VIRTUAL CLOTHES FITTING SYSTEM USING IMAGE PROCESSING

A PROJECT REPORT SUBMITTED BY

T.S.N DASA

(S / 18 / 335)

for the partial fulfillment for the requirement for the course of

CS 314 Image Processing Practical

To the **Department of Statistics and Computer Science** of the

FACULTY OF SCIENCE
UNIVERSITY OF PERADENIYA
SRI LANKA
2023

DECLARATION

I do hereby declare that the work reported in this project report was exclusively carried out by me under the supervision of Prof. Amalka Pinidiyaarachchi. It describes the results of my own independent work except where due reference has been made in the text. No part of this project report has been submitted earlier or concurrently for the same or any other degree.

Date: ……. ……

Signature of the Candidate

Certified by:

1. Instructor: **Mr. Rochana Obadage**

   Date: ……. Signature:…….

2. Supervisor: **Prof. Amalka Pinidiyaarachchi**

   Date: ……. Signature:…….

3. Head of the Department: **Dr. Sachith Abeysundara**

   Date: ……. Signature:…….

ABSTRACT

VIRTUAL CLOTH FITTING SYSTEM USING IMAGE PROCESSING

T.S.N DASA

Department of Computer Science &Statistics, University of Peradeniya, Peradeniya, Sri Lanka

Cloth Designing and Cloth fitting is a process that takes a lot of time and effort from many people such as Designers and models when working physically. And when customers are buying clothes they have to wait on a queue and get into the fitting rooms repetitively. But recently after the pandemic time People got used to online shopping systems Virtual Cloth fitting systems are more widely used and time-saving. This enhances the online shopping experience and significantly reduces the frustration associated with ill-fitting clothing, ultimately leading to higher customer satisfaction and increased sales for retailers. Customers can try out the clothing item at home and check whether those color clothes match with each other.In conclusion, the Virtual Cloth Fitting System with Image Processing represents a pioneering stride toward a future where online clothing shopping is not just convenient but also remarkably personalized, ensuring a perfect fit for every individual.

ACKNOWLEDGEMENTS

Beginning with heartfelt appreciation, I would like to extend my thanks to all those whose dedication and contributions have been pivotal in the realization of the Virtual Fitting System project.

First and foremost, I am thankful to Prof. Amalka Pinidiyaarachchi for providing the theoretical knowledge in image processing and enabling me to understand the concepts related to it without much hassle.

I would also like to thank Mr. Rochana Obadage and Mr. Kansaji Kotuwage for equipping me with the practical knowledge necessary to carry out my project while also being mentoring throughout the process.

My sincere gratitude is extended to Dr. Sachith P. Abeysundara, the head of the department of Statistics and Computer Science, for providing the necessary facilities for the successful completion of this project.

Finally, I would like to express my gratitude to my family and colleagues for their kind support and encouragement throughout this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI: Artificial Intelligence
AR: Augmented Reality

CHAPTER 01

INTRODUCTION

1.1. INTRODUCTION

Online shopping has transformed the retail landscape, offering unparalleled convenience and access to a vast array of products. However, one persistent challenge has remained – the difficulty of ensuring that purchased clothing items matches and meet the customer's expectations. Mismatching clothing items are a leading cause of customer dissatisfaction and product returns, impacting both retailers and consumers. To address this issue, we present the Virtual Clothes Fitting System, an innovative solution that leverages technology to provide a seamless and accurate virtual fitting experience, bridging the gap between online and in-store shopping. And you can experience this system at home with ease.

1.2. PROBLEM STATEMENT

The problem of sizing and fit is a longstanding and widespread concern in the world of online fashion retail. Customers often face uncertainty when selecting clothing items, as they cannot physically try them on. This uncertainty leads to increased returns and customer frustration, resulting in higher operational costs for retailers and a diminished shopping experience for consumers. The systems that are existing not created to match two clothing items to them user given image and see what is best to wear for the user. Because the image that they suggest may be not be the appearance of the user so the clotting items can be lead to a mis-match.

1.3. SOLUTION

To reduce these problems and Create more user friendly system by using image processing and with by taking user more user inputs and know about their preference correctly with they have choose what is their image and what are the clothes they want to wear by resizing the cloths image that they have given to the system.

CHAPTER 02

RELATED WORK

Related works in the field of virtual clothing fitting systems and similar technologies have emerged as the industry seeks to address the challenges of online shop owners. There are lot of online shopping web sites and apps that use machine learning technology and 3D model bodies to try the clothes on to their in build clothes which have been created to the same size ,with cropped correctly and with the image noise and background noise reduced.

