

Helwan University
Faculty of Computers and Artificial Intelligence
Computer Science Department



Virtual Try-on

A graduation project dissertation by

Batool Sherif Mohamed	202000195
Beshoy Ibrahim Asham	202000215
Tassneam Mohsen Samy	202000224
Thaowpsta Saiid Aziz	202000233
Ziad Mazhar Mahmoud	202000362
Ali Mohamed Mohamed	202000573

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence at the Computer Science Department, the Faculty of Computers & Artificial Intelligence, Helwan University

Supervised by

Dr. Salwa Osama

June 2024

جامعة حلوان
كلية الحاسبات والذكاء الاصطناعي
قسم علوم الحاسب



Virtual Try-on

رسالة مشروع تخرج مقدمة من

202000195	بتول شريف محمد
202000215	بيشوى ابراهيم عشم
202000224	تسنيم محسن سامى
202000233	ثاؤبستى سعيد عزيز
202000362	زياد مظهر محمود
202000573	على محمد محمد

رسالة مقدمة ضمن متطلبات الحصول على درجة البكالوريوس في الحاسبات والذكاء الاصطناعي،
بقسم علوم الحاسب ، كلية الحاسبات والذكاء الاصطناعي جامعة حلوان

تحت إشراف

د.سلوى اسامة

2024 يونيو

Acknowledgment

We would like to express our deep and sincere gratitude to our supervisor, **Dr. Salwa Osama**, Lecturer in Faculty of Computer and Artificial Intelligence, Helwan University, for giving us the opportunity to unleash our potential and providing invaluable guidance throughout this research while having complete faith in us. Her solid belief in our capabilities, even during moments of uncertainty about the project's direction, has been a constant source of motivation and assurance.

Dr Salwa has taught us the methodology to carry out the research and to present the research work as clearly as possible, thus enabling us to strive for excellence in our academic pursuits.

With her insightful feedback and support, it was a great privilege and honor to work and study under her guidance and we are extremely grateful for what she has offered us, the continuous encouragement, and support in finalizing this project.



Abstract

In today's digital age, consumers increasingly rely on online shopping for convenience and accessibility. However, one significant drawback of online shopping is the inability to physically try on clothing before making a purchase. This limitation often leads to uncertainty regarding style resulting in customer post-purchase dissatisfaction and higher return rates. Based on research, during the pandemic hit, items bought online are three times more likely to be returned than those bought in-store.

FITMI was conceived to bridge the gap between traditional in-store try-ons and online shopping by offering users a realistic and interactive virtual try-on experience.

Although virtual try-ons already exist, recent advancements in Artificial Intelligence have significantly enhanced their capabilities. AI has become increasingly prominent, enabling more sophisticated and realistic virtual try-on experiences than ever before.

Considering these advancements, FITMI goes beyond ordinary virtual try-ons that rely on Generative Adversarial Networks (GANs) which often produce unrealistic outputs. Instead, FITMI utilizes Latent Diffusion Models (LDMs), resulting in high-quality images with detailed textures. As a web application, FITMI facilitates virtual try-on by seamlessly integrating images of users and garments from catalogs providing a true-to-life representation of how the items would look by new approaches that sets us apart from competitors. It also expands as a trusted style advisor, enhancing the journey by recommending complementary items to elevate the chosen garment and suggesting similar options based on user preferences. This comprehensive approach not only empowers users to make informed decisions but also enhances their overall shopping experience.

Abbreviations

AI: Artificial intelligence.

GANs: Generative Adversarial Networks.

LDMs: latent diffusion models.

SDs: Stable Diffusion Techniques.

.npy: NumPy array.

EMASC: the Enhanced Mask-Aware Skip Connection.

PTEs: Pseudo-word Tokens Embeddings.

ATR: The Active Template Regression

LIP: Look Into Person

ViT: Vision Transformer.

CNNs: Convolutional Neural Networks.

MLP: Multi-Layer Perceptron.

LPIPS: Learned Perceptual Image Patch Similarity

SSIM: Structural Similarity

FID: Fréchet Inception Distance

KID: Kernel Inception Distance

Figures

1. VITON-HD dataset categories.	18
2. Dress-Code dataset categories	19
3. Proposed System	25
4. System Architecture.	28
5. System Sequence Diagram.	29
6. System Flowchart Diagram.	30
7. Image reconstruction results with and without the EMASC modules.	33
8. Overview of the proposed model	34
9. preprocessing pipeline.	35
10. Background removal-mask	36
11. Human Parsing model	37
12. Open Pose model.	38
13. detectron2 library for dense pose model.	39
14. Stable diffusion model architecture.	40
15. FITMI's Mobile home screen	47
16. FITMI's Mobile wardrobe screen.	48
17. FITMI's generate your own cloth.	49
18. FITMI's register, login, and profile pages.	50
19. Qualitative results for upper body generated by FITMI.	51
20. FITMI's Recommendation systems for upper body.	52
21. Qualitative results for lower body generated by FITMI.	53
22. FITMI's Recommendation systems for lower body	54



23. FITMI's web interface	55
24. Qualitative results for dresses generated by FITMI	56
25. FITMI's Recommendation systems for dresses.	57
26. FITMI's "Generate Your Own Cloth" page	59

Tables

1. Comparison between Dress Code and the most widely used datasets	20
2. Number of train and test pairs for each category of the Dress Code dataset . .	21
3. Quantitative results on the Viton HD dataset.	60
4. Quantitative results on the Dress Code dataset.	61
5. Quantitative results per category on the Dress Code dataset.	62

Table of contents

Acknowledgment.	3
Abstract	4
List of abbreviations.	5
List of figures.	6
List of tables	8
1. Introduction	
1.1 Comprehensive System Architecture Overview.	11
1.2 Problem Statement	12
1.3 Objective and project scope	12
1.4 Contributions.	13
1.5 Subsequent Chapters	14
2. Related Work (Literature Review)	
2.1 Related Work	15
2.2 Background	
2.2.1 Dataset	17
2.2.2 Machine learning techniques	22
3. The Proposed Solution	
3.1 System Architecture / Proposed System	
3.1.1 Proposed System.	25
3.1.2 System Architecture.	28
3.1.3 Sequence Diagram	29
3.3.4 Flow Diagram	30

3.2 Details of the model and technical aspects	
3.2.1 model.	31
3.2.2 Preprocessing	35
3.2.3 Generate your Own Cloth feature	40
4. Implementation, Experimental Setup, & Results	
4.1 Implementation Details.	42
4.2 Experimental / Simulations Setup	
4.2.1 Experimental setup.	45
4.2.2 Web Application deployment.	46
4.2.3 Mobile Application deployment.	47
4.3 Conducted Results	51
4.4 Testing & Evaluation	60
5. Discussion, Conclusions, and Future Work	
5.1 Discussion	63
5.2 Summary & Conclusion	63
5.3 Future Work	64
6. References	65



1.Introduction

1.1 Comprehensive System Architecture Overview

FITMI is a pioneering application designed and conceived as a solution to revolutionize the virtual try-on experience.

By using cutting-edge techniques such as deep learning and generative networks, FITMI addresses the limitations of traditional virtual try-on methods that are reliant on Generative Adversarial Networks (GANs), which often produce low-resolution and unrealistic outputs.

FITMI employs recent advancements in generative architectures, specifically latent diffusion models (LDMs). These models operate differently from GANs, resulting in high-quality, lifelike images with detailed textures.

FITMI's distinctive approach includes the integration of latent diffusion models and textual inversion components. This combination enables the generation of lifelike representations of users wearing in-shop garments, providing an accurate experience.

Extensive testing and validation have been conducted to ensure that FITMI delivers superior performance compared to existing virtual try-on solutions.

1.2 Problem Statement

Despite the growing popularity of online shopping, the inability to physically try on clothing remains a significant challenge, leading to uncertainties about fit and style and resulting in higher return rates and customer dissatisfaction. Existing virtual try-on methods often produce low-resolution and unrealistic outputs. This limitation requires the development of a novel solution to offer users a realistic and interactive virtual try-on experience. FITMI aims to bridge the gap between traditional in-store try-ons and online shopping by providing high-quality, lifelike representations of users wearing in-shop garments, thereby addressing the challenges associated with online apparel shopping.

1.3 Objective and project scope

FITMI targets consumers who engage in online shopping. These users often face challenges associated with the inability to physically try on clothing before making a purchase, leading to uncertainties about fit and style.

It aims to provide a high quality and accurate solution other than already existing traditional ones for a realistic and interactive virtual try-on experience that satisfies the users' needs.

Additionally, FITMI may also cater to businesses in the apparel industry, including online retailers, by providing a platform that enhances the shopping experience for their customers and potentially reduces return rates and post-purchase dissatisfaction.

1.4 Contributions

To sum up, our contributions are as follows:

- We used both datasets Dress code for lower body and dresses and Viton HD dataset for upper body.
- We integrated a category argument for upper, lower, or dresses, utilizing data from both Dress code and Viton HD datasets seamlessly without the need for manual dataset selection.
- We established customized preprocessing pipelines for input garments and person images, tailored to the specifics of Viton HD and Dress code datasets.
- We introduced sophisticated functionalities such as a garment recommendation system based on color and texture resemblances of input garments, alongside a gender-specific complementary items recommendation system.
- We developed a “Generate your own cloth” feature which integrated stable diffusion, prompt engineering, web scraping, and multi-level filtering which can significantly enhance the quality and relevance of generated content. Hence, we have developed two recommendation systems. The first one uses existing data to make personalized suggestions based on user preferences.
The second system, called 'Generate your own Cloth,' uses web scraping to gather real-time information from the internet and provide up-to-date recommendations.
- The Flask backend system employs Flask's file handling capabilities (request, send_file) and PIL for image processing, while managing paths using os and Pathlib. The primary Flask instance is responsible for controlling the secondary Flask instance, which handles all API requests, including the Cloth Generation API, the Recommendation System API, and the Complementary Items API. This architecture is designed to optimize resource usage and enhance performance by dynamically managing server states based on demand.

1.5 Subsequent Chapters

In Chapter 1, we introduced our FITMI project and its approach, including an overview (1.1), the statement of the problem (1.2) and the objectives (1.3).

In Chapter 2, focuses on Related Work, where we review existing research relevant to our study (2.1) and provide background information, including details on the datasets used (2.2.1) and the machine learning techniques applied (2.2.2).

In Chapter 3, we present the proposed solution, which includes the system architecture (Section 3.1.2), system sequence diagram (Section 3.1.3) and system flowchart diagram (Section 3.1.4) a description of our proposed system (Section 3.1.1), and a detailed explanation of the experiment and technical aspects, such as the model employed (Section 3.2.1) and the preprocessing techniques utilized (Section 3.2.2).

In Chapter 4, we dive into our Implementation details (4.1), Experimental Setup (4.2), Results (4.3), and analyze the testing and evaluation process (4.4).

Finally, in Chapter 5, We summarize our findings and draw conclusions (5.1) and propose avenues for future work (5.2).

2. Related Work (Literature Review)

2.1 Related Work

Image-based virtual try-on [1,2,15,16,22,23,24,25] aims to transfer a desired garment onto the corresponding region of a target subject while preserving human pose and identity.

Clothes Deformation: To preserve the details of a clothing item, previous approaches [16] rely on the explicit warping module to fit the input clothing item to a given person's body. One of the pioneering works in this field is VITON [16], a framework composed of an encoder-decoder generator that produces a coarse result further improved by a refinement network that exploits the warped clothing item obtained through a TPS transformation [21]. Although the warping modules have been consistently improved, the misalignment between the warped clothes and a person's body remains and results in the artifacts in the misaligned regions. Moreover, without injecting any person representation, seen poses, or explicitly considering deformations, it fails to determine where the arms and hands of a person should occur to and fails to generate photo-realistic virtual try-on results.

Segmentation Generation for Try-On Synthesis: The high-resolution virtual try-on methods [2,22,24] generally include the segmentation generation module because the importance of the segmentation map increases as the image resolution grows. Recently, VITON-HD [2] proposed a normalization technique to alleviate the issue. However, we found that the normalization method fails to naturally fill the misaligned regions with clothing texture. A common aspect linking all current methods is that the generation phase relies on GANs [22,23], where Lee et al. [22] in HR-Viton solved the misalignment problem by designing a unified pipeline that combines the warping and segmentation stages to achieve better high-resolution results. Another method, ACGPN [24], mainly mask the clothing region of the person image and reconstruct the person image with the corresponding clothes image, which require accurate human parsing but fail to preserve the characteristics of the target clothes since they excessively focus on the silhouette of the original clothes during training. Unfortunately, methods that

are heavily based on human parsing, slightly wrong segmentation results would lead to unrealistic try-on images with large artifacts.

Generation phase and refinement of the result: Another research line focuses on the generation phase and refinement of the result [1,23], where Morelli et al. [1] in dress code focused on the semantics of the generated results and proposed a semantic-aware discriminator working at the pixel level instead of the image or patch level, but it shows Low-Resolution Results in some failure cases.

Parser free method: the pioneering parser-free methods [23,25] produce noticeable artifacts due to using the same input-output pairs for both Teacher and Student networks. Addressing that issue, Yuying Ge [23] introduces PF-AFN, a new Knowledge Distillation-based training pipeline, in which the student network takes the Teacher output as its input and has its output supervised by the original images. This training methodology has become the standard for subsequent parser-free methods. RMGN [25] improves the generation part by using SPADE blocks [26], However, these methods are still based on the parser-based approach, which takes considerable time to calculate the human representation.

During the last years, a new family of generative architectures, namely diffusion models have shown superior image generation quality compared to GANs. However, considering the high computational demand typical of diffusion models, Rombach et al. [17] have recently tackled the problem by introducing a latent-based version that works in the latent space of a pre-trained autoencoder, thus finding the best trade-off between computational load and image quality. Building on this, the LaDI-vton [18] model further adapted LDMs by effectively using latent space for more detailed and controllable image synthesis. That's why our model, FITMI, is built upon the pioneering laDI-vton [18] the first latent diffusion-based approach for virtual try-on, that outperforms by a large margin the competitors in terms of realism on both Dress Code and VITON-HD datasets, two widely used benchmarks for the task where we enhanced its methodology and incorporated novel fundamentals.

2.2 Background

2.2.1 Dataset

Due to the strategic role that virtual try-on plays in e-commerce, many rich and potentially valuable datasets are proprietary and not publicly available to the research community. Public datasets, instead, either do not contain paired images of models and garments or feature a very limited number of images.

Moreover, the overall image resolution is low; mostly 256×192 . Unfortunately, these drawbacks slow down progress in the field. So, we extensively validate our architecture on two widely used virtual try-on benchmarks (i.e., Dress-Code and VITON-HD), as it outperforms by a large margin the competitors in terms of realism on both datasets. Images of both datasets have a great variety considering both the body pose of the reference models and category and textures of try-on garments. This can lead to virtual try-on architectures becoming more general and adapting to more challenging scenarios.

