TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction

Real world and virtual world are the two universes. When humans first discovered computers, they began to operate digitally. He has been making an effort to smoothly integrate the digital and virtual worlds. Many technologies were developed in an effort to close the gap between the virtual and physical worlds. Virtual reality, augmented reality, and mixed reality are three examples of software that connects the virtual and real worlds. This gave rise to an abundance of devices that enable users to simultaneously experience virtual and real worlds.

Smart technologies that streamline our activities have a significant impact on our daily lives as a result of the technology development industry's rapid growth. For example, online purchasing developed quickly. People are becoming more accustomed to using internet stores, online auctions, etc., to buy the things they are interested in. This kind of transaction has taken over as the most popular one and offers customers a tonne of convenience. However, a drawback of online clothing shopping is that customers cannot test out the item before purchasing it. The client's decision to purchase the items is influenced by how they feel after dressing. As a result, there is a growing need for virtual dressing rooms that can imitate the visual aspect of dressing.

It normally takes a lot of time to try on things in a store. Additionally, in situations like internet purchasing, it might not even be able to try on clothing. By establishing a virtual changing room setting, we hope to improve accessibility and time efficiency for trying on clothing. The alignment of the user and the cloth models with precise location, scale, rotation, and ordering is the only issue.

One of the first steps in solving the issue is identifying the user and their bodily parts. Several methods for body component detection and posture assessment have been put forth in the literature. Users of online shopping are better able to manage costs thanks to the use of web cameras. Thus, the buying experience of today might significantly change as a result of this virtual trial room software. People don't need to be afraid of hidden cameras or stand in line in front of trial room for hours to check out their clothes.

Because using this only takes a few seconds, people can quickly change their attire or try on gowns. the user saves a significant amount of time and exertion.

1.2 Deep learning in virtual reality What is Virtual Reality?

Virtual Reality (VR) is a computer-generated environment with scenes and objects that appear to be real, making the user feel they are immersed in their surroundings. This environment is perceived through a device known as a Virtual Reality headset or helmet. VR allows us to immerse ourselves in video games as if we were one of the characters, learn how to perform heart surgery or improve the quality of sports training to maximize performance.

Although this may seem extremely futuristic, its origins are not as recent as we might think. In fact, many people consider that one of the first Virtual Reality devices was called Sensorama, a machine with a built-in seat that played 3D movies, gave off odors and generated vibrations to make the experience as vivid as possible. The invention dates back as far as the mid-1950s. Subsequent technological and software developments over the following years brought with them a progressive evolution both in devices and in interface design

1.3 Deep Learning

Deep learning is a part of the family with basis of machine learning approaches based on the layer used in the artificial neural system. Learning can semi-supervised or unsupervised. Deep Learning architectures such as deep neural networks, deep belief networks, and convolutional neural networks are practiced in areas such as computers, speech recognition, natural language processing, voice recognition, social network filtering, translating engine, biotechnology, Drug design, medical analysis, material controls and board games where they get the results comparable and in some cases better than the experts.

The neural network was initially inspired by data processing and node distribution in the structures of the synaptic biological system, but the differences have different properties and functions, the biological brain structure that makes them incompatible with Nerve damage. Especially nerve networks tend to be symbolic and static, while the biological brain of living organisms is mostly plastic and analogous.

The most advanced models of deep learning are based on artificial nerves, especially on CNN. In deep learning, each level learns to change its data to an abstract and brief presentation.

Deep in deep learning refers to the number of layers that the data is changed. In particular, a deep learning system has a significant depth in the credit assignment process (CAP). CAP is a chain of change from the gateway. The CAP describes the relationship between creation and output. For feed forward neural networks, the depth of the cap is that the net and the number of hidden layers as a resultant layer are defined. For a neural nerve network, a signal can spread over one layer again, the depth of the CAP has unlimited potential.

1.4 Project objectives

The main objective of the proposed system is to enhance customer experience in clothing fitting by enabling customers to virtually try clothing on in order to check for size, fit or style. In this way, customers are able to shop and try their favorite clothing anywhere and anytime with smartphone. The main objective of the project is divided into sub objectives as shown as below

- To detect and extract human body skeleton-based joint positions using smartphone camera.
- To calculate body measurements based on the extracted body skeleton joint positions.
- To fit virtual garments onto human body according to the extracted body skeleton joint positions, body measurements and garment measurements.

1.5 Scope of the project

Compared to the early days, clothing shopping is getting easier and more convenient for customers and sellers especially through online shopping. However, the problem where it is compulsory to physically try clothing on body in order to check for size, fit or style is still cannot be solved. Therefore, one of the effective solutions to solve the problem is to develop an virtual fitting room. As the customers can see the garments on the computer, there would be no need for more customer service personnel to help customers to locate garments or to put the garments back on the shelf from the fitting room. The main objective of the proposed system is to enhance customer experience in clothing fitting by enabling customers to virtually try clothing on in order to check for size, fit or style.

In this way, customers are able to shop and try their favorite clothing anywhere and anytime with smartphone this will improve sales in clothing stores. Also, stores would not need large showrooms to display their garments. This would reduce stores’ operating expenditure.

1.6 Purpose of the project

This project also helps the clothing store managements by improving sale. If a customer cannot find clothes in one store, he or she might not go to another branch o f the same store. They might go to another store located in the neighbourhood. The VDR system will prevent these inconveniences. Furthermore, an easy way to see the fitting of the garment will attract more customers to the store. Therefore, this will improve sales in clothing stores. Also, stores would not need large showrooms to display their garments. As the customers can see the garments on the computer, there would be no need for more customer service personnel to help customers to locate garments or to put the garments back on the shelf from the fitting room. This would reduce stores' operating expenditure. The time saving and conveniences are the main reasons for my motivation to propose the VDR system.

CHAPTER 2

LITERATURE SURVEY

2.1 Literature Survey

[1]. Implementation of Virtual Fitting Room Using Image Processing by Srinivasan K. Vivek S. Department of Electronics and Instrumentation Engineering, Sri Ramakrishna Engineering College Coimbatore, India

The Virtual Dressing Room technique for virtually dressing a person requires separating the person from the background while taking into account changes in lighting and with the least amount of disruption to surrounding items. Following this, a Laplacian filter is used to detect the contour of the upper and lower body, followed by edge detection. Following that, feature points are extracted based on the fundamental human anatomy. The sample shirt is bent to precisely suit the person using these locations as a guide.

[2]. Image Processing Design Flow for Virtual Fitting Room Applications used in Mobile Devices by Cecilia Garcia, Nicolas Bessou, Anne Chadoeuf and Erdal Oruklu, Department of Electrical and Computer Engineering Illinois, Institute of Technology Chicago, Illinois, USA.

This work successfully created a mobile device application for a virtual changing room. The primary goal of creating a real-time, platform-neutral application was accomplished. Users can choose from sizes XS to XL, move the clothing with the arrow buttons, and, if necessary, change the width of the clothing with the scroll button.

They can also decide to experiment with various items before taking a photo of themselves to show how the clothing fits. The programme can monitor the user's position and movement while tracking and scaling the apparel. Retail businesses can use this app by releasing it to the Apple Store or Android Market.

[3]. A Virtual Trial Room using Pose Estimation and Homograph Kshitij Shah, Mridul Pandey, Sharvesh Patki, Radha Shankarmani. Department of Information Technology Sardar Patel Institute of Technology Mumbai, India

The development of a mobile application that allows users to virtually try on clothing utilizing open CV and TensorFlow lite technologies would greatly improve the shopping experience.

Al suggested a solution that would align the input fabric on the person representation using a Geometric Matching Module. The customer then uses the smartphone application to digitally try the clothing on. Using OpenCV, which first recognizes the customer's body and then maps the garments on to it, the mapping of the clothes is carried out. The clothing is kept on a firebase. The picture is kept in the gallery for further viewing.

2.2 Problem Statement

As e-commerce continues to dominate the retail landscape, the clothing sector faces a persistent challenge: the inability to try on clothes before purchasing them. While online shopping offers convenience, it often lacks the tactile experience of physically trying on apparel, leading to several issues for consumers. These include uncertainty about fit, style, and quality, which frequently results in shopping cart abandonment and high return rates. Despite advancements in online shopping platforms, the problem of accurately predicting size, fit, and style based on product images alone remains unresolved. For consumers, this leads to frustration, hesitation, and ultimately, a less enjoyable shopping experience.

To address these issues and revolutionize the online shopping experience, we propose the development of a **Virtual Styling Room** that uses **live video feeds** combined with **augmented reality (AR)** to enable users to try on clothing virtually before making a purchase. By using the camera on a smartphone, tablet, or computer, users can see themselves wearing the clothes in real- time, with the system adjusting the fit based on their body shape and movements. This immersive, interactive approach allows customers to evaluate how the clothing will look and feel on their own body without the need for a physical try-on.

The Virtual Styling Room would not only overlay apparel onto the user’s video feed, but it would also leverage **AI-driven algorithms** to recommend the most suitable sizes and styles based on the user’s measurements, preferences, and past shopping behavior. This system can provide tailored suggestions for outfits, mixing and matching different items to users visualize complete looks. Furthermore, it could offer personalized styling tips, enhancing the overallshopping experience and making it feel more like having a personal stylist at your fingertips.With the integration of this technology, online shoppers can interact with clothing in a way that mimics the in-store experience. The system will allow them to evaluate the fit from multiple angles, adjust for changes in posture, and even test out clothing combinations. This would reduce the anxiety associated with buying clothing online and increase consumer confidence. As a result, we expect a reduction in returns and exchanges, leading to greater satisfaction for both shoppers and retailers. The Virtual Styling Room aims to streamline the shopping process, reduce time spent browsing, and create a more engaging, personalized shopping journey that bridges the gap between online and physical retail experiences.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 SOFTWARE REQUIREMENT SPECIFICATION

1. Frontend Technologies:

- HTML5
- CSS3
- JavaScript (ES6+)

2. Backend Technologies:

- Python(Machine Learning)
- Flask Frame Work

3. Database:

- SQLIPE3

4. Development Tools:

- Code editor (e.g., Visual Studio Code)
- Version control system (e.g., Git)
- Package manager (e.g., npm)

3.2 Hardware Requirement Specification

- Processor – Intel(R) Core (TM) i7-3770 |CPU @ 3.40GHz.
- RAM – 4 GB or above.
- Hard Disk – Free disk space of above 6 GB
- Video Monitor (800 × 600 or higher resolution) with at least 256 colours (1024x768 High colour 16-bit recommended).

3.3 Functional Requirements

- The system shall detect and scan one user's body including head, upper and lower parts through smartphone camera.
- The system shall obtain and process human body skeleton joint positions information and body measurements.
- The system shall display garment collections for user to select.
- The system shall superimpose selected garment over user's body.
- The system shall allow user to change garment.
- The system shall allow user to take a picture of himself/herself with the garment.

3.4 Requirements

3.4.1. Developer side:

- IDE: Anaconda
- Programming Language: Python, HTML, CSS, BOOTSTRAP.
- Packages Used: Dlib (19.15.0), OpenCV (3.4.2.17), SciPy (1.0.0), Cascade trainer gui (1.8.0), Tkinter canvas(8.6.8), NumPy (1.18.1), Flask Web framework (1.1.1)
- Front-end Lagunages: HTML, CSS & BOOTSTRAP Data: Source (Internet)
- Back-end Language: Python (3.7.4).

