Journal of Soft Computing Paradigm (ISSN: 2582-2640)
www.irojournals.com/jscp/

Transforming E-commerce Experiences with Augmented Reality and Computer Vision: A Framework for Virtual Try-Ons and Home Decor Visualization

Anisha Kharel[1], Alija Bhujel[2], Smita Adhikari[3]

[1, 2]Student, [3]Assistant Professor, Department of Electronics and Computer Engineering, Pashchimanchal Campus, Tribhuvan University, Nepal

E-mail: [1]anishakharel456@gmail.com, [2]alijabhujel111@gmail.com, [3]smita@wrc.edu.np

Abstract

In this paper, we present an end-to-end mobile system that integrates computer vision and augmented reality to enable AR views of home decor items in e-commerce as well as real-time virtual try-ons of t-shirts and eyewear. The system is implemented using Django with SQLite for backend integration and Flutter for cross-platform mobile release. The system solves the three main e-commerce problems of low engagement, uncertainty, and product misfit. Improving preprocessing for the VITON-HD model is one of the major contributions of this research. To accomplish appropriate data preparation, we refined human parsing, cloth segmentation, and pose estimation. The LabelMe tool was used to create a personal dataset of 1000 labeled images for training, and 200 images were used for testing. Body parts like hair, face, neck, upper_body, lower_body, left_hand, right_hand, and skirt were labeled in the images. YOLOv8x, which was trained using a human parsing model for this dataset, achieved mAP50/mAP50–95 scores of 0.899/0.808 for masks and 0.915/0.833 for bounding boxes, respectively. Better synthesis outcomes and efficient segmentation were made possible by the model. To properly place and size 2D glasses using alpha blending, we employed dlib's 68-point facial landmark detector to identify eye regions during the glasses try-on process. A robust model-viewer-plus package serves as the foundation for the AR visualization of home decor product functionality, enabling the use of augmented reality to display 3D models in the

DOI: https://doi.org/10.36548/jscp.2025.3.001
Received: 29.05.2025, received in revised form: 02.07.2025, accepted: 18.07.2025, published: 01.08.2025

real world. By combining these features into a mobile application, the framework provides a simple and engaging way to encourage user satisfaction and confidence when shopping online.

Keywords: Augmented Reality, Cloth Segmentation, Human Parsing, Pose Estimation, YOLOv8x.

1. Introduction

1.1 Background

The e-commerce industry has witnessed exponential growth, offering convenience and a wide selection of products. However, clothing products searched for by online customers may not be satisfactory, as the option to try on clothes and see how they would appear on them is not provided. Additionally, the cost of returned items is borne by vendors due to customer dissatisfaction [1]. The impact of AR-based virtual try-on on purchase intention was analyzed using the Value-Based Adoption Model (VAM) with data from 318 Indonesian Gen Y and Z respondents, for which perceived value, ease of use, usefulness, enjoyment, and technology informativeness were found to significantly influence purchase intention [2]. Computer vision and augmented reality technologies enable further creativity in e-commerce. Computer vision makes it possible to understand what is being seen in real time, while augmented reality overlays digital content on top of the physical world. Combining these technologies allows customers to virtually try on clothes or visualize home decor within their actual spaces, eliminating the uncertainty associated with online purchases. In e-commerce, AR is key to turning window shopping into a hands-on digital product discovery that also fosters a sense of ownership and emotion before purchase. This adds to the decision-making process with true scale, color accuracy, and spatial alignment, which are lacking in 2D product visualization. It not only improves your store's customer experience and trust but also increases purchase satisfaction, reduces product returns, and boosts sales. The developed system uses Django for the powerful backend and Flutter for the cross-platform as a flexible choice. Leveraging AR in e-commerce enables retailers to connect digital and physical retail experiences and enhance the level of engagement and trust in the purchaser’s journey.

This research is important as it addresses practical challenges in online retail, such as misfit products, lack of visual feedback, and low purchase confidence. By bridging high-quality synthesis techniques with mobile AR technology, this work contributes toward more

personalized and immersive shopping experiences, especially in contexts where return policies or trial options are limited.

1.2 Problem Statement

E-commerce businesses have found it especially difficult to replicate the in-person shopping experience, particularly in the fashion and home furnishings industries. The novelty factor of a particular pattern or fabric makes it challenging for customers to visualize how the product will fit and appear on the body. This undermines their confidence in making a wise purchase and frequently leads to a high percentage of items being returned. Conventional product displays, which only include a static photo and a text description, lack the realism and engagement necessary to fully involve users and give them the impression that they are having a genuine buying experience.

1.3 Objective

To create an integrated platform that simulates eyewear, AR-based home decor visualization, and virtual clothing try-ons in order to replicate the authenticity of in-person shopping in e-commerce, boosting consumer confidence and lowering return rates.

1.4 Contribution

Although virtual try-on systems have been explored in prior research, most of them rely heavily on pre-trained models and lack adaptability and efficiency in real-world mobile applications. In contrast, our work presents an integrated framework that focuses mainly on optimizing the preprocessing pipeline, which is an important stage for improving the quality of try-on synthesis. The core contributions of this work are outlined below:

1. Design and implementation of a mobile-based integrated framework that combines virtual try-on and AR features in an e-commerce application using Flutter and Django for real-world deployability.
2. Optimization of the preprocessing pipeline, including pose estimation using Body25 through MediaPipe, an efficient cloth segmentation model, and an enhanced human parsing model trained on a custom dataset of 1000 annotated images and validated

using 200 images, achieving mAP50 of 0.915 and mAP50-95 of 0.833 for bounding boxes as well as mAP50 of 0.899 and mAP50-95 of 0.808 for segmentation masks.

2. Related Works

Title of the paper	Their work	Limitations
AR Shoe: Real-Time Augmented Reality Shoe Try-on System on Smartphones [3]	Offers an advanced, high-fidelity virtual shoe try-on experience, featuring realistic 3D rendering and effective occlusion handling for more accurate visualization.	It demonstrates only on desktop systems rather than smartphones, relies on a complex multibranch network and occlusion processing, and thus imposes significant computational requirements specific to the AR shoe application.
An Augmented Reality Virtual Glasses Try-On System [4]	Focuses exclusively on virtual try-on for glasses, providing an application dedicated solely to eyewear visualization.	Requires RGB-D cameras for accurate depth sensing, which makes it unsuitable for standard smartphones and limits its use specifically to eyewear applications.
Development of Augmented Reality Application for Online Trial Shopping [5]	Employs relatively basic augmented reality integration using Unity with face tracking capabilities, but does not utilize any custom datasets.	Focuses exclusively on apparel try-on and lacks deeper integration with advanced computer vision techniques.
Deep Learning in Virtual Try-On: A Comprehensive Survey [6]	Reviews deep learning techniques for virtual try-on from 2017 onward, summarizing and comparing different clothing try-on methods in a purely survey style.	Purely a survey without proposing a new system, which focuses only on clothing, analyzing existing methods and datasets.
Designing an AI-Based Virtual Try-On Web Application [7]	A web-based eyewear try-on using Vue.js and Babylon.js, with realistic 3D rendering via PRNet and the FFHQ dataset.	Limited to eyewear, prone to texture holes from occlusions needing frontal photos, and has higher processing demands than 2D methods.

