VIRTUAL TRY-ON CLOTHES

STUDENTS	ID
Basmala Akram	2103159
Bassant Mohamed	20221376715
Henar Elsayed	2103131
Nureen Ehab	20221465124
Zeinab Mohamed	20221310251

SUPERVISOR

Dr. Nermeen Elkasheif

VIRTUAL TRY-ON CLOTHES

ABSTRACT

In the rapidly evolving landscape of online shopping, Virtual Try-On (VTON) technology offers an innovative solution that combines convenience, personalization, and strategic value. This project focuses on developing a Virtual Try-On extension that allows users to visualize clothing on their own images before purchasing, creating a more immersive and confident shopping experience.

By addressing common challenges such as incorrect sizing, poor fit visualization, and uncertainty in online apparel shopping, this solution not only enhances customer satisfaction and reduces return rates but also brings substantial benefits to stakeholders. E-commerce platform owners and retailers can leverage this extension to differentiate their services, increase customer retention, boost conversion rates, and modernize their brand image.

Built using modern technologies and deep learning frameworks, the system delivers an intuitive, engaging, and realistic virtual fitting room experience. This user-friendly extension ensures a seamless shopping journey for consumers, while offering website owners a powerful tool to stay competitive, reduce logistical burdens from returns, and build stronger, data-driven customer relationships.

TABLE OF CONTENTS

TABLE OF CONTENTS

03. CHAPTER THREE

04. CHAPTER FOUR

TABLE OF CONTENTS

05. CHAPTER FIVE

TABLE OF CONTENTS

Chapter One

INTRODUCTION TO THE PROJECT

1
INTRODUCTION TO THE PROJECT

1.1
INTRODUCTION

In the evolving world of online shopping, Virtual Try-On technology offers a transformative way to blend fashion with convenience. Our VTON clothes extension aims to provide a smooth and engaging platform where users can digitally try on clothing items. This innovation delivers a unique and interactive shopping experience, allowing customers to visualize how garments fit and look on them before making a purchase.

1.2
PROBLEM STATEMENT

With the rise of online shopping, consumers face significant challenges in determining how clothing items will look on their own bodies before making a purchase. Additionally, the current practice of trying on clothes in physical stores poses a hygiene concern, as garments may have been previously worn by others, potentially transmitting diseases. This inability to try on clothes virtually, combined with health concerns, leads to hesitation, incorrect sizing choices, and a lack of personalization, contributing to high return rates and reduced customer satisfaction. Existing solutions, such as static images or limited virtual tools, fail to provide users with a realistic sense of fit, hygiene, or style tailored to individual preferences.

1
INTRODUCTION TO THE PROJECT

1.3
GOAL OF PROJECT

The primary goal of this project is to design and implement a Virtual Try-On (VTON) extension that seamlessly integrates with leading e-commerce platforms. This extension will allow users to upload a single photo and instantly visualize how garments would look and fit on their bodies, delivering a realistic, engaging, and hygienic online shopping experience. By enabling users to make more informed purchase decisions, the solution aims to significantly reduce return rates, address health-related concerns associated with physical try-ons, and ultimately enhance overall customer satisfaction.

1.4
SCOPE

The Virtual Try-On (VTON) clothes extension encompasses core features such as realistic garment visualization, image-based try-on capability, and a user-friendly interface that integrates directly with popular e-commerce platforms. The system leverages deep learning techniques for accurate human parsing, pose estimation, and clothing alignment to deliver a highly realistic and personalized fitting experience , Both the VTON model and the graphical user interface (GUI) were developed using Python, utilizing libraries such as PyTorch or TensorFlow for the model, and Streamlit for building the desktop-based GUI. This allows for a fully integrated solution that runs smoothly on local systems or can be adapted into web-based platforms in future iterations.

1
INTRODUCTION TO THE PROJECT

1.5
MOTIVATION

Online shopping has become extremely popular, offering convenience and variety. However, one big problem remains—customers cannot try on clothes before buying. This leads to doubts about how clothes will fit or look, causing wrong choices, dissatisfaction, and many returns.

The idea of adding a Virtual Try-On feature to an online shopping platform is to solve this problem. It lets users see how clothes would look on them using their own pictures, making shopping more personal and interactive. This builds confidence in buying decisions and reduces returns.

The Virtual Try-On feature also makes shopping more fun by allowing customers to try out different styles and outfits. Using advanced technology like AI, our extension aims to create a smooth, and user-friendly experience for everyone.

1
INTRODUCTION TO THE PROJECT

1.6
CURRENTLY AVAILABLE SOLUTIONS

Static Images and Videos

Online stores display clothing through static images or videos featuring models. While these provide an idea of how the garment looks, they lack personalization, making it hard for customers to imagine how the clothes will fit on their own bodies.

Virtual Mannequins

Certain websites allow users to customize virtual mannequins with approximate body shapes and sizes. However, these mannequins are not precise and fail to provide a realistic experience of trying on clothes.

Size Charts and Customer Reviews

Many online retailers rely on detailed size charts and customer reviews to help users choose the right fit. While helpful, these tools are often inconsistent across brands and depend heavily on subjective feedback. They do not provide a visual or personalized experience, leaving users uncertain about how a garment will actually look or fit on their body.

1
INTRODUCTION TO THE PROJECT

1.6
CURRENTLY AVAILABLE SOLUTIONS

Fit Prediction Tools

Some sites use AI-based fit prediction tools that recommend sizes based on user inputs like height, weight, and previous purchases. However, these do not offer a visual representation of the garment on the user.

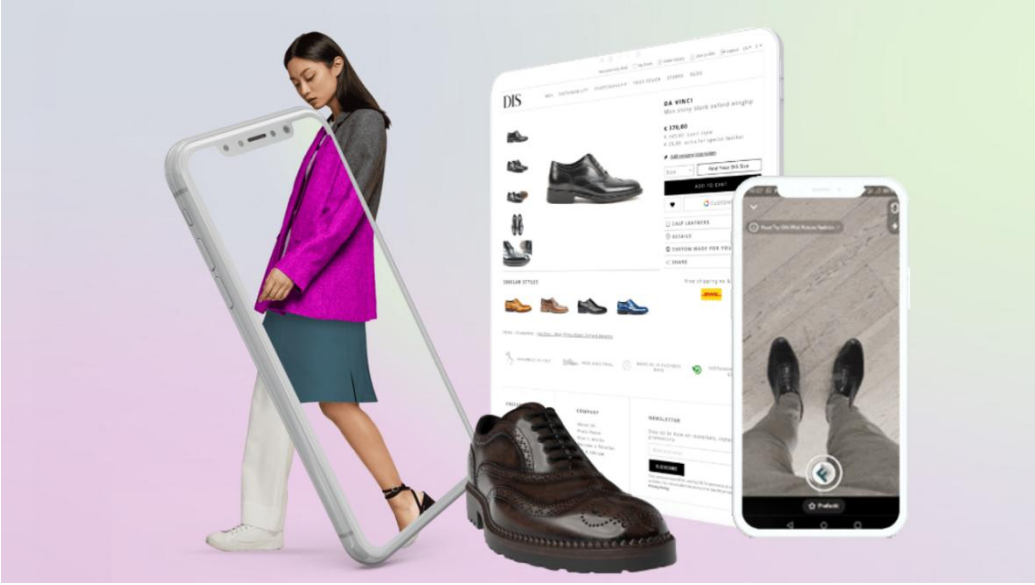
1.7
WHY IS VIRTUAL TRY-ON IMPORTANT ?

Virtual Try-On technology plays a significant role in modern society by enhancing the online shopping experience and promoting sustainability. It reduces the need for physical trials, saving time and minimizing the inconvenience of returns. Additionally, it addresses hygiene concerns associated with trying on clothes in physical stores, as garments may have been previously worn by others, potentially transmitting diseases. By offering a virtual alternative, it ensures a safer and more comfortable shopping experience.

1
INTRODUCTION TO THE PROJECT

1.7
WHY IS VIRTUAL TRY-ON IMPORTANT ?

The technology also supports inclusivity by providing personalized recommendations that cater to diverse body types and preferences. Moreover, Virtual Try-On solutions contribute to environmental sustainability by reducing returns and waste associated with packaging and transportation. It empowers consumers to make informed purchasing decisions, increasing satisfaction and confidence in online shopping. As the retail industry shifts towards digital transformation, Virtual Try-On becomes a vital tool in bridging the gap between physical and virtual retail spaces while addressing health and environmental concerns.



1
INTRODUCTION TO THE PROJECT

1.8
STATISTICS

1.8.1
Challenges in Traditional and Online Shopping

High Return Rates

ONLINE SHOPPING In 2024, the return rate for online purchases was approximately 16.9%, meaning nearly 17 out of every 100 products sold online were returned. This is significantly higher than the return rate for brick-and-mortar stores.

IN-STORE SHOPPING The return rate for in-store purchases was about 8.71%, which, while lower than online returns, still represents a substantial volume of returned merchandise.

Financial Impact of Returns

ONLINE SALES Returns accounted for $362.2 billion, or 24.5% of online sales revenue in 2024.

IN-STORE SALES Returns from brick-and-mortar stores amounted to $323.7 billion, or 8.71% of in-store sales revenue in the same year.

1
INTRODUCTION TO THE PROJECT

1.8
STATISTICS

1.8.1
Challenges in Traditional and Online Shopping

Consumer Behavior and Preferences

ONLINE SHOPPING Approximately 63% of consumer spending in the USA occurred online in 2023, indicating a strong preference for online shopping

IN-STORE SHOPPING Despite the convenience of online shopping, 45% of consumers primarily shop in brick-and-mortar stores, and 72% shop in stores on a weekly basis.

1.8.2
VTON Addresses Challenges

Reduction in Return Rates

ENHANCED DECISION-MAKING Virtual try-on tools allow customers to visualize products on themselves, leading to more informed purchasing decisions. This has been shown to reduce return rates by up to 64%.

1
INTRODUCTION TO THE PROJECT

1.8
STATISTICS

1.8.2
VTON Addresses Challenges

Reduction in Return Rates

RETAILER SUCCESS STORIES Brands implementing virtual try-on technology have reported significant reductions in return rates. For instance, Macy's reduced its return rate to less than 2% after introducing virtual fitting rooms.

Increased Sales and Customer Confidence

BOOST IN SALES The adoption of virtual try-on technology has led to a 2.5 times increase in sales conversion rates for some brands

IMPROVED CUSTOMER EXPERIENCE Virtual try-on experiences enhance customer confidence by providing a realistic preview of products, leading to higher satisfaction and reduced hesitation during the purchasing process.

1.8.3
Growth of VTON Market

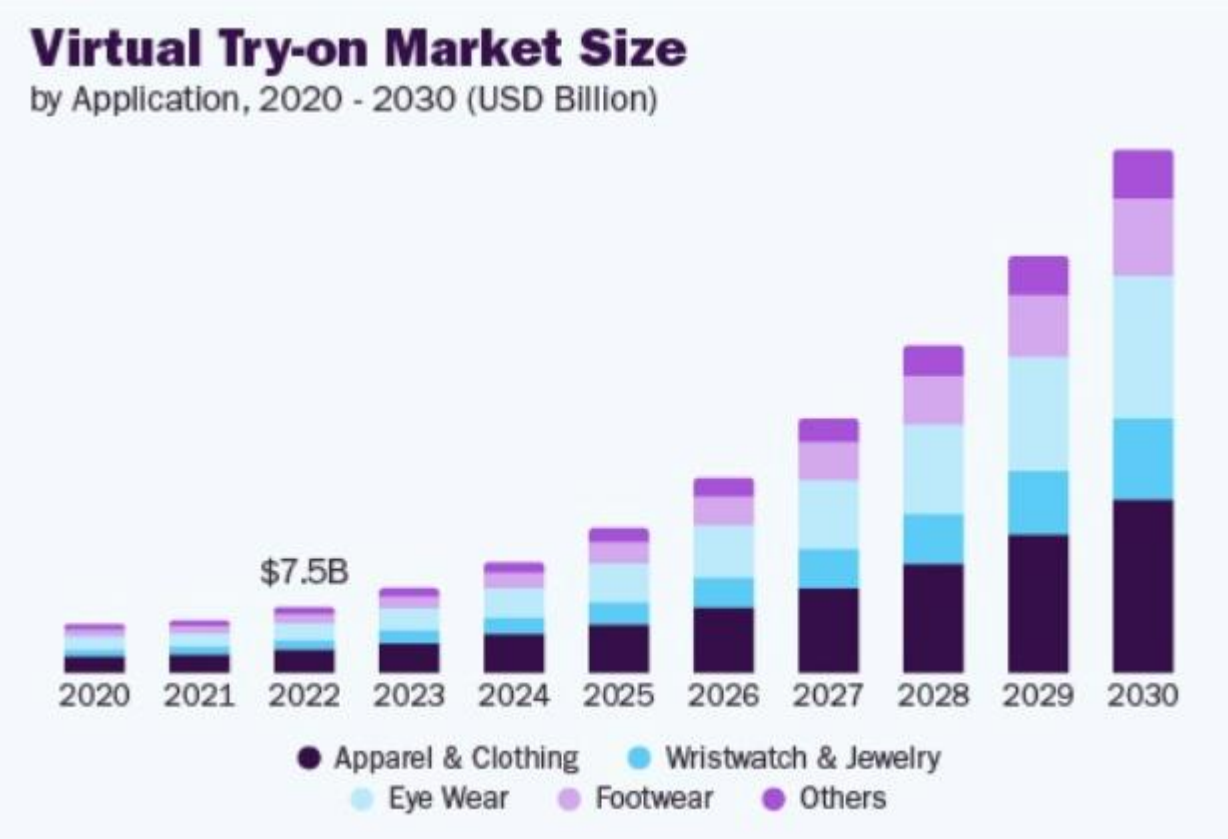
MARKET SIZE The global virtual try-on market was valued at $9.17 billion in 2023 and is projected to grow at a compound annual growth rate (CAGR) of 26.4% from 2024 to 2030, reaching $46.42 billion by 2030.

1
INTRODUCTION TO THE PROJECT

1.8
STATISTICS

1.8.3
Growth of VTON Market

REGIONAL GROWTH The virtual try-on market in the U.S. is anticipated to grow at a CAGR of over 24% from 2024 to 2030, driven by widespread smartphone usage and high-speed internet availability.



Chapter Two

TARGET AUDIENCE

2
TARGET AUDIENCE

2.1
INTRODUCTION

To ensure the successful development of our Virtual Try-On (VTO) clothes system, it is essential to understand the needs, preferences, and behaviors of our target users. This chapter presents the methods and findings of both surveys and interviews conducted with potential users and stakeholders. The goal is to gather insights that will guide the design, functionality, and usability of the system.

The VTO system is intended to serve as an extension of popular online shopping platforms such as SHEIN, Zara, and Amazon. As such, it targets everyday online shoppers—primarily females aged 18 and above—who frequently purchase clothing through e-commerce websites. To validate the necessity and practicality of the virtual try-on feature, we conducted a survey to collect quantitative data on user behaviors, challenges, and expectations related to online fashion shopping. Additionally, semi-structured interviews were conducted to gain deeper qualitative insights into individual user experiences, motivations, and concerns.

By combining both survey and interview approaches, we aim to obtain a comprehensive understanding of user needs, system requirements, and areas for improvement. The findings from this chapter will play a critical role in shaping user-centered design decisions and ensuring the final product is both functional and intuitive for end users and business partners alike.

2
TARGET AUDIENCE

2.2
USER CHARACTERSTICS

Age Range

The users will primarily be individuals aged 18 and above, with a large portion falling between 18–35, as this group shops online most frequently.

Gender Distribution

While the platform will support all users, it is expected that females will make up the majority of the user base due to their higher engagement in fashion e-commerce and interest in appearance-based shopping tools.

Technical Experience

The typical user will have basic computer and smartphone skills, including experience using websites or mobile apps for online shopping. They will be comfortable with graphical user interfaces and standard online navigation patterns (like scrolling, clicking, swiping, and zooming).

Knowledge of Technology

Most users won’t have any knowledge of advanced technology such as networking, file transfers, or machine learning. Therefore, the system should be intuitive, visually driven, and require minimal input.

2
TARGET AUDIENCE

2.2
USER CHARACTERSTICS

Time & Behavior

Typical users will be busy individuals looking for a quick and efficient experience. The try-on feature should allow users to preview clothing in seconds, with minimal loading time and no complex setup. Speed, clarity, and ease of use are key.

Access Credentials

The user should only need a username (email or phone number) and a password to register or log in. Social login options (e.g., Google, Facebook) could improve accessibility.

Motivation for Using VTO

Users are often unsure how clothes will look on them in real life. A virtual try-on tool helps build confidence in purchase decisions, reduces returns, and increases customer satisfaction.

2
TARGET AUDIENCE

2.3
DATA COLLECTION

2.3.1
Interviews

We conducted interviews with several online shoppers, and they expressed interest in our idea because they often faced challenges in determining how clothes would look and fit before making a purchase. They emphasized the need for a more interactive and personalized shopping experience to reduce uncertainty and improve satisfaction. They also highlighted the importance of features like virtual try-on tools to enhance confidence in their buying decisions and minimize returns.

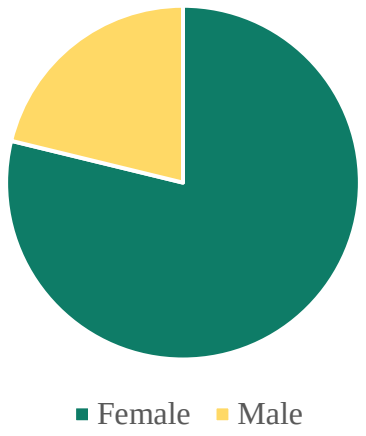
HOW INTERVIEWS HELPED?

- ✓ Understanding user needs
- ✓ Identify pain points in online shopping and virtual try-on adoption.
- ✓ Gather insights into preferences for interface design and usability.
- ✓ Feature Development.
- ✓ Highlight areas for improvement based on user feedback.
- ✓ Market Validation.
- ✓ Assess whether the virtual try-on concept solves real problems for target users.
- ✓ Trust and Security Insights.
- ✓ Ensure privacy and security concerns are addressed early in development.
- ✓ Build confidence in the AI-driven process.

2
TARGET AUDIENCE

2.3.2

Survey



Purpose gain insights into the demographics of our audience and enhance the user experience of our virtual try-on platform.

Benefits

- ✓ Understanding Target Audience by gathering gender data helps identify the primary demographic using our platform, enabling a better understanding of its appeal and reach.
- ✓ Customization and Personalization which means that the Gender-specific preferences can influence design choices, such as tailored clothing categories or unique virtual try-on experiences, enhancing user satisfaction.

Discussion of results

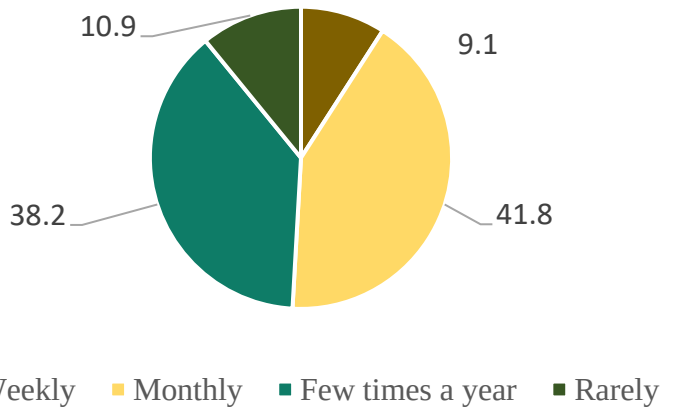
- ✓ Most of the respondents were females, these results might help us when creating the extension as females are interested in our service.

2
TARGET AUDIENCE

2.3.2

Survey

How Often do you shop for Clothes online?



Purpose To understand user habits and their frequency of engagement with online shopping.

Benefits

- ✓ Helps determine how much our target audience relies on online shopping.
- ✓ Insights into how often people might use our virtual try-on feature.

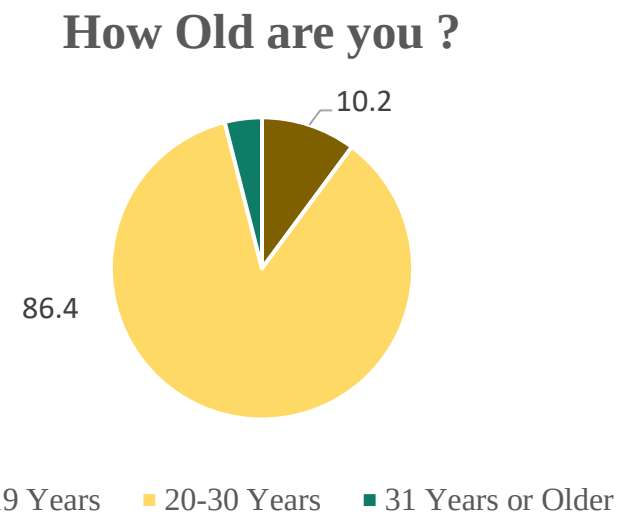
Discussion of results

- ✓ Most people chose monthly which is considered a moderate rate, asking this question helps us determine how frequently people will use our service.

2
TARGET AUDIENCE

2.3.2

Survey



Purpose To categorize respondents into age groups for better targeting.

Benefits

- ✓ Allows tailoring features, design, and marketing strategies to age-specific preferences.
- ✓ Identifies whether our service should cater more to younger or older audiences.

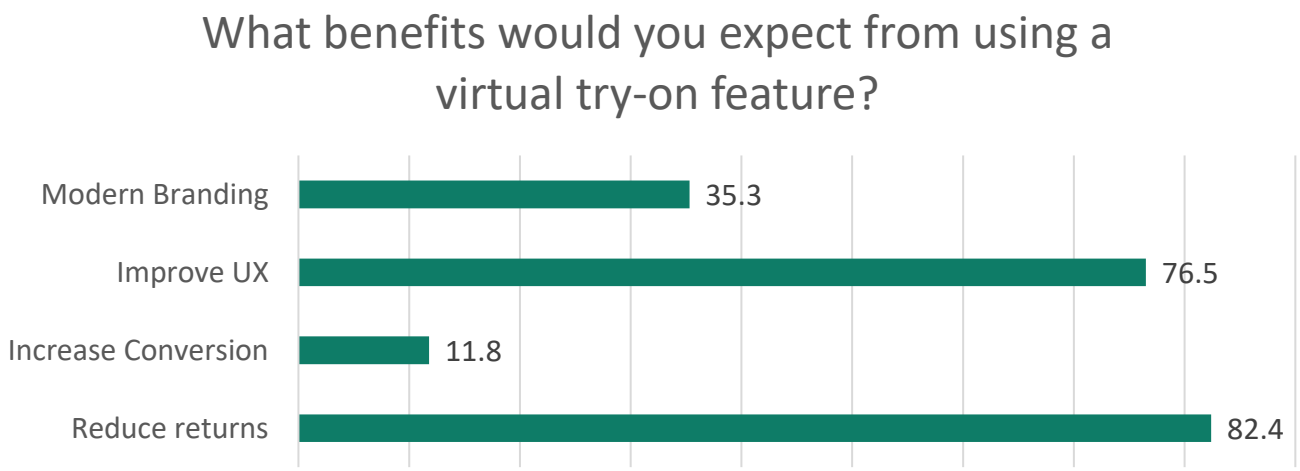
Discussion of results

- ✓ Most of the respondents aged between 20-30 (teenagers and youth) which will help us when developing the extension to make features that attract those who are interested the most in our service.

2
TARGET AUDIENCE

2.3.2

Survey



Purpose To identify what value users and business stakeholders expect from the implementation of a virtual try-on (VTO) system. By understanding their expectations, we can better align the development priorities of the system.

Benefits

- ✓ Retailers or partner platforms can use this information to justify the investment in VTO technology by matching it to clear customer expectations.
- ✓ Helps determine whether the goals of users align with what the VTO system is designed to achieve.

Discussion of results

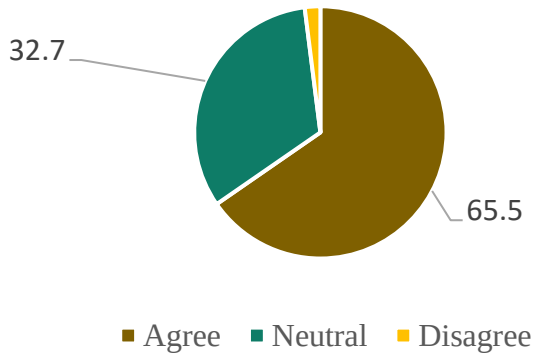
- ✓ This suggests users are often disappointed with how clothes fit or look in person, leading to frequent returns. The VTO system should therefore focus on realistic rendering and body fitting accuracy.

2
TARGET AUDIENCE

2.3.2

Survey

Would You Feel More Confident Purchasing Clothes Online After Using A Virtual Dressing Room App?



Purpose To assess the perceived value of a virtual try-on solution.

Benefits

- ✓ Helps validate the relevance of your idea.
- ✓ Understands user trust in virtual dressing rooms to improve their shopping experience.

Discussion of results

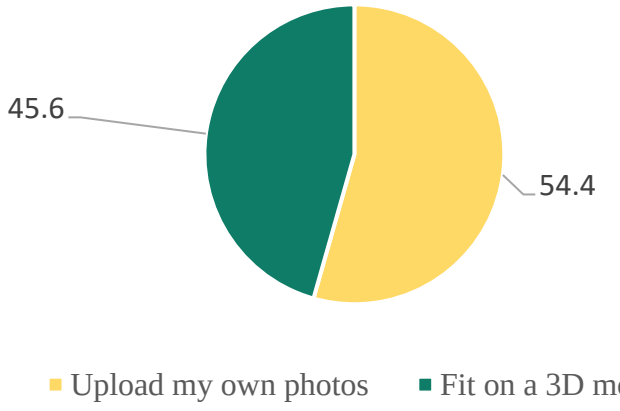
- ✓ Most people agree on feeling more confident when purchasing clothes online after using virtual dressing room app meaning That there will be many advocates for our service.

2
TARGET AUDIENCE

2.3.2

Survey

Would you rather to be able to upload your own photos to try on clothes virtually or fit on a 3D model of your body?



Purpose To explore user preferences for interaction with the website.

Benefits

- ✓ Helps prioritize technical development for either photo uploads or 3D modeling.
- ✓ Improves user experience by implementing the most preferred feature.

Discussion of results

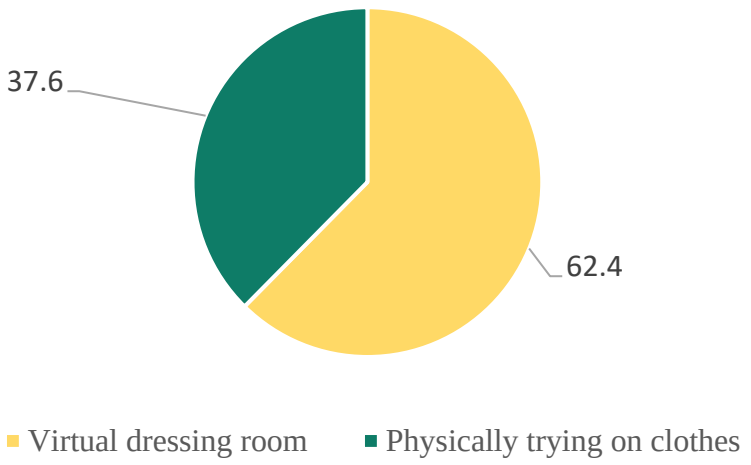
- ✓ Most people prefer realistic and personalized try-ons. Focus on improving image alignment and processing for user-uploaded photos.

