

Complex Augmented Reality Driven Online Shopping System

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

RATHANA NARASIMHUDU (Reg.No – 40111043)

RAYALA PRABHAS (Reg.No - 40111048)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

Accredited with Grade "A++" by NAAC, | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119

APRIL- 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
[DEEMED TO BE UNIVERSITY]

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of RATHANA NARASIMHUDU (40111043) and RAYALA PRABHAS (40111048) who carried out the Project entitled "COMPLEX AGUMENTED REALITY DRIVEN ONLINE SYSTEM " under my supervision from November 2023 to April 2024.

Internal Guide

Dr.J.REFONAA, M.E., Ph.D

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.,

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
[DEEMED TO BE UNIVERSITY]
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai – 600 119.

Submitted for Viva voice Examination held on

13/04/2024

13/04/24
Internal Examiner

13/04/24
External Examiner

DECLARATION

I, RATHANA NARASIMHUDU (40111043), hereby declare that the Project Report entitled "COMPLEX AGUMENTED REALITY DRIVEN ONLINE SYSTEM " done by me under the guidance of Dr.J.REFONAA, M.E., Ph.D is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 13/4/2024

PLACE: Chennai

R. Narasimhudu

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute Of Science And Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr.L.Lakshmanan, M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.J.REFONAA,M.E.,Ph.D**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Augmented reality (AR) is useful and efficient in allowing customers to examine a computer-generated view of items with a tap of a finger from their cell phones whenever it is convenient for them. By augmented reality technology, buyers may visually inspect their items in the real world, saving time spent travelling to see them in person. In this study, we created an e-commerce website that uses AR technology, allowing users to virtually sample things from the comfort of their own homes. In order to find bodies, we employed a media pipe frame. The Media Pipe Holistic channel incorporates distinct models for the hand, face, and concealment. The contribution to one element is vary depending on their areas of expertise. A lower, fixed resolution videotape frame (256x256) is used as input for the disguise estimate model as an example. The picture quality would be too low for proper articulation, for example, if the hand and face areas were cropped out of the image and sent to their respective models. In order to serve the various locations appropriately, we created Media Pipe Holistic as a multi -stage channel. AR involves transparent devices that overlay visuals and text on the user's perspective of their surroundings. These devices are often worn on top. AR enriches the actual environment by adding computer-generated objects that spontaneously emerge there. The development of an AR system combines the brain's responses to 2D interfaces by working with the visual, auditory, and kinaesthetic systems.

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	4
	2.1 Inferences from Literature Survey	
	2.2 Existing System	7
	2.2.1 Disadvantages	8
3	REQUIREMENTS ANALYSIS	9
	3.1 Scope of the Project	9
	3.2 Objective	9
	3.3 Problem Definition	9
	3.4 Hardware Requirements	10
	3.5 Software Requirements	11
	3.6 System Use Case Diagram	12
4	DESCRIPTION OF PROPOSED SYSTEM	14
	4.1 Proposed System	14
	4.2 Selected Methodology or process model	15
	4.3 System Architecture/Outline Proposed System	19
	4.4 Description of Software for Implementation and Testing plan for the Proposed Model/System	21
	4.5 Implementation Plan	28
	4.6 Testing Plan	30
	4.7 Project Management Plan	33
	4.8 Transition/Software to Operations Plan	34
5	IMPLEMENTATION DETAILS	36
	5.1 Data Flow Diagram	36
	5.2 Class Diagram	38
	5.3 Activity Diagram	39
	5.4 Sequence Diagram	40

	5.5	Algorithms	41
	5.7	Frameworks	58
6	RESULTS AND DISCUSSION		60
7	CONCLUSION		62
	7.1	Conclusion	62
	7.2	Future work	63
	REFERENCES		64
	APPENDIX		66
	A. SOURCE CODE		66
	B. SCREENSHOTS		80
	C. RESEARCH PAPER		88

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
1	System Use Case Diagram	13
2	System Architecture	19
3	Data Flow Diagram	37
4	Class Diagram	38
5	Activity Diagram	39
6	Sequence Diagram	40
7	Traditional Programming	42
8	Machine Learning	42
9	Learning Phase	43
10	Inference Model	44
11	Machine Learning Algorithm	45
12	Classified Machine Learning Algorithm	49

LIST OF ABBREVIATIONS

SI.NO	FULL FORM	SHOT FORM
1	Application Programming Interface	API
2	Augmented Reality	AR
3	Content Management System	CMS
4	File Transfer Protocol	FTP
5	Artificial Intelligence	AI
6	Uniform Resource Locater	URL
7	Short Message Service	SMS
8	Product Information Management	PIM
9	Out Of Stock	OOS
10	Natural Language Processing	NLP

CHAPTER - 1

INTRODUCTION

A Next-Gen technology called "Augmented Reality based online shopping platform" will soon be deployed in a variety of e-Commerce websites and real clothing stores. In order to provide individuals with a tool that will allow them to view a real-time simulation of several outfits without having to physically wear them, we are employing the concept of augmentation. A web browser and an internet connection are all that are need to access this Web-based application from anywhere in the globe. It can operate on any type of OS and machine, such as mobile devices, desktop computers, etc., because it is a platform-independent program.

The phrase "augmented reality" is becoming more and more well-known in the marketing and retail industries. Fascinatingly, fashion is a fantastic fit for augmented reality since it is experience-based. Future trends will have a greater impact on the virtual world than is currently apparent. By the end of 2020, augmented reality and virtual reality products will bring in 120 billion, according to Digi Capital. However, Greg Jones, Google's head of AR and VR, believes that augmented reality will play a transformation role in society. He also made an additional comment that the situation would be the same if it applied to the retail fashion business.

It is a well-known reality that businesses have a lot of issues when it comes to customer involvement and awareness. These problems can be resolved holistically with the use of virtual reality and augmented reality. In the world of fashion, augmented reality, or AR, is advancing to provide a crucial sensory link to support both in-person and online buying. Virtual reality, or VR, has an immersive science-fiction feel to it, which is why several merchants have employed it in the past. For instance, Lowe's hologram enables buyers to put on a headset and experience kitchen and bathroom remodeling ideas. However, it is crucial to highlight that this type of cutting-edge technology is only accessible in a few countries.

In addition, it is currently just being used to promote a brand rather than to increase in-store sales. However, although being less visually appealing than VR, augmented reality (AR) is adding a lot of usefulness to how online customers learn about new goods and services. AR makes use of a camera to analyse a visual trigger in the user's surroundings. On their mobile devices, which might take the shape of smart phones or tablets, users can view a genuine video of the actual world. It is noteworthy in this context that, in the opinions of the millions of Pokemon Go players, this phenomenon often has a fascinating element.

Pokemon Go is a great example of augmented reality for folks who wish to interact with characters by entering a television. The game Pokemon Go may be used to determine how useful augmented reality will be in fashion and retail. The game's participants may use their mobile devices to monitor their surroundings in real time. This is true because the augmented reality software may effortlessly combine the actual world and the animated world. The fashion sector is also affected by this. In order to provide customers, the opportunity to try out numerous lipstick hues using their front camera, Sephora has released a smart phone app called Virtual Artist.

In this perspective, it is intriguing to see that customers who appreciate their own reflection choose to "add to basket" when they see one. The fact that Sephora's Virtual Artist has attempted a staggering 45 million times is another intriguing point to consider. The premium firms are also paying attention by keeping in mind figures like these. For instance, Burberry's physical Beauty Box Outlets are looking into how to get more customers involved in augmented reality experiences for cosmetics. The four marketing stages can be guided by AR for the consumer. These four phases take the shape of raising awareness, fostering enduring loyalty, converting purchasing intent into action, and increasing contemplation.

AR would assist the shops in moving through these phases holistically. On the other hand, augmented reality (AR) can enable access to a fashion store's in-store experiences, which is advantageous for the development of a specific fashion brand. The fashion firm may ensure that the clients are readily persuaded by offering authentic experiences to them, which is a terrific strategy to increase sales. AR was

used to plan a fashion display during the Paris Fashion Show. Using a smart phone software that can disclose concealed pictures, it allowed the audience to scan models. It is intriguing how AR might help the fashion industry address one of its biggest challenges: how to show customers how the item will appear on them. When your company is an online fashion e-commerce site, this issue becomes more significant. For instance, Zara intends to include AR into their chains in order to draw clients.

Quite phenomenally, the retailer would bring this technology to more than 130 locations across the world. With the help of AR, Zara would be able to depict models sporting the latest apparels to its customers when they place their smart phones near the sensor located in the store. On the other hand, Gap has its Dressing Room application that usually allows a customer to select the most suitable body type that resembles them which is very important for active wear. Subsequently, they can see the picture of the apparel of their choice in that body type. This would give the online shoppers a fair idea on how the clothing would look like on them.

CHAPTER - 2

LITREATURE SURVEY

2.1 Inferences from Literature Survey

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration is taken into account for developing the proposed system.

The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

1) Intelligent human body tracking, modelling and activity analysis of video surveillance system: A Survey

AUTHORS: K.Srinivasan, K.Porkumaran, G.Sai Narayanan

Recent technology and market trends have demanded the need for solutions to video/camera systems and analytics. This paper highlights the video system architectures, tasks, and related analytic methods. It demonstrates that the importance of the role that intelligent video systems and analytics play can be found

in a variety of application such as homeland security, crime prevention, traffic control, accident prediction and detection, and monitoring patients, elderly and children at home, etc. Research directions are outlined in the end with a focus on what is essential to achieve the goals of intelligent video systems and analytics.

2) Segmentation by Fusion of Histogram based K-Means Clusters in different color space

AUTHORS: Max Mignotte

This paper presents a new, simple, and efficient segmentation approach, based on a fusion procedure which aims at combining several segmentation maps associated to simpler partition models in order to finally get a more reliable and accurate segmentation result. The different label fields to be fused in our application are given by the same and simple (K-means based) clustering technique on an input image expressed in different color spaces. Our fusion strategy aims at combining these segmentation maps with a final clustering procedure using as input features, the local histogram of the class labels, previously estimated and associated to each site and for all these initial partitions. This fusion framework remains simple to implement, fast, general enough to be applied to various computer vision applications (e.g., motion detection and segmentation), and has been successfully applied on the Berkeley image database. The experiments herein reported in this paper illustrate the potential of this approach compared to the state-of-the-art segmentation methods recently proposed in the literature.

3) From 2D Photos of Yourself to Virtual Try-on Dress on the Web

AUTHORS: F. Cordier, W. Lee, H. Seo, and N. Magnenat-Thalmann

This paper describes a complete methodology for cloning and dressing people by using a website. The input is simple photographs or body measurements that anybody can produce in any environment. Then the Web-based virtual-try-on allows users to see them dressed. The basic methodology uses a pre-calculated generic

database to produce personally sized bodies and animate garments on a Web application.

4) A body and garment creation method for an internet based virtual fitting room.

AUTHORS: D. Protopsaltou, C. Luible, M. Arevalo-Poizat, and N. Magnenat Thalmann

In this paper we present a new methodology for producing 3D clothes, with realistic behavior, providing users with an apposite feel for the garment's details. The 3D garments that are produced, using our methodology, originate from 2D CAD patterns of real garments and correspond to the regular sizes that can be found on a shop floor.

Our aim is to build a compelling, interactive and highly realistic virtual shop, where visitors can choose between many different types of garments designs and proceed to simulate these garments on virtually animated bodies. By merging the approach often used by the fashion industry, in designing clothes, and our own methodology for creating dressed virtual humans, we present a new technique providing trouble-free and straightforward garment visualization.