CloTH-VTON+: Clothing Three-Dimensional Reconstruction for Hybrid Image-Based Virtual Try-ON [8]	Focuses entirely on clothing virtual try-on by reconstructing a 3D clothing mesh from a single image, using SMPLify-X for 3D human pose estimation, with CloTH-VTON+ performing better for complex poses and handling strong occlusions.	Not yet optimized for mobile deployment, requires significant GPU resources for image synthesis, and is limited to clothing applications only.
VisionCart: Enhancing E-Commerce with Augmented Reality for an Immersive Shopping Experience Using Flutter and Arkit [9]	Focuses on home decor visualization and product placement in AR, mainly utilizing ARKit for iOS devices.	Limited to home decor use cases, works only on iOS with no Android support, and merely places pre-made 3D models without customization.
Revolutionizing Online Shopping With FITMI: A Realistic Virtual Try-on Solution [10]	Uses Latent Diffusion Models and the Dress-Code dataset for virtual try-on of full and lower-body garments with automatic data integration.	Struggles with complex textures, heavy garment warping, and reduced pose accuracy under harsh lighting or shadows.
Mixed Reality Virtual Clothes Try-On System [11]	Presents a purely virtual clothing try-on system using mixed reality, which adjusts skin tone to match the user's face, captures body metrics via an RGB-D sensor, and auto-generates a personalized avatar for fitting, alignment, and clothes simulation.	Requires specialized hardware like Kinect or RGB-D cameras, is not mobile-optimized, and may still need user input for accurate measurements.
Expand: An Immersive Virtual Try-On System With Augmented Reality [12]	Implements real-time video processing using MediaPipe Pose for pose tracking and Three.js for 3D model rendering to create immersive web-based AR experiences.	Does not support clothing try-on, focuses instead on items like watches and glasses, excludes home decor, is web-based without modern synthesis techniques, and lacks a proper prototype implementation.

3. Proposed Work

3.1 System Block Diagram

Figure 1 shows the architecture of the system. It is created to provide a complete online shopping experience by combining AR and computer vision techniques. The frontend has been developed on Flutter for a better responsive user interface, whereas Django and SQLite have been used for backend development, including the creation of APIs and handling overall data. For the AR experience, Model Viewer allows realistic AR views of products in your physical space. The virtual try-on pipeline consists of four stages: pre-processing, segmentation generation, clothes deformation, and try-on synthesis. In the pre-processing phase of our pipeline, we aimed to optimize the pose estimate, human parsing, and cloth segmentation to obtain the correct model inputs. The pipeline also includes dataset generation, training models, and evaluation for realistic results.

One of the interesting features is the Glasses Try-On. It identifies the position of the eyes using facial landmark mapping (built-in in dlib), calculates the translated size of the specs, and uses pixel-wise alpha blending to create a real-time overlay. This adds a personal touch to the online shopping experience.

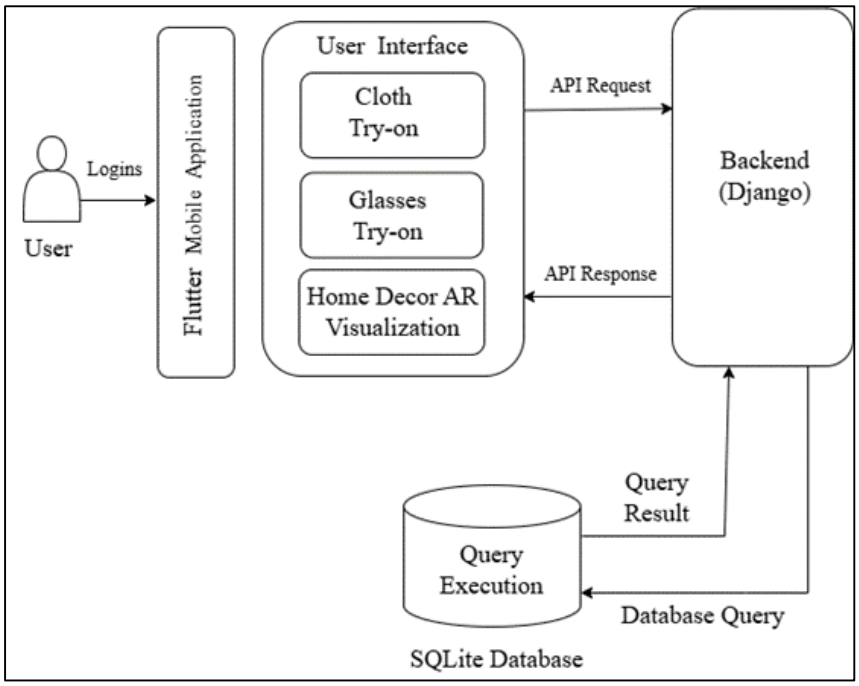

Figure 1. System Block Diagram of Proposed Application

3.2 Principal of Virtual Try-On Feature

Steps for input preprocessing, data preparation, model training and evaluation, human parsing, and final image synthesis using clothing deformation and alias normalization techniques are all shown in figure 2, the workflow diagram of the virtual try-on system pipeline.

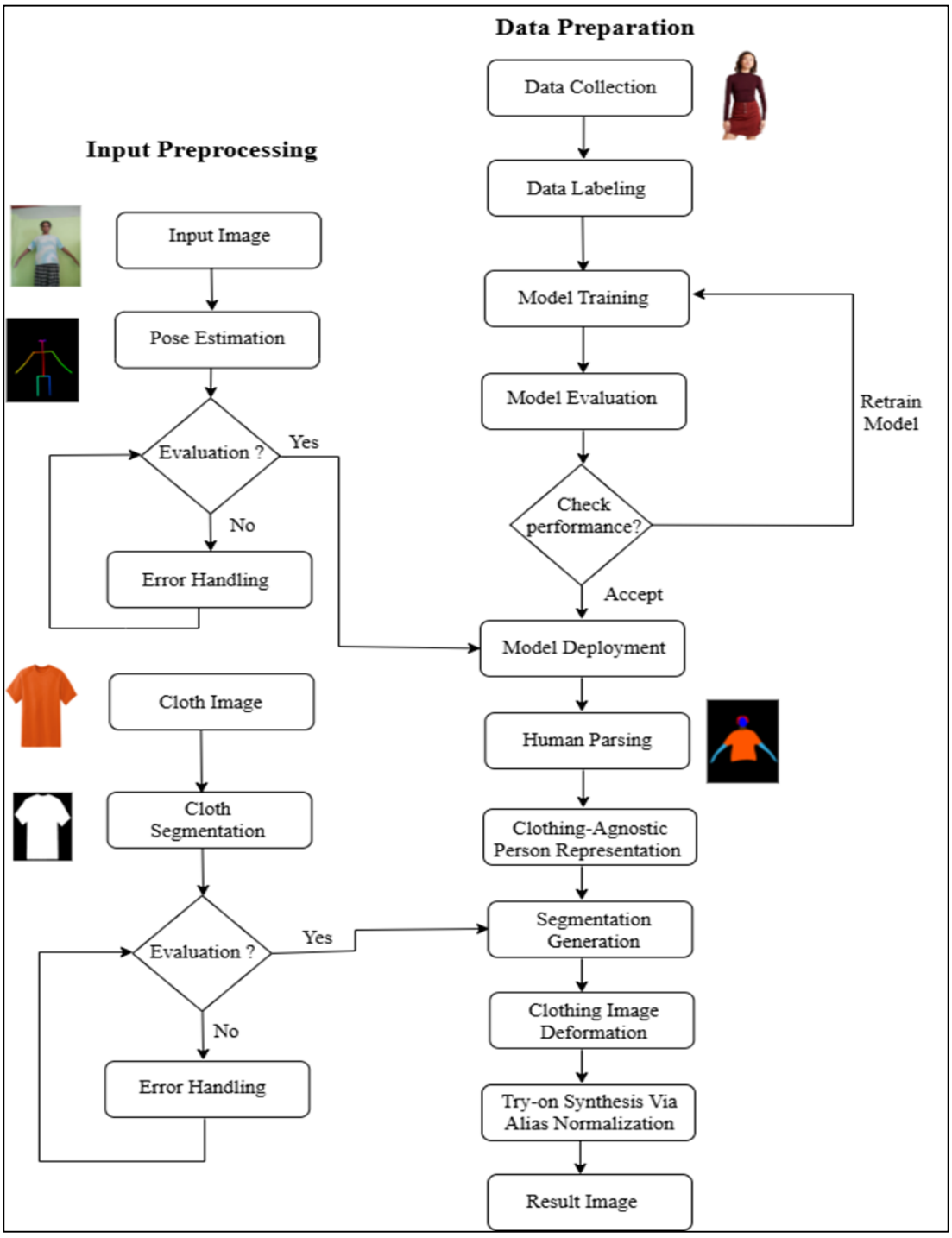

Figure 2. Flowchart of Virtual Cloth Try-On

3.2.1 Data Preparation

Dataset Collection and Annotation:

It was trained on 1000 annotated images and validated on 200 images. The annotation of a labeled image contains information about hair, face, neck, upper body, lower body, right hand, left hand, and skirt. The labeling is done with the LabelMe tool, which is both efficient and precise for annotating region-wise labels, enabling high-quality data to train our human parsing model. Figures 3–5 show the labeling process with the Labelme tool (Figure 3), the label output that corresponds to it (Figure 4), and the label contours that are visually overlaid on the original image for verification (Figure 5).

