CBS1901 TARP PROJECT REPORT

Project Title: Try your clothes online using Virtual Try-On

Team Members:
Rachita Singh (20BBS0115)
Akshita Jain (20BBS0127)
Rahul Agarwal (20BBS0089)
Atul Anurag (20BBS0095)
Shivam Kumar (20BBS0040)

Table of Contents

1. Abstract

Virtual try-on clothes software is a cutting-edge technology that allows customers to try on clothes virtually, eliminating the need to visit a physical store or try on clothes in person. The software uses advanced 3D modeling and computer vision technology to create a realistic representation of the user's body, allowing them to see how different clothing items would look on them.

To use virtual try-on clothes software, customers can simply upload a photo of themselves or use a camera to capture an image. The software then uses advanced algorithms to analyze the image and create a 3D model of the user's body. The software can take into account the user's height, weight, and body shape, as well as other factors such as skin tone and hair color, to create a highly accurate representation of the user.

Once the 3D model is created, the user can browse through a virtual catalog of clothing items and select the ones they want to try on. The software then superimposes the clothing item onto the 3D model of the user's body, allowing them to see how the clothing item would look on them from all angles.

Virtual try-on clothes software has the potential to revolutionize the way people shop for clothes. It makes it more convenient and accessible for customers to find the perfect fit and style without leaving their homes. It can also help reduce the number of returns and exchanges, as customers can see how the clothing item would look on them before making a purchase.

Virtual try-on clothes software is also beneficial for the fashion industry. It allows designers and retailers to showcase their products in a more interactive and engaging way, which can help increase sales and brand awareness. It also provides valuable data on customer preferences and body measurements, which can be used to optimize product design and inventory management.

Overall, virtual try-on clothes software is an exciting and innovative technology with significant potential to transform the fashion industry and improve the shopping experience for customers.

2. Introduction

Virtual try-on clothes software is a technology that allows customers to try on clothes virtually, which has been rapidly evolving in recent years. The software uses advanced 3D modeling and computer vision technology to create a realistic representation of the user's body, allowing them to see how different clothing items would look on them without physically trying them on.

One type of virtual try-on clothes software involves using augmented reality (AR) technology. This involves overlaying a digital image of the clothing item onto the user's live camera feed, allowing them to see themselves wearing the item in real-time. Another type of virtual try-on software uses computer-generated models that are customized to the user's body shape and size. The user can then browse a virtual catalog of clothing items and see how each item would look on their customized model.

Virtual try-on clothes software has become increasingly popular in recent years, especially as more consumers have shifted to online shopping. It offers many benefits for customers, such as the ability to try on clothes from the comfort of their own homes, avoiding the inconvenience and potential risks associated with in-person shopping. For retailers, virtual try-on clothes software can help to increase sales and reduce returns, as customers can see how the clothing item will look on them before making a purchase.

However, there are some challenges associated with the development and implementation of virtual try-on clothes software. One major challenge is ensuring the accuracy of the 3D models, which requires precise measurements and detailed data on the user's body shape and size. Another challenge is ensuring the software is compatible with different devices and platforms, as well as ensuring that the software runs smoothly and without glitches.

Despite these challenges, virtual try-on clothes software has a bright future in the fashion industry. As technology continues to evolve and improve, we can expect to see even more advanced and sophisticated virtual try-on clothes software solutions that will transform the way people shop for clothes. Additionally, virtual try-on clothes software has the potential to promote body positivity and inclusivity by providing a more accurate representation of how clothing will look on people of all sizes and shapes.

3. Motivation and feasibility study

The motivation behind the development of virtual try-on clothes software is to provide a more convenient and accessible way for customers to shop for clothes while reducing the environmental impact of the fashion industry. In addition, it allows fashion retailers to increase sales, reduce returns and exchanges, and gain valuable data on customer preferences and body measurements. Virtual try-on clothes software can also help to promote body positivity and inclusivity by providing a more accurate representation of how clothing will look on people of all sizes and shapes.

3.1 Types of Feasibility for Virtual Try-On Clothes Software:

3.1.1 Technical Feasibility:

The development of virtual try-on clothes software requires advanced 3D modeling and computer vision technology to create a realistic representation of the user's body. The software needs to accurately map the user's body shape and size to create a digital model of the user that can be used to try on clothes virtually. This requires a significant amount of computing power and data processing capabilities. In addition, the software needs to be compatible with different devices and platforms, such as desktop computers, smartphones, and tablets. It also needs to be able to run smoothly without glitches or delays, providing a seamless and realistic user experience.

3.1.2 Economic Feasibility:

The development and implementation of virtual try-on clothes software require significant investment in software development, data management, and server maintenance. This means that the cost of implementing this technology may be high. However, the potential benefits of virtual try-on clothes software may outweigh the costs. Retailers can increase sales and reduce returns and exchanges, as customers can see how clothing items will look on them before making a purchase. This can lead to improved customer satisfaction and loyalty, ultimately resulting in increased revenue.

3.1.3 Operational Feasibility:

Virtual try-on clothes software needs to be integrated into the existing organizational structure of a retailer. This means that staff needs to be adequately trained to use the software, and the software needs to be seamlessly integrated into the retailer's website or mobile app. It also requires continuous maintenance and updates to ensure that it remains effective and relevant.

3.1.4 Legal Feasibility:

Virtual try-on clothes software needs to comply with legal and regulatory requirements, such as data privacy regulations like the GDPR or CCPA. The software needs to ensure that user data is collected and processed in a legal and ethical manner. Retailers also need to ensure that the software does not infringe on any patents or copyrights.

4. Objective of the project

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 favourite clothing anywhere and anytime with smartphone. The main objective of the project is divided into subobjectives as shown as below.

1. To detect and extract human body skeleton-based joint positions using smartphone camera.
2. To calculate body measurements based on the extracted body skeleton joint positions
3. To fit virtual garments onto human body according to the extracted body skeleton joint positions, body measurements and garment measurements.
4. To develop an interactive AR virtual fitting room mobile application on Android mobile platforms.

The different processes which can be used in this project are Image Recognition, Image Segmentation and mask creation, Classification and Mask superimposition.

5. Literature Review

A. **The role of virtual try-on technology in online purchase decisions from the consumer's aspect**- Tingting Zhang, William Yu Chung Wang,Ling Cao, Yan Wang

Mechanism - Likert scale to measure all of the items, study used a web based survey to collect data, experiment, MANCOVA, mediation test

Limitation/gap identified - The study provides a single, static view of online consumer behavior. Further studies could track changes in attitudes towards VTO technology and intent to purchase apparel online.

B. **You can try without visiting: a comprehensive survey on virtually try-on outfits-**Hajer Ghodhbani, Mohamed Neji, Imran Razzak,Adel M. Alimi

Mechanism - Convolutional Neural Network (CNN) framework is used for deep human parsing-based work.

Limitation/gap identified - Try-on image generation ,Network efficiency,Virtual try-on datasets.

C. **Case study on augmented reality virtual try-on for retail evolution:**Shreeganesha K V, Sneha Pai, Abhishek Bhat, P Arunagiri, Vinayambika S Bhat

Mechanism - CNN Geometry matching PIFU, OpenGL 3D rendering, Azure, Unity and AR toolkit, SVM local feature extraction, SIFT, SDM, RGB-D & face tracker technique, TDTOS framework simulated in FPGA

Limitation/gap identified - The real-time data processing uses tech such as NoSQL, MongoDB, and Apache Kafka.A comparative analysis of data mining techniques is needed to understand customer behavior.

D. **Me or Just Like Me? The Role of Virtual Try-On and Physical Appearance in Apparel M-retailing:**Daria Plotkina, Hélène Saurel

Mechanism - An experimental study compared an AR-based VTO application with an m-commerce interface displaying pictures of human models wearing garments.

Limitation/gap identified - The experimental design has limitations, such as potential biases in the VTO video and limited AR experience due to unavailability of sophisticated AR-based VTO on mobile platforms.

E. **Mobile Application for Augmented Shopping: Virtual Shoe Try-on and Virtual Equipment Placement:**Karkour, Mustafa Liang, David AlAghbar, Amir Melaab, Rassim: Doush, Iyad Abu

Mechanism - Virtual Shoe Fitting: The virtual fitting experience in mobile shopping enhances customer satisfaction, brand loyalty, and mobile purchase intention which impacts the customer purchase.

Model Placement in augmented reality: 3D models like furniture can be placed normally in a room, however, the model itself sometimes cannot fit a room.

Limitation/gap identified - For further development, we could improve the responsiveness and accuracy of the features such as model placement and the measuring capabilities.Finally, the proposed system can be compared with similar AR technology in terms of speed, accuracy and user experience.

F. **High-quality 3D Clothing Reconstruction and Virtual-Try-On: Pants case:** Tuan, Thai Thanh Yun, Youngsik Ahn, Heejune

Mechanism - 3D Pants Reconstruction and Warping: Pants is first put into 3D based on joints on clothing, and the SMMPL model that fits the clothing is estimated using SMPLify-X.

