Project Report

On

Virtual Trial Room

Submitted
In partial fulfilment
For the award of the Degree of

PG-Diploma in Artificial Intelligence

(C-DAC Noida)

Guided By:

Mr. Dominik Irudhayaraj

Submitted By:

Amruta Makar (210920528005)
Deovart Verma (210920528015)
Megha Pandey (210920528025)
Sagar Thorat (210920528033)

Centre for Development of Advanced Computing

(C-DAC Noida)

Acknowledgement

This is to acknowledge our indebtedness to our Project Guide, **Mr. Dominik Irudhayaraj,** C-DAC Noida, for his constant guidance and helpful suggestion for preparing thisproject **Virtual Trial Room.** We express our deep gratitude towards him for his inspiration, personal involvement, constructive criticism that he provided us along with technical guidance during the course of this project.

We take this opportunity to thank Head of the department **Mr. Ravi Payal** forproviding us such a great infrastructure and environment for our overall development.

It is our great pleasure in expressing sincere and deep gratitude towards **Mrs. Saruti Gupta** (Course Coordinator, e-DAI) for their valuable guidance and constant support throughout this work and help to pursue additional studies.

Also, our warm thanks to **C-DAC Noida**, which provided us this opportunity to carryout, this prestigious Project and enhance our learning in various technical fields.

Amruta Makar (210920528005)
Deovart Verma (210920528015)
Megha Pandey (210920528025)
Sagar Thorat (210920528033)

ABSTRACT

With the introduction of smart phones and tablets, we can enjoy online shopping anytime and while sitting in any part of the world. Online shopping has certainly replaced the traditional way of shopping for daily goods and clothing. When we choose online shopping, we get the benefit of credibility. Today, almost every online store offers cash on delivery, free shipping and discounted prices. These online shopping stores eliminate the hassles of parking, getting stuck in traffic jams and standing in long queues for billing. They have also benefited those people who always complain of shortage of time. This is the reason; majority of the people have turned to online shopping. Here, they enjoy easy access to an attractive price range, prompt customer support, and free home delivery. There is no doubt that these are some of the attractive features that catches the attention of the consumers. Although there is one small issue that could make people lose interest in online shopping, it might not be possible to try-on clothes in such cases. Our motive here is to increase the time efficiency and improve the accessibility of clothes try on by creating a virtual dressing room environment.

Our proposed approach is mainly based on segmentation of the user image from the video stream using Haar Cascade Classifier, creation of masks of cloths and alignment of masks on the user. Extraction of user allows us to create a virtual environment by isolating the user area from the video stream and superimposing it onto a virtual environment in the user interface. We use the Haar Cascade Classifier, scaling and rotation in order to align the 2D cloth models with the user. Finally, the mask of cloth is superimposed on the user in real time.

Table of Contents

Chapter 1

Introduction

1.1 Introduction

Trying clothes in clothing stores is usually a time-consuming activity. Besides, it might not even be possible to try-on clothes in such cases as online shopping. Our motivation here is to increase the time efficiency and improve the accessibility of clothes try on by creating a virtual dressing room environment. In circumstances like Covid-19 where we can't physically visit the store for shopping allowing them to virtually try on clothing's before purchasing them. The problem is simply the alignment of the user and the cloth models with accurate position, scale, rotation and ordering. First of all, detection of the user and the body parts is one of the main steps of the problem. The problems can be brilliantly managed by means of simple software like OpenCV. Extraction of user image in order to create a virtual environment by isolating the user area from the video stream, creation of mask from the cloth and superimposing them in a virtual environment onto the user. Thus, the user can see a virtual image of themselves in the costume of their preference. The usage of web camera makes it easier on the cost for the users of online shopping. The implementation by OpenCV makes it more platform independent and portable and there by accessible in any form of device.

General idea about our model-

Our approach can be summarized as follows:

- Extraction of the user from the video stream by using Haar Cascade Classifier
- 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.

A sample application with user interface is developed to test practically the performance. The user interface allows the user to choose a dress by means of dropdown menu.