2
TARGET AUDIENCE

2.3.2

Survey

Would you prefer to use a virtual dressing room app over physically trying on clothes in stores?



Purpose To evaluate the potential of replacing in-store try-ons.

Benefits

- ✓ Gauges the likelihood of users adopting virtual solutions over traditional methods.
- ✓ Helps position the extension as a convenient alternative to in-store shopping.
- ✓ Guides feature development to mimic or enhance the in-store experience.

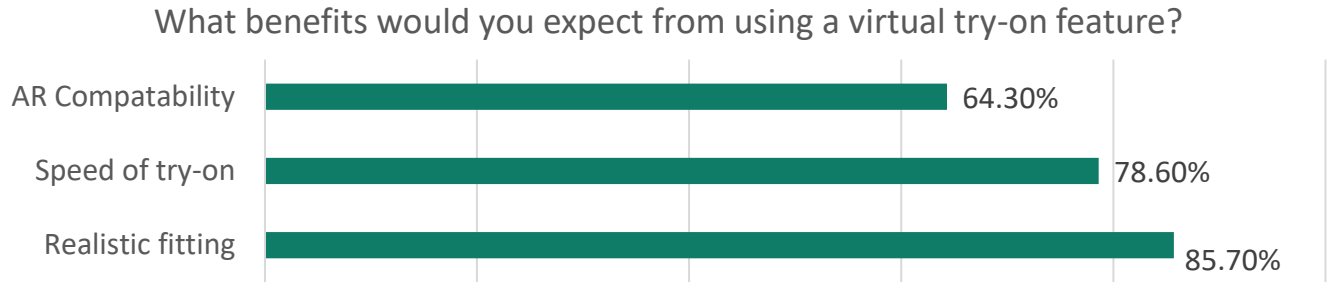
Discussion of results

- ✓ Most people chose virtual meaning that people are starting to replace in store shopping with online shopping which helps our service.

2
TARGET AUDIENCE

2.3.2

Survey



Purpose To determine which core features users value most in a virtual try-on (VTO) system. Since different users prioritize different aspects, some might want fast interaction or immersive technology, understanding their preferences helps the development team focus on building features that truly matter to the audience

Benefits

- ✓ Delivering the most valued features increases the likelihood of user satisfaction and adoption of the system.
- ✓ Understanding what users expect can help the system stand out from competitors by offering the most-desired functionality.

Discussion of results

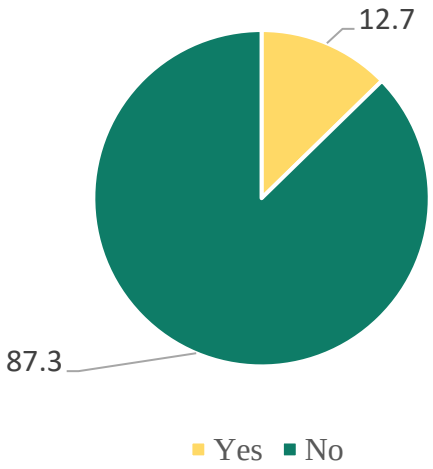
- ✓ This result emphasizes the importance of image processing, and fit modeling.
- ✓ The audience expects a fast, seamless experience with minimal delays. This indicates that performance optimization is crucial. Long loading times or complex steps may lead to frustration and drop-offs, especially among busy users.

2
TARGET AUDIENCE

2.3.2

Survey

In the past year , Have you used any virtual try-on or augmented reality tools while shopping online?



Purpose To measure user familiarity with virtual try-on technologies.

Benefits

- ✓ Identifies how aware our audience is of similar technologies.

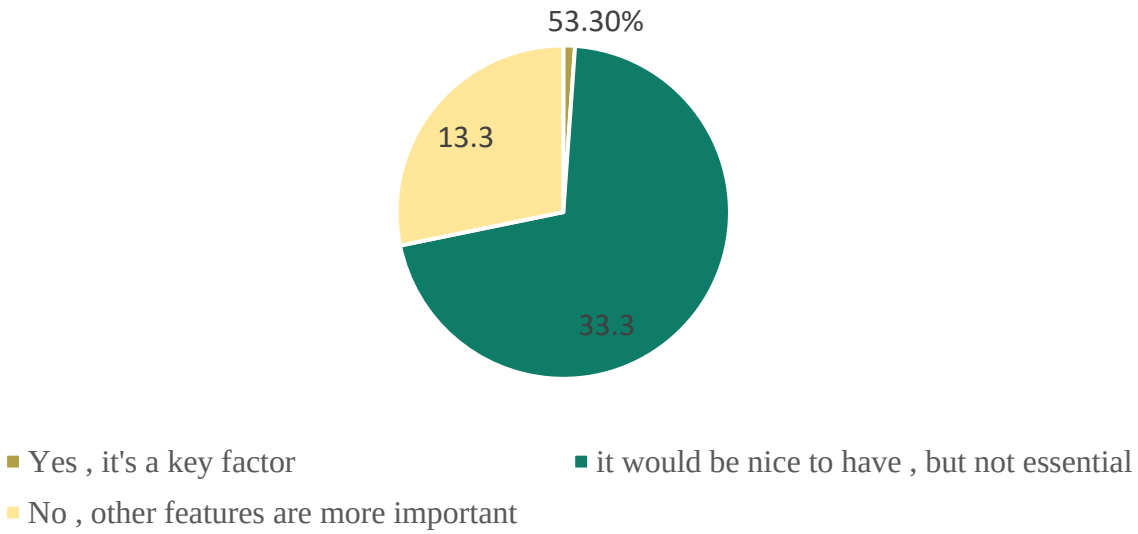
Discussion of results

- ✓ Most people didn’t use augmented reality tools while shopping online meaning that we should use techniques people be familiar with and be easy to use.

2
TARGET AUDIENCE

2.3.2

Survey



Purpose To help determine whether VTON is perceived as a must-have, a nice-to-have, or a low-priority feature in the eyes of potential customers.

Benefits

- ✓ If many users see VTON as a key factor, it can be emphasized in promotions, landing pages, and ads as a competitive advantage.

Discussion of results

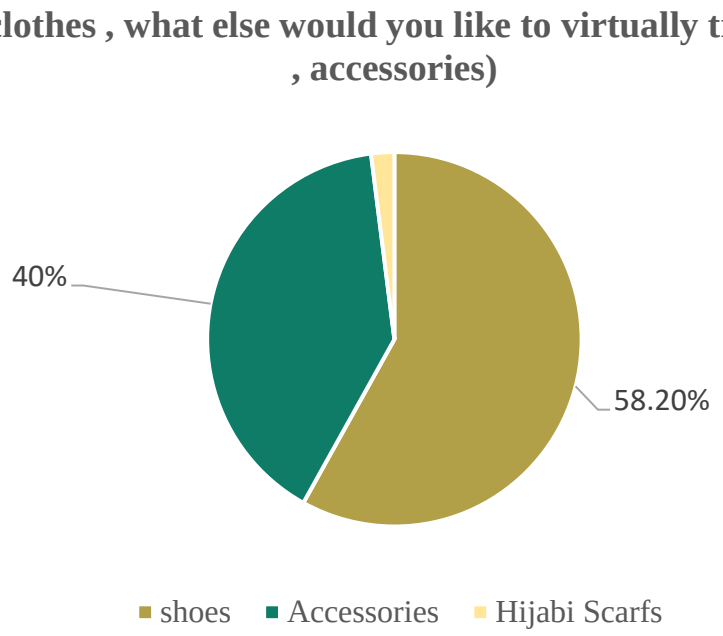
- ✓ **"Yes, it's a key factor"** is most selected:

This indicates VTON is a decisive feature for users. Platforms offering it will likely attract more traffic and loyalty, especially among users who care about visualizing their purchases before buying. The business should position VTON as a core competitive feature.

2
TARGET AUDIENCE

2.3.2

Survey



Purpose To explore opportunities for expanding your virtual try-on offerings.

Benefits

- ✓ Identifies potential areas for growth beyond clothing.

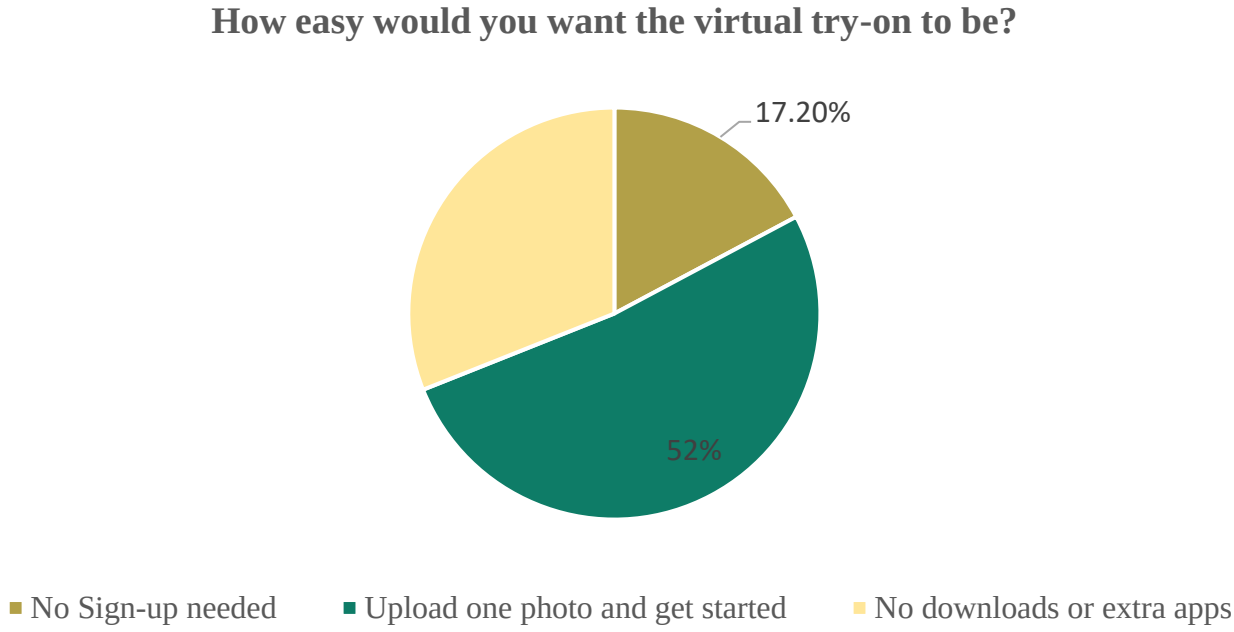
Discussion of results

- ✓ Most people chose shoes, a feature that might be useful for people and we might add it to our service.

2
TARGET AUDIENCE

2.3.2

Survey



Purpose The main purpose of this question is to assess user expectations regarding convenience and ease of access.

Benefits

- ✓ Identifies potential areas for growth beyond clothing.
- ✓ If users prefer uploading just one image to start, developers can prioritize fast, image-based onboarding over more complex avatar setup flows.

Discussion of results

- ✓ **"Upload one photo and get started"** is most selected:

Users are comfortable uploading a photo but want the process to be fast and simple. This suggests prioritizing a one-step onboarding process, possibly with clear privacy policies to ensure trust.

2
TARGET AUDIENCE

2.4
COMPETITIVE ANALYSIS

It is a strategy that involves researching major competitors to gain insight into their products sales and making tactics. implementing stronger business strategies, warding off competitors, and capturing market share are just a few benefits of conducting a Competitive market analysis.

	Wearfits Tryon	Try Outfit	Viton Try On	Letsy	Closetly
Platform Support	Android, IOS	Android, IOS	Android	Android, IOS	Extension
Types of Try-on	Shoes	Clothes	Makeup	Clothes	Clothes
Augmented Reality	Yes	No	No	No	No
Model Accuracy	Low	Intermediate	High	Intermediate	High
Ease of Use	Basic	Basic	Medium	Medium	Basic
Price/ Subscription Model	Free	Free trial	Free trial	Free trial	Free trial

	VitonTryOn	Aiuta	Reactive Reality	Try-on.io	Closetly
Platform Support	Android, IOS	Android, IOS	Extension	Web	Extension
Types of Try-on	Clothes	Clothes, and shoes	Clothes	Glasses, clothes, jewellery , bags, watches, and makeup	Clothes
Augmented Reality	No	No	Yes	Yes	No
Model Accuracy	Intermediate	Intermediate	_	Intermediate	High
Ease of Use	Medium	Medium	Medium	Basic	Basic
Price/ Subscription Model	Free trial	Free trial	Paid	Free	Free trial

Chapter Three

PROJECT MANAGEMENT

3
PROJECT MANAGEMENT

3.1
INTRODUCTION

The process models are proposed to bring the order of the software development, and bring a useful structure for the software engineering teams by defining the activities, actions, tasks, milestones and work products to produce high quality software.

3.2
AGILE MODEL

The Agile Model is an iterative and incremental software development life cycle (SDLC) model that emphasizes flexibility and collaboration. The development process progresses through repeated cycles, called sprints, allowing for continuous improvement and adaptation to change. Feedback from stakeholders is incorporated at every stage, ensuring the project evolves to meet the changing needs of the business.

3.2.1
Reasons for Choosing the Agile Model

Flexible and Adaptive Requirements

Suitable for projects where requirements are likely to change or evolve over time. Agile thrives in environments where flexibility is key, making it ideal for projects with dynamic scopes.

3
PROJECT MANAGEMENT

3.2.1
Reasons for Choosing the Agile Model

Collaborative and Interactive Approach

The iterative nature of the Agile model promotes active collaboration between developers, stakeholders, and customers. This fosters better communication and allows for the alignment of goals throughout the development process.

Emphasis on Working Software

The focus is on delivering small, incremental improvements to the system regularly, which ensures that the software is functional at every stage and ready for deployment.

Continuous Feedback and Improvement

Regular feedback from stakeholders at the end of each sprint enables the team to make adjustments, prioritize new features, or refine existing ones, ensuring the system evolves based on real-world input.

3
PROJECT MANAGEMENT

3.2.1
Reasons for Choosing the Agile Model

Flexible Timelines and Budgets

Since Agile works with iterative cycles, timelines and budgets are continuously reviewed and adjusted based on progress, enabling more accurate forecasting and adjustments as the project unfolds.

No Fixed Phases or Milestones

Unlike traditional models, Agile does not follow a strict linear sequence. Instead, the project evolves through iterative sprints, which fosters rapid delivery and allows for ongoing enhancements.

Easier Testing and Validation

Testing is integrated throughout the development process, with each sprint delivering a version of the system that can be tested. This ensures issues are identified and addressed early on, making testing a continuous activity.

Suitable for projects where ongoing testing and validation lead to better-quality products.

3
PROJECT MANAGEMENT

3.2.1
Reasons for Choosing the Agile Model

High Client Involvement

Clients are actively involved throughout the project, ensuring that their needs and priorities are met at every stage.

Best for Projects with Evolving Technology

Agile is ideal when the technology or tools used are new or evolving. The iterative nature of the model allows for the adoption of new technologies or practices as the project progresses, minimizing the risks of obsolescence.

3.2.2
Challenges

While the Agile model offers several advantages, potential challenges include difficulties in managing rapidly changing requirements and ensuring consistent collaboration among stakeholders. To mitigate this, a well-defined communication plan and regular sprint reviews will be implemented, ensuring changes are effectively managed and all team members remain aligned with project goals.

3
PROJECT MANAGEMENT

3.2.3
Conclusion

In conclusion, the selection of the Agile model for our project is based on its suitability for dynamic and evolving requirements. The iterative approach, flexibility, and emphasis on collaboration align with the nature of the project, enabling continuous feedback and rapid adaptation to user needs and technological advancements .

This decision has been made after careful consideration of the project's characteristics, and we are confident that the Agile model will foster innovation and user-centric development, ensuring the virtual try-on clothes extension meets both client and stakeholder expectations while delivering a seamless and engaging user experience.

3
PROJECT MANAGEMENT

3.3
SYSTEM REQUIREMENTS

The system requirements define the functional and non-functional specifications necessary for the successful development and operation of the virtual try-on extension. This extension allows users to upload or select person and clothing images, process them using computer vision techniques, and generate a realistic visualization of the person wearing the selected clothing. These requirements ensure that the system delivers a seamless, interactive, and user-friendly experience.

The functional requirements detail the specific behaviors and features the system must implement, such as image input, processing, try-on generation, customization, and session handling.

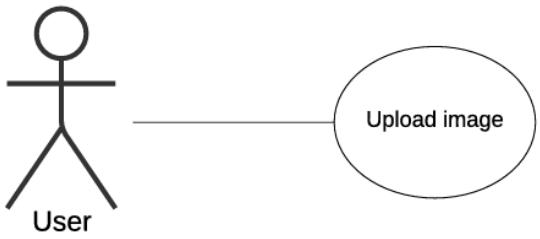
On the other hand, non-functional requirements address the quality attributes of the system, including performance, reliability, usability, and scalability.

This section provides a structured breakdown of these requirements to guide development, maintain consistency, and meet user expectations.

3
PROJECT MANAGEMENT

3.3.1
Functional Requirements

Upload Images



Actors

- ✓ User

Flow

- ✓ User clicks the “Upload Image” button.
- ✓ System allows user to upload a person image (front-facing, full body).
- ✓ System allows user to upload a clothing image (top wear).
- ✓ System confirms upload success.

Pre-conditions

- ✓ User is on the main interface of the virtual try-on system.
- ✓ The user has selected the “Upload Images" option.

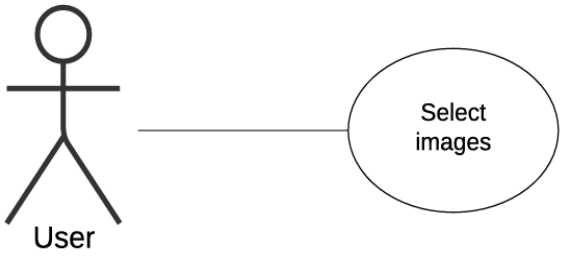
Post-condition

- ✓ User has successfully provided person and clothing images for processing.

3
PROJECT MANAGEMENT

3.3.1
Functional Requirements

Select Images



Actors

- ✓ User

Flow

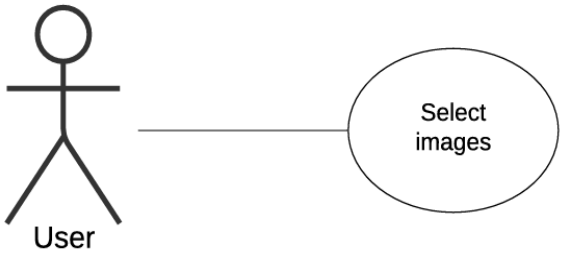
- ✓ User chooses the "Select from Presets" option.
- ✓ The system displays a grid of person images.
- ✓ User clicks a "Select" button under a person image.
- ✓ The system highlights the selected image.
- ✓ The system displays a grid of clothing images.
- ✓ User clicks a "Select" button under a clothing image.
- ✓ The system highlights the selected clothing image.
- ✓ Once both selections are made, the system loads the selected images for processing.

3
PROJECT MANAGEMENT

3.3.1

Functional Requirements

Select Images



Actors

- ✓ User

Pre-conditions

- ✓ User is on the main interface of the virtual try-on system.
- ✓ Preset image folders are available and accessible by the system.
- ✓ The user has selected the "Select from Presets" option.

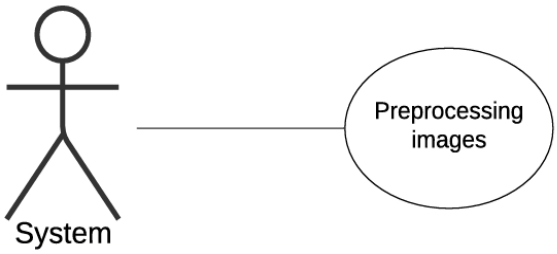
Post-condition

- ✓ The system loads the selected person and clothing images for further processing.

3
PROJECT MANAGEMENT

3.3.1
Functional Requirements

Preprocessing Images



Actors

- ✓ System

Flow

- ✓ System receives uploaded or selected images.
- ✓ System uses techniques to remove backgrounds from both images.
- ✓ Images are resized to 192x256 pixels. (same size as VTON images)
- ✓ Images are normalized on a white background for consistency.

Pre-conditions

- ✓ Person and clothing images are provided by the user.

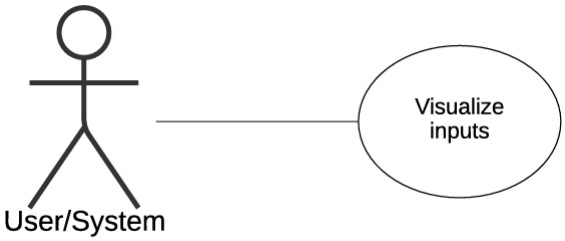
Post-condition

- ✓ Processed images are prepared for model input.

3
PROJECT MANAGEMENT

3.3.1
Functional Requirements

Visualize Inputs



Actors

- ✓ User
- ✓ System

Flow

- ✓ System displays the uploaded or selected person image.
- ✓ System displays the uploaded or selected clothing image.
- ✓ User visually verifies their inputs before proceeding.

Pre-conditions

- ✓ Image processing has been completed.

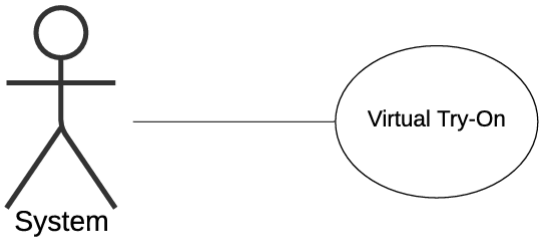
Post-condition

- ✓ User views both inputs and can proceed to try-on generation.

3
PROJECT MANAGEMENT

3.3.1
Functional Requirements

Virtual Try-On



Actors

- ✓ System

Flow

- ✓ System sends the processed images to the virtual try-on model.
- ✓ Model returns an image with the person wearing the selected clothing.
- ✓ System displays the generated try-on result.
- ✓ If generation fails, an error message is shown.

Pre-conditions

- ✓ User inputs are processed and ready.

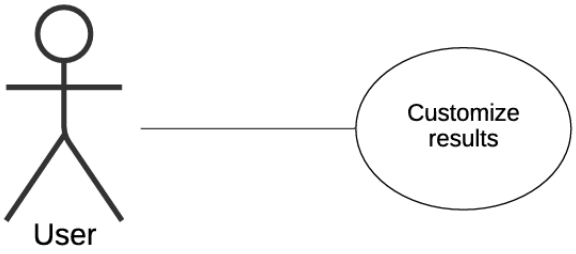
Post-condition

- ✓ User sees the generated try-on result image.

3
PROJECT MANAGEMENT

3.3.1
Functional Requirements

Customize Results



Actors

- ✓ User

Flow

- ✓ User applies grayscale filter.
- ✓ System applies the black-and-white transformation to the result image.
- ✓ User adjusts zoom using a slider interface.
- ✓ User can reset or revert changes.

Pre-conditions

- ✓ A try-on result has been generated.

Post-condition

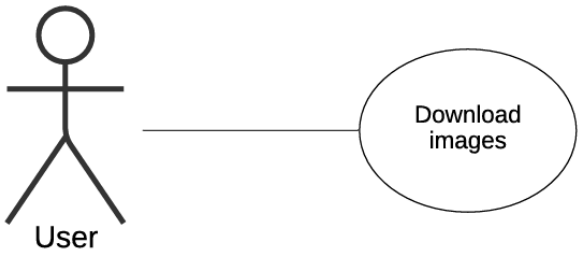
- ✓ User customizes the output image to their preference.

3
PROJECT MANAGEMENT

3.3.1

Functional Requirements

Download Images



Actors

- ✓ User

Flow

- ✓ User clicks the “Download” button.
- ✓ System prompts download of the try-on result (with all customizations applied).

Pre-conditions

- ✓ A try-on result is generated and optionally customized.

Post-condition

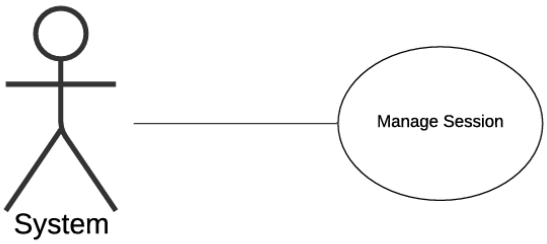
- ✓ User obtains a local copy of the result image.

3
PROJECT MANAGEMENT

3.3.1

Functional Requirements

Manage Session



Actors

- ✓ System

Flow

- ✓ System stores user-uploaded or selected images using st.session_state.
- ✓ System maintains state across user interactions.

Pre-conditions

- ✓ User has uploaded or selected images.

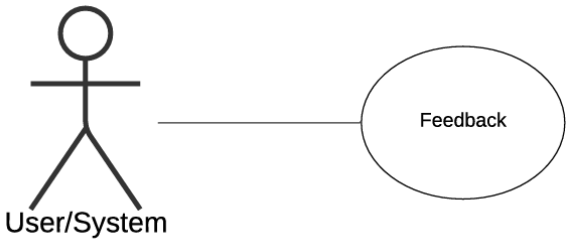
Post-condition

- ✓ System retains user selections and session state throughout their visit.

3
PROJECT MANAGEMENT

3.3.1
Functional Requirements

Feedback



Actors

- ✓ User
- ✓ System

Flow

- ✓ User interacts with various buttons (Upload, Select Preset, Generate, Customize, Download).
- ✓ System provides real-time feedback (e.g., success messages, error alerts).
- ✓ System validates actions and alerts users of missing inputs or invalid steps.

Pre-conditions

- ✓ User has uploaded or selected images.

Post-condition

- ✓ User receives interactive guidance and feedback during their session.

3
PROJECT MANAGEMENT

3.3.2
Non-Functional Requirements

Performance

- ✓ All operations initiated by users should execute promptly to deliver a seamless and responsive experience.
- ✓ The system must sustain optimal performance even during high traffic periods.
- ✓ Image and AI processing for virtual try-ons should be optimized for real-time rendering to avoid delays.

Availability

- ✓ The system must maintain consistent uptime and operate reliably.
- ✓ Servers should be robust and capable of swiftly resolving any arising issues.
- ✓ Users should have 24/7 access to the platform globally.

Scalability

- ✓ The system must scale efficiently and support large number of users.
- ✓ It should handle extensive uploads and user activity without affecting performance.

3
PROJECT MANAGEMENT

3.3.2
Non-Functional Requirements

Maintenance

- ✓ Modular architecture should be employed to simplify enhancements and modifications.

Usability

- ✓ The interface must be simple and intuitive for users with varying levels of expertise.
- ✓ Tools for virtual try-on should be easy to use, requiring minimal steps for outfit customization.
- ✓ Accessibility options, such as alt text for images, should be supported.