Target Semantic Segmentation Generation Network: The try-on image is generated by combining the warped clothing for the target human pose and shape and combining the identity region (e.g. face and hands) and reserved region (e.g. clothing not to change).

Limitation/gap identified - Data availability and quality, model complexity and computational cost, user experience and feedback, diversity and representation

G. **Research on Face Pose Estimation Method for Virtual Try- on System:** Shao, Xiwen

Mechanism - Existing Face Pose Estimation Methods, 3D Face Pose Estimation Based On Feature Points

Limitation/gap identified - In the model of the face pose yaw angle, D and D1 are approximately equal. For faces of people in various regions and people with various face shapes, D and D1 may be very different.

H. **"Virtual Try-On Using Deep Learning for Clothes: A Review":**R. Agrawal, M. Chandak, and N. Bansal

Mechanism - The paper discusses the different approaches used to develop virtual try- on systems, including image- based, video- based, and 3D- based methods.

The system uses deep neural networks to learn the features of the clothing items and the user's body shape.

Limitation/gap identified - Lack of empirical evaluation and limited discussion on 3D-based methods.

I. **"Virtual Try-On: Learning from Cross-Modal Aggregation":**L. Zhang, Y. Shen, and L. Lin

Mechanism - The proposed method consists of two main components: a cross-modal feature aggregation network and a cross-modal attention mechanism. It presents a method that combines the advantages of both image-based and 3D-based virtual try-on systems to improve the accuracy and efficiency of virtual try-on systems.

Limitation/gap identified - Limited explanation of the cross-modal attention mechanism, limitations on the impact of clothing deformation and ethical and social implications.

J. "Virtual Try-On: A Review of Augmented Reality-Based Garment Simulation" :S. Prabhu and S.B. Patil

Mechanism - It provides an overview of augmented reality (AR)-based virtual try-on systems for garment simulation. The challenges faced by AR-based virtual try-on systems, such as occlusion, lighting, and fabric deformation are discussed.

Limitation/gap identified - There is limited discussion on the technical details of these systems. The paper provides a review of various AR-based virtual try-on systems, but there is a lack of evaluation of these systems.

6. Requirement specifications

6.1 Gantt Chart

A Gantt chart is a commonly used graphical depiction of a project schedule. It's a type of bar chart showing the start and finish dates of a project's elements such as resources, planning, and dependencies.

The project development is planned from start to finish using Gantt chart. It shows the past, current and future progresses of the project based on the methodology phases: Requirement, Planning, Analysis and Design, Implementation and Testing, Deployment.

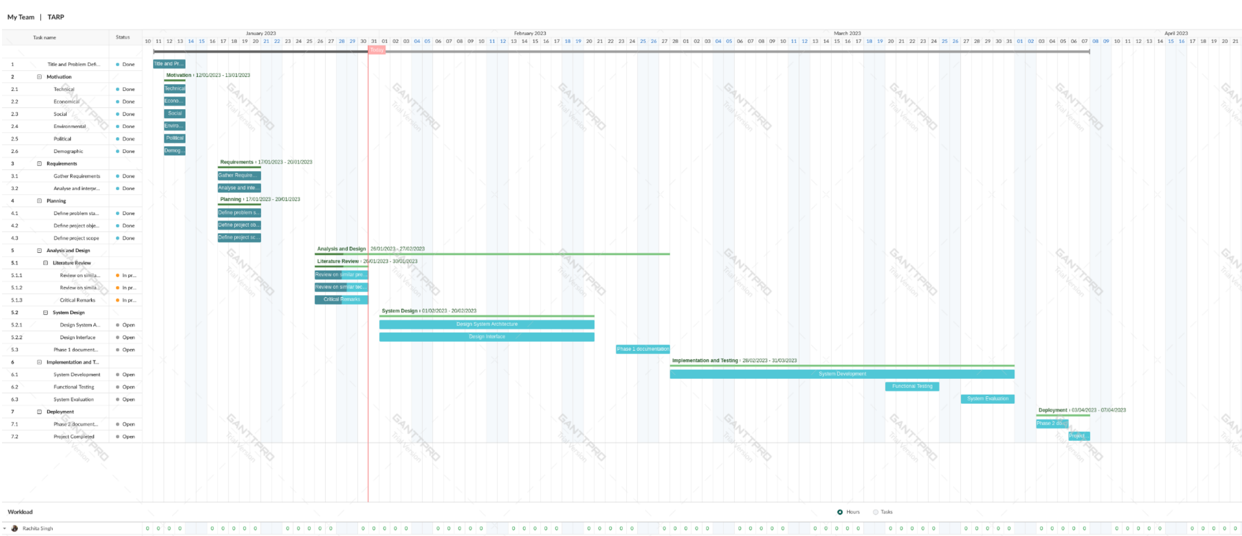


Fig6.1 Gantt chart for the project over 4 months (January 2023 - April 2023)

6.2 Functional Requirements

This section explores the functional requirements of the virtual try-on solution.

1. Image Recognition: The model should be able to recognize the different items of clothing or accessories from images, videos or 3D models.
2. Body Measurements: The model should be able to extract measurements of the user's body, including height, weight, chest, waist, and hip measurements, and use them to customize the fit of the virtual clothes.
3. Garment Simulation: The model should be able to simulate the appearance of the garment when worn, including how it drapes, folds, and stretches on the body.
4. Realistic Movement: The model should be able to animate the virtual clothes and accessories in a way that is realistic, and responds to user movements and interactions.
5. User Interaction: The model should provide the user with an intuitive and easy-to-use interface for selecting and trying on virtual clothing items and accessories.
6. Personalization: The model should be able to provide personalized recommendations based on the user's preferences, shopping history, and body measurements.
7. Integration: The model should be able to integrate with e-commerce websites and social media platforms to provide a seamless shopping experience for users.

6.3 Non-functional requirements

This section explores the non-functional requirements of the virtual try-on solution.

1. Performance: The model should be able to handle a large number of users simultaneously without slowing down or crashing.
2. Responsiveness: The model should respond quickly and accurately to user inputs, providing an immersive and engaging user experience.
3. Usability: The model should be intuitive and easy to use, with a simple and user-friendly interface.
4. Compatibility: The model should be compatible with a wide range of devices, operating systems, and web browsers.

5. Accessibility: The model should be accessible to users with disabilities, including those with visual or hearing impairments.
6. Security: The model should ensure the user's privacy and security by implementing encryption and other security measures to protect user data.
7. Reliability: The model should be reliable and robust, with minimal downtime or errors.
8. Scalability: The model should be able to scale up to handle large volumes of users and data, and accommodate future updates and improvements.
9. Maintainability: The model should be easy to maintain and update, with well-documented code and clear instructions for developers.
10. Compliance: The model should comply with relevant laws and regulations related to data privacy, security, and accessibility.

6.4 Use case diagram

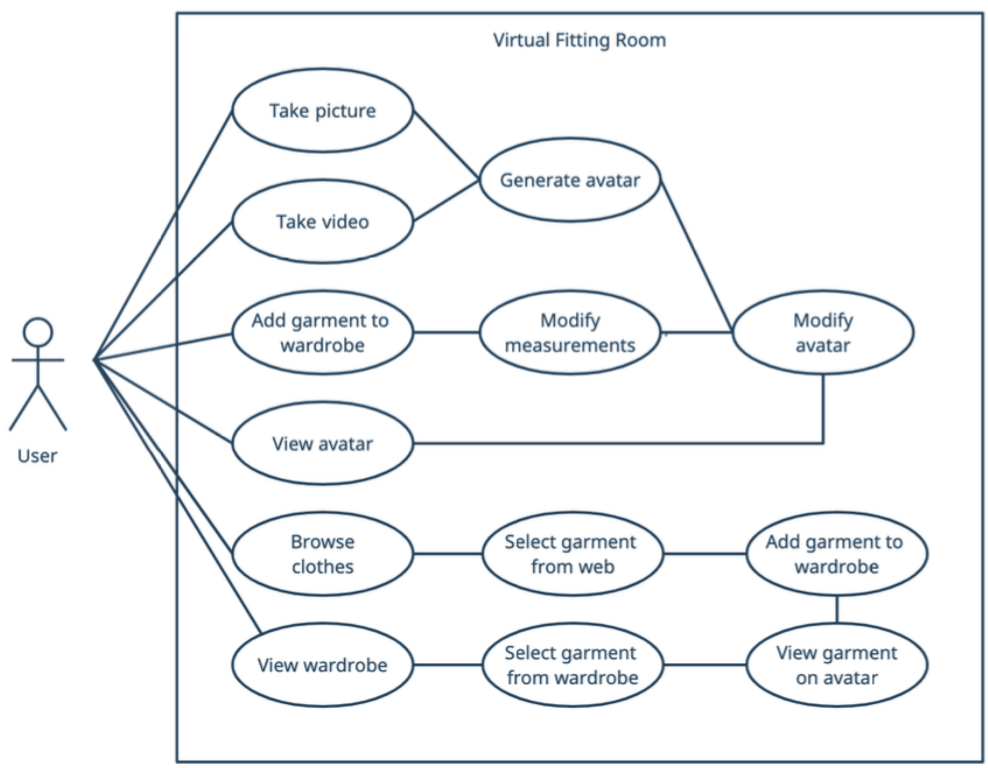


Fig 6.2 Use Case diagram for a user interacting with the virtual try-on model

The diagram shows the main interactions between the user and the virtual try-on system. The user can choose a product to try on, upload an image of themselves, adjust the try-on product to their preferences, view the product in augmented reality, and purchase the product if satisfied. The system uses ML and DL algorithms to provide an accurate and realistic representation of the product on the user's image, and to adjust the product to the user's preferences.

