ABSTRACT

In this era of technology and virtuality, the world is growing rapidly. From ordering food over to hiring handyman services everything can be done with just a click of a mouse or a swipe on the phone. One of the major contributors is the fashion and clothing industry, in which most of the people are buying garments online. The setback of this way of business being that the customers are unable to virtually try-on the clothes before purchasing them as they did in offline shopping with the help of trial rooms. The solution to this problem, we propose to build an application which seamlessly integrates the 2D images of the customer with target clothing regions and give the final output of person image with the targeted clothes, accessories. This virtual try-on clothing includes methods to overcome the segmentation and warping problems of the existing state-of-the-art models qualitatively and quantitatively in performance and promises a multipose virtual try-on experience.

CONTENTS

List of Diagrams:

1. INTRODUCTON

As the importance of online shopping increases, a technology that allows customers to virtually try on clothes is expected to enrich the customer's experience. Consumers' buying habits have changed widely as a result of online shopping. People are increasingly resorting to online retail stores for their shopping experiences. However, this change in customer behavior is producing a slew of issues for retailers and manufacturers alike. There are numerous fundamental challenges with online buying that businesses must overcome to remain competitive in today's market. Virtual try-on technology lets customers see how clothes fit themselves allowing them to virtually "try on" clothing before purchasing them. This method helps avoid challenges connected with conventional e-commerce sales channels, such as fit issues and environmental and financial returns costs. Virtual try-on refers to trying on digitally created garments or accessories in a virtual environment. That means that you can see how those items look on you, allowing you to know if something is a good fit without physically trying it on. Hence, virtual try-on technology could let you try on clothing without having it shipped or even leaving your house.

Virtual try-on helps consumers in the decision-making process. They get to experiment with products at their own pace until they find the right choice for them. Also, trying something on virtually is a fun experience that consumers want to share. Another major benefit of virtual try-ons is that it makes cross-selling easier. In the website they can then choose to purchase the product and will be taken to the corresponding product page and can add them to the cart after adding there is a button that says "Try On". Virtual try-on for apparel combines computer vision, deep learning to provide an immersive user experience. With evolving technology and customer needs, the future shop will seem quite different from what we are used to. Virtual clothing try-on occur in a variety of formats. Trying virtual apparel at home as part of the e-commerce experience or visiting physical stores. They're essentially the same thing because they're both virtual techniques of dressing. Customers may try on items via the internet at home using virtual fitting technology and a camera-equipped gadget. As the camera-equipped device captures the consumer, underlying deep

learning technology overlays a realistic virtual version of the product over their real-world image to demonstrate how the object would appear on the customer's body.

1.1 MOTIVATION

Our motivation here is to increase the time efficiency and improve the accessibility of the clothes try on by creating virtual dressing room environment. The implemented system introduced an advanced methodology which is presented for the purchase of clothing through a virtual fit on platform, which consumes far more less time than the normal process, making it easier for both seller and customer. This provides a realistic behavior for the suitability of the garment's details.

1.2 PROBLEM DEFINITION

To build an application that will help e-shopping by acting as a virtual dressing room. The images of the products will be super-imposed on the user's body using Deep learning algorithms.

Thus, helping the customer shop better by giving an effect of a real dressing room and get a better online shopping experience.

1.3 OBJECTIVE OF THE PROJECT

The main objective of this project is to provide the e-shopping customers with an interface to try on clothes virtually. Secondly, to let the customers enjoy a stand away, and allow the customers to try the apparels in different sizes. To also provide a user-friendly experience by demonstrating that with better interaction features in clothing web sites, it could affect and improve sales over the net.

It also focuses on the empowerment of the customer with additional facilities embedded in buttons to make online shopping easier and user friendly. For this project, it also provides real time customization of the customer's outfit, such as clothes and in the forthcoming modifications maybe some accessories too.

2. LITERATURE SURVEY

[1] CloTH-VTON+: Clothing Three-Dimensional Reconstruction for Hybrid Image-Based Virtual Try-On

Matiur Rahman Minar, Thai Thanh Tuan, Heejune Ahn- Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea

Image-based virtual try-on (VTON) systems based on deep learning have attracted research and

commercial interests. Although other projects show their strengths in blending the person and try-on clothing image and synthesizing the dis-occluded regions, their results for complex-posed persons are often unsatisfactory due to the limitations in their geometry deformation and texture-preserving capacity. This paper addresses these challenges, for seamlessly integrating the image-based deep learning methods and the strength of the 3D model in shape deformation. Specifically, a fully automatic pipeline is developed for 3D clothing model reconstruction and deformation using a reference human model: first, the try-on clothing is matched to the target clothing regions in the simple shaped reference human model, and then the 3D clothing model is reconstructed. The reconstructed 3D clothing model can generate a very natural pose and shape transfer, retaining the textures of clothes.

[2] A Mixed Reality Virtual Clothes Try-On System

Miaolong Yuan, Ishtiaq Rasool Khan, Farzam Farbiz, Senior Member, IEEE, Susu Yao, Arthur Niswar, and
Min-Hui Foo

Virtual try-on of clothes has received much attention recently due to its commercial potential. It can be used for online shopping or intelligent recommendation to narrow down the selections to a few designs and sizes. In this paper, it presents a mixed reality system for 3D virtual clothes try-on that enables a user to

see themselves wearing virtual clothes while looking at a mirror display, without taking off their actual clothes. The user can select various virtual clothes for trying-on. The system physically simulates
the selected virtual clothes on the user's body in real-time and the user can see virtual clothes fitting on a mirror image from various angles as they move. The major contribution of this paper is that it automatically customizes an invisible (or partially visible) avatar based on the user's body size and the skin color and use it for proper clothes fitting, alignment and clothes simulation in our virtual try-on system.

[3] ShineOn: Illuminating Design Choices for Practical Video-based Virtual Clothing Try-on

Gaurav Kuppa, Andrew Jong, Xin Liu, Ziwei Liu, Teng-Sheng Moh

Virtual try-on has garnered interest as a neural rendering benchmark task to evaluate complex object transfer and scene composition. Recent works in virtual clothing try-on feature a plethora of possible architectural and data representation choices. However, they present little clarity on quantifying the isolated visual effect of each choice, nor do they specify the hyperparameter details that are key to experimental reproduction. The paper in question, ShineOn, approaches the try-on task from a bottom-up approach and aims to shine light on the visual and quantitative effects of each experiment. This methodology limits itself from neck synthesis on zoomed view, however, lets customer's see how clothes fit themselves, allowing them to try virtually.