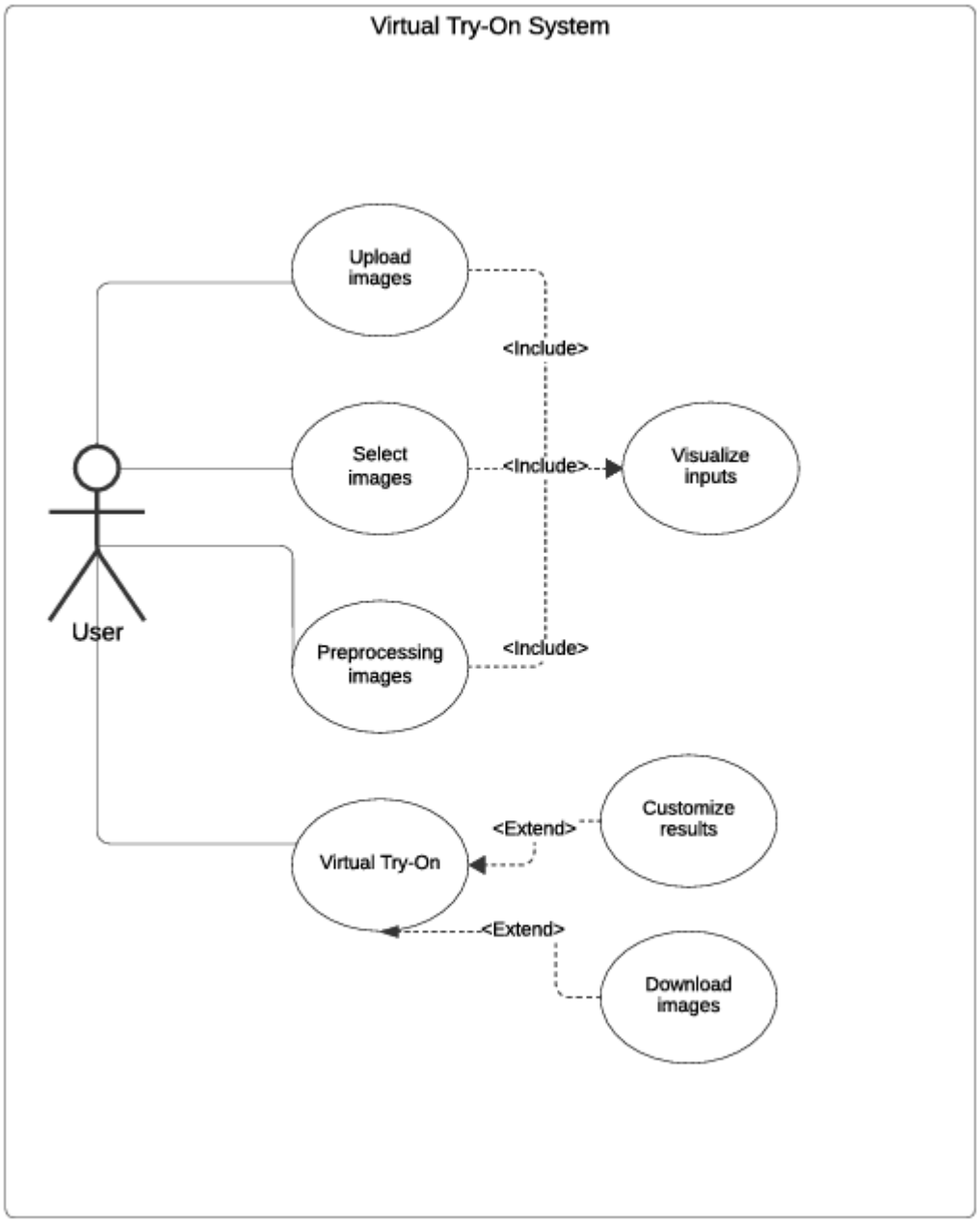
Compatibility

- ✓ The platform must be optimized for all popular browsers and devices, including desktops and smartphones.
- ✓ It should adapt responsively to different screen resolutions and sizes.

Chapter Four

SOFTWARE ARCHITECTURE

4
SOFTWARE ARCHITECTURE
4.1
USE CASE DIAGRAM

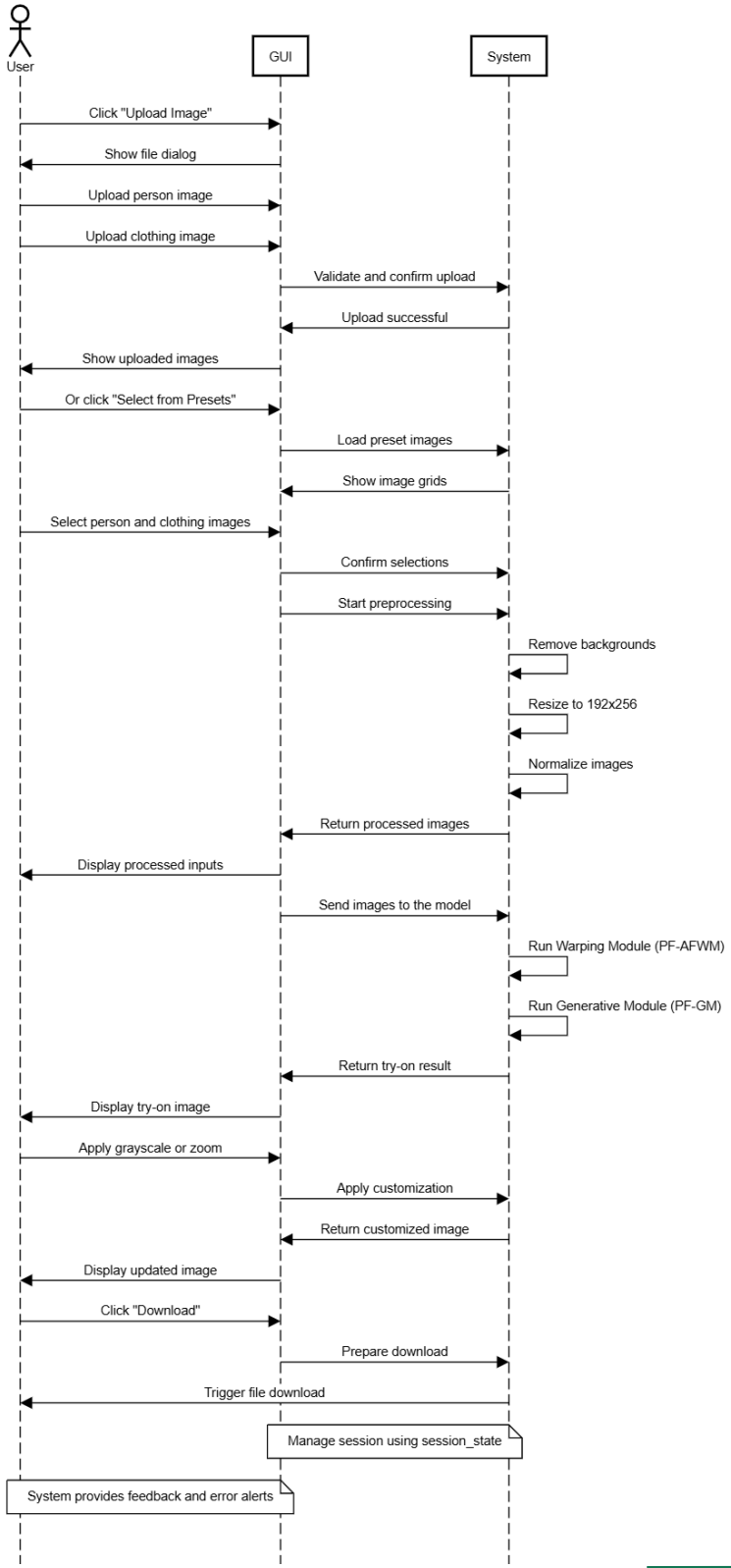


4

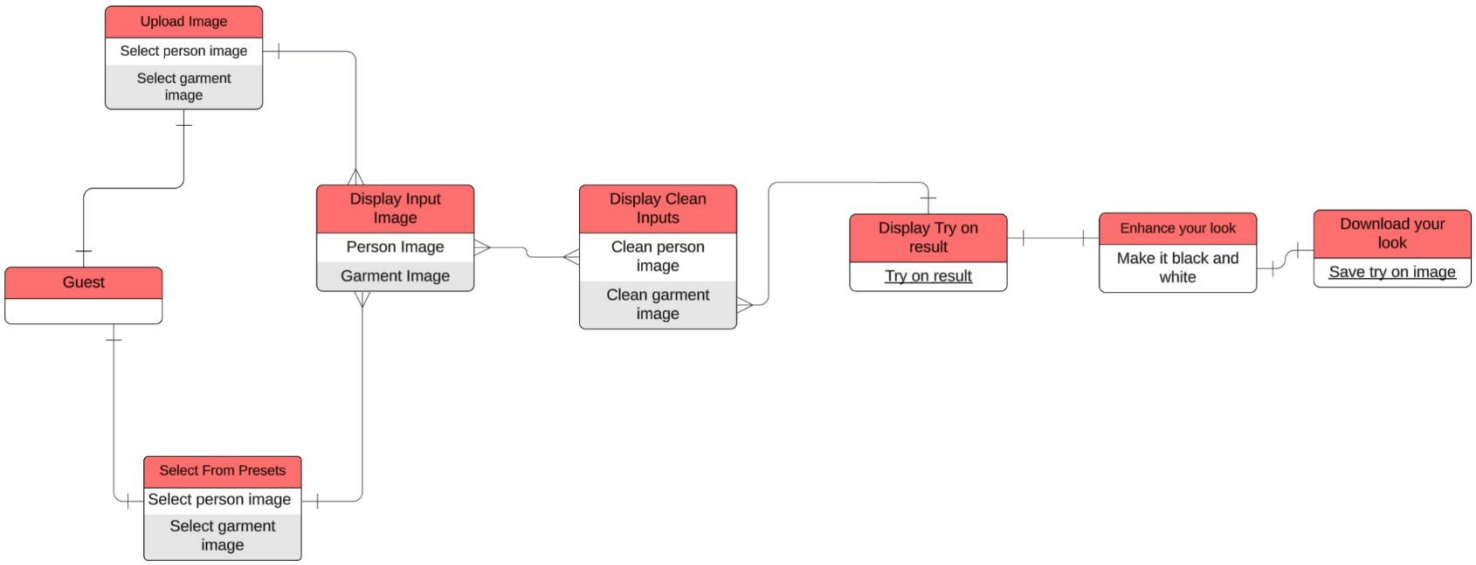
SOFTWARE ARCHITECTURE

4.2

SEQUENCE DIAGRAM

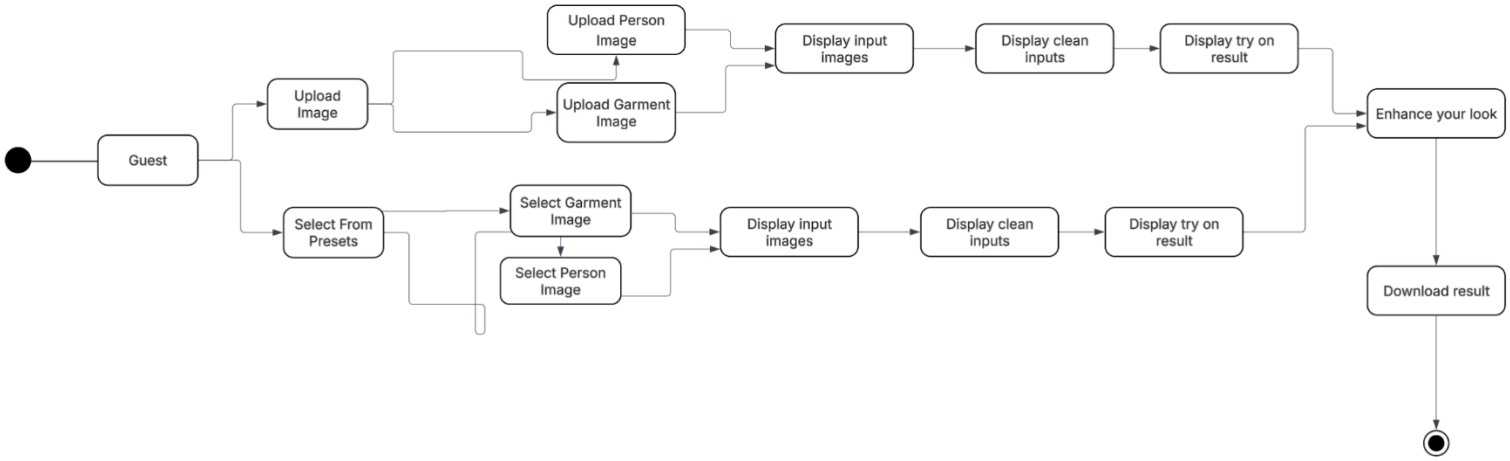


4
SOFTWARE ARCHITECTURE
4.3
ERD



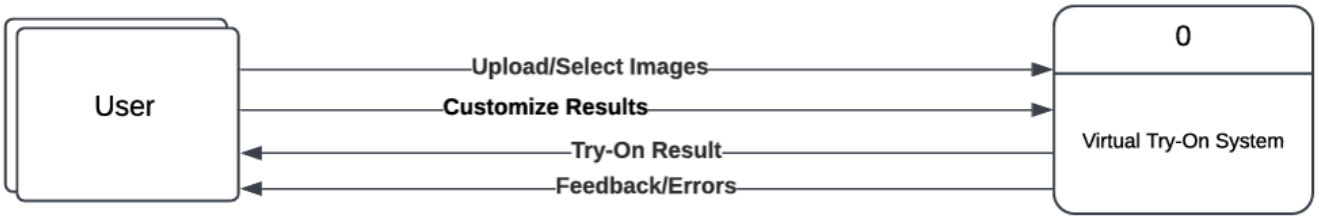
4
SOFTWARE ARCHITECTURE

4.4
STATE DIAGRAM

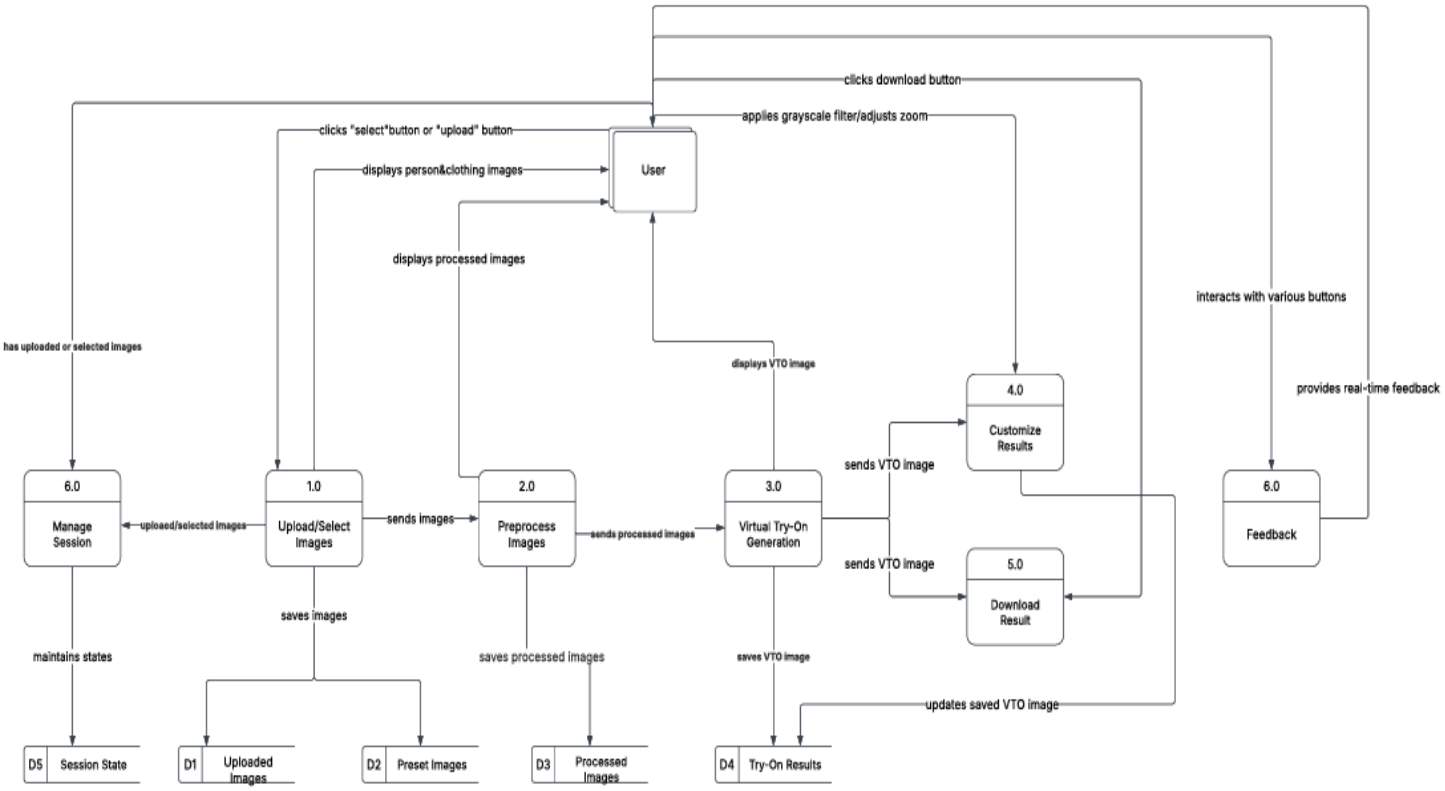


4.5
DATA FLOW DIAGRAM

4.5.1
Context Diagram



4
SOFTWARE ARCHITECTURE
4.5
DATA FLOW DIAGRAM
4.5.2
DFD-level 0



4
SOFTWARE ARCHITECTURE
4.6
PSEUDOCODE

```
FUNCTION VirtualTryOnSystem()

  // 1. Upload Images
  FUNCTION UploadImages()
    DISPLAY "Upload Person Image" button
    IF user clicks THEN
      personImage ← user uploads image
    ENDIF

    DISPLAY "Upload Clothing Image" button
    IF user clicks THEN
      clothingImage ← user uploads image
    ENDIF

    IF personImage AND clothingImage ARE uploaded THEN
      DISPLAY "Upload Successful"
    ELSE
      DISPLAY "Upload Failed: Missing Images"
    ENDIF
  END FUNCTION

  // 2. Select Images
  FUNCTION SelectFromPresets()
    DISPLAY grid of preset person images
    personImage ← user selects image

    IF personImage IS selected THEN
      HIGHLIGHT selected person image
      DISPLAY grid of preset clothing images
      clothingImage ← user selects image

      IF clothingImage IS selected THEN
        HIGHLIGHT selected clothing image
        LOAD personImage AND clothingImage
      ENDIF
    ENDIF
  END FUNCTION
```

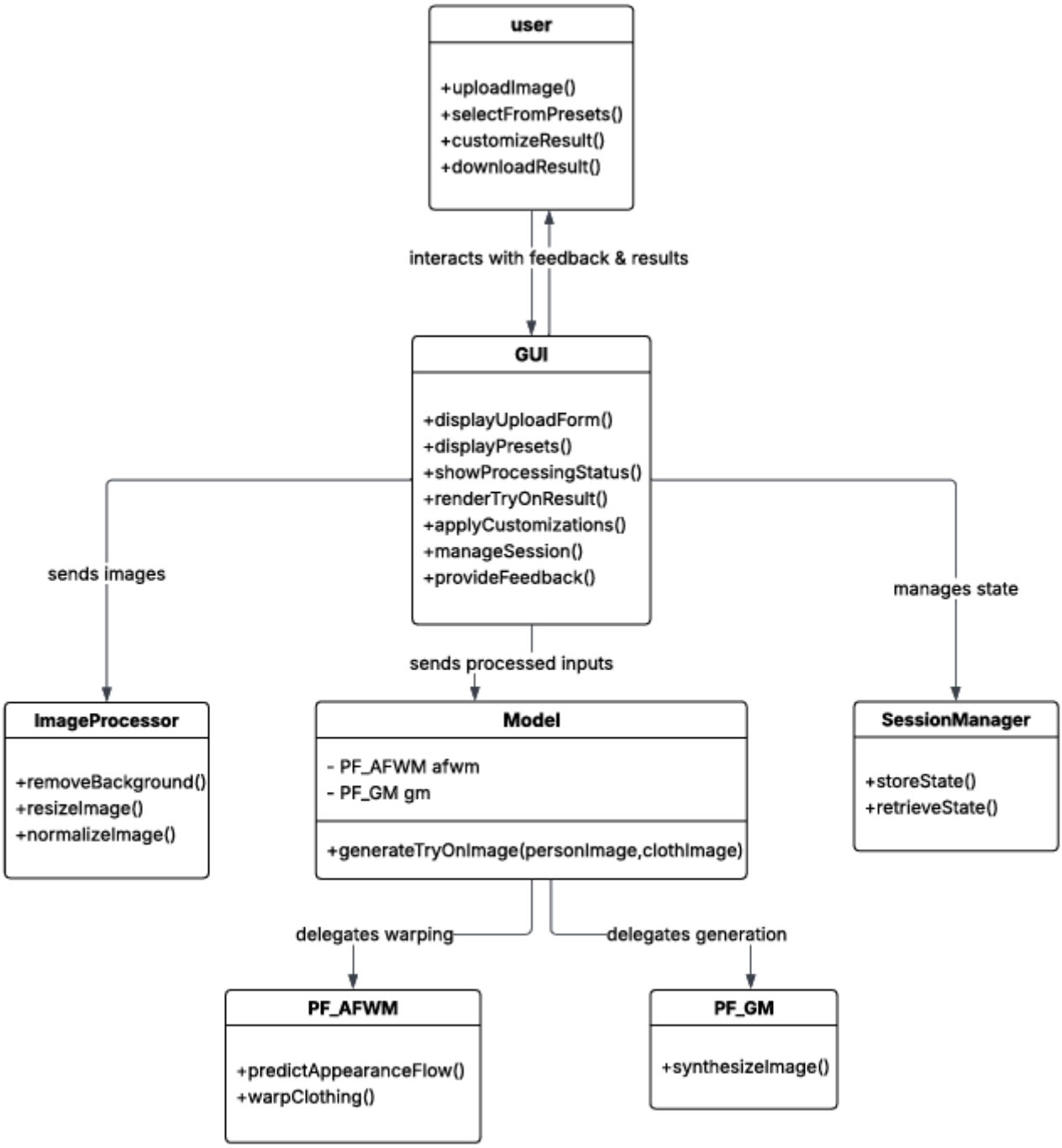
4
SOFTWARE ARCHITECTURE
4.6
PSEUDOCODE

```
// 3. Preprocess Images
  FUNCTION PreprocessImages(personImage, clothingImage)
    REMOVE background from both images
    RESIZE both images to 192x256 pixels
    NORMALIZE both images with white background
    RETURN preprocessedPersonImage, preprocessedClothingImage
  END FUNCTION

  // 4. Visualize Inputs
  FUNCTION VisualizeInputs(preprocessedPersonImage, preprocessedClothingImage)
    DISPLAY preprocessedPersonImage
    DISPLAY preprocessedClothingImage
    PROMPT user to verify images
  END FUNCTION

  // 5. Virtual Try-On
  FUNCTION GenerateTryOn(preprocessedPersonImage, preprocessedClothingImage)
    SEND both images to TryOnModel
    resultImage ← RECEIVE output from TryOnModel

    IF resultImage IS valid THEN
      DISPLAY resultImage
    ELSE
      DISPLAY "Error: Try-On Generation Failed"
    ENDIF
  END FUNCTION

  // 6. Customize Results
  FUNCTION CustomizeResult(resultImage)
    IF user applies grayscale filter THEN
      APPLY grayscale filter
    ENDIF

    IF user adjusts zoom THEN
      APPLY zoom level based on slider
    ENDIF

    IF user clicks reset THEN
      REVERT to original resultImage
    ENDIF

    RETURN customizedResultImage
  END FUNCTION
```

4
SOFTWARE ARCHITECTURE
4.6
PSEUDOCODE

```
// 7. Download Images
   FUNCTION DownloadResultImage(customizedResultImage)
      IF user clicks "Download" THEN
         PROMPT download of customizedResultImage
      ENDIF
   END FUNCTION

   // 8. Manage Session
   FUNCTION ManageSession()
      IF images ARE uploaded OR selected THEN
         STORE images in session_state
      ENDIF

      PERSIST session_state across actions
   END FUNCTION

   // 9. Feedback
   FUNCTION ProvideFeedback()
      FOR each user action (Upload, Select, Generate, Customize, Download)
         VALIDATE required steps
         IF step valid THEN
            DISPLAY "Success" message
         ELSE
            DISPLAY appropriate error or warning
         ENDIF
      ENDFOR
   END FUNCTION

END FUNCTION
```

4
SOFTWARE ARCHITECTURE
4.7
UML CLASS DIAGRAM



Chapter Five

BACKGROUND

5
BACKGROUND

5.1
INTRODUCTION

Early virtual try-on systems, such as CP-VTON, CP-VTON+, ClothFlow, ACGPN, and PF-AFN marked significant milestones in the evolution of garment transfer technology. These models paved the way for image-based clothing try-on by introducing modular architectures and novel alignment strategies. However, as the field advanced, the limitations of these methods became increasingly apparent. This chapter critically evaluates these early approaches by identifying their ineffective components, such as human parsing, local appearance modeling, and GMM-TOM modules, and explains why they were initially adopted, why they failed, and how they have been successfully replaced by more robust, modern alternative like Flow-Style-VTON.

5.2
CP-VTON

CP-VTON (Characteristic-Preserving Image-based Virtual Try-On Network), introduced at ECCV 2018, is a two-stage virtual try-on model that enables a user to "try on" new clothes using an image of a person and a separate product image of a clothing item. It focuses on accurately warping the clothes and generating realistic synthesis of the final try-on result. It is one of the earliest and most cited models in virtual try-on and laid the foundation for many successor models.

5
BACKGROUND

5.2.1

Model Structure (Two-Stage Architecture)

Stage 1: Geometric Matching Module (GMM)

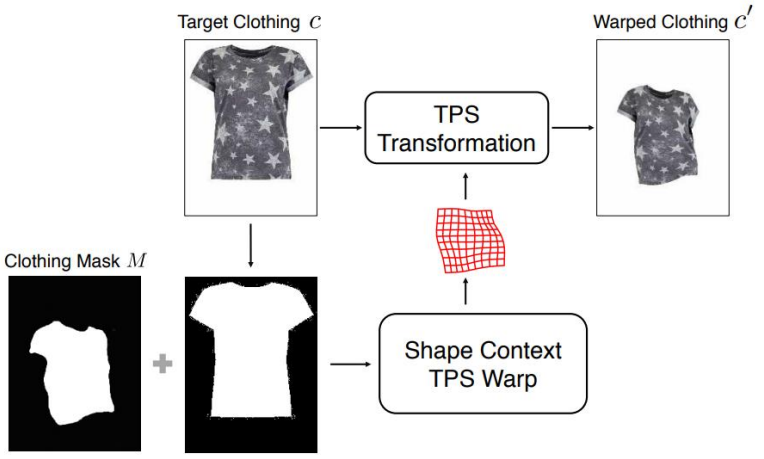
Goal: Align the in-shop clothing image with the target person's body and pose.

Inputs: Person representation (pose heatmap + body shape mask), and in-shop clothing image.

Components: Feature extractor (CNN), feature fusion & regression network, and TPS Transformation Head which predicts Thin Plate Spline parameters (18 control points).

Warping Process: Predict TPS transformation, generate dense sampling grid, then apply bilinear sampling to warp the clothing image.