7. Approach and Methodology/Techniques

The approach of this project is to create a virtual trial room for humans to try various clothing virtually. For our real-time implementation, we used the open-source library, OpenCV.

- The first step is to obtain the video stream of the user with the help of web camera. With the OpenCV, each frame is converted into a matrix in gray scale.
- Creation of cloth masks with the help of bitwise operation
- Positioning of the cloth masks by using face detection.
- Resizing of the masks of cloths by using area interpolation
- Superimposition of the masks of cloths on the user.

7.1 OpenCV:

OpenCV stands for Open-Source Computer Vision. It's an Open-Source BSD licensed library that includes hundreds of advanced Computer Vision algorithms that are optimized to use hardware acceleration. OpenCV is commonly used for machine learning, image processing, image manipulation, and much more. OpenCV has a modular structure. There are shared and static libraries and a CV Namespace. One of OpenCV's goals is toprovide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly.
The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high-resolution image of an entire scene, find similar images.

7.2 Image Processing

Image processing refers to the manipulation and analysis of digital images using various algorithms and techniques. In the context of a virtual try-on model, image processing is used to enable the model to simulate how clothing items would look on a person. This involves capturing images or videos of the clothing items and the person, processing those images to extract features like the size and shape of the clothing, and then using that information to create a simulation of how the clothing would look on the person. Image processing techniques used for virtual try-on models may include color correction, image segmentation, feature detection, and machine learning algorithms that can identify and match clothing items to a person's body shape and size.

First, we import cv2. Next, we say cap = cv2.VideoCapture(0). This will return video from the first webcam on your computer.

```
cv2.waitKey(1)
cap=  cv2.VideoCapture(0)
```

Fig7.1 Video capture from web camera

Providing images of clothes as input for masking and converting original RGB images of clothes into grey scale so, reducing the image to grayscale greatly simplifies calculations and removes redundancies. Binary image is great too but it sacrifices too many information for it to be useful in many cases.

```
ih=shirtno
i=pantno
while True:
    imgarr=["shirt1.png",'shirt2.png','shirt51.jpg','shirt6.png',]

    imgshirt = cv2.imread(imgarr[ih-1],1) #original img in bgr
    if ih==3:
        shirtgray = cv2.cvtColor(imgshirt,cv2.COLOR_BGR2GRAY)
```

```
imgarr=["pant7.jpg",'pant21.png']
imgpant = cv2.imread(imgarr[i-1],1)
imgpant=imgpant[:,:,0:3]#original img in bgr
pantgray = cv2.cvtColor(imgpant,cv2.COLOR_BGR2GRAY) #grayscale conversion
```

Fig7.2 Attire Image Processing

7.3 Mask Creation

Mask creation is a technique used in image processing to isolate or segment specific areas of an image or video. In the context of a virtual try-on model, mask creation is used to separate the clothing item from the background and the person wearing it, allowing the model to simulate how the clothing item would look on a different person or background.

Thresholding is a segmentation technique used for separating an object from its background. It involves comparing each pixel of image with a pre-defined threshold value. It divides all pixels of the input image into 2 groups (1= pixels having intensity value lower than the threshold 2= pixels having intensity value higher than the threshold).

Bitwise operations function in a binary manner and are represented as grayscale images. A given pixel is turned "off" if it has a value of zero, and it is turned "on" if the pixel has a value greater than zero.

```
ret, orig_masks_inv = cv2.threshold(shirtgray,200 , 255, cv2.THRESH_BINARY)
orig_masks = cv2.bitwise_not(orig_masks_inv)
```

Fig7.3

Fig7.4 Mask Creation

7.4 *User Detection using Haar Cascade Classifier*

Image or Object Detection is a computer technology that processes the image and detects objects in it. People often confuse Image Detection with Image Classification. Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance

A Haar classifier, or a Haar cascade classifier, is a machine learning object detection program that identifies objects in an image and video.

```
face_cascade=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

Fig7.5 User detection using Haar Cascade Classifier

7.5 Resizing of mask

As we have created the mask earlier, from the images of clothes and taking the inverse of the mask, but the mask contains some redundant data that we do not need therefore we will select the region of interest (ROI) in which we will be selecting the area in our user body and superimposing the mask on the body of the user such that it should be properly aligned with the user body.

```
pant = cv2.resize(imgpant, (pantWidth,pantHeight), interpolation = cv2.INTER_AREA) #resize a
mask = cv2.resize(orig_mask, (pantWidth,pantHeight), interpolation = cv2.INTER_AREA)
mask_inv = cv2.resize(orig_mask_inv, (pantWidth,pantHeight), interpolation = cv2.INTER_AREA)
```

Fig7.6 Resizing of the mask

7.6 Mask superimposition

Mask superimposition is a technique used in virtual try-on models to combine the mask of a clothing item with an image or video of a person wearing different clothing. This technique enables the model to simulate how a new clothing item would look on the person, as if they were actually wearing it.

To create a mask superimposition, the model uses the mask of the clothing item, as well as various image processing techniques such as perspective transformation and color correction, to adjust the mask to match the pose and lighting of the person in the image or video.
The adjusted mask is then superimposed on top of the person, allowing the model to simulate how the new clothing item would look on them.

Take ROI (Region of Interest) for shirt from background equal to size of shirt image.
roi_bg contains the original image only where the shirt is not in the region that is the size of the shirt

roi_fg contains the image of the shirt only where the shirt is

Join the roi_bg and roi_fg

Place the joined image, saved to dst back over the original image.

