Advanced AI-Driven Solutions for Virtual Fitting and Personalized Fashion E-Commerce

Heramb Haridas, Vaishnavi Borkar, Saniya Bangare, Prof. Siddhartha Chandra
Department of Computer Engineering Sardar Patel Institute of Technology Mumbai, Maharashtra

***Abstract*—The rapid growth of ecommerce in fashion has high- lighted critical challenges, such as the inability to physically try on clothes, limited personalized guidance, and the need for real- time customer support. This paper presents an AI-driven fashion ecommerce platform that addresses these issues through the integration of advanced artificial intelligence tools. Key features include a Virtual Try-On system, which leverages computer vision and deep learning to allow users to visualize clothing items on their own images, enhancing purchase confidence. Additionally, a Clothes Replacement Tool utilizes the" Segment Anything" model (SAM) and Stable Diffusion techniques to enable users to apply different clothing patterns or designs to their uploaded images, further personalizing their shopping experience. The platform also includes an AI-powered Chatbot that provides immediate customer support using natural language processing to resolve user queries in real time. Lastly, a Personalized Recommendation System offers tailored clothing suggestions based on users' preferences and current fashion trends.**

Our methodology combines pose detection, segmentation, and stable diffusion inpainting, allowing for a robust virtual try-on experience and flexible clothing customization. We achieved 92% of Pose Detection Accuracy and 94.5% of Gesture Responsive-ness for Dynamic virtual try-on. This research demonstrates a significant impact on user engagement, reducing return rates and improving satisfaction with fit and appearance. Initial testing has shown clothes replacement features offer users an immersive, interactive and fast experience that closely simulates in-store try- ons in less than 30 seconds. Through a combination of machine learning models and innovative AI approaches, this platform represents a step forward in personalized, immersive ecommerce, indicating that AI-driven fashion solutions can effectively trans- form online shopping experiences and enhance user trust in digital fashion retail.

***Keywords*- Virtual Try-On, Pose Detection, Computer Vision, Deep Learning, Chatbot Assistance, Personalized Recommendations, Stable Diffusion Inpainting, Customer Support, Segmentation Models, User Experience, Machine Learning in Fashion, Fixed Ratio, Generative AI, Digital Smart Fashion**

I. INTRODUCTION

Online fashion shopping has completely changed how we buy clothes these days. While it's super convenient to shop from home, we all know the frustrations - you can't try things on, there's no one to help you choose, and getting quick answers about items can be a pain. This often leads to disappointed customers, huge number of returns, time wastage and people being less likely to hit that 'buy' button compared to when they shop in stores. To address these issues, this research introduces an AI-driven fashion ecommerce platform that incorporates advanced machine learning and computer vision techniques. The platform aims to enhance the online shopping experience through innovative features that mimic in-store interactions. This includes a Virtual Try-On system, a Clothes Replacement Tool, an AI-powered Chatbot, and a Personalized Recommendation System.

The Virtual Try-On system leverages computer vision and deep learning to detect the user's pose and overlay virtual clothing in real-time, allowing users to visualize how an item would fit. Using pose detection from the PoseDetector module, this feature aligns clothing based on key body pose points, creating an accurate virtual fitting experience. The code for this system includes real-time webcam inputs and overlays selected clothing items onto users through pose-based image processing and resizing algorithms. This interactive component not only improves purchase confidence but also has the potential to reduce return rates by helping customers make more informed decisions. You can see how clothes look on you in real- time using your camera. The technology looks at how you're standing and places the clothing right where it should be on your body, giving you a good idea of how things will actually fit.

Additionally, the Clothes Replacement Tool utilizes segmentation and inpainting technologies. Using the "Segment Anything" model (SAM) and Stable

Diffusion inpainting, this feature allows users to experiment with various designs on selected clothing items by generating new patterns or textures based on input prompts. This code pipeline involves detecting and segmenting specific clothing areas, and then applying Stable Diffusion to replace those segments with user-defined textures, enabling a highly personalized shopping experience.

We've also built a smart chatbot that offers instant support, assisting customers with technical inquiries and guiding them through the shopping process. Built with natural language processing (NLP) algorithms, the chatbot responds to customer questions in real time, improving the overall user experience by reducing response times and addressing common issues.

Also, a Personalized Recommendation System employs collaborative and content-based filtering to provide tailored clothing suggestions based on user preferences and browsing history. This feature helps reduce choice overload and encourages users to explore styles that match their tastes and needs.

Together, these AI-driven features create a robust and immersive platform that addresses the unique challenges of online fashion shopping. In the following sections, we delve into the key components of our solution, detailing the implementation of virtual try-on, clothes replacement, chatbot assistance, and personalized recommendations. Each feature leverages advanced AI techniques, including computer vision, natural language processing, and machine learning. Our research con- tributes valuable insights into the integration of AI in fashion ecommerce, aiming to offer users a seamless, interactive, and personalized shopping experience in a rapidly evolving digital landscape.

II. RELATED WORK

A. Virtual Try-On Systems in Fashion E-commerce

Virtual try-on technology has emerged as a crucial solution to bridge the gap between online and physical shopping experiences. Patel *et al* in 3D by considering human pose, body shape, and garment style [2]. This work demonstrated the feasibility of creating realistic virtual try-on experiences through deep learning models. Building on this foundation, CP-VTON+ by Minar et al. enhanced the virtual try-on experience by focusing on preserving clothing shape and texture during the visualization process, addressing key challenges in maintaining garment details during virtual fitting [6].

A systematic review by Batool and Mou revealed that virtual fitting rooms have significantly improved customer confidence in online purchases while reducing return rates [1]. Their analysis showed that integrating computer vision and deep learning techniques in virtual try-on systems can provide users with a more accurate representation of how garments will look on their bodies.