The entire process starts from the creation of virtual bodies (either male or female), using standard measurements, which form the basis for garment modeling. Using splines, the 2D garment patterns are created and then seemed together around a virtual human body, providing the initial shape. A simulation is made using the seemed garment by applying physical parameters based on real fabric properties. Once the garment has been created, a real time platform, which has been embedded inside a web browser, is used as an interface to the Internet. We extend the web interface to add interactivity and the ability to dynamically change textures, clothes, body measurements, animation sequences and certain features of the virtual environment.

The whole methodology does not aim to build only a virtual dressing room, where customers can view garments fitted onto their own virtual bodies but to visualize made-to-measure clothes, animate them, visualize the cloth behavior and to add interactivity. The entire Virtual Try-On experience is a process of targeting the clientele, designing the clothing collection, dressing the virtual models and then using the web as a virtual shop floor.

5) Made-to-measure technologies for an online clothing store

AUTHORS: F. Cordier, H. Seo, and N. Magnenat-Thalmann

The Internet is a compelling channel for selling garments. Several recent initiatives by companies such as Nordstrom, Macy's, and Lands End focus on made-to-measure manufacturing and shopping via the Internet. We present a Web application that provides more powerful access to and manipulation of clothing to facilitate clothing design, pattern derivation, and sizing. We apply 3D graphics technology to help create and simulate the virtual store

2.1 EXISTING SYSTEM

In the existing there are various applications similar to the Virtual Trial Room, some of them are being referred. The Fitter Room (Published by IEEE) focuses mainly on the Kinetic software used the Kinetic sensors are used to find the body functions and 3d modeling techniques are used for superimposing an image. Blender software is mainly used for 3d modeling and also hardware components i.e., sensors are used whereas we will not be using any hardware components to keep it user- friendly and easy to install

The Machine Learning approach (Published by IEEE) contains machine learning algorithms to detect facial and eye movements to gain coordinates for the user. It has an approach for more than 60,000 algorithms to gain accuracy. Image Processing Techniques (Published by IEEE) contains techniques of image processing using

filters such as Gaussian filters, mean filters, Gaussian smoothing, Conservative Smoothing, unsharp filter, Laplacian/Laplacian of gaussian filter, high pass and low pass filter, edge enhancement filter, edge detection filter, etc. Virtual Trial Room using augmented reality (Published by International journal of advanced computer technology IJACT): The Kinetic Sensor are paired with a depth sensor as well as an RGB sensor. When the user is allowed to stand in front of the webcam, the Kinetic will track the body joints and will show the skeleton of the user's body.

2.1.1 Disadvantages

- The main disadvantage of the existing system is it will not work without the use of hardware components. Some systems, such as the ones that use augmented reality technology, may require additional hardware such as sensors to function properly. This can make the system less accessible or user-friendly for some customers.
- The existing system has lower accuracy when implemented in the real time application.
- The existing system may depend on user-uploaded images, which may not always be of high quality or accurate, leading to inaccurate virtual try-ons.
- The existing systems may have been costly to implement, which could limit their adoption by smaller retailers or those with limited budgets.
- Even with advanced technology, virtual trial rooms may not be widely adopted by customers due to a lack of awareness, trust, or interest in using the technology which is very hard to understand.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 SCOPE OF THE PROJECT

The project's goal is to provide everyone in the globe the wonderful ease of trying on clothing without donning it.

3.2 OBJECTIVE

- I. The objective of the project is to provide the world with a great convenience of trying clothes without even wearing them physically.
- II. We are creating an Augmented Reality based application, which can be used to visualize the clothes on your body virtually using web application.
- III. This will give user the freedom of trying clothes and will make them able to try as many clothes as possible in lesser amount of time.

3.3 PROBLEM DEFINITION

By using augmented reality technology, an online shopping system can provide customers with a more realistic and engaging experience. This system would allow customers to view products in 3D and visualize them in their real-world environments. They would be able to rotate, zoom in and out, and interact with products to see how they function and how they look from different angles.

The augmented reality based online shopping system also addresses the issue of returns and exchanges. Customers can be unsure about the size or appearance of a product when shopping online, which can lead to a higher rate of returns and exchanges. With augmented reality, customers can get a better idea of how products

will fit and look in their homes, reducing the need for returns.

3.4 HARDWARE REQUIREMENT

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and it also shows how it should be implemented. It specifies the speed of the system that should be used in this project. The hardware requirement for this project is mentioned below.

System	: Pentium i3 Processor
Hard Disk	: 500 GB.
Monitor	: 15" LED
Input Devices	: Keyboard, Mouse
Ram	: 2 GB

The Pentium i3 Processor is a CPU manufactured by Intel, which is designed to provide good performance for basic tasks such as web browsing, office work, and multimedia consumption. The processor has two cores and supports Hyper-Threading, which means that it can execute four threads simultaneously. This allows the system to handle multiple tasks at once without slowing down.

The 500 GB hard disk provides a considerable amount of storage space, which is sufficient for storing a large number of files and documents, as well as some multimedia content. However, if you plan to store large video files or multimedia libraries, you may want to consider additional storage options, such as an external hard drive or cloud storage.

The 15-inch LED monitor is a standard size for laptops and desktops, and it uses LED technology to produce bright and clear images. LED monitors are generally more energy-efficient and have a longer lifespan than traditional LCD monitors.

The keyboard and mouse are common input devices that allow for easy and efficient navigation of the computer. The keyboard features a standard layout with alphanumeric keys, as well as function keys and special keys for specific tasks. The mouse provides a pointing device that allows you to click and select items on the screen, and it may also feature additional buttons for customization.

The 2GB RAM is the amount of memory that the computer has available for running programs and applications. While this may be sufficient for basic tasks, such as web browsing and office work, it may limit the number of programs and applications that can run simultaneously. If you plan to run multiple programs at once or use memory-intensive applications, you may want to consider upgrading the RAM to increase performance.

3.5 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team and tracking the team's progress throughout the development activity. The below are the software requirements for this project:

Operating system	: Windows 10
Coding Language	: python

Windows 10 is a widely-used operating system developed by Microsoft. It has a user-friendly interface and provides various features such as Cortana, a virtual assistant, and a built-in web browser, Microsoft Edge. It also has a robust security system to protect against malware and other threats.

Python is a popular high-level programming language used for a wide range of

applications, including web development, data analysis, artificial intelligence, and scientific computing. It is known for its simple syntax, readability, and ease of use, making it a great language for beginners to learn. It has a vast library of modules and packages that can be used for various tasks, making it a versatile language. Python is compatible with Windows 10, and many developers use it to build applications on this operating system

3.6 System Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

In this use case diagram, the primary actors are the customer and the e-commerce website. The customer interacts with the website through a series of use cases, including browsing products, searching products, viewing product details, adding products to their cart, checking out their cart, and completing the payment process.

The final use case in this diagram is augmented reality, which allows customers to use their device's camera and AR technology to see products in their real-world environment. This feature enhances the shopping experience by allowing customers to see how products will look and fit in their homes before making a purchase.

Overall, this use case diagram represents the key interactions between the customer and the e-commerce website, highlighting the use of augmented reality as a key feature that enhances the shopping experience.

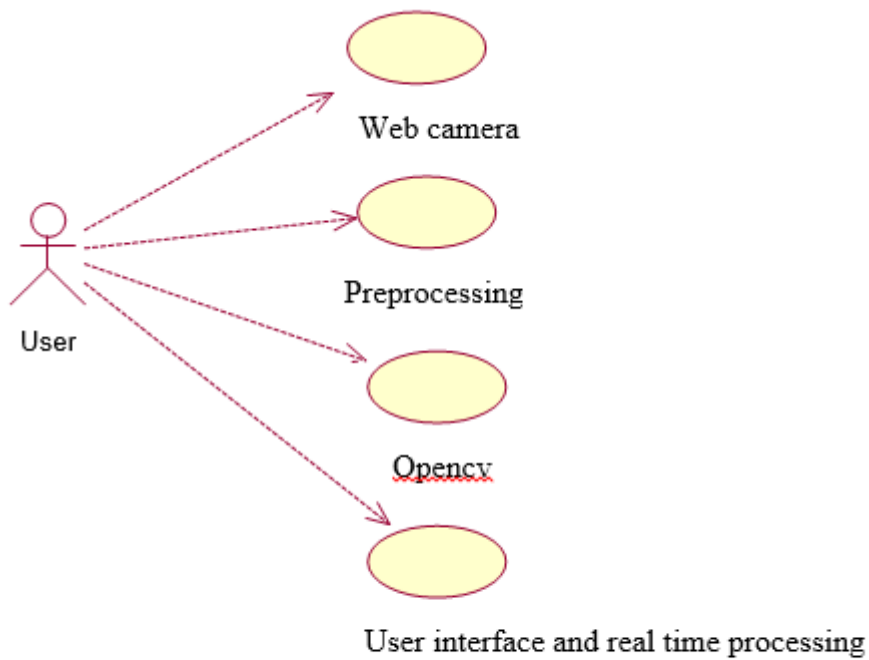


Fig 1. System Use Case Diagram

The use cases shown in the diagram are:

- I. View Augmented Content: The User can view augmented content on their device by pointing their camera at a specific object or location.
- II. Track Object: The User can track an object through the camera of their device, allowing the application to display relevant information related to the object.
- III. Display Information: The System can display relevant information related to the augmented content, such as product details or historical information.
- IV. Access Data: The System can access data, such as user preferences or location information, to deliver personalized and relevant content to the User.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 PROPOSED SYSTEM

The goal of the virtual trial room system is to provide users with an intuitive web application. The system is created with Flask, HTML, CSS, and JavaScript using various features and buttons to make it appealing and eye-catching for users. The interface is created in local languages, specifically English, and can be readily understood by any user. The "Tryon" button must be selected by the user if they want to try on clothing.

The user's body is overlaid with an image of the clothing using OpenCV input from the camera. If the user likes the clothing, they can either buy it or try it on right now. Our proposed system is focused on developing web application that records a video of the user using the device's camera, splits the video into individual frames, and then extracts the user's body from those frames. Using functions to morph, rotate, and scale the wearable image onto the user in real-time, as well as functions to extract information on the locations of joints in the body.

The primary goal of this project is to create a working prototype for an online virtual trial area where buyers may virtually try on clothing. This effort will be highly helpful both during and after the COVID 19 epidemic. For the customers, it will make things simpler and more effective.

4.1.1 *Advantages*

- The proposed system implementation of the Virtual Trial Room system helps the user to explore billions of options sitting right at the home rather than physically visiting the shops.
- By using this application users can have a look at how an attire fits and looks on them.
- This application is useful in this pandemic situation as well where you don't

have an option to try on clothes while buying them. During COVID-19 pandemic, offline stores will limit trials. During those times this technology will be very useful.

- Also, it saves the retailer from damaged garments due to the try-on facility given to the customers in their shops.
- We will build a system that has consistently higher precision and individual members are more satisfied.
- If we compare this project to other existing systems, one of the main differences is the absence of any hardware components in our project.

4.2 Selected Methodology or process model

4.2.1 Detecting and Sizing the Body

First step of the proposed Online Virtual Trial Room method is the acquisition of the shape of the body, head or neck depending upon the wearable to get reference points. Reference points are then used to determine where to display the particular cloth or ornament. In order to obtain the body shape, we applied several techniques: i) Filtering with threshold, Canny edge detection, K-means, and ii) Motion detection or skeleton detection wherein multiple frames were analyzed for any movement.