1. VITON-HD:

VITON-HD is a 1024×768 virtual try-on dataset which contains only upper-body clothes with 13,679 frontal-view woman and top clothing image pairs. The pairs are split into a training and a test set with 11,647 and 2,032 pairs, respectively. One can either use the pairs of a person and a clothing image to evaluate a paired setting or shuffle the clothing images for an unpaired setting. The paired setting is to reconstruct the person image with the original clothing item, and the unpaired setting is to change the clothing item on the person image with a different item.

It utilizes a blend of seven techniques in fig.2 including masks to differentiate the person from the background, Dense Pose and Open Pose for detailed pose estimation, and parsing representations for segmenting the body into distinct parts. The system also incorporates agnostic masks and parsing to focus on body shapes rather than specific clothing, and an agnostic representation of the person that abstracts away physical and clothing details using pose and segmentation maps. Additionally, it includes a specific mask for the clothing items.

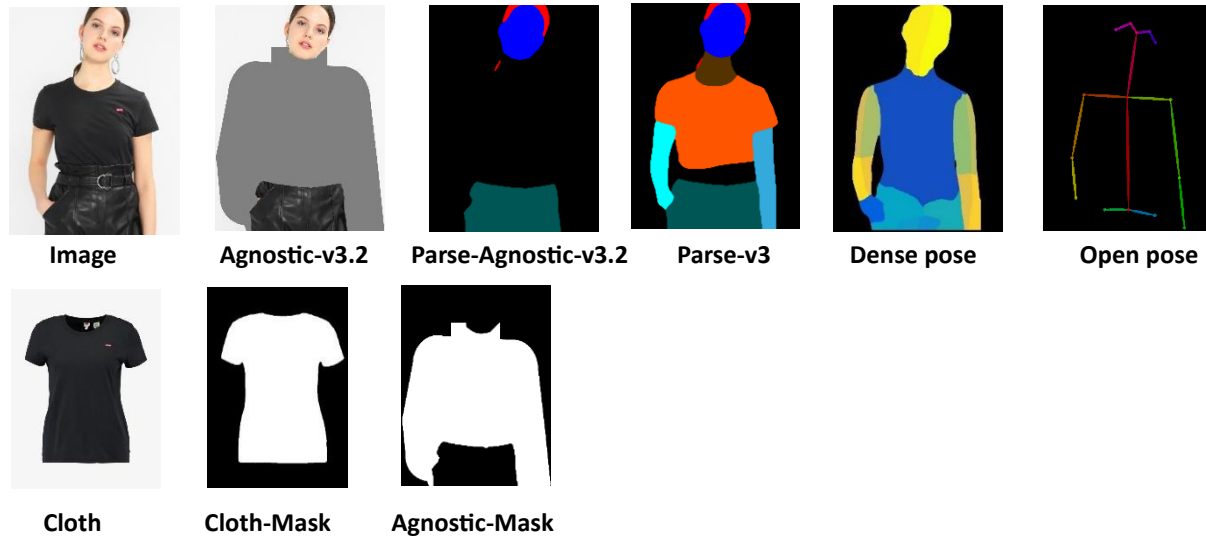


Figure 1. VITON-HD dataset categories

2. Dress-Code:

Dress-Code is more than 3× larger than publicly available datasets for image-based virtual try-on and features high-resolution paired images (1024×768) with front-view, full-body reference models containing more than 50k image pairs of try-on garments and corresponding catalog images where each item is worn by a model. A dataset for virtual try-on with a large number of images is more preferable than other datasets with the same overall characteristics but smaller size. It is the first publicly available dataset featuring lower-body and full-body clothes. As a plus, all images have high resolution (1024×768). This makes Dress-Code more than 3× larger than VITON-HD which contain only upper-body clothes, Dress Code features three categories: upper-body (composed of tops, T-shirts, shirts, sweatshirts, and sweaters), lower-body (composed of skirts, trousers, shorts, and leggings), and full-body clothes (composed of dresses), as well as full body images of human models. To preserve the models' identity, we partially anonymize all images by cutting them at the level of the nose. In this way, information about the physiognomy of the human models is not available. Overall, the dataset is composed of 53,795 image pairs: 15,366 pairs for upper-body clothes, 8,951 pairs for lower-body clothes, and 29,478 pairs for dresses.

Dress-Code contains images paired with corresponding clothing items that undergo six preprocessing steps, including joint coordinates in (.npy) loading arrays, which likely denote key points on the body for pose estimation. Moreover, the dataset includes dense-pose information, which maps image pixels to the 3D surface of the human body for detailed pose understanding. Additionally, it offers cloth align masks and cloth align parses, likely to assist in aligning the clothing properly on the body. Furthermore, there are cloth warped images, which probably depict the clothing items adjusted to fit the body shape. Finally, the dataset provides parsing information of the person, segmenting the image into different semantic parts, for all three categories of clothing: upper, lower, and dresses.

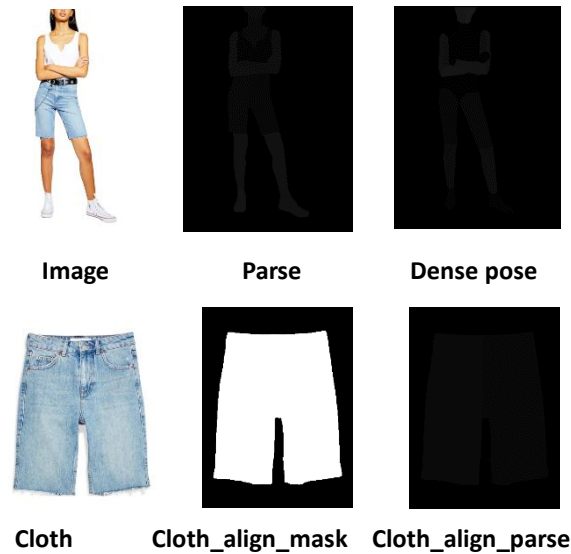


Figure 2. Dress-Code dataset categories

Table 1 reports the main characteristics of the Dress Code dataset in comparison with existing datasets for virtual try-on and fashion-related tasks. Although some proprietary and non-publicly available datasets have also been used, almost all virtual try-on literature employs the VITON dataset [16] to train the proposed models and perform experiments. We believe that the use of Dress Code could greatly increase the performance and applicability of virtual try-on solutions. In fact, when comparing Dress Code with the VITON dataset, it can be seen that our dataset jointly features a larger number of image pairs (i.e. 53,795 vs 16,253 of

the VITON dataset), a wider variety of clothing items (i.e. VITON only contains t-shirts and upper-body clothes), a greater variance in model images (i.e. Dress Code images can contain challenging backgrounds, accessories like bags, scarfs, and belts, and both male and female models), and a greater image resolution (i.e. 1024×768 vs 256×192 of VITON images)

Table 1. Comparison between Dress Code and the most widely used datasets for virtual try-on and other related tasks.

Dataset	Public	Multi -Category	# Images	# Garments	Resolution
O-viton [7]	×	✓	52,000	-	512 x 256
TryOnGAN [8]	×	✓	105,000	-	512 x 512
Revery AI [9]	×	✓	642,000	321,000	512 x 512
Zalando [10]	×	✓	1,520,000	1,140,000	1024 x 768
VITON -HD [2]	✓	×	27,358	13,679	1024 x 768
FashionOn [11]	✓	×	32,685	10,895	288 x 192
DeepFashion [12]	✓	×	33,849	11,283	288 x 192
MVP [13]	✓	×	49,211	13,524	256 x 192
FashionTryOn [14]	✓	×	86,142	28,714	256 x 192
LookBook [15]	✓	✓	84,748	9,732	256 x 192
VITON [16]	✓	×	32,506	16,253	256 x 192
Dress Code [1]	✓	✓	107,584	53,795	1024 x 768

Table 2. Number of train and test pairs for each category of the Dress Code dataset.

	Images	Training pairs	Test pairs
Upper body	30,736	13,563	1,800
Lower body	17,902	7,151	1,800
Dresses	58,956	27,678	1,800
All	107,584	48,392	5,400

In conclusion, we chose Viton HD and Dress-Code datasets for their extensive coverage, detailed annotations, and suitability for virtual try-on applications. While Viton HD offers paired evaluation settings and focuses on upper-body clothing, Dress-Code stands out for its larger size, covering a variety of clothing categories including lower-body and full-body garments. Both datasets provide essential preprocessing steps and annotations, ensuring accurate alignment of clothing items and realistic simulations. Overall, these datasets offer comprehensive data for training and evaluating our virtual try-on model effectively.

2.2.2 Machine learning techniques

Vision Transformer (ViT):

ViT is a state-of-the-art architecture for image classification and understanding, originally introduced for processing visual data using transformers.

ViT layer is employed as part of the textual inversion network to extract visual features from garments and generate pseudo-word token embeddings (PTEs) in the CLIP token embedding space representing the garments.

Multi-Layer Perceptron (MLP):

MLPs are a class of feedforward neural networks consisting of multiple layers of nodes (or neurons), each connected to the next layer. Within the textual inversion network, MLPs are employed after the ViT layer to process visual features and generate pseudo-word token embeddings.

Latent Diffusion Models (LDMs):

LDMs Forms the basis for performing the virtual try-on task in the project. They are utilized for image inpainting, specifically in replacing garments in human-based images with target garments provided by users.

Our proposed method builds upon the Stable Diffusion which is a latent text to image diffusion model, which consists of an autoencoder with encoder and decoder components, a text time-conditional U-Net denoising model, and a CLIP text encoder.

Stable diffusion techniques (SDs)

SDs, in image generation specifically refer to methods used in neural network-based models to produce high-quality images. These techniques involve a process where an initial low-resolution image is progressively refined through multiple

steps, enhancing both the visual quality and level of detail. Each step involves applying noise to the image and allowing it to diffuse or spread, which helps in maintaining stability and fidelity during the generation process. This approach is particularly effective in generating realistic images with intricate details.

Convolutional Neural Networks (CNNs):

CNNs are fundamental in image processing tasks, known for their effectiveness in feature extraction and pattern recognition and are extensively used within the Stable Diffusion architecture for tasks such as image encoding, decoding, and inpainting. In the Autoencoder with Enhanced Mask-Aware Skip Connection (EMASC) module, CNNs are used to implement convolutional layers for enhancing image reconstruction within the Stable Diffusion model.

Enhanced Mask-Aware Skip Connections (EMASC):

EMASC modules are introduced to enhance the autoencoder's reconstruction capabilities. These modules propagate relevant information from different layers of the encoder to corresponding layers of the decoder, focusing on preserving details during the inpainting process.

AdamW regularization Algorithm:

AdamW is a variant of the Adam optimization algorithm that incorporates weight decay directly into the optimization step, which helps prevent overfitting by penalizing large parameter values.

It is utilized as the optimizer during training of both the textual inversion network and the EMASC modules, ensuring efficient convergence and parameter updates.

L1 and VGG Loss Functions:

L1 and VGG loss functions' purpose is to Optimize the reconstruction quality of the Stable Diffusion model.

L1 loss (Mean Absolute Error) measures the absolute difference between predicted and target values.

VGG loss, also known as perceptual loss, is computed using a pre-trained VGG network, which captures high-level image features. It involves passing both the predicted and target images through the VGG network and comparing the high-level feature representations extracted from specific layers. By comparing the feature representations rather than pixel-wise differences, VGG loss encourages the model to capture perceptually meaningful details and textures in the reconstructed images.

These loss functions are used during training of the EMASC modules to optimize the reconstruction quality of the Stable Diffusion model.

3.1 System Architecture / Proposed System

3.1.1 Proposed System

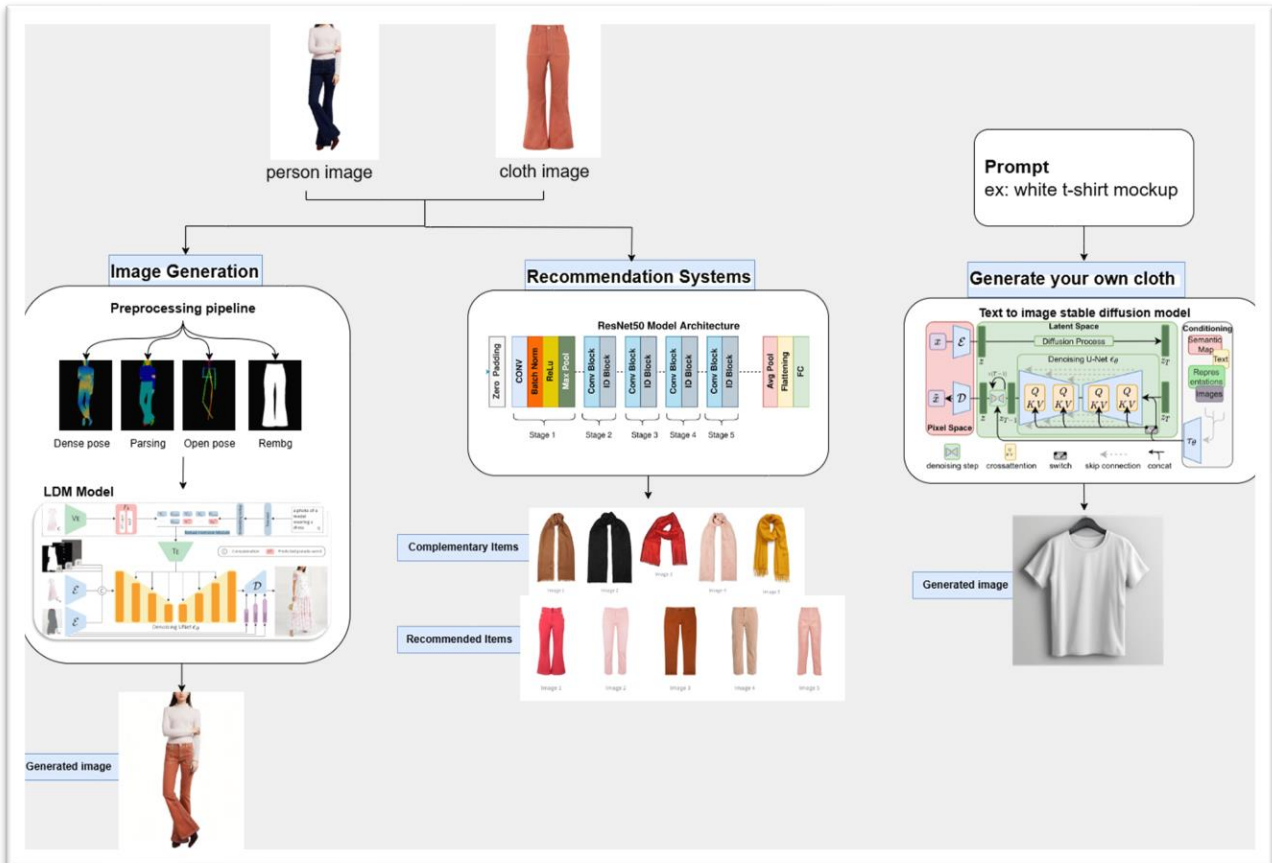


Figure 3. Proposed System

Our proposed system as shown in fig.3 presents a novel approach to virtual try-on by integrating Latent Diffusion Models (LDMs) with advanced techniques in image-text fusion. At its core, the system leverages Stable Diffusion, an LDM architecture well-known for its superior image generation capabilities over traditional Generative Adversarial Networks (GANs). Building upon this foundation, we introduce enhancements tailored specifically for virtual try-on scenarios. By incorporating dual inputs - the target garment and the pose of the model - our system achieves a detailed and realistic depiction that preserves the model's physical characteristics, pose, and identity. The integration of a text time-

conditional U-Net denoising model and a CLIP text encoder further enhances the precision of garment details and textures, crucial for maintaining fidelity in virtual try-on applications. Additionally, our system features an Enhanced Mask-Aware Skip Connection (EMASC) module to address challenges related to high-frequency detail retention, ensuring the preservation of intricate features during the image reconstruction phase.