3. ANALYSIS

3.1 EXISTING SYSTEM

When it comes to the existing system, it does not have a system requirement for accurate pose for fitting virtual clothes try on user's image. It suffers from neck synthesis on zoomed view and improper distinction between torso and limbs of the user.

Existing system such as Cloth-VTON uses both 2D image-based and 3D model-based methods, where 2D uses neural network methods which generate disclosed human parts, where 3D method provide realistic deformation of clothing but the application range of the proposed clothing 3D reconstruction and deformation method is limited to rather simpler and tighter clothing. [6]

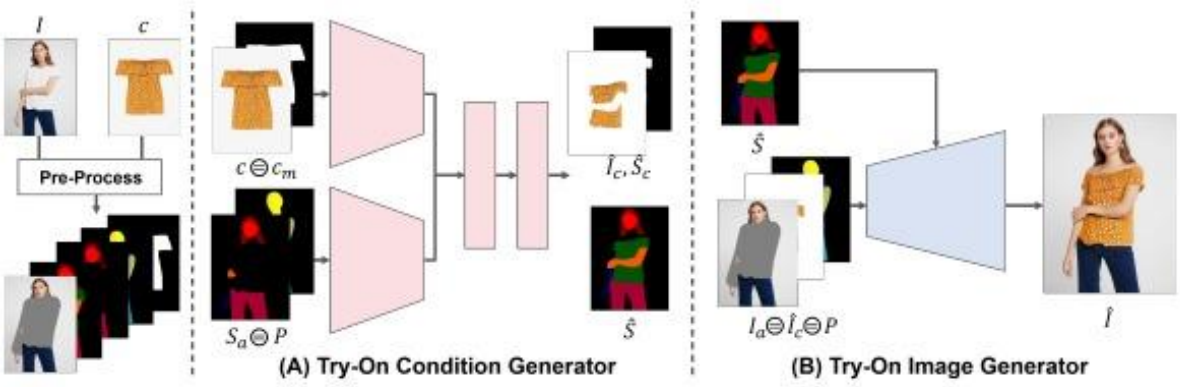


Fig 3.1 Existing System Architecture

3.2 PROPOSED SYSTEM

Our platform defers vastly in terms of efficiency and accuracy. The resultant picture is better as it overcomes some of the limitations of the existing system.

This platform/tool uses techniques such as image segmentation and pose transformation which helps in reshaping of the clothes, outfit matching and texture transfer. It helps avoid challenges connected with conventional e-commerce sales channels, such as fit issues, returns costs and travelling charges, and is very user-friendly.

The platform which we created has a website where users can check different products available in the website, they can add their products to the cart and try on virtually.

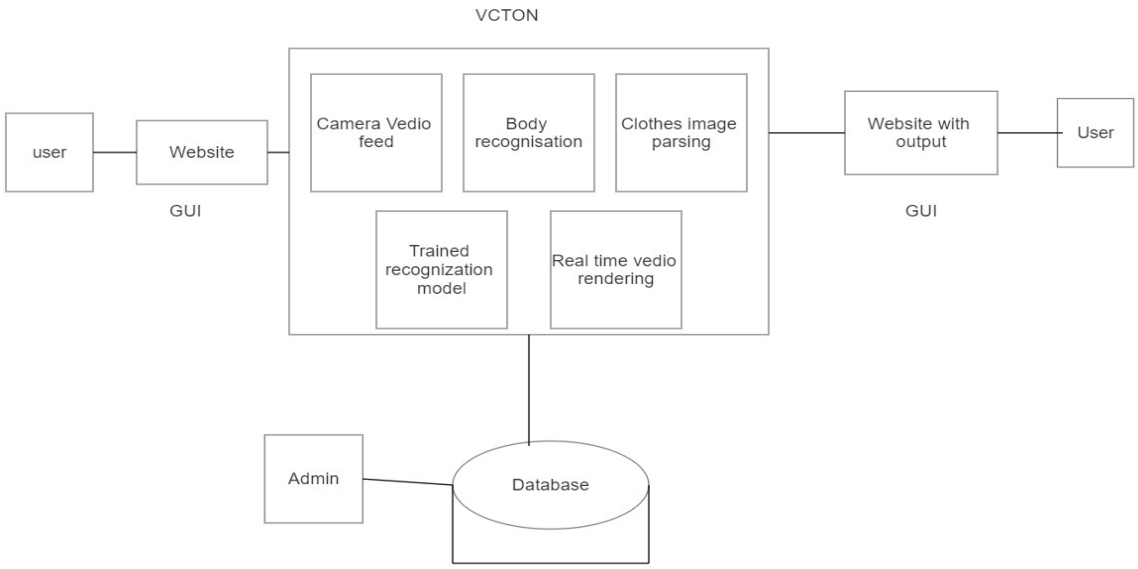


Fig 3.2 Proposed System Architecture

The try on in website helps the users able to check whether their outfit suits are not. The specialty of our proposed system is that one outfit can be tried by multiple users at a same time. This system is user friendly and creates a positive environment around the user when they are trying virtually. This is also an innovative outreach which helps the users to shop more items.

3.3 SOFTWARE REQUIREMENT SPECIFICATIONS

Software & Hardware Requirements:

Operating System: Windows 10 Home (32 or 64 Bit)

RAM: 8GB (recommended)

Hard Disk: 128GB (minimum)

Image Input

System Requirements:

Windows* 7, Vista,XP Me,2000,or 98SE;

Intel* Pentium* IV 500 MHZ or higher

256 MB RAM, 200 MB free hard drive space

Available USB 1.1 OR 2.0 Port (USB 2.0 Recommended)

3.3.1 PURPOSE

The main aim of the Virtual Clothing Try-On is the customers can virtually try on shirts, T-shirts, Pants or any kind of dress by using their camera. The virtual clothing try-on technology allows users to swipe through various dresses, ensuring that the particular article of clothing they choose is right for them.

