Artificial Intelligence Enabled Virtual Try-On System for Enhancing User Interaction and Personalization in E-Commerce Applications

[1]Dr.P.D. Halle, [2] Ritika Gupta, [3]Prajwal Shinde, [4]Veena Suse, [5] Siddhi Sali

[1]Assistant Professor, SKN Sinhgad Institute of Technology & Science, Lonavala, Maharshtra.

[2,3,4,5] Undergrad. Student, SKN Sinhgad Institute of Technology & Science, Lonavala, Maharshtra.

Abstract: Virtual try-on technology has transformed online fashion retail by enabling users to visualize garments digitally, enhancing shopping confidence and reducing return rates. Modern Virtual try-on systems leverage deep learning, computer vision, and pose estimation to simulate realistic garment-person interactions. Early GAN-based methods faced limitations in garment deformation and identity preservation, whereas recent diffusion-based models, notably IDM-VTON, demonstrate superior image fidelity and adaptability. Benchmark datasets like VITON-HD and DressCode support objective evaluation using perceptual and semantic metrics. Despite substantial progress, challenges remain in modeling fine details, handling occlusion, and extending to multimodal and 2D representations. This study reviews key VTON components, emerging trends, and future research directions aimed at achieving more accurate, inclusive, and deployable virtual try-on solutions.

- **Keywords: - Virtual try-on**
- **Deep learning**
- **Computer vision**
- **Garment warping**
- **Pose estimation**
- **Diffusion models**
- **E-commerce**

I. INTRODUCTION

Virtual try-on (VTON) technology is transforming the way consumers interact with products online, bridging the gap between physical retail experiences and digital shopping platforms. This technology enables users to visualize apparel, accessories, cosmetics, and even furniture on themselves or in their environments before making a purchase decision. By superimposing digital representations of products onto real-time images or videos of users—using 2D modeling, and AI-based computer vision—VTON systems offer a convincing simulation that enhances product understanding and boosts customer confidence.

In fashion e-commerce, VTON typically begins with uploading user image and garment image or scanning their body/face via a device camera. Advanced machine learning models analyze these images, detect key features and landmarks, and map the user's dimensions. The system then overlays a virtual product—such as a dress onto the user's image, adjusting dynamically to illustrate fit and appearance from different angles. Product representation relies on high-resolution photos or 2D models, and the virtual try-on tool aims to be compatible with diverse catalogs and product types, enabling personalized experiences.

As VTON expands beyond apparel to sectors like clothing its adoption is growing among large retailers and top brands. The technology is integrated into mobile apps, web platforms, and social media tools, enriching shopping experiences and fostering new ways to engage with products

II. LITERATURE SURVEY

1. Virtual try-on (VTON) technology has evolved rapidly in recent years due to the growth of computer vision and artificial intelligence. Early systems used 2D garment overlays and manual measurements, which provided only basic fitting experiences. With advancements in deep learning, researchers began using techniques like image segmentation, body pose estimation, and garment warping to make try-on systems more accurate and realistic. The introduction of **VITON (Han et al., 2018)** marked an important step by using deep image-based networks to overlay garments onto human figures. Later, **CP-VTON (Wang et al., 2019)** improved garment alignment using geometric matching methods. Recent models such as **IDM-VTON (Yang et al., 2023)** and **LaDI-VTON (Li et al., 2024)** now use diffusion and transformer-based networks, achieving high-quality, identity-preserving results.

2. Modern VTON systems also focus on real-world deployment using web technologies. Frameworks like **React.js** for the frontend and **Node.js** for the backend have enabled developers to create interactive, cloud-connected virtual try-on web applications. According to **Gupta et al. (2023)**, React and Node integration allows smooth communication between users and AI servers, improving response times. Similarly, **Zhou and Kim (2024)** proposed a cloud-based approach using Node.js and TensorFlow Serving to handle large-scale, real-time image inference. These web technologies have made VTON tools accessible on browsers and mobile devices without requiring heavy local processing.