Preceding these groundbreaking advancements is our comprehensive preprocessing stage, essential for accurate data classification and pattern recognition within our datasets. This preprocessing workflow orchestrates a symphony of four distinct models:

—Mask and Background Removal, Parsing, Open Pose, and Dense Pose—. Each meticulously integrated together to execute consecutively.

Complementing these technical details are the user-centric features embedded within our FITMI application. Leveraging advanced machine learning algorithms, FITMI offers personalized garment recommendations based on user preferences.

We used the Dress Code dataset categories (upper body, lower body and dresses) and generated its embeddings using a pre-trained ResNet50 convolutional neural network (CNN) model to extract image features and the extracted features were normalized and saved as embeddings.

Furthermore, it goes beyond individual garment suggestions by providing complementary item recommendations, ensuring users can effortlessly complete their desired looks with curated accessories, shoes, or outerwear that caters to users of all genders equally. We also generated embeddings for complementary items using the same process used for our garment recommendation, but other datasets were collected. The datasets were curated from the Real Fashion dataset [19], which focuses on specific categories such as accessories, footwear, and outerwear. Additionally, datasets were assembled for both genders, ensuring that the embeddings captures gender-specific preferences and styles.

Additionally, we introduced 'Generate Your Own Cloth', a feature that utilizes web scraping to create personalized clothing photos based on user descriptions [37].



These photos can be seamlessly transferred to a virtual wardrobe, allowing users to try on their custom designs virtually.

3.1.2 System Architecture

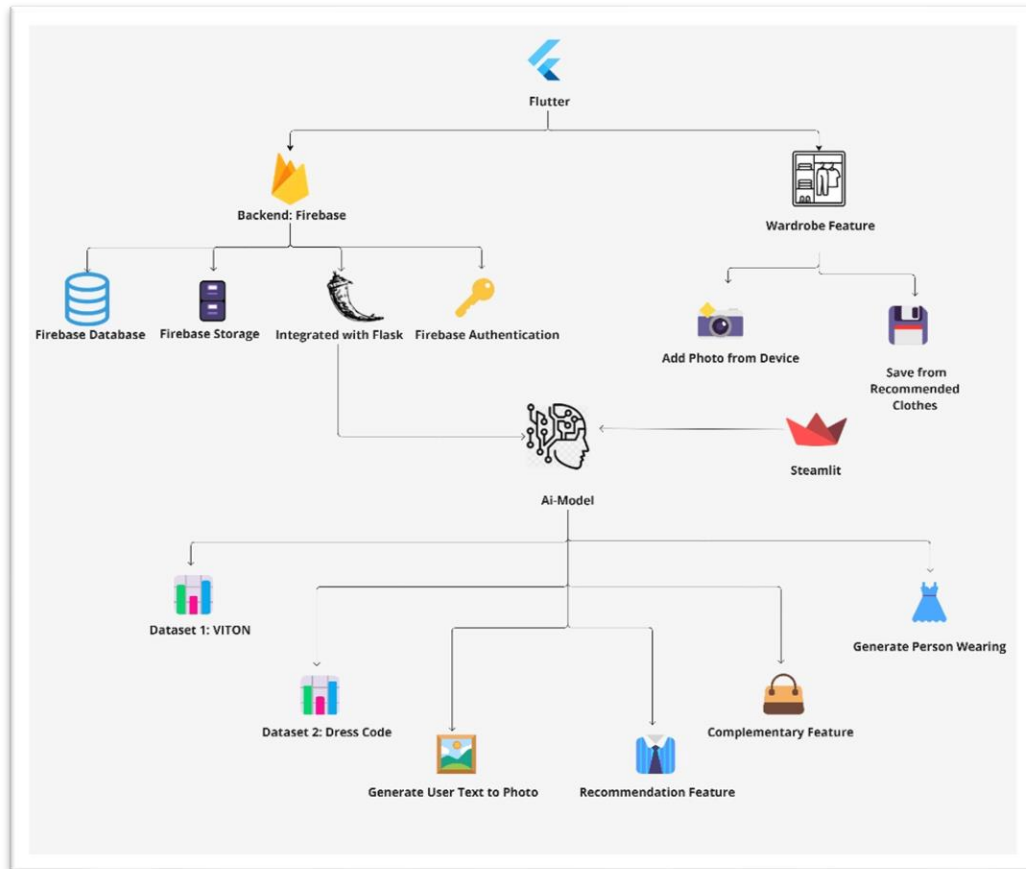


Figure 4. System Architecture

Figure 4 illustrates a Flutter-based application integrated with Firebase for wardrobe management and AI-powered clothing recommendations. The frontend is built with Flutter, providing features for managing a wardrobe by adding photos from the device or saving recommended clothes. The backend utilizes Firebase for data storage, media handling, and user authentication, with Flask facilitating AI model integration. The central AI model processes inputs using VITON and Dress Code datasets to generate visual representations, person-wearing images, and clothing recommendations, enhanced by complementary item suggestions.

Streamlit serves as the interface for interactive user engagement with the AI model.

3.1.3 Sequence Diagram

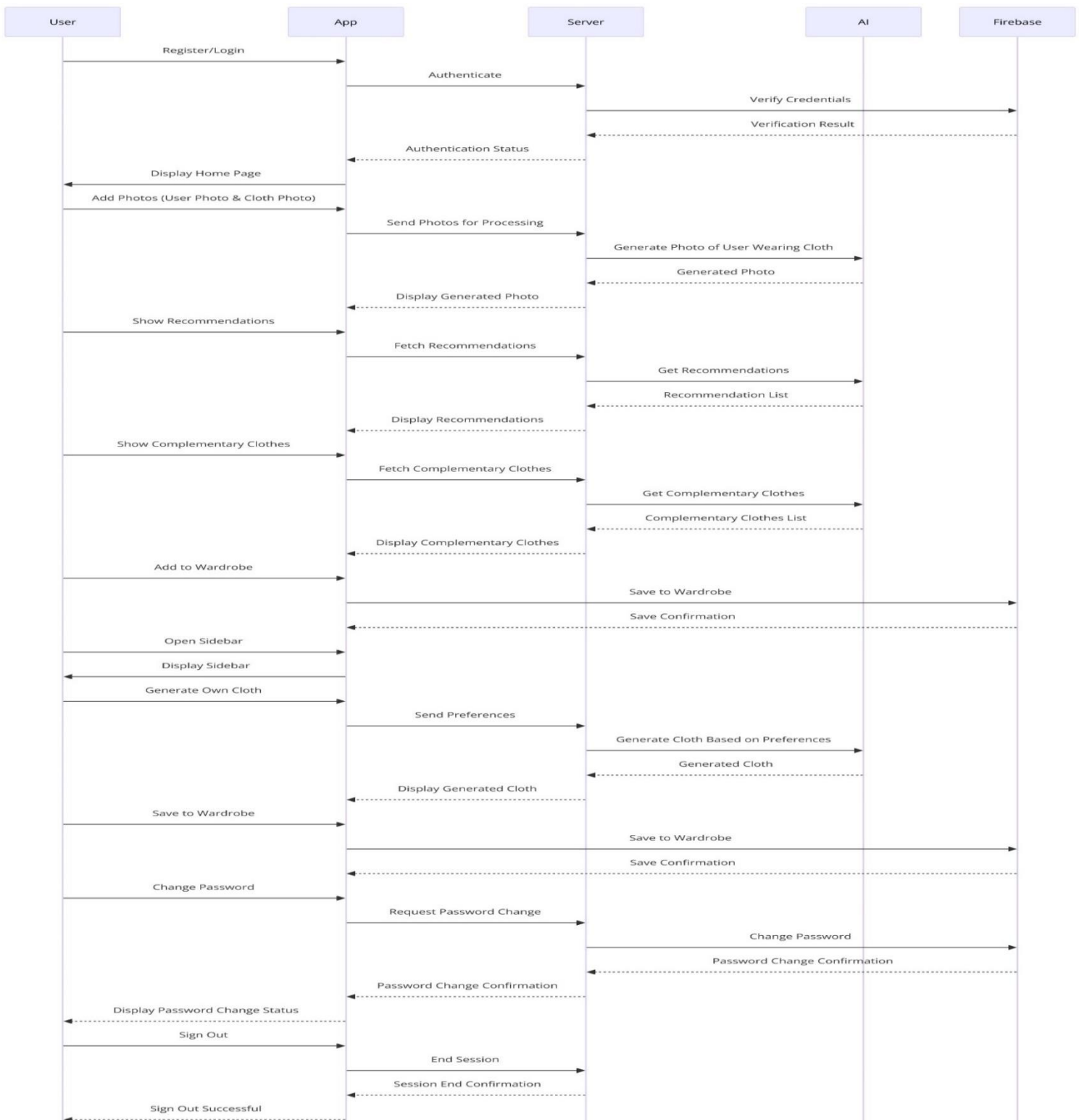


Figure 5. System Sequence Diagram

3.1.4 Flow Diagram

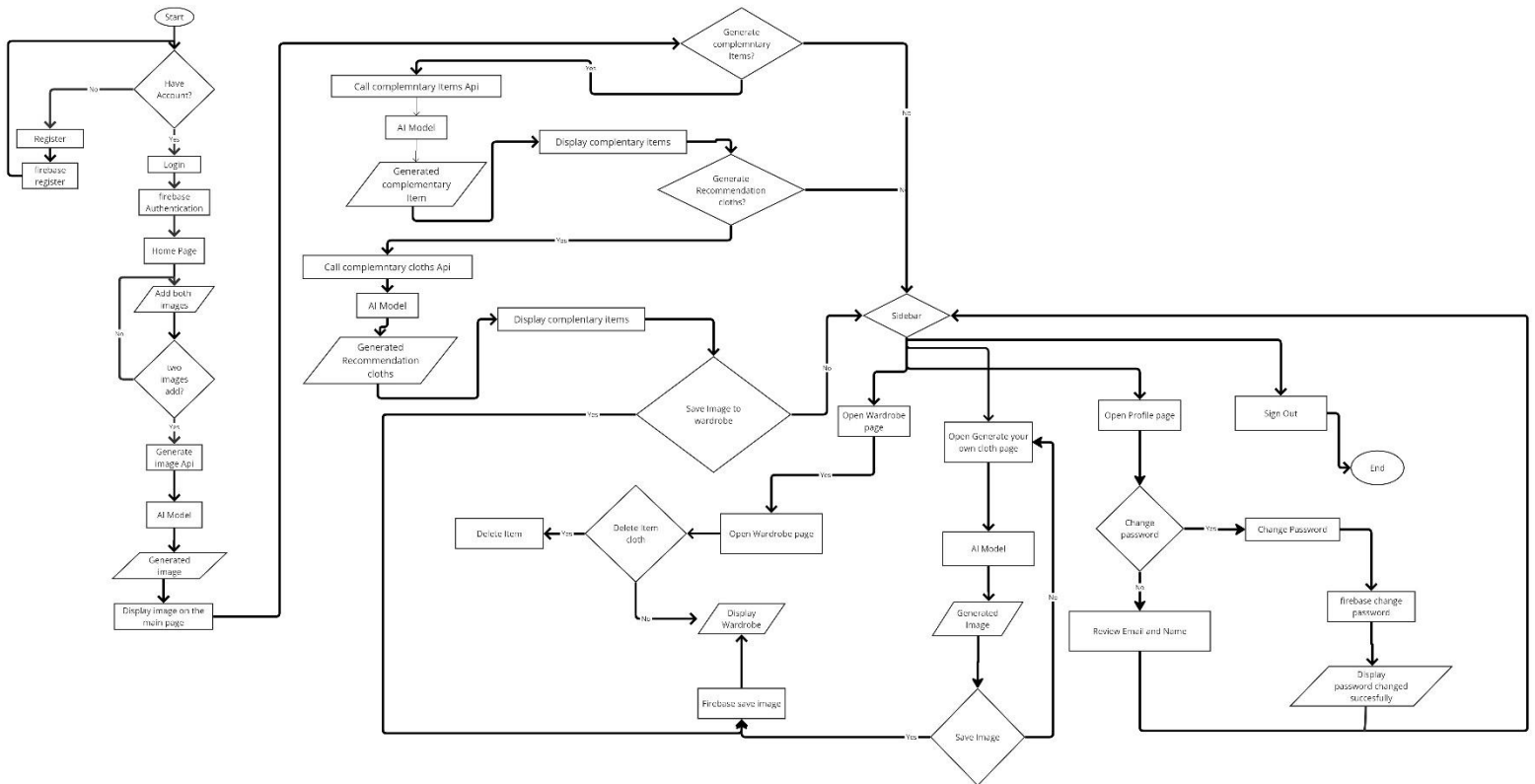


Figure 6. System Flowchart Diagram

The above flowchart Fig.6 illustrates the comprehensive workflow of a mobile application, starting from user authentication, where users either register or log in using Firebase Authentication. Once logged in, users can add images and generate new ones through an API, which are then displayed on the main page. The application also features an AI-powered system to generate and display complementary items and recommended clothes. Users can save these items to their wardrobe, view or delete them, and manage their profile by updating information or changing passwords. The process concludes with the user signing out, completing the application's user journey.

3.2 Details of the model and technical aspects

3.2.1 Model

Our approach innovatively integrates Latent Diffusion Models (LDMs) with the latest advancements in image and text integration, drawing inspiration from successful precedents and extending their capabilities.

The backbone of our method lies in the adoption of LDMs, which have been shown to surpass the capabilities of traditional Generative Adversarial Networks (GANs) in image generation tasks. This shift was initially inspired by the work presented by Rombach et al. [17], which mitigated some of the inherent limitations of GANs. Building on this, the LaDI-vton [18] model further adapted LDMs by effectively using latent space for more detailed and controllable image synthesis. To better accommodate the complexities of virtual try-on, we have modified the standard input structure of these models. Moving beyond the typical single-input systems that focus solely on the garment, our model uniquely incorporates dual inputs: the garment itself and the pose of the target model. This enables a more detailed depiction that not only conforms the garment to the model but also precisely aligns it with the model's pose, ensuring that the physical characteristics, pose, and identity are convincingly maintained.