Buyer confidence is especially important with the recent shift toward ecommerce, as online shopping precludes the "experience" stage of in-person shopping.

3.3.2 SCOPE

Functional Requirements:

1. The user should be able to give input through camera.
2. The user should be able to generate live picture with selected garment.
3. The user should be able to view themselves in different poses.
4. The user should be able to browse through and selected clothing items.
5. The user should be able to add garments to their wardrobe.
6. The user should be able to view and select garments from their wardrobe.
7. The user should be able to browse the web for garments from the platform.
8. The user should be able to try different clothing items at same time.

Non-Functional requirements:

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

Time: Tasks should be executed in a timely manner. Tasks in progress should terminate if loading takes too long.

Memory: The project should not take a lot of memory space.

3.3.3 OVERALL DESCRIPTION

Virtual try-on for apparel combines computer vision, artificial intelligence, recommendation algorithms, augmented reality (AR), or virtual reality (VR) to provide an immersive user experience.

Customers may try on items via the internet at home using virtual fitting technology and a camera-equipped gadget. As the camera-equipped device captures and demonstrate how the object would appear on the customer's body. If you have used Snapchat, then you know how this works.

Fashion is, primarily, a physical experience. According to a survey, 82% of internet customers prefer to see and touch things before purchasing. Using virtual fitting rooms allows consumers to preview how an item will appear on their body, but they can't experience the material.

4. DESIGN

4.1 UML DIAGRAMS

4.1.1 USE CASE DIAGRAM

Use Case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. It can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

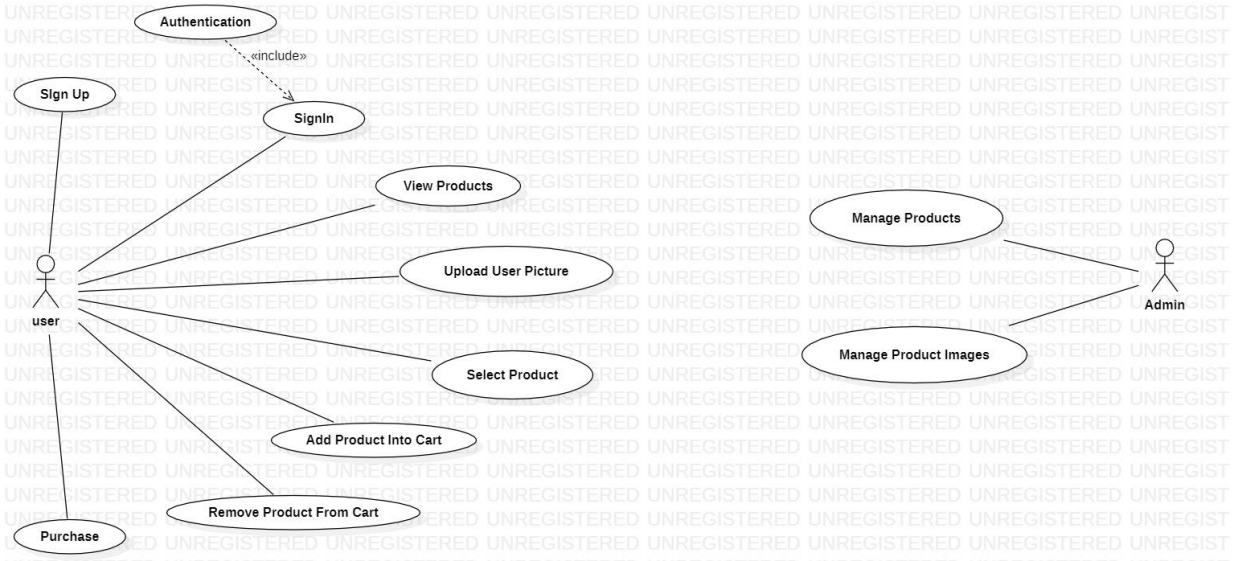


Fig 4.1.1 Use Case Diagram

4.1.2 CLASS DIAGRAM

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

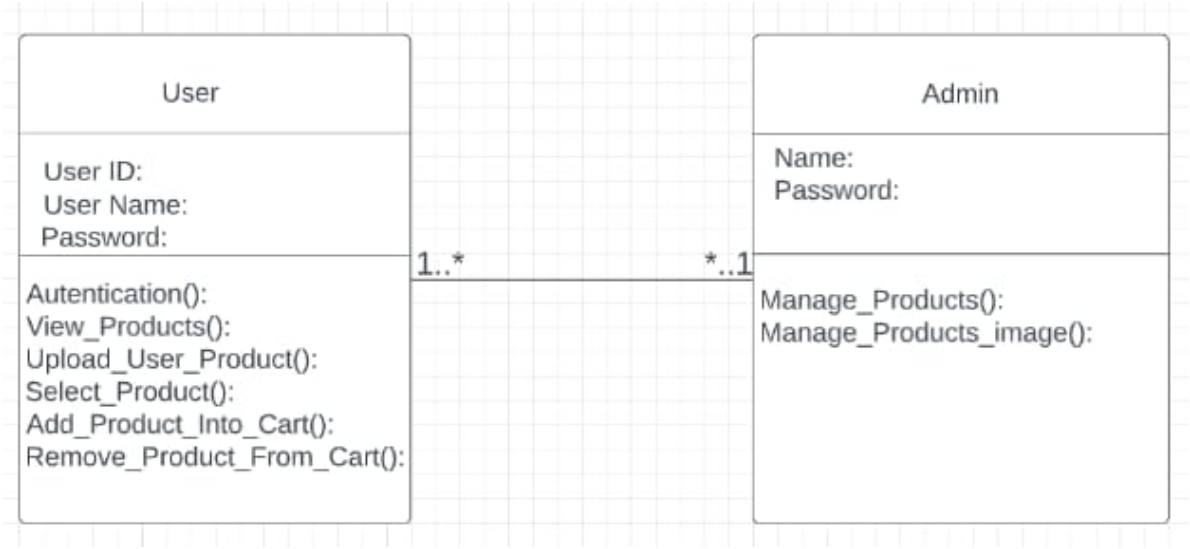


Fig 4.1.2 Class Diagram

4.1.3 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. It is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