3.4.2. User side:

- The user must identify the target human body's parts and transfer objects to the appropriate body parts.
- For body part detection we were used haar cascade dataset.
- The image's moving *24*24 target window, which includes characteristics like (line features, rectangular feature,edge feature etc.)
- Convolutional kernals and haar features are pertinent characteristics for object detection.

3.5 Programming language

- **Python**

  Python is a popular computer programming language used to create software and websites, automate processes, and analyses data. Python is a general-purpose, high-level, interpreted programming language. Code readability is prioritized in its design philosophy, which employs heavy indentation.

3.6. Packages used

- **OpenCV**

  OpenCV-Python is a library of Python bindings designed to solve computer vision problems. OpenCV array structures are converted to and from NumPy arrays. This also makes it easier to integrate with other libraries that use NumPy such as SciPy.

- **SciPy**

  An open source, BSD-licensed library for mathematics, science, and engineering, SciPy is a scientific library for Python. The NumPy library, which offers simple and quick N-dimensional array manipulation, is a prerequisite for the SciPy library.

- **NumPy**

  Large, multi-dimensional arrays and matrices are supported by NumPy, a library for the Python programming language, along with a substantial number of high-level mathematical operations that may be performed on these arrays.

- **Cascade trainer GUI**

  An application called Cascade Trainer GUI is used to train, test, and enhance cascade classifier models. It makes it simple to utilize OpenCV tools for training and testing classifiers by using a graphical interface to set the settings.

CHAPTER 4

DESIGN & ANALYSIS

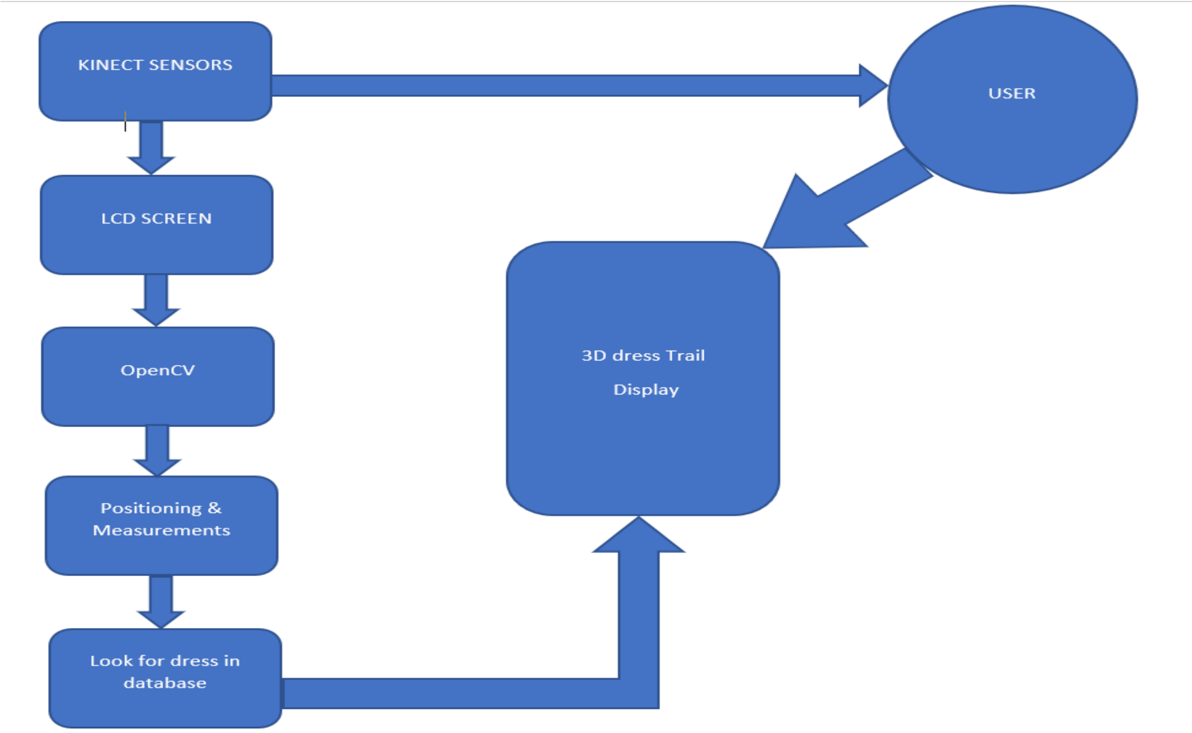


Fig 4.1: **System Architecture**

The working of the project goes like:

Step1: The user must stand in front of the device's computer screen or LED screen.

Step 2: The camera of the gadget will interact with the human body in a manner similar to how it identifies the human body's structure and, with the aid of specific alignments, enables the addition of a specific product to the user's body.

Step 3: Since it is a real-time project, OpenCV and its various modules will respond to the user's alignment and assign it to the user in a suitable manner.

Step 4: The NumPy framework will take into account how the user is situated and how big the human body is. While the user is testing a certain product, the results of the calculation will be displayed on the LED screen.

Step5: The source is where the project gets its database from (via. Internet).

Step 6: After the product's database-based calculations and alignment, it is now available for the user to test out.

Step 7: On the system's LED screen, the entire process is visible.

Step 8: By this, the project is Ready for its trial.

4.1 Methodology

Our goal is to provide a detailed concept of a real-time system that effortlessly tries on countless items of clothing without leaving the comfort of your home. In addition, people can also try to wear good looking dresses when they wanted to go from home to a party or other places. People use mirrors on a daily basis to see how they look and choose clothes to wear for the day before leaving home. Also, many mirrors are placed in clothing stores to help customers decide on clothes that fit and look good. In this sense, detailed concepts for a realtime dressing system can answer your questions about dressing as well as clothing sizing without the need for physical dressing and undressing. The need for virtual dressing in real time system are obvious. Firstly, benefits for customers are to save don and undressing time and easily estimate your body measurements for tailored dresses. Customers normally try on many things and spend a lot of time dressing and undressing to buy a dress. It is very inconvenient for them to take the dress they want to try on, go to the dressing room, take it off and put it back on whenever they find an attractive dress. Second, store owners can save costs because they no longer need changing rooms. In addition, the waste of clothes tried on by customers will be reduced.

4.2 Use case diagram

The interaction between elements is graphically shown in the use case diagram. A use case is a series of procedures or events that define the interactions between a role and a system in order to accomplish a particular objective. The major components and operations that make up the system are shown on the use case diagram. The procedures are referred to as "use cases," and the key components are referred to as "actors."

An actor graph, a collection of use cases bounded by a system boundary, and a relationship between communication between actors and use cases make up a use case diagram. Each use case illustrates a feature that the system offers users, and it defines how the system communicates with external actors. The stick figure that represents the actor and his name is included in the use case diagram along with ellipses that include the name of the use case. There is a rectangle enclosing all use cases. Figure 4.2 and Figure 4.3 shows the use case diagram for Admin and the User for the Virtual styling room respectively. The admin can add and remove clothes. The user can select, try and also buy the clothes.