So when the customer came to the online shopping web site or app they can choose between the in-build clothing items that are selling right now. Shoppers can virtually try on clothing items or beauty products without physically touching the items themselves. The augmented reality (AR) or artificial intelligence (AI) technology places the item over live imaging of the customer, so they can check the size, style, and fit of a product they're thinking of buying.

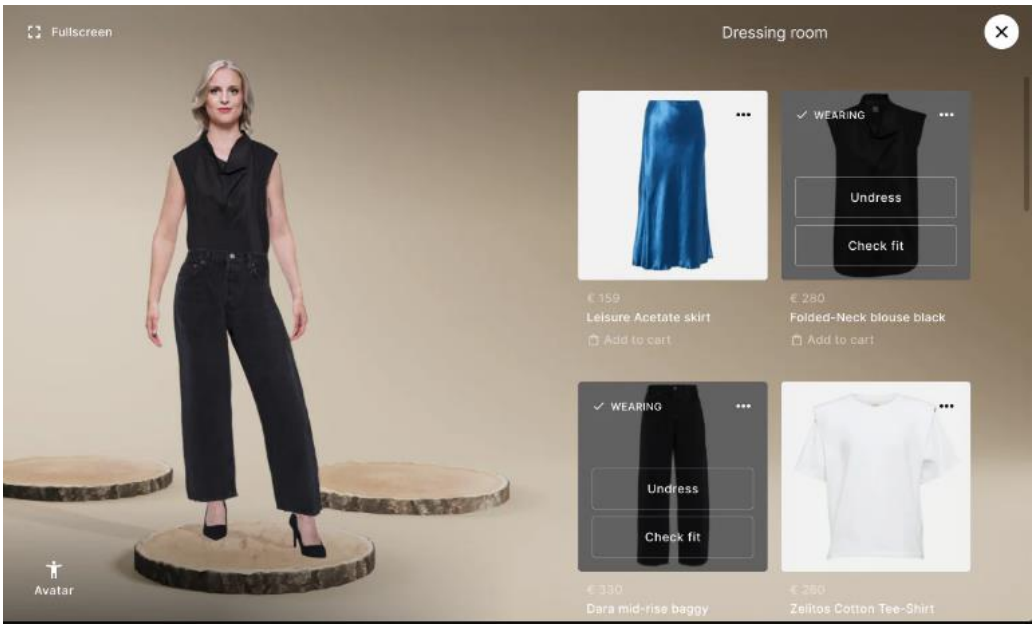

Figure 1: Virtual Cloth Fitting Application using 3D body model

CHAPTER 03

METHODOLOGY

3.1. IMAGE PROCESSING PIPELINE

The figure below shows the image processing pipeline followed by this system.