Purpose of Activity Diagrams

The basic purposes of activity diagrams are similar to other four diagrams. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

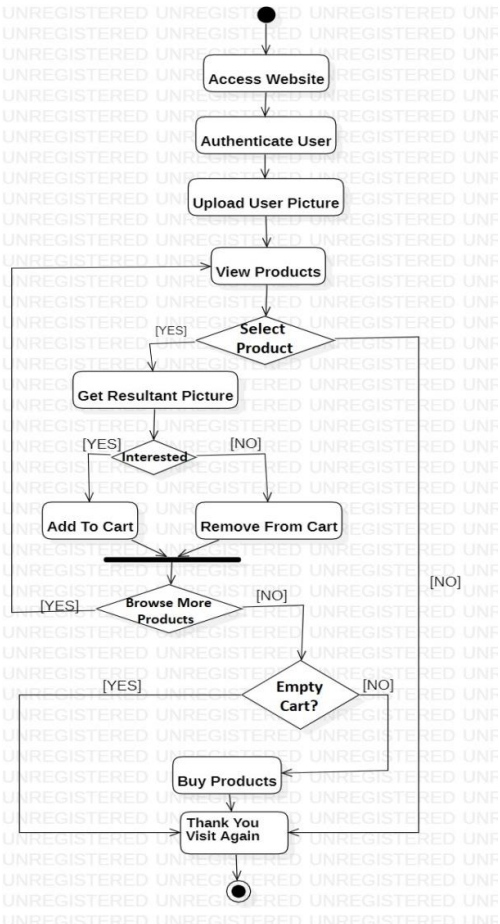


Fig 4.1.3 Activity Diagram

4.1.4 SEQUENCE DIAGRAM

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios.

It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

Purpose of a Sequence Diagram

To model high-level interaction among active objects within a system.

To model interaction among objects inside a collaboration realizing a use case.

It either models' generic interactions or some certain instances of interaction.

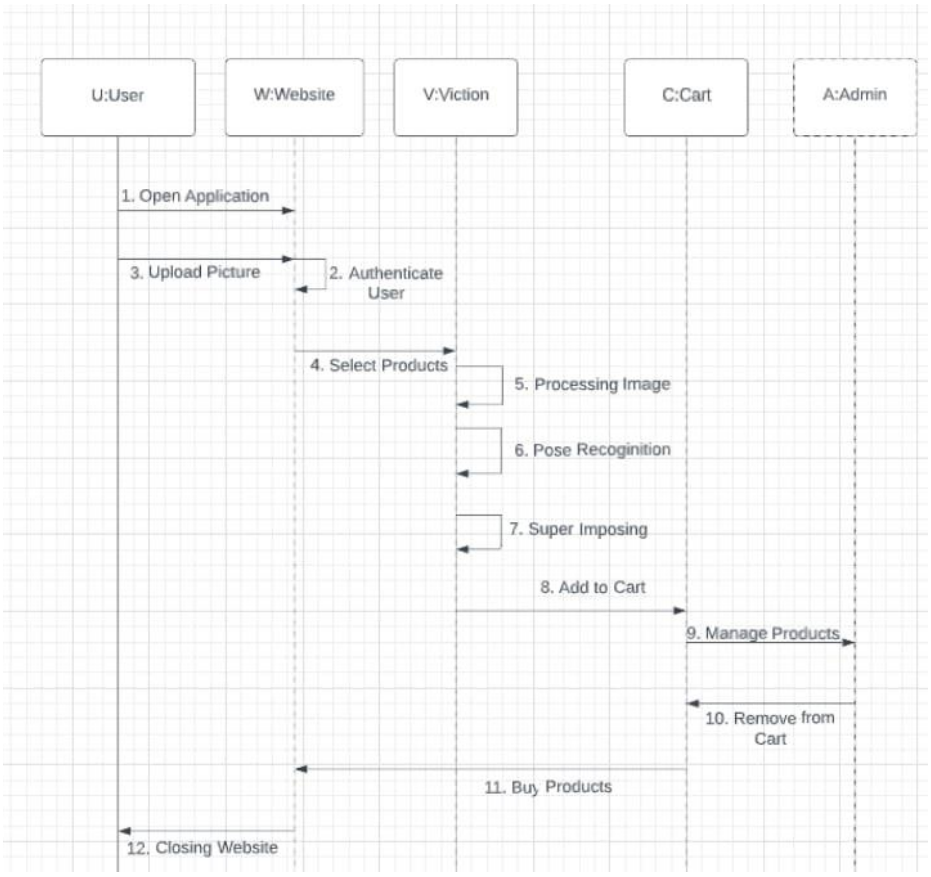


Fig 4.1.4 Sequence Diagram

5. IMPLEMENTATION

5.1 MODULES USED

A module is a collection of source files and build settings that allow you to divide your project into discrete units of functionality. Your project can have one or many modules, and one module may use another module as a dependency. You can independently build, test, and debug each module.

Additional modules are often useful when creating code libraries within your own project or when you want to create different sets of code and resources for different device types, such as phones and wearables, but keep all the files scoped within the same project and share some code. Below are some of the modules we used in our project and their descriptions.

5.2 MODULES DESCRIPTION

Pandas:

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users. Data set merging and joining. Flexible reshaping and pivoting of data sets. Provides time-series functionality. Powerful group by functionality for performing split-apply-combine operations on data sets.

Flask:

Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. [4] Advantages of using Flask framework are:

1. There is a built-in development server and a fast debugger provided.
2. Lightweight
3. Secure cookies are supported.
4. Templating using Jinja2.
5. Request dispatching using REST.
6. Support for unit testing is built-in

CV2:

OpenCV is an open-source library which is very useful for computer vision applications such as video analysis, CCTV footage analysis and image analysis. OpenCV is written by C++ and has more than 2,500 optimized algorithms. When we create applications for computer vision that we don't want to build from scratch we can use this library to start focusing on real world problems.

OS:

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.

Sys:

The sys module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment. It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter.