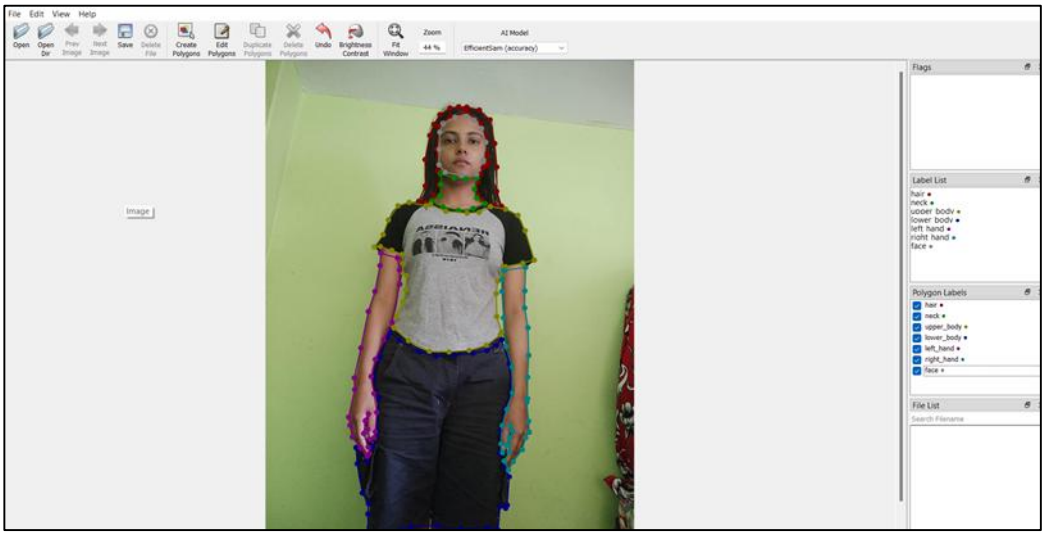

Figure 3. Labelling using Labelme Tool

Figure 4. Label

Figure 5. Label Visualization

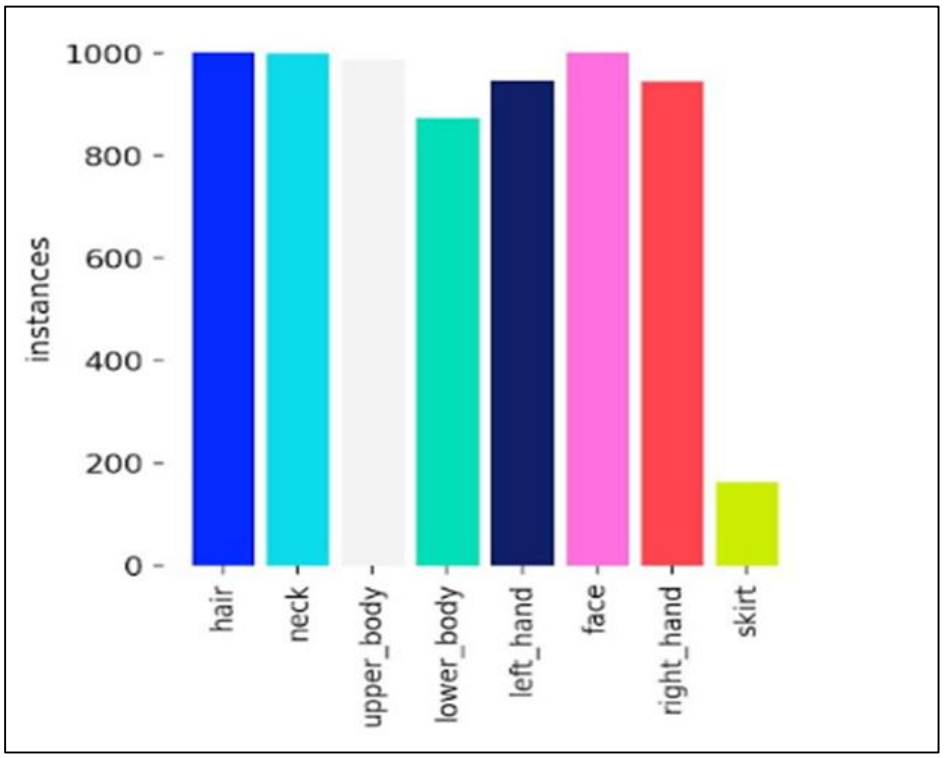

Figure 6. Number of Labels in Training Dataset

Figure 6 shows the distribution of label instances in the training dataset by body part, with fewer instances for the skirt and more for the categories of face, neck, and hair.

Pose Estimation:

Pose estimation is a computer vision application that estimates the position and orientation of a person or an object in an image or video. It is used to detect and track landmarks such as a user's body joints or features of an object so that the spatial layout and movement can be understood with precision. The Body25 pose estimation format was chosen over the other two popular formats of pose estimation, COCO and MPII, as it covers the human body in detail. The body25 format from OpenPose consists of 25 key points on the whole body, including the head, hands, and feet, and is appropriate for applications requiring fine-grained motion analysis. Although VITON-HD is compatible with the Body25 format, OpenPose is the most commonly employed pose estimator. MediaPipe was used instead of OpenPose, which is extremely computationally expensive and complex to set up. This design choice permitted a more efficient, smaller-weight implementation. The raw input image used for pose estimation is displayed in Figure 7, while the skeletal keypoint output in the Body-25 format is shown in Figure 8 for further processing.

Figure 7. Input Image

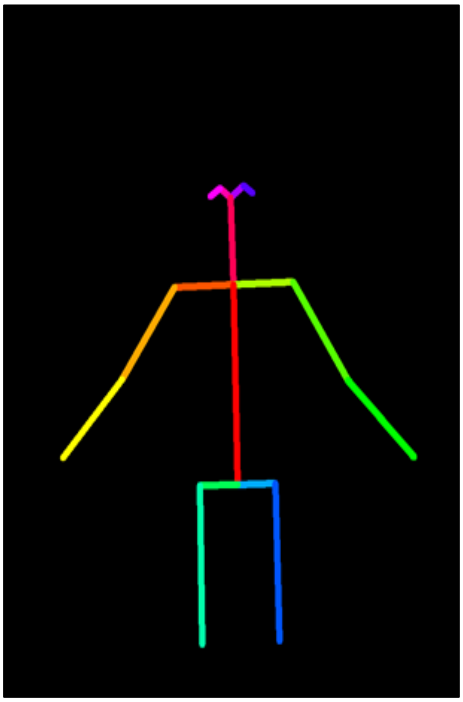

Figure 8. Body-25 Format

Clothes Segmentation:

Clothes segmentation is a computer vision task for segmenting clothes, which involves dividing clothing in an image by recognizing and isolating different articles of clothing precisely. This is often accomplished through machine learning and deep learning-based models like U-Net, Mask R-CNN, and fully convolutional networks (FCNs). The clothes segmentation model is chosen for the best trade-off between speed and accuracy in this research. For real-time or near-real-time virtual try-on applications, the model is appropriate because it can process and segment a single image in about one second.