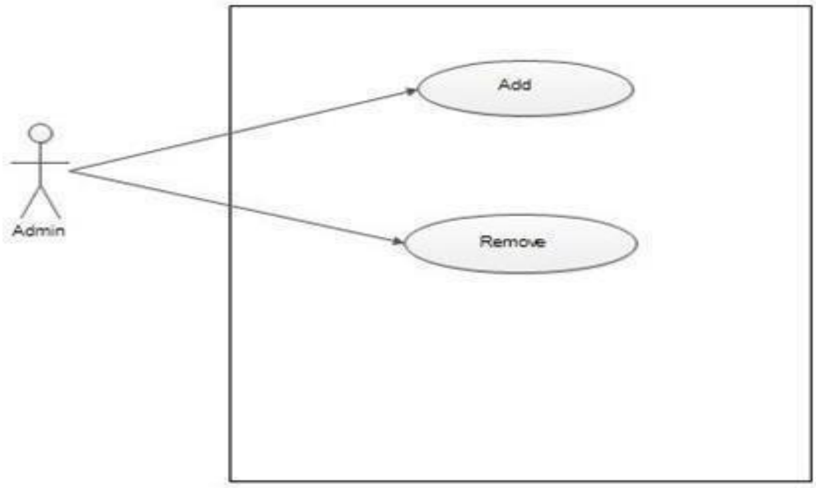


Fig 4.2: Use Case Diagram for Admin

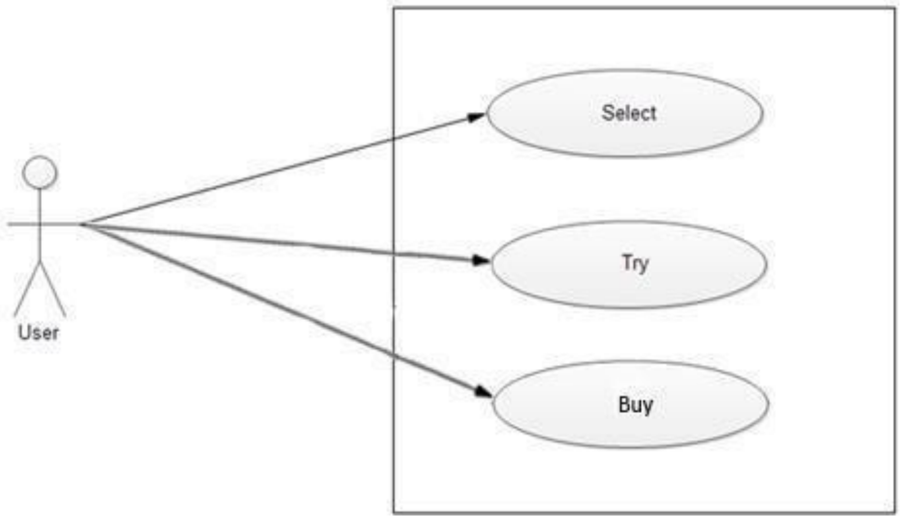


Fig 4.3: Use Case Diagram for User

4.3 Control Flow Diagram

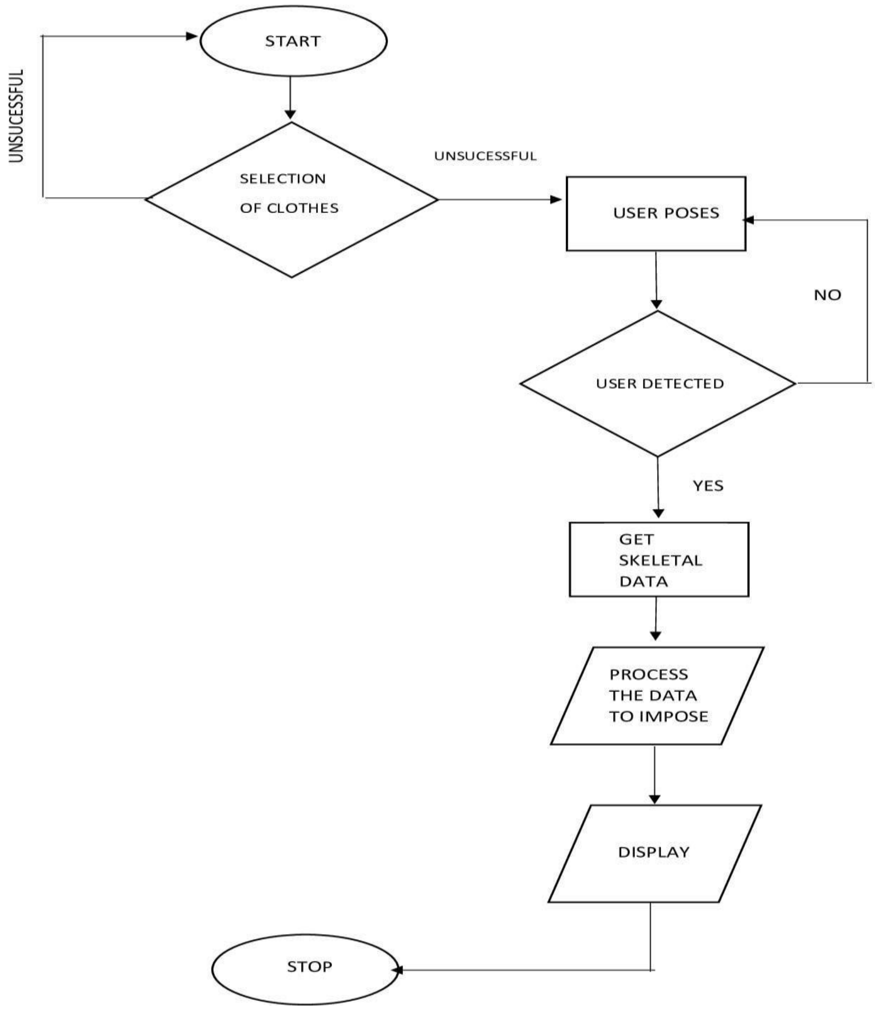


Fig 4.4: Control Flow Diagram

The main steps in a systematic process are depicted graphically and sequentially in a system flow diagram. A system flow diagram demonstrates the types of data that will be input into and exported from the system, as well as their origin, destination, and storage locations. The control flow diagram for the virtual style room with live video feed is shown in Figure 4.4.

The image you've shared illustrates a Data Flow Diagram (DFD) for a Real-Time Virtual Dressing System, which outlines the sequence of steps from the start of the application to the display of the virtual try-on result. Each block and arrow in the diagram represents a critical stage in the virtual dressing pipeline.

- **Start**

The process begins with the "Start" node, which initializes the virtual dressing system. At this point, the user has launched the application or activated the system on a device like a smartphone, tablet, or interactive kiosk. The system is now ready to accept user input and initiate the try-on process.

- **Selection of Clothes**

The first meaningful action required from the user is the "Selection of Clothes." This stage is essential because the entire virtual try-on process depends on the garment selected by the user. The user browses through a catalog or wardrobe database and picks a clothing item (e.g., a shirt, dress, or pants) they wish to try on. The selected item becomes the reference model to be virtually imposed on the user's body later in the process.

If the user fails to make a selection or the system fails to retrieve the selected clothing item (perhaps due to a backend error), the process is marked as "Unsuccessful", and the application loops back to the start, prompting the user to retry or resolve the issue.

- **User Poses**

Once a clothing item is selected, the system prompts the user to pose in front of the camera. This is a crucial step as the virtual dressing system needs a clear visual feed of the user to carry out subsequent image processing tasks. The user's pose helps the system capture body posture, scale, and relative positioning of limbs, which are all vital for accurate garment mapping.

The system constantly checks whether the user is posing correctly and visibly. If not, it loops the process and keeps prompting the user until an acceptable pose is detected.

- **User Detected**

After capturing a video frame or image of the user, the next decision point in the diagram is "User Detected." This decision box checks whether the system has successfully identified the presence of a human figure in the video stream. Technologies such as OpenCV, MediaPipe, or other body detection algorithms might be used here to recognize and confirm a human silhouette or skeleton. If the user is not detected (due to poor lighting, obstruction, or improper camera alignment), the process cycles back to the "User Poses" step until a proper detection is achieved.

- **Get Skeletal Data**

Upon successful user detection, the system moves to the "Get Skeletal Data" block. Here, the system captures critical points of the user's body using pose estimation algorithms. These algorithms generate a digital skeletal structure based on joint detection (e.g., shoulders, elbows, hips, knees), which enables the software to map clothing realistically.

The skeletal data helps determine:

- The size of the body
- The orientation and posture of limbs
- Real-time movement (for dynamic adjustments)
- Proportions for scaling the clothes

This skeletal framework forms the basis on which virtual garments will be adjusted and displayed.

- **Process the Data to Impose**

With the skeletal data in hand, the system proceeds to process this data to impose the selected garment onto the user's image. This involves several complex tasks such as:

- Scaling the garment to match the user's proportions
- Warping or bending the clothing to fit the pose and orientation
- Layering the clothing image over the user's body without distorting the look
- Color and shadow correction to make the garment appear realistic and naturally blended with the user's live video feed Advanced image processing and deep learning techniques may be employed here, including Homography, Geometric Matching Modules (GMM), and Texture Mapping.
- The processing ensures that the clothing not only fits but also follows the user's motion, simulating a lifelike dressing experience.

- **Display**

Once the processing is complete, the result is rendered in the "Display" step. The final augmented image or live video feed is shown on the device's screen, displaying the user wearing the selected garment virtually.

This output provides a real-time or near-real-time preview of how the clothing looks and fits on the user. At this point, users can:

- View the garment from multiple angles
- Try different poses to test garment behavior
- Change the clothing selection and repeat the process

  The display step is the culmination of all the previous processes and provides the interactive, visual result than the user desires.

- **Stop**

Finally, the process ends at the "Stop" node. Upon successful user detection, the system moves to the "Get Skeletal Data" block. Here, the system captures critical points of the user's body using pose estimation algorithms. These algorithms generate a digital skeletal structure based on joint detection (e.g., shoulders, elbows, hips, knees), which enables the software to map clothing realistically The user may choose to stop the application, close the virtual dressing room, or proceed to take additional actions such as:

- Saving the image
- Sharing the look on social media
- Adding the item to a shopping cart
- Requesting size recommendations or similar items

The virtual dressing system begins with the Start phase, where the application is launched on a device such as a smartphone, tablet, or interactive kiosk. Once initialized, the system waits for the user to engage. The first critical step is the Selection of Clothes, where the user browses a virtual catalog and chooses a garment they wish to try on. If a selection is not made or if the system encounters an error retrieving the item, the process is marked as unsuccessful and loops back to the start. After a garment is selected, the user is prompted to pose in front of the camera during the User Poses stage. This step ensures that the system captures a clear image of the user's posture and positioning. The next step is User Detected, where body detection algorithms confirm the presence of a human figure in the camera frame. If detection fails—due to poor lighting, incorrect alignment, or obstructions—the process loops back to the posing stage. Once the user is detected, the system proceeds to Get Skeletal Data, where it uses pose estimation techniques to map key body joints and create a digital skeletal structure.

4.4 Activity Diagram

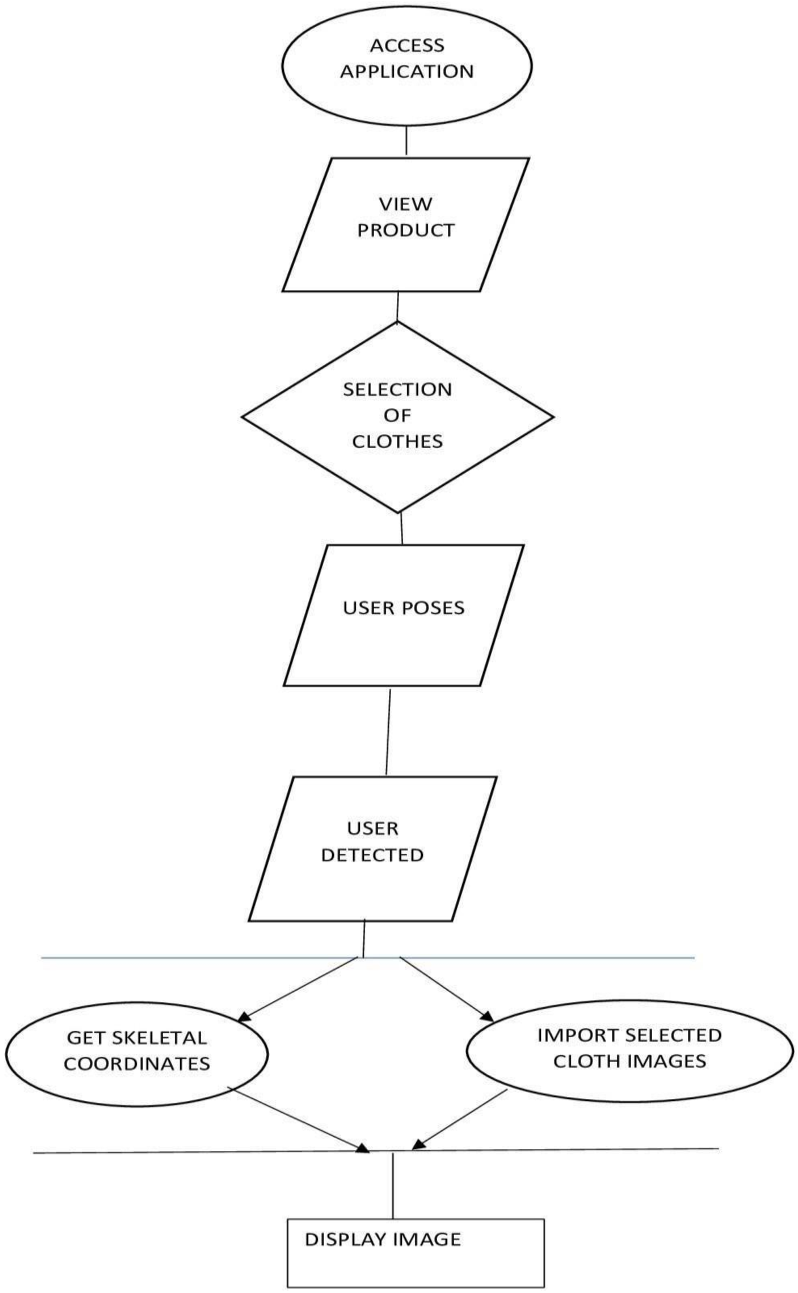


Fig 4.5: Activity flow diagram

Another crucial diagram in the UML used to depict the system's dynamic elements is the activity diagram. An activity diagram is a flow graphic that shows how one activity leads to another. The action might be referred to as a system operation. One operation leads to the next in the control flow. This flow may be concurrent, branching, or sequential. By utilizing various parts like fork, join, etc., activity diagrams deal with all sorts of flow control.

Activity Diagrams can help you create a use case by outlining the actions that must be taken and when they must happen. It describes modelling applications with simultaneous processes and difficult sequential algorithms. Activity diagrams shouldn't go into specifics about how the objects interact or behave. The Activity Diagram for the Virtual Styling Room Using Live Video Feed is shown in Figure 4.5

4.5 Sequence diagram

A sequence diagram is an interaction diagram that shows how processes operate with one another and shows the order of process operation. It depicts the participants in an interaction and the sequence of messages among them. It also shows the interaction of the system with its actors to perform a use case. A sequence diagram shows the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with the use case realization in the Logical View of the system under development.

In the sequence diagram, each actor as well as system is represented by a vertical line called the life line and each message by a horizontal arrow from sender to receiver. The parallel vertical lines show different objects that live simultaneously. Time proceeds from top to bottom. This step ensures that the system captures a clear image of the user's posture and positioning. The next step is User Detected, where body detection algorithms confirm the presence of a human figure in the camera frame. If detection fails—due to poor lighting, incorrect alignment, or obstructions—the process loops back to the posing stage. Once the user is detected, the system proceeds to Get Skeletal Data, where it uses pose estimation techniques to map key body joints and create a digital skeletal structure.