The sys modules provide variables for better control over input or output. We can even redirect the input and output to other devices. This can be done using three variables –

stdin - It can be used to get input from the command line directly. It is used for standard input. It internally calls the input () method. It, also, automatically adds '\n' after each sentence.

stdout - A built-in file object that is analogous to the interpreter's standard output stream in a file.

PIL:

PIL is an additional, free, open-source library for the Python programming language that provides support for opening, manipulating, and saving many different images file formats.

Thread:

The threading module exposes all the methods of the thread module and provides some additional methods. The methods provided by the Thread class are as follows −

1. run() − The run() method is the entry point for a thread.
2. start() − The start() method starts a thread by calling the run method.
3. join([time]) − The join() waits for threads to terminate.
4. isAlive() − The isAlive() method checks whether a thread is still executing.
5. getName() − The getName() method returns the name of a thread.
6. setName() − The setName() method sets the name of a thread.

Dlib:

Dlib for face detection and facial landmark detection. The frontal face detector in dlib works really well. It is simple and just works out of the box. [5] This script requires two command line arguments.

1.input image

2.model weights

Imutils:

A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much easier with OpenCV and both Python 2.7 and Python 3.

Math:

Python provides the math module to deal with such calculations. Math module provides functions to deal with both basic operations such as addition (+), subtraction (-), multiplication (*), division (/) and advance operations like trigonometric, logarithmic, exponential functions. Math module provides various the value of various constants like pi, tau.

5.3 INTRODUCTION TO TECHNOLOGIES USED

Programming Framework:

Python:

Python was used to implement the models used in the virtual try-on solution. It offers libraries and packages adequate for Deep Learning tasks.

Flask Facial:

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. Flask Facial is a Web-Based Application for Detection of Faces on video using Flask OpenCV and Face_Recognition.

Fig 5.3 a) Python b) OpenCV c) Flask

Body Recognition:

OpenCV:

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library.

Dlib, Haar Cascades Dataset:

Dlib dataset is a landmark's facial detector with pre-trained models, the dlib is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person's face.

These points are identified from the pre-trained model where the iBUG300-W dataset was used.

Haar Cascades dataset is used for Face, Profile, Eyes, Smile, Upper Body Detection. It is an Object Detection Algorithm used to identify faces in an image or a real time video.

Front End:

HTML & CSS:

HTML (Hyper Text Markup Language) HTML is a syntax used to format a text document on the web.CSS (Cascading Style Sheets) is a style sheet language used for describing the look and formatting of a document written in a markup language.

Fig 5.3 d) HTML, CSS & JavaScript

JavaScript:

Java Script (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side. It is used to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. Java Script is used to create pop-up windows displaying different alerts in the system like “User registered successfully”,” Product added to cart” etc.

5.4 SAMPLE CODE

main.py file

```
import pandas
from flask import Flask, render_template, Response, redirect, request
from camera import VideoCamera
import os

app = Flask(__name__)

CART = []

pandas
@app.route('/checkOut')
def checkOut():
  return render_template('checkout.html')



@app.route('/tryon/<file_path>', methods=['POST', 'GET'])
def tryon(file_path):
    file_path = file_path.replace(',', '/')
    os.system('python tryOn.py ' + file_path)
    return redirect('http://127.0.0.1:5000/', code=302, Response=None)

@app.route('/tryall', methods=['POST', 'GET'])
def tryall():
    CART = request.form['mydata'].replace(',', '/')
    os.system('python test.py ' + CART)
    return render_template('checkout.html', message='')
@app.route('/')
```

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

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

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

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

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

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route("/cart/<file_path>", methods=['POST', 'GET'])
```

```
def cart(file_path):
    global CART
    file_path = file_path.replace(',', '/')
    print("ADDED", file_path)
    CART.append(file_path)
    return render_template("checkout.html")

@app.route('/video_feed')
def video_feed():
    return Response(gen(VideoCamera()),
                    mimetype='multipart/x-mixed-replace; boundary=frame')

if __name__ == '__main__':
    app.run()
```

camera.py file

```
import cv2

class VideoCamera(object):
    def __init__(self):

        self.video = cv2.VideoCapture(0)

    def __del__(self):
        self.video.release()