However, the results were unreliable and not good enough to obtain reference points for displaying the wearable. Therefore, we introduced a new detection methodology based on locating the face of the user, adjusting a reference point at his/her neck and displaying the wearable based on that point. In addition, another point of reference can be obtained by using an Augmented Reality (AR) marker. Though this was enough for small attires like glasses or ornaments but it was not enough to map the clothes onto the user body. For obtaining the size of the user, we follow a similar automated body feature extraction. The idea is to set up the user in front of the camera and hold him at the beginning at a certain predetermined distance.

The algorithm extracts points on the shoulders and the belly. Measuring the distance between these points and knowing the distance from the user to the camera, the size

of the user can be obtained. When the image (video frame) is acquired, a canny edge detection filter is applied to obtain only the silhouette of the body. Canny edge detection is really susceptible to noise that is present in unprocessed data; therefore, it uses a filter where the raw image is convolved with a Gaussian filter. After convolution, four filters are applied to detect horizontal, vertical and diagonal edges in the processed image. Morphological functions are also applied to obtain a closed silhouette. Finally, an 8-points Freeman chain code, is applied to assign a direction to each pixel. OpenCV's built-in Haar cascades were used to detect the user's body and isolate it from the background

4.2.2 Face Detection

As when the user comes in front of the screen, to detect the user the discrete structure which is to be identified is the face. So, to detect the face, we use Haar feature-based cascade classifiers. In haar classifier instead of using intensity values of the pixel, it uses the change in contrast values between adjacent groups of pixels. Then the variance difference between the pixel groups is used to determine the relative light and dark areas in the image.

It is a machine learning approach. So, to work well with the algorithm the cascade function is trained from a lot of negative and positive images. A lot of negative images (images without faces) and positive images (images with faces) are shown to the classifier to train it so it can extract features from it. The purpose of using OpenCV makes it easier as it comes with pre-trained classifiers for face, eyes, smile etc. It comes with a trainer and a detector; we can train it with our own classifier easily for any object detection. If it finds a match it returns Rect (x, y, w, h) implying coordinates for left, top, bottom and right.

4.2.3 Image Masking

In this simply the image masked has some of its pixel intensity values set to zero. In

the image wherever the pixel intensity value is zero automatically the pixel intensity of the resulting masked image will be set to the background value which is normally zero. Or To define the mask the ROIs for each slice is used. If required, masking can be controlled on slice by slice basis in ROI toolkit. In ROI toolkit, masking operations does not affect a slice without ROI.

4.2.4 Edge Detection

There are various edge detection techniques. We have used Canny Edge detection technique as discussed before for detection of body. To perform this edge detection technique Gaussian filters are used. These filters cut out the noise in a digital image to prevent any false detection by the processor. This does the work of smoothening and reducing the effect of noise on the image for the proper functioning of the processor. With this the intensity gradients of the image are not found out.

The edges in the image can point in various directions like horizontal, vertical and in diagonal edges, so this algorithm uses four filter to detect all kind of edges in blurred image. After this process non-maximum suppression is applied to make the edge thin. This suppression results in quite accurate edge pixels in reference to present real edges. Also, some pixels may be caused by the noise, then for such pixels we apply double threshold on them.

4.2.5 Scaling of Attire

Scaling means re-sizing of the image according to the circumstances. As when the user moves in front of the screen it should change the sizing of the attire and place it on the body accordingly. When the user moves towards the screen the image size should be increased according to user but the actual measurements of the attire should not increase. This is done by scaling method.

4.2.6 Body Parts Tracking

Open CV's pose estimation algorithms like OpenPose or AlphaPose were used to track the user's body parts like head, torso, arms. This helped us map the user's body to the virtual clothes

4.2.7 Data Preparation

We collected a dataset of images of clothes from different angles and sizes. We then preprocessed the images to remove any background noise and standardize the image size and format

4.2.8 Virtual Clothes Overlay

Computer vision techniques like homograph, image warping, and blending were used to overlay the virtual clothes on the user's body in real-time. This created the illusion that the user was wearing the clothes

4.2.9 User Interface

A simple user interface was developed using OpenCV's GUI toolkit, allowing users to select different types of clothes and accessories. This made the application easy to use and intuitive.

4.2.10 Real-time Processing

The algorithms were optimized to achieve real-time processing and rendering of the virtual clothes on the user's body. This enabled the user to see themselves in the virtual clothes in real-time.

4.3 System Architecture/Outline of Proposed System

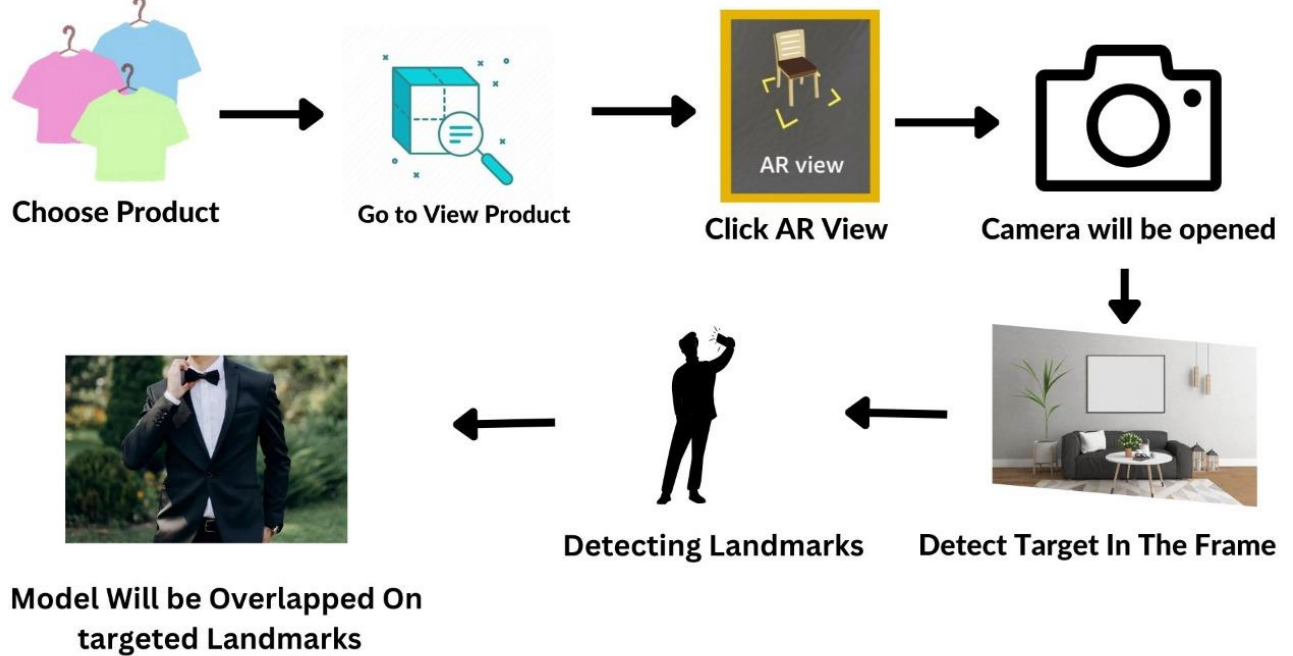


Fig 2. System Architecture

4.3.1 Outline for System Architecture

4.3.1.1 Mobile Application: The AR-based online shopping system requires a mobile application that allows users to browse and purchase products. This application should be designed to work on multiple platforms, including iOS and Android.

4.3.1.2 AR Engine: The AR engine is the heart of the AR-based online shopping system. It is responsible for rendering the 3D models of products and placing them in the real-world environment of the user. This engine requires significant computing power and must be optimized for speed and accuracy.

4.3.1.3 Product Database: The product database is a critical component of the system. It contains all the information about the products that users can browse and purchase, including product descriptions, images, 3D models, and pricing information.

4.3.1.4 Image Recognition System: The image recognition system is used to identify the real-world objects that the user is viewing through their device's camera. This system is required for the AR engine to place the 3D models of the products in the correct position and orientation in the user's environment.

4.3.1.5 Payment Gateway: The payment gateway is a critical component of the AR-based online shopping system. It allows users to securely purchase products using their preferred payment method.

4.3.1.6 Analytic and User Tracking: The analytic and user tracking component is used to collect data on user behavior and preferences. This information is used to improve the user experience and provide personalized recommendations to users.

4.3.1.7 Content Management System: The content management system is used to manage the content of the mobile application, including product information, pricing, and user reviews.

4.3.1.8 Cloud Infrastructure: The AR-based online shopping system requires a robust cloud infrastructure to handle the computational and storage requirements of the system. This infrastructure should be designed to handle high traffic volumes and be scalable to accommodate future growth.

4.4 Description of Software for Implementation and Testing plan of the Proposed Model/System

4.4.1 Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

4.4.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

4.4.3 Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.

- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let us understand how to set up our Python environment.

4.4.4 Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.
- Follow the link for the Windows installer python-XYZ. msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

4.4.5 First Python Program

Let us execute programs in different modes of programming.

4.4.5.1 Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
$ python
Python2.4.3(#1, Nov112010,13:34:43)
[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

4.4.5.2 Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

```
print "Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

```
$ python test.py
```

This produces the following result –

```
Hello, Python!
```

4.4.6 What is Python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

4.4.7 What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.
-

4.4.8 Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Net beans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

4.4.9 Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download it for free from the following website: <https://www.python.org/>

4.4.10 Python QuickStart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.

```
helloworld.py  
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

4.4.11 The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

From there you can write any python, including our hello world example from earlier in the tutorial:

```
C:\Users\Your Name>python
```

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on  
win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
Hello, World!
```

Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

```
exit()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

```
>>> print("Hello, World!")
```

Hello, World!

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

Python Indentations

Where in other programming languages the indentation in code is for readability only, in Python the indentation is very important.

Python uses indentation to indicate a block of code.

Example

```
if 5 > 2:
```

```
    print("Five is greater than two!")
```

Python will give you an error if you skip the indentation:

Example

```
if 5 > 2:
```

```
print("Five is greater than two!")
```

Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment.

4.5 Implementation Plan

- **Define Requirements:** The first step is to define the requirements for the system. This includes identifying the target audience, defining the features and functionality of the system, and determining the technical requirements.

- **Select AR Platform:** Next, select the AR platform that will be used for the system. There are several options available, including ARKit for iOS devices and ARCore for Android devices.
- **Develop AR Engine:** Develop the AR engine that will be used to render 3D models of products in the real world. This engine should be optimized for speed and accuracy and be able to handle a large number of 3D models simultaneously.
- **Build Product Database:** Build the product database that will be used to store information about the products available for purchase. This includes product descriptions, images, 3D models, and pricing information.
- **Implement Image Recognition System:** Implement the image recognition system that will be used to identify real-world objects and place 3D models of products in the correct position and orientation.
- **Develop Mobile Application:** Develop the mobile application that will be used by users to browse and purchase products. This application should be designed to work on multiple platforms, including iOS and Android, and be optimized for the AR engine.
- **Integrate Payment Gateway:** Integrate a payment gateway into the system to enable users to securely purchase products using their preferred payment method.
- **Implement Analytics and User Tracking:** Implement an analytics and user tracking system that will be used to collect data on user behavior and preferences.
- **Implement Content Management System:** Implement a content management system that will be used to manage the content of the mobile

application, including product information, pricing, and user reviews.

- **Deploy Cloud Infrastructure:** Deploy a cloud infrastructure that will be used to handle the computational and storage requirements of the system.
- **Test and Refine:** Test the system to ensure that all components work together seamlessly and refine the system as necessary to improve performance and user experience.
- **Launch and Maintain:** Launch the system and maintain it to ensure that it continues to provide a high-quality shopping experience for users.

4.6 Testing Plan

4.6.1 Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

4.6.2 Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

4.6.3 Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

4.6.4 Basics of software testing

There are two basics of software testing: black box testing and white box testing.