Latent Diffusion Models ‘Stable Diffusion’:

Understanding the underlying components of Stable Diffusion, an LDM architecture is essential. The model comprises an autoencoder with distinct encoder and decoder components, a text time-conditional U-Net denoising model, and a CLIP text encoder. Figure 8 depicts an overview of the proposed model.

text time-conditional U-Net denoising model paired with a CLIP text encoder:

Central to our method is the refined use of a text time-conditional U-Net denoising model paired with a CLIP text encoder within the diffusion model

framework. The denoising network, which is pivotal in the diffusion process, minimizes a loss function associated with reconstructing images that have been deliberately corrupted with Gaussian noise. Furthermore, the CLIP model, A vision-language model integral to the textual inversion technique aligns visual and textual data within a unified embedding space where visual features of the garment are encoded as textual token embeddings. We refer to these embeddings as Pseudo-word Tokens Embeddings (PTEs) since they do not correspond to any linguistically meaningful entity but rather are a representation of the in-shop garment visual features in the token embedding space. Such an arrangement allows the model to fasten the diffusion process to these tokens, facilitating a more seamless integration of text and image features, and enhancing the precision of the garment's details and textures in the final output. This is fundamental in maintaining the integrity and fidelity of complex textile patterns and colors in virtually tried-on clothing.

Enhanced Mask-Aware Skip Connection module:

As we said LDMs are augmented with a textual inversion network. However, LDMs struggle with high-frequency details in the pixel space due to spatial compression performed by the autoencoder, consequently, to address the challenges of high-frequency detail retention of LDMs in areas like hands, feet, faces, and particularly in areas where the new garment is incorporated through techniques that can be seen as a particular type of inpainting that is specialized in replacing garment information in human based images according to a target garment image provided by the user, we have implemented the Enhanced Mask-Aware Skip Connection (EMASC) module. EMASC is an architectural improvement to boost the model's performance further. Notably, masked skip connections are incorporated within the autoencoder structure of the Stable Diffusion model. These connections are crucial for maintaining high-quality image outputs, remarkably through their role in better preserving details during the image reconstruction phase. This enhancement significantly boosts the model's ability to transfer intricate details from the encoding phase back to the decoding phase.



Figure 7. Image reconstruction results from the Stable Diffusion auto encoder with and without the EMASC modules.

The effectiveness of our approach has been rigorously validated against prominent virtual try-on benchmarks such as VITON-HD and Dress Code. Our method demonstrates superior quantitative and qualitative performance compared to existing state-of-the-art techniques.

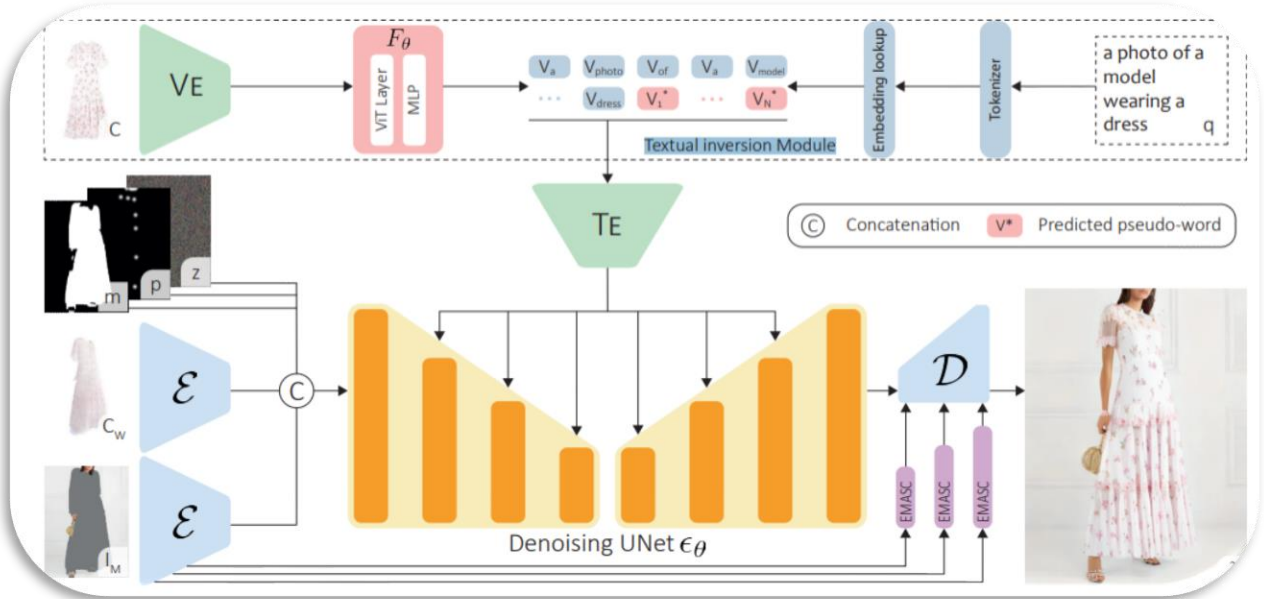


Figure 8. Overview of the proposed model. On the top, the textual inversion module generates a representation of the in-shop garment. This information conditions the Stable Diffusion model along with other convolutional inputs. Decoder \mathcal{D} is enriched with the Enhanced Mask-Aware Skip Connection (EMASC) modules to reduce the reconstruction error, improving the high-frequency details in the final image.

3.2.2 Preprocessing

Preprocessing stands as a foundational stage within our methodology, as our proposed model accurately classifies data by establishing patterns within our datasets. Our preprocessing workflow in fig. 9, orchestrates a symphony of four distinct models—Mask and Background Removal, Parsing, Open Pose, and Dense Pose—each meticulously integrated together to execute consecutively.

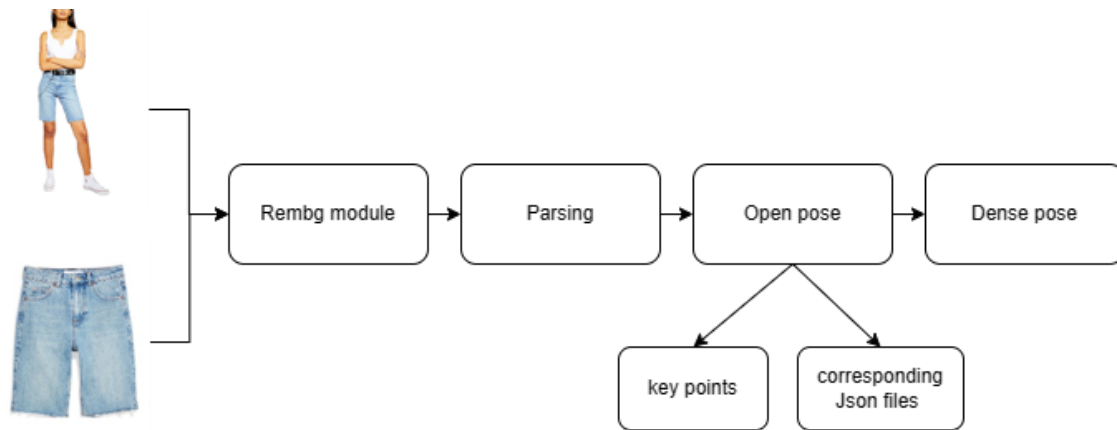


Figure 9. preprocessing pipeline

background removal-mask:

For the background removal process, we have implemented a strategy to enhance the quality of our results by utilizing images with a white background. This deliberate choice has been observed to yield superior outcomes. To streamline this aspect of our pipeline, we have integrated the Rembg module [6]. This powerful library specializes in the removal of backgrounds from images, offering a robust solution for our preprocessing needs. Importantly, we leverage the optional parameter "mask" within the Rembg module, which plays a vital role in isolating specific areas or objects within an image. This strategic approach ensures the precise extraction of both the garment and the person from their backgrounds with optional masks.



Figure 10. background removal-mask

Parsing:

Parsing, also known as segmentation or label mapping, serves as a technique for partitioning the human body into distinct semantic regions, encompassing crucial areas such as the head, torso, arms, legs, and finer details like hair and clothing. We used in our approach the Self Correction for Human Parsing model [3] . This sophisticated model was trained on three diverse datasets. Among the available datasets, we found the ATR dataset and LIP dataset to be particularly well-suited for our purposes. Notably, the ATR dataset boasts 18 semantic category labels, including crucial fashion-related elements such as ‘face, sunglasses, hat, scarf, hair, upper clothes, left and right arms, belt, pants, left and right legs, skirt, left and right shoes, bag, dress, and background’. Meanwhile, the LIP dataset provides 20 semantic category labels, including 'Coat' and 'Jumpsuits' in addition to those present in ATR. By prioritizing fashion-centric semantic categories, our parsing methodology ensures understanding of garment-related features.



Figure 11. Human Parsing model

Open pose:

Open Pose is used for providing accurate human pose estimation and intricate insights into key body joints. The primary objective of Open Pose within our framework is to preserve the fidelity of the model's body pose throughout the processing stages. To accomplish this, we have integrated the PyTorch implementation of the Open Pose model [4], selected for its compatibility with our workflow and its ability to extract key information critical for our model's functioning. The PyTorch Open Pose model efficiently extracts two crucial categories essential for our model's operation: the key points indicative of pose estimation and the corresponding Json files containing the precise coordinates of these key points. Specifically, the key points extracted by Open Pose serve as the foundation for computing the 18-channel pose heatmap. In this heatmap, each channel corresponds to a distinct body key point, providing a comprehensive representation of the subject's pose. By harnessing this detailed information, our preprocessing pipeline ensures the preservation and accurate representation of human body poses.

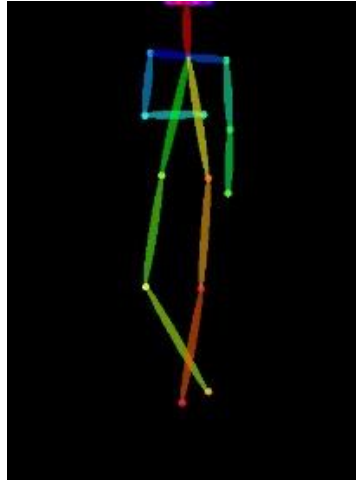


Figure 12. Open Pose model

Dense pose:

Dense Pose strives to establish dense correspondence between every pixel within an image and specific points on the human body surface. This intricate mapping facilitates a pixel-to-surface correlation, wherein each pixel is assigned a unique body surface identifier along with local coordinates, yielding a comprehensive understanding of the body's shape and appearance. Given the computational demands and processing time associated with Dense Pose, we opted for integration with the detectron2 library [5]. Leveraging detectron2's capabilities, we focused on estimating the U coordinates for various body parts. This strategic choice enables us to attain a detailed representation of body part positioning while mitigating the computational overhead typically associated with Dense Pose. The output of Dense Pose comprises two critical components: the 25-channel label map and the 2-channel UV map. These maps encapsulate invaluable information regarding body part segmentation and surface coordinates, respectively. In our preprocessing pipeline, we concatenate these maps without further processing.

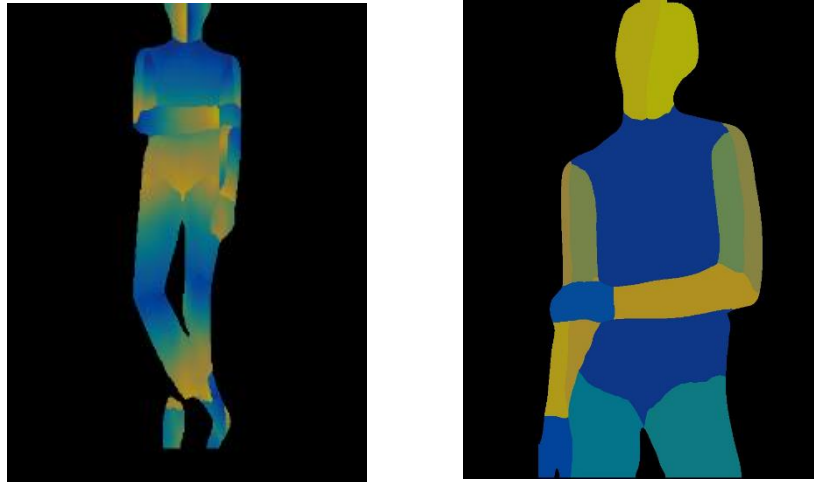


Figure 13. detectron2 library for dense pose model

Conclusion:

Our preprocessing pipeline orchestrates a series of sophisticated techniques, each is used to enhance the quality and relevance of the data input into our proposed model. Beginning with the Mask Model, we delicately extract the garment and person from the background. Parsing extracts semantic information about the human body and attire, while Open Pose and Dense Pose offer detailed insights into body pose and surface correspondence to the body's shape and appearance, respectively. By integrating these components seamlessly, our preprocessing pipeline ensures the extraction of comprehensive data, empowering our model to make accurate classifications. This precisely crafted workflow not only optimizes accuracy but also fosters robustness and efficiency in our methodology.

3.2.3 Generate Your own Cloth Feature

This section provides a comprehensive guide on the "Generate Your Own Cloth" feature, which utilizes a Stable Diffusion model with a focus on generating clothing designs. The feature integrates prompt engineering and web scraping to enhance the quality and variety of generated clothing items.

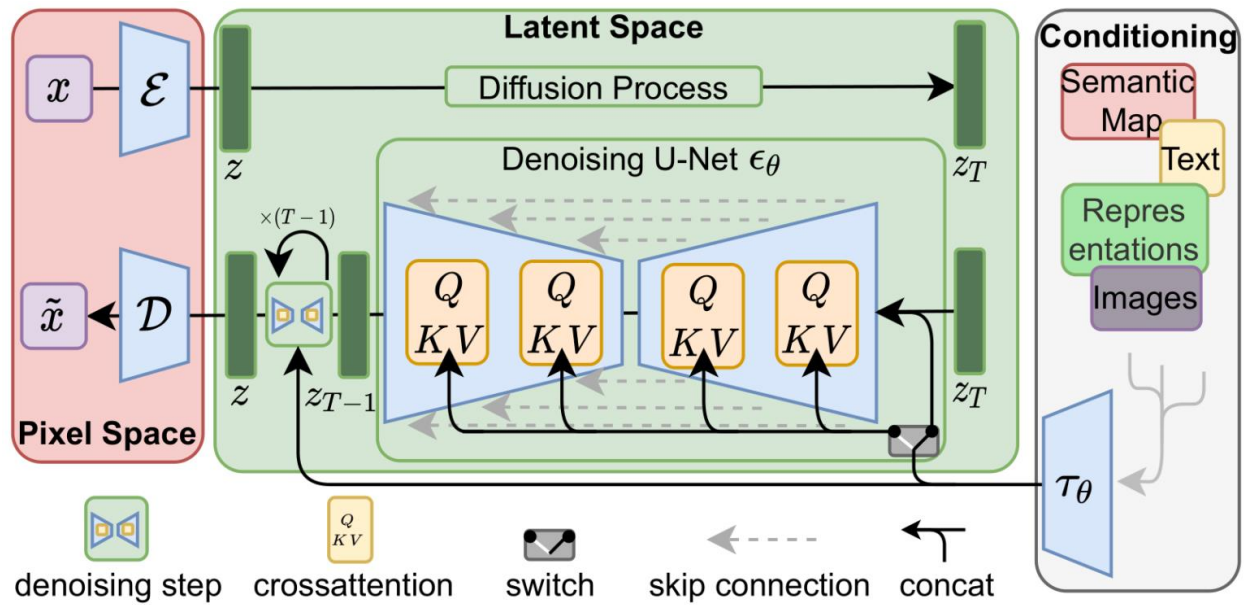


Figure 14. Stable diffusion model architecture and samples

The depicted model is a sophisticated architecture designed to enhance image quality through denoising and conditioning. On the left side of the flowchart, the process begins with the initial noisy input image, denoted as ' x .' The first step

involves denoising, which aims to remove noise and imperfections from this image. This denoising operation, represented by a blue square with a sigma sign, transforms the noisy image 'x' into a denoised version ' \tilde{x} ,' which serves as an improved representation free from noise artifacts [34, 35]. Following the denoising process, the encoder 'E' processes ' \tilde{x} ' further, transforming it into a latent space representation 'z.' The latent space captures essential features of the image while reducing dimensionality [36]. Within the green rectangle labeled "Latent Space," we find the intricate "Denoising U-Net," which involves query 'Q,' key 'K,' and value 'V' blocks [37, 38]. These blocks interact through cross-attention, switch operations, skip connections, and concatenation, ultimately enhancing the latent representation 'z' [39, 40]. On the right side, a separate pathway introduces conditioning semantic map text ' ΣT .' This semantic information interacts with the latent representation 'zT' before generating the final conditioned denoised image. The latent representation 'zT' then undergoes decoding via a decoder 'D,' resulting in the output, which is the conditioned denoised image ' $\tilde{x}T$ ' [41, 42, 43].