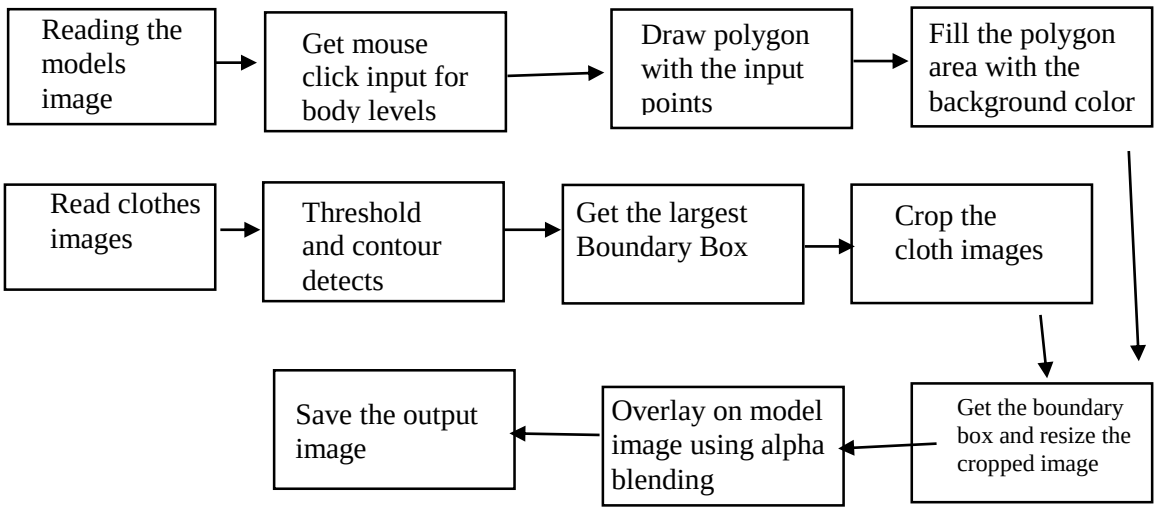

Figure 2: Image processing pipeline flowchart

3.2. USED FUNCTIONS AND THEIR APPLICATIONS

- cv2.imread():To read user images and clothes images
- cv2.cvtColor():To convert color space to another color space
- cv2.shape():To get the height and width of the image
- cv2.findContours():To detect the contours in the image
- cv2.resize():To resize the image to the measured size
- cv2.fillPoly():To fill the polygon with the color
- cv2.bitwise_and(): To get the bitwise AND operation between 2 images
- cv2.bitwise_not():To get the bitwise NOT operation between 2 images
- cv2.copy():To get a copy of image
- cv2.len():To get the size of the array

3.3. USED TOOLS, TECHNOLOGIES, AND LIBRARIES

Libraries used

- Opencv
- Numpy
- matplotlib
- Tkinter
- PIL (Python image library)

For the creation of dataset I obtain the male and female models' images ,kid models images and the clothes images from free image-accessing websites such as shutterstock, istock and freepik. These used images are strictly taken in the white background .

Figure 3: Some of the images in the dataset for the project

3.4. PROCEDURE

First the model images, and clothes images are read to the system and then the user was ask to select the gender of the person. Then as the gender choice if main, there will be available clothing choices as shirt and trousers, short and shirt, shirt and sarrom. As for the female users can choose between blouse and skirt , blouse and trouser, and frock. Then as per their choice overlaying will be done. Before that user will be asked to input the mouse clicks for left shoulder, right shoulder, left hip, right hip, left ankle, right ankle, left knee, right knee left hand and right hand (all 10 mouse click events).Then as per there system will determine the shoulder level ,hip level, knee level ,ankle level and hand level (5 body levels).Then by connecting those points the system will draw a polygon around the body and it will be filled with the color from the background. After that For the clothes images will be threshold and find the contour and get the maximum contour and draw the boundary box for the clothing image and crop it along that box and Blending the transformed clothes image onto the model's image using alpha blending using the above given five spots. And then show the output for the user.

3.5. PROCEDURE EXPLAINED

The images are read using the imread() function. And Form the interface the gender choice and the clothing choice will be taken to the interface and store in the variables.
Then user will be ask to draw polygon and mark levels on the models image as a user input and with this function The it will count until 10 mouse click function and label those points with the chosen naming.

```
def label_point(event):
    global body_polygon, canvas,labeled_body_levels

    if len(labeled_body_levels) >= 10:
        messagebox.showinfo("Info", "You have already labeled 10 body levels.")
        return

    x, y = event.x, event.y


    for level, (lx, ly) in labeled_body_levels:
        if abs(x - lx) < 10 and abs(y - ly) < 10:
            messagebox.showinfo("Info", "This point is close to an existing labeled point.")
            return


    body_polygon[level_var.get()] = (x, y)
    labeled_body_levels.append((level_var.get(), (x, y)))
    canvas.create_text(x, y - 10, text=level_var.get(), fill="red", font=("Helvetica", 10, "bold"))
    label_var.set(f"Labeled {len(labeled_body_levels)} body levels")
```

Figure 4: Function for show labels on the image

Then For the taken points in the body System will connect those points with together and draw the polygon around the body and ask the user to exit from the drawing mode by clicking a button

```
def start_drawing_polygon():
    global body_polygon,canvas,labeled_body_levels

    polygon_window = tk.Toplevel(root)
    polygon_window.title("Draw Body Polygon")

    canvas = tk.Canvas(polygon_window, width=400, height=600)
    canvas.pack()

    if model_image is not None:
        img = Image.fromarray(model_image)
        img = ImageTk.PhotoImage(image=img)
        canvas.create_image(0, 0, anchor=tk.NW, image=img)
        canvas.image = img

    canvas.bind("<Button-1>", label_point)

    body_polygon = {}
    labeled_body_levels = []
    exit_button =tk.Button(polygon_window,text="Exit Drawing Mode",command=polygon_window.destroy)
    exit_button.pack()
```