Output: Warped clothing image **Cwarped**.



5
BACKGROUND

5.2.1

Model Structure (Two-Stage Architecture)

Stage 2: Try-on Module (TOM)

Goal: Blend the warped clothing image with the person image to generate the final try-on result.

Inputs: Warped clothing **Cwarped,** and person representation (pose, and shape).

Components: Encoder-decoder network (U-Net style).

Predicts: rendered image **Irendered**, and composition mask **M** ∈ [0,1].

Training Losses

- ✓ L1 loss between generated and ground-truth image.
- ✓ Perceptual loss (VGG-based).
- ✓ Mask regularization (to encourage realistic blending).

5.2.2

Contribution

Introduction of Geometric Matching Module (GMM): CP-VTON was one of the first virtual try-on models to incorporate a learnable geometric alignment mechanism.

5
BACKGROUND

5.2.2
Contribution

Person Representation with Pose and Parsing Information: CP-VTON introduces a rich person representation which is a combination of the pose keypoints, body segmentation map, and preserved regions (face, hair, body). This guides the try-on synthesis network to focus on the right spatial structure and preserve important identity features.

Benchmarked Evaluation and Dataset: CP-VTON introduced a benchmark dataset and evaluation protocol, helping standardize comparison for future virtual try-on research.

5.2.3
Limitations

Reliance on Human Parsing: Requires accurate segmentation maps and pose estimations. Errors in segmentation (e.g., mislabeling hair, arms, or background) lead to unrealistic results.

Texture and Detail Loss: Even with GMM, fine textures (e.g., logos, embroidery) are not preserved well. Warping can distort detailed patterns, and TOM might generate blurry outputs.

Occlusion and Complex Poses: Fails to realistically handle self-occlusions (arms covering torso, sitting poses). Struggles with in-the-wild images with diverse lighting and backgrounds.

5
BACKGROUND

5.3
CP-VTON+

CP-VTON+ (Clothing Shape and Texture Preserving Image-Based Virtual Try-On**)** is an enhanced version of the original CP-VTON model for virtual try-on. It was proposed to improve the alignment accuracy and visual quality of the synthesized image when trying on a clothing item digitally. CP-VTON+ mainly improves upon the geometric warping process(Geometric Matching Module), which is responsible for warping the in-shop clothing item to align with the target person's pose and body shape, which was a significant limitation in the original CP-VTON.

5.3.1
Model Structure (two main stages)

Improved Geometric Matching Module (GMM+)

Purpose: Warps the in-shop clothing to fit the target person based on their pose and parsing.

Inputs

- ✓ Person representation: pose heatmap, human body mask, face, hair, lower-body clothing.
- ✓ In-shop clothing image.

5
BACKGROUND

5.3.1

Model Structure (two main stages)

Improved Geometric Matching Module (GMM+)

Architecture Improvements

1. **Improved human parsing:**

✓ Fixes parsing errors in the chest/neck area and occlusions by hair.

✓ Adds a skin label to better define body shape.

2. **GMM input modification:**

Uses clothing mask instead of RGB image during geometric matching to focus on shape, not color.

3. **TPS Warping with Regularization:**

Predicts Thin Plate Spline (TPS) parameters to warp the clothing image.

Introduces grid-level warping regularization:

✓ Penalizes large differences between neighboring grid points to reduce distortion.

✓ Regularization loss is applied directly to the deformation grid.

5
BACKGROUND

5.3.1

Model Structure (two main stages)

Improved Try-On Module (TOM+)

Inputs

- ✓ Warped clothing (from GMM).
- ✓ Person representation (pose, face, hair, lower-body, etc.).
- ✓ Clothing mask.

Architecture Enhancements

Adds warped clothing mask to help TOM distinguish between cloth and background.

Uses ground-truth composition mask instead of learned one:

- ✓ Strengthens alpha blending between warped clothing and preserved body parts.
- ✓ Helps avoid blurry or misaligned overlays.

Final synthesis via ResNet-based encoder-decoder.

5
BACKGROUND

5.3.2
Contribution

- ✓ Fixes human parsing errors in CP-VTON.
- ✓ Adds warping regularization to reduce clothing distortion.
- ✓ Uses ground-truth masks for better blending.
- ✓ Better at handling long sleeves, occlusion, and complex textures.

5.3.3
Limitations

- ✓ **Still Relies on Parsing:** Human parsing segmentation maps are used, which can be brittle.
- ✓ **Pose Misalignment:** If pose estimation is noisy, the warped clothing may still misalign.
- ✓ **Fails on Unusual Poses:** Struggles with complex poses or occlusions.
- ✓ **Limited Generalization:** May not work well for garments with novel shapes or styles outside the training set.

5
BACKGROUND

5.4
ClothFlow

ClothFlow is a virtual try-on model consists of three-stage virtual try-on and proposed to improve clothing alignment and realism by modeling appearance flow(a dense displacement field that shows how each pixel of a garment should move to align with the target body). This warps clothing images directly based on human pose and body features, rather than relying on sparse control points like in TPS. It was designed to overcome the limitations of TPS-based models like CP-VTON that struggle with complex poses or non-linear deformations.

5.4.1
Model Structure

Key Goal: Generate a target person image **It** in a new pose **pt**, using a source image **Is** of the same person, while preserving fine-grained appearance and texture details, especially of clothing.

Main Components of the Architecture:

Conditional Layout Generation

Inputs: Source image **Is**, source segmentation map, and target pose **pt**.

Output: Target semantic layout.

Purpose: Generate a structural guide for how the person should look in the target pose (body parts and clothing regions).

5
BACKGROUND

5.4.1
Model Structure

Cascaded Clothing Flow Estimation

Goal: Compute clothing flow – a dense 2D appearance flow field.

Architecture: A cascaded network is used to gradually refine the flow.

Function: Determines which pixels from the source image can be used to warp clothes to match the new pose. This module explicitly handles pose-induced deformation in the clothing region.

Why This Is Better Than Prior Work:

- ✓ More expressive than TPS: It learns a pixel-level flow, not just a few control points.
- ✓ Operates at multiple resolutions, preserving both large-scale and fine-grained alignment.
- ✓ Uses learned, image-aware warping rather than relying on keypoints or hardcoded masks.
- ✓ Handles occlusions and partial clothes better than standard optical flow methods.

5
BACKGROUND

5.4.1
Model Structure

Cascaded Clothing Flow Estimation

How It Works:

I. **Feature Pyramid Networks (FPNs):** Two separate FPNs:

✓ **Source FPN:** Encodes features from the source clothes and segmentation map.

✓ **Target FPN:** Encodes features from the target layout (i.e., the desired output pose/layout).

Both FPNs generate multi-resolution feature maps (pyramids): {S1...SN} and {T1...TN}.

II. **Cascaded Warping**

✓ Starts at lowest resolution (smallest feature maps):

Concatenate **SN** and **TN**, pass through convolution to estimate initial clothing flow **FN**.

✓ Then go up the pyramid:

Upsample flow **FN**, warp source features using it, and refine the flow with residual updates.

This continues from level N down to 1, giving progressively finer flow maps.

5
BACKGROUND

5.4.1
Model Structure

Cascaded Clothing Flow Estimation

How It Works:

III. Warping Clothes

Final flow **F1** is used to warp the actual source clothing image, generating a new image **c′s** that matches the target pose.

Rendering Stage

Inputs: Warped clothes, source image, target pose, and semantic layout.

Output: Final synthesized image **It**.

This stage blends all the warped and contextual information to generate a photo-realistic image with preserved clothing texture.

5.4.2
Contribution

Dense Appearance Flow Estimation

ClothFlow introduces a method to estimate dense appearance flows, capturing pixel-level correspondences between source and target clothing regions. This allows for precise modeling of complex clothing deformations due to pose changes.

5.4.2

Contribution

Cascaded Warping Network

The model employs a cascaded warping network with dual feature pyramid networks (FPNs) for the source and target. This architecture progressively refines the clothing flow estimation from coarse to fine levels, enhancing the alignment of clothing details.

Clothing-Preserving Rendering

In the final stage, ClothFlow uses the warped clothing, along with the source image, target semantic layout, and target pose, to generate the final output image. The rendering process incorporates perceptual and style losses to maintain clothing textures and overall image quality.

5.4.3

Limitations

Still dependent on human parsing: which can be error-prone and affect the final output quality.

Handling of Large Misalignments and Occlusions: struggle with significant pose variations and occlusions, leading to artifacts in the synthesized images.

Local Appearance Flow Estimation: local appearance flow estimation may not effectively capture global context, leading to less coherent results in challenging scenarios.

5
BACKGROUND

5.5
ACGPN

The ACGPN (Adaptive Content Generating and Preserving Network) is an advanced virtual try-on model designed to realistically synthesize images of a person wearing new clothes, especially handling complex poses, occlusions, and detailed clothing preservation better than prior methods like CP-VTON or ClothFlow. It preserves the body pose, the facial identity, and the target clothing's texture and structure. ACGPN introduces an adaptive, multi-stage pipeline that separates structure alignment and content fusion more effectively.

5.5.1
Model Structure

Pipeline Overview: ACGPN consists of three key stages, each handled by a dedicated network:

Stage1: Semantic Alignment

Network: Semantic Generation Module (SGM).

Purpose: Predicts semantic layout of the target image, including body part and warped cloth masks.

Inputs: reference image (person image), corresponding segmentation mask, pose map, and target clothing image.

Outputs:

- ✓ $MS\omega$**:** Synthesized body part masks (head, arms, pants).
- ✓ MSc**:** Synthesized clothing region mask.

5
BACKGROUND

5.5.1
Model Structure

Stage2: Cloth Warping

Network: Clothes Warping Module (CWM).

Purpose: Warps the target clothing image using TPS with second-order constraints.

Inputs: Target clothing image, and clothing region mask from SGM.

Outputs:

- ✓ **TWc**: Warped clothing image.
- ✓ **α**: Learned blending weight map.
- ✓ **TRc**: Refined clothing image (after blending).

Stage3: Try-On Synthesis

Network: Content Fusion Module (CFM).

Purpose: Fuses all semantic, cloth, and body part information into a final image.

Inputs: Refined clothing image (**TRc** from CWM), body part image with clothing region removed, synthesized clothing mask (**MSc** from SGM), and composite mask of preserved body part.

Output:

- ✓ **Itryon**: Final virtual try-on result (person wearing the target clothes).

5
BACKGROUND

5.5.2
Contribution

Semantic Layout Prediction in ACGPN

Semantic layout allows ACGPN to selectively preserve the body parts not affected by the clothing change and synthesize new regions where occlusion or transformation is needed. This adaptive generation-preservation approach improves visual consistency and realism.

Improved Realism with Adaptive Content Preservation

- ✓ Semantic Generation Module (SGM): Guides the next steps to maintain structural consistency.
- ✓ Clothes Warping Module (CWM): The warped result fits more naturally over the person's body and aligns with predicted contours.
- ✓ Content Fusion Module (CFM): Maintains image-level realism and spatial accuracy.

Thanks to these three modules, ACGPN achieves fewer unnatural artifacts, and clear preservation of non-garment features (e.g., face and arms).

5
BACKGROUND

5.5.3
Limitations

Dependency on Human Parsing

The entire pipeline depends on high-quality semantic parsing.

Problem: If the parser fails (e.g., mislabels an arm or misses hair), the final try-on image inherits that flaw.

This makes ACGPN fragile in unconstrained environments.

Handling of Occlusions and Complex Poses

- ✓ ACGPN struggles with self-occlusions, such as folded arms or hands in pockets.
- ✓ Generated results can be blurry, distorted, or anatomically incorrect.
- ✓ The layout sometimes fails to generalize to non-standard poses.

Local Warping Limitations

CWM operates mostly locally, focusing on fine-grained pixel-level alignment.

Problem: When person and clothing images differ significantly in scale or pose, the warping may introduce stretching artifacts, misaligned edges, and unnatural garment draping.

5
BACKGROUND

5.6
PF-AFN

PF-AFN (Parser-Free Appearance Flow Network) is a virtual try-on model that aims to overcome the key limitations found in traditional virtual try-on systems that rely on human parsing maps. In conventional methods such as CP-VTON, ClothFlow and ACGPN the generation of realistic try-on images hinges on accurate segmentation of the human body into discrete parts like arms, legs, torso, and face. These parsing maps guide the placement and deformation of garments, helping align them with the target body pose. However, these systems suffer significantly when the parsing fails, which can occur due to occlusion, unusual poses, or ambiguous boundaries between body parts. Parsing errors often result in misaligned clothes, unrealistic warping, or visual artifacts in the final output.

To address this, PF-AFN introduces a parser-free paradigm, removing the dependency on explicit human segmentation. At the core of PF-AFN is an innovative teacher-tutor-student knowledge distillation framework. Rather than relying on human parsing at inference time, the system leverages a pre-trained teacher model that still uses parsing to guide training. The knowledge distilled from this teacher is then transferred to a student model through an intermediate tutor representation. This allows the student network to learn how to align and warp garments directly from visual data without ever needing parsing inputs during generation.

5
BACKGROUND

5.6
PF-AFN

PF-AFN stands for Parser-Free Appearance Flow Network that means:

- ✓ **Parser-Free:** Indicates that the model does not rely on human parsing (i.e., body part segmentation) during inference, unlike earlier methods such as CP-VTON or ACGPN.
- ✓ **Appearance Flow:** Refers to the learned dense flow fields that guide how pixels from the clothing image are spatially transformed to align with the person.
- ✓ **Network:** Represents the deep learning architecture that combines flow estimation, knowledge distillation, and image synthesis into an end-to-end system.

5.6.1
Model Structure

There are two networks involved:

- ✓ PB-AFN (Parser-Based Appearance Flow Network) — the tutor.
- ✓ PF-AFN (Parser-Free Appearance Flow Network) — the student.

5
BACKGROUND

5.6.1
Model Structure

PB-AFN Training

PB-AFN is trained with human parsing (segmentation) and pose information, which helps it align clothes to the body using detailed guidance.

Inputs: Clothes image, person image (with the same clothes), parsing map (hair, face, lower body), and pose estimation → forms a feature **p***.

Process:

- ✓ The Appearance Flow Warping Module (AFWM) takes **p*** and clothes image to generate appearance flow (**uf**).
- ✓ The clothes image is warped to **uw** using **uf**.
- ✓ A generative module (PB-GM) combines **uw**, preserved regions of the person, and pose to generate the synthetic try-on image.

Try-On Generation for Distillation

After PB-AFN is trained:

- ✓ A new clothes image **Ie_c** is selected.
- ✓ PB-AFN generates a fake try-on image **ue_I** (person wearing new clothes).
- ✓ This try-on image serves as "tutor knowledge" to train the PF-AFN.

5
BACKGROUND

5.6.1
Model Structure

PF-AFN Training (Student Network)

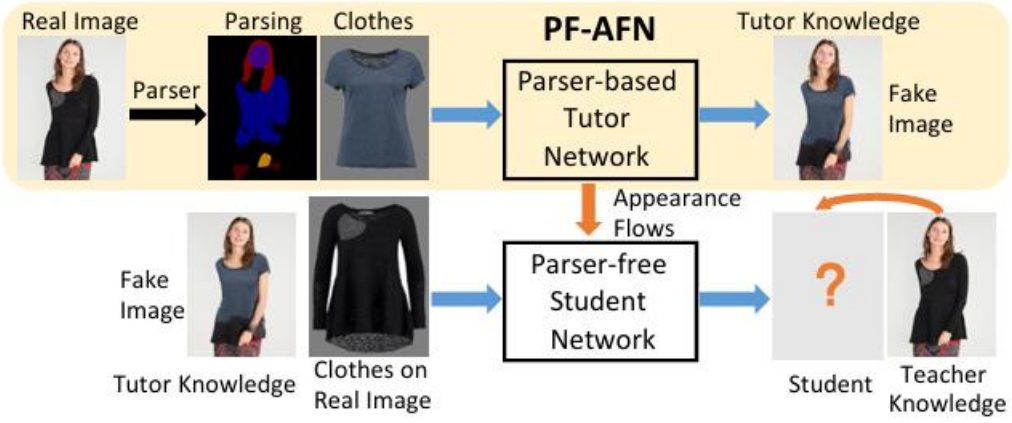
PF-AFN does not rely on human parsing, making it more lightweight and robust.

Inputs: Try-on image **ue_I** (from PB-AFN), and original clothes image **Ic**.

Process:

- ✓ PF-AFWM (same structure as PB-AFWM) predicts flow **sf** between **ue_I** and **Ic**, used to warp **Ic** to **sw**.
- ✓ A generative module (PF-GM) combines **sw** and **ue_I** to produce final result **sI**.
- ✓ This output **sI** is compared against the real image **I** using losses (pixel, perceptual, and smoothness).

Teacher-Tutor-Student Framework



5
BACKGROUND

5.6.1
Model Structure

Knowledge Distillation

The PB-AFN transfers not just images (**ue_I**), but also appearance flows **uf** to PF-AFN. This helps the student network learn better correspondences even without parsing information.

What is Knowledge Distillation?

In general, knowledge distillation is a technique where a simpler or smaller "student" model learns from a larger, more accurate "teacher" model. The teacher's output (e.g., predictions or intermediate representations) provides extra supervision to improve the student’s learning.

Goal:

Distill appearance flows from a strong parser-based model (PB-AFN) into a lightweight parser-free model (PF-AFN) to help the student better match clothing to human images.

However, there's a problem:

The teacher model (PB-AFN) depends on human parsing and densepose estimation, which can be unreliable. If parsing fails, the teacher might produce incorrect supervision, which would harm the student model.

5
BACKGROUND

5.6.1
Model Structure

Knowledge Distillation

Solution: Adjustable Knowledge Distillation

To solve the above problem, the authors introduce adjustable knowledge distillation — a mechanism that selectively enables distillation only when the teacher's output is better than the student’s.

Key Components of Adjustable Distillation

Inputs:

- ✓ PB-AFN (Teacher) gets rich inputs: parsing, pose, densepose.
- ✓ PF-AFN (Student) gets simpler inputs: just the person and clothes images.

Outputs Compared:

- ✓ **uI**: Image generated by PB-AFN
- ✓ **sI**: Image generated by PF-AFN
- ✓ **I**: The real person image

If PB-AFN's result is closer to the real image than PF-AFN’s (i.e., ||uI - I||1 < ||sI - I||1), then its features and flows are considered trustworthy and used for distillation.

5
BACKGROUND

5.6.2
Contribution

Addresses Limitations of Parser-Based Methods

Traditional virtual try-on models like CP-VTON and ACGPN rely heavily on human parsing maps, pixel-level segmentations that label body parts (arms, torso, legs, etc.). These maps are used to guide garment warping and synthesis. While effective under ideal conditions, this dependency has several drawbacks:

Parsing errors: If the human parser incorrectly identifies body parts (e.g., confusing an arm for background), the try-on result may warp clothing improperly or cause artifacts.

Lack of robustness: parsing models often fail on occluded or unusual poses.

Added complexity: parsing requires additional preprocessing steps and models for segmentation. PF-AFN overcomes this by limiting the need for human parsing, thereby reducing reliance on an external component that can introduce fragility into the pipeline.

Example of misaligned regions of parsing:

(a) Warped Clothes on Reference Image | (b) Warped Clothes on Segmentation | (c) Misaligned Regions

5
BACKGROUND

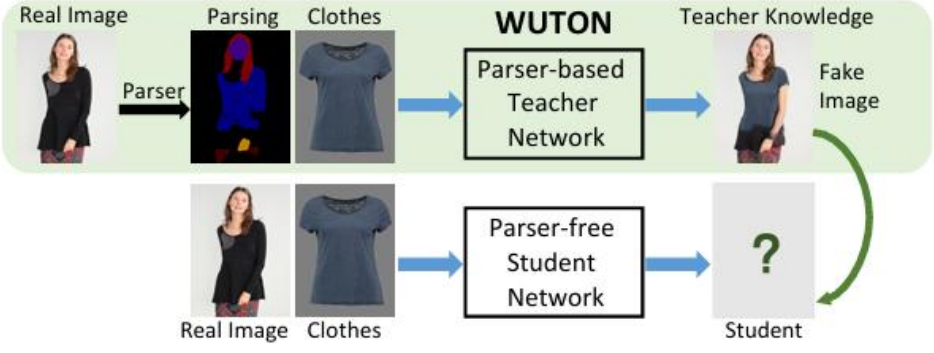
5.6.2
Contribution

Innovative Teacher-Tutor-Student Knowledge Distillation Framework

This approach combines the structure and quality of parser-based methods with the flexibility and robustness of parser-free models. It allows PF-AFN to achieve high-quality synthesis, preserve garment details and human structure, and generalize better to diverse poses and occlusions.

Improves Upon Earlier Parser-Free Models

Earlier parser-free models like **WUTON** treats a parser-based network as a “teacher” network and distills the fake person image produced by the teacher to a parser free “student” network, making the student directly mimic the try on ability of the parser-based teacher. In comparison, PF-AFN treats the fake person image as “tutor knowledge” and uses it as the input of the parser-free “student” network, which is supervised by the real person image (teacher knowledge). The parser-based “tutor” network further distills the appearance flows between the person image and the clothes image, facilitating the high-quality image generation in the “student” network.



5
BACKGROUND

5.6.3
Limitations

Struggles with Extreme Poses and Rich Textures

Despite being parser-free, PF-AFN still struggles in complex scenarios such as:

Extreme body poses: e.g., crouching, stretching, or twisting poses where body parts are occluded or significantly distorted.

Intricate garment textures: e.g., detailed prints, embroidery, transparent fabrics, or complex shading patterns.

These limitations stem from:

- ✓ The local nature of appearance flow estimation, which may fail to model long-range pixel correspondences.
- ✓ The absence of global context modeling, which is critical for reasoning about body layout and garment structure.

To address this, FlowStyle introduces a global style encoder and a flow generator that learns long-range dependencies. This allows it to better warp garments across non-rigid body deformations and produce visually realistic results even in challenging scenarios.

Chapter Six

IMPLEMENTATION ASPECTS

6
IMPLEMENTATION ASPECTS

6.1
Tools, Technologies and Programming Languages used

Our project combines deep learning, image processing, and a web-based interface. Let's break it down into core components:

1. Deep Learning Framework: PyTorch

Used for: Building and training the Flow-Style-VTON model (AFWM + ResUnetGenerator).

Why PyTorch?

- Easy to define custom networks.
- Strong GPU support.
- Widely used in research and production.
- Native support for dynamic computation graphs (important for warping).

Key Components Used:

✓ torch.nn – model architecture (AFWM, ResUnetGenerator).

✓ torchvision.transforms – preprocessing and image handling.

✓ torch.nn.functional.grid_sample – used for applying the appearance flow to warp the garment image.

2. Frontend GUI: Streamlit

Used for: Creating an interactive user interface to upload images, trigger inference, and display results.

6
IMPLEMENTATION ASPECTS

6.1
Tools, Technologies and Programming Languages used

Why Streamlit?

- Python-native.
- No need for HTML/CSS/JS.
- Fast to prototype ML demos.
- Automatically handles file uploads, image previews, layout, etc.

Some Features Used:

- st.file_uploader() – to upload person and garment images.
- st.image() – to preview input and output images.
- st.button() – to trigger model inference.

3. Web Tunneling: ngrok

Used for: Exposing the locally hosted Streamlit app on Google Colab to the public internet.

Why ngrok?

- Colab runs on a remote VM and cannot bind ports directly.
- ngrok creates a secure tunnel from a public URL to your localhost port (usually 8501 for Streamlit).
- Enables demo sharing even without deployment.

6
IMPLEMENTATION ASPECTS

6.1
Tools, Technologies and Programming Languages used

4. Experiment Environment: Google Colab

Used for: Running the model on GPU, hosting the GUI, and prototyping without needing a local machine setup.

Why Colab?

- Free access to GPUs.
- Quick to share notebooks.
- Integrates easily with ngrok and Streamlit for live demos.

5. Image Processing: PIL & OpenCV

Used for:

- Loading and saving images.
- Preprocessing (resizing, cropping, color corrections).
- Applying transformations before sending data into the model.

Libraries:

- Pillow (PIL) – Lightweight and used with PyTorch transforms.
- OpenCV (cv2) – (optional) For more advanced preprocessing or background removal.

6
IMPLEMENTATION ASPECTS

6.1
Tools, Technologies and Programming Languages used

6. Python

Used for: Everything — from model definition to GUI logic.

Why Python?

- Dominant language in AI/ML.
- PyTorch and Streamlit are Python-based.
- Easy to script, extend, and debug.

Role of Python:

- Loads and preprocesses images.
- Runs model inference.
- Handles all GUI logic via Streamlit.
- Integrates Colab, PyTorch, PIL, and ngrok scripts together.

7. Auxiliary Tools & Libraries:

- Torchvision: Pretrained models, transforms, image tools.
- Numpy: Array manipulation, tensor-image conversion.
- matplotlib (Optional): for visualizing flow fields or debugging.
- Rembg: for background removal in images.
- glob, os: For file system handling (batch inference, etc.).

6
IMPLEMENTATION ASPECTS

6.1
Tools, Technologies and Programming Languages used

This stack was chosen for **rapid development**, **accessibility**, and **demonstrability**:

- PyTorch + Streamlit let us go from a research model to a usable product in one notebook.
- ngrok bridges the gap between local and public access.
- Python glues it all together efficiently.

6.2
OVERALL ARCHETICTURE

6.2.1
Workflow

1. Upload or Select Images

Frontend Role:

- User uploads or selects a person image and a clothing image.
- GUI displays selected images and confirms success.
- Backend Trigger: Waits until both images are confirmed before preprocessing.

6
IMPLEMENTATION ASPECTS

6.2.1
Workflow

2. Preprocessing

Frontend View: Shows loading indicator or "Processing images..." message.

Backend Tasks:

- Removes background (if needed).
- Resizes images to 192×256 (VITON format).
- Normalizes on white background.
- Prepares inputs for the model.

3. Visualize Inputs

Frontend:

- Displays person and clothing images side by side.
- Allows user to confirm input before generation.

4. Virtual Try-On (Backend)

Backend Process:

A. **Warping Module (PF-AFWM):** Predicts dense appearance flow between person and cloth image. Warps clothing accordingly.

B. **Generative Module (PF-GM):** Synthesizes a realistic try-on image using warped clothes and the person's features.

C. **Outputs:** Try-on result image (person wearing the selected clothing).

Frontend:

Displays the final try-on image. Shows error messages if generation fails.

6
IMPLEMENTATION ASPECTS

6.2.1
Workflow

5. Customize Results (GUI)

Frontend Tools: Apply enhancements on result such as grayscale filter or zoom using slider.

Backend Role: Applies basic image transformations (locally).

6. Download Image (GUI)

Frontend: "Download" button prompts download of final (possibly customized) image.

Backend: Prepares image blob for download, including modifications.