This model is integral to our innovative feature, "Generate Your Own Cloth," which enables users to create custom clothing designs based on their textual prompts and preferences. By leveraging the power of semantic map text and latent space transformations, the model ensures that the generated clothing images are not only visually appealing but also align closely with user specifications, providing a personalized and unique design experience [34-43].

4. Implementation, Experimental Setup, & Results

4.1 Implementation Details

Our implementation focuses on utilizing LaDI vton pretrained model [18] consisting of three primary modules: EMASC, textual inversion adapter, and warping component. Without conducting any additional training, we solely utilize the pretrained model for testing purposes. Weight freezing is applied to all modules except the textual inversion adapter, which undergoes evaluation alongside the proposed enhanced Stable Diffusion pipeline. Image generation is conducted at a resolution of 512×384 pixels. The textual inversion network ($F\theta$) integrates a single ViT layer followed by a multi-layer perception with specific configurations. The visual encoder (VE) leverages the pretrained OpenCLIP ViT-H/14 model. Testing of the diffusion virtual try-on model is executed with predefined optimizer and scheduling strategies. A guidance technique is utilized during testing, incorporating the fast variant of the multi-conditional classifier-free guidance. For the autoencoder with EMASC, we utilize the pretrained EMASC modules without further training. Testing involves the application of a combination of L1 and VGG loss functions. Throughout testing, the encoder and decoder remain frozen, and only the EMASC modules are assessed for performance.

Enhancing the proposed model, we customized a preprocessing pipeline to accommodate the specific requirements of the two datasets we are using: Dress-code for lower body and dresses, and Viton HD for upper body. Both datasets undergo a standardized preprocessing pipeline, differing only in parsing and dense pose methodologies.

For the Dress code dataset, parsing is facilitated by the Self Correction for Human Parsing model [3] pretrained on the ATR dataset that boasts 18 semantic category labels while Viton HD parsing model is pretrained on the LIP dataset provides 20 semantic category labels.

Notably, a divergence occurs in the utilization of Dense Pose detectron2 [5] parameters. In the Viton HD dataset, the `dp_segm` parameter is employed to

generate segmentation masks for annotated persons, optimizing pose estimation accuracy. Conversely, in the Dress code dataset, the `dp_u` parameter is utilized for point annotation colored according to their U coordinate in part parameterization.

Despite parsing and dense pose variations, all other models in the pipeline remain consistent for both datasets. The input garment image and the person image both traverse through the Rembg module [6], followed by parsing [3], Open Pose [4], and Dense Pose [5] models sequentially. Subsequently, they are fed into the pretrained model, generating the final output of the person wearing the new desired garment.

For our features, we implemented a recommendation system for fashion items based on garment input texture and color. We also implemented a complementary items recommendation system based on gender. Both systems utilize state-of-the-art deep learning techniques for feature extraction and similarity search to provide personalized recommendations to users. It begins by loading a pretrained ResNet50 model. The model is initialized with weights pretrained on the ImageNet dataset [20] and is configured to exclude the top classification layer, ensuring that it captures high-level features relevant to garment images. Additionally, for computing embeddings and filenames, we employed a custom dataset for each system consisting of garment and accessories images. The process of creating embeddings from these datasets involved several steps allowing for efficient feature retrieval during recommendation. Garment images are preprocessed and passed through the pretrained ResNet50 model to extract high-dimensional feature vectors. These feature vectors encode semantic information about the garments, capturing their visual characteristics in a compact representation. To ensure consistency and comparability, the extracted features are normalized to unit length. The recommendation system employs a Nearest Neighbors algorithm to find similar garment images based on the extracted features. By calculating the Euclidean distance between feature vectors, the system identifies the most visually similar garments to the input image. The number of nearest neighbors, as well as the distance metric, are configurable parameters that can be adjusted to fine-tune the recommendation results to deliver personalized and visually appealing fashion recommendations to users.

For our “Generate Your Own Cloth” feature we used the stable diffusion model which is an advanced technique in machine learning, particularly powerful for generating diverse and realistic images. Implementation begins with neural networks trained on extensive datasets to effectively learn diffusion dynamics. Prompt engineering is crucial, involving the design of prompts guiding models to specific outputs, utilizing filters ensuring content aligns with predefined criteria. These filters can include style transfer algorithms to maintain consistency, sentiment analysis for emotional tone control, and topic modeling for thematic relevance. Additionally, web scraping techniques are often employed to gather diverse and up-to-date datasets for training and fine-tuning the model. The model's capability to produce 50 possible outputs at the same prompt ensures a wide range of creative options, making it a versatile tool for applications such as creative content generation and artistic expression. Rigorous testing and fine-tuning are essential to optimize model performance, ensuring outputs are stable, diverse, and high quality. Together, these elements make the stable diffusion model with prompt engineering a robust solution for image generation across various domains.

Our project includes “Flask,” a backend server system consisting of two interconnected Flask applications: one for handling user interactions and generating try-on images, and the other for managing server operations. The first Flask application hosts several key APIs: the Cloth Generation API (`/api/upload_and_generate_tryon`) for creating virtual clothing items using machine learning, the Recommendation System API (`/get_cloth_rec`) for personalized clothing suggestions, and the Complementary Items API (`/get_comp_rec`) for recommending accessories. The backend leverages Python and Flask, with image processing managed by Pillow (PIL), and path management facilitated by `os` and `Pathlib`. The secondary Flask application optimizes resource usage by managing server states, enabling the system to turn the primary application off or on based on requests, thereby saving resources and maximizing performance. This setup also facilitates the integration of our mobile app through these APIs, allowing users to interact via their mobile devices and receive personalized recommendations based on their inputs and current fashion trends.

4.2 Experimental / Simulations Setup

4.2.1 experimental setup:

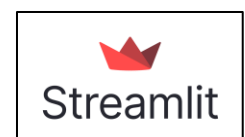
For our virtual try-on application, we leveraged two specific datasets to handle different segments of attire - upper body and lower body including dresses:

VITON-HD: This dataset is primarily used for experiments involving the upper body garments. It features high-resolution images that are crucial for maintaining the quality and realism in virtual try-ons.

Dress Code: we utilized the Dress Code dataset, which encompasses upper and lower garments as well as dresses. However, due to performance limitations, we focused solely on lower garments and dresses, as the dataset's effectiveness was not optimal for upper garments.

Both datasets were instrumental in training our models to generate realistic and accurate garment overlays on user-provided images. They provided a wide variety of garment types and styles, which enhanced the versatility and robustness of our application.

We created a web app using Google Colab and Streamlit for seamless data integration and interactive visualizations.



We also created a mobile app using Flutter for a native user experience, leveraging Dart for development. For seamless data integration and interactive features, we utilized Firebase technologies, including Firestore as our database and Firebase Storage for handling media files.



4.2.2 Web Application Deployment:

Due to the intensive GPU requirements of this project, which surpasses the capabilities of standard laptops and desktops, we executed all computational tasks on Google Colab.

Google Colab provides a robust cloud-based service with access to high performance GPUs, which is essential for handling the computational load efficiently and effectively in our experiments. This environment ensures that all users can replicate our results without the need for specialized hardware. In addition to computational tasks, we also utilized Google Colab to run our web application using Streamlit.

Streamlit is an open-source app framework specifically designed for Machine Learning and Data Science projects. It allows us to create beautiful, interactive web applications quickly and with minimal code. By using Streamlit, we were able to build and deploy a user-friendly interface for our project. Streamlit allowed us to turn data scripts into shareable web apps in a very short amount of time. It seamlessly integrates with popular data science libraries which we heavily used in our project. This integration allowed us to visualize complex datasets and model outputs directly within the web application. This was crucial for iterative testing and feedback enabling users to interact with our models and visualizations directly through their web browser.

Deploying a Streamlit app on Google Colab involved minimal setup. We used specific port forwarding techniques and public URLs to access our Streamlit web application directly from Colab, which gave us the advantage of a high-powered computational backend coupled with an efficient, easy-to-use frontend.

This combination not only maximized our productivity but also enhanced the accessibility of our project, making it possible for anyone with internet access to interact with our advanced models without requiring any downloads or local installations.

4.2.3 Mobile Application Deployment

FITMI is designed to simplify outfit selection and wardrobe management. FITMI leverages Dart and Flutter for the frontend, while backend services and storage system are powered by Firebase. Additionally, FITMI integrates with an AI model through Flask, providing powerful endpoint APIs for generating photos and enhanced outfit recommendations.

Home Page: Start by capturing photos of your clothing items and yourself. Choose your gender and select the category of clothing (upper, lower for men) (upper, lower, dresses for women). FITMI uses these photos to generate the outfit first then, personalized outfit recommendations and suggests complementary items based on your selections and AI insights.

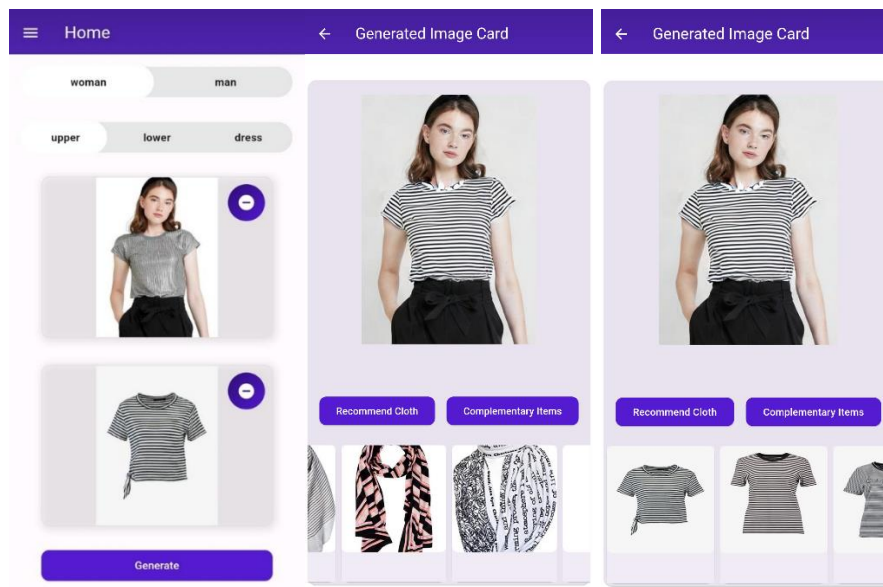


Figure 15. FITMI's Mobile home screen.

Wardrobe Page: Manage your clothing items and save outfit recommendations effortlessly. Upload images from your gallery or show the recommended images generated from the home page. Organize items by the chosen categories and keep track of your favorite outfits.

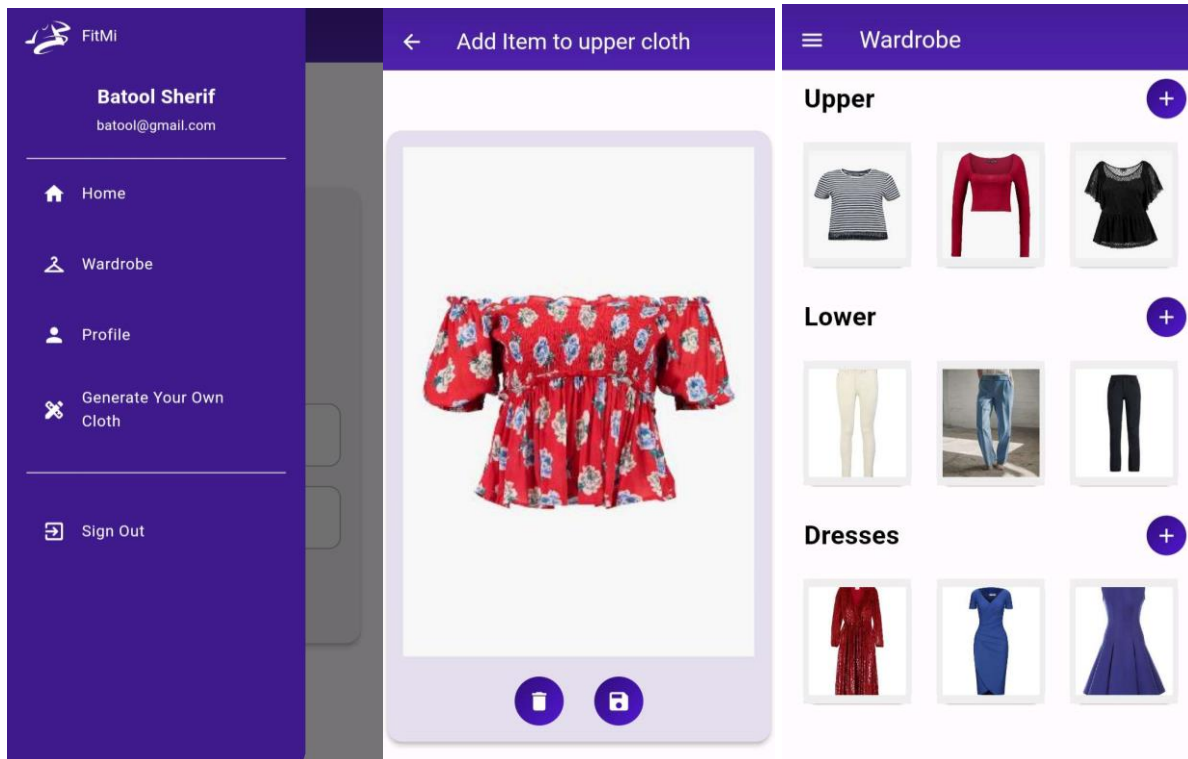


Figure 16. FITMI's Mobile wardrobe screen.