```
rois = img[y1s:y2s, x1s:x2s]
num=rois

roi_bgs = cv2.bitwise_and(rois,num,mask = masks_inv)
roi_fgs = cv2.bitwise_and(shirt,shirt,mask = mask)
dsts = cv2.add(roi_bgs,roi_fgs)
```

Fig7.7 Mask superimposition

8. Implementation

8.1 Hardware specifications:

For a web based virtual try-on model, the laptop system requirements are specified as follows –

System Type	64-bit Operating System
Processor	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz
Graphics Processor	Intel(R) UHD Graphics 630
Graphics Card	NVIDIA GeForce GTX 1050
RAM	16.0 GB DDR4 2666MHz
Operating System	Microsoft Windows 10 Pro

Table 1

8.2 Software specifications:

1. Open source computer vision (OpenCV)

   OpenCV is the leading open source computer vision and machine learning software library of programming functions mainly aimed at real-time computer vision. OpenCV provides a common infrastructure for computer vision applications by offering several hundreds of computer vision algorithms.

2. TensorFlow Lite

   TensorFlow Lite is a lightweight open source deep learning framework for on- device machine learning inference. It provides machine learning models for developers to perform classification, detection, regression and any other machine learning algorithms.

3. Anaconda:

   It is a package management software with free and open-source distribution of the Python and R programming language for scientific computations (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify deployment.

4. Spyder:

Spyder, the Scientific Python Development Environment, is a free open- source integrated development environment (IDE) that is included with Anaconda written in Python, for Python, and designed by and for scientists, engineers and data analysts. It includes editing, interactive testing, debugging, and introspection features with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

Fig8.1

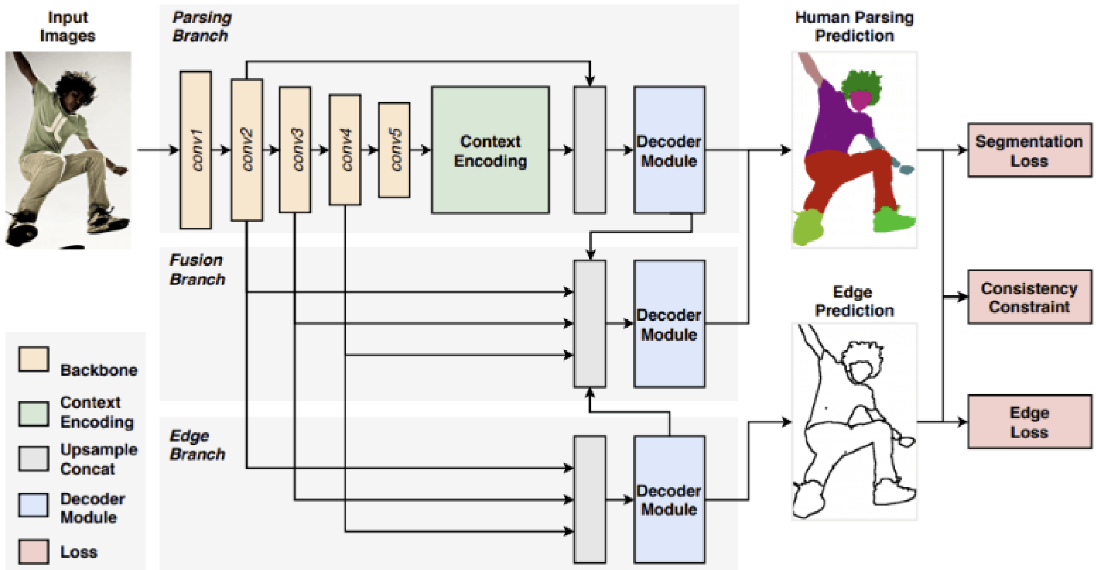


Fig8.2 Implemenation after hardware and software requirements are met

8.3 Human Body Recognition, Detection and Motion Tracking

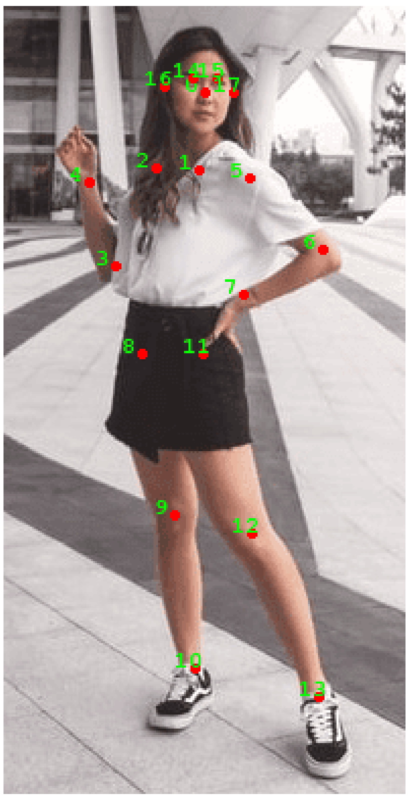
TensorFlow is the primary tool used for Human Body Recognition, Detection, and Motion Tracking, with the application utilizing a pose estimation model for mobile that operates using TensorFlow Lite. The model detects 14 joint positions, including top, neck, right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist, right hip, right knee, right ankle, left hip, left knee, and left ankle. To use the model, the application imports the model file as an asset.

The Virtual Try-On app employs the OpenCV library to process image data and supply it to the TensorFlow Lite pose estimation model. OpenCV is a widely used open-source computer vision library that offers various functions for image analysis and processing. Within this app, OpenCV reads the frame and feeds image data into the pose estimation model classifier for pose estimation. The native libraries of OpenCV are also imported into the jniLibs folder to ensure seamless usage of OpenCV functions within the application.

To improve the image quality and classification accuracy, the image data undergoes Gaussian Blur processing before being fed into the classifier. Gaussian Blur is a common technique in computer vision applications that reduces noise and smooths out the image.

Once the model and OpenCV libraries are successfully loaded, heat maps are generated, capturing features at various scales. These heat maps predict the probability of skeleton joints appearing at each pixel of the image.

The pose estimation model then returns the classification result to the application in real-time, allowing for accurate and efficient recognition, detection, and tracking of human body movements in live video.



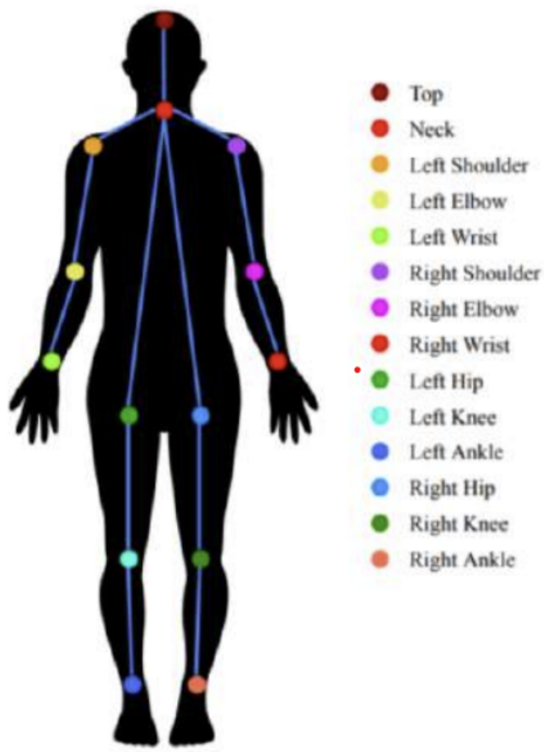


Fig8.3 **Fig 8.4**

8.4 Generation of Garment Model

In order to improve the virtual clothing fitting experience and provide accurate size and style to users, the garment model used in this project is customized to match the skeleton joint positions. This customization is necessary due to the lack of suitable apparel datasets available. The garment model is created using an actual garment with a transparent background and saved in the Portable Network Graphics (PNG) file format. This format is chosen for its ability to support lossless data compression and preserve data without any loss each time it is saved or opened. Additionally, the PNG format also supports transparency, which is a clear advantage over other file formats such as Joint Photographic Experts Group (JPEG).

Fig8.5

8.5 Superimposition of Garment Model over Human Body

The virtual garment is placed over the body by utilizing data of body measurements, body skeleton joint positions, and garment measurements. To achieve this, the garment model is first customized to match the skeleton joint positions since there is no available apparel dataset. The model is generated based on an actual garment with a transparent background and saved in the lossless Portable Network Graphics (PNG) format, which supports transparency.

Next, the garment model is converted into a Bitmap class, which creates two-dimensional matrices composed of pixels, with each pixel containing a color value. Each pixel of the Bitmap is processed in a customized way to align with the body skeleton joint points. The android.graphics.Matrix package is also integrated to perform scaling and transformation of the garment model to generate a matrix, based on the body measurements. These measurements include shoulder width, left and right arm length, left and right limb length.

By using the package, the scaling of the garment model is implemented based on the calculated distances between each of the body skeleton joint positions. The resulting matrix is used to place the virtual garment over the body, resulting in a more realistic and accurate virtual clothing fitting experience for the user.

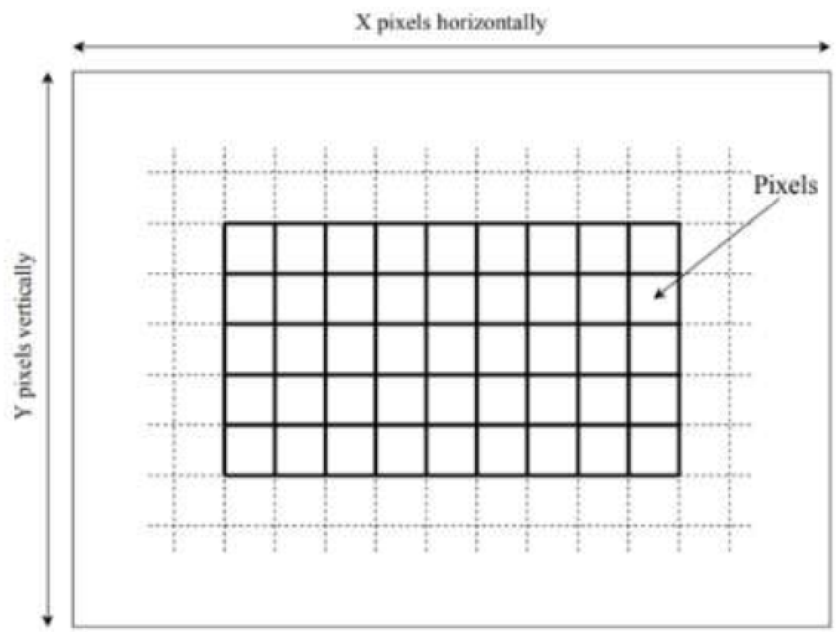


Fig8.6 Pixel description of the bitmap

Fig8.7

8.5 Flask

Flask is a popular web framework for building server-side applications in Python. It is known for its simplicity and ease of use, making it a popular choice for building web applications and APIs. Flask provides developers with a range of tools for building web applications, including URL routing, template rendering, and support for interacting with databases.

In the context of a virtual try-on software, Flask can be used to build a web server that allows users to upload images, select clothing items, and view the results of the virtual try-on process. Flask can also be used to integrate with other technologies, such as TensorFlow and OpenCV to provide a seamless user experience. By leveraging Flask's flexibility and ease of use, developers can quickly and efficiently build a virtual try-on system that meets the needs of their users.

8.6 Implemented Code:

```
from flask import Flask, render_template, request
#import json
from flask_cors import CORS
#import numpy as np
import cv2                              # Library for image processing
#from math import floor

app = Flask(__name__)
CORS(app)

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

@app.route('/pant.html')
def ploty():
    return render_template('pant.html')