7. Session Management (GUI + System)

System Maintains State: Uses st.session_state or similar to store:

- Uploaded images
- Current try-on result
- Customization state

8. Real-time Feedback (GUI)

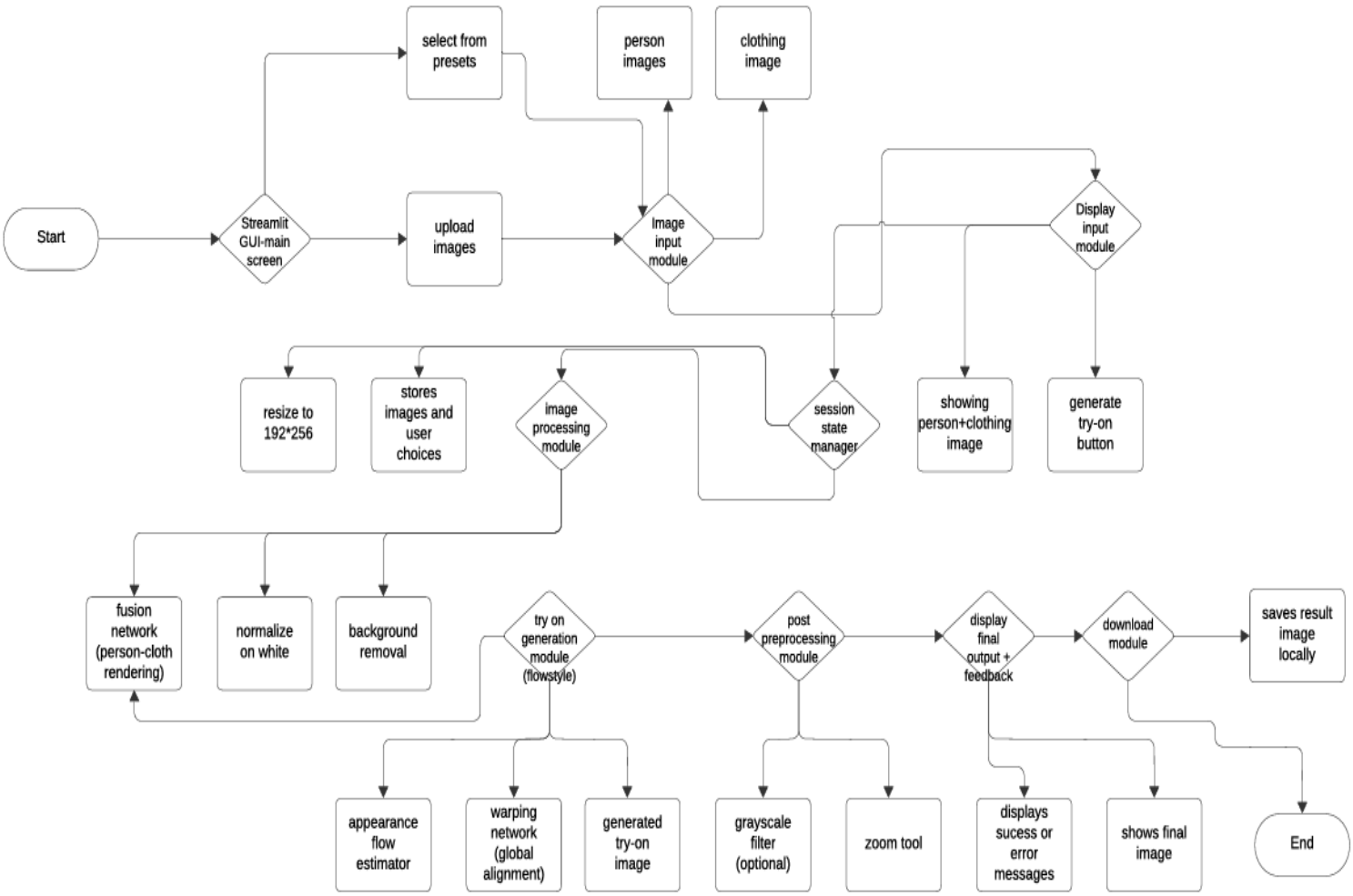
Frontend:

- Provides feedback messages:
- Upload successful
- Generation started/complete
- Error messages

6
IMPLEMENTATION ASPECTS

6.2.2
Block Diagram

Block diagram outlines the workflow and architecture of the overall virtual try-on system using a Streamlit GUI:



6
IMPLEMENTATION ASPECTS

6.3
DATASET

Dataset Components

- **Person Images (person)**

Contains images of people wearing simple or baseline clothing.

Resolution: Typically 256x192 pixels.

Format: .jpg or .png.

- **Clothing Images (cloth)**

Images of the clothing items (e.g., tops, shirts, dresses) to be tried on.

These are generally front-facing images of the clothing item without a person.

Resolution: Same as the person images (256x192).

Format: .jpg or .png.

- **Segmentation Maps (person_segmentation)**

Precomputed segmentation masks for the person images.

Each mask is typically a grayscale image, where different pixel values represent different body parts:

- ✓ 0: Background
- ✓ 1: Hair
- ✓ 2: Face
- ✓ 3: Upper body
- ✓ 4: Arms
- ✓ 5: Legs Format: .png or .mat files.

6
IMPLEMENTATION ASPECTS

6.3
DATASET

Dataset Components

- **Pose Landmarks (pose)**

JSON or .txt files containing the coordinates of key body landmarks (e.g., shoulders, elbows, wrists, hips, knees).

Extracted using tools like OpenPose.

- **Warped Cloths (Optional Intermediate Dataset) (warped_cloth)**
- **Annotations (pair.txt or train_pairs.txt)**

A file listing the correspondence between person images and clothing items.

6
IMPLEMENTATION ASPECTS

6.3
DATASET

```
dataset/
├── train/
│   ├── person/              # Person images for training
│   │   ├── 0001.jpg
│   │   ├── 0002.jpg
│   │   └── ...
│   ├── cloth/               # Clothing images for training
│   │   ├── cloth_0001.jpg
│   │   ├── cloth_0002.jpg
│   │   └── ...
│   ├── person_segmentation/   # Segmentation maps for persons
│   │   ├── 0001.png
│   │   ├── 0002.png
│   │   └── ...
│   ├── pose/                # Pose landmarks (JSON or .txt files)
│   │   ├── 0001.json
│   │   ├── 0002.json
│   │   └── ...
│   ├── train_pairs.txt       # Pairs of person and cloth
│   └── warped_cloth/         # Warped cloths (optional)
│       ├── 0001_warped.jpg
│       ├── 0002_warped.jpg
│       └── ...
└── test/
    ├── person/              # Person images for testing
    ├── cloth/               # Clothing images for testing
    ├── person_segmentation/   # Segmentation maps for testing
    ├── pose/                # Pose landmarks for testing
    └── test_pairs.txt        # Pairs of person and cloth for testing
```

6
IMPLEMENTATION ASPECTS

6.4
SAMPLE OUTPUT

Person Image

Clothes Image

VTON result Image

Chapter Seven

APPLIED MODEL AND ENHANCEMENTS

7

APPLIED MODEL AND ENHANCEMENTS

7.1

INTRODUCTION

In our project, we present a novel approach for virtual try-on (VTON), a task that aims to realistically visualize how an in-shop clothing item would look when worn by a person in a given photo. Traditional methods typically rely on local garment warping techniques, which struggle with large pose variations, occlusions, and spatial misalignments between the person and the garment. To overcome these limitations, our project utilizes a StyleGAN-based architecture to estimate a global appearance flow. This global flow allows the model to effectively capture the overall structure and context of both the person and the garment, enabling more accurate and visually coherent garment alignment, even in challenging scenarios involving complex body poses or significant differences in scale and position. By incorporating both global context and local refinement, our method ensures that the try-on result not only aligns well but also preserves the fine-grained details of the clothing.

7.2

KEY FEATURES OF OUR MODEL

Parser-Free Design

Our model eliminates the need for human parsing during inference, making it more robust and easier to deploy in real-world applications.

7

APPLIED MODEL AND ENHANCEMENTS

7.2

KEY FEATURES OF OUR MODEL

Global Appearance Flow Estimation

By using a StyleGAN-based architecture, the model captures global context, allowing for accurate garment alignment even with large pose differences or misalignments

Local Flow Refinement

Fine-grained alignment is achieved through a refinement layer that corrects local deformation, preserving garment details like sleeves and textures.

Robust to Occlusions and Poses

The combination of global and local modeling allows the system to handle complex body poses, occluded regions, and diverse clothing styles effectively.

7
APPLIED MODEL AND ENHANCEMENTS

Why we chose this architecture

We selected the style-based global appearance flow architecture for its ability to overcome key limitations found in previous virtual try-on methods. Traditional models relying on local warping struggle with misalignment and fail to generalize to natural, in-the-wild images. The StyleGAN framework, however, provides a powerful way to capture global structure through a style vector while still supporting detailed, fine-grained refinement through local correspondence. This hybrid design allows our model to deliver both accuracy in garment placement and high visual quality, making it particularly well-suited for real-world virtual try-on applications.

7.3
MODEL COMPONENTS AND KEY CONCEPTS

What Is StyleGAN

StyleGAN (Style-Based Generative Adversarial Network) is an advanced generative model introduced by NVIDIA, designed to synthesize high-resolution and photorealistic images. Unlike conventional GAN architectures, which map a latent noise vector directly to an output image, StyleGAN separates the learning of image structure and visual attributes by incorporating an intermediate latent space and a style modulation mechanism.

7

APPLIED MODEL AND ENHANCEMENTS

7.3

MODEL COMPONENTS AND KEY CONCEPTS

What Is StyleGAN

This architecture enables multi-scale control over the generation process, allowing distinct levels of the network to influence coarse features such as pose or shape and fine details such as texture or color independently. The modulation is achieved by injecting style vectors into each convolutional layer, thereby enhancing both the diversity and controllability of the generated images. Due to its ability to produce highly realistic and structurally coherent outputs, StyleGAN has become a foundational model in various image generation tasks, including virtual try-on systems, where capturing detailed appearance and accurate spatial alignment is essential.

Role of StyleGAN in the model

In our virtual try-on architecture, StyleGAN is employed as the core mechanism for global appearance flow estimation, which is responsible for aligning the in-shop garment with the person image. Traditional methods rely on local feature correspondences, which often fail when there are large misalignments, pose variations, or occlusions. By contrast, StyleGAN enables the model to learn a global style vector that captures high-level structural relationships between the garment and the body.

7
APPLIED MODEL AND ENHANCEMENTS

7.3
MODEL COMPONENTS AND KEY CONCEPTS

Role of StyleGAN in the model

This style vector is used to modulate feature maps during the warping process, allowing the model to estimate how the garment should deform to fit the target body shape and pose. The use of style modulation ensures that the appearance flow has a global receptive field, making it more robust in complex scenarios where local methods struggle. Thus, StyleGAN is crucial in our approach for achieving accurate garment alignment and maintaining visual coherence in the final try-on output.

What is global appearance flow

Global appearance flow is a technique in computer vision that refers to the estimation of a dense, pixel-wise flow field representing how the visual content in a source image should be spatially transformed or warped to match the structure of a target image. Unlike local appearance flow, which focuses on small, localized regions and uses limited receptive fields, global appearance flow incorporates context from the entire image to determine more coherent and large-scale transformations.

7
APPLIED MODEL AND ENHANCEMENTS

7.3
MODEL COMPONENTS AND KEY CONCEPTS

What is global appearance flow

This approach is particularly beneficial in scenarios where structural differences between source and target images are significant, such as large pose variations, viewpoint changes, or object deformations. By leveraging global context, the model can infer more meaningful and consistent correspondences across the image, leading to improved accuracy in tasks such as image alignment, view synthesis, and motion transfer.

Role of global appearance flow in the model

Global appearance flow estimation serves as the foundation for spatially transforming the garment image to match the body structure and pose of the target person. Instead of relying on isolated, local feature correspondences, our model predicts a full-resolution flow field that captures how each part of the garment should be repositioned across the entire image. This global mapping enables the model to understand long-range dependencies and structural relationships, such as how the shoulder area of a shirt should align with a tilted torso or how a sleeve should follow a bent arm. By estimating the deformation at a global level, the model ensures that the warped garment maintains anatomical correctness and visual continuity, which is critical for generating believable try-on results.

7
APPLIED MODEL AND ENHANCEMENTS

7.3
MODEL COMPONENTS AND KEY CONCEPTS

Role of global appearance flow in the model

This holistic approach reduces artifacts caused by incorrect local matches and enables better handling of complex body poses and garment shapes.

What is local refinement

Local refinement is a technique used in deep learning and computer vision to enhance the precision of an initially coarse prediction by focusing on fine-grained details at a local level. Local refinement solves many problems by operating within smaller spatial neighborhoods to correct minor errors, enhance texture consistency, and improve alignment or resolution. It typically involves processing high-resolution feature maps or applying convolutional layers to refine specific regions of interest. This approach is widely used in tasks such as image synthesis, object detection, and pose estimation, where maintaining both global coherence and local accuracy is essential for producing high-quality results.

7
APPLIED MODEL AND ENHANCEMENTS

7.3
MODEL COMPONENTS AND KEY CONCEPTS

Role of local refinement in the model

In our virtual try-on model, local refinement is employed to enhance the precision of the garment warping process after the initial global appearance flow has been estimated. While the global flow provides a coarse alignment of the garment to the person's body, it may lack the detail necessary to accurately capture small and complex regions, such as sleeves, collars, or hands. To address this, local refinement modules operate at multiple resolutions to fine-tune the appearance flow by leveraging local feature correspondences between the person and garment images. This allows the model to correct subtle misalignments and preserve fine-grained garment details, ensuring that the final try-on image appears natural, anatomically consistent, and visually accurate. By combining global context with localized corrections, the model achieves a more realistic and detailed virtual try-on result.

7

APPLIED MODEL AND ENHANCEMENTS

7.4

MODEL OVERVIEW

Use of global and local refinement

Interaction between global and local refinement

Our virtual try-on model is designed to digitally dress a person in a garment by intelligently modifying the garment image so it fits naturally on the target person's body. To achieve realistic and well-aligned results, the model first estimates a global deformation—a broad transformation that aligns the general shape and position of the garment with the person's pose and body structure. This step ensures that the garment covers the right body areas. However, because global alignment can miss small details, the model includes a local refinement step that corrects finer regions like sleeves, cuffs, or overlapping limbs. By combining global understanding with local precision, the model produces visually coherent and naturally fitting try-on images.

7
APPLIED MODEL AND ENHANCEMENTS

7.4
MODEL OVERVIEW

Use of global and local refinement

Benefits of using this approach

The effectiveness of combining global and local refinement lies in the way each component addresses a different level of spatial complexity within the try-on process. Global estimation provides the model with a holistic view, enabling it to account for large-scale transformations, such as adjusting the garment to fit different body orientations or correcting overall positioning. However, real-world images often contain subtle variations and fine structural elements that require more precise treatment. This is where local refinement becomes essential—it introduces spatial sensitivity by focusing on small, context-specific regions that global processing may overlook. Rather than treating garment warping as a single, uniform transformation, this layered approach allows the model to adaptively refine specific areas based on localized features. The result is a system that balances macro-level alignment with micro-level precision, significantly improving visual accuracy and making the final output appear more lifelike and tailored to the individual's pose and body shape.

7
APPLIED MODEL AND ENHANCEMENTS

7.4
MODEL OVERVIEW

Teacher student training strategy

To develop a robust and efficient parser-free virtual try-on model, we adopt a two-stage training strategy that involves a teacher model and a student model. This approach allows the student model to learn accurate garment alignment and image synthesis without relying on external parsing information at inference time.

Step 1: Train the parser-based Teacher model

The first step involves training a teacher model that utilizes rich semantic information, including human parsing maps, keypoint poses, and dense pose data. This model has access to detailed body structure annotations and is capable of learning precise spatial relationships between the person and garment images. By using these high-level inputs, the teacher model can generate high-quality try-on results and intermediate feature representations that reflect strong structural understanding. Although this model depends on parsing data, it is only used during the training phase to guide the learning process.

7

APPLIED MODEL AND ENHANCEMENTS

7.4

MODEL OVERVIEW

Teacher student training strategy

Step 2: Train the parser-free student model with knowledge distillation

Once the teacher model is trained, the next step is to train a parser-free student model that operates using only raw person and garment images. During this phase, the student does not receive semantic inputs; instead, it learns by mimicking the internal feature representations and output behavior of the teacher model. This is achieved through knowledge distillation, where the student's encoder is guided to reproduce the intermediate features generated by the teacher. As a result, the student model learns to perform accurate garment warping and try-on synthesis without needing parsing data. This enables the final model to be more lightweight, efficient, and robust, particularly in real-world scenarios where semantic annotations may be unavailable or unreliable.

7
APPLIED MODEL AND ENHANCEMENTS

7.4
MODEL OVERVIEW

Teacher student training strategy

Person image

Garment image

Human parsing map

Pose Keypoints

Teacher Model

Inputs for parser-based training network: Person image, Garment image, Human parsing map, pose Keypoints, and dense pose.

Outputs for parser-based training network: Synthesized Try on image

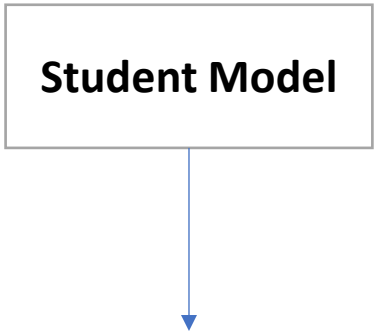
7
APPLIED MODEL AND ENHANCEMENTS

7.4
MODEL OVERVIEW

Teacher student training strategy

Person image

Garment image



Inputs for parser-free student network: Person image, Garment image.

Outputs for parser-free student network: Final Try on image

The output image of the parser-based training network is the input of the student parser free model in the training phase. The student model only is used during inference to generate final try on image.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Problem Formulation

The virtual try-on task in our model is formulated as a conditional image synthesis problem. The input consists of a person image $p \in R^{3\times H\times W}$ and a corresponding in-shop garment image $g \in R^{3\times H\times W}$, where H and W denote the height and width of the images, and the number 3 represents the RGB color channels. The objective is to generate a try-on image $t \in R^{3\times H\times W}$, in which the person from image p is realistically visualized wearing the garment from image g. The generated image t must satisfy two critical requirements: it should preserve the identity and body structure of the person in the original image p, ensuring that facial features, pose, and unaltered body parts remain consistent; and it should retain the visual characteristics of the garment, including texture, color, shape, and design details. Achieving this requires accurately warping the garment to fit the person's body and seamlessly blending it with the original image while maintaining high visual realism.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Feature Extraction

Feature extraction is a critical component in our virtual try-on model, responsible for encoding the visual and structural information from both the person image and the garment image. This process transforms raw input images into multi-scale feature representations that serve as the foundation for subsequent warping and synthesis operations.

Dual Encoders for person and Garment images

Our architecture employs two separate convolutional encoders:

• Ep for the person image p

• Eg for the garment image g

Each encoder processes its respective image and outputs a set of feature maps at multiple spatial resolutions. These feature maps capture information at different levels of abstraction, from low-level texture and edges to high-level semantic structures such as pose and garment shape.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

ResNet- based blocks

Both the person and garment encoders in our model are built upon a shared architectural foundation based on ResNet-style residual blocks, a widely adopted design in deep convolutional networks. The key innovation of a residual block lies in its shortcut connection, also known as a skip connection, which allows the input of the block to bypass one or more convolutional layers and be directly added to the output. This architectural feature addresses a common challenge in deep networks—the vanishing gradient problem—by ensuring that gradients can flow more easily during backpropagation. As a result, residual blocks facilitate the training of much deeper models without degradation in performance, leading to more effective learning of hierarchical visual features. In each encoder, the input image passes through a stacked sequence of residual blocks, where each block contains multiple convolutional layers, typically followed by batch normalization and a nonlinear activation function. As the data moves deeper into the network, each residual block performs a combination of spatial downsampling and channel expansion.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

ResNet- based blocks

As the input image passes through these blocks, the network extracts increasingly abstract and semantically rich features , this enables the network to gradually transition from capturing low-level visual cues, such as edges and textures, to more abstract and semantically meaningful representations, such as body pose, garment shape, and structural alignment. Importantly, because both the person encoder and garment encoder share the same ResNet-based design, the extracted features at each level are structurally aligned and compatible. This architectural consistency facilitates more effective feature comparison, fusion, and warping in the later stages of the model, particularly in the appearance flow estimation and garment warping processes. The hierarchical nature of the residual block structure ensures that the network captures information at multiple scales, which is crucial for combining both global structural context and fine-grained local details during the try-on synthesis process.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Multi-Level Feature Extraction

At each stage of the encoder, we extract intermediate feature maps from both the person and garment images. These are denoted as:

• {p1,p2,...,pN}: Feature maps from the person encoder

• {g1,g2,...,gN}: Feature maps from the garment encoder

Here, N represents the total number of feature levels , corresponding to different spatial resolutions. The early layers retain fine-grained details with higher resolution but limited semantic abstraction, while the deeper layers contain low-resolution but semantically richer representations.

Each feature map pi or gi is a 3D tensor of shape Ci×Hi×Wi, where:

• Ci: number of feature channels at level i

• Hi, Wi height and width of the feature map at level i

These multi-level features are essential for the model to:

• Understand the global structure via deeper layers

• Preserve local garment texture and alignment via shallower layers

7
APPLIED MODEL AND ENHANCEMENTS

7.5
MODEL ARCHITECTURE AND WORKFLOW

Multi-Level Feature Extraction

The hierarchical structure of these multi-level feature maps is not only important for representing information at different resolutions, but also plays a key role in enabling the model to reason across spatial scales. In our model, these features are used at different stages for different purposes. The deeper features , which contain rich semantic information, are particularly useful for generating the global style vector. This vector informs the model about how the garment should be warped globally to fit the person's body structure and pose. On the other hand, the shallower feature maps retain higher-resolution texture information and fine-grained appearance details. These are essential for the local refinement module, which performs pixel-level adjustments to preserve the visual fidelity of garment regions like sleeves, patterns, and edges. By maintaining a consistent correspondence between person and garment feature maps at each level, the model can perform layer-wise warping. Specifically, at each stage of the warping process, the garment feature map gi is warped according to a predicted flow field and then aligned with the corresponding person feature map pi.

7
APPLIED MODEL AND ENHANCEMENTS

7.5
MODEL ARCHITECTURE AND WORKFLOW

Multi-Level Feature Extraction

This allows the model to combine semantic alignment and appearance matching progressively, from coarse-to-fine. Moreover, using multiple levels of features allows the network to better handle challenges such as occlusion, scale variation, and pose differences by dynamically focusing on appropriate spatial contexts. For instance, deeper layers help manage large deformations across body parts, while shallower layers ensure continuity and accuracy in smaller regions. This multi-level feature extraction strategy is fundamental to achieving both global structural alignment and local texture consistency, making it a core strength of our virtual try-on architecture.

Purpose of using this architecture

The multi-level features extracted by the person and garment encoders serve as foundational inputs for both deformation and synthesis tasks within the model. High-level semantic features are utilized to encode overall spatial context, enabling the system to compute a deformation strategy that accounts for the person's body layout and garment structure. Simultaneously, lower-level features are preserved to enhance the visual fidelity of garment textures and edges, supporting a coherent and realistic try-on experience across all spatial scales.

7
APPLIED MODEL AND ENHANCEMENTS

7.5
MODEL ARCHITECTURE AND WORKFLOW

Style based global appearance flow module

The Style-Based Global Appearance Flow Module is the core component in our architecture responsible for aligning the in-shop garment with the person image in a globally coherent way. Unlike traditional local warping methods, which operate based only on nearby pixels or features, this module leverages global context to estimate how the garment should deform across the entire image space. This is achieved using a global style vector and a stack of style modulation blocks, enabling the model to perform coarse, structure-aware alignment as a foundation for the final try-on result.

How does style based global appearance flow module work

Step 1: Global Style Vector

The process begins with the computation of a global style vector, which plays a central role in capturing the overall spatial relationship between the person and the garment.

Input: The deepest feature maps from the encoders, pN and gN are used.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Style based global appearance flow module

How it's computed:

- These feature maps are first passed through fully connected layers.
- The resulting vectors are concatenated or fused to produce a single style vector s∈Rd, where d is the dimensionality of the style space.

Purpose: This vector contains high-level contextual information that describes the alignment between body shape, pose, and garment structure. It is not spatially distributed like a feature map but instead acts as a global control signal for the next stage.

Step 2: Style Modulation Blocks (Warping Blocks)

After computing the global style vector, the model uses a sequence of Style Modulation Blocks, also referred to as Warping Blocks, to iteratively align the garment features with the person features. This alignment is done at multiple spatial scales, starting from deep, low-resolution features to shallow, high-resolution ones. Each block consists of two main stages: style-guided coarse warping and local refinement, followed by progressive integration across layers.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Style based global appearance flow module

a) Style-Based Modulated Convolution (Coarse Warping)

At each level i, the model focuses on the corresponding garment feature map g_{N+1-i}, starting from the deepest feature level and moving toward the shallower ones. Here's how coarse warping is performed:

Upsampling:

The feature map f_{i-1} from the previous warping block or an initial zero tensor for the first block is upsampled to match the resolution of g_{N+1-i}. This ensures consistent spatial dimensions for fusion.

Concatenation and Modulated Convolution:

The upsampled flow features and current garment feature map g_{N+1-i} are concatenated and passed through a modulated convolution layer.

- The convolution weights are modulated using the global style vector s, which encodes the overall garment-person relationship.
- This modulation allows the convolution to adapt its behavior dynamically, enabling the network to apply context-aware filters that respond to the spatial structure of the person’s pose.

7
APPLIED MODEL AND ENHANCEMENTS

7.5
MODEL ARCHITECTURE AND WORKFLOW

Style based global appearance flow module

Output:

This operation produces a coarse appearance flow field fc,i, which describes how pixels in the garment feature map should be displaced to align with the target body pose at a global level. However, since this estimation is based on coarse, high-level information, it lacks precision in local regions.

(b) Local Refinement (Flow Residual Correction)

To enhance the accuracy of the coarse flow, the model applies a local refinement step that fine-tunes the appearance flow using high-resolution, spatially detailed features:

Warping with Coarse Flow:

The garment feature map gN+1−i is first warped using the coarse flow fc,i. This provides an intermediate aligned garment feature.

Feature Fusion:

The warped garment feature and the corresponding person feature map pN+1−I are concatenated along the channel dimension. This combination captures both the current alignment and the target structure.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Style based global appearance flow module

Refinement Convolution:

The concatenated features are passed through a regular convolutional block to produce a flow residual fri. This residual flow corrects any local misalignments or artifacts introduced by the coarse warping.

Final Flow Calculation:

The final appearance flow at this level is computed by:

fi=fci+fri, This flow combines both global context and local detail, resulting in a more accurate and smooth warping field.

Final Garment Warping:

The final flow fi is used to warp the original garment feature map gN+1−i, producing the aligned garment feature at this level.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Style based global appearance flow module

(c) Progressive Warping Across Multiple Levels

This entire process—coarse warping followed by local refinement—is repeated over N levels, starting from the deepest features and moving toward the shallowest features.

At each level, the model:

- Incorporates more spatial detail
- Improves alignment accuracy
- Preserves fine garment features such as sleeves, edges, and textures

The flow from each level is incrementally upsampled and refined in the next level, creating a progressive deformation path from global structure to pixel-level alignment.This progressive warping strategy allows the model to synthesize try-on images that are not only aligned at a structural level but also exhibit high-fidelity garment detail and natural transitions.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Style based global appearance flow module

Summary of the module

- The global style vector provides semantic guidance based on the overall image context.
- The modulated convolutions enable the model to apply pose-aware and structure-sensitive transformations.
- The multi-level flow prediction and refinement strategy ensures that warping is both globally aligned and locally accurate.
- The result is a set of warped garment features that are spatially aligned with the person image and ready for final try-on synthesis.