Generate Your Own Cloth Page: Explore new clothing styles based on your preferences. Enter your style preferences and FITMI retrieves clothing styles from the internet with the help of an AI model. Save these generated photos as references in your wardrobe for future inspiration.

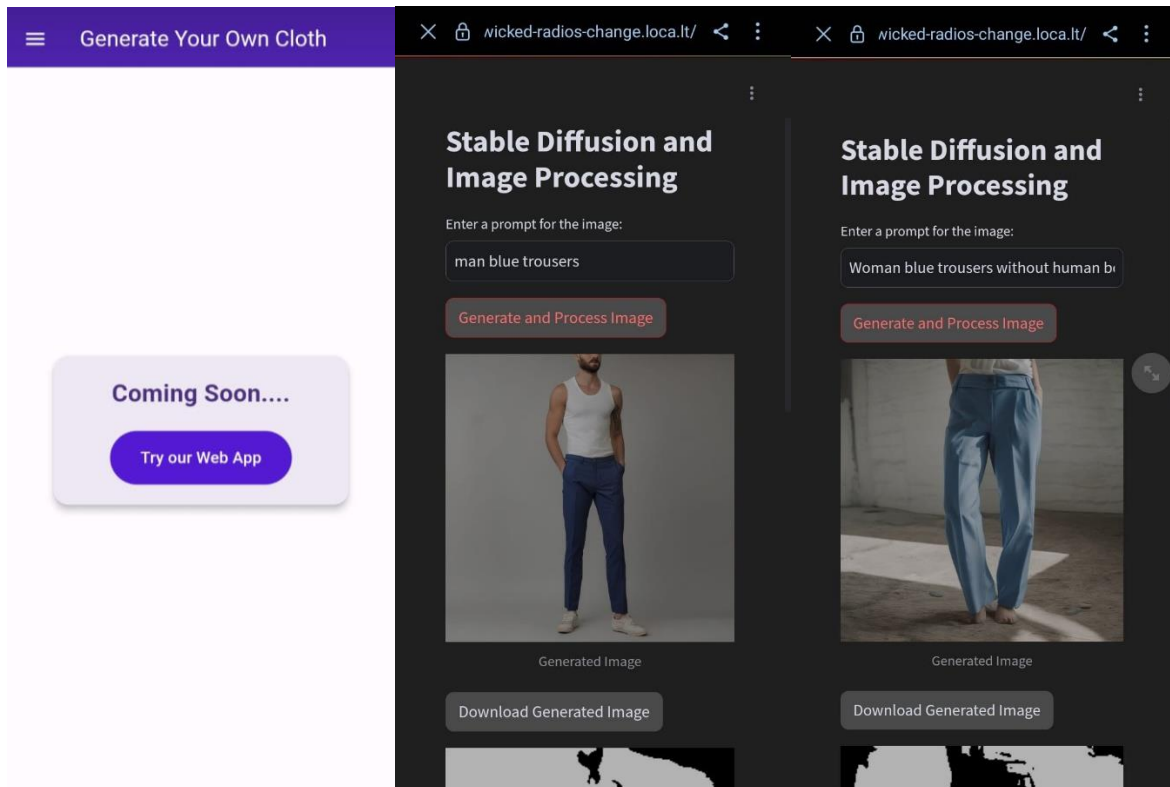
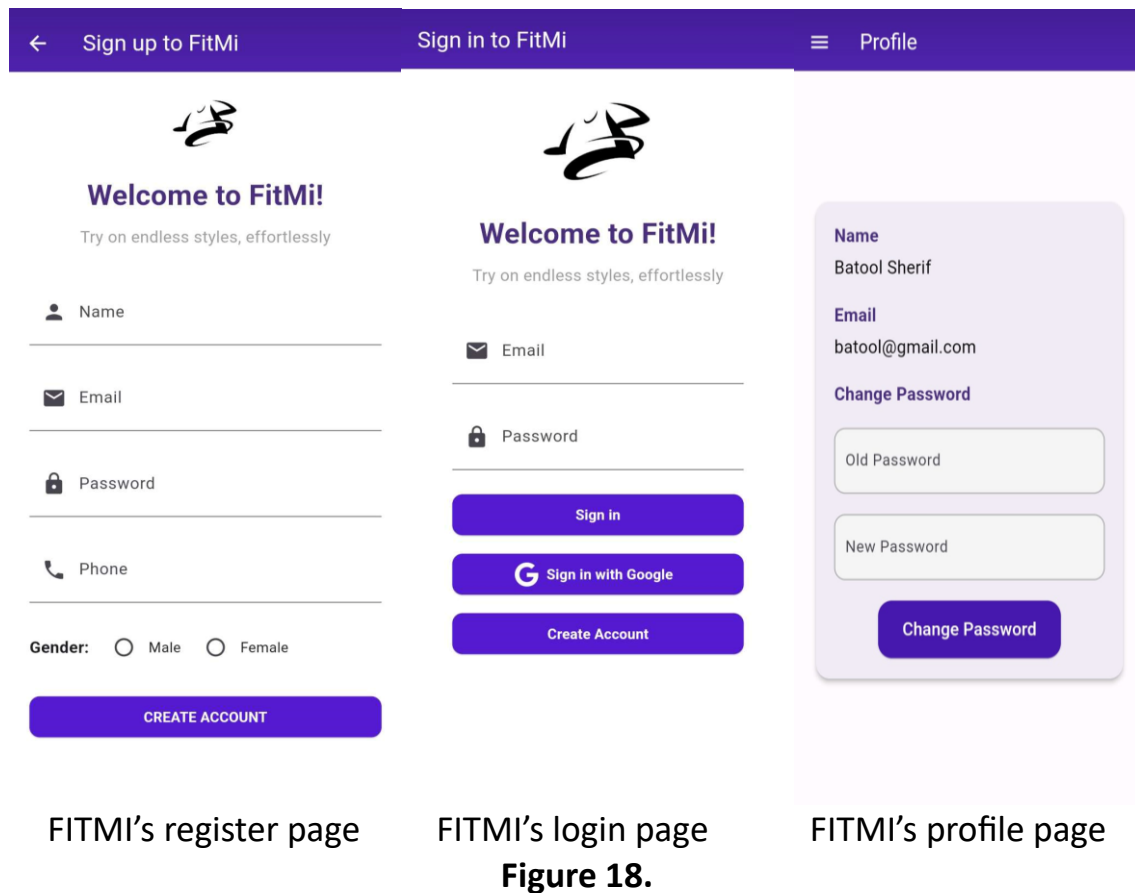


Figure 17. FITMI's generate your own cloth.



Additional Pages: The Profile page allows you to update your password and view your user information. The Registration and Sign-In pages facilitate easy account creation and login.



Future Enhancements: In future updates, we plan to integrate additional clothing style APIs for more diverse recommendations, add social sharing features to share outfit combinations, and become real time app.

FITMI is designed to simplify your outfit selection process and ensure your wardrobe stays organized and stylish. Download FITMI today and let us help you make getting dressed a breeze!

4.3 Conducted Results

The input garment and person undergo our preprocessing pipeline, where various transformations occur, although these intermediate preprocessing images are not directly shown to the user. Instead, they serve as inputs to the model. Behind the scenes, which then generates the results.

Upper body

We used Viton HD dataset that focused solely on upper garments.

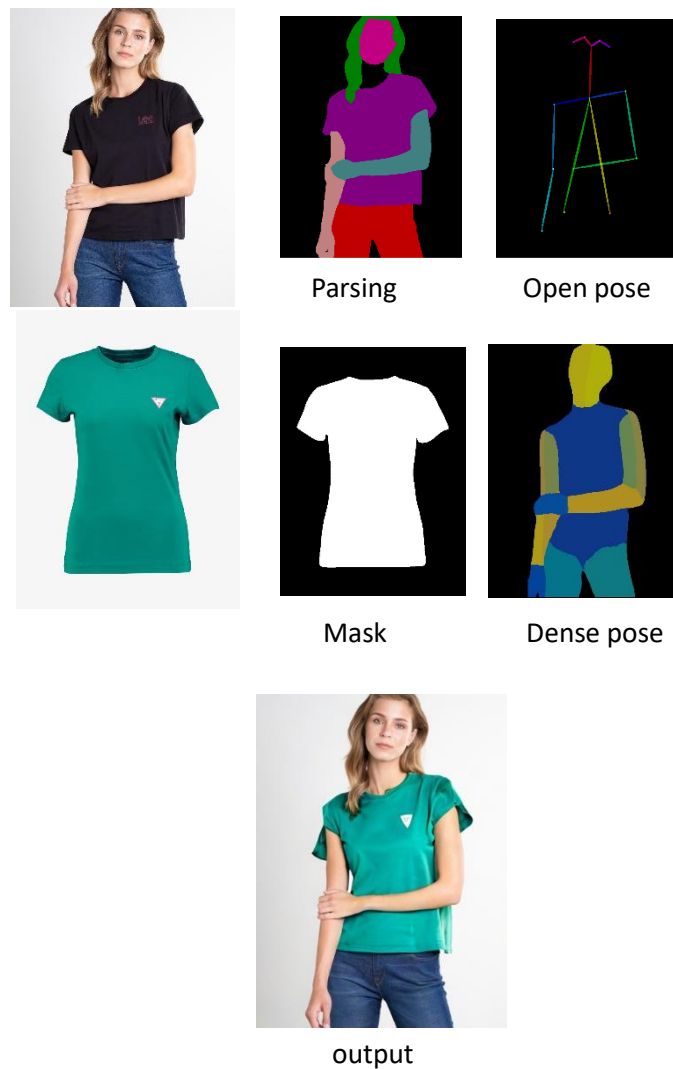


Figure 19. Qualitative results for upper body generated by FITMI

Images in fig. 20 represent the recommendation system and complementary items' outputs, where recommendations are adapted based on the input garment, and complementary are provided based on the user's gender.

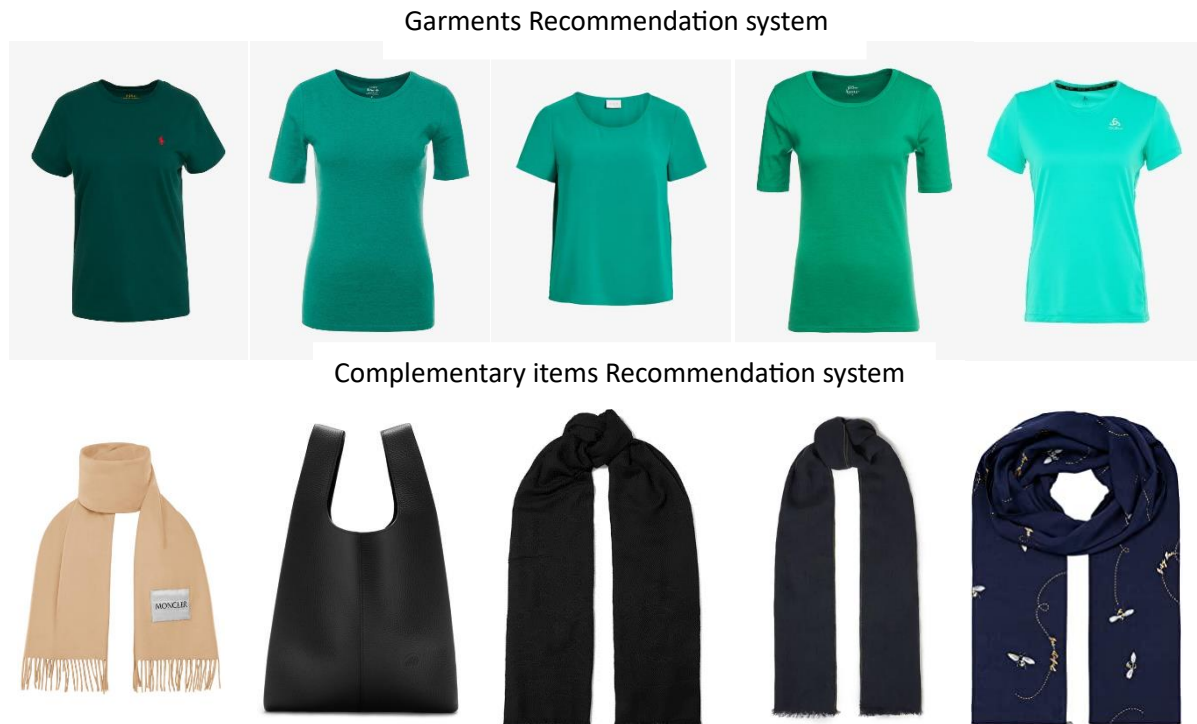


Figure 20. FITMI's Recommendation systems for upper body

Lower body

We used Dresscode dataset that focused solely on lower garments and dresses.

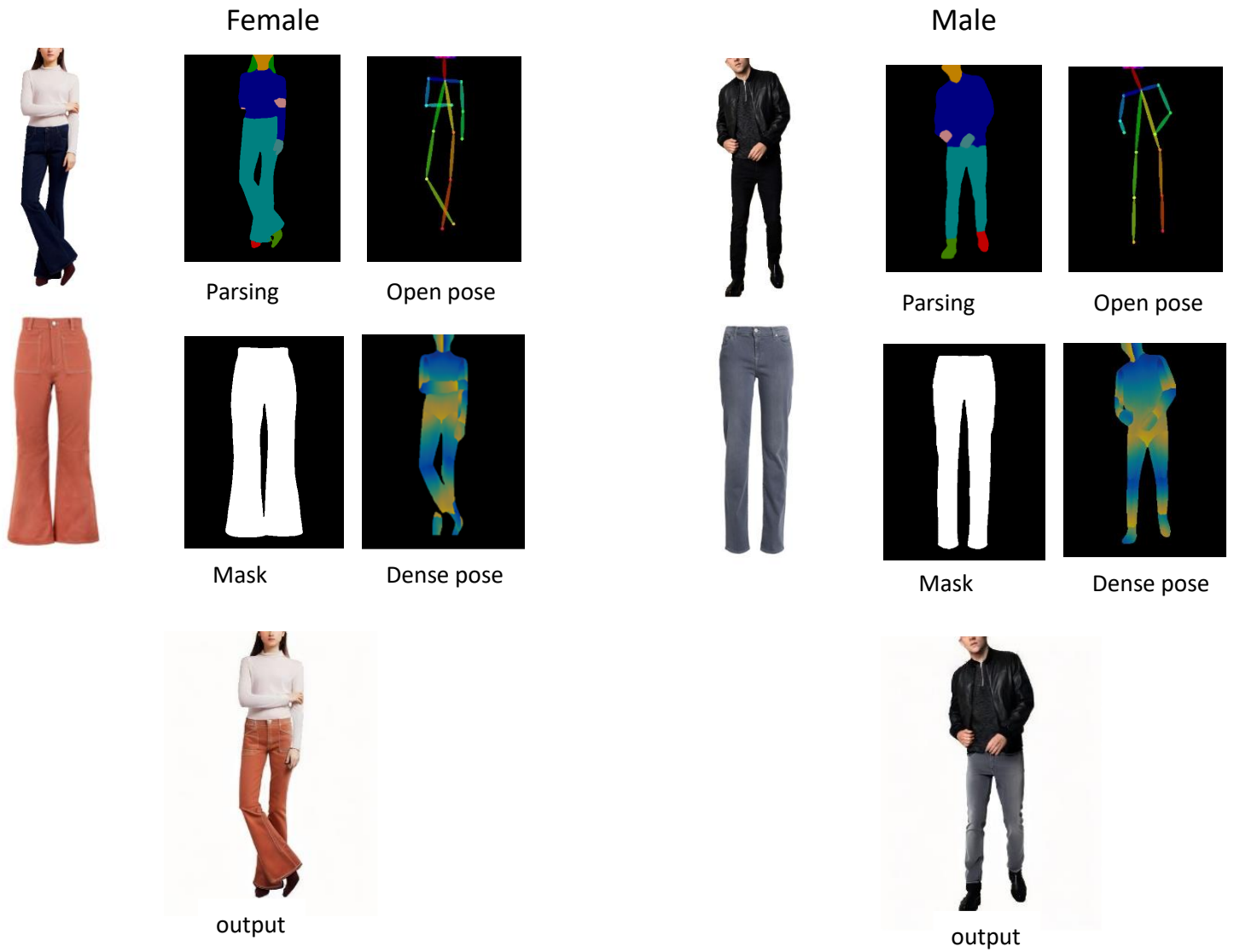


Figure 21. Qualitative results for lower body generated by FITMI

Images in fig. 22 represent the recommendation system and complementary items' outputs, where recommendations are adapted based on the input garment, and complementary are provided based on the user's gender.