@app.route('/predict/', methods=['GET','POST'])
def predict():
    shirtno = int(request.form["shirt"])
    pantno = int(request.form["pant"])

    cv2.waitKey(1)
    cap=  cv2.VideoCapture(0)
    ih=shirtno
    i=pantno
    while True:
        imgarr=["shirt1.png",'shirt2.png','shirt51.jpg','shirt6.png',]

        imgshirt = cv2.imread(imgarr[ih-1],1) #original img in bgr
        if ih==3:
            shirtgray = cv2.cvtColor(imgshirt,cv2.COLOR_BGR2GRAY)
            ret, orig_masks_inv = cv2.threshold(shirtgray,200 , 255, cv2.THRESH_BINARY)
            orig_masks = cv2.bitwise_not(orig_masks_inv)
```

```
        else:
            shirtgray = cv2.cvtColor(imgshirt,cv2.COLOR_BGR2GRAY) #grayscale conver
            ret, orig_masks = cv2.threshold(shirtgray,0 , 255, cv2.THRESH_BINARY) #
            orig_masks_inv = cv2.bitwise_not(orig_masks)
        origshirtHeight, origshirtWidth = imgshirt.shape[:2]
        imgarr=["pant7.jpg",'pant21.png']
        imgpant = cv2.imread(imgarr[i-1],1)
        imgpant=imgpant[:,:,0:3]#original img in bgr
        pantgray = cv2.cvtColor(imgpant,cv2.COLOR_BGR2GRAY) #grayscale conversion
        if i==1:
            ret, orig_mask = cv2.threshold(pantgray,100 , 255, cv2.THRESH_BINARY) #
            orig_mask_inv = cv2.bitwise_not(orig_mask)
        else:
            ret, orig_mask = cv2.threshold(pantgray,50 , 255, cv2.THRESH_BINARY)
            orig_mask_inv = cv2.bitwise_not(orig_mask)
        origpantHeight, origpantWidth = imgpant.shape[:2]

        face_cascade=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

        ret,img=cap.read()

        height = img.shape[0]
        width = img.shape[1]

        resizewidth = int(width*3/2)
        resizeheight = int(height*3/2)

        cv2.namedWindow("img",cv2.WINDOW_NORMAL)
        cv2.resizeWindow("img", resizewidth,resizeheight)

        gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
        faces=face_cascade.detectMultiScale(gray,1.3,5)
```

```
        for (x,y,w,h) in faces:  #strt pt  end pt
           # cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
            #cv2.rectangle(img,(x-100,y+100),(x+w-100,y+h+100),(255,255,25

            pantWidth =  3 * w  #approx wrt face width
            pantHeight = pantWidth * origpantHeight / origpantWidth #prese

            # Center the pant..just random calculations..
            if i==1:
                x1 = x-w
                x2 =x1+3*w
                y1 = y+5*h
                y2 = y+h*10
            elif i==2:
                x1 = x-w/2
                x2 =x1+2*w
                y1 = y+4*h
                y2 = y+h*9
            else :
                x1 = x-w/2
                x2 =x1+5*w/2
                y1 = y+5*h
                y2 = y+h*14
            # Check for clipping(whetehr x1 is coming out to be negative c

            #two cases:

            """
            close to camera: image will be to big
            so face ke x+w ke niche hona chahiye warna dont render at all
            """
```

```
        if x1 < 0:
            x1 = 0 #top left ke bahar
        if x2 > img.shape[1]:
            x2 =img.shape[1] #bottom right ke bahar
        if y2 > img.shape[0] :
            y2 =img.shape[0] #nichese bahar
        if y1 > img.shape[0] :
            y1 =img.shape[0] #nichese bahar
        if y1==y2:
            y1=0
        temp=0
        if y1>y2:
            temp=y1
            y1=y2
            y2=temp



        # Re-calculate the width and height of the pant image(to resize the image when it wud be pasted
        pantWidth = int(abs(x2 - x1))
        pantHeight = int(abs(y2 - y1))
        x1 = int(x1)
        x2 = int(x2)
        y1 = int(y1)
        y2 = int(y2)

        # Re-size the original image and the masks to the pant sizes



        pant = cv2.resize(imgpant, (pantWidth,pantHeight), interpolation = cv2.INTER_AREA) #resize all
        mask = cv2.resize(orig_mask, (pantWidth,pantHeight), interpolation = cv2.INTER_AREA)
        mask_inv = cv2.resize(orig_mask_inv, (pantWidth,pantHeight), interpolation = cv2.INTER_AREA)
```

```
roi = img[y1:y2, x1:x2]
    # roi_bg contains the original image only where the pant
    # in the region that is the size of the pant.
num=roi
roi_bg = cv2.bitwise_and(roi,num,mask = mask_inv)
    # roi_fg contains the image of the pant only where the p
roi_fg = cv2.bitwise_and(pant,pant,mask = mask)
# join the roi_bg and roi_fg
dst = cv2.add(roi_bg,roi_fg)
    # place the joined image, saved to dst back over the ori


# Blur the original input  image/feed when filter is superim

#slecting the area to blurr
top=img[0:y,0:resizewidth]
bottom=img[y+h:resizeheight,0:resizewidth]
midleft=img[y:y+h,0:x]
midright=img[y:y+h,x+w:resizewidth]
blurvalue=5

# Apply gaussian blurr
top=cv2.GaussianBlur(top,(blurvalue,blurvalue),0)
bottom=cv2.GaussianBlur(bottom,(blurvalue,blurvalue),0)
midright=cv2.GaussianBlur(midright,(blurvalue,blurvalue),0)
midleft=cv2.GaussianBlur(midleft,(blurvalue,blurvalue),0)

#
img[0:y,0:resizewidth]=top
img[y+h:resizeheight,0:resizewidth]=bottom
img[y:y+h,0:x]=midleft
img[y:y+h,x+w:resizewidth]=midright
img[y1:y2, x1:x2] = dst
```

```
            shirtWidth =  3 * w  #approx wrt face width
            shirtHeight = shirtWidth * origshirtHeight / origshirtWidth #preserving aspect ratio of original
            # Center the shirt..just random calculations..
            x1s = x-w
            x2s =x1s+3*w
            y1s = y+h
            y2s = y1s+h*4
            # Check for clipping(whetehr x1 is coming out to be negative or not..)

            if x1s < 0:
                x1s = 0
            if x2s > img.shape[1]:
                x2s =img.shape[1]
            if y2s > img.shape[0] :
                y2s =img.shape[0]
            temp=0
            if y1s>y2s:
                temp=y1s
                y1s=y2s
                y2s=temp



            # Re-calculate the width and height of the shirt image(to resize the image when it wud be pasted)
            shirtWidth = int(abs(x2s - x1s))
            shirtHeight = int(abs(y2s - y1s))
            y1s = int(y1s)
            y2s = int(y2s)
            x1s = int(x1s)
            x2s = int(x2s)


            # Re-size the original image and the masks to the shirt sizes
            shirt = cv2.resize(imgshirt, (shirtWidth,shirtHeight), interpolation = cv2.INTER_AREA) #resize al
            mask = cv2.resize(orig_masks, (shirtWidth,shirtHeight), interpolation = cv2.INTER_AREA)
            masks_inv = cv2.resize(orig_masks_inv, (shirtWidth,shirtHeight), interpolation = cv2.INTER_AREA)
```

```
            rois = img[y1s:y2s, x1s:x2s]
                # roi_bg contains the original image only where the shirt is not
                # in the region that is the size of the shirt.
            num=rois
            roi_bgs = cv2.bitwise_and(rois,num,mask = masks_inv)
            # roi_fg contains the image of the shirt only where the shirt is
            roi_fgs = cv2.bitwise_and(shirt,shirt,mask = mask)
            # join the roi_bg and roi_fg
            dsts = cv2.add(roi_bgs,roi_fgs)
            img[y1s:y2s, x1s:x2s] = dsts # place the joined image, saved to dst back over the original image
            #print "blurring"

            break
        cv2.imshow("img",img)
        if cv2.waitKey(100) == ord('q'):
            break

    cap.release()                           # Destroys the cap object
    cv2.destroyAllWindows()                 # Destroys all the windows created by imshow