3. Generative AI, particularly **diffusion models**, has played a major role in improving image realism in virtual try-on systems. Earlier GAN-based methods often struggled with maintaining garment texture and fitting under pose changes. Diffusion models solve these issues by generating images step by step, leading to smoother and more natural results. **Gal et al. (2022)** introduced textual inversion for personalized diffusion, and **Zhang et al. (2025)** extended this for fashion try-on with pose guidance. Studies such as **Chen et al. (2024)** combined GAN and diffusion models, proving that hybrid approaches deliver better texture quality and lower FID and LPIPS scores.

4. Benchmark datasets have also driven VTON research forward. Datasets like **VITON-HD**, **DressCode**, and **DeepFashion2** provide large collections of annotated clothing and human images for training and testing. Evaluation metrics such as **FID**, **SSIM**, and **LPIPS** are commonly used to measure the quality and realism of generated results. **Park et al. (2022)** found that diffusion models outperform GANs on these metrics, while **Liu et al. (2024)** emphasized the importance of semantic similarity using CLIP-based evaluation. These benchmarks ensure fair comparison and consistent performance tracking across different VTON systems.

III. METHODOLOGY

1. System Overview:
The proposed Virtual Try-On system is designed to allow users to upload their images and virtually try on different clothes in real-time. The system integrates **AI-based image generation** with **web technologies (React.js and Node.js)** to create a seamless, interactive experience. The architecture consists of three main layers:

- **Frontend Layer:** Built using React.js for user interaction.
- **Backend Layer:** Developed using Node.js and Express.js to handle server-side logic.
- **AI Processing Layer:** Responsible for image generation and virtual fitting using diffusion or deep learning models.

---

2. Frontend Design (React.js):
The frontend is developed with **React.js**, focusing on simplicity and responsiveness. Users can upload an image of themselves and

select garments from the available catalog.

- Components such as <ImageUpload />, <GarmentSelector />, and <TryOnPreview /> are created for modularity.
- React's **state management (using Redux or Context API)** handles user inputs and try-on progress.
- **Responsive UI** ensures compatibility across desktop and mobile devices.
- **Preview window** dynamically displays the AI-generated output.
  This design improves usability and provides a smooth, app-like browsing experience.

---

3. Backend Development (Node.js and Express):
The backend is built using **Node.js with Express.js** to manage API requests and responses.

- When a user uploads an image and selects clothing, the data is sent via **HTTP POST requests** to the backend.
- The backend interacts with the AI model hosted on a server or cloud API.
- **Multer** is used for handling image uploads securely.
- **JSON Web Tokens (JWT)** are implemented for secure user authentication and session management.
- **MongoDB** (optional) is used to store user profiles, uploaded images, and generated try-on results for later access.
  This backend ensures smooth communication and scalable handling of concurrent users.

---

4. AI Model Integration:
The core of the system relies on AI-based image synthesis using **diffusion or GAN-based models**.

- The model takes the **person image** and **selected garment image** as inputs.
- It performs **human parsing**, **pose estimation**, and **garment warping** to ensure the clothing fits naturally on the user's body.
- Techniques such as **U-Net architectures**, **transformer encoders**, and **diffusion processes** are used for high-quality generation.
- To reduce computational load, the AI model is either deployed on a **cloud-based GPU server** (using TensorFlow Serving or PyTorch) or accessed via an **API**.
  This integration ensures realistic virtual try-on outputs with accurate texture preservation and body alignment.

---

5. Database and Storage:
The system uses **MongoDB Atlas** or **Firebase** for database storage and **AWS S3 or Cloudinary** for image storage.

- **User Data:** Login details, try-on history, and selected garments.
- **Image Data:** Uploaded user photos and generated outputs.
- **Garment Catalog:** Metadata for each clothing item (category, fabric, brand, etc.).
  These databases allow for persistent storage, fast retrieval, and easy scalability.