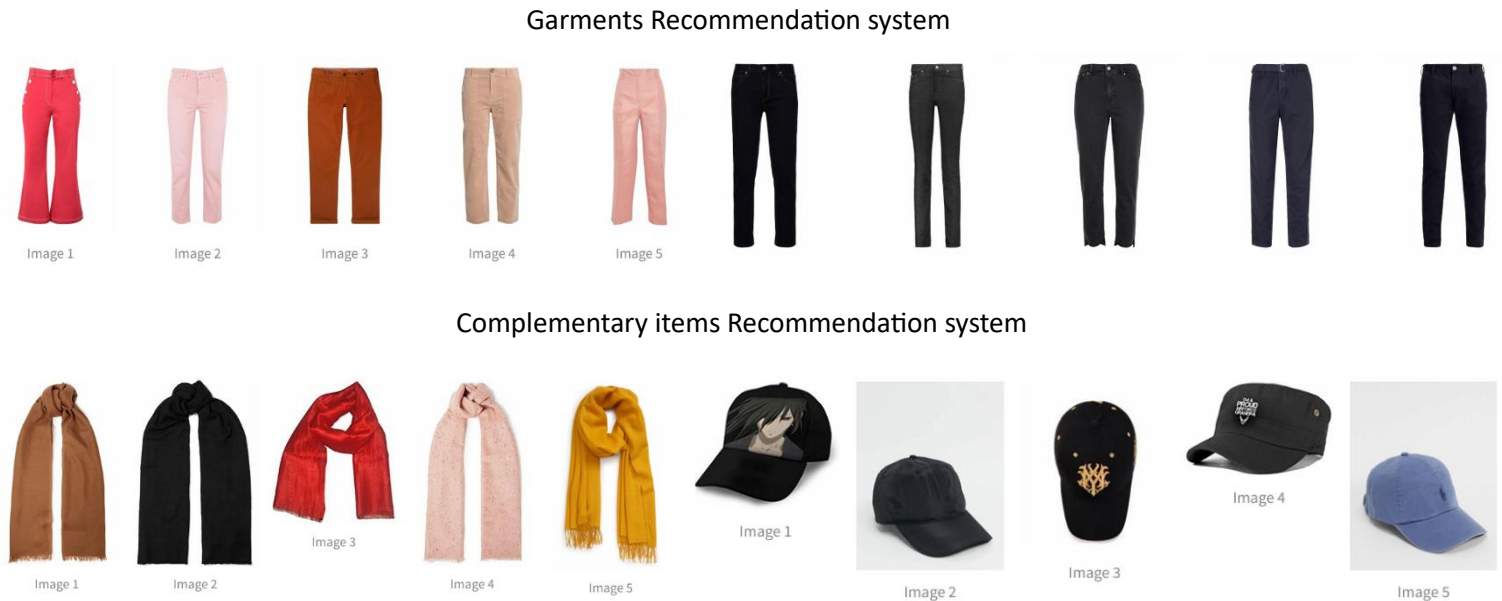



Figure 22. FITMI's Recommendation systems for lower body

Fig. 23 illustrates our website interface, showcasing both the input and output sections, alongside our two recommendation systems. The input section allows users to input garment and personal details, while the output section displays the results generated by our model, and our main features.



Navigation

Select Page

Try Clothes

Try Clothes

Category

Select category

lower_body

Gender

Select gender

female

Cloth image

Select option:

☒ Upload image

☐ Capture from camera


Upload cloth image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

014195_1.png 115.3KB



Cloth image

Person image

Select option:

☒ Upload image

☐ Capture from camera


Upload person image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

013644_0.png 70.9KB



Person image

Generate




Recommended Clothes



Complementary Items





Navigation

Select Page

Try Clothes

Try Clothes

Category

Select category

lower_body

Gender

Select gender

male

Cloth image

Select option:

☒ Upload image

☐ Capture from camera


Upload cloth image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

014044_1.png 102.1KB



Cloth image

Person image

Select option:

☒ Upload image

☐ Capture from camera


Upload person image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Browse files

99e8ef450d5183db751ef1... 43.6KB



Person image

Generate



Recommended Clothes



Complementary Items



Figure 23. FITMI's web interface

Dresses

We used Dresscode dataset that focused solely on lower garments and dresses.

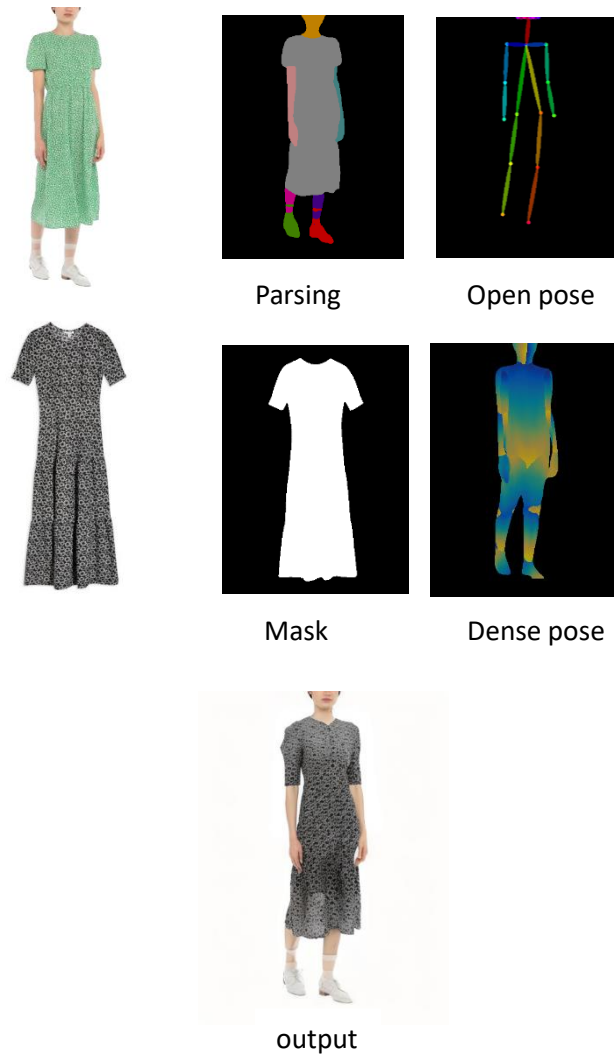


Figure 24. Qualitative results for dresses generated by FITMI

Images in fig. 25 represent the recommendation system and complementary items' outputs, where recommendations are adapted based on the input garment, and complementary are provided based on the user's gender.

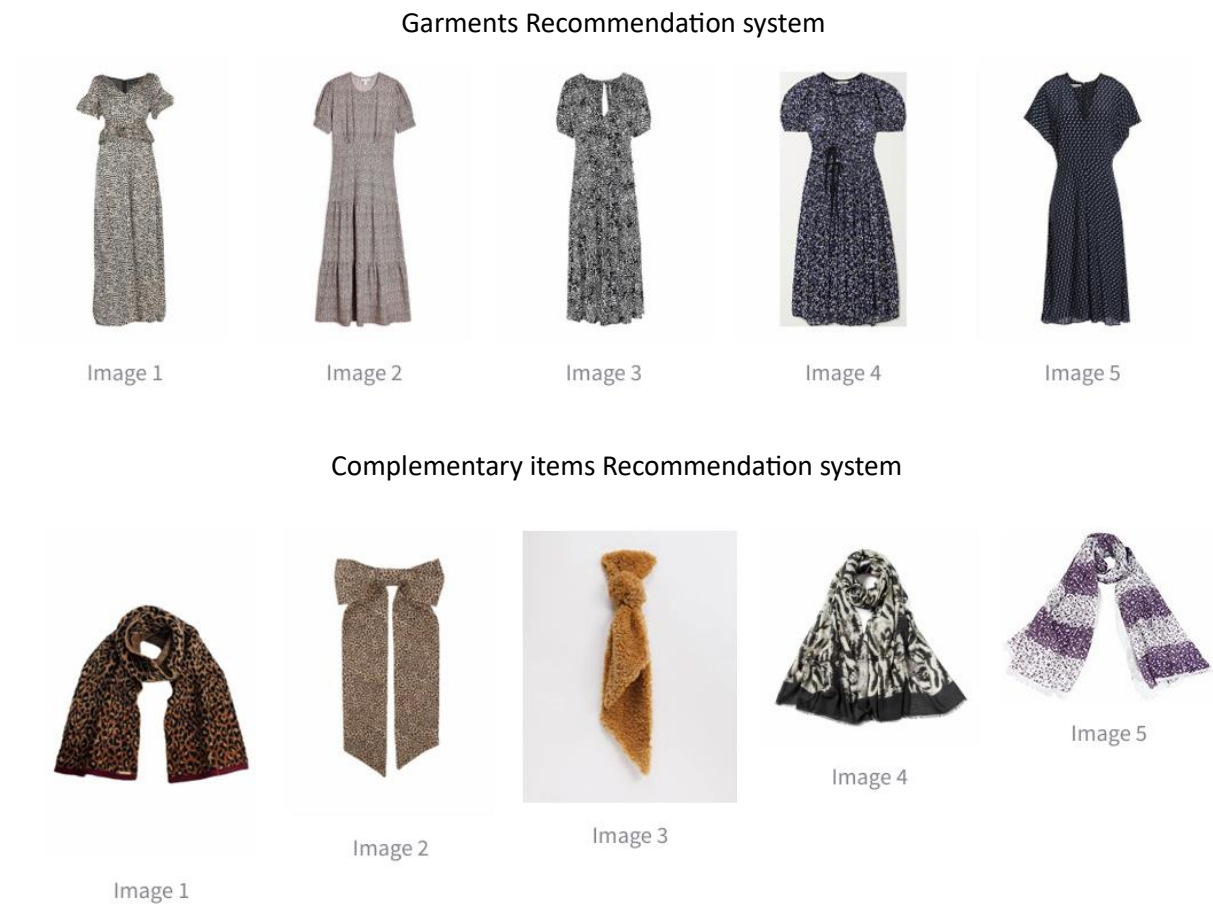


Figure 25. FITMI's Recommendation systems

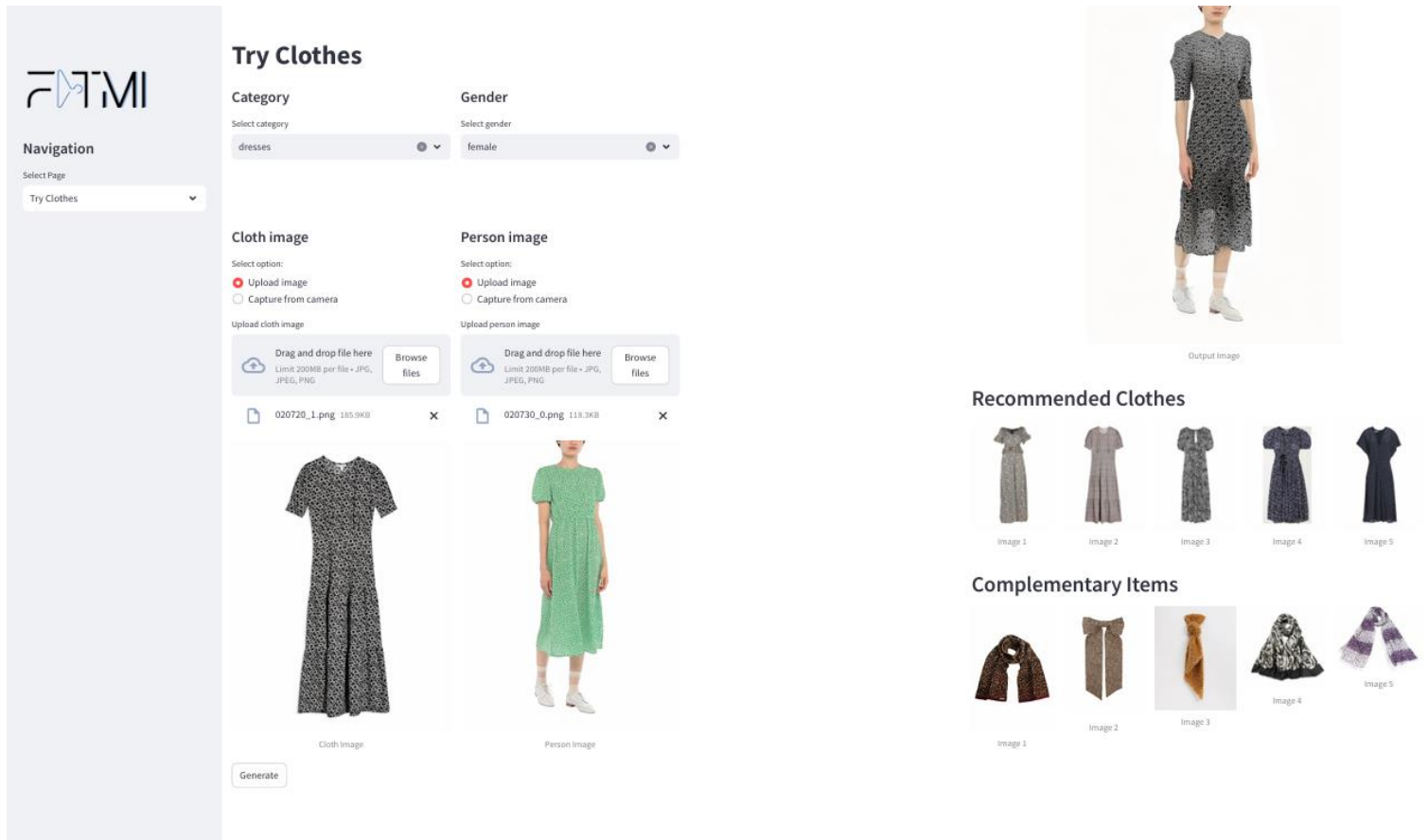


Figure 23. FITMI's web interface

Generate your own cloth

Images in fig. 26 represent the recommendation system using the stable Diffusion and Image Processing interface where recommendations are adapted based on the user descriptions.