4.6.5 Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

4.6.6 White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

4.6.7 Types of testing

There are many types of testing like

- i. Unit Testing
- ii. Integration Testing
- iii. Functional Testing
- iv. System Testing
- v. Stress Testing
- vi. Performance Testing
- vii. Usability Testing
- viii. Acceptance Testing
- ix. Regression Testing

4.6.7.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

4.6.7.2 Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

4.6.7.3 Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

4.6.7.4 System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

4.6.7.5 Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

4.6.7.6 Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

4.6.7.7 Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

4.6.7.8 Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

4.6.7.9 Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing

4.7 Project management plan

A project management plan for an augmented reality-based online shopping system would typically include the following components:

1. **Scope:** This section would define the goals and objectives of the project, as well as the deliverables and milestones that need to be achieved. It would also outline the requirements and specifications for the augmented reality technology, as well as the integration with the online shopping system.
2. **Schedule:** This section would include a detailed timeline for the project, with tasks and activities broken down into smaller, manageable components. It would also include deadlines and milestones, as well as resource allocation and dependencies.
3. **Budget:** This section would outline the financial resources required for the project, including the costs of hardware and software, development and testing, and any other expenses related to the project. It would also include contingency plans and risk mitigation strategies.
4. **Quality management:** This section would outline the quality standards and procedures for the project, including testing and validation protocols, as well as acceptance criteria and user feedback mechanisms.

5. Human resources: This section would outline the roles and responsibilities of the project team, as well as their skills and experience requirements. It would also include recruitment and training plans, as well as performance evaluation and feedback mechanisms.
6. Communication: This section would outline the communication channels and protocols for the project, including regular status updates, progress reports, and stakeholder engagement. It would also include conflict resolution and escalation procedures.
7. Risk management: This section would outline the risks and uncertainties associated with the project, as well as the risk management strategies and contingency plans to mitigate them. It would also include a risk register and risk assessment methodology.

Overall, a project management plan for an augmented reality-based online shopping system would be a comprehensive and detailed document that outlines the goals, objectives, and requirements of the project, as well as the resources, schedules, and risks associated with it. It would also include mechanisms for quality assurance, communication, and stakeholder engagement, to ensure that the project is completed successfully and within budget and timeline constraints.

4.8 Transition/ Software to Operations Plan

A transition/software to operations (TTO/SO) plan for an augmented reality-based online shopping system would typically include the following components:

- ✓ System description: This section would provide an overview of the augmented reality-based online shopping system, including its purpose, features, and functionality. It would also describe the system architecture, hardware and software requirements, and any other relevant technical details.

- ✓ Deployment strategy: This section would outline the deployment strategy for the system, including the installation and configuration process, as well as any dependencies or prerequisites. It would also include a timeline for deployment, as well as any testing or validation procedures.
- ✓ Operations and maintenance: This section would describe the operational and maintenance procedures for the system, including routine monitoring and maintenance tasks, as well as any troubleshooting or issue resolution procedures. It would also include a schedule for software updates and patches, as well as any backup and recovery procedures.
- ✓ Support and training: This section would outline the support and training procedures for the system, including help desk support, user training, and documentation. It would also include a plan for ongoing user education and support, as well as any communication channels and feedback mechanisms.
- ✓ Performance monitoring: This section would describe the performance monitoring procedures for the system, including the metrics and benchmarks used to evaluate performance, as well as the tools and techniques used to monitor performance. It would also include a plan for ongoing performance improvement and optimization.
- ✓ Security: This section would outline the security procedures for the system, including access controls, data protection, and vulnerability management. It would also include a plan for ongoing security monitoring and risk management, as well as any incident response procedures.
- ✓ Governance: This section would describe the governance procedures for the system, including the roles and responsibilities of key stakeholders, as well as any compliance requirements or regulations. It would also include a plan for ongoing governance and oversight, as well as any reporting requirements.

CHAPTER – 5

IMPLEMENTATION DETAILS

5.1 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

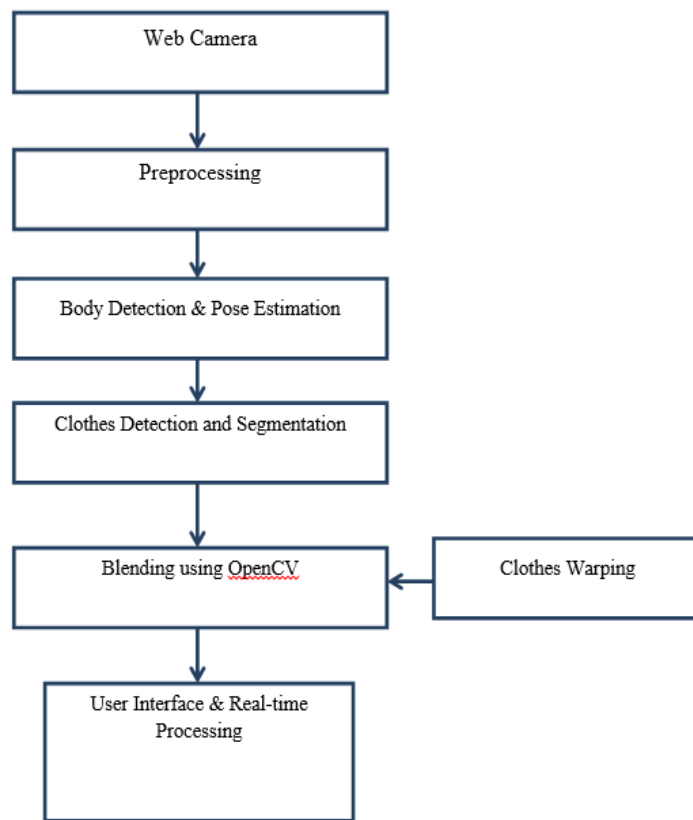


Fig 3. Data Flow Diagram

The data flows shown in the diagram are:

Image/Video Stream: This flow represents the input from the device's camera, which is received by the "Sensor/Tracking System" and used to track objects in the environment.

Augmented Content: This flow represents the output from the "Image/Video Processing System," which creates the augmented content and sends it to the "Augmented Reality Display System" for display.

Augmented Reality Display: This flow represents the output from the "Augmented Reality Display System," which displays the augmented content to the user.

5.2 CLASS DIAGRAM

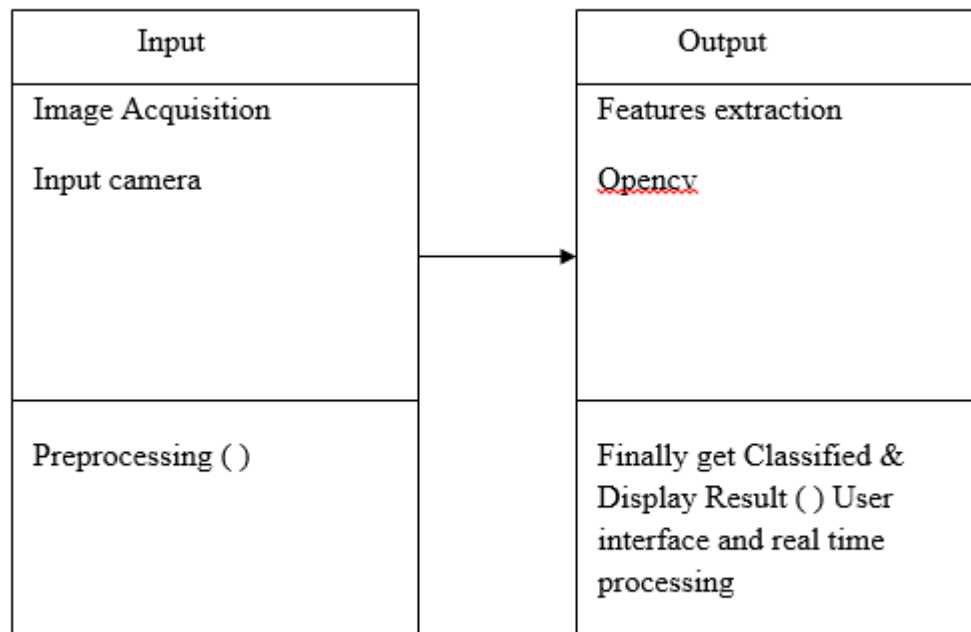


Fig 4. Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

5.3 Activity Diagram

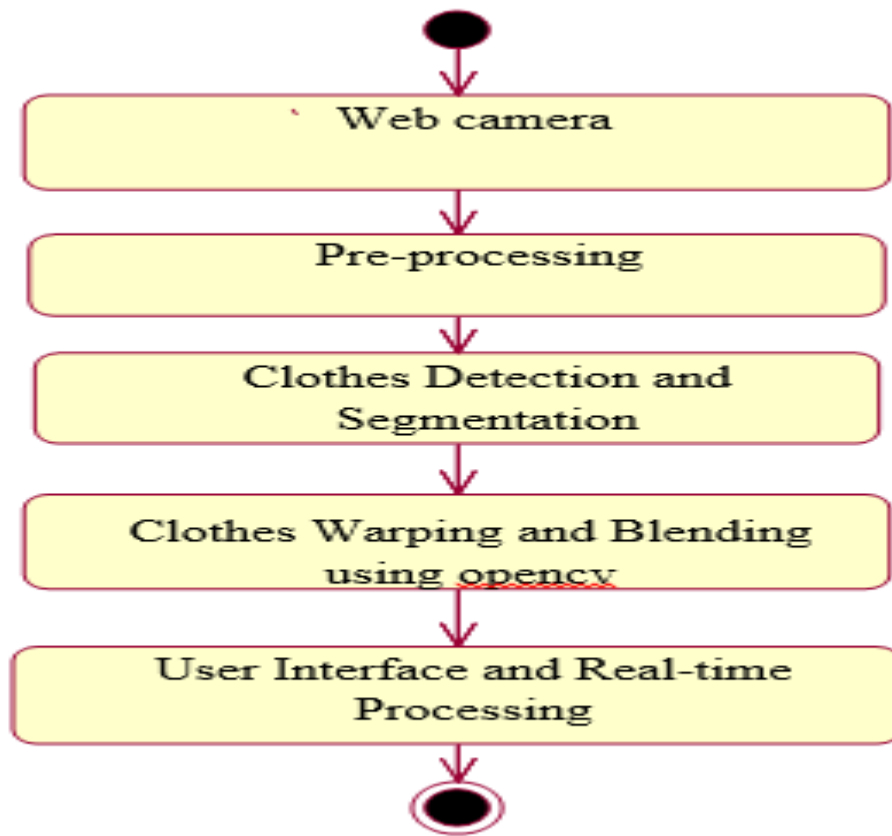


Fig 5. Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step work flows of components in a system. An activity diagram shows the overall flow of control.