Figure 5: Function for draw polygon around the body points

Then the system will call out the function for clearing and filling the drawn polygon part with the fillPoly() function. This can be not lead to a best output if the user didn't choose the all 10 points or when those points are out of the clothes boundary or if the selected point are still inside the point on the clothes ,the modes image maybe be not cleared well and some parts of the clothes can be shown of after overlaying on it.

```
def clear_model_image(model_image, body_polygon):
    hand_level = body_polygon['hand']
    hand_neighborhood = model_image[hand_level[1] - 50:hand_level[1] + 50, hand_level[0] - 50:hand_level[0] + 50]
    body_color = np.median(hand_neighborhood, axis=(0, 1)).astype(np.uint8)

    cv2.fillPoly(model_image, [np.array(body_polygon.values(), dtype=np.int32)], body_color)

    return model_image
```

Figure 6: Function for clear the model image inside the polygon

Then for the clothes images after converting colors and thresholding with cvtColor() to gray scale image should find Contour around the clothing and get the maximum contour and draw the boundary box around the border and crop the clothing image.

```
import matplotlib.pyplot as plt
import numpy as np

cover = cv2.imread(r"images/lady2.png")
#cover = cv2.cvtColor(cover,cv2.COLOR_BGR2RGB)

h = cover.shape[0]
w = cover.shape[1]

#-----------------------------------------------------------------------------------------
#GET THE CROPED SHIRT IMAGE USING CANNY EDGE DETECTION AND CONTOURS DRAWING
shirt = cv2.imread(r"images/shirt2.jpeg")
shirt = cv2.cvtColor(shirt,cv2.COLOR_BGR2RGB)
# Convert the image to grayscale
gray = cv2.cvtColor(shirt, cv2.COLOR_RGB2GRAY)
#-----------------------------------------------------------------------------------------

# Convert the image to grayscale
gray = cv2.cvtColor(shirt, cv2.COLOR_BGR2GRAY)

# Apply binary thresholding to extract the shirt region
_, threshold = cv2.threshold(gray, 1, 255, cv2.THRESH_BINARY)

# Find contours of the shirt region
contours, _ = cv2.findContours(threshold, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

# Find the contour with the largest area
contour = max(contours, key=cv2.contourArea)

# Create a mask with the same shape as the image
mask = np.zeros_like(shirt)
```

Figure 7: Function for create the cropped clothing images

then after that, the clothing images should be resized to the size of the model's upper body or lower body For that we should draw the boundary box around the marked body levels and the clothes should be resized according its gender and clothing choice. If the gender is male and the clothing choice is shirt and trousers then the height of the upper body cloth will be set to the difference of shoulder level and hip level. Then for the width of the clothing item, the difference between the left shoulder level and right shoulder level should be taken into account and then the addWeighted() function which we have to use to get the alpha blending can be used to blend the cleared model images and the cropped clothing image.

```python
def overlay_clothing(model_image, clothing_images, shoulder_level, hip_level, ankle_level):
    if len(clothing_images) != 2:
        raise ValueError("Two clothing images are required.")

    upper_clothing_height = hip_level[1] - shoulder_level[1]
    lower_clothing_height = ankle_level[1] - hip_level[1]


    upper_clothing_aspect_ratio = clothing_images[0].shape[1] / clothing_images[0].shape[0]
    upper_clothing_width = int(upper_clothing_height * upper_clothing_aspect_ratio)
    resized_upper_clothing = cv2.resize(clothing_images[0], (upper_clothing_width, upper_clothing_height))

    if upper_clothing_width <= 0:
        raise ValueError("Invalid upper clothing dimensions.")

    # Resize the lower clothing image to match the width between hip and ankle
    hip_to_ankle_width = ankle_level[0] - hip_level[0]
    lower_clothing_width = int(clothing_images[1].shape[1] * (hip_to_ankle_width / clothing_images[1].shape[0]))
    if lower_clothing_width <= 0 or lower_clothing_height <= 0:
        raise ValueError("Invalid lower clothing dimensions.")

    resized_lower_clothing = cv2.resize(clothing_images[1], (lower_clothing_width, lower_clothing_height))


    upper_clothing_mask = np.zeros_like(resized_upper_clothing)
    lower_clothing_mask = np.zeros_like(resized_lower_clothing)

    upper_pts = np.array([shoulder_level, hip_level, (hip_level[0], shoulder_level[1])], np.int32)
    lower_pts = np.array([hip_level, ankle_level, (ankle_level[0], hip_level[1])], np.int32)


    cv2.fillPoly(upper_clothing_mask, [upper_pts], (255, 255, 255, 255))  # RGBA format with full opacity (255)
    cv2.fillPoly(lower_clothing_mask, [lower_pts], (255, 255, 255, 255))

    # Invert the masks to keep the clothing areas transparent
    upper_clothing_mask = cv2.bitwise_not(upper_clothing_mask)
    lower_clothing_mask = cv2.bitwise_not(lower_clothing_mask)

    # Apply the masks to the clothing images
    resized_upper_clothing = cv2.bitwise_and(resized_upper_clothing, upper_clothing_mask)
    resized_lower_clothing = cv2.bitwise_and(resized_lower_clothing, lower_clothing_mask)

    model_image = model_image & (255 - upper_clothing_mask)
    model_image = model_image & (255 - lower_clothing_mask)

    # Overlay the upper and lower clothing on the model image
    model_image[shoulder_level[1]:hip_level[1], :] = cv2.addWeighted(
        model_image[shoulder_level[1]:hip_level[1], :], 1, resized_upper_clothing, 1, 0
    )

    model_image[hip_level[1]:ankle_level[1], :] = cv2.addWeighted(
        model_image[hip_level[1]:ankle_level[1], :], 1, resized_lower_clothing, 1, 0
    )

    return model_image
```