Purpose of using style based global appearance flow module

This module enables the model to compute spatial transformations that consider the full-body layout rather than relying on isolated regions. By integrating a style representation extracted from high-level features, it guides the warping of garment features in a way that reflects the overall geometry of the person. Its role is to provide a strong structural foundation for garment alignment before any fine adjustments are applied, improving robustness in complex cases such as complex poses or significant spatial shifts.

7

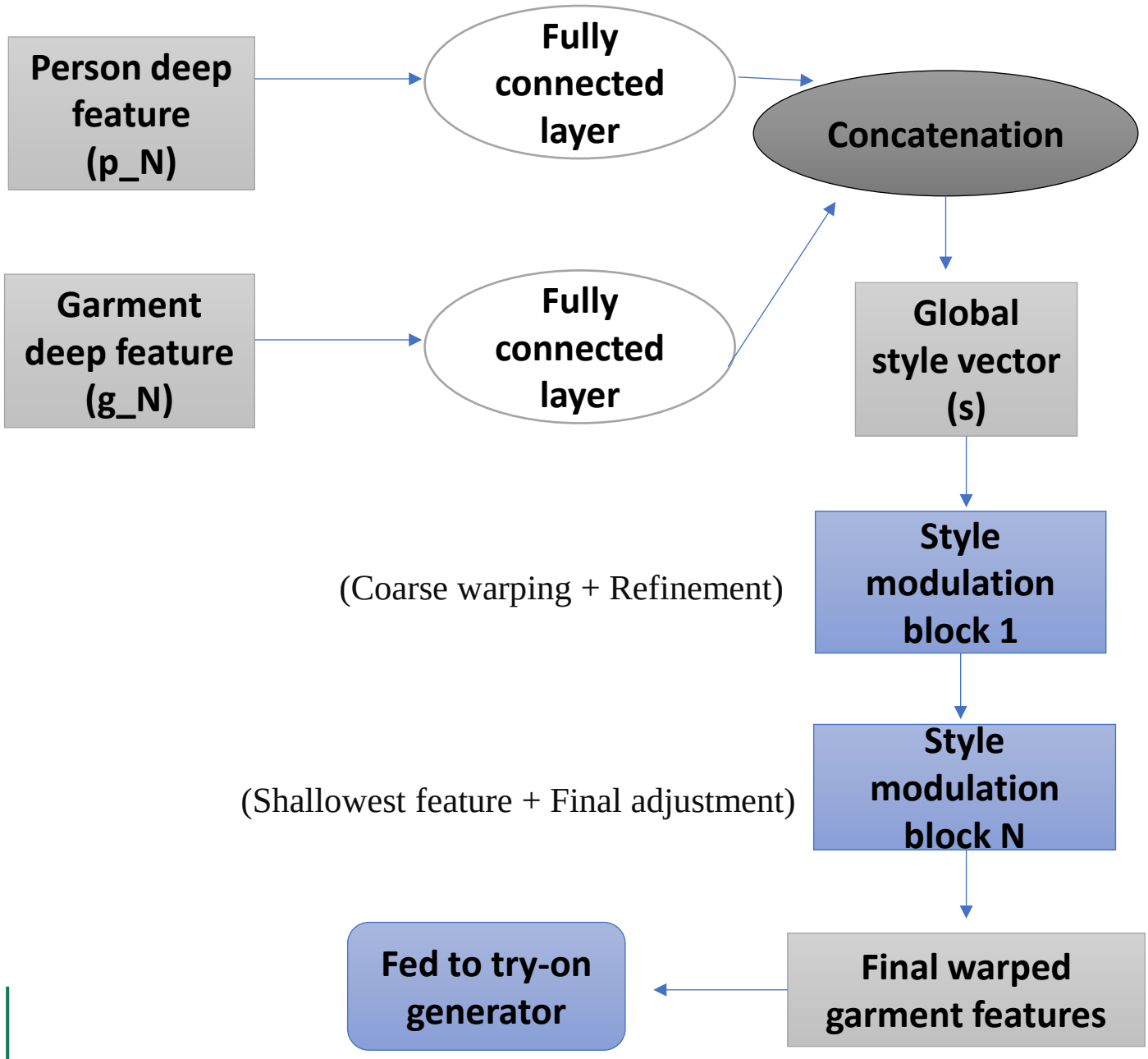
APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Style based global appearance flow module

Style based global appearance flow module diagram



7
APPLIED MODEL AND ENHANCEMENTS
7.5
MODEL ARCHITECTURE AND WORKFLOW
Garment warping and try-on image generation

After the appearance flow has been progressively refined through multiple warping stages, each incorporating both global structure and local detail,the model produces a final, high-resolution appearance flow field, denoted as fN. This flow field is a dense spatial map where each vector indicates how a specific pixel in the original garment image should be repositioned or moved to match the geometry of the target person. Essentially, fN encodes a transformation that deforms the garment in a way that aligns it with the pose, size, and body contours of the person in the reference image. Importantly, fN operates at the pixel level, meaning it can produce fine-grained, non-rigid deformations that are crucial for realistic virtual try-on, where garments must naturally wrap around various body parts. At this point in the pipeline, the model transitions from feature space where garment and person representations are manipulated as abstract tensors to image space, where visual information is directly warped. The flow field fN is applied to the original garment image using a differentiable sampling function often implemented as bilinear sampling to generate a spatially transformed garment image g^. This image reflects the precise pixel movements needed to adapt the flat or misaligned garment image to the body of the person while retaining the garment's original texture, design, and appearance. This warped garment is now ready for fusion with the person image in the final synthesis stage.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Garment warping and try-on image generation

How does garment warping and try on image generation work

Step 1: Final Garment Warping

The final appearance flow fN, obtained after passing through all style modulation and refinement blocks, represents a dense pixel-wise deformation field. Each vector in this flow field indicates how a particular pixel in the garment image g should be shifted to align with the person's body. This transformation is not rigid; it varies at every pixel, allowing the network to perform non-linear, fine-grained warping that accounts for body pose and clothing deformation. To apply this transformation, the model uses a sampling function, denoted as S(g,fN), to generate the final warped garment image g^. This function is responsible for mapping each pixel location in the target image space back to the corresponding location in the original garment image, based on the displacement vectors defined in fN.

What is Sampling Function S:

The sampling function is a differentiable spatial transformer, typically implemented using bilinear interpolation.

7
APPLIED MODEL AND ENHANCEMENTS
7.5
MODEL ARCHITECTURE AND WORKFLOW
Garment warping and try-on image generation

What is Sampling Function S:

How it works:

• For each output pixel location (x′,y′), the function computes a source coordinate (x,y) by subtracting the flow vector at that location.

• The pixel value at (x′,y′) is then estimated by interpolating between the nearby pixel values in the original image g, using bilinear weighting.

• This allows the warping process to remain smooth and differentiable, which is crucial for end-to-end training via backpropagation.

Mathematically:

g^(x′,y′)=∑i,j g(i,j)·wij(x,y) where wij are the bilinear interpolation weights based on proximity to the non-integer coordinate (x,y).

Purpose and importance of this step:

This warping step plays a critical role in the entire try-on pipeline:

• It ensures that the garment contour, size, and shape are properly adjusted to match the person’s pose.

• It helps maintain the original appearance of the clothing such as fabric patterns, colors, and textures while adjusting its geometry.

7
APPLIED MODEL AND ENHANCEMENTS
7.5
MODEL ARCHITECTURE AND WORKFLOW
Garment warping and try-on image generation

Purpose and importance of this step:

- It bridges the gap between feature space alignment and image space synthesis, producing a visually aligned garment ready to be blended into the final try-on image.

Unlike feature-level deformation, this step directly manipulates visual pixels, meaning any errors here can visibly affect realism. Accurate flow and smooth sampling are therefore essential for producing seamless, artifact-free try-on results.

Step 2: Concatenation with Person Image

After generating the warped garment image g^ using the final appearance flow fN, the next step involves preparing the input for the image synthesis stage. To do this, the model concatenates the warped garment g^ with the original person image p, forming a composite input that combines both visual sources of information. This concatenation is typically done along the channel dimension, meaning the RGB channels of both images are stacked to produce a unified tensor.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Garment warping and try-on image generation

What is concatenation in our case:

Concatenation is a common operation in deep learning where two tensors are joined along a specified axis. In this case, concatenation along the channel axis allows the model to retain independent color and feature information from both the person and the garment, while processing them as a single input during generation. Concatenation is different from addition which fuses values directly; it preserves full information from both sources so that the network can learn how to best blend them later.

Purpose and importance of this step:

This step plays a crucial role in ensuring that the generator has access to all the necessary context to produce a high-quality try-on result.

It provides the following:

- Person Identity and Body Context:

  The original person image contains important information such as facial features, skin tone, hairstyle, body pose, and other personal attributes that should be preserved in the final try-on image.
- Aligned Garment Appearance:

  The warped garment image brings the new clothing into the scene, already spatially aligned with the person’s body

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Garment warping and try-on image generation

Purpose and importance of this step:

By feeding both together, the model allows the generator to understand the interaction between the body and the garment such as where the garment should end, how it should curve around arms or shoulders, and how it should blend into the background naturally. This input design encourages the generator to focus not just on garment overlay but also on seamless integration, so that clothing appears naturally worn, not simply pasted onto the person. It enables the generator to reconstruct occluded or partially hidden regions such as under the arms or behind the neck more realistically, using cues from both inputs.

Step 3: Try-On Image Generation via Generator G

After the person image p and the warped garment image g^ are concatenated into a composite input, the final step in the pipeline is to generate the realistic try-on image t. This is achieved using a deep convolutional neural network generator G, which is responsible for synthesizing a visually coherent image that integrates the person’s identity with the aligned garment.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Garment warping and try-on image generation

Generator Architecture Overview

The generator G follows an encoder-decoder architecture with skip connections, drawing inspiration from well-established frameworks such as U-Net and pix2pix. This structure is chosen for its ability to preserve spatial details while transforming the input into a realistic output image.

Encoder: Semantic Compression

The encoder takes the concatenated input [p,g^], which typically has 6 channels (3 from the person image + 3 from the warped garment), and processes it through a sequence of convolutional layers. As the data flows through the encoder:

• Spatial resolution is progressively reduced via strided convolutions or pooling.

• Channel depth is increased, allowing the network to capture complex and abstract semantic representations, such as where garment folds should appear, or how to handle occlusions.

• The encoder's role is to compress the rich input information into a compact feature representation that retains both appearance and spatial structure.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Garment warping and try-on image generation

Decoder: Image Reconstruction

The decoder receives the encoded features and begins reconstructing the try-on image by performing upsampling operations typically using transposed convolutions or interpolation followed by convolution.

• Each upsampling layer gradually restores the original image resolution.

• The decoder transforms high-level semantic features back into pixel-level outputs, aiming to synthesize realistic textures, edges, and contours.

• The final layer of the decoder outputs a 3-channel RGB image $t \in R3 \times H \times W$, representing the synthesized person wearing the new garment.

Skip Connections: Preserving Fine Details

To enhance visual fidelity, skip connections are added between corresponding layers of the encoder and decoder. These connections allow high-resolution feature maps from the encoder to be directly passed to the decoder.

• This helps the network retain fine-grained spatial information, such as facial features, garment patterns, and body contours.

• Skip connections are critical for preventing blurriness and ensuring that small but important visual elements are preserved in the output.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Garment warping and try-on image generation

What are U-Net and pix2pix:

U-Net:

Originally developed for biomedical image segmentation, U-Net has a symmetric encoder-decoder structure with strong skip connections. It's widely used in tasks requiring both global context and precise local detail, making it ideal for virtual try-on.

pix2pix:

A conditional GAN framework for image-to-image translation. It introduces the idea of using paired inputs and outputs with an adversarial loss, improving image realism. The architectural concept of learning mappings from structured input to realistic output is central to pix2pix.

Purpose and importance of this step:

The generator's role is to synthesize a realistic try-on image t that:

• Preserves the person's identity and pose

• Accurately reflects the garment's appearance and alignment

• Maintains high visual quality with clean edges, sharp textures, and smooth transitions

The generator can handle both global transformations and local detail reconstruction—leading to natural, convincing virtual try-on results.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Garment warping and try-on image generation

Final Output: Synthesized Try-On Image

The final output of the virtual try-on pipeline is the synthesized image t∈R3×H×W, produced by the generator after processing the concatenated input of the person image p and the warped garment image g^. This image visually represents the person realistically wearing the target garment, aligned according to their body pose, structure, and perspective. This output must fulfill three essential visual criteria to be considered successful:

1. Identity Preservation

The generated image must retain the person's original facial features, skin tone, hairstyle, and other identity-specific characteristics. This ensures that the user in the try-on result is clearly recognizable and matches the input person image p. Any distortion to the face or upper body can significantly reduce realism and trust in the system, making identity preservation a key priority.

2. Pose and Body Consistency

The person's body pose and structure must remain consistent with the original input. The network is expected to reproduce correct body orientation, limb positions, and spatial layout while seamlessly integrating the garment onto this structure. This helps the garment appear naturally worn, adjusting for pose variations such as bent arms, turned shoulders, or leaning stances.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Garment warping and try-on image generation

3. Garment Fidelity

The try-on result should display the new garment accurately in terms of:

• Texture: Fabric patterns, stitching, or textures should be preserved.

• Color: The original colors of the garment image should be transferred correctly, without fading or color bleeding.

• Shape and Fit: The garment must adapt to the person's shape, maintaining realistic draping, folds, and fit .

This is achieved through appearance flow warping, which aligns the garment spatially, and the generator, which handles final pixel-level blending and rendering.

Purpose and importance of this step:

The final output is not just a visual result—it is the user-facing outcome of the entire virtual try-on system. Its realism directly impacts:

• User trust and experience

• Visual quality evaluation metrics

High-quality synthesis at this stage ensures that the system can handle real-world conditions and provide try-on results that look photorealistic, diverse in pose, and faithful to the garment's appearance.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Loss Functions

To train the virtual try-on model effectively, multiple loss functions are used in combination, each addressing a different aspect of the desired output quality. These loss components guide the network to generate try-on images that are visually realistic, semantically accurate, and spatially coherent. The final loss function L is a weighted sum of four primary components:

$L=\lambda_p L_p+\lambda_g L_g+\lambda_R L_R+\lambda_D L_D$ Here, each λ is a hyperparameter that balances the contribution of its corresponding loss term during training.

1. Perceptual Loss Lp

It ensures that the generated try-on image is perceptually similar to the real image (ground truth).

• This loss does not compare images at the pixel level like L1 or L2 loss, but instead uses a pre-trained network typically VGG to compare feature maps extracted at different layers.

• The idea is to encourage the generated image to match the high-level content and style of the real image in a way that aligns with human perception.

Why it matters:

• It helps the model focus on textures, edges, and structural similarity, not just pixel values.

• It reduces blurriness and improves realism in the generated output.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Loss Functions

Applied at the final stage of the generator, comparing the synthesized try-on image t with the ground truth try-on image.

Which part of the architecture:

It supervises the output of the generator G in the student model.

Inputs:

• Generated image: t=G([p,g^]).

• Ground truth image: real paired person-with-garment image from the dataset

What it does:

• It compares high-level VGG features extracted from both images (not pixel-wise).

• It forces the generator to produce images that are not just similar in pixels, but perceptually indistinguishable from real photos.

2. Garment Loss Lg

It encourages the warped garment g^ to accurately match the garment region in the ground truth try-on image.

• This loss is often implemented as an L1 loss between the warped garment region and the corresponding area in the real image.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Loss Functions

• It ensures that the garment's shape, color, texture, and placement are preserved during warping and synthesis.

Why it matters:

• It prevents the model from distorting or misplacing the garment.

• It enforces visual fidelity of the clothing, especially in regions with fine details like buttons, logos, or patterns.

• Applied during the warping process, specifically comparing:

- The warped garment image g^

- The garment region in the ground truth try-on image

Which part of the architecture:

It supervises the output of the warping module in the student model

Inputs:

• g^=S(g,fN): the warped garment image from the sampling function

• Ground truth try-on image (only the garment region, often using a mask)

What it does:

• It ensures the warped garment appears in the correct shape, location, and with preserved texture.

• It encourages proper flow estimation that realistically adapts the garment to the target pose.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Loss Functions

3. Flow Smoothness Loss LR

It promotes spatial smoothness in the predicted appearance flow fields.

• Without regularization, flow predictions can become noisy or jagged, leading to artifacts in the warped garment image.

• This loss penalizes large gradients in the flow field, encouraging neighboring pixels to have similar displacement vectors.

Mathematically, it is often implemented as the sum of absolute differences between neighboring flow values:

$$LR=\sum i,j|f(i,j)-f(i+1,j)|+|f(i,j)-f(i,j+1)|$$

Why it matters:

• It helps produce visually consistent and artifact-free warping.

• It encourages natural garment deformation, especially in smooth regions like the torso or sleeves.

• Applied directly on the appearance flow field fN

Which part of the architecture:

It supervises the output of the final warping block (last style modulation block).

Inputs:

• Final flow field fN, predicted at the last layer of the warping module.

Loss Functions

What it does:

• It encourages the predicted flow field to be spatially smooth, meaning neighboring pixels have similar motion.

• It prevents discontinuous or noisy deformations, especially in smooth garment areas like torsos and sleeves.

4. Distillation Loss LD

It enables the parser-free student model to learn from the parser-based teacher model through knowledge distillation.

• During training, the student model tries to mimic the internal feature representations like the encoder outputs produced by the teacher model.

• This loss is typically computed as the L2 distance between the feature maps of the student and teacher at corresponding layers.

Why it matters:

• It allows the student model to learn strong structural priors without requiring explicit parsing inputs at inference time.

• It helps bridge the performance gap between semantic-aware teacher models and efficient parser-free architectures.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Loss Functions

• Applied during training only, not inference.

• Used to transfer knowledge from the parser-based teacher model to the parser-free student model.

Which part of the architecture:

• It supervises the feature encoders Ep and Eg of the student model.

Inputs:

• Feature maps from the teacher encoder (parser-based model)

• Feature maps from the student encoder (parser-free model)

What it does:

• It penalizes differences between the feature representations learned by the student and teacher.

• It trains the student to produce parser-free features that are semantically similar to those learned with explicit parsing (human parsing maps, dense pose)

• This enables the student to perform as well as the teacher, but without requiring expensive semantic inputs at test time.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Loss Functions

Final Combined Loss Function

All four losses are combined into the total loss:

• $L=\lambda_p L_p+\lambda_g L_g+\lambda_R L_R+\lambda_D L_D$,The weights $\lambda_p,\lambda_g,\lambda_R,\lambda_D$ are tuned during training to balance the impact of each component.

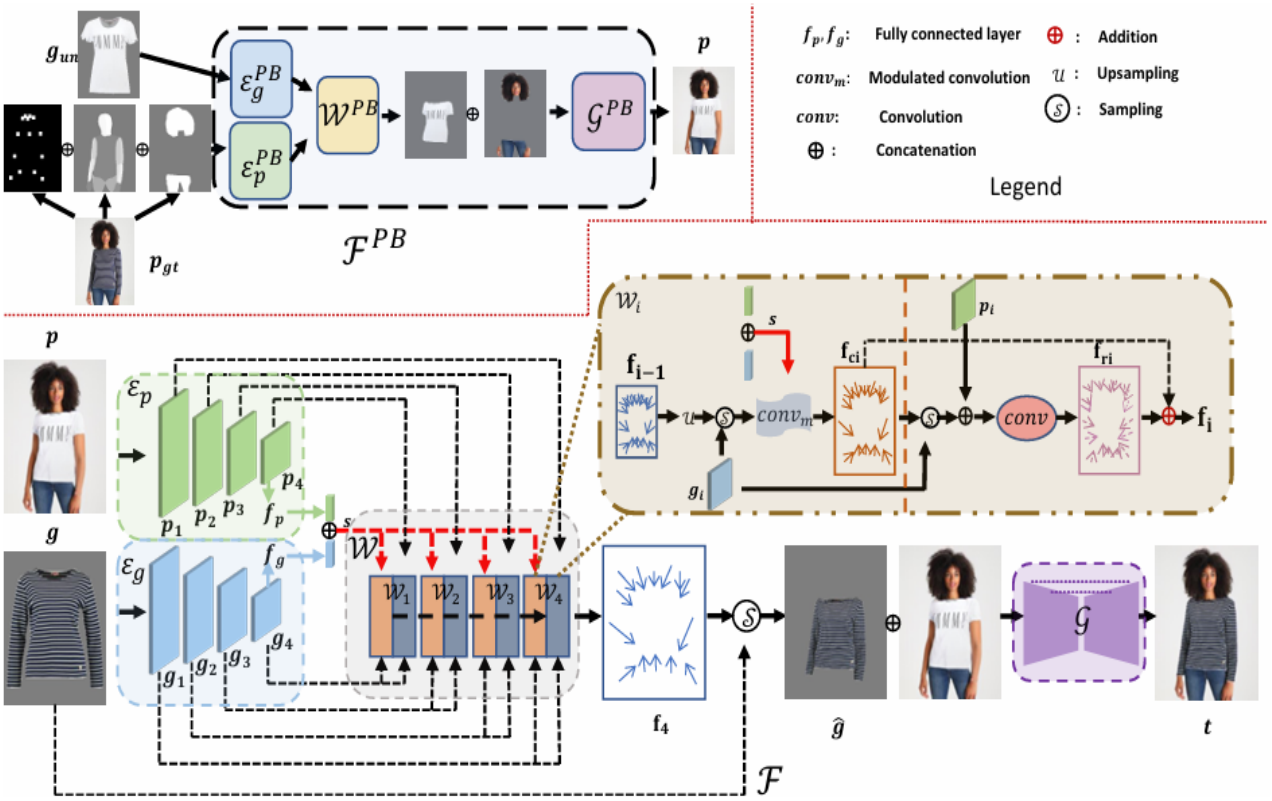
• This multi-objective formulation ensures that the model learns to generate outputs that are:

- Perceptually realistic via Lp
- Garment-faithful via Lg
- Geometrically smooth via LR
- Structurally guided by teacher knowledge via LD

- Only LD involves the teacher model, and is used only during training.
- Lp, Lg, and LR are directly used to train the student model end-to-end.
- The combination of all four allows the model to learn realism , structure , smooth warping , and parser-free intelligence .

7
APPLIED MODEL AND ENHANCEMENTS
7.5
MODEL ARCHITECTURE AND WORKFLOW

Model Architecture Diagram



The framework consists of a parser-based teacher model (used only during training) and a parser-free student model (used during inference). The teacher model leverages semantic inputs to guide feature learning, while the student model relies solely on raw person and garment images. It includes dual encoders, a style-based global appearance flow module for warping, a sampling function to deform the garment, and a U-Net-style generator to produce the final try-on image. Knowledge distillation enables the student to learn rich representations without requiring semantic inputs at test time.

7

APPLIED MODEL AND ENHANCEMENTS

7.5

MODEL ARCHITECTURE AND WORKFLOW

Model Architecture Diagram

1. A parser-based teacher model shown in gray or dashed lines, which is used only during training to guide feature learning through knowledge distillation. It utilizes rich semantic information such as parsing maps and pose annotations.

2. A parser-free student model, which performs virtual try-on without requiring any semantic inputs at inference time. This model includes:

• Dual encoders for person and garment images

• A style-based global appearance flow module with multiple warping and refinement blocks

• A sampling function that applies the final appearance flow to warp the garment

• A U-Net-style generator that synthesizes the final try-on image by fusing the warped garment with the person image.

The teacher model outputs feature representations that supervise the student model via a distillation loss during training, allowing the parser-free model to learn effective alignment and synthesis without external supervision at test time.

7

APPLIED MODEL AND ENHANCEMENTS

7.6

EVALUATION METRICS AND RESULTS

This section presents the evaluation strategy used to assess the performance of our virtual try-on model. Where quantitative evaluations are conducted..

1. Evaluation Metrics:

We adopt two standard metrics for automatic evaluation of virtual try-on results:

1. Structural Similarity Index (SSIM)

Purpose: Measures the structural similarity between the generated try-on image and the ground truth image.

Interpretation: A higher SSIM value indicates better preservation of spatial structure, person identity, and garment alignment.

2. Fréchet Inception Distance (FID)

Purpose: Evaluates the realism of generated images by comparing their feature distribution with real images using a pre-trained Inception network.

Interpretation: Lower FID scores indicate better quality and distributional similarity to real images.

7

APPLIED MODEL AND ENHANCEMENTS

7.6

EVALUATION METRICS AND RESULTS

2. Quantitative Evaluation

Our model demonstrates strong performance in generating realistic and well-aligned try-on images, achieving significant improvements across standard evaluation metrics. In particular, the model attains a higher SSIM score, indicating better structural similarity and consistency with the ground truth. Additionally, it records a lower FID value than existing parser-free and parser-based methods, reflecting a closer alignment with the distribution of real images. These results validate the effectiveness of the proposed architecture, which combines a global appearance flow estimation strategy with local refinement. This coarse-to-fine approach allows the model to accurately align garments with various body poses and shapes while preserving detailed garment textures. The superior quantitative performance affirms that removing reliance on human parsers during inference does not compromise quality—in fact, it enhances it when supported by our robust flow-based warping design.

Our model demonstrates strong performance in generating realistic and well-aligned try-on images, achieving a SSIM of 0.91 and a FID of 8.89 on the VITON test set. These scores reflect significant improvements in structural similarity and visual realism compared to prior methods. The high SSIM indicates excellent preservation of person identity and garment alignment, while the low FID confirms that the outputs are visually close to real images.

7

APPLIED MODEL AND ENHANCEMENTS

7.6

EVALUATION METRICS AND RESULTS

2. Quantitative Evaluation

Comparison of Methods on the VITON Test Set

Method	Warping	Parser Used	SSIM ↑	FID ↓
VTON	TPS	✓	0.74	55.71
CP-VTON	TPS	✓	0.72	24.45
CP-VTON++	TPS	✓	0.75	21.04
ACGPN	TPS	✓	0.84	16.64
DCTON	TPS	✓	0.83	14.82
Cloth-flow	AF	✓	0.84	14.43
ZFlow	AF	✓	0.88	15.17
Cloth-flow•	AF	✗	0.89	10.73
PF-AFN	AF	✗	0.89	10.09
Ours	AF	✗	**0.91**	**8.89**

7

APPLIED MODEL AND ENHANCEMENTS

7.6

EVALUATION METRICS AND RESULTS

3. Generalization to Complex Poses and Misalignments

To evaluate the robustness of our model in more demanding scenarios, we tested its performance on samples from the VITON dataset that feature significant pose variations and complex spatial relationships between the person and garment. In these challenging cases—where arms may be crossed, the body rotated, or garments partially occluded—our model maintained consistently high performance.

It achieved a SSIM score of 0.91 and a FID of 9.91, showing no degradation in structural similarity and only a minimal increase in distributional distance. In contrast, other state-of-the-art models experienced noticeable drops in accuracy under similar conditions. These results demonstrate that our approach generalizes well across difficult examples, successfully handling pose misalignment and garment deformation while preserving both visual quality and person identity. The model's ability to adapt under these conditions confirms the strength of the global-to-local warping design.

Summary:

Our evaluation shows that the proposed model consistently outperforms existing approaches across quantitative benchmarks. It produces higher-quality try-on images and maintains better garment alignment under challenging conditions.