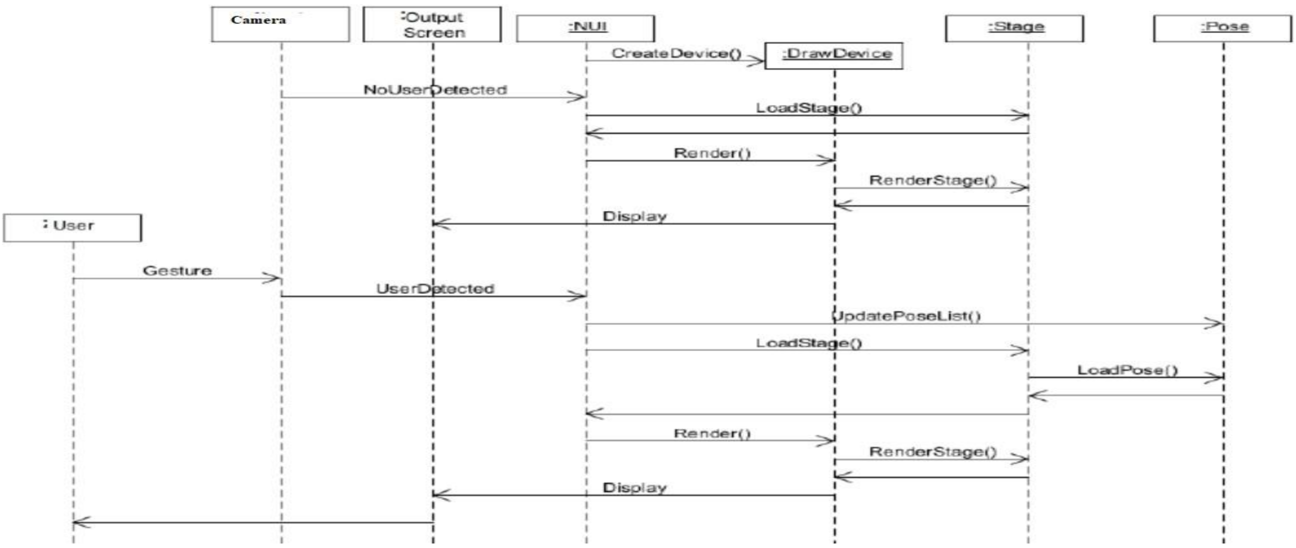


Figure 4.6: shows the sequence diagram for Virtual Mirror.

4.6 Class diagram

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

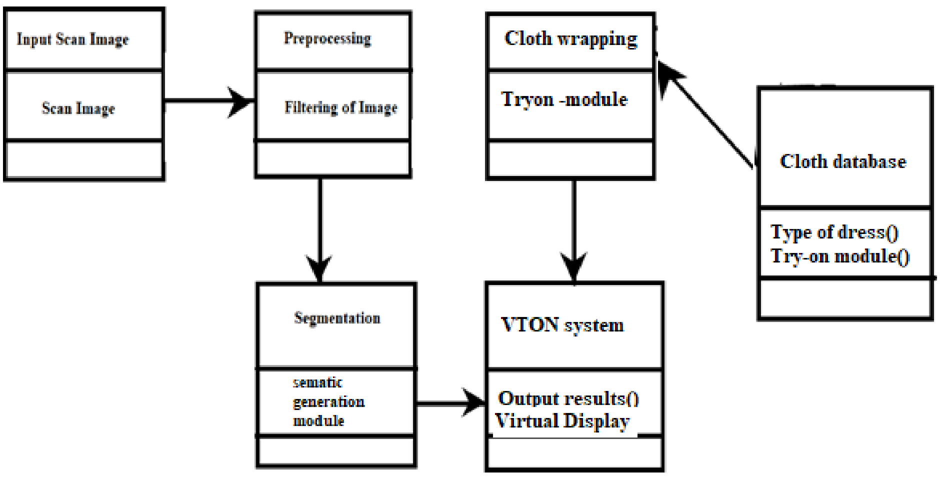


Fig 4.7: Class diagram

CHAPTER 5

IMPLEMENTATION

A technical specification or method is implemented when it is made into a programme, piece of software, or other type of computer system. The goal of this phase is to implement the system's design as effectively as possible by translating the concept into code. In the life cycle of a system, implementation is critical. It is a phase where the design is turned into a functional module.

The crucial and concluding stage of software development is implementation. It speaks of the transformation of a novel system design into a function. Implementation results in robust, reusable, and expandable code. The process of guiding a customer from purchase to the hardware or software that was purchased is known as implementation. This covers user regulations, user training, system integration, customization, scope analysis, and delivery.

5.1 Pseudocode

Pseudocode is a colloquial term for a high-level, informal description of how an algorithm or computer programme works. Although it follows standard programming language structural conventions, it is written for human rather than machine reading. It is used to develop a program's rough draught or blueprint. Pseudocode condenses a program's flow but omits supporting information. To make sure that programmers comprehend the specifications of a software project and align their code correctly, system designers create pseudocode. The model implemented has been done with a focus on tops, with complete apparel transfer being potential future work.

5.1.1 The Segmentation Algorithm

For this, we initially needed to be able to implement a segmentation algorithm. Even though open- source state-of-the-art models could have been used to implement this, we stuck with robust image processing techniques for segmentation. With the idea being to localize the face and understand the skin color of the model from the face image to be able to divide an image into hair, clothes, skin, and background.

5.1.2 Geometric Matching Module

Once we have the clothing segment, we can now geometrically compare this clothing segment to the in-shop clothing. Our goal is now to be able to learn transforms on the in-shop clothing to make it as geometrically similar to the model clothing. To visually describe this one can refer to the image below. The example is a grid of six images. Top left is the in-shop clothes, top right being the clothing segment of the model, the top middle being the transform (bottom left) applied on the in-shop clothes.

Fig 5.1: **Geometric Matching**

The above examples are generated during training and hence the in shop clothing and the model clothing are the same. This can also lead to an easier qualitative assessment. The network architecture to learn this transform is briefly described below.

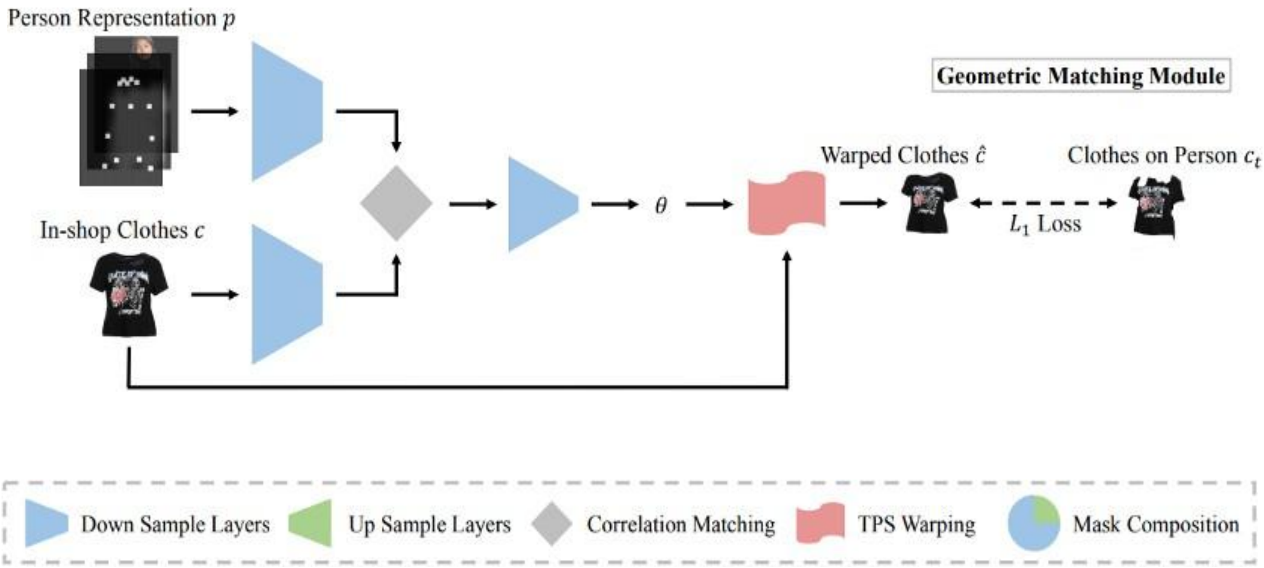


Fig5.2: **Geometric Matching Module**

We call the learning of this transformation as the Geometric Matching Module, as it matches the in-shop clothing to the current clothing trying to get them to match geometrically. Some of the results after training are shown below.

5.1.3 Try-on Module

The instinctive approach to imposing the new clothing now is to simply paste it over the image, but as one can see this will cause problems due to overlap with hair and hands, and the previous clothing stays, making it look very unrealistic. The solution to this was the try-on module, where we implement an encoder-decoder network to smoothen out the image.

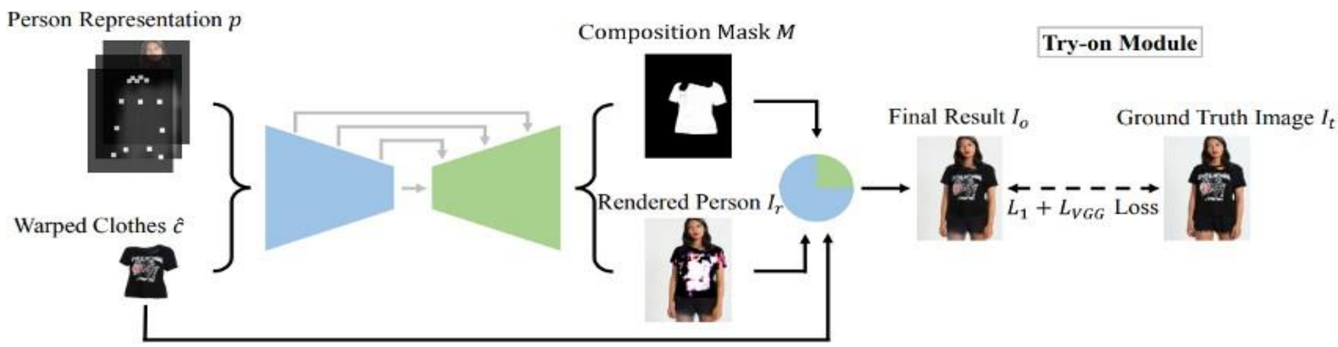


Fig 5.3: Try-on Module

This gives a smoothened image that looks much more realistic then the results we would have if we were to paste the image over the model. The article has avoided any in-depth description of the work done and for a thorough description of the model and training strategy. The image illustrates a Try-on Module, which is a critical component of a virtual try-on pipeline that follows the Geometric Matching Module. This module is designed to synthesize a photorealistic image of a person wearing a given item of clothing. It combines the information from the warped clothes (c^\hat{c}c^) generated earlier and the person representation (ppp) to produce the final output. The Try-on Module handles challenges like blending clothing textures with human body features, preserving occlusions like hair or arms, and ensuring the final output appears natural.

The process starts by inputting both the warped clothing and the person representation into an encoder decoder network, typically a U-Net architecture. This structure enables the model to retain high-resolution spatial details from the input through skip connections between encoding and decoding layers. These connections are essential for preserving person-specific features such as facial details, pose, and body shape while integrating the warped clothing.