1.2 Objective and specification:

The project aims to deliver a software program, exploiting latest OpenCV libraries and functions. The program works towards the objective of Image Recognition, Image Segmentation and Classification. Using the model thus we created a virtual trial room.

So, the objective of this project is to create a virtual trial room for humans to try various clothing virtually, instead of trying on in real.

The different processes which can be used in this project are:

- Image Recognition
- Image Segmentation and mask creation
- Classification
- Mask superimposition

Chapter 2

Methodology and Techniques

2.1 Approach & 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.

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 to provide 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.

2.2 Image Processing

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)
```

Fig 2.1 Video capture from web camera

Providing images of clothes as input for masking.
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
```

Fig 2.2 Attire Image Processing

2.3 Mask Creation

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)
```

Fig 2.3 Mask Creation

2.3 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')
```

Fig 2.4 User detection using Haar Cascade Classifier

2.4 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 al
mask = cv2.resize(orig_mask, (pantWidth,pantHeight), interpolation = cv2.INTER_AREA)
mask_inv = cv2.resize(orig_mask_inv, (pantWidth,pantHeight), interpolation = cv2.INTER_AREA)
```

Fig 2.5 Resizing of the mask

2.1 Mask superimposition

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)
```

Fig 2.6 Mask Superimposition

Chapter 3

Implementation

3.1 Implementation

1. Use of Python Platform for writing the code with **OpenCV** and **FLASK**.

2. Hardware and Software Configuration:

Hardware Configuration:

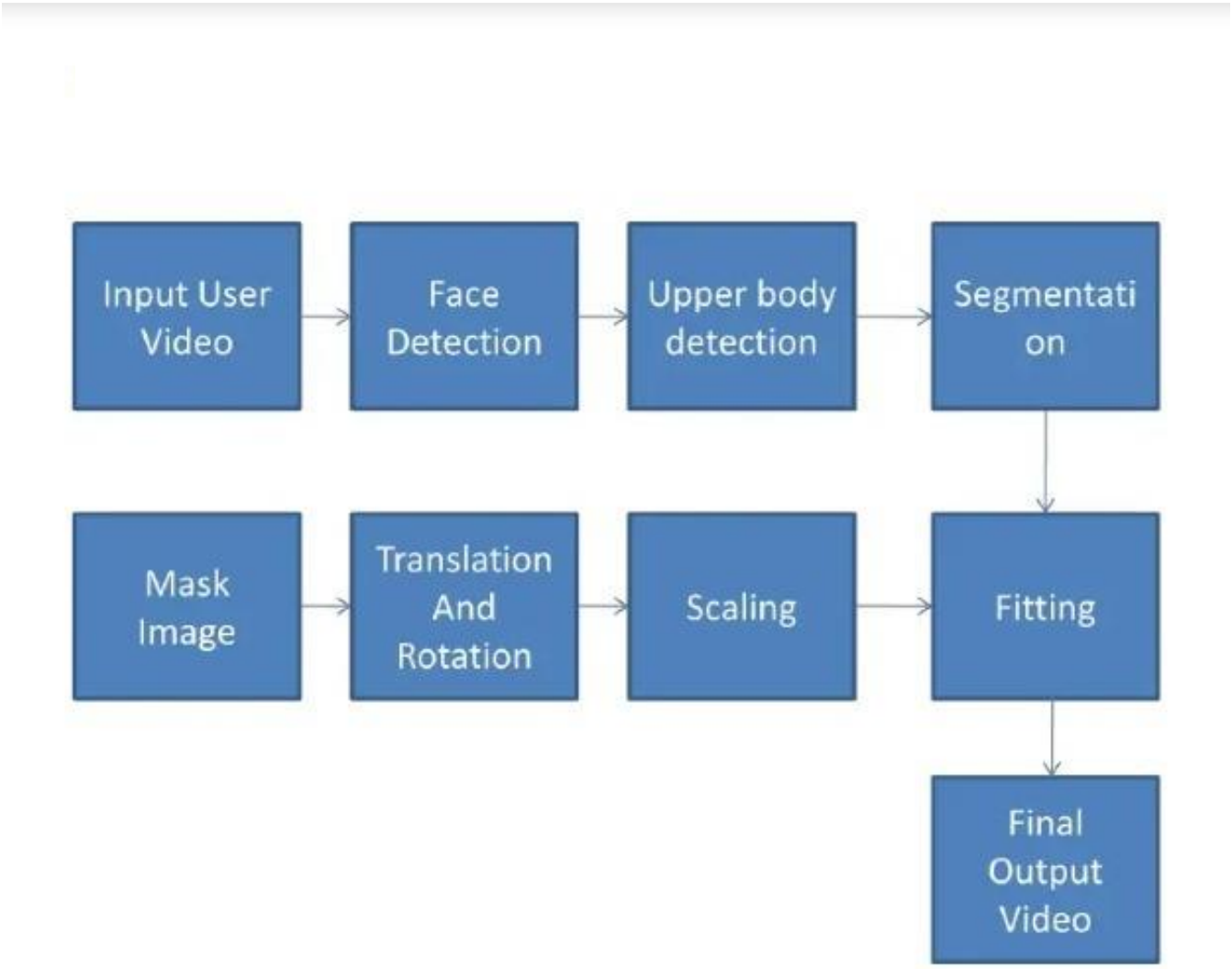
- CPU: 8 GB RAM, Quad core processor
- GPU: 4GB RAM **Nvidia's GTX 1650Ti**
- Laptop webcam