Stable Diffusion and Image Processing

Enter a prompt for the image:

woman white dress

Generate and Process Image



Stable Diffusion and Image Processing

Enter a prompt for the image:

man blue trousers

Generate and Process Image



Figure 26. FITMI's Generate Your Own Cloth page

4.4 Testing & Evaluation

We perform experiments on two virtual try-on datasets, namely Dress Code [1] and VITON-HD [2], that feature high-resolution image pairs of in-shop garments and model images in unpaired settings. In the unpaired setting, a different garment from the model is selected for the virtual try-on task.

To quantitatively evaluate our model, we employ evaluation metrics to estimate the coherence and realism of the generation. We use the Learned Perceptual Image Patch Similarity (LPIPS) [28] and the Structural Similarity (SSIM) [29] to evaluate the coherence of the generated image compared to the ground-truth. We compute these metrics on the unpaired setting of both datasets. To measure the realism, we instead employ the Fréchet Inception Distance [30] and the Kernel Inception Distance [31] in unpaired (i.e., FID_u and KID_u) settings. For the LPIPS and SSIM implementation, we use the torch-metrics Python package [32], while for the FID and KID scores, we employ the implementation in [33].

VITON-HD

The VITON-HD dataset [2] comprises 13,679 image pairs, each composed of a frontal view woman and an upper-body clothing item with a resolution equal to 1024×768. The dataset is divided into training and test sets of 11,647 and 2,032 pairs, respectively.

In Table 3, we show the quantitative analysis of the VITON-HD dataset. FITMI surpasses all other competitors by a large margin in terms of FID and KID, demonstrating its effectiveness in this setting.

Table 3. Quantitative results on the Viton HD dataset

Model	LPIPS↓	SSIM↑	FID _u ↓	KID _u ↓
CP-VTON [27]	-	0.791	30.25	40.12
ACGPN [24]	-	0.858	14.43	5.87
VITON-HD [2]	0.116	0.863	12.96	4.09
HR-VTON [22]	0.097	0.878	13.06	4.72
FITMI	0.091	0.876	9.41	1.60

Dress code

The Dress Code dataset [45] features over 53,000 image pairs of clothes and human models wearing them. The dataset includes high-resolution images (i.e., 1024×768) and garments belonging to different macro-categories, such as upper-body clothes, lower-body clothes, and dresses. In our experiments, we employ the original splits of the dataset where 5,400 image pairs (1,800 for each category) compose the test set and the rest the training one

Table 4 reports the quantitative results on the Dress Code dataset. As can be seen, FITMI achieves comparable results in terms of coherence with the inputs (i.e., LPIPS and SSIM), while significantly outperforming all competitors in terms of realism in unpaired settings.

Table 4. Quantitative results on the Dress Code dataset

ALL				
Model	LPIPS↓	SSIM↑	FID _u ↓	KID _u ↓
PF-AFN [23]	-	-	-	-
HR-VTON [22]	-	-	-	-
CP-VTON [27]	0.186	0.842	31.19	25.17
FITMI	0.064	0.906	6.48	2.20

As a complement to Table 4, Table 5 presents the complete quantitative results for each category of the Dress Code dataset. Our method, denoted as FITMI, demonstrates superior performance compared to all competitors across all three Dress Code categories in terms of realism metrics such as FID and KID in the unpaired settings. When assessing input adherence metrics such as LPIPS and SSIM, our approach achieves better results than CP-VTON [27].

Table 5. Quantitative results per category on the Dress Code dataset

	Upper-body		Lower-body		Dresses	
Model	FID \downarrow	KID \downarrow	FID \downarrow	KID \downarrow	FID \downarrow	KID \downarrow
PF-AFN [23]	14.32	-	18.32	-	13.59	-
HR-VTON [22]	16.86	-	22.81	-	16.12	-
CP-VTON [27]	48.31	35.25	51.29	38.48	25.94	15.81
FITMI	13.26	2.67	14.80	3.13	13.40	2.50

Notably, our solution can generate highly realistic images and preserve the texture and details of the original in-shop garments, as well as the physical characteristics of target models and Results show that the proposed FITMI model outperforms by a large margin the competitors in terms of realism on both Dress Code and VITON-HD datasets, two widely used benchmarks for the task.

5. Discussion, Conclusions, and Future Work

5.1 Discussion

The superiority of FITMI over existing methods in image-based virtual try-on systems can be attributed to several factors, including the model architecture it utilizes, particularly latent diffusion models (LDMs), which have shown superior image generation quality compared to traditional methods such as GANs.

By building upon the pioneering LaDI-vton model [18], which is the first latent diffusion-based approach for virtual try-on, FITMI inherits the strengths of LDMs while enhancing its methodology and incorporating novel fundamentals.

FITMI's allows for more detailed and controllable image synthesis, which is critical for virtual try-on applications where preserving the texture, shape, and physical characteristics of both garments and target models is paramount.

The performance of FITMI surpasses existing methods outlined in the related work. It consistently achieves lower (FID) and (KID) scores compared to its competitors across both the Dress Code and VITON-HD datasets. These metrics directly measure the realism and fidelity of the generated images, indicating FITMI's ability to produce virtual try-on images that closely resemble real-world counterparts. Despite achieving lower FID and KID scores, FITMI achieves comparable or even better (LPIPS) and (SSIM) scores compared to other models. This signifies FITMI's capability to accurately transfer clothing items onto target subjects while preserving texture, shape, and overall coherence.

5.2 Summary & Conclusion

This documentation has detailed the development and optimization of the FITMI application, leveraging the LDMs with two well-known benchmarks: Dress Code and VITON-HD Datasets. To increase the detail retention of the input in-shop garment, we exploit the textual inversion technique, demonstrating its capability in conditioning the generation process. Moreover, we introduce the EMASC modules that can enhance the inpainting output image quality reducing the

autoencoder compression loss of LDMs. This advancement notably improves the human perceived quality of high-frequency human body details. In the preprocessing phase, four critical techniques were utilized. Each technique was integrated and executed consecutively to ensure optimal input quality for the virtual try-on model. Central to FITMI application's user appeal is its sophisticated recommendation system. The application not only enables users to see themselves in desired outfits but also intelligently recommends garments based on individual style preferences. This personalization is further enhanced by FITMI's ability to suggest complementary items, effectively simulating a personalized shopping assistant. We also Introduced 'Generate Your Own Cloth', a feature that utilizes web scraping to create personalized clothing photos based on user descriptions. These photos can be seamlessly transferred to a virtual wardrobe, allowing users to try on their custom designs virtually.

Through numerous experiments, we have refined these processes to achieve the best possible accuracy and realism in the virtual try-on experience. These experiments were critical in helping us determine the most effective configurations and settings for our models, ultimately leading to a more reliable and user-friendly application.

5.3 Future Work

We are excited to explore several enhancements to further improve our FITMI application. Key among these is the development of real-time virtual try-on. This feature will allow users to see how clothes fit on their moving image instantly, making the virtual try-on experience even more dynamic and interactive.

Additionally, we plan to expand FITMI's functionality to serve not just individual consumers but also businesses within the apparel industry. By integrating FITMI with online retail platforms, we can offer a tool that enhances the shopping experience for customers, potentially increasing engagement, and reducing return rates.

6. References

- [1] Davide Morelli, Matteo Fincato, Marcella Cornia, Federico Landi, Fabio Cesari, Rita Cucchiara, **Dress-Code**: High-Resolution Multi-Category Virtual Try-On, University of Modena and Reggio Emilia, Italy, 2022.
- [2] Seunghwan Choi, Sunghyun Park, Minsoo Lee, Jaegul Choo, **VITON-HD**: High-Resolution Virtual Try-On via Misalignment-Aware Normalization, South Korea, 2021.
- [3] Peike Li, Yunqiu Xu, Yunchao Wei, Yi Yang, **Self-Correction-Human-Parsing**, Centre for Artificial Intelligence, University of Technology Sydney, 2020.
- [4] Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh, Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, **PyTorch Open-pose**, The Robotics Institute, Carnegie Mellon University, 2017.
- [5] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, Ross Girshick, **Detectron2 dense-pose**, 2019.
- [6] Daniel Gatis, **Rembg** remove images background, 2020.
- [7] Neuberger, A., Borenstein, E., Hilleli, B., Oks, E., Alpert, S.: Image Based Virtual Try-On Network From Unpaired Data. **O-viton** 2020.
- [8] Lewis, K.M., Varadharajan, S., Kemelmacher-Shlizerman, I.: **TryOnGAN**: Body-Aware Try-On via Layered Interpolation. 2021.
- [9] Li, K., Chong, M.J., Zhang, J., Liu, J.: Toward Accurate and Realistic Outfits Visualization with Attention to Details. **Revery AI** In: CVPR 2021.
- [10] Yildirim, G., Jetchev, N., Vollgraf, R., Bergmann, U.: Generating high-resolution fashion model images wearing custom outfits. **Zalando** In: ICCV Workshops 2019.
- [11] Hieh, C.W., Chen, C.Y., Chou, C.L., Shuai, H.H., Liu, J., Cheng, W.H.: **FashionOn**: Semantic-guided image-based virtual try-on with detailed human and clothing information. In: ACM Multimedia 2019.

- [12] Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: **DeepFashion**: Powering robust clothes recognition and retrieval with rich annotations. In: CVPR 2016.
- [13] Dong, H., Liang, X., Shen, X., Wang, B., Lai, H., Zhu, J., Hu, Z., Yin, J.: Towards Multi-Pose Guided Virtual Try-on Network. **MVP** In: ICCV 2019.
- [14] Zheng, N., Song, X., Chen, Z., Hu, L., Cao, D., Nie, L.: Virtually Trying on New Clothing with Arbitrary Poses. **FashionTryOn** In: ACM Multimedia 2019.
- [15] Yoo, D., Kim, N., Park, S., Paek, A.S., Kweon, I.S.: Pixel-level domain transfer. **LookBook** In: ECCV 2016.
- [16] Han, X., Wu, Z., Wu, Z., Yu, R., Davis, L.S.: **VITON**: An Image-based Virtual Try-On Network. In: CVPR 2018.
- [17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis With **Latent Diffusion Models**. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
- [18] Davide Morelli, Alberto Baldrati, Giuseppe Cartella, Marcella Cornia, Marco Bertini, Rita Cucchiara, 2023 **LaDI-VTON**: Latent Diffusion Textual-Inversion Enhanced Virtual Try-On.
- [19] Hammaad ali, **Real Fashion**, real dataset which contains images of various kinds of clothing accessories, 2020.
- [20] Jia Deng et al. **ImageNet**: A large-scale hierarchical image database.
- [21] Jean Duchon. 1977. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In Constructive Theory of Functions of Several Variables.
- [22] Sangyun Lee, Gyojung Gu, Sunghyun Park, Seunghwan Choi, and Jaegul Choo. 2022. **HR-VITON**, High-Resolution Virtual Try-On with Misalignment and Occlusion-Handled Conditions. In Proceedings of the European Conference on Computer Vision.
- [23] Yuying Ge, Yibing Song, Ruimao Zhang, Chongjian Ge, Wei Liu, and Ping Luo. 2021. **PF-AFN**, Parser-free virtual try-on via distilling appearance flows. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[24] Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. 2020. **ACGPN**, towards photo-realistic virtual try-on by adaptively generating-preserving image content. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[25] C. Lin, Z. Li, S. Zhou, S. Hu, J. Zhang, L. Luo, J. Zhang, L. li Huang, and Y. He. **Rmgn**: A regional mask guided network for parser-free virtual try-on. In IJCAI, pp. 1151–1158, 2022.

[26] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization, **SPADES**. In CVPR, pp. 2337–2346, 2019.

[27] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. 2018. **CP-VTON**, Toward characteristic-preserving image-based virtual try-on network. In Proceedings of the European Conference on Computer Vision.

[28] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. **LPIPS**, The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[29] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. **SSIM** Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13, 4 (2004), 600–612.

[30] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. 2017. **FID**, GANs trained by a two time-scale update rule converge to a Nash equilibrium. In Advances in Neural Information Processing Systems.

[31] Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. 2018. **KID**, Demystifying MMD GANs. In Proceedings of the International Conference on Learning Representations.

- [32] Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh Jha, Teddy Koker, Luca Di Liello, Daniel Stancu, Changsheng Quan, Maxim Grechkin, and William Falcon. 2022. **TorchMetrics**-Measuring Reproducibility in PyTorch. Journal of Open Source Software 7, 70 (2022), 4101.
- [33] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. 2022. On Aliased Resizing and Surprising Subtleties in GAN Evaluation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
- [34] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. 2023. SpaText: Spatio-Textual Representation for Controllable Image Generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 18370–18380
- [35] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2023. When and why Vision-Language Models behave like Bags-of-Words, and what to do about it?. In International Conference on Learning Representations. <https://openreview.net/forum?id=KRLUvxh8uaX>
- [36] Yu Zeng, Zhe Lin, Jianming Zhang, Qing Liu, John Collomosse, Jason Kuen, and M. Patel, Vishal. 2022. SceneComposer: Any-Level Semantic Image Synthesis. arxiv (2022).
- [37] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. 2023. Text-to-image Diffusion Models in Generative AI: A Survey. arXiv:2303.07909 [cs.CV]
- [38] Chenshuang Zhang, Chaoning Zhang, Sheng Zheng, Mengchun Zhang, Maryam Qamar, Sung-Ho Bae, and In So Kweon. 2023. A Survey on Audio Diffusion Models: Text To Speech Synthesis and Enhancement in Generative AI. arXiv:2303.13336 [cs.SD]
- [39] Lvmin Zhang and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. arXiv:2302.05543 [cs.CV]
- [40] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. 2023. Uni-ControlNet: All-in-One Control to Text-to-Image Diffusion Models. arXiv preprint arXiv:2305.16322 (2023).

- [41] Yufan Zhou, Bingchen Liu, Yizhe Zhu, Xiao Yang, Changyou Chen, and Jinhui Xu. 2022. Shifted Diffusion for Text-to-image Generation. arXiv preprint arXiv:2211.15388 (2022).
- [42] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. 2023. ELITE: Encoding Visual Concepts into Textual Embeddings for Customized Text-to-Image Generation. arXiv:2302.13848 [cs.CV]
- [43] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. 2023. Better Aligning Text-to-Image Models with Human Preference. arXiv:2303.14420 [cs.CV]

THANK YOU!