Google Images. Background Removal: Shirt images are cleaned using remove.bg API.

Fit Score Calculation: Measures alignment of shirt dimensions to user's shoulders and torso using geometric formulas. Overlay Mechanism: Shirt image is scaled, rotated, and blended onto the frame based on pose data.

IV. MODULES DEVELOPED

The system has been developed in a modular architecture to ensure separation of concerns and maintainability. Below are the key modules implemented in the virtual shirt try-on application:

4.1. User Interface Module

Technologies Used: HTML5, CSS3, JavaScript, Bootstrap Functionality:

Captures real-time webcam video using HTML5 and JavaScript. Provides input field for users to search shirts (e.g., "formal white shirt"). Displays the output with live shirt overlay and fit score.

Provides buttons to start/stop webcam and change shirt.

4.2. Flask Backend Module

Technologies Used: Flask (Python), OpenCV, Flask-SocketIO Functionality:

Routes web requests and handles client-server communication. Streams live webcam frames to the frontend.

Coordinates between all modules: pose detection, shirt fetching, background removal, and overlay rendering.

4.3. MediaPipe Pose Detection Module Technologies Used: MediaPipe (Google), Python Functionality:

Detects 33 body keypoints including shoulders, hips, and torso center. Extracts relevant keypoints (left and right shoulders) to guide shirt placement. Continuously updates pose information frame-by-frame.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, June 2025
Impact Factor: 7.67

4.4. Shirt Fetching Module (SerpAPI) Technologies Used: SerpAPI, Python requests Functionality:
Accepts user input (e.g., "blue checked shirt") and performs Google image search using SerpAPI. Retrieves the top shirt image result (preferably a front-facing model or mannequin).
Passes the image to the remove.bg module for preprocessing.

4.5. Background Removal Module (remove.bg API) Technologies Used: remove.bg API
Functionality:
Removes background from the shirt image.
Returns a transparent PNG image (only the shirt remains). Optimizes the image size for overlaying on the user.

4.6. Shirt Overlay and Transformation Module Technologies Used: OpenCV
Functionality:
Resizes and transforms the transparent shirt image according to MediaPipe keypoints. Aligns shirt shoulders with detected human shoulders.
Handles scaling and rotation for realistic fit visualization.

4.7. Fit Score Estimation Module
Technologies Used: Python (custom logic using math and OpenCV) Functionality:
Calculates the shoulder width of the user from pose keypoints. Compares it with the shirt's shoulder width.
Generates a "fit score" (e.g., 92% fit), which is displayed on the screen.

4.8. Output Renderer Module
Technologies Used: OpenCV, Flask Streaming Functionality:
Renders the final augmented video frame combining webcam feed and shirt overlay. Displays the fit score visually in the frame.
Streams the output back to the web interface in real-time.

V. METHODOLOGY

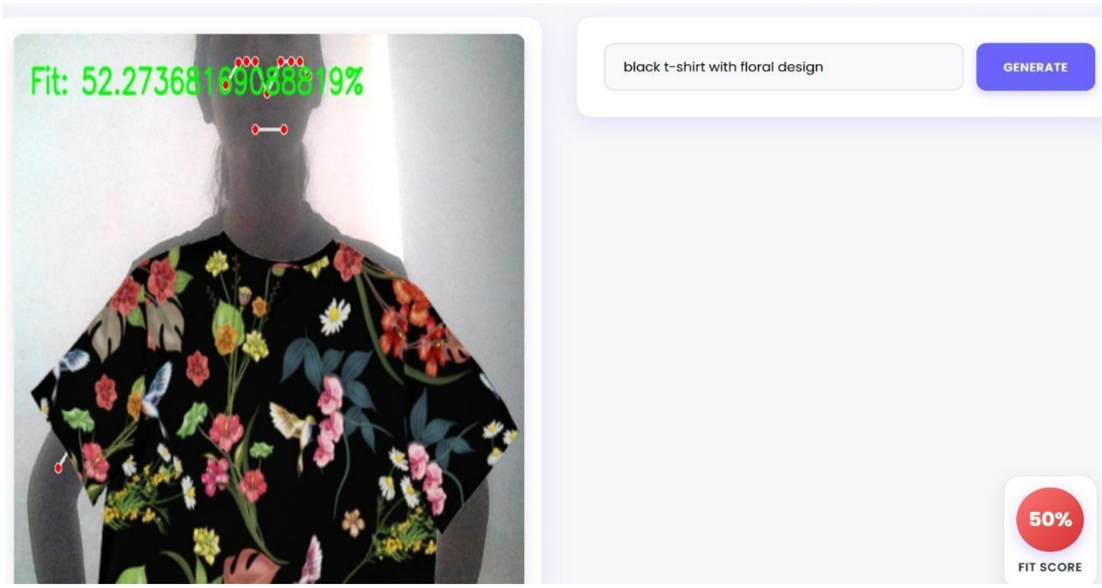
5.1 Pose-Based Shirt Placement Shirt positioning is computed using:
Shoulder and hip landmarks. Calculated angle between shoulders.
Torso length and shoulder width to scale shirt dimensions.

5.2 Fit Score Metric
A composite score out of 100 is computed using:
Width and height ratios (shirt vs. user body). Shoulder alignment angle.
Weighted formula: fit_score = (width_ratio * 0.5 + height_ratio * 0.3 + angle_score * 0.2) * 100.

5.3 Real-Time Frame Rendering
The video feed is processed at runtime:
Each frame is analyzed for landmarks.
The shirt is rendered using alpha blending and rotation.
Fit score is overlaid on-screen and sent to the frontend via WebSocket.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, June 2025
Impact Factor: 7.67

VI. OUTPUT



VII. CONCLUSION

The generative fashion design web platform represents a significant step forward in personalizing the shopping and clothing design experience. By combining face image recognition, body shape customization, and generative design algorithms, this platform offers users a powerful tool for trying on clothes virtually, which could revolutionize the way fashion is designed and consumed. Although there are still challenges to address, the potential impact on consumer behavior, sustainability, and fashion innovation is immense.

REFERENCES

[1]. Han, Xintong, et al. "ClothFlow: A Flow-Based Model for Clothed Person Generation." ICCV 2019.

[2]. Wang, Bo, et al. "Toward Characteristic-Preserving Image-based Virtual Try-On Network (CP-VTON)." ECCV 2018.

[3]. Lugaresi, Camillo, et al. "MediaPipe: A Framework for Building Perception Pipelines." arXiv:1906.08172 (2019).

[4]. Cao, Zhe, et al. "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields." CVPR 2017.

[5]. remove.bg API Documentation – Kaleido AI

[6]. SerpAPI Documentation – Real-Time Google Image Search API

[7]. Saito, Shunsuke, et al. "PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization." ICCV 2019.

[8]. Jaderberg, Max, et al. "Spatial Transformer Networks." NeurIPS 2015.

[9]. Khamis, Sameh, et al. "Learning an Efficient Model of Human Shape and Pose." CVPR 2015.

[10]. Maji, Subhransu, et al. "Fine-grained Visual Classification of Apparel." CVPR 2013