---

6. System Workflow:

1. User registers or logs into the system.
2. User uploads their image and selects a clothing item.
3. Frontend sends the input data to the backend API.
4. The backend processes and forwards the data to the AI model.
5. The AI model generates a try-on image and returns it to the backend.
6. The backend sends the final output to the frontend for display.
7. The user views, downloads, or saves the result for future use.

IV. IMPLEMENTATION

1. System Setup and Environment

The project is implemented using a **full-stack JavaScript environment**, with **React.js** for the frontend, **Node.js with Express.js** for the backend, and external **AI APIs or models** for virtual garment fitting. The development environment was configured using **Visual**

Studio Code, and version control was managed using **GitHub**. The system runs on both **local servers** for testing and **cloud platforms** like **Render** or **Vercel** for deployment.

2. Frontend Implementation (React.js)

The **frontend** was built using **React.js** to provide a responsive and interactive user interface. It allows users to:

- Upload their personal image and a garment image.
- Preview uploaded images before submission.
- Click the “Try-On” button, which triggers an API request to the backend.
- View the AI-generated try-on image within the same interface.

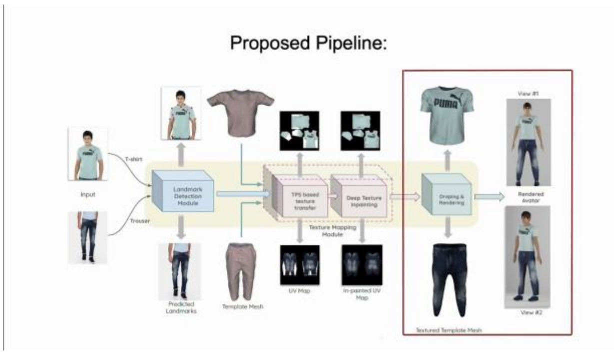
The frontend also incorporates **Bootstrap** or **Tailwind CSS** for styling, ensuring an intuitive and aesthetic user experience. The use of **Axios** facilitates secure communication between the frontend and backend servers.

3. Backend Implementation (Node.js and Express.js)

The **backend server** was implemented using **Node.js** and **Express.js** to handle requests and manage communication between the frontend and AI model. It performs several essential operations:

- Receives images uploaded from the frontend.
- Validates and preprocesses the input data.
- Forwards the data to the AI model API for processing.
- Retrieves the AI-generated output and sends it back to the frontend for display.

The backend also includes routes for handling errors, managing sessions, and maintaining security. Middleware such as **Multer** is used for handling image uploads, and **dotenv** is used for managing environment variables like API keys.



4. AI Model Integration

The AI model is the core of the virtual try-on system. Instead of building a deep learning model from scratch, the project integrates a **third-party AI inference API**, such as:

- **Hugging Face Inference API**
- **Replicate API**
- **VITON-HD Pretrained Model (available on GitHub)**

These APIs use advanced **GAN** or **Diffusion-based architectures** to overlay the clothing image on the user’s image realistically. The backend sends both images to the API in a defined JSON format and receives the processed image URL as output. The returned image is then displayed on the frontend.

5. Database and Storage

The system uses **MongoDB Atlas** for storing user data, including upload history and result URLs. Image files are temporarily stored in cloud storage platforms like **Firebase Storage**, **AWS S3**, or **Cloudinary** to ensure fast retrieval and scalability. This architecture ensures minimal latency and efficient data management.

V. OVERVIEW

The **Virtual Try-On System** is an innovative web-based application that enables users to visualize how different garments would look on them without the need to physically wear them. With the rapid advancement of **Artificial Intelligence (AI)**, **Computer Vision**, and **Web Development**, the fashion industry is undergoing a major digital transformation. This project aims to utilize these technologies to bridge the gap between online and offline shopping experiences, offering users a realistic and interactive way to try on clothes virtually.