5.4 Sequence Diagram

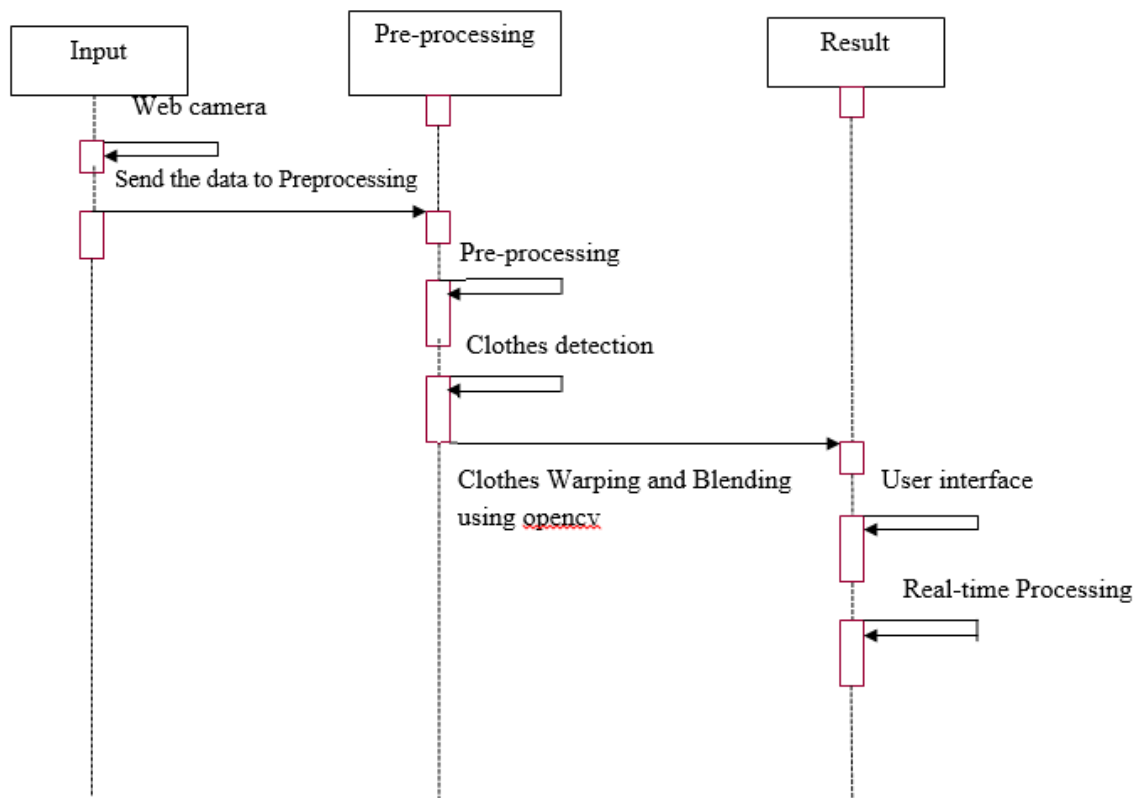


Fig 6. Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

5.5 Algorithms

5.5.1 What is Machine Learning?

Machine Learning is a system of computer algorithms that can learn from example through self-improvement without being explicitly coded by a programmer. Machine learning is a part of artificial Intelligence which combines data with statistical tools to predict an output which can be used to make actionable insights.

The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input and uses an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of tasks like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

5.5.2 Machine Learning vs. Traditional Programming

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

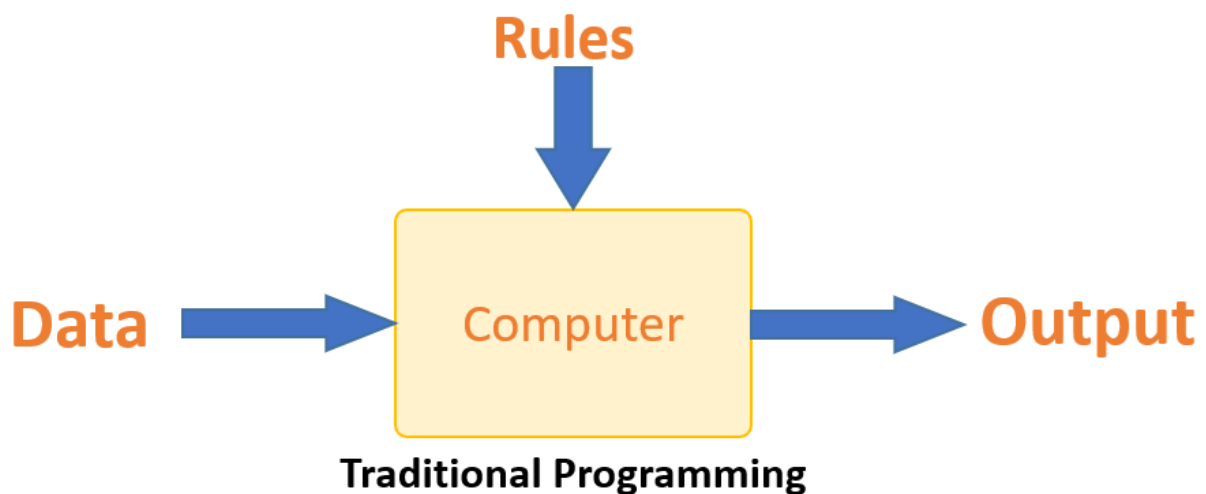


Fig 7: Traditional Programming

Machine learning is supposed to overcome this issue. The machine learns how the input and output data are correlated and it writes a rule. The programmers do not need to write new rules each time there is new data. The algorithms adapt in response to new data and experiences to improve efficacy over time.

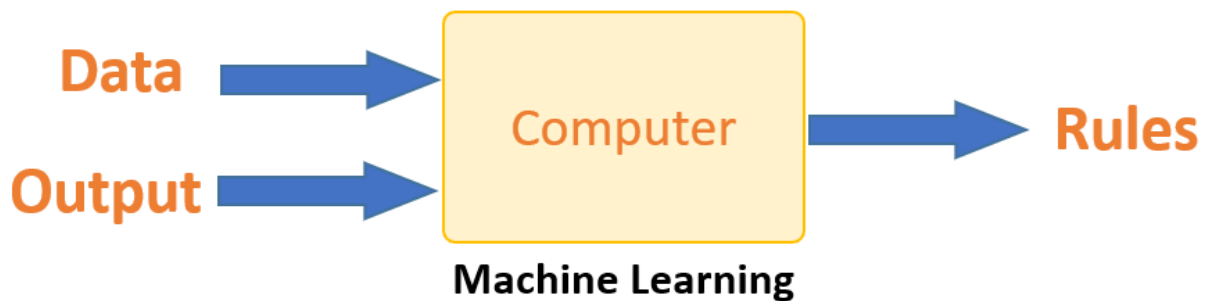


Fig 8. Machine Learning

5.5.3 How does Machine Learning Work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if it's feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning** and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector**. You can think of a feature vector as a subset of data that is used to tackle a problem.

The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.

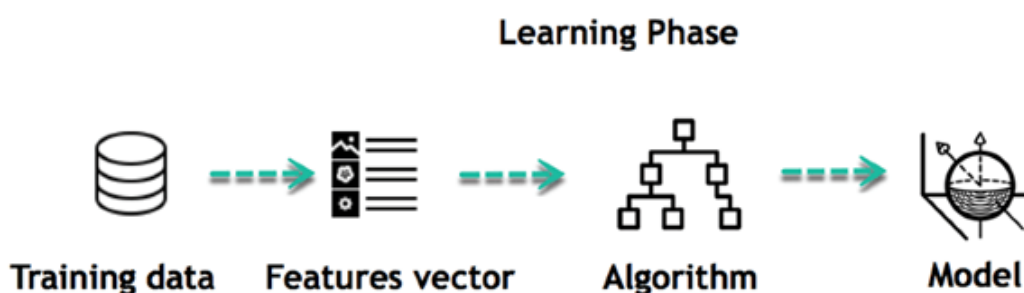


Fig 9. Learning Phase

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

5.5.4 Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

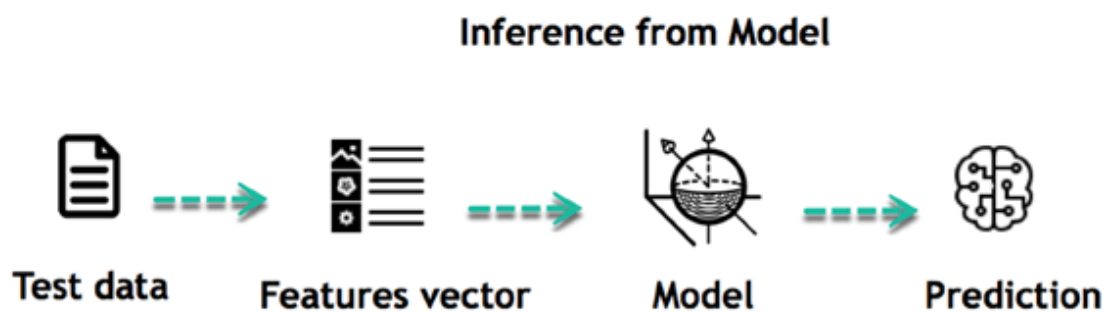


Fig 10. Inference from Model

The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

5.5.5 Machine Learning Algorithms and Where they are Used?

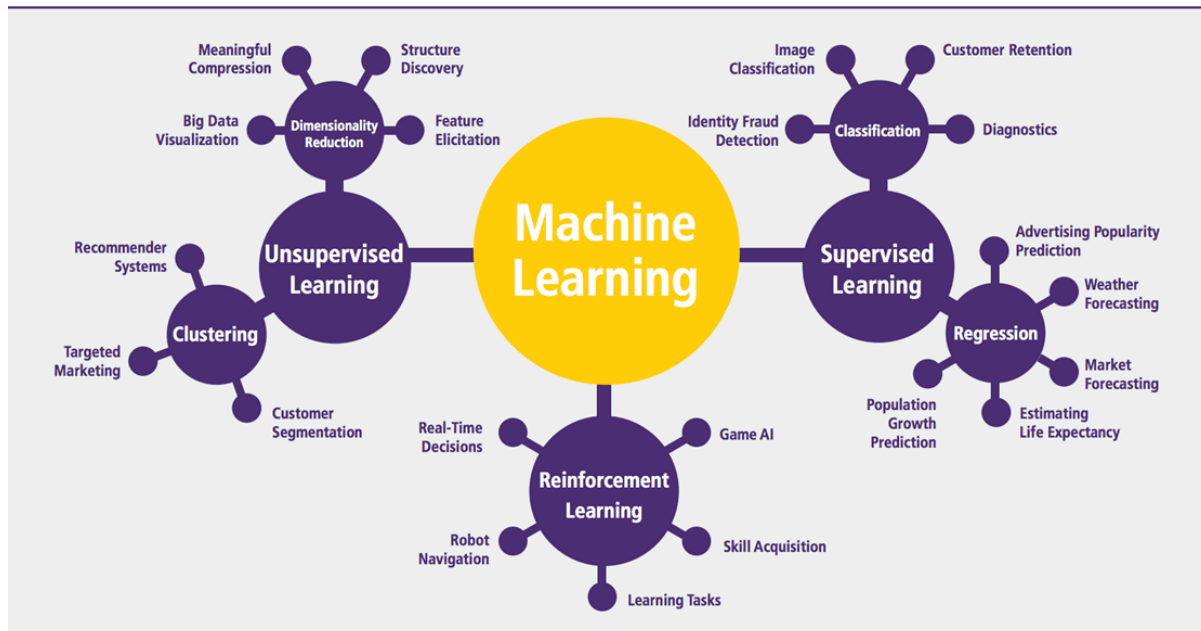


Fig 11: Machine learning Algorithms

Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

5.5.6 Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

- Classification task
- Regression task

5.5.6.1 Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above Machine learning example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

5.5.6.2 Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

Algorithm Name	Description	Type
Linear regression	Finds a way to correlate each feature to the output to help predict future values.	Regression
Logistic regression	Extension of linear regression that's used for classification tasks. The output variable is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors)	Classification
Decision tree	Highly interpretable classification or regression model that splits	Regression

	Data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made	Classification
Naive Bayes	The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior Knowledge of an event with the independent probability of each feature that can affect the event.	Regression Classification
Support vector machine	Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver.	Regression (not very common) Classification
Random forest	The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction.	Regression Classification
AdaBoost	Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	Regression Classification
Gradient-boosting trees	Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it.	Regression Classification

Table 1: Supervised Learning

5.5.7 Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

Algorithm	Description	Type
K-means clustering	Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans)	Clustering
Gaussian mixture model	A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters)	Clustering
Hierarchical clustering	Splits clusters along a hierarchical tree to form a classification system. Can be used for Cluster loyalty-card customer	Clustering
Recommender system	Help to define the relevant data for making a recommendation.	Clustering
PCA/T-SNE	Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances.	Dimension Reduction

Table 2: Unsupervised Learning

5.5.7.1 How to Choose Machine Learning Algorithm

Machine Learning (ML) algorithm:

There are plenty of machine learning algorithms. The choice of the algorithm is based on the objective. In the Machine learning example below, the task is to predict the type of flower among the three varieties. The predictions are based on the length and the width of the petal. The picture depicts the results of ten different algorithms. The picture on the top left is the data set. The data is classified into three categories: red, light blue and dark blue. There are some groupings. For instance, from the second image, everything in the upper left belongs to the red category, in the middle part, there is a mixture of uncertainty and light blue while the bottom corresponds to the dark category. The other images show different algorithms and how they try to classified the data.