CHAPTER 6
SYSTEM ANALYSIS

Problem statement

Designing a solution for reducing human time and making their online shopping better by designing a virtual styling room using live video feed it also Provides a virtual room to try apparel through e- commerce websites before buying it.

6.1 Existing system

The present virtual changing rooms are useful for clothing items for individual body parts such the head, feet, arms, and face. The current approach requires us to try on each and every item of clothing, which most people find difficult or inconvenient to accomplish. Existing efforts that use images to test have issues with the model's stability when processing pictures of individuals taken in various lighting situations, different environmental settings, and unique stances. Many e- commerce companies, including Lenskart, Purple, and others, use AI fitting rooms to produce results relevant to spectacles or makeup.

6.2 Current Applications and Scope

Most existing virtual changing room systems focus on clothing items that correspond to individual body parts such as the head (hats, scarves), feet (shoes, socks), arms (gloves, sleeves), and the face (makeup, spectacles). These segmented approaches are often simpler to implement because they require overlaying virtual objects onto a relatively limited area of the user’s body. For instance, AR try-ons for eyewear or makeup typically involve detecting facial landmarks and applying virtual frames or cosmetics These segmented approaches are often simpler to implement because they require overlaying virtual objects onto a relatively in real-time. Similarly, shoe or glove try-on systems use foot or hand tracking to visualize the item in a natural way.

6.3 Challenges and Limitations of Existing Models

Despite the impressive capabilities, current virtual changing room models have notable limitations that affect their usability and acceptance. One major issue is the requirement for users

to try on each clothing item individually, which can be time-consuming and inconvenient. Because the systems typically simulate
one item or one category of apparel at a time, customers must repeat the process for every shirt, jacket, or pair of pants they are interested in. This can lead to frustration, especially when browsing large catalogs.
Another significant challenge is the instability and inconsistency of models when processing images taken under diverse conditions. The quality and realism of virtual try-ons depend heavily on accuratesegmentation and fitting of garments onto the user's body. However, lighting variations, environmental backgrounds, and unique body poses can negatively impact the model's ability to correctly interpret the image. For example, shadows, reflections, or complex backgrounds may confuse the segmentation algorithms, leading to poor garment alignment or visual artifacts. This problem is further amplified when users upload photos taken from different angles or with varying resolutions, resulting in an inconsistent experience.

6.4 Disadvantages

- Due to the pandemic it causes second thoughts before trying on an outfit.
- On online websites we can only observe how the garments look on others.
- The size & colour variance is not always seen clearly in the posted photos.
- People always have problem in choosing colour which looks good on them.
- Buying products without trials may lead to return & exchanges.
- Lack of human touch and emotional connection.
- High initial costs and implementation challenges.
- Interpretability and transparency issues.
- Difficulty with complex or ambiguous requests.

6.5 Proposed system:

This project's primary goal is to improve and simplify the online buying experience for users. In order to save time, it seeks to develop an "Augmented Reality" fitting room. gives customers a way to try on different clothing without actually touching it before making a purchase. decreases the necessity for manual or physical clothing putting on, which also lowers the chance of contracting covid. It enables consumers to make wiser decisions. The project's primary goal is to create a genuine connection between the user and virtual clothing.

Fig 6.1 Virtual Trial Room

6.6 Advantages:

- Allows people to complete their tasks in a smooth & timely manner
- Gives the shoppers access to various more options to try & check.
- Convenient and faster shopping experience.
- Quicker and Easier to try clothes.
- Reduces Exchange and Return Polices.
- No hidden camera issues as for female customers.
- By providing your customers a way to see how each item looks on their body, they can make more informed decisions. Therefore, fewer of them will want to return their orders.
- Drawing customers in.
- Notably greater turnover. Positive image of your brand.

CHAPTER 7

SOURCE CODE

App.py

```
import sqlite3
import random
import os import
csv import
base64

from PIL import Image from
io import BytesIO
from flask import Flask, render_template, request, send_from_directory, session, url_for,
redirect

from src.utils.all_utils import read_yaml, create_directory
import os
from PIL import Image

import numpy as np
import pickle
import tensorflow as tf

from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import GlobalMaxPooling2D
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input

from sklearn.neighbors import NearestNeighbors
from numpy.linalg import norm

app = Flask(_name_)

app.secret_key = '\xf0?a\x9a\\\xff\xd4;\x0c\xcbHi'
```

```
def base64_to_image(base64_data): base64_data =
base64_data.split(",")[-1] image_data =

base64.b64decode(base64_data) image_stream =
BytesIO(image_data)

image = Image.open(image_stream) return
image

connection = sqlite3.connect('user_data.db')
cursor = connection.cursor()

command = """CREATE TABLE IF NOT EXISTS user(name TEXT, password TEXT, mobile
TEXT, email TEXT)"""
cursor.execute(command)

def total_iems():
return len(os.listdir("static/collections"))
config = read_yaml('config/config.yaml')



params = read_yaml('params.yaml')
artifacts = config['artifacts'] artifacts_dir
= artifacts['artifacts_dir']

#upload
upload_image_dir = artifacts['upload_image_dir'] uploadn_path
= os.path.join(artifacts_dir, upload_image_dir)

# pickle_format_data_dir
pickle_format_data_dir = artifacts['pickle_format_data_dir']
img_pickle_file_name = artifacts['img_pickle_file_name']
raw_local_dir_path = os.path.join(artifacts_dir, pickle_format_data_dir)
```

```
pickle_file = os.path.join(raw_local_dir_path, img_pickle_file_name)

#Feature path
feature_extraction_dir = artifacts['feature_extraction_dir']
extracted_features_name = artifacts['extracted_features_name']

feature_extraction_path = os.path.join(artifacts_dir, feature_extraction_dir)
features_name = os.path.join(feature_extraction_path, extracted_features_name)

#params_path
weight = params['base']['weights']
include_tops = params['base']['include_top']

#loading
feature_list = np.array(pickle.load(open(features_name,'rb')))
filenames = pickle.load(open(pickle_file,'rb'))

#model
model = ResNet50(weights= weight,include_top=include_tops,input_shape=(224,224,3))
model.trainable = False

model = tf.keras.Sequential([
model, GlobalMaxPooling2D()
])

defsave_uploaded_file(uploaded_file): try:
create_directory(dirs=[uploadn_path])

with open(os.path.join(uploadn_path,uploaded_file.name),'wb') as f:

f.write(uploaded_file.getbuffer())

return 1 except:
```

```
return 0

def feature_extraction(img_path,model):
img = image.load_img(img_path, target_size=(224, 224))

img_array = image.img_to_array(img) expanded_img_array =
np.expand_dims(img_array, axis=0) preprocessed_img =
preprocess_input(expanded_img_array)

result = model.predict(preprocessed_img).flatten()
normalized_result = result / norm(result)
return normalized_result

def recommend(features,feature_list):
neighbors = NearestNeighbors(n_neighbors=6, algorithm='brute', metric='euclidean')
neighbors.fit(feature_list)

distances, indices = neighbors.kneighbors([features])
return indices

@app.route('/') def
index():
return render_template('index.html')
@app.route('/home')

def home():
items = os.listdir('static/data') files =
random.sample(items, 12)

return render_template('home.html', ti = total_iems(), files=files)
@app.route('/recommendation') def

recommendation():
```

```
return render_template('recommend.html', ti = total_iems())

@app.route('/trial/<img>') def
trial(img):
session['filename'] = 'static/data/'+img
return render_template('trialroom.html', ti = total_iems())

@app.route('/cart') def
cart():
List = [] prices
= []

for im in os.listdir("static/collections"): f =
open("prices.csv", "r")
reader = csv.reader(f)

File = im.split('.')[0] for i in
reader:
if File in i: prices.append(i[1])
List.append("http://127.0.0.1:5000/static/collections/"+im)

return render_template('cart.html', ti = total_iems(), List = List, prices = prices, n = len(List))
@app.route('/userlog', methods=['GET', 'POST'])

def userlog():
if request.method == 'POST':
connection = sqlite3.connect('user_data.db') cursor =
connection.cursor()

name = request.form['name'] password =
request.form['password']

query = "SELECT name, password FROM user WHERE name = '"+name+"' AND password=
```

```
'"+password+"'

cursor.execute(query)
result = cursor.fetchall()

if len(result) == 0:
return render_template('index.html', msg='Sorry, Incorrect Credentials Provided, Try
Again')
else:

return redirect(url_for('home'))
return render_template('index.html')

@app.route('/userreg', methods=['GET', 'POST'])

def userreg():
if request.method == 'POST':

connection = sqlite3.connect('user_data.db') cursor =
connection.cursor()

name = request.form['name'] password =
request.form['password'] mobile =
request.form['phone']

email = request.form['email']
print(name, mobile, email, password)

command = """CREATE TABLE IF NOT EXISTS user(name TEXT, password TEXT, mobile
TEXT, email TEXT)"""


cursor.execute(command)
```

```
cursor.execute("INSERT INTO user VALUES ('"+name+"', '"+password+"', '"+mobile+"',
'"+email+"')")
connection.commit()

return render_template('index.html', msg='Successfully Registered')
return render_template('index.html')

@app.route('/imagetest', methods=['GET', 'POST']) def
imagetest():

if request.method == 'POST': fileName =
session['filename'] fileName1 =
request.form['img2']

File = fileName.split('.')[0]
File = File.replace('static/data/', '') fileName1 =
"static/test_img/"+fileName1

os.system(f"python detection.py --input_image {fileName1} --input_cloth {fileName}")
f = open("prices.csv", "r") reader
= csv.reader(f)

for i in reader: if File in
i:
name = i[1] import
random

random_range = random.randint(80, 90)
return render_template('trialroom.html', ti = total_iems(), dress=fileName, price=name,

image=fileName1,

output="static/result/"+os.listdir("static/result")[0],random_range=random_range)
```

```
return render_template('recommend.html' , ti = total_iems())

@app.route('/livetest', methods=['GET', 'POST'])

def livetest():
if request.method == 'POST': fileName =
session['filename'] filedata =
request.form['img2']

dlist = os.listdir('static/testpicture') for
item in dlist:
os.remove("static/testpicture/"+item)

name1 = str(random.randint(1000, 9999)) result_image =
base64_to_image(filedata)

result_image.save('static/testpicture/'+name1+'.png')

File = fileName.split('.')[0]
File = File.replace('static/data/', '') fileName =
fileName

fileName1 = 'static/testpicture/'+name1+'.png'
os.system(f"python detection.py --input_image {fileName1} --input_cloth {fileName}")
f = open("prices.csv", "r")
reader = csv.reader(f) for i in
reader:
if File in i: name = i[1]

return render_template('trialroom.html', ti = total_iems(), dress=fileName, price=name,
image=fileName1,

output="static/result/"+os.listdir("static/result")[0])
```

```
return render_template('recommend.html', ti = total_iems())

##@app.route('/uploads/<filename>')
##def uploaded_file(filename):
##
return send_from_directory(uploadn_path, filename)

@app.route('/choose', methods=['POST'])

def choose():
Type = request.form['Type']
session['type'] = Type

if Type == 'small': images
= [] names = []
for img in os.listdir('static/artifacts/SMALL'):
images.append('static/artifacts/SMALL/'+img)

names.append(img)
return render_template('recommend.html', names=names, images=images, Type=Type,
n=len(images))

if Type == 'medium':
images = [] names = []

for img in os.listdir('static/artifacts/MEDIUM'):
images.append('static/artifacts/MEDIUM/'+img)

names.append(img)
return render_template('recommend.html', names=names, images=images, Type=Type,
n=len(images))

if Type == 'large': images
```

```
= [] names = []

for img in os.listdir('static/artifacts/LARGE'):
images.append('static/artifacts/LARGE/'+img)
names.append(img)

return render_template('recommend.html', names=names, images=images, Type=Type,
n=len(images))

if Type == 'x-large': images
= [] names = []

for img in os.listdir('static/artifacts/EXTRA_LARGE'):
images.append('static/artifacts/EXTRA_LARGE/'+img) names.append(img)

return render_template('recommend.html', names=names, images=images, Type=Type,
n=len(images))
return render_template('recommend.html')

@app.route('/upload/<name1>') def
upload(name1):
print(name1)

if session['type'] == 'small': filename =
'SMALL\\'+name1

if session['type'] == 'medium': filename =
'MEDIUM\\'+name1



if session['type'] == 'large': filename =
'LARGE\\'+name1
ifsession['type'] == 'x-large':
```

```
filename = 'EXTRA_LARGE\\'+name1

features = feature_extraction(os.path.join(artifacts_dir, filename), model)
indices = recommend(features, feature_list)
result = []

for i in filenames:
result.append(i.replace("data\\", ''))

return render_template('recommend.html', ti = total_iems(), filenames=result, indices=indices[0])
##

@app.route('/logout') def
logout():
return render_template('index.html')

if _name_ == "_main_": app.run(debug=True,
use_reloader=False)
```