Figure 9. Target Cloth

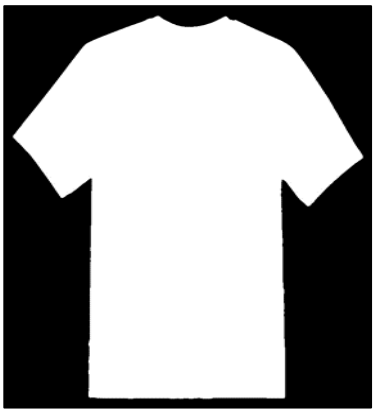

Figure 10. Cloth Segmentation

The target clothing item for virtual try-on is shown in Figure 9, and the matching segmentation mask for isolating the garment during processing is shown in Figure 10.

Human Parsing:

Human parsing is an image segmentation technique that divides the human body into different regions, with each region defining a body part. Each segmented region is assigned a color, providing a detailed understanding of the human image. This segmentation plays an important role in generating accurate virtual try-on outputs. We tried various pre-trained models for this, but their results were not satisfactory enough for the requirements of the VITON-HD model. Therefore, a model using the YOLOv8x architecture was trained with our custom dataset to obtain accurate result. Once the model is trained, it accepts a person's image as input and produces a pixel-wise segmented map where each pixel is assigned to one of the predefined body part categories. Such a segmentation result is employed as input to the pre-processing stage of the VITON-HD pipeline to guide the correct fashion item placement, alignment, and blending in the virtual try-on process for realistic and well-aligned items.

Figure 11. Input Image

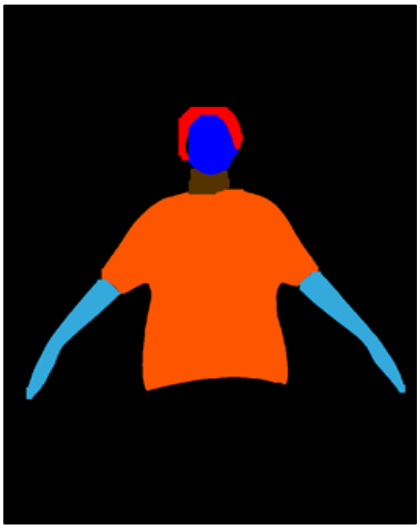

Figure 12. Human Parsing Image

3.2.2 Virtual Try-On Synthesis

The final stage of the virtual try-on pipeline involves the realistic overlay of the target clothing item onto a person's image. To enhance the quality of this overlay, a high-resolution virtual try-on method, VITON-HD, was proposed to overcome the limitations of previous low-resolution approaches (e.g., 256×192) by generating realistic 1024×768 try-on images. To

address misalignment issues and preserve fine clothing details, Alignment-Aware Segment (ALIAS) normalization and an ALIAS generator were introduced [13]. To achieve this, the core stages of the VITON-HD framework were adopted, and its publicly available pretrained checkpoints were utilized to produce high-fidelity try-on results. For tasks such as final image synthesis, geometric matching, and segmentation generation, we used these checkpoints independently. While pre-processing components such as pose estimation and human parsing were developed and trained using a custom dataset with YOLOv8x segmentation, the synthesis stage was executed using the pretrained VITON-HD pipeline to ensure visual realism, structural alignment, and preservation of fine-grained details.

The key stages of the VITON-HD pipeline, which were used in the synthesis process, are given below:

1. **Segmentation Generation:** A person-specific segmentation map was produced based on the clothing-agnostic input through the checkpoint.
2. **Geometric Matching:** The target clothing item was warped to fit the target body using the checkpoint. A Thin-Plate Spline (TPS) transformation was applied to make sure the clothing aligned properly with the body.
3. **ALIAS Synthesis:** The final try-on image was generated using the checkpoint, and the ALIAS (Alignment-Aware Segment) normalization was imposed to preserve fine details and maintain high-resolution reality.

Figure 13. Selecting Input and t-Shirt Image

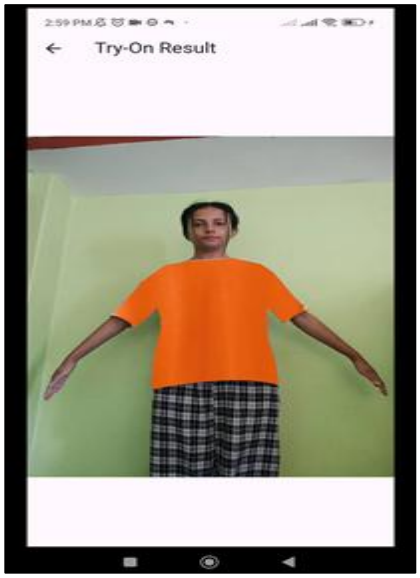

Figure 14. Virtual Try-On Result

The interface for choosing the input person image and the preferred t-shirt for virtual try-on is shown in Figure 13, and the final synthesized output, with the chosen t-shirt realistically superimposed on the person, is shown in Figure 14.

3.3 Principle of Home Decor Visualization Feature

The following steps were performed in order to render home decor objects to provide Augmented Reality visualization:

1. **Dataset Acquisition:** The AR visualization process is initiated by gathering a set of high-quality 3D models that cover different home decor items such as furniture, decorative objects, and household appliances. For this purpose, we used publicly available .glb files from online repositories. The .glb format, which is the binary version of glTF (GL Transmission Format), is particularly used for mobile applications and includes the 3D model geometry, material definitions, textures, and a separate file for each animation. This is highly effective in reducing load time and memory requirements, which is essential to keep the app running smoothly on mobile devices.
2. **Integration with Flutter:** Once the 3D assets were acquired, we integrated them into a mobile application. For 3D model rendering, we used the model-viewer-plus package, which is a powerful way to see 3D objects in glTF and .glb formats directly inside the Flutter UI. It offers interactive functions, including object rotation, zooming, sliding, and lighting adjustments, which help the users look at products from different angles and get a vivid impression of shape, color, size, detail, etc. This kind of interactivity makes it a very worthwhile supplement to enhance the shopping experience, providing confidence to users in their choices before they enter AR mode.
3. **Augmented Reality Implementation:** The AR view was implemented using the model-viewer-plus Flutter package, which provides a cross-platform interface for rendering interactive 3D models in glTF and GLB formats by embedding Google's model-viewer component, enabling real-time placement and visualization of 3D objects in real-world environments using the device's camera [14, 15]. When the user enables the AR mode, the app takes a live video feed of the real-world environment from the device's camera, and the app overlays the selected 3D object on the open space. This type of technology addresses some of the tasks in AR surface detection, environmental joining, object resizing, and so on. Users can navigate around the virtual object, see it

from other angles, adjust its size, and position it elsewhere, replicating what the object will look like in their real home. This lifelike visualization experience offers a tangible extension of the online shopping experience in which consumers can more easily understand how products fit into their private environments. The result is a more immersive experience that helps us make better decisions so we can buy with confidence, and those that actually cut down returns.

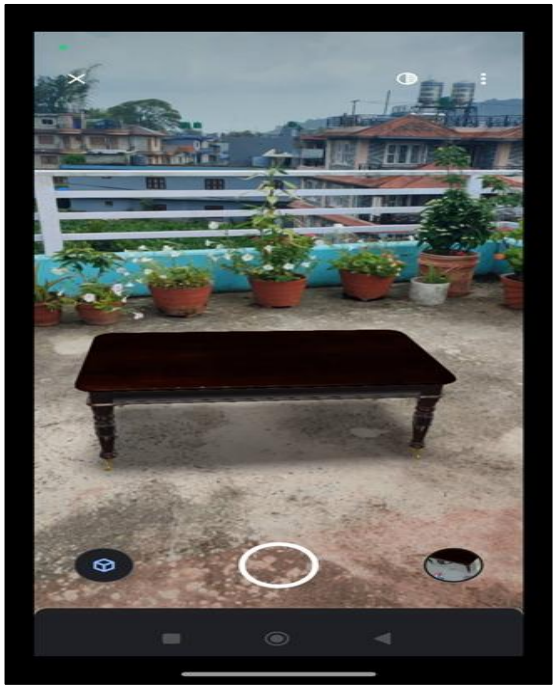

Figure 15. AR View of a Table using Model Viewer

Using Model Viewer, an augmented reality (AR) view of a virtual table placed in a real-world setting that demonstrates realistic spatial integration is shown in Figure 15.