Figure 8: Function for resizing clothes items and overlay on models image

Then generate the output as the body level input and with the overlay function to show output to the user and the user can download the image or reset the software back to the basic.

```
def generate_output():
    global model_image, clothing_images, output_image, body_polygon

    if model_image is None or clothing_images[0] is None or clothing_images[1] is None:
        messagebox.showerror("Error", "Please select a model image and two clothing images.")
        return

    if not body_polygon:
        messagebox.showerror("Error", "Please label body levels by drawing a polygon.")
        return

    gender_choice = gender_var.get()
    clothing_choice = clothing_type_var.get()

    shoulder_level = body_polygon.get("shoulder")
    hip_level = body_polygon.get("hip")
    knee_level = body_polygon.get("knee")
    ankle_level = body_polygon.get("ankle")

    if gender_choice == "Female":
        if clothing_choice == "Blouse & Trouser":
            output_image = oncloth.overlay_clothing(model_image.copy(), clothing_images, shoulder_level, hip_level, knee_level, ankle_level)
        elif clothing_choice == "Blouse & Skirt":
            blouse_height = knee_level[1] - shoulder_level[1]
            blouse_width = model_image.shape[1]
            skirt_height = ankle_level[1] - knee_level[1]
            skirt_width = model_image.shape[1]

            resized_blouse = cv2.resize(clothing_images[0], (blouse_width, blouse_height))
            resized_skirt = cv2.resize(clothing_images[1], (skirt_width, skirt_height))

            output_image = model_image.copy()
            output_image[shoulder_level[1]:knee_level[1], :] = resized_blouse
            output_image[knee_level[1]:ankle_level[1], :] = resized_skirt
        elif clothing_choice == "Frock":
            frock_height = ankle_level[1] - shoulder_level[1]
            frock_width = model_image.shape[1]

            resized_frock = cv2.resize(clothing_images[0], (frock_width, frock_height))

            output_image = model_image.copy()
            output_image[shoulder_level[1]:ankle_level[1], :] = resized_frock
        else:
            messagebox.showerror("Error", "Invalid clothing choice for Female.")
    elif gender_choice == "Male":
        if clothing_choice == "Shirt & Trouser":
            output_image = oncloth.overlay_clothing(model_image.copy(), clothing_images, shoulder_level, hip_level, ankle_level)
        elif clothing_choice == "Shirt & Short":
            shirt_height = hip_level[1] - shoulder_level[1]
            shirt_width = model_image.shape[1]
            short_height = ankle_level[1] - knee_level[1]
            short_width = model_image.shape[1]

            resized_shirt = cv2.resize(clothing_images[0], (shirt_width, shirt_height))
            resized_short = cv2.resize(clothing_images[1], (short_width, short_height))

            output_image = model_image.copy()
            output_image[shoulder_level[1]:hip_level[1], :] = resized_shirt
            output_image[knee_level[1]:ankle_level[1], :] = resized_short
        else:
            messagebox.showerror("Error", "Invalid clothing choice for Male.")
    else:
        messagebox.showerror("Error", "Invalid gender choice.")

    if output_image is not None:
        display_image(output_image, output_image_label)
```

Figure 9: Function for generating output

CHAPTER 04

RESULTS AND DISCUSSION

4.1. MAJOR RESULTING STEPS

As for the major resulting steps the system can take correctly all the 10 body levels for the resizing calculating and if the user selected the body points correctly the polygon will be drawn correctly.