B. AI-Powered Fashion Recommendation Systems

Fashion recommendation systems have evolved significantly with the integration of AI technologies. Chakraborty et al. conducted a comprehensive review of fashion recommendation models, highlighting the effectiveness of combining collaborative and content-based filtering approaches [12]. Their research demonstrated that personalized recommendations can significantly improve user engagement and satisfaction in fashion e-commerce platforms. Further advancing this field, Sivaranjani et al. proposed a machine learning-based fashion recommendation system that considers both user preferences and current fashion trends [13]. Their work showed that incorporating multiple data points, including browsing history and style preferences, leads to more accurate and personalized recommendations.

C. Chatbots in E-commerce Customer Service

The integration of chatbots in e-commerce has revolution- ized customer service delivery. Hossain et al. developed an e-commerce platform with an integrated sales chatbot that demonstrated improved customer satisfaction through instant response capabilities [8]. Their research showed that natural language processing (NLP) enabled chatbots can effectively handle common customer queries and provide personalized shopping assistance. Joshi's work on e-commerce chatbots further emphasized the importance of combining sales analysis with chatbot functionality to provide more informed and contextual responses to customer inquiries [9]. This integration allows for more personalized and effective customer support while reducing response times.

D. Advanced Image Processing in Fashion

Recent developments in image processing have significantly impacted fashion e-commerce. Zhang et al.'s survey on the Segment Anything Model (SAM) highlighted its potential in fashion applications, particularly in clothing segmentation and pattern manipulation [14]. Their research demonstrated how foundation models and prompt engineering can be effectively applied to fashion-related tasks, enabling more sophisticated virtual try-on and clothing customization features. The application of AI in fashion retail, as explored by Ademtsu et al., has shown that combining computer vision with deep learning can transform both physical and online shopping experiences [11]. Their work emphasized how AI technologies can create more immersive and personalized shopping experiences while addressing traditional e-commerce challenges.

III. PROPOSED SOLUTION

A. *Dynamic Virtual Try-On*

1) *System Architecture:* The Virtual Try-On system uses real-time pose detection to superimpose selected clothing (like shirts) onto a user's live video feed. The setup as demonstrated in Fig. 1 includes several key components that work together to create an immersive and seamless experience:

- Input Module: Used to capture live feed from the user via webcam using OpenCV.
- Pose Detection Module: Detects landmarks using the PoseDetector from the CVZone library. It locates relevant points such as shoulders and arms for accurate placement of shirts.
- Shirt Overlay Module: After the points are located, this module dynamically resizes and places the shirts

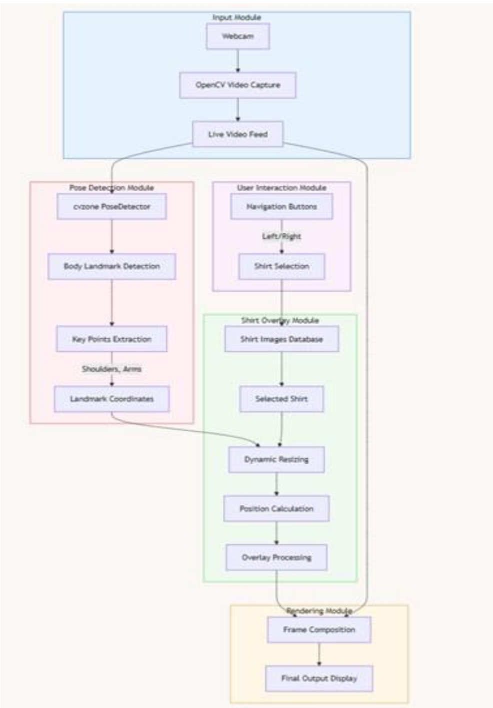

Fig. 1: System Architecture for Dynamic Virtual Try-On

corresponding to the detected landmarks. This ensures that the shirt is aligned proportionately with the user's body.

- User Interaction Module: This module incorporates navigation buttons (right and left) for cycling through available shirt options, enhancing user experience.
- Rendering Module: Combines the processed shirt over- lay along with the live video feed and renders it to the user.

2) *Tech Stack:*

- OpenCV: OpenCV is an essential component for capturing video feed, processing images, and rendering augmented outputs. The *cv2.VideoCapture()* module allows live video capture from the webcam, used for ana- lyzing the user's pose and body alignment. OpenCV's image processing capabilities, such as resizing and manipulating images, are integral to the project. Image overlay functions, like cvzone.overlayPNG(), seamlessly render augmented shirt images on top of the real-time video feed.
- cvzone: This module is crucial for detecting the adequate pose and simplifying complex operations like overlaying PNG images. The cvzone. PoseModule() detects

33 body landmarks with high accuracy, essential for determining positions like shoulders, elbows, and hips, which are key for accurate shirt alignment.

- Numpy: Used for mathematical computations such as calculating angles and performing array operations, which are vital for pose validation and alignment.
- Dataclasses: Used to define simple and immutable containers like PoseValidation, facilitating clean and efficient handling of pose validation results. Dataclasses eliminate verbose class definitions, enhancing code read- ability and maintainability.
- Python File I/O: Used to organize and access resources such as shirt images and pose guide overlays stored in structured directories.
- cv2.imshow: Provides a Graphical User Interface (GUI) to display the live video feed and augmented outputs, allowing users to experience real-time virtual try-ons.
- Event Handling: cv2.waitKey() handles events, taking keyboard input to switch shirts ('n'), go back ('p'), or close the camera ('q'), providing

an intuitive way for users to interact with the system.

- Real-Time Feedback Mechanism: Vital for guiding users into the correct posture to achieve optimal virtual try-on results.