CHAPTER 8

RESULTS

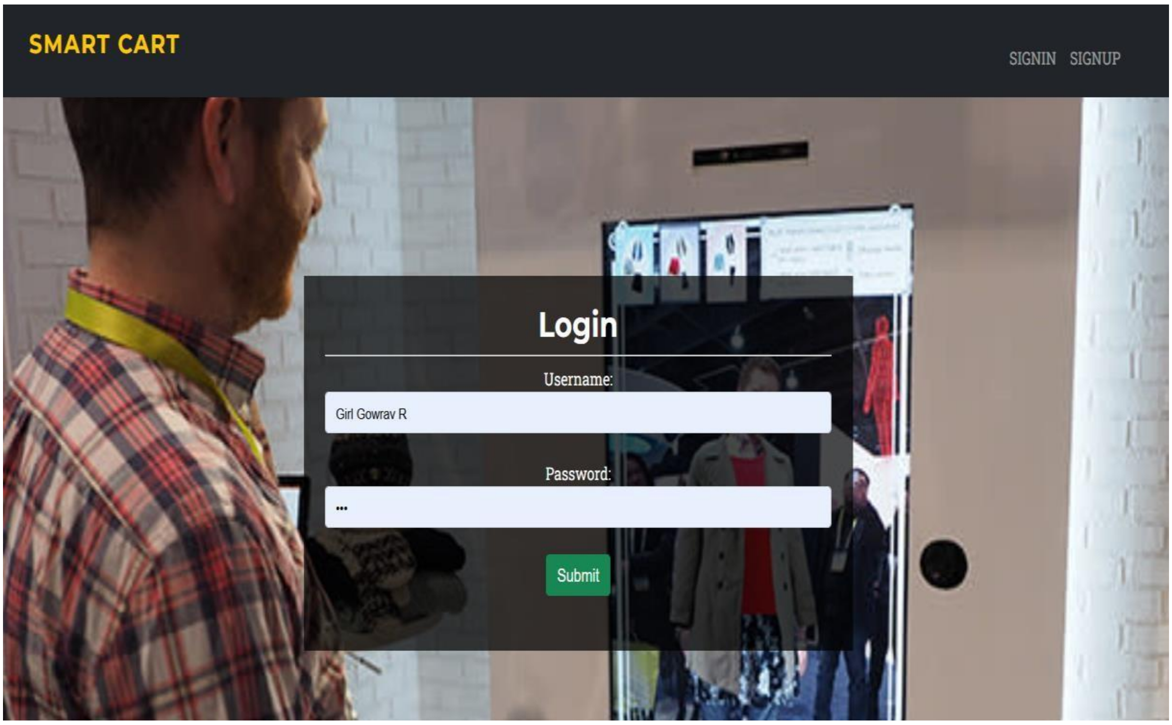


Fig 8.1: Home Page

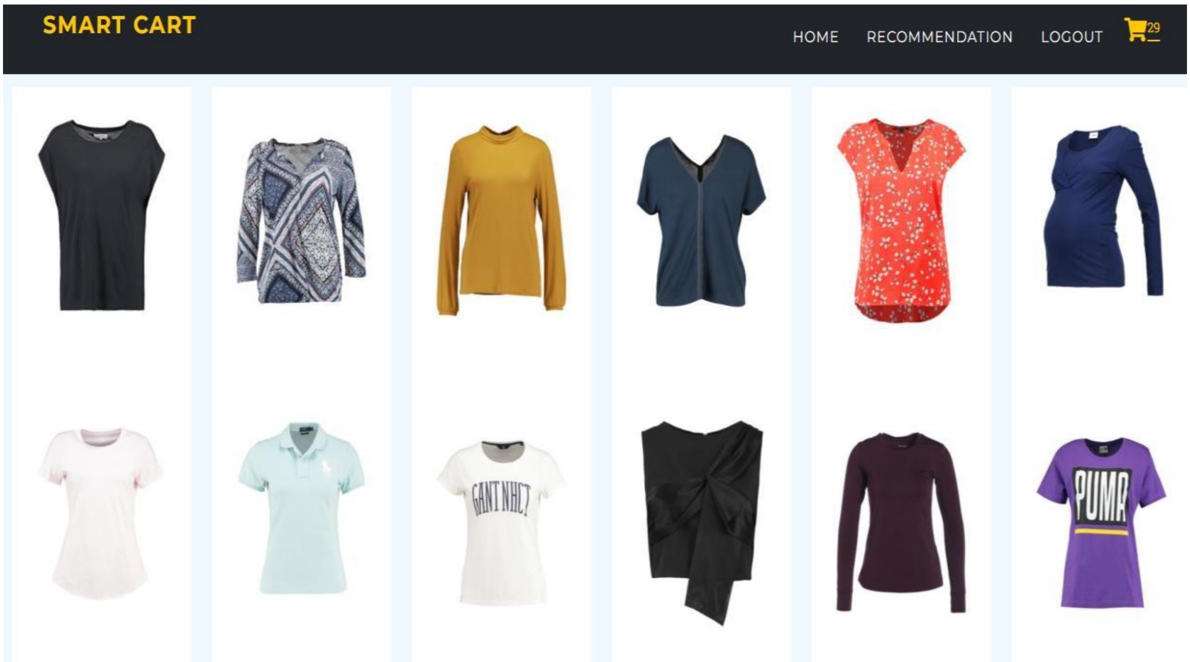


Fig 8.2 : Login Page

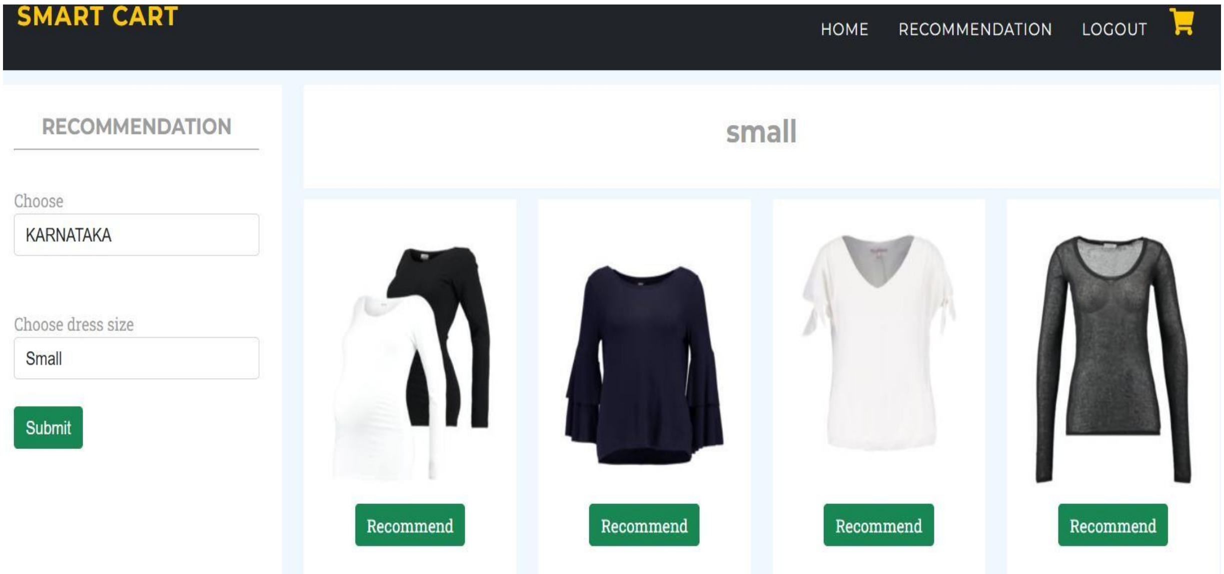


Fig 8.3 : Recommendation [State and Size]