    def get_frame(self):
        success, image = self.video.read()
        ret, jpeg = cv2.imencode('.jpg', image)
        return jpeg.tobytes()
```

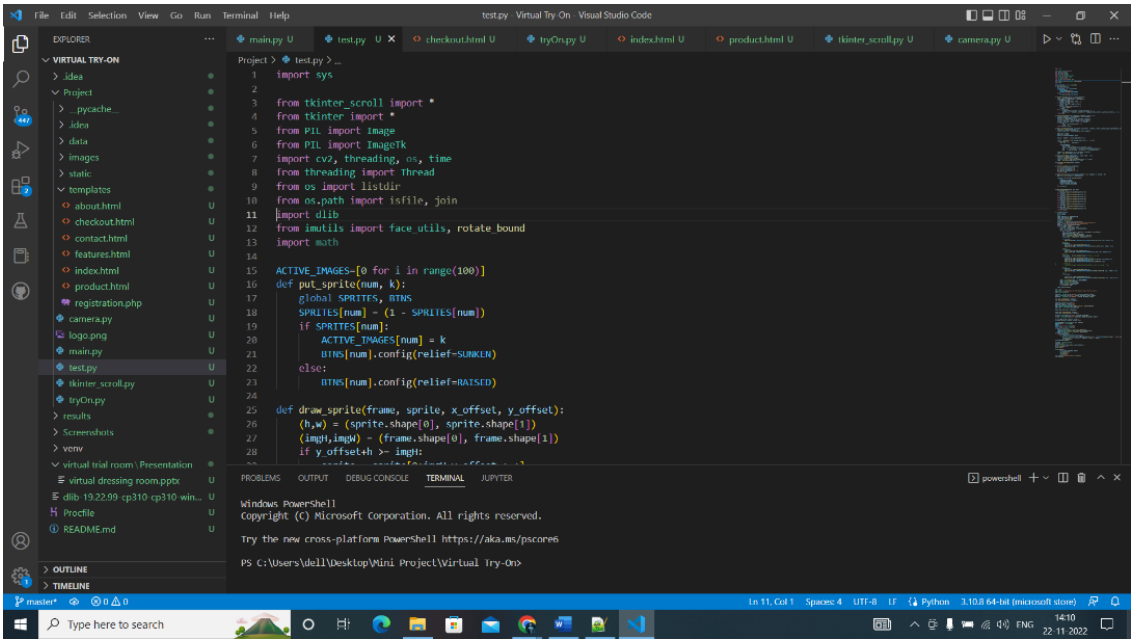


Fig 5.4 a) Implementation Code

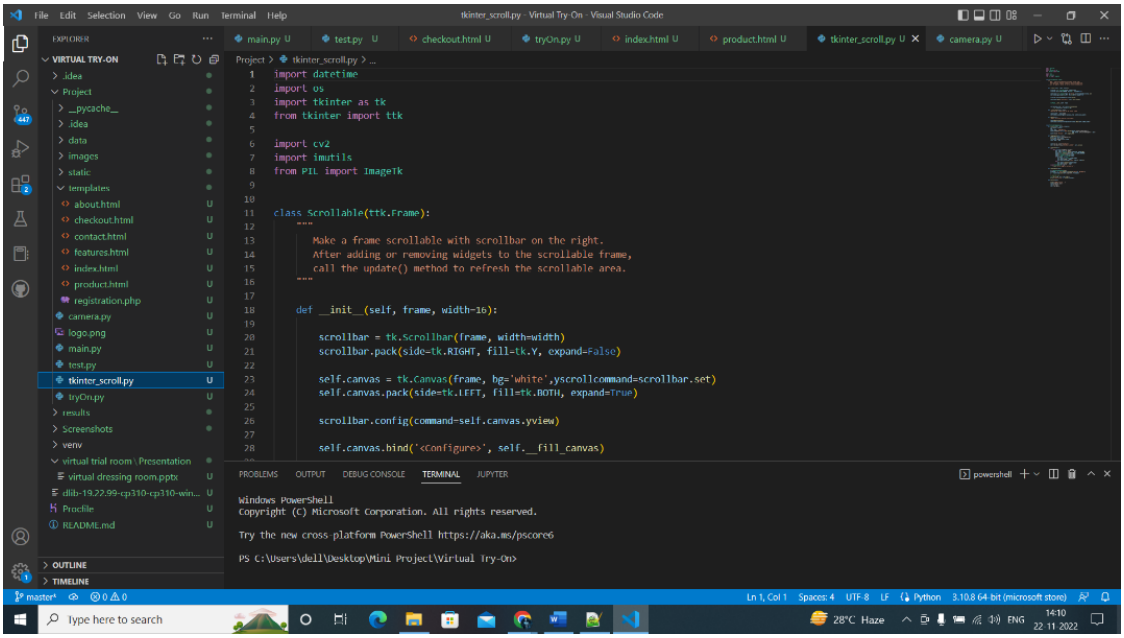
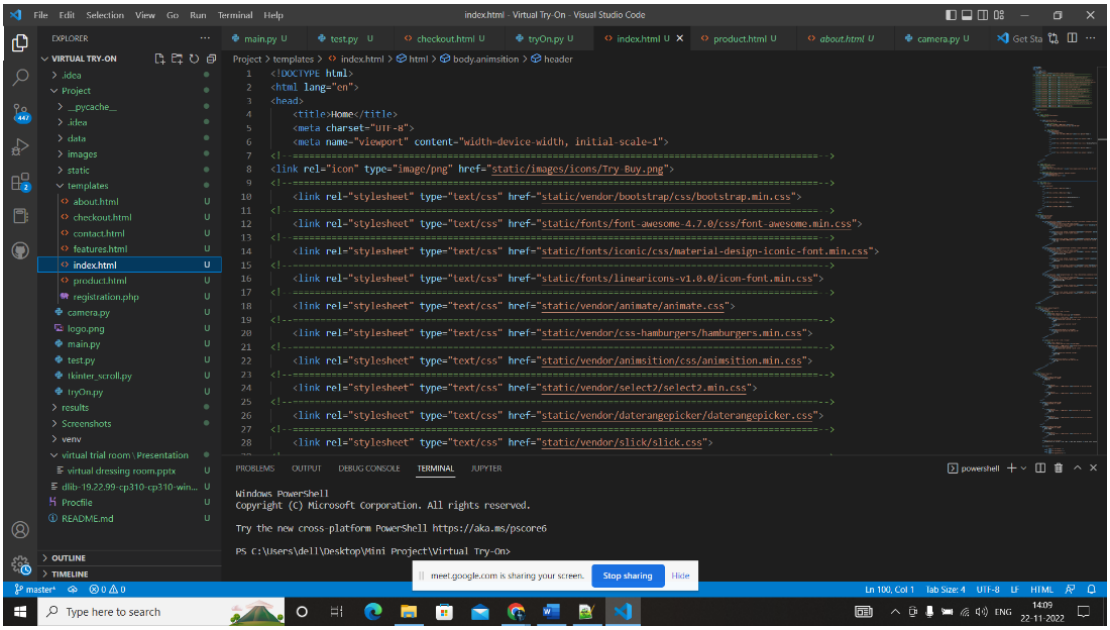


Fig 5.4 b) Implementation Code



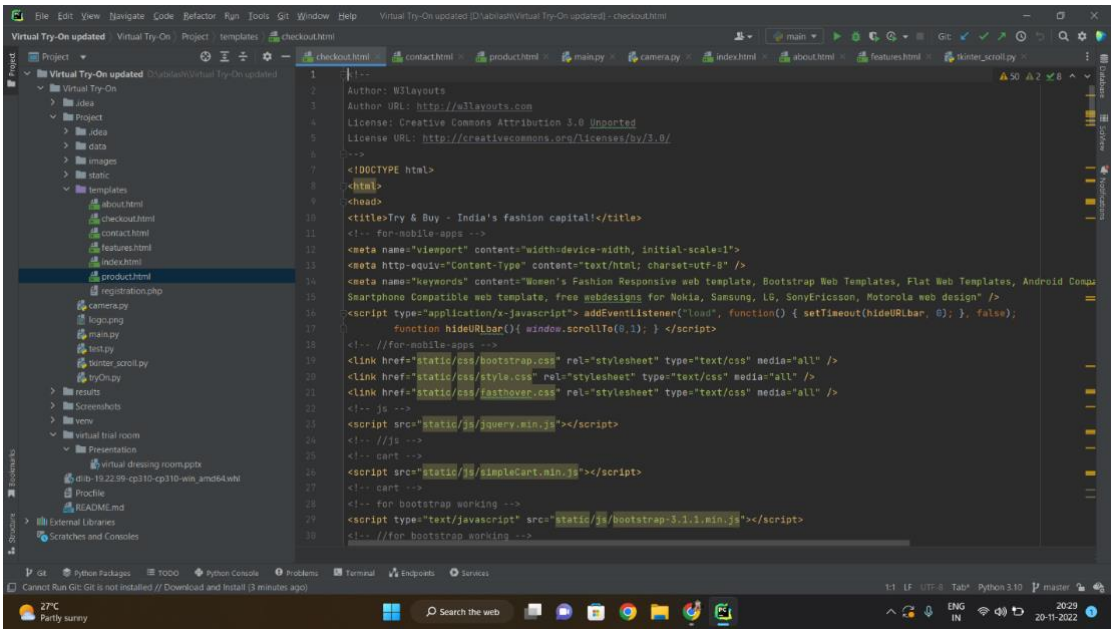


Fig 5.4 c) Implementation Code (Webpages)

6. TEST CASES

Test Case	Expected Output	Actual Output	Outcome (Pass/Fail)
Image Capturing (Image/Video)	Captured Image/Video	Captured Image/Video	Pass
Recognize Body	Body-Pose Model	Body-Pose Model	Pass
Clothes Transfer	Parsing Clothes	Parsing Clothes (Torso)	Pass
Real Time Output	Try-On Image/Video	Try-On Image/Video	Pass

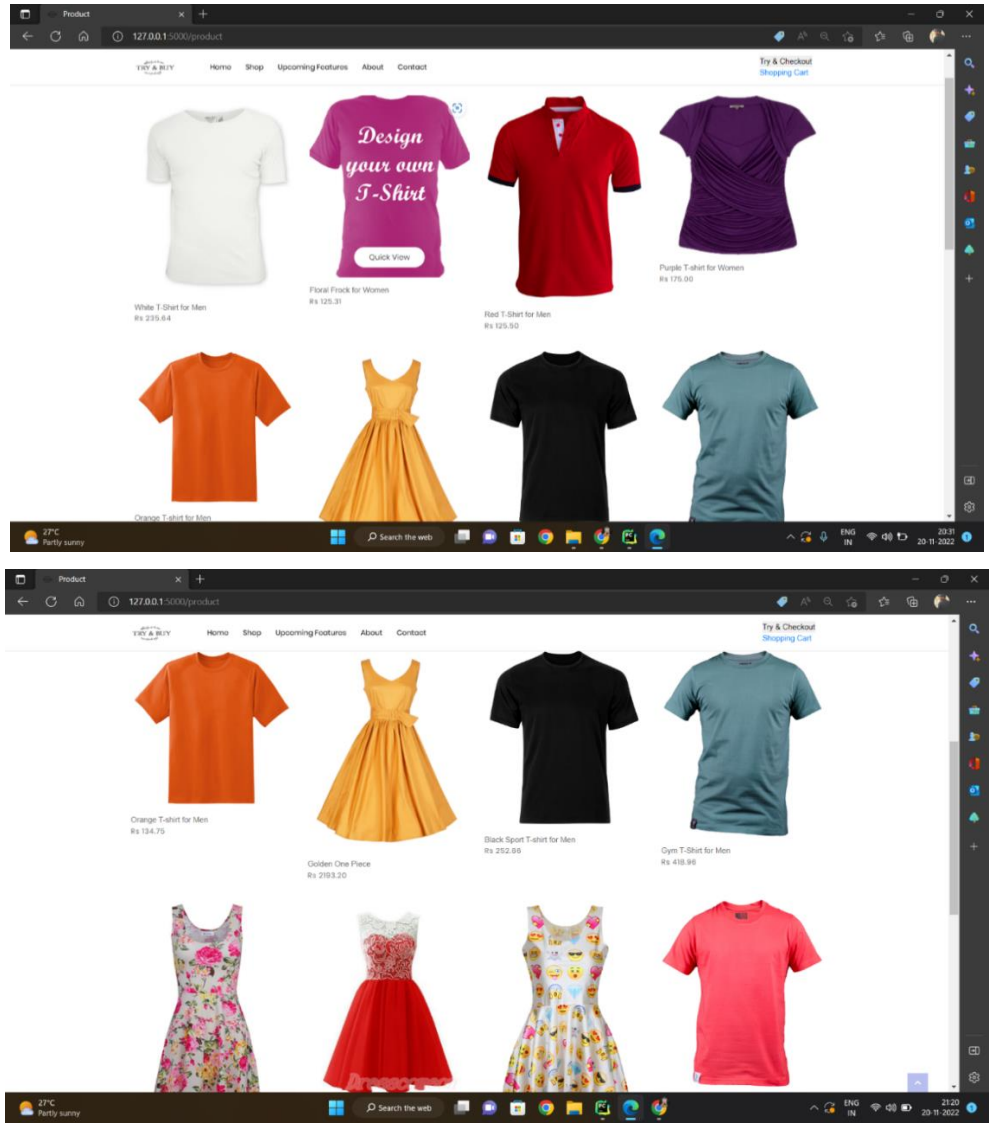
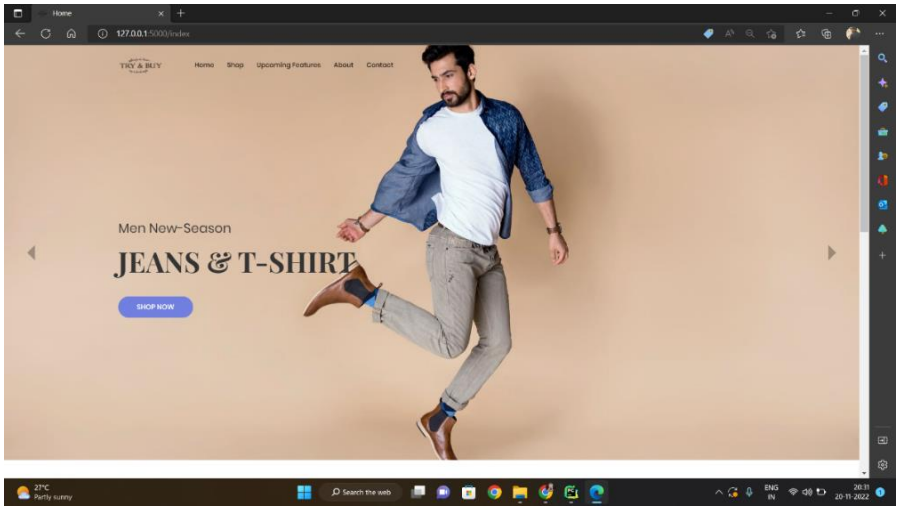
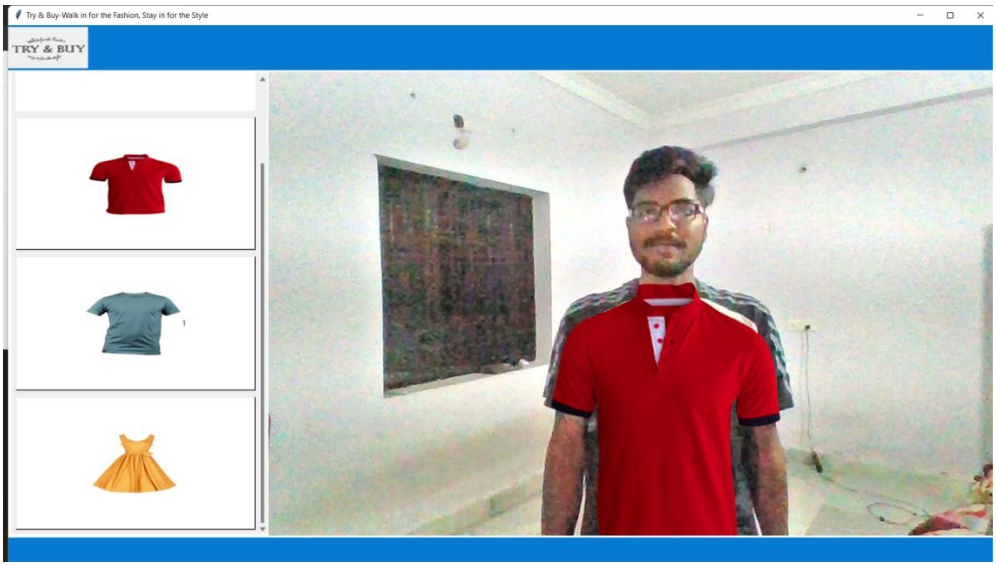


Fig 6 Pictures of the clothes that are displayed on our website

7. SCREENSHOTS





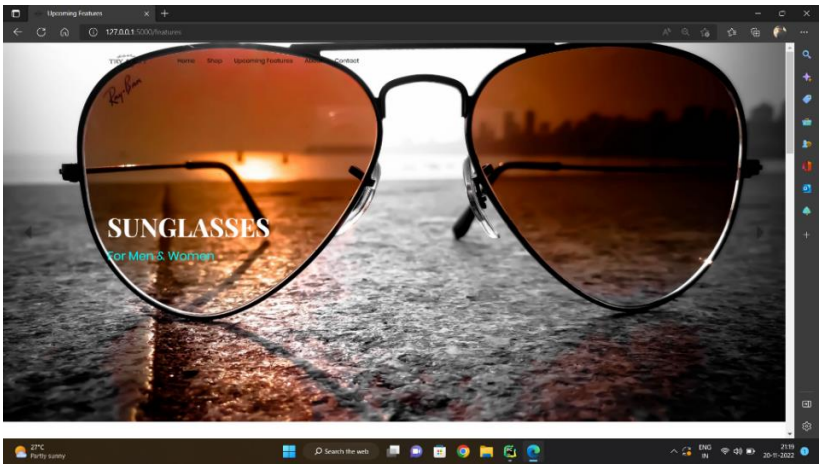


Fig 7 Screenshots of our Project

8. CONCLUSION