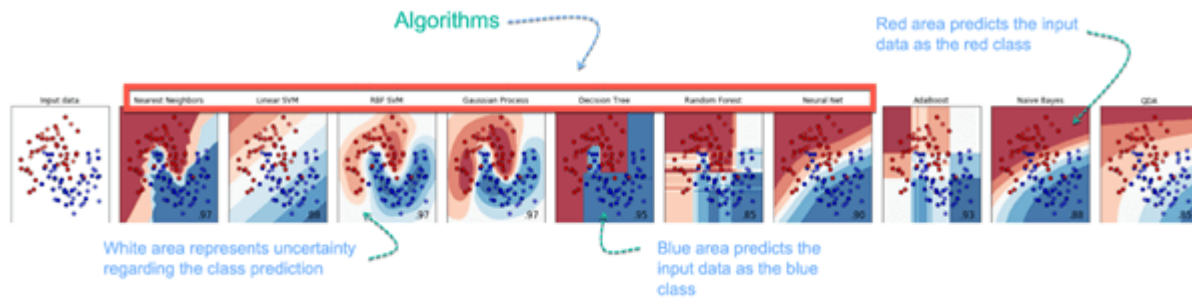


Fig 12: Classified Machine Learning Algorithms

5.5.7.2 Challenges and Limitations of Machine Learning

The primary challenge of machine learning is the lack of data or the diversity in the data set. A machine cannot learn if there is no data available. Besides, a data set with a lack of diversity gives the machine a hard time. A machine needs to have heterogeneity to learn meaningful insight. It is rare that an algorithm can extract information when there are no or few variations. It is recommended to have at least 20 observations per group to help the machine learn. This constraint leads to poor evaluation and prediction.

5.5.8 Application of Machine Learning

5.5.8.1 Augmentation:

- Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

5.5.8.2 Automation:

- Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

5.5.8.3 Finance Industry:

- Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

5.5.8.4 Government organization

- The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

5.5.8.5 Health care industry

- Health care was one of the first industry to use machine learning with image detection.

5.5.8.6 Marketing

- Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

5.5.9 Example of application of Machine Learning in Supply Chain

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse data set. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.

For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

5.5.10 Example of Machine Learning Google Car

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

5.5.11 Why is Machine Learning Important?

Machine learning is the best tool so far to analyze, understand and identify a pattern in the data. One of the main ideas behind machine learning is that the computer can be trained to automate tasks that would be exhaustive or impossible for a human being. The clear breach from the traditional analysis is that machine learning can take decisions with minimal human intervention. Take the following example for this ML tutorial; a retail agent can estimate the price of a house based on his own experience and his knowledge of the market.

A machine can be trained to translate the knowledge of an expert into features. The features are all the characteristics of a house, neighborhood, economic environment, etc. that make the price difference. For the expert, it took him probably some years to master the art of estimate the price of a house. His expertise is getting better and better after each sale.

For the machine, it takes millions of data, (i.e., example) to master this art. At the very

beginning of its learning, the machine makes a mistake, somehow like the junior salesman. Once the machine sees all the example, it got enough knowledge to make its estimation. At the same time, with incredible accuracy. The machine is also able to adjust its mistake accordingly.

Most of the big company have understood the value of machine learning and holding data. Mc Kinsey have estimated that the value of analytic ranges from \$9.5 trillion to \$15.4 trillion while \$5 to 7 trillion can be attributed to the most advanced AI techniques.

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytic.

Overview

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed.

For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST data set of handwritten digits has often been used.

5.5.12 Machine learning approaches

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

5.5.12.1 Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

5.5.12.2 Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

5.5.12.3 Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

Other approaches have been developed which don't fit neatly into this three-fold

categorization, and sometimes more than one is used by the same machine learning system. For example, topic modeling, dimensional reduction or meta learning.

As of 2020, deep learning has become the dominant approach for much ongoing work in the field of machine learning.

5.5.13 History and relationships to other fields

The term machine learning was coined in 1959 by Arthur Samuel, an American IBM and pioneer in the field of computer gaming and artificial intelligence. A representative book of the machine learning research during the 1960s was the Nilsson's book on Learning Machines, dealing mostly with machine learning for pattern classification. Interest related to pattern recognition continued into the 1970s, as described by Duda and Hart in 1973. In 1981 a report was given on using teaching strategies so that a neural network learns to recognize 40 characters (26 letters, 10 digits, and 4 special symbols) from a computer terminal.

Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T , as measured by P , improves with experience E ." This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?".

Modern day machine learning has two objectives, one is to classify data based on models which have been developed, the other purpose is to make predictions for future outcomes based on these models. A hypothetical algorithm specific to classifying data may use computer vision of moles coupled with supervised learning in order to train it to classify the cancerous moles. Whereas, a machine learning algorithm for stock trading may inform the trader of future potential predictions.

5.5.14 GENETIC ALGORITHM

Genetic algorithm is a type of meta-heuristic optimization algorithm that is inspired by the process of natural selection and evolution. It is used to solve complex optimization problems where traditional methods may not be effective.

The genetic algorithm works by starting with a population of potential solutions (i.e. chromosomes), and then iteratively applying selection, crossover, and mutation operations to produce new generations of solutions. The algorithm evaluates each solution based on a fitness function that quantifies how well it performs in solving the optimization problem. The fittest individuals are more likely to be selected for the next generation, while less fit individuals are less likely to be selected.

Genetic algorithms have been used in a variety of applications, including engineering design, financial forecasting, and image processing, among others. In the context of augmented reality based online shopping system, genetic algorithms can be used to optimize various aspects of the system, such as the placement of virtual objects, the design of user interfaces, or the recommendation of products based on user preferences.

One of the challenges of designing an augmented reality based online shopping system is to find the optimal placement of virtual objects such as furniture, clothing items, or accessories within the user's physical environment. This requires considering a large number of factors such as the size and shape of the object, the user's preferences and requirements, and the available space and lighting conditions.

To tackle this problem, a genetic algorithm can be used to generate and evaluate different placement configurations and select the ones that are more likely to satisfy the user's needs. Here is a high-level overview of the steps involved:

- I. Encoding: The placement configuration can be encoded as a chromosome, where each gene represents a parameter of the configuration such as the

position, orientation, or scale of the virtual object.

- II. Initialization: A population of initial placement configurations can be randomly generated or based on some prior knowledge of the user's preferences.
- III. Evaluation: Each placement configuration is evaluated based on a fitness function that takes into account various factors such as the user's preferences, the spatial constraints, and the aesthetics of the arrangement.
- IV. Selection: The fittest placement configurations are selected to form the next generation of solutions. This can be done using various selection methods such as tournament selection or roulette wheel selection.
- V. Crossover: The selected placement configurations can be combined through crossover operations to create new offspring configurations.
- VI. Mutation: The offspring configurations can be randomly mutated to introduce new variations in the population.

Repeat: Steps 3 to 6 are repeated for multiple generations until a satisfactory placement configuration is found.

By using a genetic algorithm, the augmented reality based online shopping system can optimize the placement of virtual objects based on the user's preferences and the available physical environment. This can lead to a more personalized and engaging shopping experience, and increase the chances of the user making a purchase.

5.6 Testing

When it comes to testing genetic algorithms, there are several aspects that should be taken into consideration in order to ensure their correctness and effectiveness. Here are some testing plans that can be used for genetic algorithms:

5.6.1 Functional Testing: This type of testing focuses on testing the core functionality of the genetic algorithm, such as the initialization, selection, crossover, mutation, and termination processes. This can be done by generating test cases that cover various scenarios and inputs, and verifying that the output meets the expected results. For example, testing if the algorithm can generate diverse and optimal solutions for a given problem, or if it can handle different types of constraints and objectives.

5.6.2 Performance Testing: This type of testing focuses on testing the performance and scalability of the genetic algorithm, such as the execution time, memory usage, and parallelism. This can be done by measuring the algorithm's performance under different load conditions and input sizes, and comparing it to the expected results. For example, testing if the algorithm can handle large populations and generations, or if it can take advantage of multi-core processors or GPUs for faster execution.

5.6.3 Integration Testing: This type of testing focuses on testing the integration of the genetic algorithm with other components of the system, such as the fitness function, the encoding and decoding mechanisms, or the visualization and reporting tools. This can be done by generating test cases that cover different combinations of inputs and components, and verifying that the output meets the expected results. For example, testing if the algorithm can work with different types of fitness functions, or if it can produce meaningful and useful reports and visualizations.

5.6.4 Robustness Testing: This type of testing focuses on testing the robustness and fault-tolerance of the genetic algorithm, such as the ability to handle errors, exceptions, and unexpected inputs or outputs. This can be done by generating test cases that simulate different types of errors and faults, and verifying that the

algorithm can recover or handle them gracefully. For example, testing if the algorithm can handle missing or invalid input data, or if it can detect and prevent common pitfalls such as premature convergence or over fitting.

By using these testing plans, genetic algorithms can be thoroughly tested and validated to ensure their correctness, effectiveness, and robustness in various scenarios and use cases.

5.7 Frameworks

Flask is a Python web framework that allows developers to quickly build web applications with minimal overhead and boilerplate code. Flask is lightweight, modular, and easy to learn, making it a popular choice for small to medium-sized web applications and APIs.

Here are some key features and benefits of Flask:

5.7.1 Lightweight: Flask has a small code base and minimal dependencies, which makes it easy to install, configure, and deploy. Flask also follows the "micro" philosophy, which means it focuses on doing one thing well - handling HTTP requests and responses - and allows developers to add functionality using extensions and plugins.

5.7.2 Modular: Flask is designed to be modular and flexible, which allows developers to customize and extend the framework to fit their specific needs. Flask uses a "blueprint" system to organize routes and views, which makes it easy to break down large applications into smaller, reusable components.

5.7.3 Easy to learn: Flask has a simple and intuitive API that is easy to learn, even for beginners. Flask also has a large and active community, which provides documentation, tutorials, and support to help developers get up to speed quickly.

5.7.4 Pythonic: Flask follows the Python philosophy of "batteries included" and "easy

to read, easy to write". Flask uses Python idioms and conventions, which makes it easy to integrate with other Python libraries and frameworks, and allows developers to write clean and maintainable code.

5.7.5 Scalable: Flask can be used to build scalable and high-performance web applications and APIs, thanks to its support for asynchronous programming, caching, and load balancing. Flask can also be deployed on a variety of platforms and cloud services, such as Heroku, AWS, and GCP.

Flask is a versatile and powerful web framework that allows developers to build web applications and APIs quickly and efficiently, using Python's simplicity and elegance.