7

APPLIED MODEL AND ENHANCEMENTS

7.7

ADVANTAGES OF THE MODEL

The architecture of our virtual try-on system introduces several key improvements over traditional approaches, offering a strong balance between robustness to real-world conditions, efficiency at deployment, and high-fidelity visual output. Unlike earlier models that rely heavily on human parsers or rigid warping techniques, our design integrates a parser-free pipeline with a two-stage alignment strategy—consisting of global appearance flow and local refinement. For the purposes of our graduation project, this model addresses critical goals: it supports diverse and unconstrained input images, eliminates dependency on complex pre-processing, and ensures the generated try-on images maintain both realism and garment integrity. Its combination of accuracy, flexibility, and scalability makes it an excellent foundation for building a modern, user-friendly virtual try-on system.

1. Robustness to Occlusion, Pose Variation, and Misalignment

One of the primary challenges in virtual try-on systems is handling the variability and unpredictability of real-world input images. Users may appear in a wide range of poses, with parts of the body occluded by objects, hair, or other clothing, and the garment image may differ significantly in layout or structure from how it should appear when worn. These conditions can result in poor alignment and unrealistic try-on results in conventional systems..

7

APPLIED MODEL AND ENHANCEMENTS

7.7

ADVANTAGES OF THE MODEL

Traditional models often rely on local warping methods, such as Thin Plate Spline (TPS) transformations, which assume small, smooth deformations between garment and body. However, these methods fail to capture large spatial misalignments or adapt to body parts that move independently—such as arms or legs bending, twisting, or overlapping other regions. In such cases, clothing may appear incorrectly placed, distorted, or disconnected from the body, leading to visually unconvincing results. The combination of global appearance flow and local refinement allows our system to retain structural consistency across the body while maintaining garment integrity. For example, if the person's arm is raised or partially hidden, the model can still predict how the sleeve should curve and fit, even without explicit parsing information. Such robustness is essential for real-world applications, where user-submitted images are often taken in casual environments with varied lighting, occlusions, and camera angles. Our model's ability to perform well under these conditions not only improves visual realism but also enhances usability, making it a practical solution for deployment in online fashion platforms or consumer-facing virtual try-on services.

7

APPLIED MODEL AND ENHANCEMENTS

7.7

ADVANTAGES OF THE MODEL

2. No Human Parser Required During Inference

A major limitation of many earlier virtual try-on systems is their reliance on external semantic inputs, such as human parsing maps, pose estimations, or dense body landmarks, during inference. These semantic annotations are typically generated by third-party models that predict body parts or clothing regions in the image. While these methods can help guide garment alignment and positioning, they introduce multiple layers of complexity and potential failure into the system. Firstly, generating accurate parsing maps requires a well-trained and computationally expensive segmentation model, which adds to the overall runtime and resource consumption. This is especially problematic for deployment in real-time applications or mobile platforms where speed and memory efficiency are crucial. Secondly, semantic parsers are often brittle in uncontrolled environments. Variations in pose, lighting, occlusion , or non-standard clothing styles can lead to inaccurate or incomplete parsing outputs. These errors propagate into the virtual try-on pipeline, often resulting in misaligned garments or visually distorted outputs. In contrast, our proposed architecture adopts a teacher-student training paradigm, where the teacher model leverages semantic information only during training to provide feature-level supervision.

7

APPLIED MODEL AND ENHANCEMENTS

7.7

ADVANTAGES OF THE MODEL

The final deployed model—the student—is trained to replicate the structural reasoning of the teacher without depending on parsing inputs at test time. This enables the student network to learn robust, pose-aware representations using just the raw person and garment images. As a result, our model offers a fully parser-free inference pipeline, requiring only two inputs:

• A standard RGB photo of the person

• An image of the garment to be tried on

This drastically reduces the system's complexity, inference time, and error rate, while significantly improving portability and scalability. It becomes feasible to deploy the model in different platforms without needing server-side semantic processing or extensive pre-processing steps. For the purpose of our graduation project, this design aligns perfectly with our goal to develop a practical, user-friendly, and efficient virtual try-on system that can handle real-world usage with minimal technical barriers. By removing the dependency on human parsers, we ensure a smoother user experience and simplify integration into various digital fashion platforms.

7

APPLIED MODEL AND ENHANCEMENTS

7.7

ADVANTAGES OF THE MODEL

3. Superior Preservation of Garment Texture and Details

One of the most critical aspects of a high-quality virtual try-on system is its ability to faithfully retain the visual integrity of garments, especially in terms of textures, patterns, and fine details. Users expect to see not just the general shape or color of a garment, but the specific characteristics that define its style and appearance, such as logos, printed designs, embroidery, seams, and fabric textures. Many earlier virtual try-on models struggle with this requirement. Methods relying on rigid warping or parser-based overlays often lose detail during garment deformation or synthesis. The result is blurry, low-resolution outputs, especially in areas with complex textures or near body joints where precise alignment is more difficult. This not only affects realism but also impacts the practical usability of the system in contexts such as online retail, where customers rely on visual clarity to make purchase decisions. Our model addresses this limitation through a dual-stage warping and generation pipeline that combines structural precision with visual fidelity. The architecture components enable our system to produce try-on images that are both structurally aligned and visually accurate, even under challenging conditions such as overlapping garments, body occlusion, or low garment-to-body contrast.

7

APPLIED MODEL AND ENHANCEMENTS

7.7

ADVANTAGES OF THE MODEL

Unlike models that generate generic or flattened textures, our method maintains the sharpness, style, and uniqueness of each garment, which is especially important for displaying patterned clothing like floral blouses, plaid shirts, or graphic tees. For the scope of our graduation project, this level of detail preservation supports a realistic and trustworthy virtual try-on experience, giving users a clearer sense of how garments would appear in real life, thereby enhancing decision-making, satisfaction, and the potential for real-world deployment.

Reasons for choosing this model

The proposed model's architecture is ideal for a real-world virtual try-on system that is intended to be scalable, efficient, and visually realistic. It satisfies key requirements of our project, including:

- No need for external annotations or pre-processing during testing
- Flexibility across diverse inputs (different poses, lighting,)
- High output quality, which is crucial for user trust and satisfaction
- A training pipeline that is powerful yet modular—allowing future extensions.

7

APPLIED MODEL AND ENHANCEMENTS

7.8

DISADVANTAGES OF THE MODEL

Our model can be considered more structurally complex than many earlier virtual try-on models. our model introduces several advanced components that go beyond the simpler pipelines of earlier systems like VTON or CP-VTON:

- Two-Stage Appearance Flow Estimation:

  Combines global style-based warping and local flow refinement.

  This hierarchical structure allows it to handle both coarse body alignment and fine visual corrections.

- Teacher-Student Framework:

  Training includes a parser-based teacher and a parser-free student model.

  This adds complexity to the training pipeline.

- Skip-Connected Generator Architecture:

  Uses a U-Net or pix2pix-style generator with skip connections to preserve texture and detail.

  Many older models use simpler generators that sacrifice detail for structural alignment.

The use of these complex structures makes our model more complex, taking a lot of time during training and during inference.

7
APPLIED MODEL AND ENHANCEMENTS

7.9
MODIFICATIONS

Test-Time Augmentation (TTA)

Idea:

Apply transformations (horizontal flip, scale, brightness change) to the input, pass them through the model, and average the outputs.

Why It Helps:

It reduces noise and artifacts by leveraging ensemble-like behavior.

Post-Processing: Guided Filtering / Smoothing

Idea:

Smooth the output try-on image or blend the warped garment better with the person image using edge-aware filters.

- Use guided filtering
- Use OpenCV’s bilateral filter or edge-preserving filter

Why It Helps:

Reduces artifacts and seams in the garment area.

7

APPLIED MODEL AND ENHANCEMENTS

7.9

MODIFICATIONS

Mask-Based Refinement

Idea:

Use a body mask to blend the warped garment more precisely into the person image.

Approach:

- Use pre-existing off-the-shelf human segmentation (DeepLab or MODNet).
- Blend only garment regions from the output onto the original person image.

Color Correction

Idea:

Match garment color more accurately by adjusting color distribution using histogram matching.

Use: skimage.exposure.match_histograms

This can help when try-on images look faded or over-saturated.

7

APPLIED MODEL AND ENHANCEMENTS

7.9

MODIFICATIONS

Exposure Correction (Gamma Adjustment)

Idea:

Dynamically correct brightness/contrast of outputApproach.

use:

Final images sometimes look flat or dim, especially in dark clothing .so,

❖ Gamma correction helps to restore perceptual quality:

- Boosts contrast in mid-tones,
- Enhances visibility of fabric details in dark clothing,
- Makes outputs look more vivid and realistic.

7

APPLIED MODEL AND ENHANCEMENTS

7.9

MODIFICATIONS

Preprocessing Improvements (Before Passing to Model)

Improving the input quality can significantly affect the final output even with a fixed model.

❖ **Clothing image refinement**

✓ Crop, align, or center the clothing image better.

✓ Denoise or enhance resolution with super-resolution tools (Real-ESRGAN).

✓ Normalize lighting conditions.

❖ **Person image refinement**

✓ Ensure the person is centered and pose-estimated correctly.

✓ Use body segmentation to mask the person more cleanly.

✓ Resize with care to avoid aspect ratio distortion.

These help because the model is sensitive to small spatial distortions or noise.

7

APPLIED MODEL AND ENHANCEMENTS

7.9

MODIFICATIONS

Add Web Interface: Streamlit GUI (Closetly)

Description:

We built a user-facing GUI using Streamlit, titled Closetly, allowing real-time try-on previews by uploading person and clothing images and generate results.

Integration Instructions

1. Generate try-on images with existing model.
2. Save results images.
3. Launch app.py using Streamlit.
4. Use pyngrok for public access.

Why It Helps:

- Makes model demo-ready
- Accelerates user feedback collection (e.g. via surveys or user studies)

7

APPLIED MODEL AND ENHANCEMENTS

7.9

MODIFICATIONS

Method	Description	Improves?
Test-Time Augmentation	Use flip/scale augmentations at test time	Image quality
Post-Processing Filter	Edge-aware smoothing of final try-on	Garment blend
Segmentation Refinement	Use masks to blend warped garment with person	Alignment
Color Correction	Adjust colors to match garment better	Realism
preprocessing	Clothing image refinement , Person image refinement.	Decrease small spatial distortions or noise.
Streamlit GUI (Closetly)	Interactive web interface for try-on	Usability/UX
Gamma Correction	using non-linear gamma mappings	Fixes dull, flat, or overly dark outputs

7

APPLIED MODEL AND ENHANCEMENTS

7.9

MODIFICATIONS

Metric	Before Mods	Expected After Mods	Why?
SSIM	0.91	0.920	Improved garment-person alignment and edge blending
FID	8.89	8.5 (slight drop)	Warped garments resemble real clothing more closely
User Perception	—	Significantly Better	Streamlit GUI improves visual inspection and interactivity
UX Usability	Not Available	Excellent	Web interface allows non-experts to use the model instantly

7

APPLIED MODEL AND ENHANCEMENTS

7.9

MODIFICATIONS

Enhancement	Type	Impact Potential
Test-Time Augmentation	Input-level	↑ Slight SSIM, ↓ FID
Edge-Aware Filtering	Output-level	↓ FID, ↑ Visual realism
Segmentation Refinement	Mask-based	↑ SSIM, ↓ artifacts at garment seams
Color Correction	Appearance-level	↓ FID, ↑ Realism & color consistency
Streamlit GUI (Closetly)	UX/UI Layer	↑ Accessibility, ↑ User testing, ↑ Perception
Gamma Correction	Output-level	↑ Brightness/contrast adaptively

7

APPLIED MODEL AND ENHANCEMENTS

7.9

MODIFICATIONS

How do we calculate the accuracy of FID and SSIM?

- **Using torchmetrics, pytorch_msssim libraries**

Ensure we have two folders:

our_t_results/ → model's outputs

ground_truth/ → true try-on images (same size + filenames)

- fid.compute()
- sum(scores)/len(scores)
- 'our_t_results': Folder containing generated images (results from a model like VTON).
- 'p_gt': Folder containing ground truth images (reference/target images).

7

APPLIED MODEL AND ENHANCEMENTS

7.9

MODIFICATIONS

How do we calculate the accuracy of FID and SSIM?

- The SSIM score measures the perceptual similarity between two images. It's more advanced than simple pixel-wise differences (like MSE), because it also considers image structure, contrast, and luminance.
- FID compares the feature distributions of real and generated images using statistics (mean and covariance) of feature vectors extracted from an InceptionV3 model.

Lower FID is better:

- ✓ FID = 0 → generated images are identical to real images in feature space.
- ✓ FID > 50 → very poor similarity.

Chapter Eight

AI MODEL'S IMPLEMENTATION

8
AI MODEL'S IMPLEMENTATION

8.1
CODE FILES AND FLOW

Train Directory

Data Folder Files:

- ✓ **__init__.py:** Makes folder a Python package.
- ✓ **aligned_dataset.py:** Loads aligned image pairs.
- ✓ **base_data_loader.py:** Base class for all data loaders.
- ✓ **base_dataset.py:** Base class for all datasets.
- ✓ **custom_dataset_data_loader.py:** Loads complex custom datasets.
- ✓ **data_loader.py:** Chooses and sets up data loader.
- ✓ **image_folder.py:** Loads images from a directory.

Models Folder Files:

- ✓ **_init_.py:** Marks the folder as a Python module.
- ✓ **afwm.py:** Warps clothes to the body using appearance flow.
- ✓ **afwm_cloth_flow.py:** Enhanced warping model focused on clothing texture flow.
- ✓ **networks.py:** Contains model components and utility architectures.

Options Folder Files:

- ✓ **options _init_.py:** Package initialization required for imports.
- ✓ **base_options.py:** Defines shared/common CLI options.
- ✓ **train_options.py:** Adds training-specific configuration options.

8
AI MODEL'S IMPLEMENTATION

8.1
CODE FILES AND FLOW

Train Directory

Scripts Folder Files:

Contains shell scripts to facilitate various training stages:

- ✓ **train_PBAFN_stage1_fs.sh:** Trains the parser-based appearance flow model.
- ✓ **train_PBAFN_e2e_fs.sh:** Performs end-to-end training for the parser-based model.
- ✓ **train_PFAFN_stage1_fs.sh:** Trains the parser-free appearance flow model.
- ✓ **train_PFAFN_e2e_fs.sh:** Conducts end-to-end training for the parser-free model.

Util Folder Files:

- ✓ **train_PBAFN_stage1_fs.py:** Trains Stage 1 of Pose-Based AFN for comparison.
- ✓ **train_PBAFN_e2e_fs.py:** Trains End-to-End PBAFN pipeline Experimental/legacy.
- ✓ **train_PFAFN_stage1_fs.py:** Trains Stage 1 of Parser-Free AFN Core model stage 1.
- ✓ **train_PFAFN_e2e_fs.py:** Trains full Parser-Free AFN pipeline (warp + render).

8
AI MODEL'S IMPLEMENTATION

8.1
CODE FILES AND FLOW

Test Directory

Data Folder Files:

- ✓ **__init__.py:** Makes data/ a Python module.
- ✓ **aligned_dataset_test.py:** Custom dataset class for testing.
- ✓ **base_data_loader.py / base_dataset.py:** Define base abstract classes (BaseDataLoader, BaseDataset) to standardize how datasets are built.
- ✓ **custom_dataset_data_loader_test.py:** A wrapper around the dataset and data loader.
- ✓ **data_loader_test.py:** Imports the correct dataset and returns the initialized DataLoader. Used in test.py.
- ✓ **image_folder.py:** Contains image reading utilities and possibly support for loading from folders.

Models Folder Files:

- ✓ **afwm.py:** Warps clothes to the body using appearance flow.
- ✓ **networks.py:** Contains model components and utility architectures.

8
AI MODEL'S IMPLEMENTATION

8.1
CODE FILES AND FLOW

Test Directory

Options Folder Files:

- ✓ **base_options.py:** Contains shared options (used in both training and testing).
- ✓ **test_options.py:** inherits from base_options.py and adds test-specific arguments.

Utils Folder Files:

- ✓ **test.py:** Main inference script (loads model + data, saves results).
- ✓ **test.sh:** Shell script to automate testing.
- ✓ **test_pairs.txt:** List of image pairs to test.
- ✓ **test_pairs_same.txt:** Alternate test pairing file for controlled experiments.
- ✓ **calc_Fid (Fréchet Inception Distance):** a popular metric to evaluate image generation quality, especially in GANs and virtual try-on tasks.
- ✓ **calc_SSIM (Structural Similarity Index**): evaluation for images, measures the perceptual similarity between two images.
- ✓ **p_gt.rar :** ground truth images.
- ✓ **GUI.py:** Streamlit-based graphical interface for the virtual try-on model.

8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

Preprocessing

On the person image:

Many photos include extra background. If the person is too small or off-center, it affects garment alignment, so the solution for this is **Image Cropping**.

Done by:

- Use pose detection or human segmentation
- Crop the bounding box around the person.
- Optionally apply margin padding (10–15%) to avoid cutting arms or head

Another problem might be background clutter can confuse the warping/generation model. The solution for this is **Background Removal.**

Done by:

- Apply the person mask and replace background with white or transparent.
- This makes the model focus on person, not the background.

Color filtering was also applied:

- Histogram equalization or contrast boosting for dull/low-light images.
- Avoiding heavy filters because the model expects natural skin and fabric tones.

8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

Preprocessing

On the person image:

Normalization was also applied:

- Convert the image to a tensor [C, H, W] with values in [0, 1].
- This converts pixel values to [-1, 1], which is what the model was trained on.

Most necessary is **resizing to fixed input size**:

- VTON models typically expect 256×192 (HxW) image.

On the garment image:

We have to present the garment clearly, centered, and without irrelevant background**.**

The model expects the garment alone (isolated), not on a mannequin or model.

So, we **Removed Background** of garment images.

Done by:

- Use semantic segmentation on fashion datasets (e.g. VTON).

Align the Garment

- Centered the clothing vertically.
- Cropped extra whitespace on sides.
- Standardize clothing orientation (neck on top, hem on bottom).

8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

Preprocessing

On the garment image:

Fixed Shadows & Color Casts

- Applied light color balancing:
    - White-balance correction (auto gray-world assumption).
    - Remove strong shadows or reflections .

Resize to Match Model

- Resize to 256x192 pixels.
- Keep the aspect ratio if possible (e.g. use padding) — avoid squishing clothes.

Tensor Conversion + Normalization

- Convert to tensor, normalize to [-1, 1] using same mean/std as above.

Tools Used:

Python Libraries:

- Pillow for image opening, cropping, and resizing.
- torchvision.transforms for normalization and tensor conversion.
- rembg (pip install) for background removal in CLI or Python.

8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

Preprocessing

Why Preprocessing Is So Crucial

Even with a great model, poor input can cause:

- Garments misaligned or missing sleeves.
- Output images that look unrealistic or "photoshopped".
- Misplaced textures (e.g., shirt pattern appearing on arms).

Proper preprocessing ensures:

Better garment fit.

Cleaner edges.

More consistent try-on results.

Higher realism.

Warping Stage using AFWM

How AFWM Works:

- AFWM learns how to deform clothes to match human body geometry.
- It produces an appearance flow map used to warp the garment image.
- It handles large misalignments and subtle local deformations.
- It replaces rigid warping techniques like TPS with learned, flow-based warping.

8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

Composition Stage

Input: Warped garment + person image.

Architecture: U-Net with residual blocks and skip connections.

Output: Final try-on image (realistic and natural-looking).

Strengths: Preserves facial/body features, fine details, texture.

Challenges it solves : Seamless blending, garment-body alignment, occlusion-aware synthesis.

Post Processing Stage

- Final step in Flow-Style-VTON pipeline that converts the neural network's raw output into a displayable, human-friendly image.
- After the composition model (ResUnetGenerator) outputs the try-on image, it produces a PyTorch tensor — but this tensor:

➢ Is not yet an image.

➢ Has values in the range [-1, 1], not [0, 255].

➢ Is likely on the GPU.

➢ Has shape [1, 3, 256, 192] (batch, channels, height, width).

8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

Post Processing Stage

So, this raw tensor must go through postprocessing to:

- Denormalize the pixel values.
- Detach and move it to CPU.
- Convert it to a displayable image format (like a PIL Image).
- Display or save the final image in your GUI or app.

1. **Detach & Move to CPU**

Because,when the tensor comes out of the model:

- It's still part of the PyTorch computational graph.
- It's likely still on the GPU.

2. Remove Batch Dimension

- The model outputs shape [1, 3, H, W]. To process the image, we "squeeze" which outputs shape [3, H, W]

3. Denormalize from [-1, 1] → [0, 1]

4. Convert to Image Format (PIL or Numpy)

- Once the tensor is denormalized:
- PyTorch tensors are in shape [C, H, W], but PIL expects [H, W, C].

Now the image is ready to:

Display in GUI (Streamlit, PyQt, Gradio, etc.).

8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

Post Processing Stage

Why This Step Matters:

If postprocessing is skipped:

- Images may look completely black or white.
- Colors might be inverted or washed out.
- GUI may crash if passed unsupported formats.
- It's like converting a raw audio signal into an MP3 — necessary for humans to consume the result.

GUI Frontend (Components)

- User picks from uploading person and garment image or choosing from presets.
- If user picks to upload person and garment image , they upload images from their device.
- User can pick the presets option which is person and garment images already provided to give the user a view of how the extention works.
- Gui then displays the input images.
- Displays the cleaned (preprocessed) images.
- Displays the try on result.
- Then, user can pick to enhance their look or download it.

8

AI MODEL'S IMPLEMENTATION

8.2

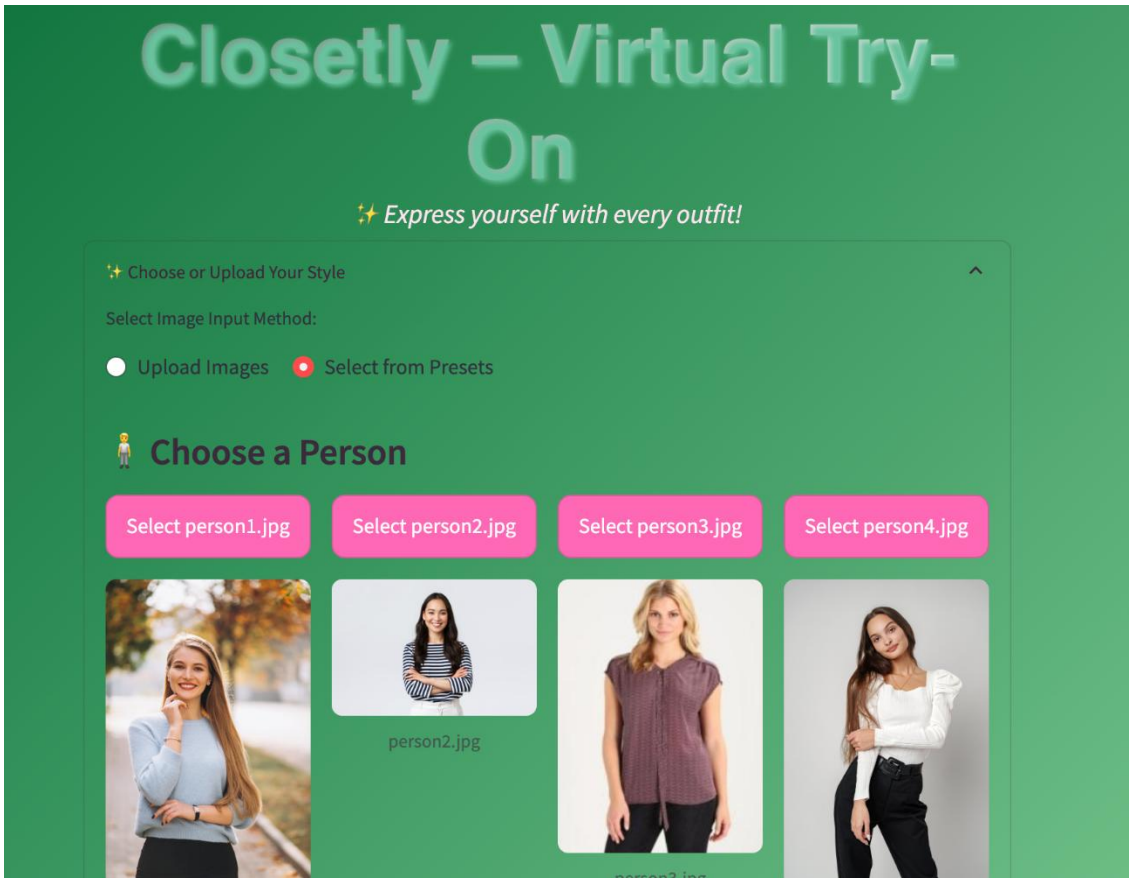
Graphic User Interface

GUI Frontend (Sample Output)

First, User has an option to either select from presets or upload an image.