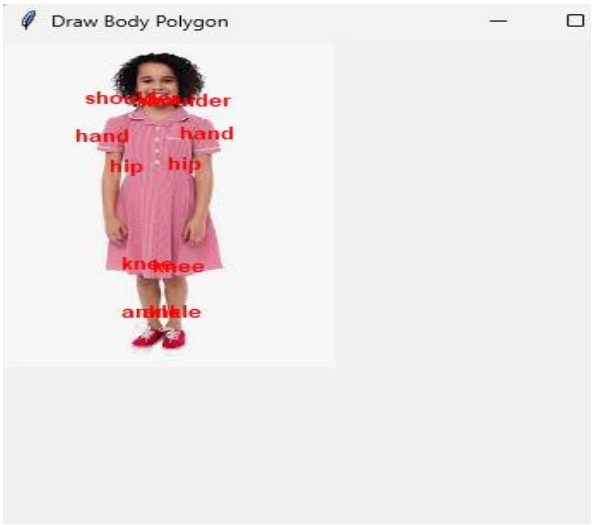

Figure 10: Canvas to get the mouse click events from user

Below Figure (see Figure 11) this is when the polygon is drawn around the person's body and the area will be cropped according to the mouse click points and fill the area with the given color.

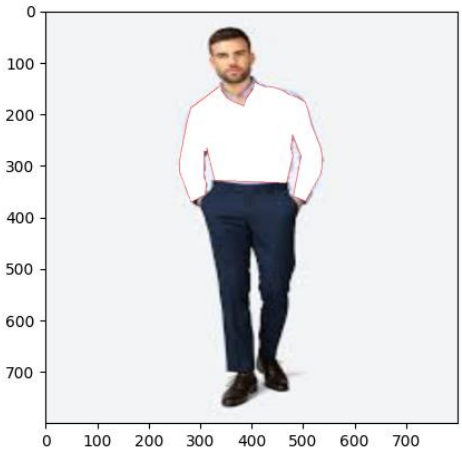

Figure 11: Clearing the models image after drawing polygon

4.2. SOME TESTED IMAGE RESULTS

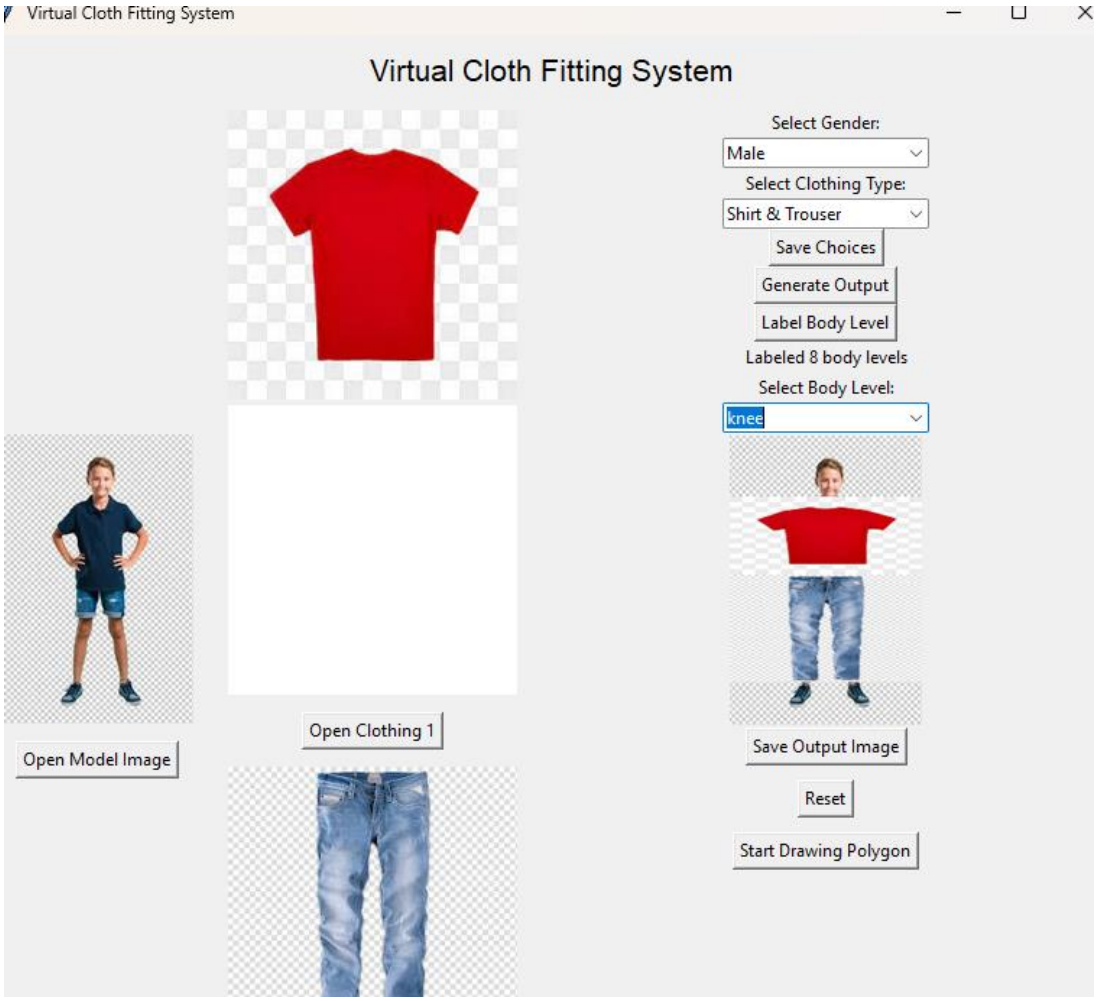

Figure 12: tested with the kid models image with t-shirt and trouser

For the tested results (see **Figure 12**) is the interface where the system has checked with the boy kid model image and added 2 clothing items a shirt and a trousers and then drew the polygon by giving the body levels on the model's image and the output will be shown as above. You can download the output image by saving the output button.

Then you can reset the interface.

This can be tried out for other clothing types and other genders too (see **Figure 13**).

Then the appropriate input values should be included or else it will raise an error message.