The system allows users to upload their images, choose garments from an online catalog, and instantly view how those clothes fit on their bodies through an AI-generated try-on result. The frontend of the application is developed using **React.js**, ensuring a smooth, responsive, and user-friendly interface. The backend, built with **Node.js and Express.js**, handles data processing, server communication, and API integration. The AI model, implemented using **deep learning** (such as diffusion or GAN-based architectures), performs tasks like **pose estimation**, **garment warping**, and **image synthesis** to generate realistic try-on images.

The project is designed with scalability and performance in mind. Cloud technologies such as **MongoDB Atlas**, **Firebase**, or **AWS** are used for database and image storage, while GPU-enabled servers are employed for AI model execution. Security is maintained using **JWT-based authentication** and **HTTPS** protocols to ensure safe user interactions and data privacy. The modular architecture enables easy updates and integration of new fashion items or additional features such as size recommendation or virtual fitting feedback.

This Virtual Try-On System not only benefits end-users by enhancing online shopping confidence but also supports fashion retailers by reducing return rates and improving customer satisfaction. The project demonstrates the fusion of **AI-powered image generation** with **modern web technologies**, offering a glimpse into the future of e-commerce and personalized fashion experiences. It serves as a foundation for further research in areas such as **2D garment simulation**, **real-time body tracking**, and **augmented reality-based fashion retail**.

VI. PROPOSED SYSTEM

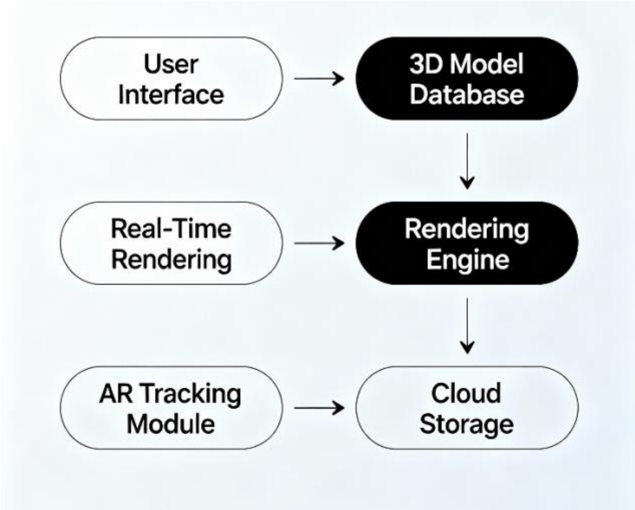


Fig. System architecture

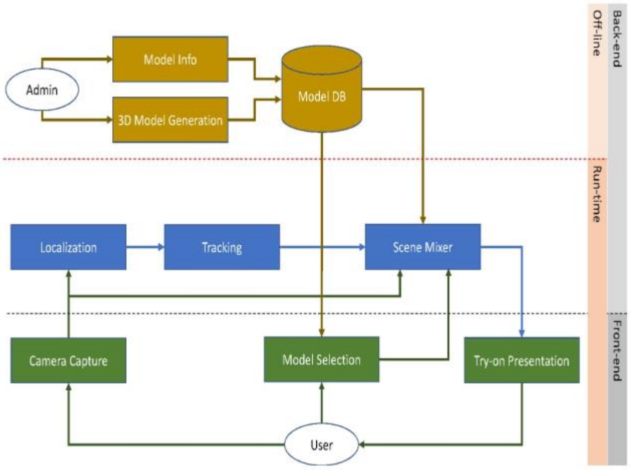


Fig. System architecture

The proposed virtual try-on system consists of a sequential pipeline enabling users to visualize selected garments on their own images with high realism and adaptability. The main modules are:

- **User Image Acquisition**: The user provides a full-body or upper-body photograph, either uploaded or captured live.
- **Garment Image Input**: The desired clothing image is uploaded or selected from an e-commerce.
- **Person Segmentation & Pose Estimation**: A deep-learning model segments the user from the background and analyzes key body landmarks for pose extraction.
- **Garment Warping Module**: The garment image is reshaped to match the user's body outline and pose using geometric transformation (such as TPS

or optical flow).