    return render_template('index.html')
if __name__ == '__main__':
    app.run(host='0.0.0.0',debug=True,port=5000)
```

9. Model testing

9.1 Verification plan, analysis and result

The system should be able to recognize and detect human body in order to obtain body skeleton joint positions and body measurements in most of the common condition types. The system should also be able to fit garment onto human body in real-time through smartphone camera. In order to verify the functionalities of the system, a few of tests in different situations are planned to be carried out after the system is developed.

Method	Testing
Applicable Requirements	Detect user's body skeleton joint position and fit garment onto user's body in real-time.
Purpose / Scope	To detect human body and fit garment onto human body in simple scene.
Items Under Test	Live video
Special Conditions/Limitations	• If the exposed human body is partial, the system may not be able to detect the human body accurately.
Data Recording	None
Acceptance Criteria	The system fits garment onto human body or do not display garment.
Procedures	1. Open camera. 2. Select garment. 3. Expose whole body to camera. 4. The system fit garment onto user's body.
Troubleshooting	Repeat the procedure.
Post-Test Activities	None

Table 2

The above plan is carried out to test the developed web application. the application superimposes the virtual garment over the body after successfully extracted the body skeleton joint positions. Based on the real-time video of body in simple background, the application is able to fit the virtual garment onto the detected body in an appropriate way. In this case, it proves that the application succeeds in presenting the visualisation of wearing garment for user.

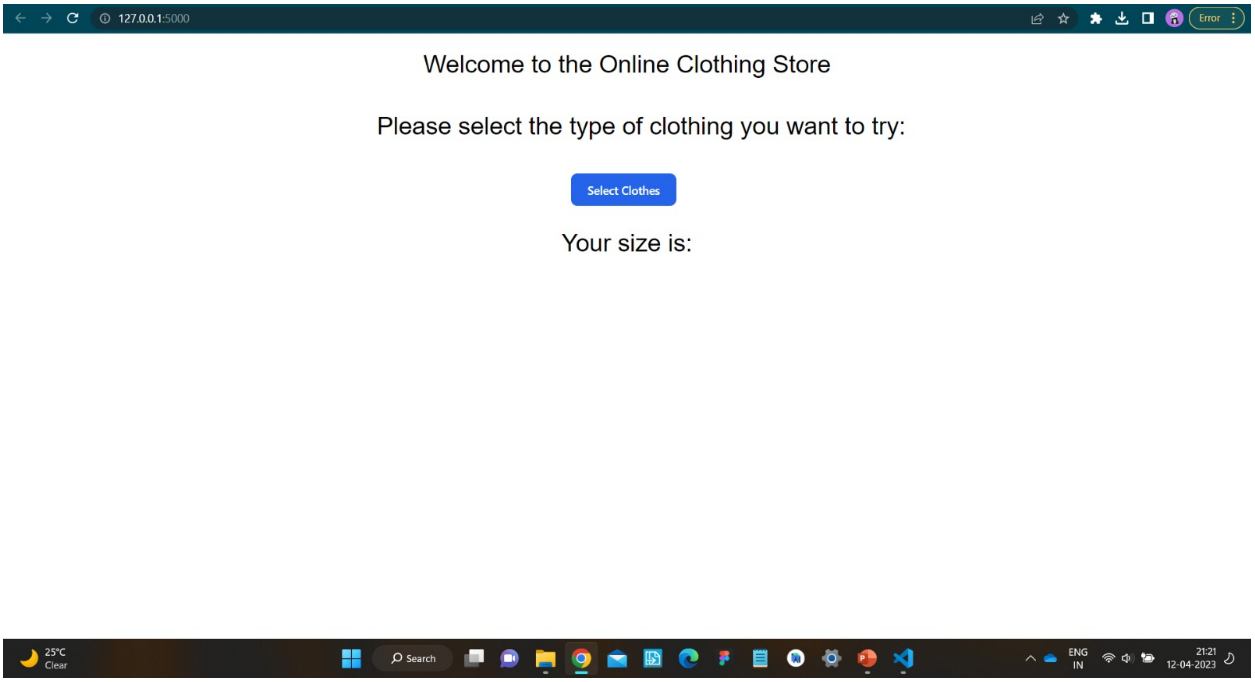
9.2 Use Case Testing

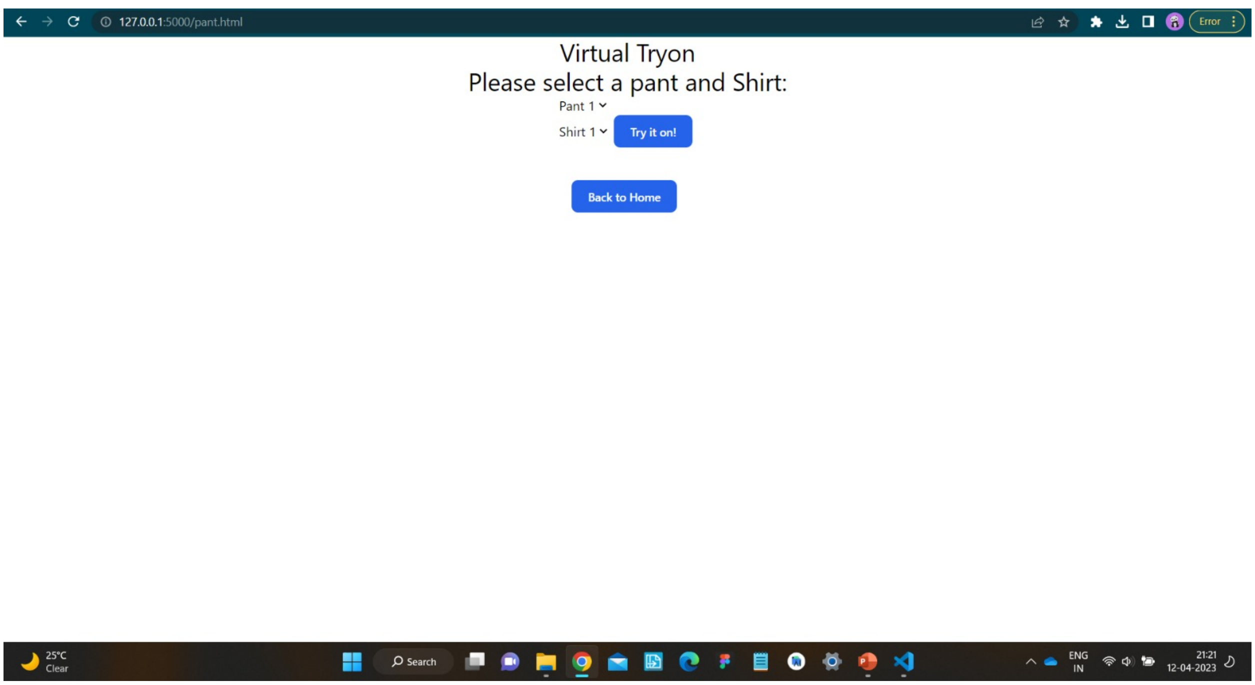
Use case testing is a type of software testing that focuses on evaluating the system's behaviour and functionality from the perspective of the end user. It involves testing the software using specific scenarios that represent how users would interact with the system in real-world situations. The objective of use case testing is to ensure that the software behaves correctly under a variety of different conditions and user interactions. This type of testing can help identify potential issues or bugs in the software that may not have been caught during other types of testing, such as unit testing or integration testing.

Objective	Try Garment On		
Condition	**Set Up**	**Expected Result**	**Pass/Fail**
Main Flow	1. User captures body skeleton joints successfully in real-time 2. System superimposes the garment over the body in real-time.	System displays the garment fitted onto the body in real-time.	Pass

Table 3

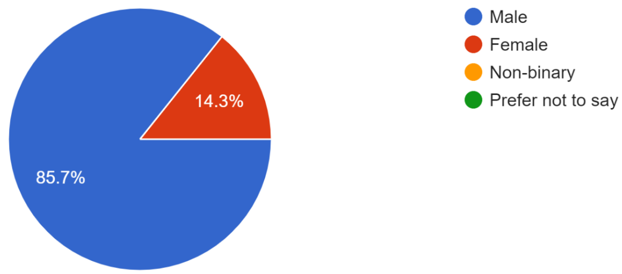
A frontend page was deployed to carry out system testing. Here, the user can select can select the shirt type along with the pants he/she would like to try. Once the items are selected, the user can view how it looks in the virtual try-on software.



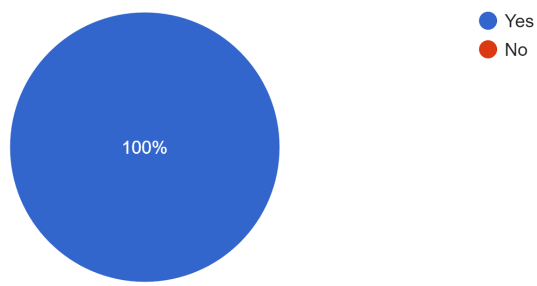


A survey was conducted as a part of the testing procedure -

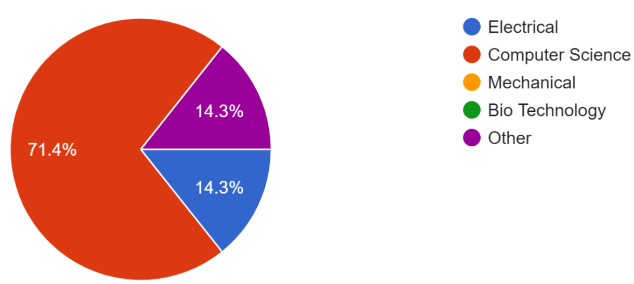
What is your gender?
7 responses



Are you currently enrolled in college?
7 responses

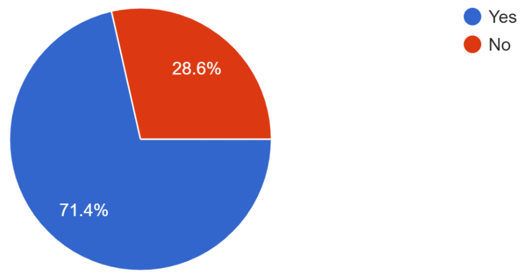


What is your college major or field of study?
7 responses



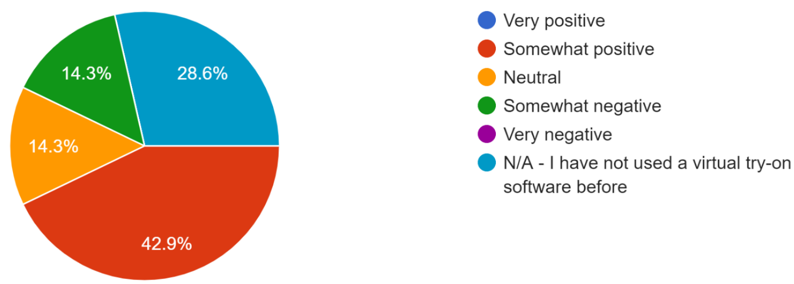
Have you ever used a virtual try-on software before?

7 responses



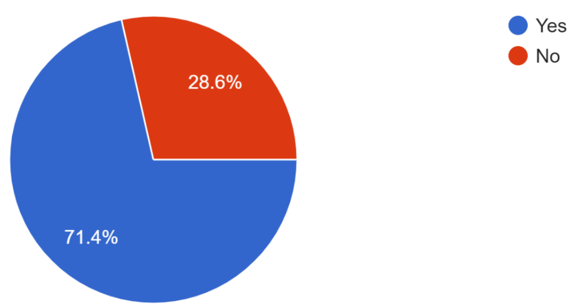
If you have used a virtual try-on software before, what was your experience like?

7 responses



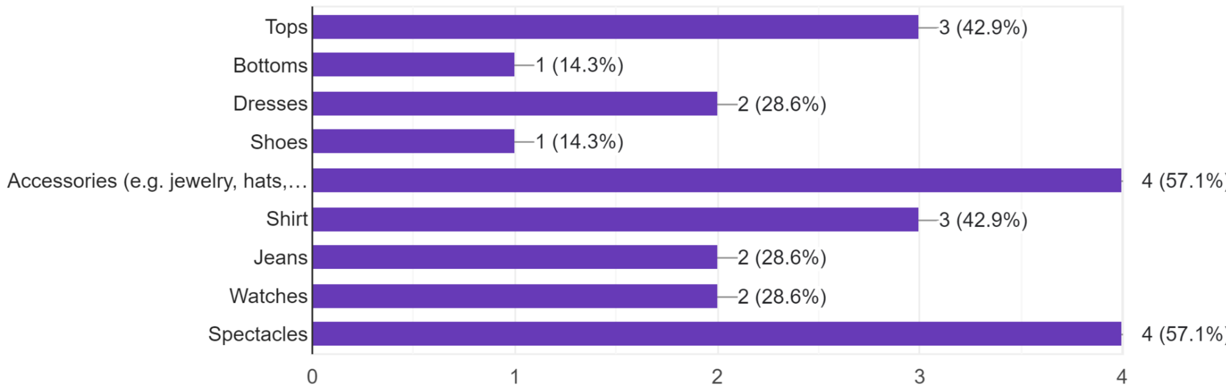
Would you be interested in using a virtual try-on software for trying on clothes and accessories?

7 responses



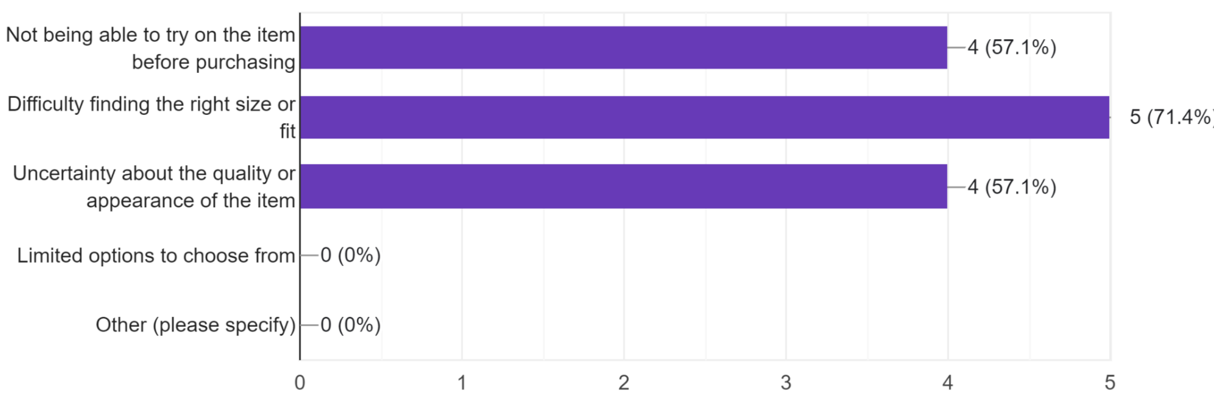
What type of clothing or accessories would you like to try on using a virtual try-on software? (Select all that apply)

7 responses