3) Implementation Details:

- Pose Detection: The PoseDetector from cvzone.PoseModule identifies 33 body (as shown in Fig. 2) landmarks using a live webcam feed. The program checks the shoulder alignment, elbow angles and pose symmetry. These landmarks are crucial for calculating the dimensions and placement of the virtual shirt and the results are shown in Fig. 3.
- Pose Guide and Validation for user feedback: The system checks shoulder alignment, elbow angles, and pose symmetry. Calculates angles between key landmarks using trigonometric functions. If pose parameters match the T-pose thresholds (e.g., shoulder and elbow angles), the pose is validated. This ensures the shirt overlay happens only when the user's pose matches the required posture, preventing misalignment or awkward fits(see Fig. 4).

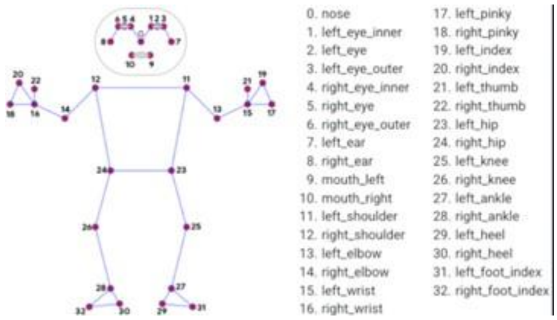

Fig. 2: Pose Tracking Full Body Landmarks

- Shirt Overlay and Fitting : The Virtual Try-On system utilizes a sophisticated approach to ensure the selected clothing item is seamlessly integrated with the user's appearance. The key to this process lies in accurately measuring and aligning the garment based on the user's body dimensions. The width of the virtual shirt is calculated using the horizontal distance between the left and right shoulders, scaled by a fixed ratio (fixedRa- tio=262/ 90), while the height-to-width ratio of the shirt (shirtRatioHeightWidth= 581 / 440) ensures the proper aspect ratio during resizing. The ShirtOverlay module determines whether the shirt should be overlayed, based on shoulder visibility and body orientation. A Shoulder Visibility Check validates that both shoulders are visible and their confidence scores exceed a defined threshold, while a Rotation Calculation ensures that the user is front-facing by determining the angle of rotation between the shoulders. For better fitting, dimensions are dynamically adjusted based on the distance between the left and right shoulders (shoulder width) and the distance from shoulders to hips (shirt height). These measurements can be scaled using predefined ratios (fixed Ratio, shirt Ratio Height Width) to maintain realistic proportions, and the resized shirt is placed at a calculated offset to align properly with the detected pose. The formulas to calculate fixed Ratio and shirt Ratio Heigh Width is given below: fixed Ratio=User Shoulder Width (landmark distance) / Shirt Shoulder Width (pixels) shirt Ratio Height Width=Shirt Width (pixels) / Shirt Height (pixels)

These formulas can be adjusted to accommodate variations in shirt dimensions or user body proportions.

- User Interaction: The Virtual Try-On enables the users to toggle between two options by their wrist movements. Two buttons are positioned on the screen: the right button navigates to the next shirt, and the left button moves to the previous one. Wrist landmarks (points 15 and 16 - ref to Fig. 6 and Fig. 7) are tracked, and when a wrist hovers over a button for a predefined duration, the system triggers the corresponding action. This touchless navigation provides an intuitive and engaging experience, eliminating the need for manual inputs while ensuring seamless toggling between shirt options.
- Shirt Loading and Management: All shirt images are stored in a structured directory (Resources/Shirts) for efficient management and scalability. The system dynamically loads these images, allowing seamless updates and supporting intuitive toggling between options using interactive buttons. This flexible design ensures the plat- form can easily accommodate new shirt additions without requiring changes to the core functionality.

4) Challenges:

- Pose Detection Accuracy: Ensures accurate detection of body landmarks under varying conditions, such as when the user is out of frame or in different lighting conditions.

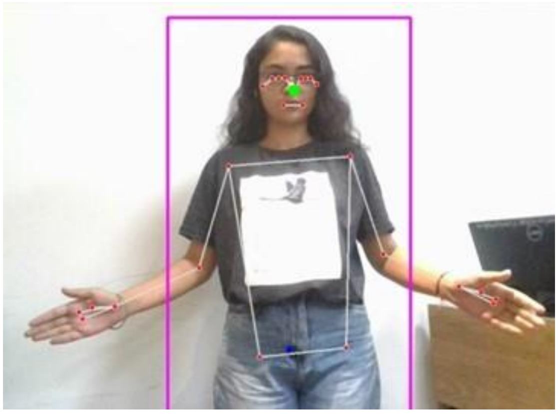

Fig. 3: Pose Detection

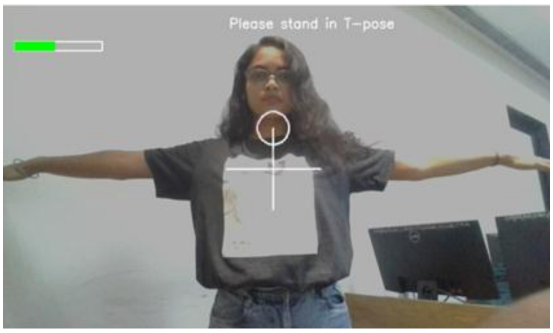

Fig. 4: Pose Validation

- Dynamic Resizing: Maintains realistic shirt proportions and alignment across diverse body types and poses.
- Real-Time Performance: Achieves smooth performance even on low-end devices while processing video, pose detection, and overlays in real time.
- Scalability: Manages a growing collection of shirt de- signs without compromising system performance.

B. Static Virtual Try-on

1) *System Architecture:* The system is designed for a generative AI-based clothes replacement application. It uses advanced tools like the Segment Anything Model (SAM) and Stable Diffusion for processing, segmentation, and inpainting. The workflow integrates hardware acceleration with CUDA GPUs and relies on several Python libraries to handle the processing pipeline. The system flow is highlighted in Fig 8 for better understanding.