Our project helps users to try their clothes virtually and find out whether it suits them or not. Our project has a user-friendly website with many features. It also helps the multiple users to try the same clothes at the same time. Through our project people can able to try their clothes virtually during online shopping. We also identify that our design choice using optical flow improves temporal smoothness. We introduced several methodological innovations to improve the quality of image synthesis, and demonstrated that our method is able to generate substantially better realistic looking virtual try-on images than the state-of-the-art methods.

9. FUTURE ENHANCEMENTS

This project offers a lot of scope for adding newer features. Since our project is Virtual Clothing Try-On, we can add more features like accessories such as watches, sunglasses, necklaces, ear rings and also shoes. Here we can find all the variety collections of accessories with different colors and patterns of different brands. This is a user-friendly website you can easily check whether those accessories that will fit on you and those which match your style.

10.BIBLIOGRAPHY

[1] CloTH-VTON+: Clothing Three-Dimensional Reconstruction for Hybrid Image-Based Virtual Try-On (Matiur Rahman Minar, Thai Thanh Tuan, Heejune Ahn-Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea)

[2] A Mixed Reality Virtual Clothes Try-On System (Miaolong Yuan, Ishtiaq Rasool Khan, Farzam Farbiz, Senior Member, IEEE, Susu Yao, Arthur Niswar, and Min-Hui Foo)

[3] ShineOn: Illuminating Design Choices for Practical Video-based Virtual Clothing Try-on (Gaurav Kuppa, Andrew Jong, Xin Liu, Ziwei Liu, Teng-Sheng Moh)

[4]https://www.analyticsvidhya.com/blog/2021/10/easy-introduction-to-flask-framework-for-beginners/

[5] https://towardsdatascience.com/cnn-based-face-detector-from-dlib-c3696195e01c

[6] High-Resolution Virtual Try-On with Misalignment and Occlusion-Handled Conditions
Sangyun Lee, Gyojung Gu, Sunghyun Park, Seunghwan Choi, and Jaegul Choo (Soongsil University, Korea Advanced Institute of Science and Technology & Nestyle Inc.)