3.4 Principle of Glass Try-On Feature

3.4.1 Facial Landmark Detection using dlib

The implementation of the Glasses Try-On feature begins with accurate facial landmark detection to determine the position and alignment of the eyes. For facial landmark detection in the virtual glass try-on module, we utilized the 68-point facial landmark predictor and the HOG face detector provided by the Dlib library [16]. The input image is first converted to grayscale to improve detection accuracy and efficiency. Using Dlib's Histogram of Oriented Gradients (HOG)-based frontal face detector, the face region is identified. Once the face is detected, 68 facial landmarks are predicted, from which the landmark indices 37 to 42 (for the left eye) and 43 to 48 (for the right eye) are extracted. These eye landmarks are employed to determine the

centers of the left and right eyes, respectively, and to determine the distance between the centers of the two eyes, since these two eye centers become crucial for the adjustment and alignment of an optical overlay of virtual glasses on the face of the virtual mirror. Applying dlib allows for a solid and effective real-time facial feature localization within AR apps like virtual try-on systems.

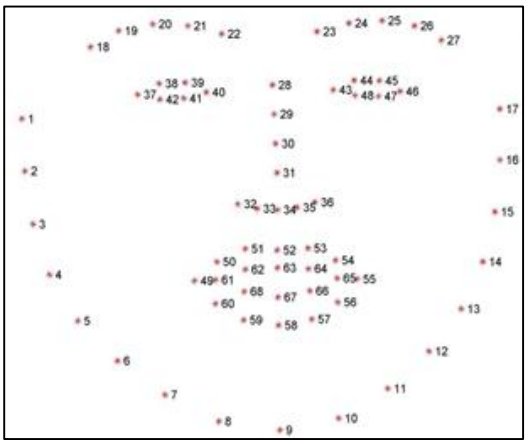

Figure 16. Dlib Facial Landmarks 68 Points Index Reference

Figure 17. Facial Landmark Detection using Dlib's 68-Point Model

Standard landmark locations for facial analysis are shown in Figure 16 using Dlib's 68-point facial landmark index. Figure 17 illustrates accurate facial feature localization on an input image using Dlib's 68-point model for facial landmark detection.

3.4.2 Calculating Position and Size of Glasses

The optimal width and height of the glasses image were estimated based on the extraction of the eye centers and the inter-eye distance. The width of the eye patch is approximately 2.8 times the eye distance being arranged, so the glasses are worn to cover the

eyes naturally, consistently, and straight. The ratio of the glasses image is retained so as not to be distorted. The glasses images were resized with the resize() function of OpenCV, with the top-left positioning coordinates of the glasses calculated based on the left eye center, slightly adjusting the offset to make the placement more realistic.

3.4.3 Overlaying Glasses with Alpha Blending

The resized glasses image (with an alpha channel) was alpha-blended onto the original face image. The Region of Interest (ROI) was extracted for the placement of the glasses in the face image. If the ROI matched the dimensions of the resized glasses, each color channel (R, G, B) was blended with the corresponding channel in the glasses image using the alpha values. This blending maintained the transparency and shape of the glasses, resulting in a natural overlay. The final image, with the glasses applied, was then returned as a PNG image.