- Input Module: This module captures the image from the user. It also ensures file validation, i.e., ensures that the image is in a compatible format – JPEG, PNG, etc.
- Preprocessing Module: Vital for image enhancement and ensures that the input image adheres to the require- ments of the segmentation model. It also normalizes pixel values to reduce them to the required threshold.

Fig. 5: Shirt Overlay

Fig. 6: Navigating between T-shirts

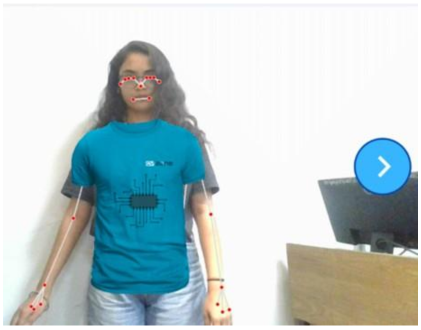

Fig. 7: Toggled to next T-shirt

- Segmentation Model: SAM (Segment Anything Model) is used to identify different regions (or segments) in the given image. This model can target different segments and regions, with a major focus on the clothing area.
- Mask Management Module: Used for handling the various masks generated by the segmentation model. It allows users to select a particular mask (or the region of the cloth) and make the necessary customizations.
- Inpainting Prompt Module: This functionality allows users to input prompts describing the desired replacement garment (e.g., ”red shirt with stripes”, ”Galaxy pattern shirt”).
- Inpainting Module: This module uses Stable Diffusion

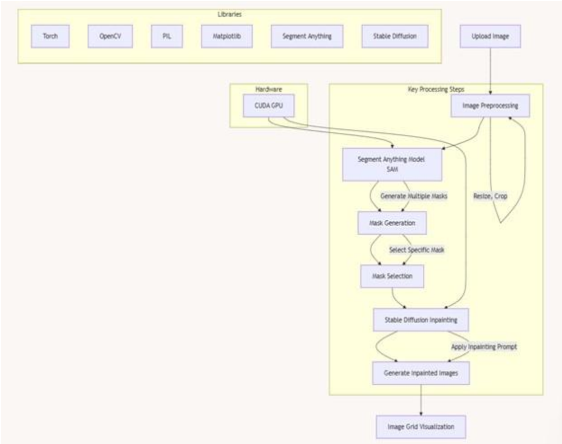

Fig. 8: System Architecture for Clothes Replacement

Inpainting to replace the selected cloth within the masked region. This functionality also helps maintain a realistic blend with the original image.

- Postprocessing Module: It fine-tunes the selected in- painted image by applying color correction and blend- ing. It ensures that the final output looks seamless and professional.
- User Interaction Module: Provides a smooth interface for users to interact with the system. This includes up- loading the image, selecting the region, and giving the prompt.
- Hardware Acceleration Module: Leverages CUDA GPU to speed up computationally intensive tasks such as segmentation, mask handling, and inpainting.

2) *Tech Stack:*

- Pillow (PIL): Used for preprocessing the input image, such as resizing it to a square aspect ratio and crop- ping unnecessary areas to ensure compatibility with the segmentation model. It also converts numpy arrays back into image objects after processing, making it easier to visualize segmentation masks and final results.
- Segment Anything Model (SAM): SAM is a pre-trained vision transformer model that performs semantic segmentation to detect and isolate specific clothing regions. By generating highly accurate segmentation masks, SAM eliminates manual effort and ensures precise masking for subsequent inpainting. It also provides parameters like predicted IoU (ideally, 0.9 ¡ IoU ¡ 1.1) and area for each mask to facilitate selection.
- Diffusers: The stable diffusion inpainting pipeline is deployed for a seamless clothing experience. It generates new designs based on user-defined text prompts, modify- ing the selected regions while preserving the rest of the image. Guidance scale and inference steps are important parameters as they control the quality and style of the inpainted output.
- PyTorch: PyTorch is the backbone for running SAM and Stable Diffusion models. It provides GPU acceleration for efficient computation, handling model loading, execution, and memory management.
- Matplotlib: This library is used to visualize the segmentation results, thus allowing the user to have a clear comparison between the original and the inpainted output.
- Numpy: NumPy handles mathematical operations and array manipulations required for generating segmentation masks and inpainting outputs.
- Regex & tqdm: These are used for parsing text-based inputs from the user.
- Scipy: Used for advanced mathematical operations like filtering, and image manipulations during preprocessing.
- xformers: Optimizes memory usage and speeds up at- tention mechanisms in Stable Diffusion.
- PyCocoTools, ONNXRuntime, and ONNX: Enables lightweight and optimized inference for models like SAM in production environments and handles dataset annotations and segmentation masks.

3) *Implementation Details:*

- Image Processing: Prepare the input image for segmentation and inpainting by resizing it to a relevant and fixed resolution for efficient computation. The image can be loaded using the PIL library. Standard resolution for the image can be 512 512 pixels(Fig. 9). This can be done using Lanczos resampling, preserving detail while minimizing distortion.
- Segmentation with SAM: Load the SAM model with a pre-trained ViT architecture (vit_h) and a checkpoint file for accurate segmentation. The mask generator function (Sam Automatic Mask Generator) creates masks for regions in the input image by sampling the points across a 32×32 grid and then filtering the masks based on the IoU threshold, stability score threshold (ideally, ~ 0.9), and minimum mask area (about 100 pixels). Multiple segmented masks are generated in the input image. Each mask is color-coded and labeled with a different

number as shown in Fig. 10.