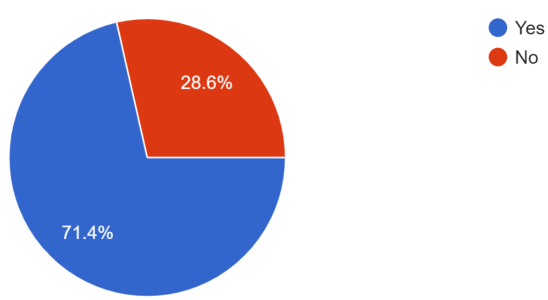
What are the biggest challenges you face when shopping for clothing or accessories online? (Select all that apply)

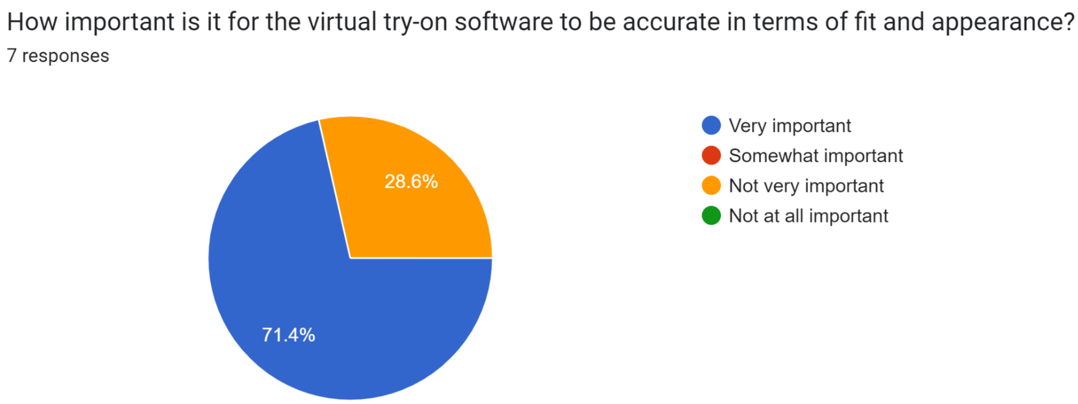
7 responses



Would you be more likely to make a purchase if you could try on the item virtually before buying it?

7 responses





The responses show that customers are inclined towards trying clothes before buying them, and a virtual try-on solution is feasible with respect to e-commerce and online shopping, as it enables them to have an idea about the fitting virtually and saves their time when compared to trying clothes at a walk-in brand showroom.

Two use cases have been considered for a college student -

1. When the student is going for a college interview and needs to select formal clothes to look professional.
2. Students also need to decide their daily wear while going to classes or while going outside the campus

The results of the use cases have been included in the next section as Trial 1 and Trial 2.

10. Result

One can expect to see a 3D version of the virtual product superimposed onto the user's image or avatar through the application. The process involves utilizing various algorithms including image segmentation, pose estimation, 3D modeling, texture mapping, as well as lighting and shading, in order to create an authentic visualization of the virtual product. The user can engage with the product, by adjusting their pose or physical movements, to witness how the product appears and fits from different perspectives. Moreover, the application may offer valuable insights regarding product features, such as size, color, and material, to enable users to make informed purchase decisions.

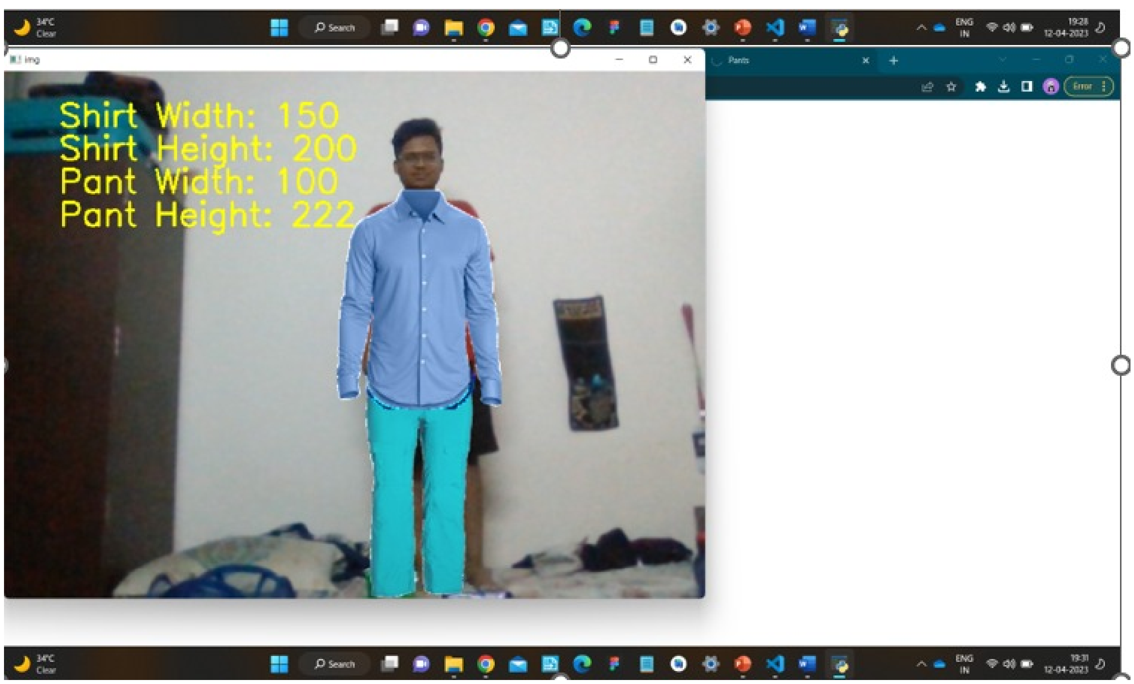


Fig10.1 Trial 1

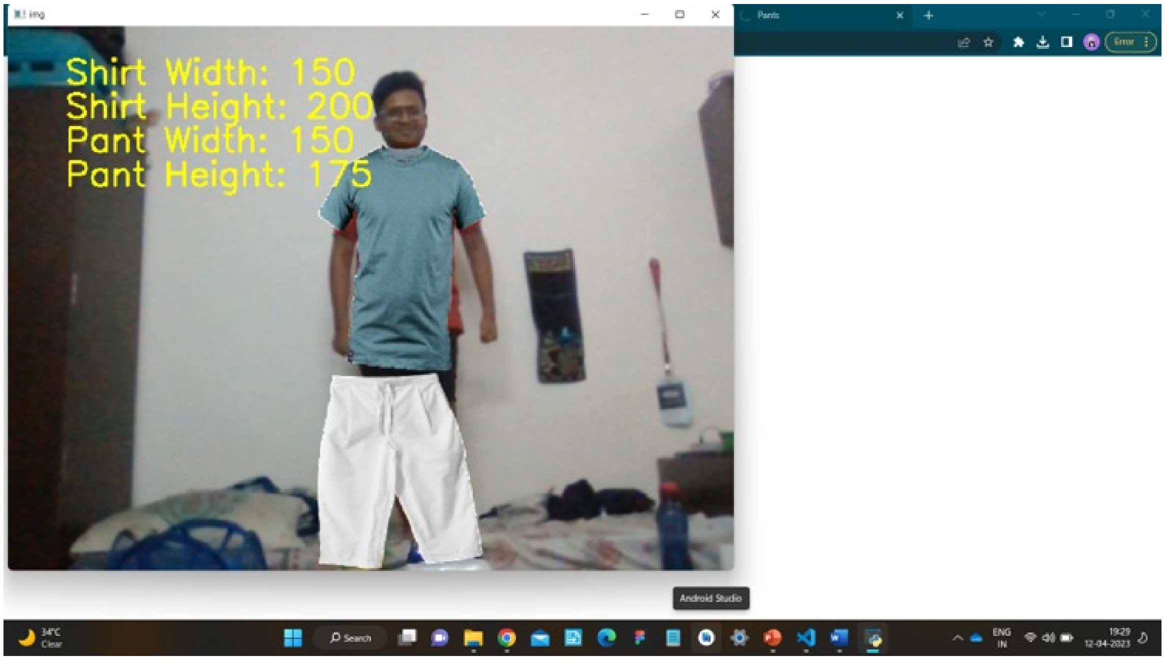


Fig10.2 Trial 2

11. Limitations and scope for future work

The research on virtual try-on and dress fitness prediction using DL/ML techniques has some limitations that warrant consideration in future work. First, the accuracy of the predictions heavily relies on the quality and representativeness of the training data. If the training data is biased, incomplete, or limited in diversity, it may affect the performance and generalization of the DL/ML models. Additionally, the reliance on digital avatars or virtual representations of users may not fully capture the individual variations in body shape, size, and posture, which could impact the accuracy of fitness prediction.

Another limitation is the computational complexity and resource requirements of DL/ML models. Deep learning models often require substantial computing power, memory, and storage, which may not be readily available to all users or in all environments. This can limit the scalability and accessibility of virtual try-on and dress fitness prediction systems, particularly for smaller retailers or users with limited computing resources.

Furthermore, virtual try-on technology is heavily dependent on the availability and quality of 3D models of clothing items. Creating accurate and realistic 3D models of various types of clothing, fabrics, and styles can be challenging and time-consuming, and may require specialized expertise or equipment. The lack of comprehensive and high-quality 3D models could impact the visual fidelity and realism of the virtual try-on experience.

Despite these limitations, there are several promising areas for future work in this field. This includes the development of more diverse and comprehensive datasets for training and evaluation, improving clothing modeling algorithms to enhance accuracy and realism, enhancing the user experience through advancements in virtual try-on technologies, and addressing ethical and social concerns. Further research can also explore novel DL/ML techniques, such as GANs and reinforcement learning, for virtual try-on and dress fitness prediction. Additionally, investigating the potential of incorporating user feedback, customization, and personalization in virtual try-on systems can enhance their practical applications in the fashion industry and beyond. Overall, addressing the limitations and exploring new avenues of research can further advance the field of virtual try-on and dress fitness prediction using DL/ML techniques, making them more reliable, user-friendly, and ethically responsible.