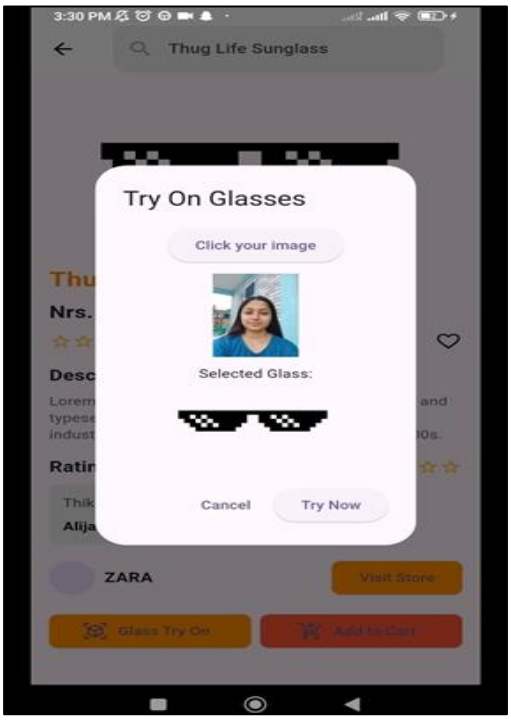

Figure 18. Selection of Input and Glass Image

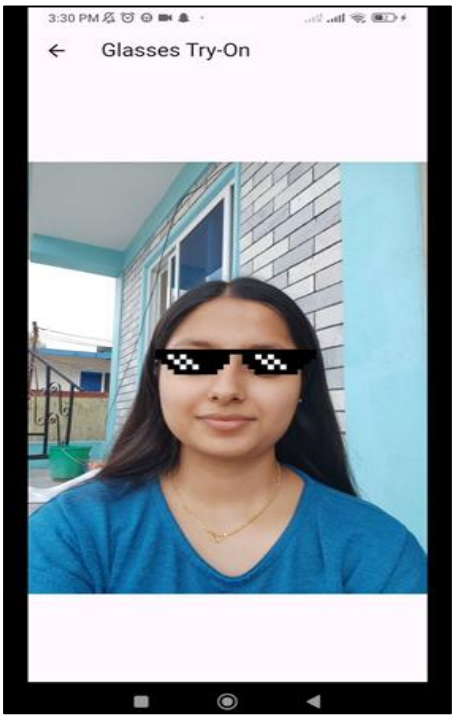

Figure 19. Result of Glass Try-On

3.5 System Work Flow Diagram

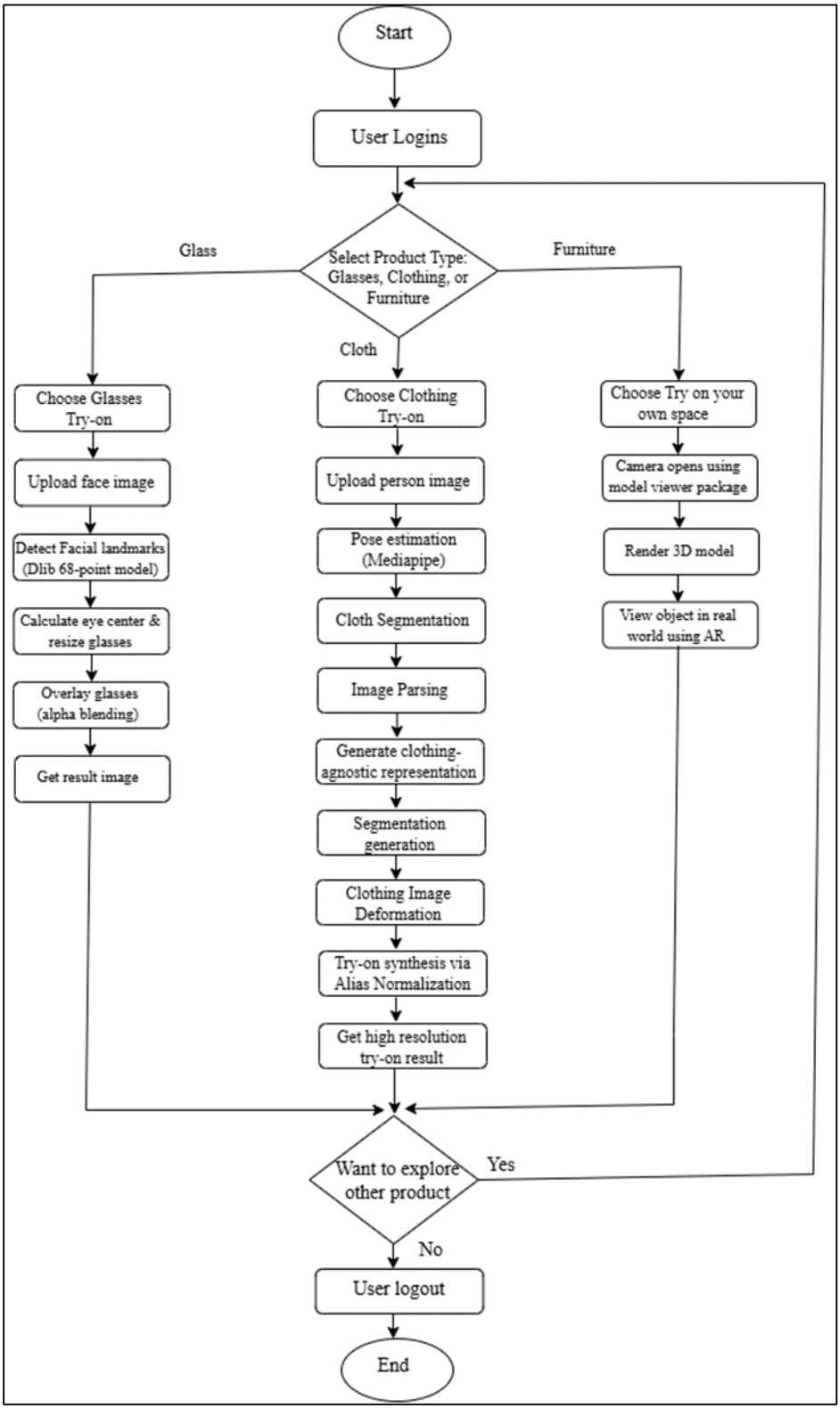

Figure 20. System Work Flow Diagram

4. Results and Discussion

The performance metrics of training and validation were continuously checked over 100 epochs, and the performance of the model was evaluated based various loss functions and metrics such as precision, recall, and mean Average Precision (mAP) as illustrated in figure 21.

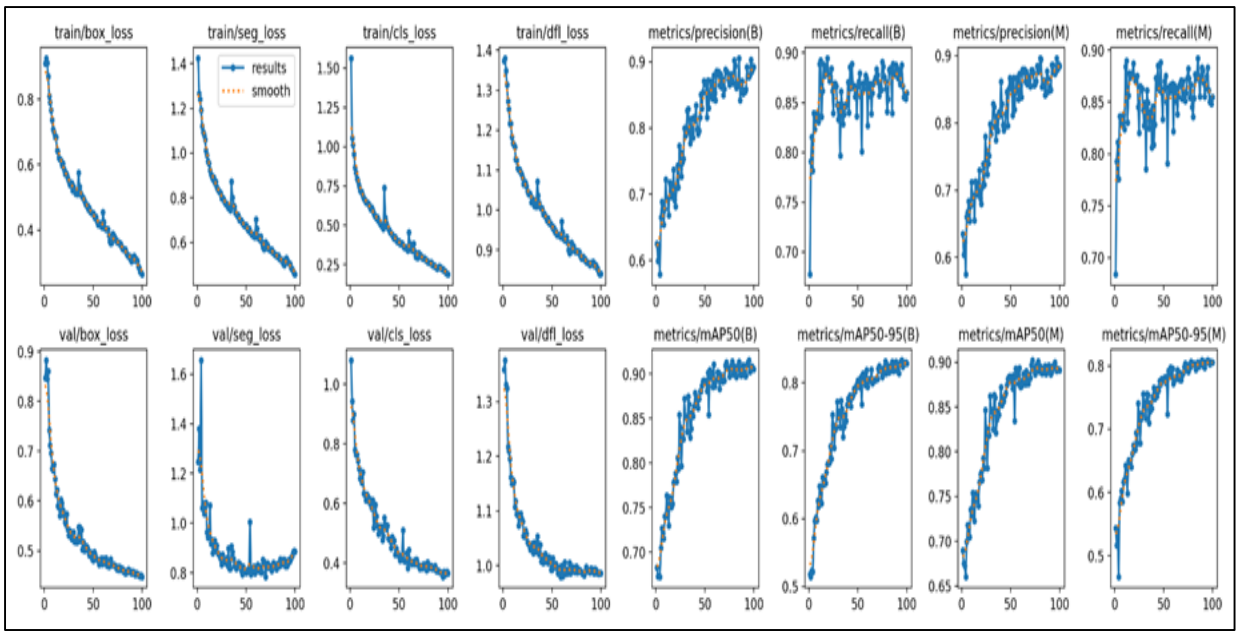

Figure 21. Training and Validation Loss Curves with Evaluation Metrics

4.1 Training and Validation Losses

Box loss decreased steadily during training and even more so in validation, indicating that the model has successfully learned the localization of bounding boxes. The segmentation loss, which was high at first, fell gradually with stabilization, indicating good segmentation. Similarly, the classification loss reduced significantly across epochs, demonstrating the model's success in learning class labels. Furthermore, the Decision Focused Learning (DFL) decreased continuously, which indicates that the model learned class probability distributions for object localization effectively.

4.2 Evaluation Metrics

The precision and recall of Bounding Box (B) and Mask (M) improved continuously during training. The precision of both bounding boxes and masks was around 0.9, while the recall was in the range of 0.85-9. The average precision at a 50% IoU threshold (mAP50) exceeded 0.9 for bounding boxes and was 0.833 for masks, which suggests that the detection

accuracy is high. Additionally, the mAP50-95 metric was strict, resulting in a large gain, which was 0.899 for bounding boxes and 0.808 for masks.

4.3 Performance

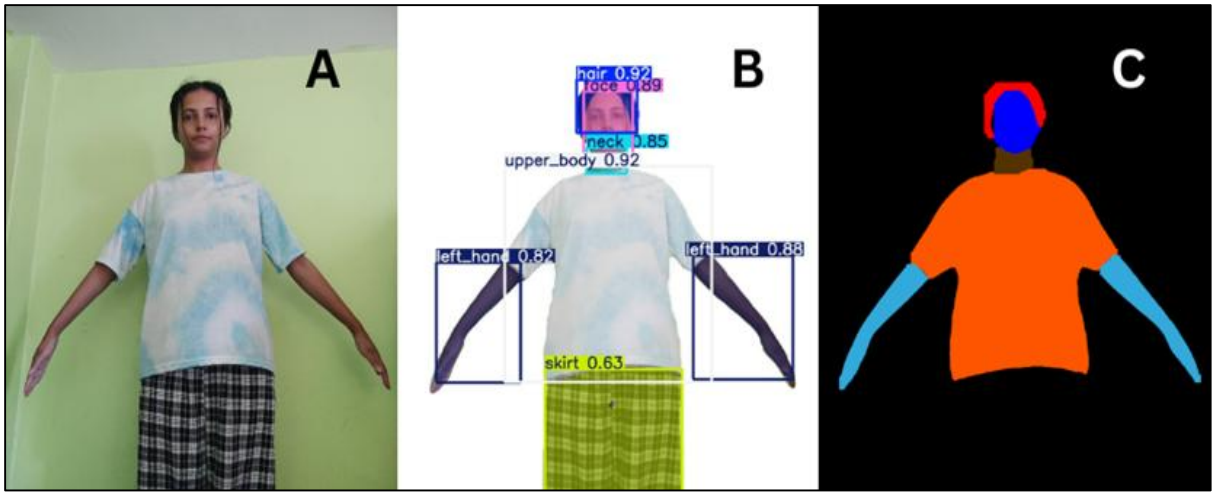

Figure 22. A. Input Image **B.** Inference Image **C.** Human Parse Image

The results of the custom-trained YOLOv8x model, which is crucial for the VITON-HD model, are shown graphically in the figure above (22). The model is fed the first image (A) as its raw input. The bounding box output, where the YOLOv8x model correctly identifies the important body parts, is displayed in the second image (B). The human parse image, which serves as a filter to separate clothing from the body, is finally shown in the third image (C). In order to accurately overlay the new article of clothing on the appropriate body parts, this mask is necessary for the try-on synthesis process. The results provided by the model confirm that it correctly parses input images, which is a key step for achieving realistic virtual try-on outputs.