- Mask Selection and Visualization: Masks are sorted by area to prioritize larger regions, assuming these are more likely to correspond to clothing. A user-friendly interface for mask selection is made, ensuring accurate targeting of the desired clothing region.
- Clothing Replacement with Stable Diffusion Inpainting: After the cloth is selected, initialize the stable diffusion pipeline for high-quality outputs. We use a user- provided prompt (e.g., ”Blue Floral Shirt”), thus allowing the inpainting model to replace the masked region with the desired design. The two important parameters for this process are – Guidance Scale (range lies between 7 and 12) & Number of Steps (value can range from 60 to 200).
- Guidance Scale ensures that the output aligns closely with the text prompt.
- Number of Steps is pivotal for refining the inpainting process.

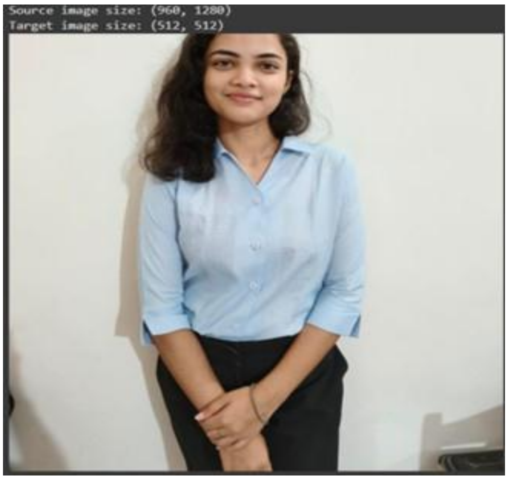

Fig. 9: Input image - Converted to 512 x 512 pixels

Fig. 10: Segmented Image with masks

Fig.13 and Fig. 14 clearly depicts high-fidelity clothing replacement, seamlessly integrating the new design with the surrounding image.

- Tech Stack and Hardware : NVIDIA CUDA-enabled GPU for accelerating segmentation and inpainting tasks.

4) *Challenges:*

- High Computational Requirements: The module demands significant GPU resources for running SAM and Stable Diffusion, which might require high computational devices.
- Storage Constraints: Large pre-trained models require extensive storage space, thus complicating the process of deployment.
- Slow Processing Speed: Generating masks and inpainting designs can be time-intensive, especially when handling high-resolution images or multiple prompts.

Fig. 11: Visualizing the Selected Mask

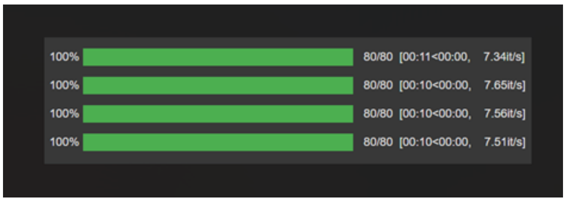

Fig. 12: Guidance Scale and Number of inference

Fig. 13: Results of generative clothing replacement - ‘Galaxy Pattern and Tiger Pattern

- Local Hosting Challenges: Hosting the solution locally can be difficult due to the heavy dependencies and computational overhead, necessitating cloud-based deployment for scalability.
- Dynamic Mask Challenge: Ensuring accurate mask selection, especially in cases with overlapping regions, can lead to errors or suboptimal outputs.

Scalability: Adapting the module to handle diverse clothing types, textures, and patterns requires extensive model fine-tuning and evaluation.

Fig. 14: Leaf Print T-shirt and Blue Floral T-shirt

C. Recommendation Systems

1) System Architecture:

The recommendation system is built to help the users find their perfect fashion preference amidst millions of options in the huge clothing and branding world. It delivers personalized suggestions to users based on their ratings of the product, reviews, product interactions like clicks and hovering on the product, product attributes and preference survey. A visual representation of the architecture is provided in Fig. 1. The system integrates:

- User Data Processor: Collects and preprocesses user data.
- Hybrid Recommendation Module: Combines both col- laborative and content-based approaches to enhance accu- racy, suggesting products based on user details, interests, and other product attributes.

Data flows from the frontend (user actions) into the backend, where it is processed by the recommendation engine to generate real-time suggestions displayed to the user.

2) Tech Stack:

- Frameworks: Tensor Flow and Scikit-learn for model development.
- Dataset: Fashion Clothing Segmentation Dataset from Kaggle.
- Architecture: ResNet50 for image classification and feature extraction.
- Tools: Python for development due to its extensive libraries and scalability.

These technologies were chosen for their scalability, flexibility, and ability to handle large datasets effectively. Tensor Flow supports efficient model training.

3) Implementation Details:

The recommendation system uses deep convolutional neural network ResNet50 which is 50 layers deep. Users upload an image which is processed to get the features. Nearest neighbours are searched through the database and then recommended to the users.

4) Challenges:

- Managing the dataset of a huge user base is difficult with all the processing for the recommendation system in such a complex and large industry. Since we made our project at a small scale, we didn't deal with this.
- We used a few metrics of collaborative and content- based filtering to increase the accuracy and scalability of the recommendation system. To increase the scalability at a high level, we found a solution to use distributed computing to manage large user and product datasets efficiently.

D. Customer help Chatbot

1) System Architecture:

Ecommerce platforms require chatbot assistance in order to streamline the user experience and resolve their small but important queries related to transaction and order. Our chatbot developed using Amazon Lex revolves around five user cases which are as follows: Transaction issue, Order status and tracking , Refund and return , Delivery and Customer Compensation. Lex is used for building the conversational interface using Intents and Slots attributes. Following are the components mentioned:

1) Amazon Lex:
- Used for building the conversational interface of the chatbot.
- Provides speech-to-text and intent recognition capa- bilities.
- Handles user inputs, identifies the intent (e.g., prod- uct recommendation, order status), and extracts en- tities.

2) AWS Lambda:
- Processes logic for chatbot responses.
- Executes functions for user queries, such as fetching product details or integrating with recommendation models.
- Ensures that responses are dynamically

generated based on the user's intent.