12. Conclusion

Virtual try-on and dress fitness prediction using DL/ML techniques hold great promise in revolutionizing the fashion industry by providing virtual fitting experiences for online shoppers. These technologies have shown significant advancements in recent years, enabling users to virtually try on clothes and predict their fitness with accuracy. However, there are still limitations that need to be addressed, such as data availability, clothing modeling complexities, technology limitations, user experience, and real-world deployment challenges.

Despite these limitations, the potential for future research and advancements in DL/ML techniques, technologies, and user-centric design is vast. Further development of large and diverse datasets, incorporation of physics-based simulations, multi-modal inputs, and improved garment modeling techniques, optimization of user experience and interaction design, and practical deployment and integration in real-world retail settings can enhance the effectiveness and adoption of virtual try-on and dress fitness prediction technologies.

With continued research and innovation, virtual try-on and dress fitness prediction using DL/ML have the potential to transform the online shopping experience, reduce returns, and improve customer satisfaction in the fashion industry. As technology continues to evolve, these technologies are expected to play an increasingly significant role in the future of fashion retail, offering personalized and immersive virtual fitting experiences for shoppers, and driving the growth and competitiveness of the fashion industry in the digital era.

13. References

1. Lassner, C.; Romero, J.; Kiefel, M.; Bogo, F.; Black, M. J.; Gehler, P. V. Unite the people: Closing the loop between 3D and 2D human representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4704–4713, 2017h
2. Pachoulakis, I. 2012, Augmented Reality Platforms for Virtual Fitting Rooms. *The International journal of Multimedia & Its Applications*, 4(4), pp.35-46.
3. https://docs.opencv.org/master/d9/df8/tutorial_root.html
4. M2E-Try On Net: Fashion from Model to Everyone(https://arxiv.org/pdf/1811.08599v1.pdf)
5. https://www.academia.edu/6940511/A_REAL_TIME_VIRTUAL_DRESSING_ROOM_APPLICATION_USING_OPENCV_A_PROJECT_REPORT
6. https://www.academia.edu/17537494/Virtual_Fitting_Room_using_Webcam
7. A. Mir, T. Alldieck, and G. Pons-Moll, "Learning to Transfer Texture From Clothing Images to 3D Humans," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
8. C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 7, pp. 1325–1339, 2014.
9. M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in International Conference on Machine Learning (ICML), 2017, pp. 214–22w
10. L:h J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4690–469q
11. LXh H. Dong, X. Liang, X. Shen, B. Wang, H. Lai, J. Zhu, Z. Hu, and J. Yin, "Towards Multi-Pose Guided Virtual Try-on Network," in International Conference on Computer Vision (ICCV), 2019, pp. 9026–9035h
12. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. ArXiv, abs/2208.12242, 202L

13. Szymczyk, M. 2019, Augmented Reality Virtual Dressing Room Use Cases For Shopper Engagement. *Zugara*. Available from: https://zugara.com/augmented-reality-virtual-dressing-room-use-cases-for-shopper-engagement
14. Takahashi, D. 2011, DEMO: Zugara engages online shoppers with dress-up visualization. *VentureBeat*. Available from: https://venturebeat.com/2011/02/28/demo-zugara-engages-online-shoppers-with-dress-up-visualization/
15. A. Radke, “Five Legal Challenges of the Fashion Industry,” The Fashion Law Chronicles, 08-Nov-2018.