Table 1. Performance Metrics per Class

Class	Images	Instances	Box(P)	R	mAP 50	mAP 50-95	Mask(P)	R	mAP 50	mAP 50-95
all	200	1371	0.877	0.888	0.915	0.833	0.872	0.883	0.899	0.808
hair	200	200	0.876	0.89	0.932	0.866	0.876	0.89	0.932	0.81
neck	200	200	0.933	0.91	0.93	0.785	0.933	0.91	0.922	0.787
upper_body	194	194	0.913	0.897	0.925	0.877	0.908	0.892	0.925	0.879
lower_body	174	174	0.889	0.891	0.937	0.864	0.884	0.885	0.927	0.867

left_hand	189	189	0.871	0.86	0.909	0.8	0.887	0.876	0.919	0.785
face	200	200	0.997	1	0.995	0.964	0.997	1	0.995	0.982
right_hand	192	192	0.874	0.927	0.958	0.857	0.874	0.927	0.954	0.825
skirt	22	22	0.662	0.727	0.732	0.651	0.62	0.682	0.621	0.531

The table 1 provides the performance of different class labels in our detection and segmentation pipeline. Performance metrics such as precision, recall, and mean average precision (mAP) are listed for each class at two IoU thresholds, one at 50% (mAP50) and another at between 50% and 90% in steps of 5% (mAP50-95) for both bounding box and mask predictions. For the widely represented classes, such as face, hair, and upper body, our model performed well, it models detected faces with nearly perfect precision and recall. However, the performance of the skirt class was low, as it was insufficiently represented in the learned model. All these findings indicate that our model worked well for common and well-defined classes but raises issues for underrepresented ones.

4.4 Confusion Matrix

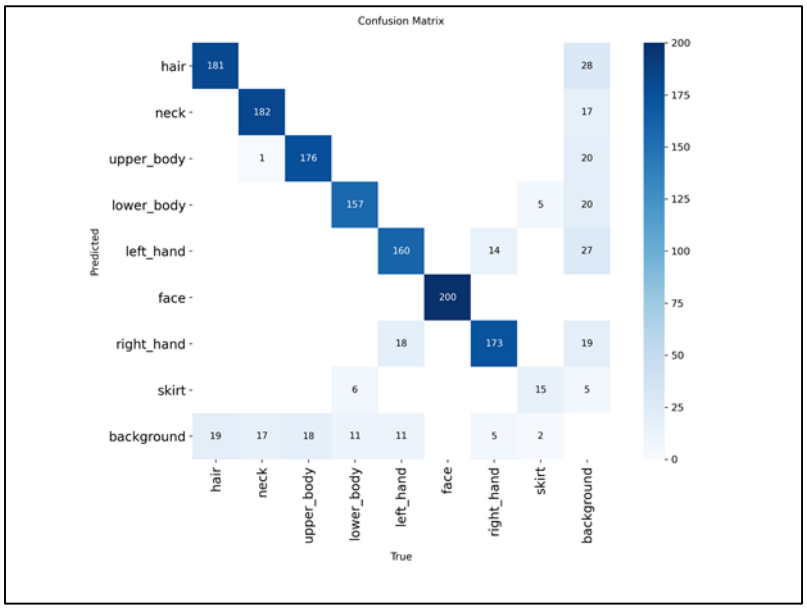

Figure 23. Confusion Matrix During Validation

We examined the confusion matrix shown in figure 23 to confirm the effectiveness of our in-house-trained human parsing YOLOv8x model. The classification output for nine classes face, neck, hair, upper and lower body, left and right hands, skirt, and background is displayed in the matrix. A large diagonal value denotes good classification performance, which

was demonstrated by the model's capacity to identify distinguishing body parts in the face (200 correct classifications), neck (182), and hair (181). Misclassifications point to areas that need work. For instance, there were 20 misclassifications of the upper body and background, 20 misclassifications of the lower body and background, and 14 and 18 misclassifications of the left and right hands, respectively. Visual similarity or partial occlusion in images would cause misclassifications. Furthermore, a slight misunderstanding of the distinction between the foreground and background classes suggests that segmentation boundaries need to be better understood. The model's strong ability to segment human body parts by striking the ideal balance between precision and recall is demonstrated by overall performance, even in the face of obstacles like these.

4.5 Model Performance Evaluation Curves

Four metric curves were used to evaluate the performance of the model, as shown in Figure 24. Both the precision-confidence and recall-confidence curves showed strong confidence in the model, with the majority of the classes displaying high precision and recall of over 0.9 and 0.85 across a wide spectrum of confidence thresholds, respectively. The optimal threshold value on the F1-confidence curve was 0.432, at which the balanced F1-score was 0.88. The precision-recall curve consistently showed high average precision, which was observed for most of the classes, and a mAP50 of 0.915.

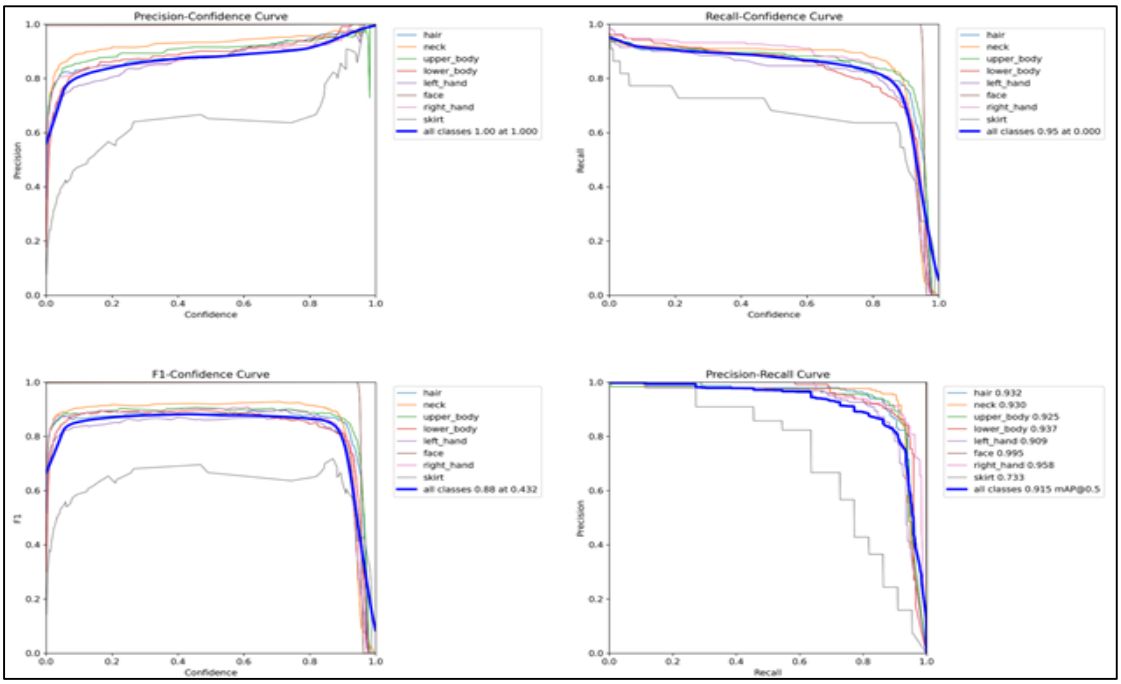

Figure 24. Confidence and Precision-Recall based Evaluation Metrics (Box)