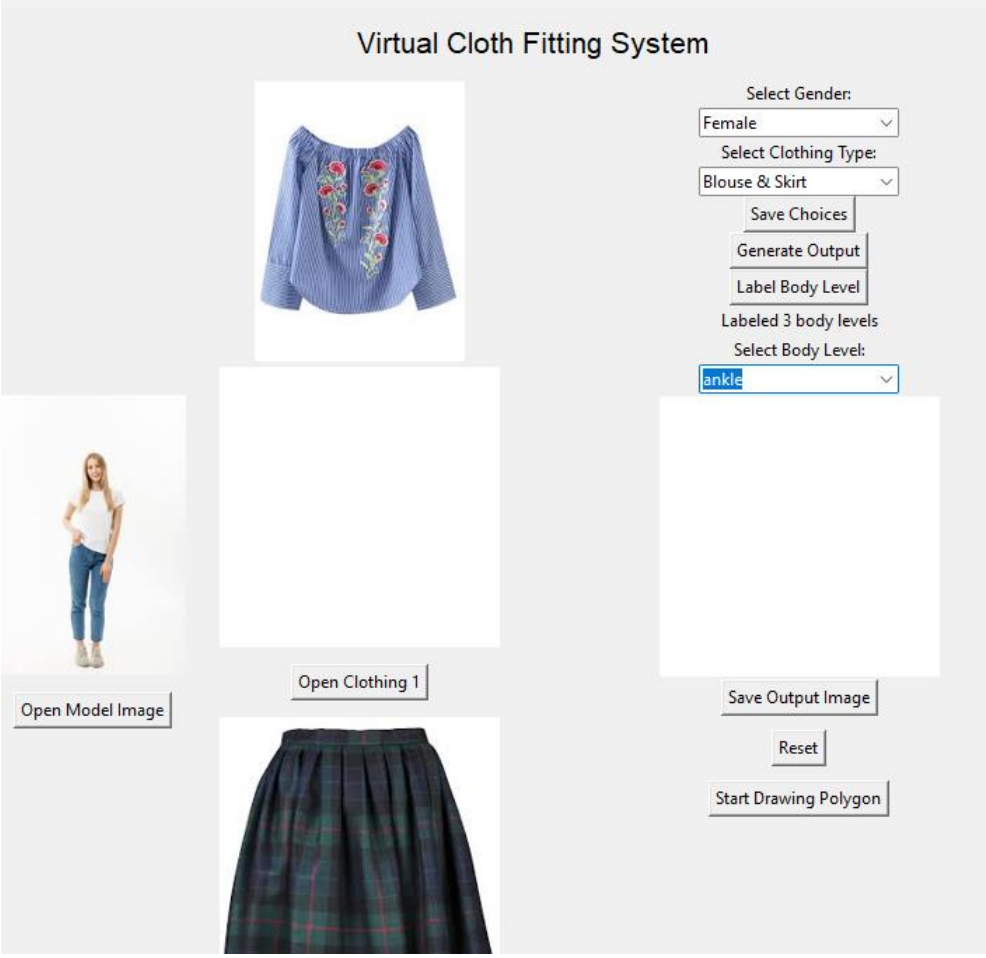

Figure 13: tested with the Female models image with blouse and skirt

4.3. DISCUSSION

The introduction of the Virtual Clothes Fitting System, underpinned by advanced image processing, signifies a noteworthy evolution in the domain of online fashion retail. This system adeptly addresses the longstanding quandary faced by online shoppers: the uncertainty surrounding clothing fit and appearance. By seamlessly integrating cutting-edge computer vision and machine learning technologies, it offers a compelling virtual fitting experience, incorporating precise body recognition, posture estimation, and the realistic overlay of virtual clothing. This innovation is poised to empower consumers with newfound confidence in their online attire selections, enhancing user satisfaction and potentially reducing return rates. Despite inherent challenges, the Virtual Clothes Fitting System holds substantial potential to reshape the landscape of online fashion retail.

CHAPTER 05

CONCLUSIONS

5.1. ADVANTAGES OF THE IMPLEMENTED SYSTEM

The Virtual Clothes Fitting System, implemented through image processing technologies, offers a multitude of advantages for both consumers and retailers alike. One of the most notable benefits is the substantial increase in customer confidence. By providing users with a highly realistic virtual try-on experience, the system alleviates the longstanding problem of uncertainty when making online clothing purchases. Users can now see how clothing items will fit and look on their own bodies, leading to a greater sense of assurance in their buying decisions.

Moreover, the Virtual Clothes Fitting System presents substantial cost-saving opportunities for retailers. By optimizing inventory management and curbing return-related expenses, retailers can improve their overall profitability and competitiveness in the online market. Furthermore, the system's contribution to environmental sustainability is noteworthy. With fewer returns and reduced reliance on physical try-ons, the fashion industry can mitigate its environmental impact, particularly in terms of transportation and packaging waste.