3) Amazon DynamoDB:
- Stores chat histories, user preferences, and session data.
- Enables personalized and contextual responses.

4) Amazon S3:
- Hosts static assets such as images for virtual try-ons, product catalogs, or promotions.
- Ensures secure and scalable delivery of media files to users.

5) Amazon API Gateway:
- Facilitates communication between Lambda and external services, if needed.

2) Tech Stack:

The chatbot extensively leverages AWS services for its implementation, ensuring scalability, reliability, and seamless integration. Key AWS tools like Lambda, Lex, and DynamoDB are used to deliver efficient and respon- sive interactions.

3) Implementation Details:

1) Frontend Interaction:
- The website's frontend is built using modern frame- works like React.js or Vue.js, running on Amazon S3 and Amazon CloudFront for fast delivery and scalability.
- The chatbot interface is integrated into the frontend using the Amazon Lex Web UI Kit. Users can interact with the chatbot directly from the website.

2) Backend Processing:
- User queries are sent to Amazon Lex, which inter- prets the intent and forwards the request to an AWS Lambda function.
- Lambda functions perform logic, such as querying DynamoDB for user-specific recommendations or fetching product details stored in S3.

3) Recommendation System Integration:
- Lambda can integrate with a pre-trained recommendation engine stored in an AWS SageMaker end- point. This allows personalized recommendations to be served to the user in real time.

4) Response Delivery:
- The processed response from Lambda is sent back to Amazon Lex.
- Lex formats the response and sends it to the frontend for display in the chatbot interface.

4) Challenges:
- Scalability: Solution: Leveraged AWS's serverless services (Lambda, DynamoDB, S3) to handle large user interactions efficiently.
- Latency: Integrated Amazon Cloud Front for frontend delivery and optimized Lambda function execution.
- Personalization: Integrated with a recommendation system using AWS Sage Maker for real-time insights.

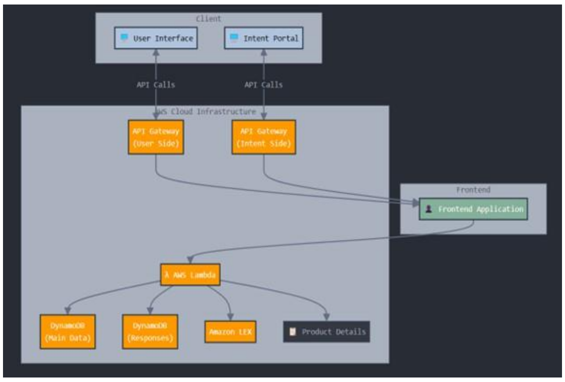

Fig. 15: System Architecture for Chat bots

V. RESULTS AND EVALUATION

1. Virtual Try-On

Here is a comparative analysis of some previous works in the same field.

Metric	Dynamic Virtual Try-On	CP-VTON+	TryOnGAN
Pose Detection Accuracy	0.92	0.89	0.87
Processing Time (ms/frame)	55 ms	70 ms	110ms
Gesture Responsiveness	94.5%	Not Supported	Not Supported
Alignment Symmetry	91%	92%	90%

TABLE I

Graphical Interpretation:

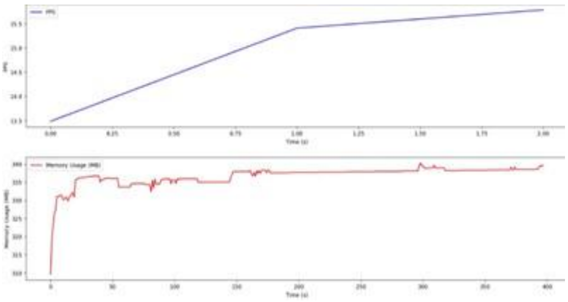

Fig. 16: Graphs for - FPS vs Time and Memory Usage vs Time

FPS starts at approximately 13.5 and increases steadily, crossing 15.5 by the end of the timeline. The linear increase suggests that the system optimizes its frame rate over time, potentially as the application initializes or caches data. There is no drop in F PS, indicating stable and improving performance during the observed period.

Memory usage starts at approximately 310 MB and quickly increases to about 330 MB within the initial stages. After reaching a peak of around 340 MB, the memory usage stabilizes with minor fluctuations around 335 MB for the majority of the timeline. The minor fluctuations suggest dynamic memory allocation, but the stability indicates no memory leaks.

The boxplot compares the distribution of 4 metrics - FPS, Processing Time, Garment Switch Time, Memory usage(see Fig. 17). And histogram in Fig. 18 shows the frequency distribution of garment switch times.

The Virtual Try-On system developed in this project seamlessly integrates advanced computer vision and AI techniques, delivering a user-friendly and efficient virtual try-on solution. Performance analysis showcases optimized frame rates, low processing times, and manageable memory usage, ensuring a smooth and responsive experience. Despite some challenges, the system demonstrates scalability and reliability, positioning it as a transformative force in the e-commerce landscape. By