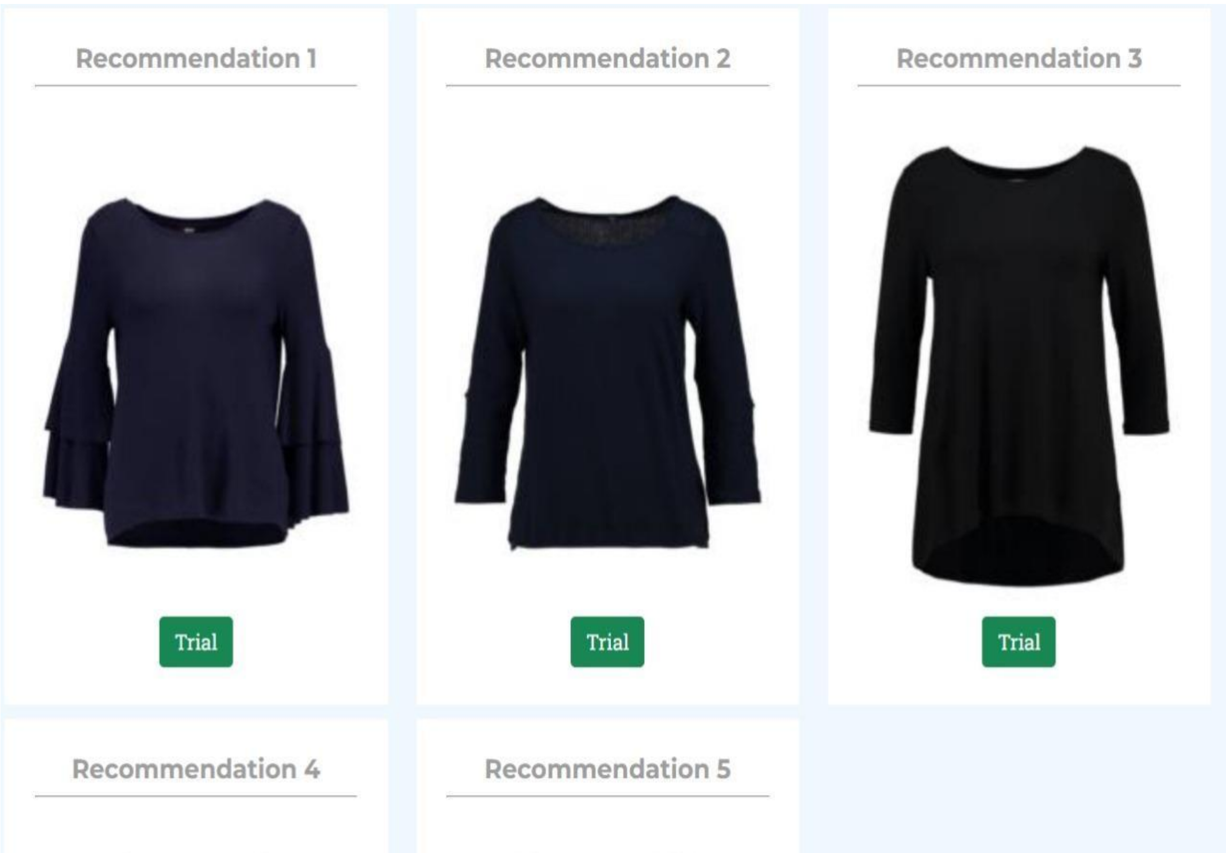
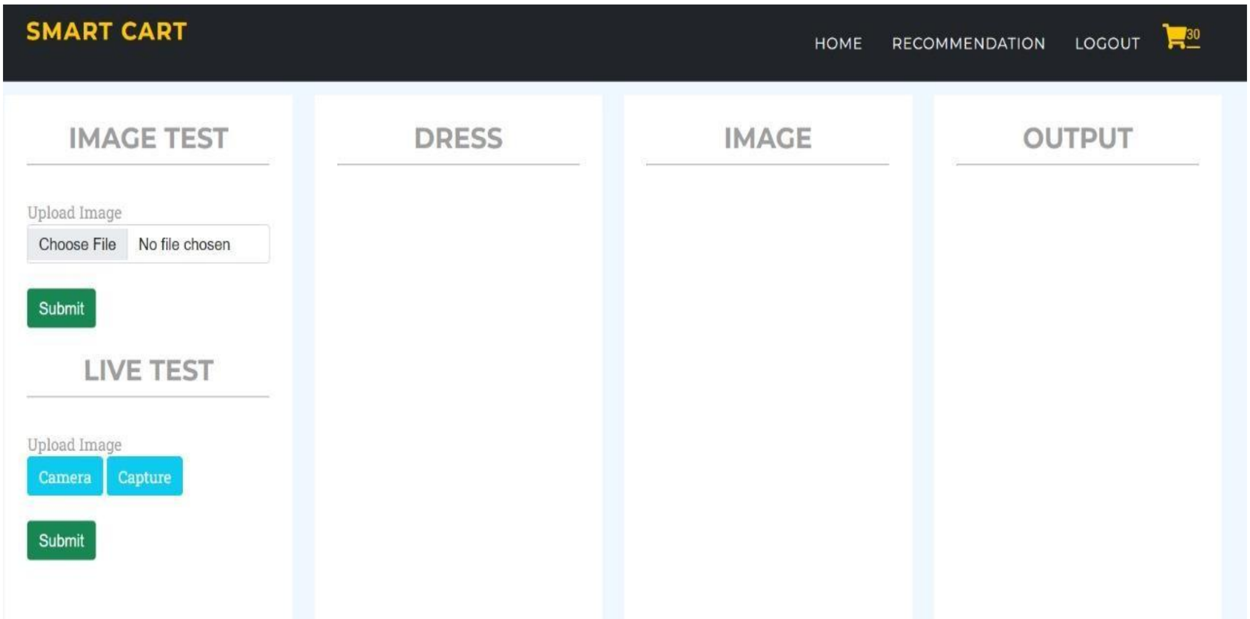