CHAPTER - 6

RESULTS AND DISCUSSION

The results of an augmented reality-based online shopping system can be quite positive, as this technology has the potential to enhance the customer experience and drive sales. Here are some potential benefits and outcomes of using an augmented reality-based online shopping system:

Enhanced customer experience: By allowing customers to visualize and interact with products in a realistic and immersive way, augmented reality can help to enhance the customer experience and make online shopping more engaging and satisfying. This can lead to increased customer loyalty and repeat business.

- I. Increased sales: Augmented reality can help to increase sales by providing customers with a better understanding of products and their features, which can lead to increased confidence and purchase intent. Augmented reality can also help to reduce returns and product dissatisfaction by providing more accurate and detailed product information.
- II. Competitive advantage: By adopting augmented reality technology, online retailers can differentiate themselves from competitors and position themselves as innovative and customer-centric. This can help to attract new customers and retain existing ones, as well as increase brand awareness and loyalty.
- III. Operational efficiency: Augmented reality can also help to improve operational efficiency by reducing the need for physical product samples and prototypes, as well as by streamlining the product visualization and customization process. This can lead to reduced costs and faster time-to-market for new products.

IV. Data insights: Augmented reality can also provide valuable data insights into customer behavior and preferences, such as which products are most popular or which features are most important. This can help online retailers to optimize their product offerings and marketing strategies, as well as improve customer segmentation and targeting.

An augmented reality-based online shopping system has the potential to deliver significant benefits and outcomes for both customers and retailers, by enhancing the shopping experience, increasing sales, improving efficiency, and providing valuable data insights.

CHAPTER – 7

CONCLUSION

7.1 conclusion

The rise of online shopping and the desire for customers to fully utilize it while ensuring complete personal satisfaction with their dress purchases has necessitated the development of a system that virtually dresses individuals in their chosen clothing. To meet this requirement, our proposed system has been developed to work reliably without the need for external light adjusters or any hardware devices, which are commonly used in current algorithms and create limitations on their usage. This proposed system offers a more accessible solution for all people, enabling them to try on different clothes with fewer restrictions, providing greater pleasure for online shoppers.

In conclusion, the Virtual Trial Room for online shopping is a revolutionary technology that has the potential to transform the way people shop for clothes online. The development of this technology using advanced technologies like augmented reality, computer vision, and machine learning has addressed the challenge of not being able to try on clothes before making a purchase, providing an immersive and personalized shopping experience to customers.

The Virtual Trial Room offers several benefits, including the ability to visualize how clothes would look on the customer, the convenience of shopping from home, and the ability to compare different styles and sizes easily. It also helps businesses by reducing the likelihood of returns due to sizing issues and increasing customer satisfaction.

The project's success relies on the simple usage of the system without any additional hardware except a web camera which is already available in Laptop or PC. Additionally, the use of OpenCV in the project provides several advantages, including real-time image processing, cost-effectiveness, and scalability.

7.2 Future Work

1. Improving the user experience: AR shopping systems have the potential to revolutionize online shopping by providing a more immersive and interactive experience for users. However, to fully realize this potential, the user experience needs to be optimized. Future work could focus on improving the speed, accuracy, and ease-of-use of AR shopping systems, as well as developing more intuitive user interfaces.
2. Enhancing product visualization: One of the main benefits of AR shopping systems is that they allow users to see products in 3D and in real-world contexts. Future work could focus on improving the quality and realism of product visualizations, as well as developing new visualization techniques that enhance the user experience.
3. Personalization: AR shopping systems have the potential to offer highly personalized experiences for users. Future work could focus on developing algorithms and techniques that enable AR shopping systems to tailor product recommendations and visualizations to each user's preferences and needs.
4. Social shopping: AR shopping systems could also enable social shopping experiences where users can shop with friends and family members in a virtual environment. Future work could focus on developing social shopping features that enhance the sense of community and collaboration among users.
5. Integration with other technologies: AR shopping systems could be integrated with other technologies such as voice assistants, smart home devices, and wearable technology. Future work could focus on developing integrations that enhance the overall shopping experience and make it more seamless and convenient for users.

REFERENCES

- [1]. Bianco.S, Schettini.R, “Color constancy using faces”, in 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 65–72, 2012.
- [2]. Brouet.R, Sheffer.A, Boissieux.L, Cani.M-P, “Design preserving garment transfer”, *ACM Trans. Graph.*, vol. 31, No. 4, pp. 36:1–36:11, Jul. 2012.
- [3]. Cordier.F, Lee.W, Seo.H, Magnenat-Thalmann.N, “From 2D Photos of Yourself to Virtual Try-on Dress on the Web”, Springer, pp. 31–46, 2001.
- [4]. Cordier.F, Seo.H, Magnenat-Thalmann.N, “Made-to-measure technologies for an online clothing store”, *IEEE Comput. Graph. Appl.*, vol. 23, no. 1, pp. 38–48, Jan. 2003.
- [5]. Ehara.J, Saito.H, “Texture overlay for virtual clothing based on pca of silhouettes”, in *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ser. ISMAR '06. IEEE Computer Society, pp.139–142, 2006.
- [6]. Guan.P, Freifeld.O, Black.M.J, “A 2D human body model dressed in eigen clothing”, in *Proceedings of the 11th European Conference on Computer Vision: Part I*, pp. 285–298, 2010.
- [7]. Luh.G-C, “Face detection using combination of skin colour pixel detection and viola jones face detector”, in 2014 IEEE International Conference on Machine Learning and Cybernetics(ICMLC), pp.364-370, 2014.
- [8]. Mignotte.M, “Segmentation by Fusion of Histogram based K-Means Clusters in different color space” *IEEE Transactions on Image Processing*, Vol.17, pp.780-787, 2008.
- [9]. Protopsaltou.D, Luible.C, Arevalo-Poizat.M, Magnenat Thalmann.N, “A body and garment creation method for an internet based virtual fitting room”, in *Proc. Computer Graphics International 2002 (CGI '02)*. Springer, pp. 105–122, 2002.

- [10]. Srinivasan.K, Porkumaran.K, Narayanan.G, “Background subtraction techniques for human body segmentation in indoor video surveillance”, *Journal of Scientific and Industrial Research*, Vol.73, pp.342-345, 2014.
- [11]. Srinivasan.K, Porkumaran.K, Narayanan.G, “Enhanced Background Subtraction Techniques for Monocular Video Applications”, *International Journal of Image Processing and Applications*, Vol. 1, no. 2, pp.87-93, 2010.
- [12]. Srinivasan.K, Porkumaran.K, Narayanan.G, “Intelligent human body tracking, modeling and activity analysis of video surveillance system: A Survey”, *Journal of convergence in engineering, technology and science*, Vol.1, pp.1-8, 2009.
- [13]. Srinivasan.K, Porkumaran.K, Narayanan.G, “Skin colour segmentation based 2D and 3D human pose modelling using Discrete Wavelet Transform”, *Journal of Pattern recognition and image Analysis*, Springer, Vol.21, pp.740-753, 2011.
- [14]. Tong.J, Zhou.J, Liu.L, Pan.Z, Yan.H, “Scanning 3D full human bodies using kinects”, *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE Virtual Reality)*, Vol. 18, No. 4, pp. 643–650, 2012.
- [15]. Weng.Y, Xu.W, Wu.Y, Zhou.K, Guo.B, “2D shape deformation using nonlinear least squares optimization”, *Vis. Comput.*, vol. 22, no. 9, pp. 653–660, 2006.
- [16]. Xu.W, Umentani.N, Chao.Q, Mao.J, Jin.X, Tong.X, “Sensitivity optimized rigging for example-based real-time clothing synthesis”, *ACM Trans. Graph. (Proc. of SIGGRAPH 2014)*, Vol. 33, No. 4, Aug. 2014.
- [17]. Zhou.Z, Shu.B, Zhuo.S, Deng.X, Tan.P, Lin.S, “Image-based clothes animation for virtual fitting”, in *SIGGRAPH Asia 2012 Technical Briefs*, pp. 33:1–33:4, 2012.

APPENDIX

SOURCE CODE

```
from flask import Flask, render_template, Response, redirect, request
from camera import VideoCamera
import os
import sys
app = Flask(__name__)

CART=[]
```

```
@app.route('/checkout', methods=['POST', 'GET'])
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)
    #return Response(gen(VideoCamera()), mimetype='multipart/x-mixed-replace; boundary=frame')
```

```
@app.route('/tryall', methods = ['POST', 'GET'])
def tryall():
    print("YESSS")
    if request.method == 'POST':
        cart = request.form['mydata'].replace(',', '/')
        print("tryall cart type= ", type(cart))
        print("tryall cart= ", cart)
        os.system('python test.py ' + cart)
        return redirect('http://127.0.0.1:5000/', code=302, Response=None)
```

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

```
def gen(camera):
    while True:
        frame = camera.get_frame()
        #print("frame= ", 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(debug=True)
```

```
from tkinter import *
from PIL import Image
from PIL import ImageTk
import cv2, threading, os, time
from threading import Thread
from os import listdir
from os.path import isfile, join
import sys
import dlib
from imutils import face_utils, rotate_bound
import math
```

```
def put_sprite(num):
    global SPRITES, BTNS
    SPRITES[num] = (1 - SPRITES[num])
    # if SPRITES[num]:
    #     BTNS[num].config(relief=SUNKEN)
    # else:
    #     BTNS[num].config(relief=RAISED)
```

```
def draw_sprite(frame, sprite, x_offset, y_offset):
    print("sprite>>>>>>", sprite.shape, "type=", type(sprite.shape))
    (h,w) = (sprite.shape[0], sprite.shape[1])
    (imgH,imgW) = (frame.shape[0], frame.shape[1])
```

```
if y_offset+h >= imgH:
    sprite = sprite[0:imgH-y_offset,:,:]
```

```
if x_offset+w >= imgW:
    sprite = sprite[:,0:imgW-x_offset,:]
```

```
if x_offset < 0:
    sprite = sprite[:,abs(x_offset)::,:]
```

w = sprite.shape[1]

x_offset = 0

```
for c in range(3):
    try:
        frame[y_offset:y_offset+h, x_offset:x_offset+w, c] = \
            sprite[:,c] * (sprite[:,c]/255.0) + frame[y_offset:y_offset+h, x_offset:x_offset+w, c]
    * (1.0 - sprite[:,c]/255.0)
    except Exception as e:
        print(e)
        pass
return frame
```

```
def adjust_sprite2head(sprite, head_width, head_ypos, ontop = True):
    (h_sprite,w_sprite) = (sprite.shape[0], sprite.shape[1])
    factor = 1.0*head_width/w_sprite
    sprite = cv2.resize(sprite, (0,0), fx=factor, fy=factor)
    (h_sprite,w_sprite) = (sprite.shape[0], sprite.shape[1])
```

```

y_orig = head_ypos-h_sprite if ontop else head_ypos
if (y_orig < 0):
    sprite = sprite[abs(y_orig):,:,:]
    y_orig = 0
return (sprite, y_orig)

```

```

def apply_sprite(image, path2sprite,w,x,y, angle, ontop = True):
    sprite = cv2.imread(path2sprite,-1)
    sprite = rotate_bound(sprite, angle)
    (sprite, y_final) = adjust_sprite2head(sprite, w, y, ontop)
    image = draw_sprite(image,sprite,x, y_final)

```

```

def calculate_inclination(point1, point2):
    x1,x2,y1,y2 = point1[0], point2[0], point1[1], point2[1]
    incl = 180/math.pi*math.atan((float(y2-y1))/(x2-x1))
    return incl