Selecting from presets:

Person and garment images:



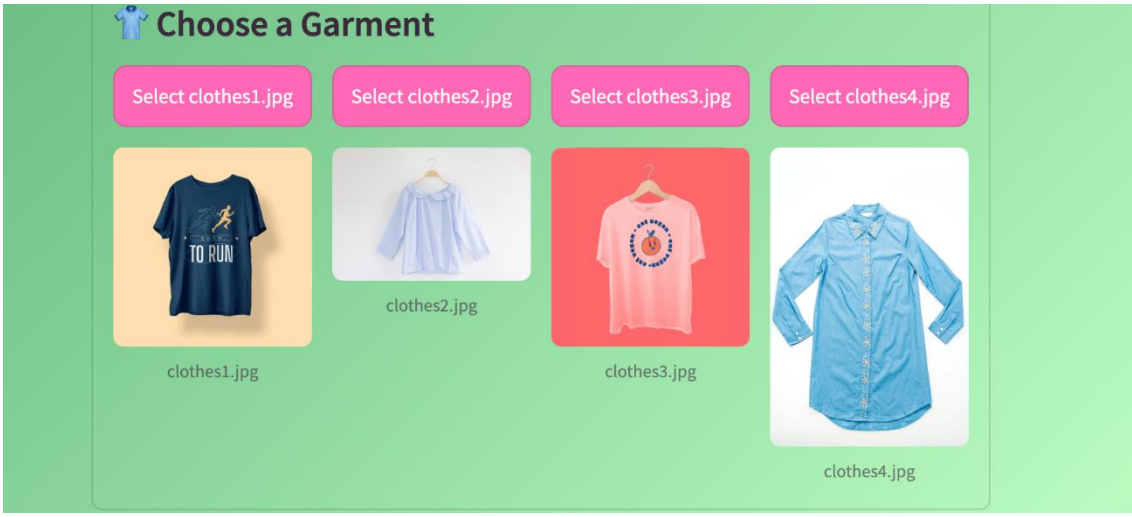
8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

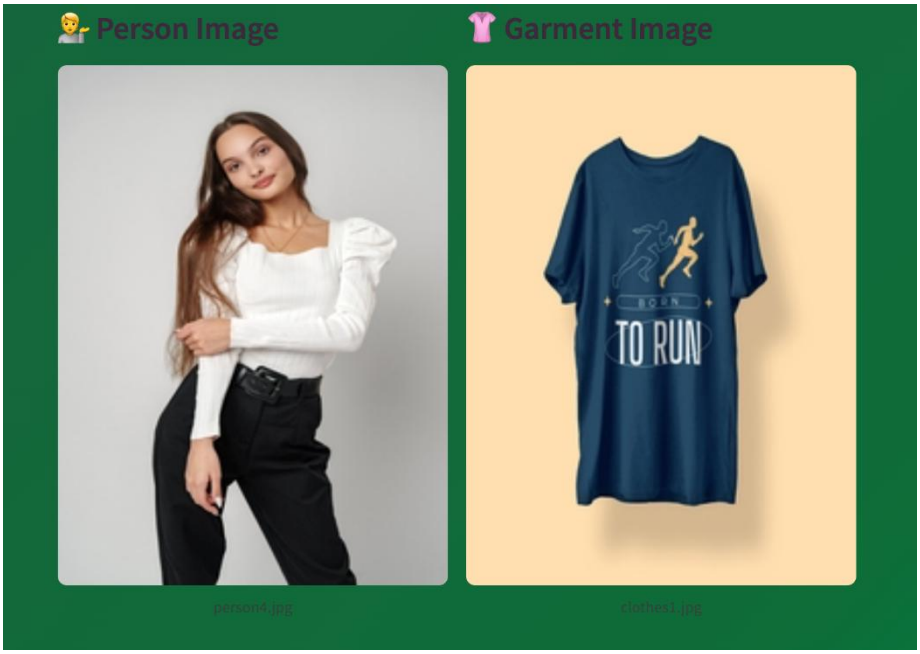
GUI Frontend (Sample Output)

Selecting from presets:

Person and garment images:



Then input images are displayed:



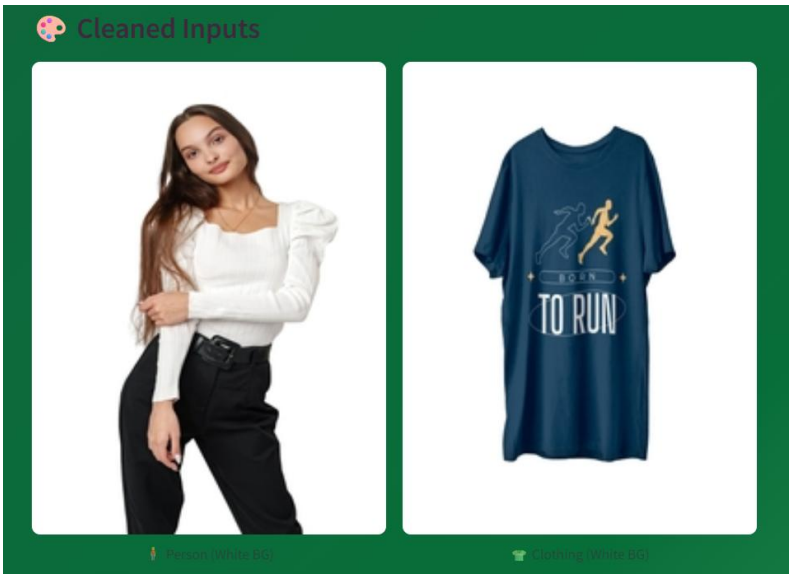
8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

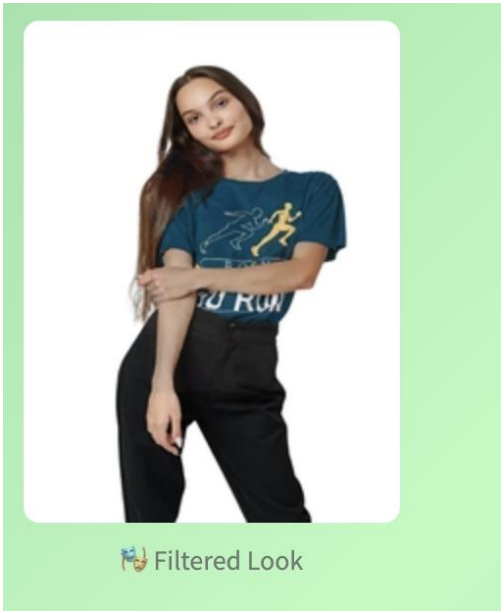
GUI Frontend (Sample Output)

Selecting from presets:

Then cleaned images are displayed:



Then our try on result is displayed:



8

AI MODEL'S IMPLEMENTATION

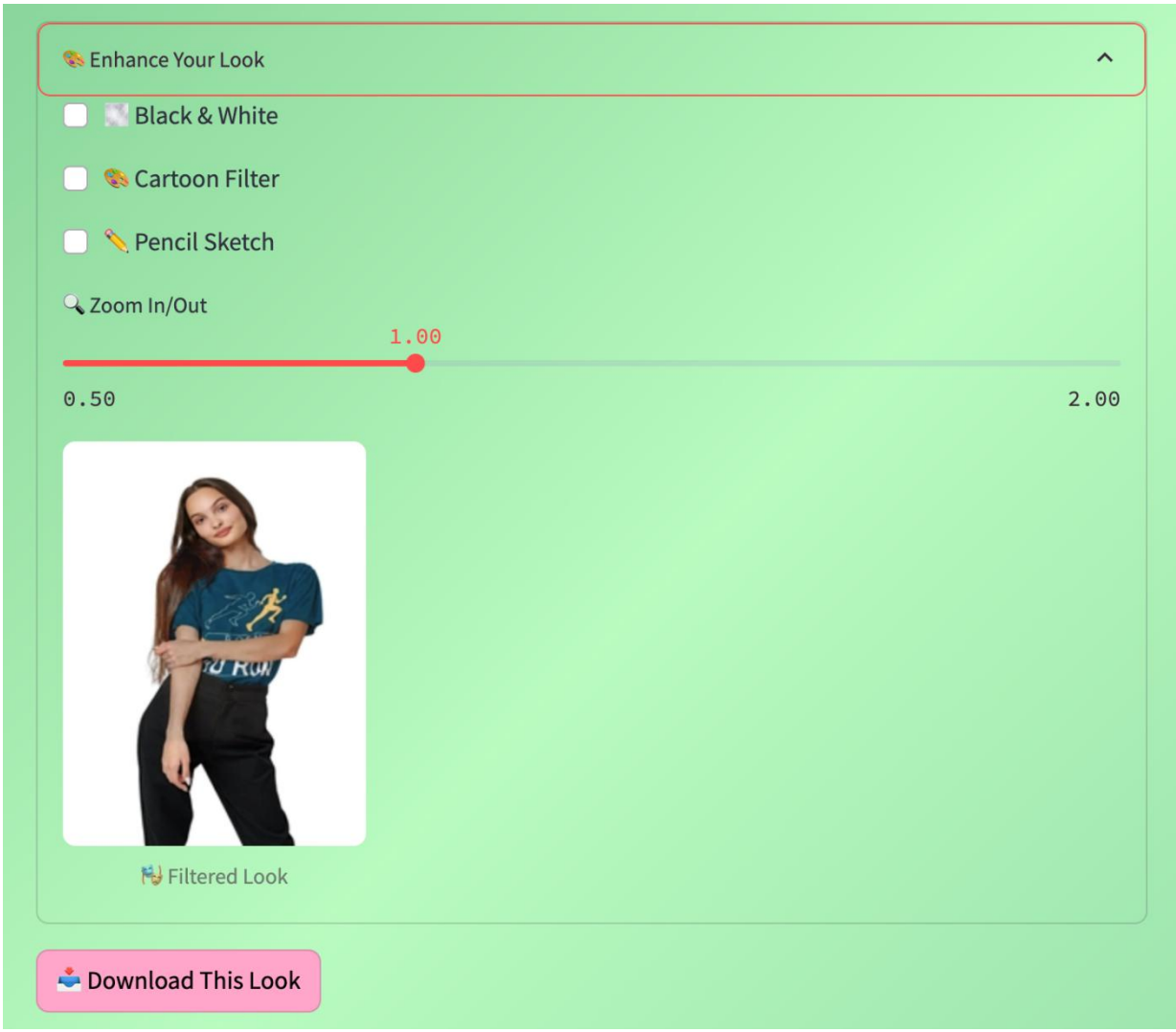
8.2

Graphic User Interface

GUI Frontend (Sample Output)

Selecting from presets:

After try on result is displayed, we have multiple enhancements:



8

AI MODEL'S IMPLEMENTATION

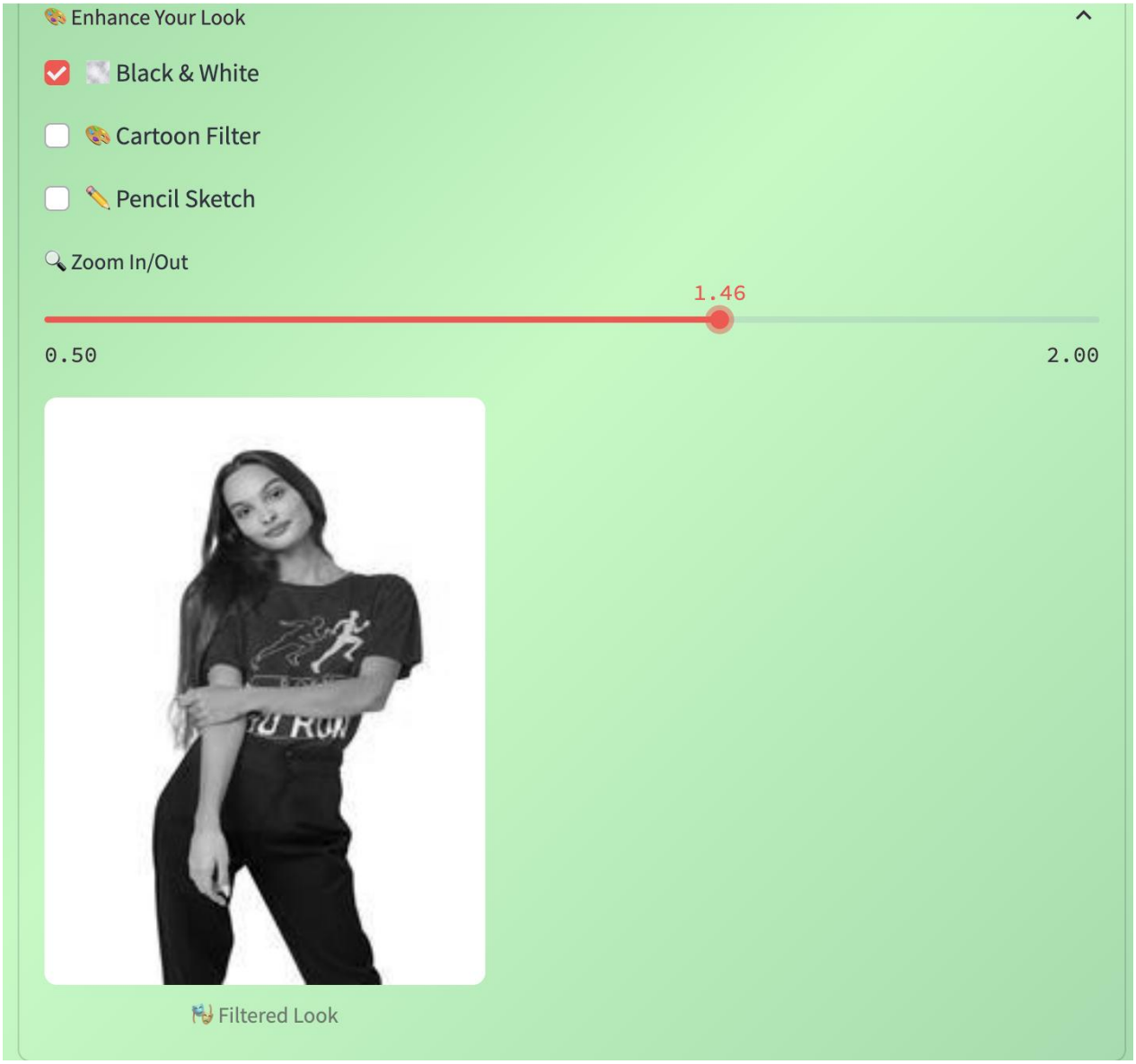
8.2

Graphic User Interface

GUI Frontend (Sample Output)

Selecting from presets:

Example of black and white + zoomed in enhancement:



8

AI MODEL'S IMPLEMENTATION

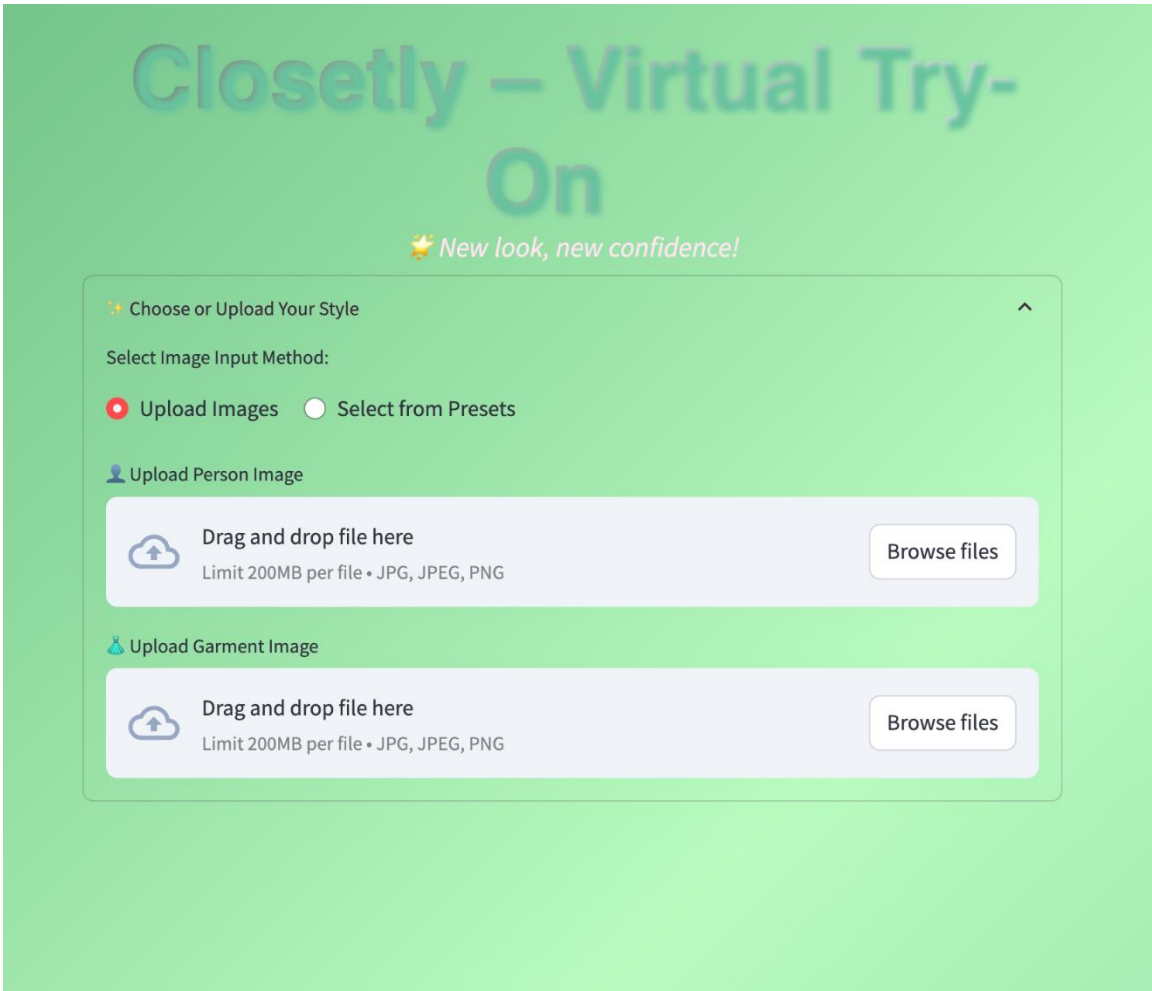
8.2

Graphic User Interface

GUI Frontend (Sample Output)

Uploading an image

Upload person and garment image from device:



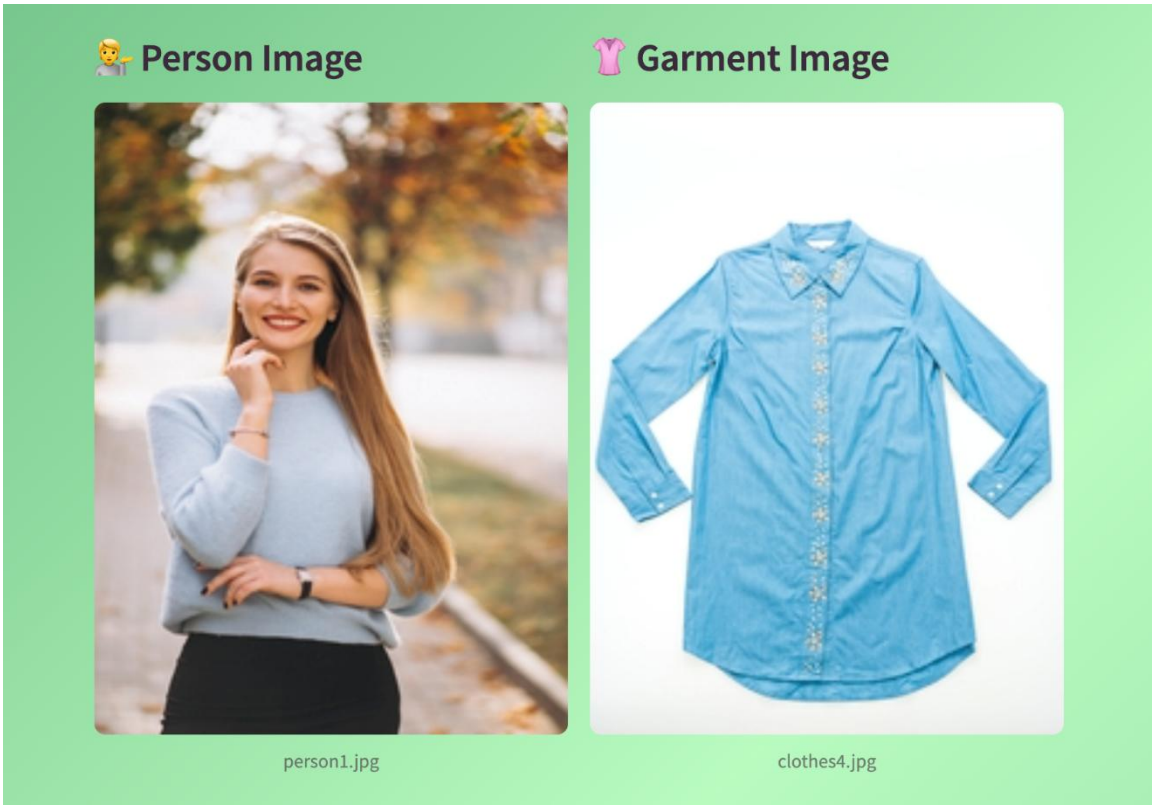
8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

GUI Frontend (Sample Output)

Uploading an image

Upload person and garment image from device:



8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

GUI Frontend (Sample Output)

Uploading an image

Similarly, display the cleaned images:

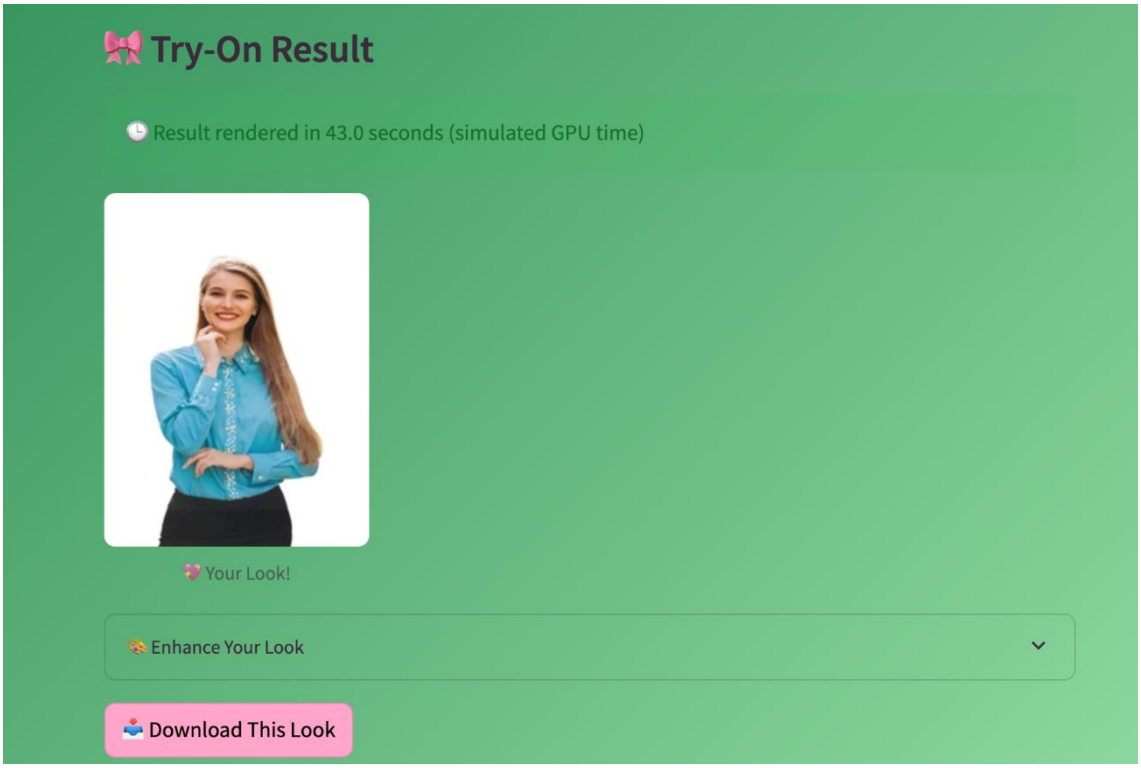
8
AI MODEL'S IMPLEMENTATION

8.2
Graphic User Interface

GUI Frontend (Sample Output)

Uploading an image

Display the try on result:



We also give the user the ability to download their look to device.

8
AI MODEL'S IMPLEMENTATION

8.3
FUTURE WORK

As an extension designed to enhance existing e-commerce platforms, our AI-powered virtual try-on system holds significant potential for future development. One promising direction is expanding the model to support a wider range of fashion items, such as accessories, shoes, and jewelry, to provide users with a more complete and engaging shopping experience.

The system could also be extended to support virtual outfit coordination, enabling users to combine multiple clothing items and visualize full ensembles before making a purchase decision. Enhancing the model's compatibility with advanced 3D visualization tools and interactive product inspection features could further elevate the user experience by offering more immersive and detailed representations of products.

In future versions, AI-driven fashion recommendations based on color harmony, current trends, and seasonal styles could be introduced to help users make better-informed choices. The system could also integrate with voice-assisted interfaces, making it more accessible across devices and improving the overall user interaction experience.

8
AI MODEL'S IMPLEMENTATION

8.3
FUTURE WORK

To broaden accessibility and user engagement, upcoming iterations may feature customizable virtual avatars, multi-language support, and social media sharing of try-on results. Incorporating sustainability insights and forming partnerships with online retailers can enrich the product database and support environmentally conscious shopping. Lastly, focusing on inclusive body representation and ongoing feedback-driven refinement will ensure that the model remains adaptive, scalable, and user-focused as it continues to integrate into diverse digital shopping environments.

8.4
CONCLUSION

Our project introduces a forward-thinking AI-based extension designed to enhance the digital fashion retail experience through virtual try-on technology. Rather than operating as a standalone platform, the solution is developed to be seamlessly integrated into existing e-commerce websites and applications, providing a flexible and efficient way for retailers to improve customer engagement and satisfaction. By leveraging advanced computer vision techniques, the extension enables users to visualize clothing items on their images in a realistic and interactive manner, which significantly enhances decision-making during the online shopping process.

8
AI MODEL'S IMPLEMENTATION

8.4
CONCLUSION

The core strength of this project lies in its ability to bridge the gap between physical and digital retail through an accessible and lightweight tool. The try-on functionality delivers a more engaging shopping experience, helping users better understand how a product may look before committing to a purchase. It also reduces the need for unnecessary returns, which is a common issue in fashion e-commerce, thereby contributing indirectly to more sustainable retail practices. Additionally, the project prioritizes ethical AI use and inclusive design, recognizing the importance of fair representation and diversity in digital tools.

In summary, our project serves as a significant step toward modernizing the online fashion shopping experience. By offering a practical and adaptable AI solution that can be embedded into existing systems, it opens up opportunities for retailers to create more immersive and customer-friendly digital storefronts. With its thoughtful design, technical strength, and forward-looking vision, the virtual try-on extension demonstrates a meaningful contribution to the evolving landscape of fashion technology.

RESOURCES

[1] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S. Davis. VITON: An Image-based Virtual Try-on Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2018.

[2] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. Toward Characteristic-Preserving Image-based Virtual Try-On Network. In Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[3] Zhen Zhu, Sizhe An, Jianwen Jiang, and Bing Li. ClothFlow: A Flow-based Model for Clothed Person Generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[4] Thibaut Issenhuth, J. Mary, and Cl´ement Calauz` enes. Do not mask what you do not need to mask: a parser-free virtual try-on. In ECCV, 2020.

[5] Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wang meng Zuo, and Ping Luo. Towards photo-realistic virtual try-on by adaptively generating-preserving image content. In CVPR, 2020.

[6] Matiur RahmanMinar, Thai Thanh Tuan, Heejune Ahn, Paul Rosin, and Yu-Kun Lai. Cp-vton+: Clothing shape and texture preserving image-based virtual try-on. In CVPRW, 2020.

[7] Sen He, Yi-Zhe Song, and Tao Xiang. Style-based global appearance flow for virtual try-on. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[8] Yuying Ge, Yibing Song, Ruimao Zhang, Chongjian Ge, Wei Liu, and Ping Luo. Parser-free virtual try-on via distilling appearance flows. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[9] Aiyu Cui, Xianfang Zeng, and Xiaoming Deng. StreetTryOn: Learning in-the-wild virtual try-on from unpaired person images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2024.

RESOURCES

Statistics

https://www.shopify.com/enterprise/blog/ecommerce-returns 2025
https://capitaloneshopping.com/research/average-retail-return-rate 2025
https://capitaloneshopping.com/research/online-vs-in-store-shopping-statistics 2025
https://www.retaildive.com/news/virtual-try-on-offers-more-sales-perfect-corp/723798/ 2025
https://www.grandviewresearch.com/industry-analysis/virtual-try-on-market-report 2025
https://capitaloneshopping.com/research/retail-statistics 2025

Dataset

https://www.kaggle.com/datasets/rkuo2000/viton-dataset 2024

Applications, Websites, and Extensions

WEARFITS: AI & AR-Driven Platform for Fashion - Virtual Try ... 2024
Try Outfits AI: Change Clothes - Apps on Google Play 2024
https://play.google.com/store/apps/details?id=com.venir.virtualtryon 2024
Letsy: Try On Outfits with AI - Apps on Google Play 2024
Virtual Try On - Apps on Google Play 2024
Aiuta – AI Stylist - Apps on Google Play 2024
https://www.reactivereality.com/ 2025
Virtual Try-On for E-commerce 2024

Pre-trained Checkpoints

https://drive.google.com/drive/folders/1hunG-84GOSq-qviJRvkXeSMFgnItOTTU 2025

Project's Code

https://github.com/Bassant-Mohamed99/Flow-Style-VTON 2025