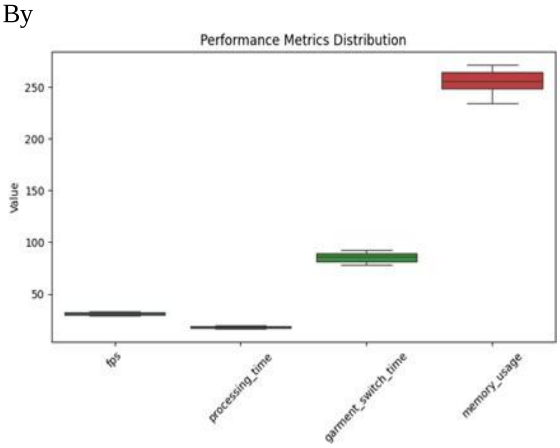

Fig. 17: Performance Metrics Distribution

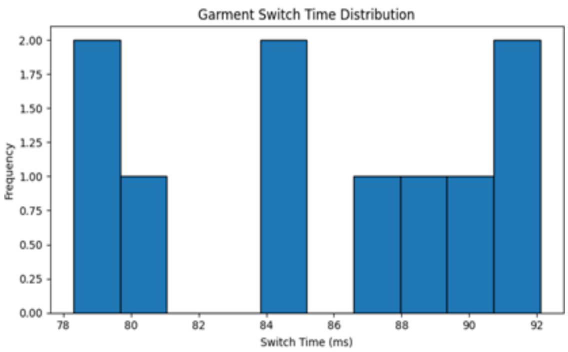

Fig. 18: Switch Time vs Frequency

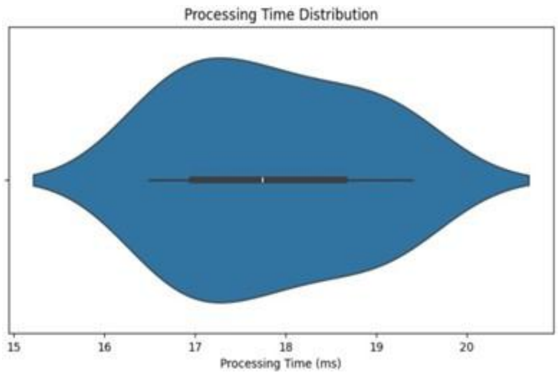

Fig. 19: Violin Plot for Image Segmentation and Generation

bridging the gap between physical and virtual shopping, the Virtual Try-On project offers innovative solutions that can revolutionize online retail, enhancing customer engagement and driving increased sales through a more immersive purchasing journey.

2. Clothes Replacement

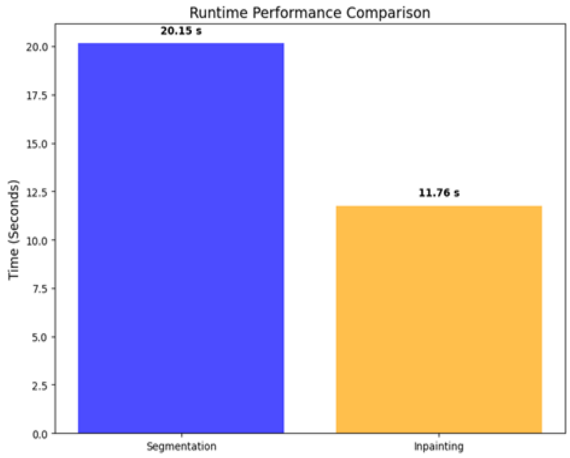

Fig. 20: Runtime Performance Comparison

As illustrated in Fig. 20 , the segmentation process takes significantly longer due to the computational complexity involved in generating accurate masks and model inference. In contrast, the inpainting process is relatively faster, as Stable Diffusion's inpainting mechanism appears to be optimized for image generation once the segmentation is complete. The Clothes Replacement system demonstrates the seam- less integration of cutting-edge AI models like Stable Diffusion and Segment Anything, enabling precise segmentation and realistic inpainting of garments. By effectively isolating specific regions and transforming them into custom designs, the system offers an interactive and visually compelling user experience. Through optimized prompts and efficient pipeline configurations, it achieves high-quality outputs while maintaining reasonable computational performance. Despite minor challenges like mask refinement and balancing guidance parameters, the solution showcases immense potential for revolutioniz- ing fashion e-commerce by enabling virtual garment customization and personalization. This project bridges the gap between user preferences and digital creativity, paving the way for scalable, user-centric applications in the online shopping domain.

3. Recommendation System

The recommendation system was evaluated using a clustering-based approach to assess its accuracy in suggesting similar images. A K-Means clustering algorithm was applied to the feature embeddings of the dataset, grouping the images into distinct clusters based on their visual similarity. For each query image, the system's top- 5 recommendations were analyzed to determine whether they belonged to the same cluster as the query image.

The evaluation resulted in an accuracy of 63%, indicating that approximately two-thirds of the recommendations

Fig. 21: User input to get recommends

successfully matched the expected cluster. While the results demonstrate moderate alignment with the clustering approach, there is potential for improvement through fine-tuning the feature extraction model, optimizing the clustering process, or increasing the number of clusters to capture more nuanced categories.

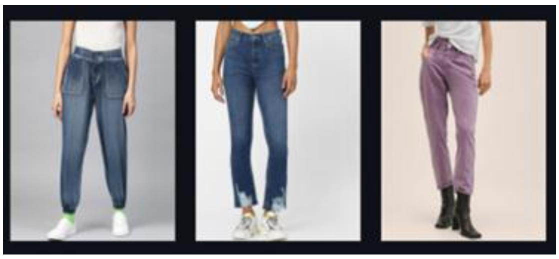

Fig. 22: Recommendations given by the system

Fig. 23: Another set of recommendations

4. Chatbot

The chatbot effectively resolves the basic query or needs of the users related to their transaction, order status and delay and even guide them about static and dynamic virtual-try on features.

VI. FUTURE SCOPES & CONCLUSION

As the e-commerce and fashion industry keeps on evolving everyday and hence keeping pace with the cutting edge technology is important to deliver the best user experience. Our website's foundation in static and dynamic virtual try-on, recommendation systems and AI-driven chatbots provides a strong starting point but there is huge potential to expand and innovate further.

The future scope includes working on the efficiency and accuracy of individual modules. From improving the fit aspect of virtual try-on features to enabling the economic feasibility of the customized cloth replacement to AI- powered chatbot , we envision a system that bridges the gap between shopping and in-store experiences. More- over, we can use emerging technologies like augmented reality(AR) and virtual reality (VR) which will solve most of the challenges that we face, considering it is predominantly built for such use cases like virtual try on. The main aim at the end is to increase user engagement and satisfaction and inclusivity in online fashion retail. Following are the important future ideas that could be included.