5.2. ISSUES

The complexity of clothing also poses challenges. Certain clothing items, particularly those with intricate patterns, drapes, or textures, may be more challenging to simulate accurately. The system's effectiveness in accommodating various clothing styles and materials.
In addition to that this may not give a very accurate answer since this resizing is not classified into different clothes sizes or material type it may not be satisfied with the customers satisfaction in the output.

5.3. CONCLUSION

In conclusion, the Virtual Clothes Fitting System, driven by image processing technologies, represents a significant advancement in the realm of online fashion retail. Its advantages, including boosting customer confidence, reducing returns, enhancing personalization, and delivering cost savings for retailers, are substantial and compelling.

Continuous research and development in image processing, machine learning, and user experience design are essential to further enhance the system's accuracy, security, and usability. As we continue to refine these technologies, we move closer to achieving the full potential of virtual clothing fitting systems in revolutionizing the online fashion retail landscape.

REFERENCES

(1) Gonzalez, R. C., & Woods, R. E. (2020). Digital Image Processing. 3rd edition. Pearson.

(2) OpenCV Tutorial, http://www.cs.iit.edu/~agam/cs512/lectnotes/opencv-intro/opencv-intro.html

(3) Garcia, C., Bessou, N., Chadoeuf, A., & Oruklu, E. (2012). Image Processing Design Flow for Virtual Fitting Room Applications Used in Mobile Devices. Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, Illinois, USA.

(4) Srinivasan, K., & Vivek, S. (2017). Implementation of virtual fitting room using image processing in IEEE International Conference on Computer, Communication and Signal Processing (ICCCSP-2017)

(5) 3DLook. (Website). Retrieved from https://3dlook.ai/