Software Required:

- **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.

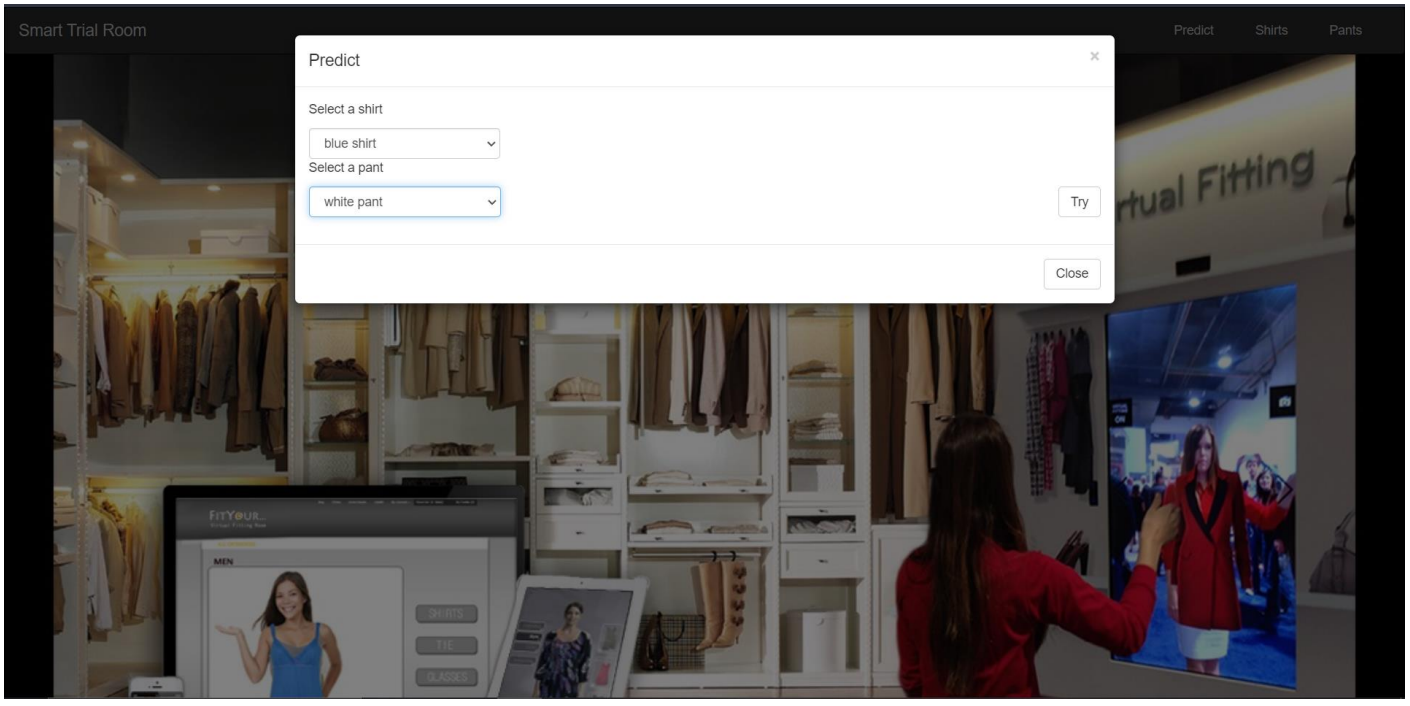
- **Spyder**: Spyder, the Scientific Python Development Environment, is a free open-source integrated development environment (IDE) that is included with Anaconda written in Python, for Python, and designed by and for scientists, engineers and data analysts. It includes editing, interactive testing,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