1) Developing Advanced AR and VR Features: Expanding the platform to include Augmented Reality (AR) features for mobile and web apps, as well as exploring immersive Virtual Reality (VR) try-ons for platforms like Meta Quest or Apple Vision Pro.
2) Improving Visual Quality with High-Resolution GANs: Currently, the Segment Anything Model (SAM) focuses on segmentation and not image resolution. Future work could involve integrating SAM with high- resolution GANs such as ESRGAN and Style GAN. This would enhance visual quality and sharpness, enabling highly detailed, user-specific customization of

even minute body parts. Improved resolution would also facilitate better segmentation by leveraging parameters like Intersection over Union (IoU).

3) Trend-Aligned Recommendations: Incorporating deep learning models like BERT embeddings can streamline the generation of trend-aligned recommendations. These models can efficiently process large databases to ensure users are updated with new fashion trends while considering their individual clothing preferences.
4) Enhancing Chatbot Functionality: Future developments can integrate transformers like GPT-4 to pro- vide accurate responses and better user interaction. The website could also include features like abandoned cart reminders and personalized product suggestions via the chatbot to improve user engagement.
5) Fine-Tuning SAM Architecture: Enhancing the SAM architecture by upgrading the feature extractor with ResNet or Vision Transformer (ViT) backbones and incorporating contextual modules like Atrous Spatial Pyramid Pooling (ASPP) in DeepLab or Atrous Convo- lution. This would improve the understanding of global image context, essential for inferring occluded regions.
6) Community Interaction Features: Adding a community chatting feature where users can share virtual try- on results and newly designed clothes via prompts. This features can also help sellers to actually provide the cloth having high upvotes.

VII. REFERENCES

[1] Batool and Jian Mou Raheela, *A systematic literature review and analysis of try-on technology: Virtual fitting rooms*, *Data and Information Management*, Volume 8, No. 2, June 2024, 10.1016/j.dim.2023.100060.

[2] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-M, *TailorNet: Predicting Clothing in 3D as a Function of Human Pose, Shape and Garment Style*, *CVF Open Access*, pp. 7365–7375, Jun. 2020, 10.1109/CVPR42600.2020.00739.

[3] Navneet Garg, Ankita Pareek, Ajinkya Lade, and S K Jagannath Charya, *Evolution in E-commerce with Augmented Reality*, *Conf. Series: Materials Sci- ence and Engineering*, 2021, pp. 1-10, 10.1088/1757-899X/1012/1/012041.

[4] Xiuquan Qiao, Pei Ren, and Schahram Dustdar, *Web AR: A Promising Future for Mobile Augmented Reality*, *Proceedings of the IEEE*, Volume 107, No. 4, April 2019, pp. 651-666, 10.1109/JPROC.2019.2895105.

[5] Travis Moniaga and Viany Utami Tjhin, *The Use Of Augmented Reality In E-Commerce: A Bibliometric Study*, *Journal of Theoretical and Applied Information Technology*, July 2022, Vol. 100, No. 13, pp. 4814-4826.

[6] item Matiur Rahman Minar, Thai Thanh Tuan, Heejune Ahn, Paul L. Rosin, and Yu-Kun, *CP-VTON+: Clothing Shape and Texture Preserving Image-Based Virtual Try- On*, 2020.

[7] Sang-Heon Shim, Jiwoo Chung, and Jae-Pil Heo, *Towards Squeezing-Adverse Virtual Try-On via Sequential Deformation*, *AAAI*, Dec. 2023, 10.48550/arXiv.2312.15861.

[8] Mostaqim Hossain, Mubassir Habib, and Mainuddin Hassan Faria Soroni and Mohammad Monirujjaman Khan, *Research and Development of an E-commerce with Sales Chatbot*, *2022 IEEE World AI IoT Congress (AIIoT)*, Jun. 2022, 10.1109/AIIoT54504.2022.9812772.

[9] Shreyas Joshi, *E-Commerce Website with Chatbot, Sales Analysis and Price Optimiza- tion, IJNRD*, Vol. 8, N o 4, p p 607-610, https://www.ijnrd.org/papers/IJNRD2304279.pdf.

[10] Gaolu Huang, *Artificial Intelligence Enabled Apparel Design Research*, *Applied Mathematics and Nonlin- ear Sciences*, 9, No. 1, Jan. 2024, pp. 1-16, doi: 10.2478/amns.2023.00338.

[11] James Tetteh Adentsu, Dr. Pragya Pathak, and Osei Dinah Bridget Oduroa, *Role of AI in Changing the Physical and Online Shopping Experience of Clothes and Fashion Products*, *International Journal of Multi-disciplinary Research and Analysis*, Vol. 6, No. 10, Oct. 2023, pp. 4981-4991, 10.47191/ijmra/v6-i10-57.

[12] Samit Chakraborty, Md. Saiful Hoque, Naimur Rah- man Jeem, Manik Chandra Biswas, Deepayan Bardhan, and Edgar Lobaton, *Fashion Recommendation Systems, Models and ....*

[13] Lingala Sivaranjani, Sandeep Kumar Rachamadugu, B.V. Suresh Reddy, *Basi reddy*

A, M.Sakthivel and Sivakumar Depuru, Fashion Recommendation System using Machine Learning, 24th International Conference on Smart Electronics and Communication (ICOSEC), Sep. 2023,.

[14] Chaoning Zhang et. al, A Survey on Segment Anything Model (Sam);*Vision Foundation Model Meets Prompt Engineering, arXiv, Vol.1, May 2023,.*