Fig 8.4 : Recommendation With Providing Trail Room



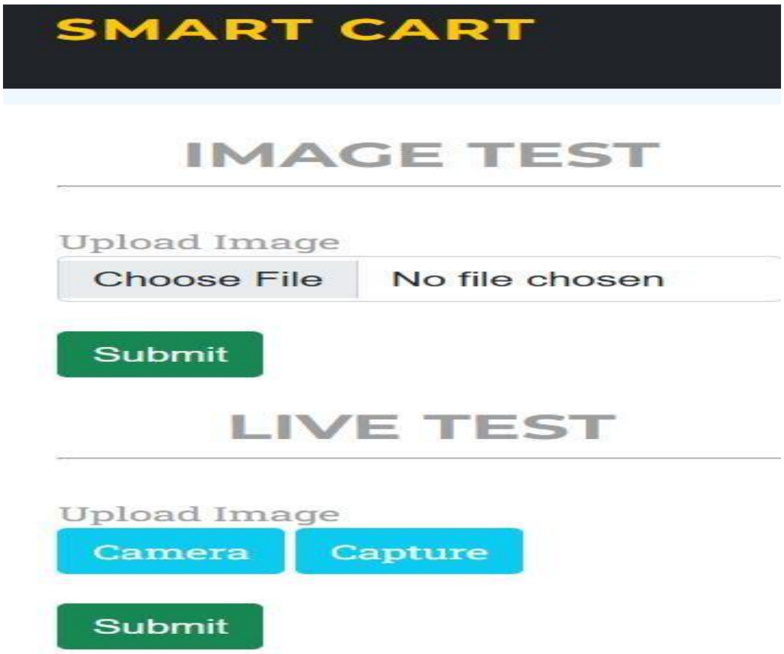


Fig 8.5: Trail Room By Uploading Image or Live Capture Of Person

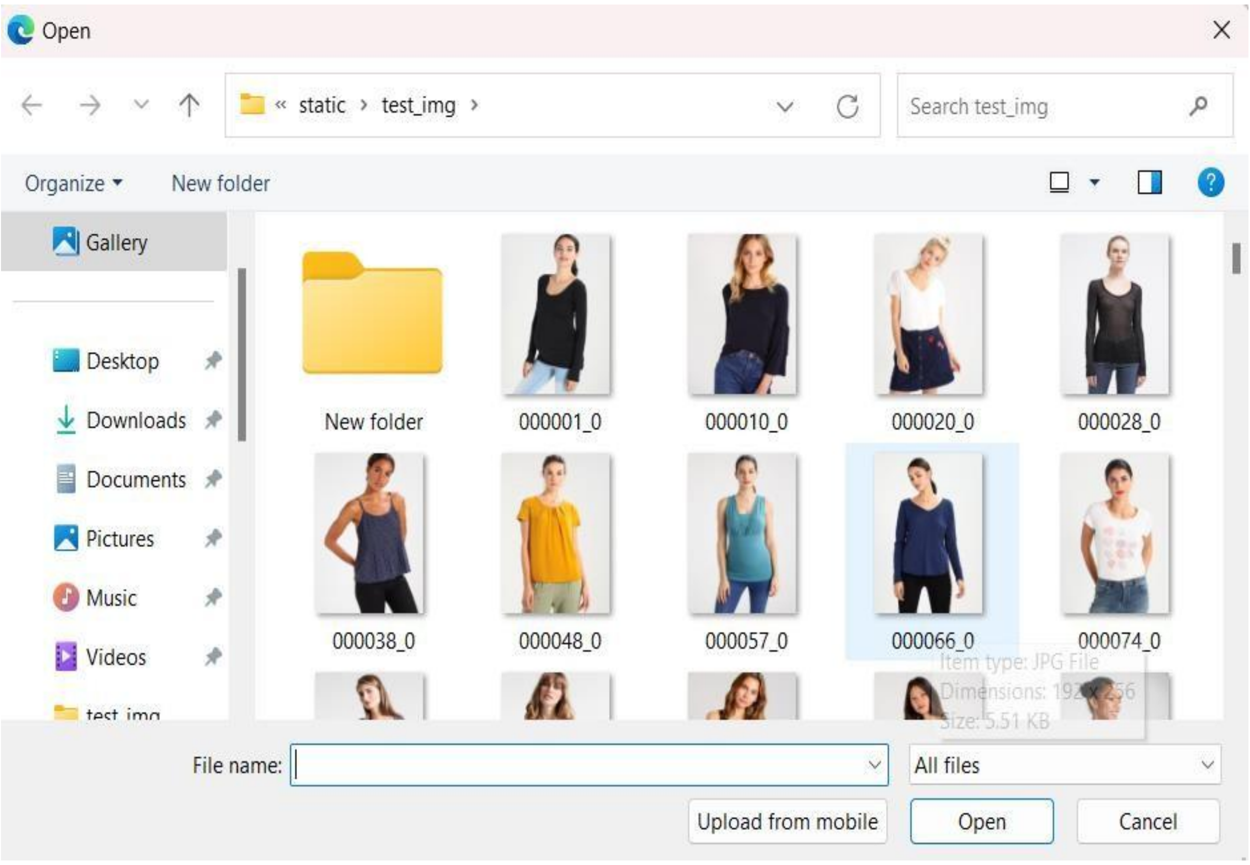


Fig 8.6 : Uploading An Image

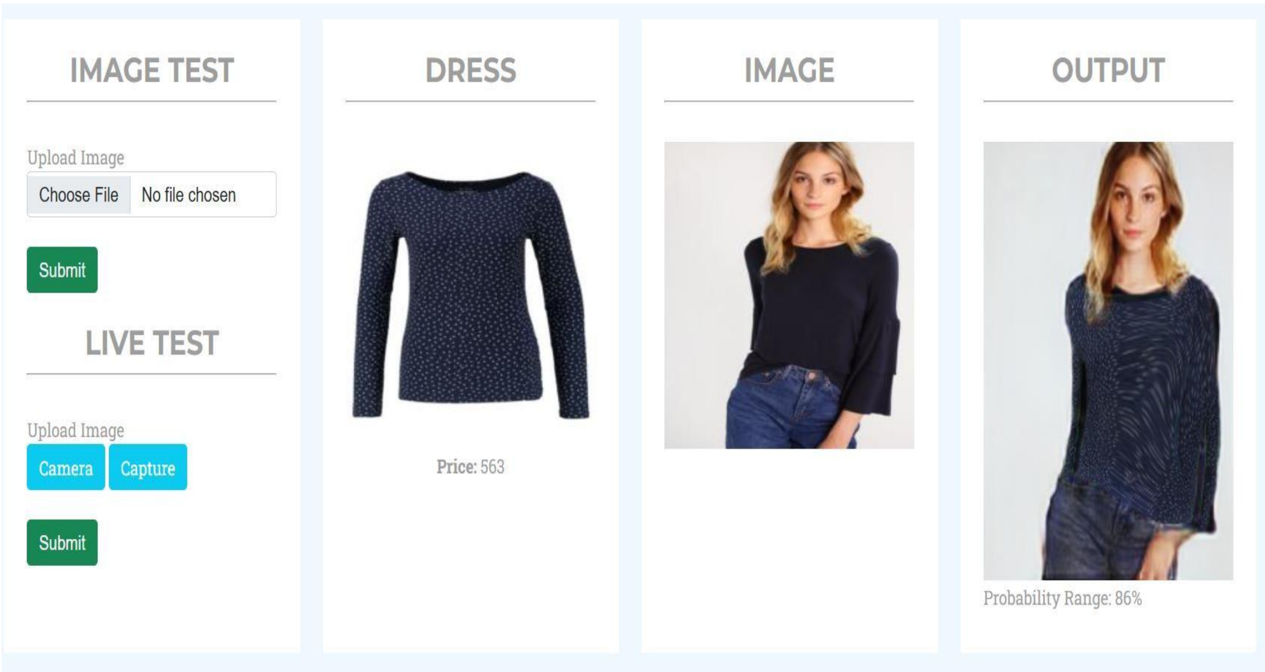


Fig 8.7: Output Window

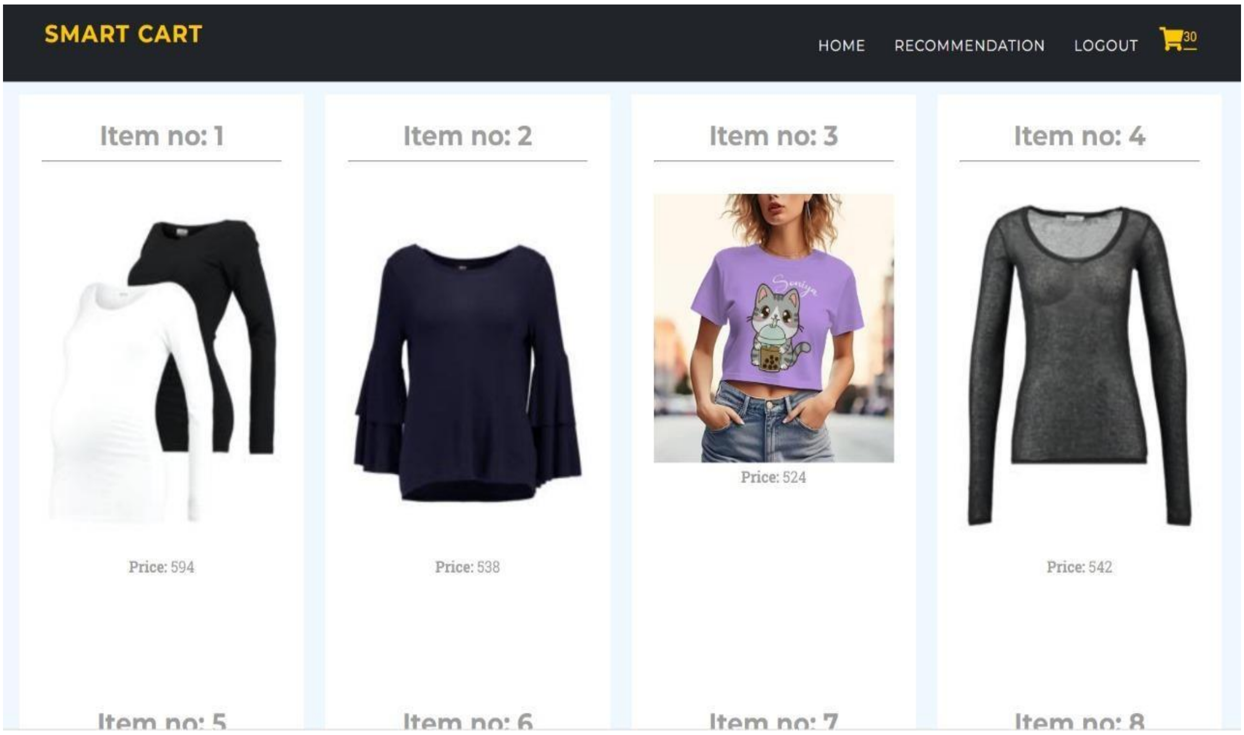


Fig 8.8: Cart For Items

CHAPTER 9

FUTURE ENHANCEMENT

The current virtual try-on application is designed primarily for customers, allowing them to try on garments in real time using a live video feed and body tracking technology. This enhances the shopping experience by eliminating the need for physical trial rooms and enabling users to explore various styles and fits conveniently. However, to extend the functionality and support the business aspect of retail, we are also developing a dedicated application specifically for store owners or shopkeepers. This retailer-focused module will function as an analytical dashboard, providing valuable insights such as the number of users who tried on virtual garments, how many of them proceeded to make a purchase, which clothing items are
the most popular, and recommendations on which inventory to retain or restock based on usage patterns. These insights will help retailers make data-driven decisions, manage inventory more effectively, and plan targeted marketing campaigns.

In addition to this, we plan to integrate a machine learning-based recommendation system into the customer application. This system will analyze previous user behavior, such as clothing choices and style preferences, to suggest new apparel that aligns with the user's taste. By incorporating techniques like collaborative filtering and content-based filtering, the model will continuously improve its recommendations over time. It may also consider contextual factors such as seasonality, occasion, or body type to deliver highly personalized suggestions. This feature not only improves the customer's shopping experience but also enhances user engagement and increases the chances of purchase. To ensure the realism and precision of the virtual try-on process, the application employs a deep learning architecture that includes three specialized neural networks: the Pose Alignment Network (PAN), the Texture Refinement Network (TRN), and the Fitting Network (FTN). PAN is responsible for aligning the selected garment with the user's body pose by mapping skeletal keypoints and transforming the garment accordingly. This ensures that the clothing conforms accurately to the posture and orientation of the user. Next, TRN enhances the visual details of the aligned garment, preserving elements such as textures, logos, and patterns while correcting any distortions introduced during alignment. This step is crucial for maintaining a high level of visual fidelity. Finally, FTN adjusts the fit of the garment by refining its scale and position based on the user's body contours, ensuring a natural and snug appearance in the final output. These three networks work in harmony to produce realistic, responsive, and visually appealing results.

In summary, the application not only offers a modern, efficient alternative to traditional fitting rooms but also serves as a powerful tool for retailers. By integrating real-time analytics, personalized recommendations, and advanced neural networks for pose alignment, texture refinement, and fitting, the system creates a comprehensive solution that benefits both shoppers and business owners. This dual purpose platform enhances user satisfaction, streamlines retail operations, and represents a forward thinking approach to digital fashion and commerce.To ensure realistic and high-quality virtual try-on results, especially after the system identifies key body sections in the target image,implement a multi-network training architecture. Specifically, we propose the use of three distinct neural networks:

- **Pose Alignment Network (PAN):** This network is responsible for aligning the in-shop garment with the posture of the person in the target image. It takes into account the skeletal keypoints and adjusts the clothing to fit the pose accurately, ensuring that the virtual outfit conforms to the body's structure and orientation.
- **Texture Refinement Network (TRN):** After alignment, the texture refinement network enhances the clothing's visual details, such as patterns, logos, and fabric textures. It ensures that these features remain realistic and are not distorted during the pose transformation, maintaining visual fidelity in the final image.
- **Fitting Network (FTN):** Finally, the fitting network fine-tunes the garment's placement and scale to match the contours and proportions of the body. It focuses on making the clothing appear naturally fitted, improving realism and reducing visual artifacts like misalignment or unnatural edges. Together, these networks form the core of a robust virtual dressing system capable of delivering realistic, responsive, and personalized shopping experiences for both users and retailers.

CHAPTER 10

CONCLUSION

The popularity of online shopping and people's desire to utilize it to the fullest extent possible when buying clothes justifies the necessity to create an algorithm that digitally dresses them in the chosen clothing. The requirement to spend hours physically trying on a range of outfits is a regular issue client run into when shopping for clothing. The time available might not be enough, and this might be exhausting. The utilization of a virtual styling room that serves as a trial room using live video feed is the suggested remedy for this issue. The human body's nodes and points are plotted using a Kinect sensor, and this information is then utilized to create an image of clothing over the user's body, obviating the need for actual fittings and saving time.

The ability to check out themselves in different outfits with less limits thanks to this technology would be greatly appreciated by online buyers.We came to the conclusion that this exercise really saves time. It doesn't demand extra work. Anyone who is not technically savvy can use this virtual machine. It doesn't call for a lot of technical expertise. It is hence accessible. Therefore, it is the perfect addition for a clothier. Overall, the suggested virtual dressing room appears to be a solid option for precise and speedy virtual clothing fitting.The growing popularity of online shopping has revolutionized the retail landscape, offering consumers convenience and a vast selection of products at their fingertips. However, one major challenge that persists, especially in the clothing sector, is the inability to try on garments before purchasing. This limitation often leads to uncertainty about fit, style, and overall appearance, sometimes resulting in dissatisfaction and high return rates.

To bridge this gap between physical and online shopping, the development of an algorithm that can digitally dress individuals in their chosen clothing items has become not only relevant but necessary. Traditional shopping often requires customers to spend considerable time in fitting rooms, trying on multiple outfits to decide which one suits them best. This process can be both time-consuming and exhausting, especially when there are long queues or limited trial room availability in busy stores. Furthermore, in the context of increasing health and hygiene awareness, particularly after the global pandemic, many customers may feel uncomfortable physically trying on clothes that others have worn. exhausting,

especially when there are long queues or limited trial room availability in busy stores.

Therefore, a contactless, efficient alternative that provides a similar or even enhanced shopping experience is in high demand. A promising solution to this issue is the implementation of a virtual dressing room, which functions as a digital trial room using a live video feed. Leveraging advanced technologies like Kinect sensors, the system is capable of detecting and mapping key human body nodes and skeletal points.

Furthermore, it contributes to sustainability by reducing unnecessary returns and the associated logistical costs and environmental impact. In conclusion, the development of a virtual dressing room powered by real-time body mapping and digital clothing overlay presents a highly effective solution to the common challenges faced in clothing retail. It offers time efficiency, comfort, accessibility, and a personalized shopping experience. As online shopping continues to grow, such innovative tools will become indispensable in delivering customer satisfaction and enhancing the overall retail journey. The proposed system is not only a technological advancement but also a practical, user-centric approach to modern retail.
By combining efficiency with ease of use, it paves the way for a smarter, more engaging, and future-ready fashion shopping experience.

REFERENCES

[1] Srinivasan K. Vivek S, Implementation of Virtual Fitting Room Using Image Processing, Department of Electronics and Instrumentation Engineering, Sri Ramakrishna Engineering College Coimbatore, India, ICCCSP-2017.

[2] Cecilia Garcia, Nicolas Bessou, Anne Chadoeuf and Erdal Oruklu, Image Processing Design Flow for Virtual Fitting Room Applications used in Mobile, Devices by Department of Electrical and Computer Engineering Illinois, Institute of Technology Chicago, Illinois, USA.

[3] Kshitij Shah, Mridul Pandey, Sharvesh Patki, Radha Shankarmani, A Virtual Trial Room using Pose Estimation and Homograph, Department of Information Technology Sardar Patel Institute of Technology Mumbai, India.

[4] Vlado Kitanovski, Ebroul Izquierdo Multimedia and Vision Research Group – “3d Tracking of facial Features for Augmented Reality Applications”, Queen Mary, University of London, UK

[5] A. Hilsmann and P. Eisert, “Realistic cloth augmentation in single view video,” in Proc. Vis.,Modell., Visualizat. Workshop, 2009, pp.

[6] LanZiquan, Augmented Reality – Virtual fitting Room using Kinect, Department of Computer Science,School of Computing, National University of Singapore 2011/2012

[7] Vipin Paul, Sanju Abel J, Sudarshan S, Praveen M, “Virtual Trial Room”, South Asian Journal of Engineering and Technology Vol3, No5(2017)87-96

[8] Nikita Deshmukh, Ishani Patil, Sudehi Patwari, Aarati Deshmukh, Pradnya Mehta, “Real Time Virtual Dressing Room”, IJCSN International Journal of Computer Science and Network, Volume 5, Issue 2, April 2016.