3.2 Flow Chart of Application



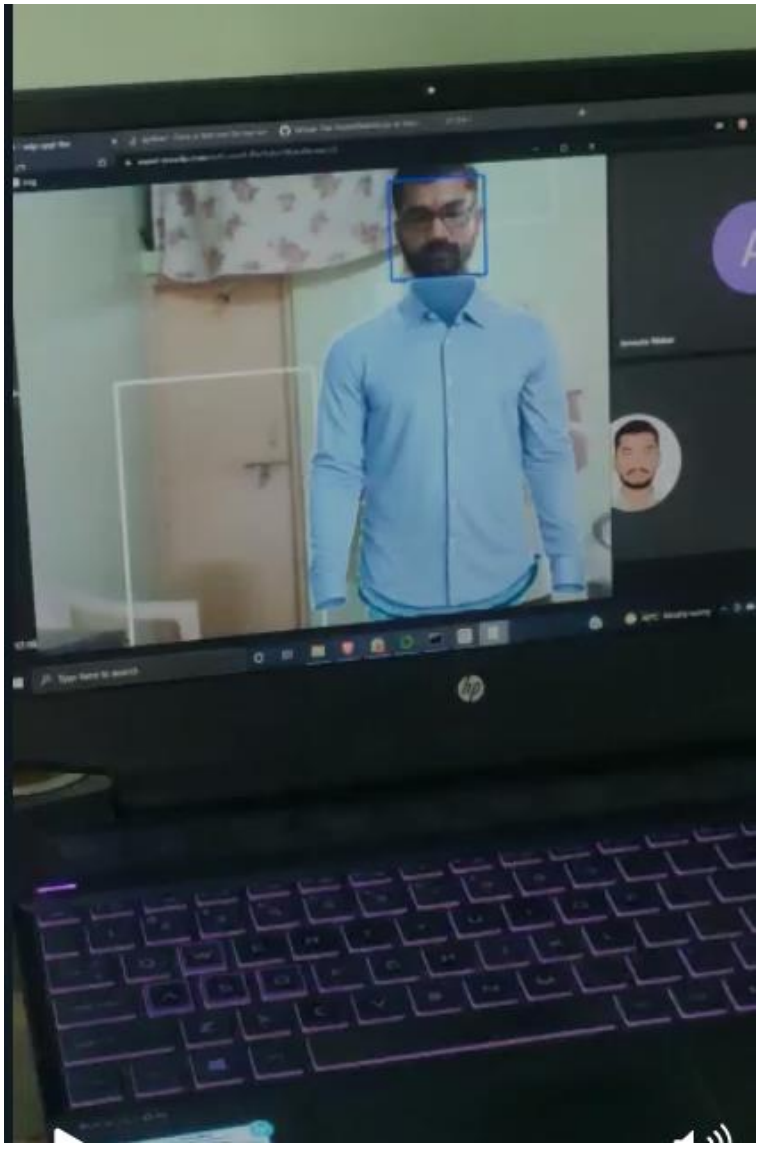
User Interface of our project using Flask:

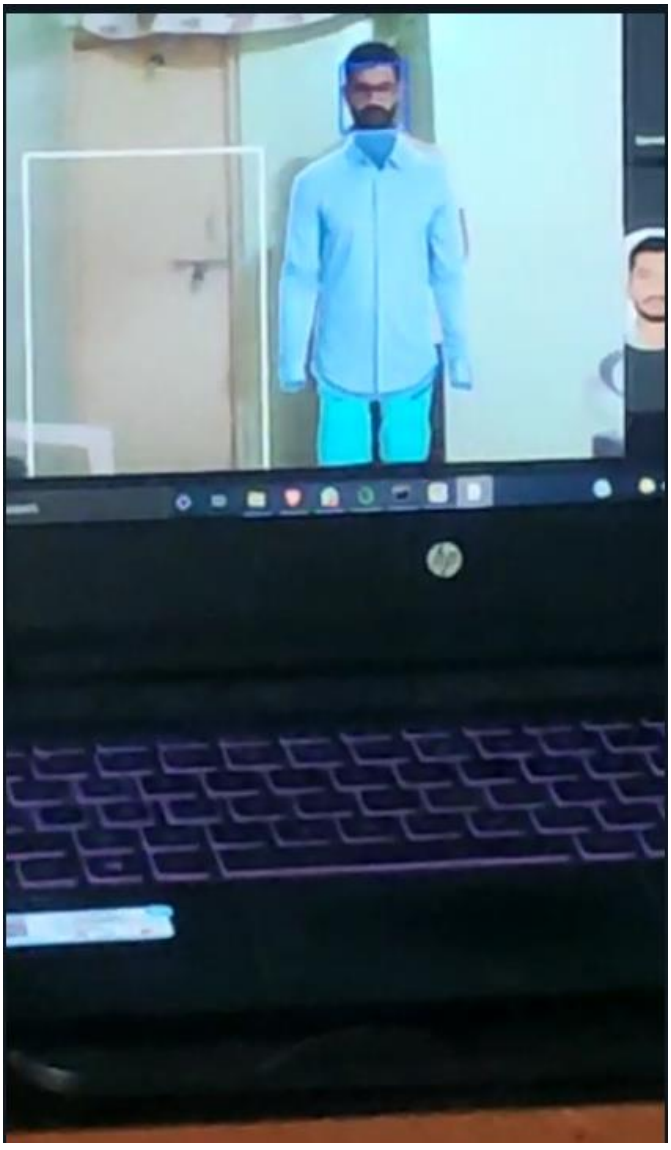
User Panel:



Chapter 4

Result





Chapter 5

Conclusion

5.1 Conclusion and Future Scope:

- In this Python project, we have built a Virtual Trial Room. We used OpenCV and Haar Cascade Classifier to detect user and using ROI for better preciseness.
- This model can be integrated with E-commerce platforms.
- Contactless clothing try on can be achieved with the help of this model in Covid-19 like situations.
- This model as of now can be implemented for 2-D application only.
- There are lots of scope for improvement in this model then in further iterations we will try to build the same on a live video feed of the user, for a 3D modelling.

"The model can be adapted for scalability, better convergence, and better accuracy."

Chapter 6

References

References:

- **https://docs.opencv.org/master/d9/df8/tutorial_root.html**
- **M2E-Try On Net: Fashion from Model to Everyone(https://arxiv.org/pdf/1811.08599v1.pdf)**
- **https://www.academia.edu/6940511/A_REAL_TIME_VIRTUAL_DRESSING_ROOM_APPLICATION_USING_OPENCV_A_PROJECT_REPORT**
- **https://www.academia.edu/17537494/Virtual_Fitting_Room_using_Webcam**