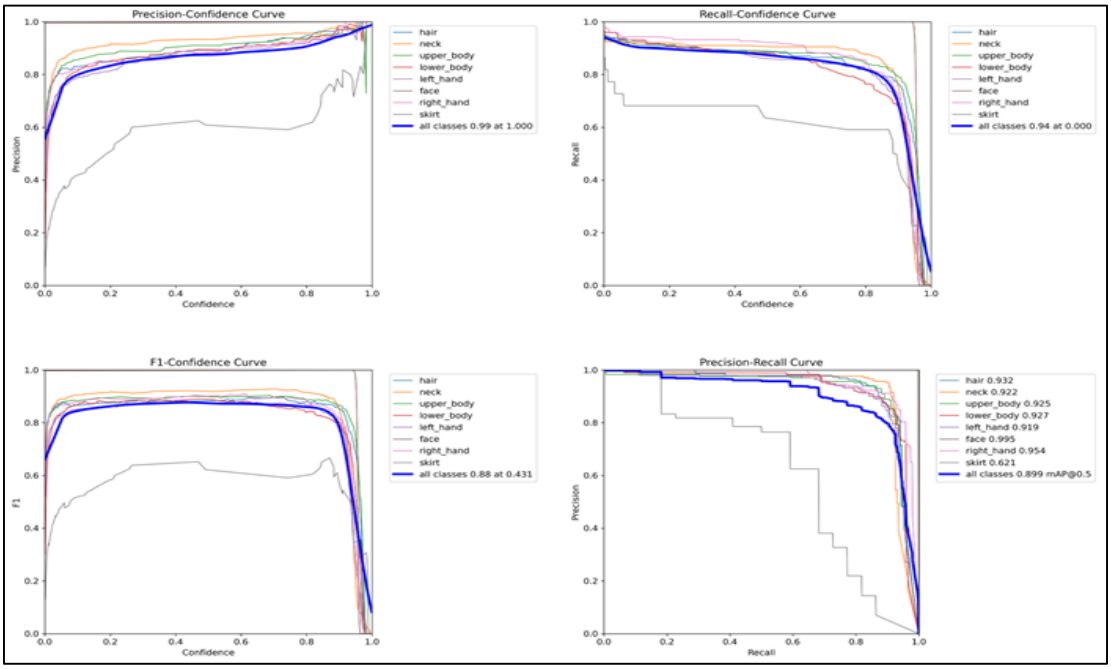

Figure 25. Confidence and Precision-Recall based Evaluation Metrics (Mask)

The experimental results of the custom-trained YOLOv8x model, based on the performance of precision-confidence, recall-confidence, F1-score, and the precision-recall curves, have shown that the overall segmentation performance is strong. The best threshold for a balanced F1-score occurred at a confidence level of 0.431, with an F1-score of 0.88. The mAP50 was 0.899 among all the classes, confirming high reliability as illustrated in figure 25.

5. Limitations

Even though the proposed project offers important features, there are still some aspects that could be improved. Unlike the glasses try-on feature, the current virtual t-shirt try-on does not allow real-time try-on via live camera input. To increase visual realism, the user's image can be further blended with a t-shirt and glasses. Small discrepancies in size perception could result from the AR-based home decor visualization feature's current lack of accurate scaling based on actual room dimensions. These restrictions create opportunities for future development to further enhance the user experience.

6. Future Enhancement

Future improvements will include adding a Virtual Shoe Try-On feature, improving AR visualization by adjusting decor item dimensions to avoid unrealistic scaling while allowing

close-up inspection, and improving the realism of blending clothing and eyewear on the user's face and body.

7. Conclusion

In summary, the strategy presented in this paper provides a way to enhance e-commerce app experiences by enabling virtual try-on capabilities for glasses and t-shirts in addition to AR-grounded visualizing of home furniture items like tables and cabinets. The VITON-HD model, which is specifically made for virtual try-on applications, is what we used in this work. The enhancement of the pre-processing of virtual t-shirt try-ons is the work's most important contribution. An effective cloth segmentation model, pose estimation using Body25 through MediaPipe, and an improved human parsing model that was trained on a custom dataset of 1000 annotated images and validated on 200 images using the YOLOv8x architecture achieved mAP50 of 0.815 and mAP50-95 of 0.833 for bounding boxes and mAP50 of 0.899 and mAP50-95 of 0.808 for segmentation masks. Prior to beginning the project's main phase, we concentrated primarily on getting the input data ready. The VITON-HD-based synthesis process's input quality was greatly enhanced by these enhancements, allowing for more precise and aesthetically pleasing t-shirt overlays. For the virtual glasses try-on feature, a lightweight approach was employed using dlib’s 68-point facial landmark detector to detect the eye regions and achieve precise alignment and realistic placement of 2D glasses through alpha blending. Flutter's model-viewer-plus package was also used to create an augmented reality feature that enables the real-time 3D placement of home décor items in real-world settings. Overall, by eliminating uncertainty and offering a captivating shopping experience, the suggested system shows a successful strategy for raising user confidence and customer engagement in online shopping. Real-time performance enhancements and future extensions to other product categories are also possible with this framework.

References

[1] Alzamzami, Ohoud, Sumaiya Ali, Widad Alkibsi, Ghaidaa Khan, Amal Babour, and Hind Bitar. "Smart fitting: an augmented reality mobile application for virtual try-on." Romanian Journal of Information Technology and Automatic Control 33, no. 2 (2023): 103-118.

[2] Harjati, Fitriana, Yolanda Masnita, and Kurniawati Kurniawati. "Value-Based Adoption Model to Increase Purchase Intention in the Use of Virtual Try-On." Almana: Jurnal Manajemen dan Bisnis 9, no. 1 (2025): 27-41.

[3] An, Shan, Guangfu Che, Jinghao Guo, Haogang Zhu, Junjie Ye, Fangru Zhou, Zhaoqi Zhu, Dong Wei, Aishan Liu, and Wei Zhang. "ARShoe: Real-time augmented reality shoe try-on system on smartphones." In Proceedings of the 29th ACM International Conference on Multimedia, pp. 1111-1119. 2021.

[4] Azevedo, Pedro, Thiago Oliveira Dos Santos, and Edilson De Aguiar. "An augmented reality virtual glasses try-on system." In 2016 XVIII Symposium on Virtual and Augmented Reality (SVR), IEEE, (2016): 1-9.

[5] Balamurugan, S., K. J. Ganesh, M. Rohith Reddy, S. Aadarsh Teja, and M. J. Suganya. "Development of augmented reality application for online trial shopping." In 2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC), IEEE, (2022): 735-740.

[6] Islam, Tasin, Alina Miron, Xiaohui Liu, and Yongmin Li. "Deep learning in virtual try-on: A comprehensive survey." IEEE Access 12 (2024): 29475-29502.

[7] Marelli, Davide, Simone Bianco, and Gianluigi Ciocca. "Designing an AI-based virtual try-on web application." Sensors 22, no. 10 (2022): 3832.

[8] Minar, Matiur Rahman, and Heejune Ahn. "Cloth-vton: Clothing three-dimensional reconstruction for hybrid image-based virtual try-on." In Proceedings of the Asian conference on computer vision. 2020.

[9] Rajapandian, P., Priyadharshini. K, and Tharanie. P. 2024. “VisionCart: Enhancing E-commerce With Augmented Reality for an Immersive Shopping Experience Using Flutter and Arkit.” International Journal for Research in Applied Science and Engineering Technology 12 (5): 2588–92. https://doi.org/10.22214/ijraset.2024.62140.

[10] Samy, Tassneam M., Beshoy I. Asham, Salwa O. Slim, and Amr A. Abohany. "Revolutionizing online shopping with FITMI: A realistic virtual try-on solution." Neural Computing and Applications 37, no. 8 (2025): 6125-6144.

[11] Yuan, Miaolong, Ishtiaq Rasool Khan, Farzam Farbiz, Susu Yao, Arthur Niswar, and Min-Hui Foo. "A mixed reality virtual clothes try-on system." IEEE Transactions on Multimedia 15, no. 8 (2013): 1958-1968.

[12] Shiba, Rana, Ahmed Fadel, Ahmed Elbashaar, Kyrillos Emad, and Rania Elgohary. "Expand: An Immersive Virtual Try-On System with Augmented Reality." International Integrated Intelligent Systems 2, no. 1 (2025).

[13] Choi, Seunghwan, Sunghyun Park, Minsoo Lee, and Jaegul Choo. "Viton-hd: High-resolution virtual try-on via misalignment-aware normalization." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, (2021): 14131-14140.

[14] "Model_Viewer | Flutter Package." n.d. Dart Packages. Accessed August 17, 2024. https://pub.dev/packages/model_viewer.

[15] "3D Model-viewer Embed." n.d. Accessed August 17, 2024. https://modelviewer.dev/.

[16] "Dlib C++ Library." n.d. Accessed December 12, 2024. https://dlib.net/.