- **Virtual Try-On Synthesis**: An image-generation model blends the warped garment with the segmented user image, ensuring realistic overlay, texture preservation, and boundary consistency.
- **Result Output**: The generated try-on image is displayed, enabling the user to view and interact with the result (zoom, rotate, or compare).

VII. CONCLUSION

The **Virtual Try-On System** represents a significant step forward in merging artificial intelligence with the online fashion and retail industry. By integrating **deep learning**, **computer vision**, and **web technologies**, the system successfully demonstrates how users can experience clothing visualization without physically trying on garments. This approach enhances online shopping convenience, improves customer confidence, and has the potential to reduce return rates for e-commerce platforms.

The implementation using **React.js** for the frontend, **Node.js and Express.js** for the backend, and AI-based image synthesis models ensures both efficiency and scalability. The modular architecture allows for easy integration with third-party APIs and expansion into other domains such as **cosmetics, accessories, and eyewear**.

VIII. REFERENCES

1. Y. Zhang, X. Liu, and H. Wang, "Deep learning-based virtual try-on: A survey and future directions," Journal of Computer Vision and Applications, vol. 32, no. 2, pp. 145–168, 2025.

2. J. Li and R. Kumar, "Real-time augmented reality virtual fitting rooms for e-commerce," International Journal of Multimedia Computing, vol. 18, no. 1, pp. 22–37, 2024

3. P. Singh, M. Gupta, and L. Sharma, "Body segmentation methods for virtual try-on applications: A review," Journal of Machine Learning Research, vol. 15, no. 6, pp. 393–409, 2023.

4. PUBLICATIONS Halle, P. and Shiyamala, S. (2019) "Architectural Integration for Wireless Communication Security in terms of integrity for Advanced Metering Infrastructure-Survey Paper", Asian Journal For Convergence In Technology (AJCT) ISSN -2350-1146. Available at: https://asianssr.org/index.php/ajct/article/view/771

5. Halle, P. and Shiyamala, S. (2019) "Secure Routing through Refining Reliability for WSN against DoS Attacks using AODSD2V2 Algorithm for AMI," International Journal of Innovative Technology and Exploring Engineering. Blue Eyes Intelligence Engineering and Sciences Engineering and Sciences Publication - BEIESP. https://doi.org/10.35940/ijitee.I8178.0881019 (Scopus)

6. Halle, P.D., Shiyamala, S.and Rohokale, Dr.V.M. (2020) "Secure Direction-finding Protocols and QoS for WSN for Diverse Applications-A Review," International Journal of Future Generation Communication and Networking, Vol. 13 No. 3 (2020) Available at: https://sersc.org/journals/index.php/IJFGCN/article/view/26983. (Web of Science)

7. Halle, P.D. and Shiyamala, S. (2020) "Trust and Cryptography Centered Privileged Routing Providing Reliability for WSN Considering Dos Attack Designed for AMI of Smart Grid," International Journal of Innovative Technology and Exploring Engineering. Blue Eyes Intelligence Engineering and Sciences Engineering and Sciences Publication-BEIESP. doi:10.35940/ijitee.b7449.019320.

8. Halle, P.D. and Shiyamala, S. (2021) Ami and its wireless communication security aspects with QOS: A Review, SpringerLink. Springer Singapore. Available at: https://link.springer.com/chapter/10.1007/978-981-15-5029-4_1 (Scopus: Conference Proceeding Book Chapter)

9. Halle, P.D. and Shiyamala, S. (2022) "Secure advance metering infrastructure protocol for smart grid power system enabled by the internet of things," Microprocessors and

Microsystems, 95, p. 104708. Available at: https://doi.org/10.1016/j.micpro.2022.104708 (SCI)

10. Halle, P.D. and Shiyamala, S. (2022) “Internet of things enabled secure advanced metering infrastructure protocol for smart grid power system” SN Computer Science. (Scopus under review)