```

```

def calculate_boundingBox(list_coordinates):
    x = min(list_coordinates[:,0])
    y = min(list_coordinates[:,1])
    w = max(list_coordinates[:,0]) - x
    h = max(list_coordinates[:,1]) - y
    return (x,y,w,h)

```

```

def detectUpperBody(image):
    cascadePath = 'data/haarcascade_upperbody.xml'
    result = image.copy()
    imageGray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    cascade = cv2.CascadeClassifier(cascadePath)
    Rect = cascade.detectMultiScale(imageGray, scaleFactor=1.1, minNeighbors=1, minSize=(1,1))
    if len(Rect) <= 0:
        return False
    else:
        return Rect

```

```

def get_face_boundingBox(points, face_part):
    if face_part == 1:
        (x,y,w,h) = calculate_boundingBox(points[17:22])
    elif face_part == 2:
        (x,y,w,h) = calculate_boundingBox(points[22:27])
    elif face_part == 3:
        (x,y,w,h) = calculate_boundingBox(points[36:42])
    elif face_part == 4:
        (x,y,w,h) = calculate_boundingBox(points[42:48])
    elif face_part == 5:
        (x,y,w,h) = calculate_boundingBox(points[29:36])
    elif face_part == 6:
        (x,y,w,h) = calculate_boundingBox(points[0:17])
    elif face_part == 7:
        # (x,y,w,h) = calculate_boundingBox(points[48:68]) #mouth
        (x,y,w,h) = calculate_boundingBox(points[1:5])
    elif face_part == 8:
        (x,y,w,h) = calculate_boundingBox(points[12:16])
    return (x,y,w,h)

```

```
image_path = ''
```

```
def add_sprite(img):
    global image_path
    image_path = img
    print("rsplit of imgpath>>>>>>",img.rsplit('/',1))
    print(">>>>>>>>>>",int(img.rsplit('/',1)[0][-1]))
    put_sprite(int(img.rsplit('/',1)[0][-1])) #here it will return the integer value from 1 to 6, which
will denotes that which catagory is the apparel belongs to i.e either frock or tops or etc.
```

```
#Principal Loop where openCV (magic) occurs
# Face detection starts from here
```

```
def cvloop(run_event):
    global panelA
    global SPRITES
    global image_path
    i = 0
    video_capture = cv2.VideoCapture(0) #read from webcam
    (x,y,w,h) = (0,0,10,10) #whatever initial values
```

```
#Filters path
detector = dlib.get_frontal_face_detector()
```

```
model = "data/shape_predictor_68_face_landmarks.dat"
predictor = dlib.shape_predictor(model) # link to model:
http://dlib.net/files/shape\_predictor\_68\_face\_landmarks.dat.bz2
```

```
while run_event.is_set():
    ret, image = video_capture.read()
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) #converting the colour image to gray
    faces = detector(gray, 0)
```

```
    for face in faces:
        (x,y,w,h) = (face.left(), face.top(), face.width(), face.height()) #reading the coordinates
in sequence
```

```
        shape = predictor(gray, face)
        shape = face_utils.shape_to_np(shape)
        incl = calculate_inclination(shape[17], shape[26]) #inclination based on eyebrows
```

```
        # condition to see if mouth is open
        is_mouth_open = (shape[66][1] -shape[62][1]) >= 10 #y coordiantes of landmark points of
lips
```

```
        if SPRITES[0]:
            apply_sprite(image,image_path,w,x,y+40, incl, ontop = True)
```

```
        if SPRITES[3]:#Tiara
            apply_sprite(image, image_path,w+45,x-20,y+15, incl, ontop = True)
```

```
        #Necklaces
        if SPRITES[1]:
            (x1,y1,w1,h1) = get_face_boundingBox(shape, 6)
            apply_sprite(image, image_path,w1,x1,y1+110, incl, ontop = False)
```

```
        #Goggles
```

```
if SPRITES[6]:
    (x3,y3,_,h3) = get_face_boundingBox(shape, 1)
    apply_sprite(image, image_path,w,x,y3-10, incl, ontop = False)
```

```
#Earrings
(x0,y0,w0,h0) = get_face_boundingBox(shape, 6) #bound box of mouth
if SPRITES[2]:
    (x3,y3,w3,h3) = get_face_boundingBox(shape, 7) #nose
    apply_sprite(image, image_path,w3,x3-40,y3+30, incl,ontop=False)
    (x3,y3,w3,h3) = get_face_boundingBox(shape, 8) #nose
    apply_sprite(image, image_path,w3,x3+30,y3+75, incl)
```

```
# if SPRITES[5]:
#     apply_sprite(image,image_path,w,x,y, incl, ontop = True)

#Frocks
if SPRITES[5]:
    (x1,y1,w1,h1) = get_face_boundingbox(shape, 8)
    apply_sprite(image, image_path,w1+590,x1-300,y1+70, incl, ontop = False)
```

```
#Tops
if SPRITES[4]:
    # (x,y,w,h) = (0,0,10,10)
    # apply_sprite2feature(image, IMAGES[7][ACTIVE_IMAGES[7]], fullbody, w//4, 2*h//3, h//2,
True, w//2, x, y, w, h)
    (x1,y1,w1,h1) = get_face_boundingbox(shape, 8)
    apply_sprite(image, image_path,w1+300,x1-200,y1+40, incl, ontop = False)
```

```
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
image = Image.fromarray(image)
image = ImageTk.PhotoImage(image)
#image = ImageTk.PhotoImage(image.resize((500,500))) #this is for photo mode
panelA.configure(image=image)
panelA.image = image
```

```
video_capture.release()
```

```
# Initialize GUI object
root = Tk()
root.title("Virtual trial room")
this_dir = os.path.dirname(os.path.realpath(__file__))
btn1 = None
```

```
def try_on(image_path):
    btn1 = Button(root, text="Try it ON", command = lambda:add_sprite(image_path))
    btn1.pack(side="top", fill="both", expand="no", padx="5", pady="5")
panelA = Label(root)
panelA.pack( padx=10, pady=10)
```

```
SPRITES = [0,0,0,0,0,0,0,0]
BTNS = [btn1]
```

```
print(">>>>>>>>>>>>>>>>>>>",sys.argv[1])  
try_on(sys.argv[1]) #here sys.argv will return the path of apparels.  
run_event = threading.Event()  
run_event.set()  
action = Thread(target=cvloop, args=(run_event,))  
action.setDaemon(True)
```

```
action.start()
```

```
def terminate():
    global root, run_event, action
    run_event.clear()
    time.sleep(1)
    root.destroy()
```

```
root.protocol("WM_DELETE_WINDOW", terminate)
root.mainloop()
```

```
from tkinter_scroll import *
from tkinter import *
from PIL import Image
from PIL import ImageTk
import cv2, threading, os, time
from threading import Thread
from os import listdir
from os.path import isfile, join
import dlib
from imutils import face_utils, rotate_bound
import math
import sys
ACTIVE_IMAGES=[0 for i in range(100)]
def put_sprite(num, k):
    global SPRITES, BTNS
    SPRITES[num] = (1 - SPRITES[num])
    if SPRITES[num]:
        ACTIVE_IMAGES[num] = k
        BTNS[num].config(relief=SUNKEN) #buttons to click
    else:
        BTNS[num].config(relief=RAISED)

def draw_sprite(frame, sprite, x_offset, y_offset):
    (h,w) = (sprite.shape[0], sprite.shape[1])
    (imgH,imgW) = (frame.shape[0], frame.shape[1])
    if y_offset+h >= imgH:
        sprite = sprite[0:imgH-y_offset,:,:]
    if x_offset+w >= imgW:
        sprite = sprite[:,0:imgW-x_offset,:]
    if x_offset < 0:
        sprite = sprite[:,abs(x_offset):,:,:]
        w = sprite.shape[1]
        x_offset = 0
    for c in range(3):
        frame[y_offset:y_offset+h, x_offset:x_offset+w, c] = \
            sprite[:, :,c] * (sprite[:, :,3]/255.0) + frame[y_offset:y_offset+h, x_offset:x_offset+w, c]
    * (1.0 - sprite[:, :,3]/255.0)
    return frame
```

```
def adjust_sprite2head(sprite, head_width, head_ypos, ontop = True):
    (h_sprite,w_sprite) = (sprite.shape[0], sprite.shape[1])
    factor = 1.0*head_width/w_sprite
    sprite = cv2.resize(sprite, (0,0), fx=factor, fy=factor)
    (h_sprite,w_sprite) = (sprite.shape[0], sprite.shape[1])
    y_orig = head_ypos-h_sprite if ontop else head_ypos
    if (y_orig < 0):
        sprite = sprite[abs(y_orig):,:,:]
```

```

        y_orig = 0
    return (sprite, y_orig)

```

```

def apply_sprite2feature(image, sprite_path, haar_filter, x_offset, y_offset, y_offset_image,
adjust2feature, desired_width, x, y, w, h):
    sprite = cv2.imread(sprite_path,-1)
    (h_sprite,w_sprite) = (sprite.shape[0], sprite.shape[1])

```

```

    xpos = x + x_offset
    ypos = y + y_offset
    factor = 1.0*desired_width/w_sprite

```

```

    sub_img = image[y + y_offset_image:y+h,x:x+w,:]

```

```

    feature = apply_Haar_filter(sub_img, haar_filter, 1.3 , 10, 10)
    if len(feature)!=0:
        xpos, ypos = x, y + feature[0,1] #adjust only to feature in y axis (eyes)

```

```

        if adjust2feature:
            size_mustache = 1.2 #how many times bigger than mouth
            factor = 1.0*(feature[0,2]*size_mustache)/w_sprite
            xpos = x + feature[0,0] - int(feature[0,2]*(size_mustache-1)//2) #centered respect to
width
            ypos = y + y_offset_image + feature[0,1] - int(h_sprite*factor) #right on top

```

```

    sprite = cv2.resize(sprite, (0,0), fx=factor, fy=factor)
    image = draw_sprite(image, sprite, xpos, ypos)

```

```

def apply_sprite(image, path2sprite,w,x,y, angle, ontop = True):
    sprite = cv2.imread(path2sprite,-1)
    sprite = rotate_bound(sprite, angle)
    (sprite, y_final) = adjust_sprite2head(sprite, w, y, ontop)
    image = draw_sprite(image, sprite, x, y_final)

```

```

def calculate_inclination(point1, point2):
    x1,x2,y1,y2 = point1[0], point2[0], point1[1], point2[1]
    incl = 180/math.pi*math.atan((float(y2-y1))/(x2-x1))
    return incl

```

```

def calculate_boundingbox(list_coordinates):
    x = min(list_coordinates[:,0])
    y = min(list_coordinates[:,1])
    w = max(list_coordinates[:,0]) - x
    h = max(list_coordinates[:,1]) - y
    return (x,y,w,h)

```

```

def apply_Haar_filter(img, haar_cascade,scaleFact = 1.05, minNeigh = 3, minSizeW = 30):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

```

```

    features = haar_cascade.detectMultiScale(
        gray,
        scaleFactor=scaleFact,
        minNeighbors=minNeigh,
        minSize=(minSizeW, minSizeW),
        flags=cv2.CASCADE_SCALE_IMAGE
    )
    return features

```

```
def get_face_boundingbox(points, face_part):
    if face_part == 1:
        (x,y,w,h) = calculate_boundingbox(points[17:22])
    elif face_part == 2:
        (x,y,w,h) = calculate_boundingbox(points[22:27])
    elif face_part == 3:
        (x,y,w,h) = calculate_boundingbox(points[36:42])
    elif face_part == 4:
        (x,y,w,h) = calculate_boundingbox(points[42:48])
    elif face_part == 5:
        (x,y,w,h) = calculate_boundingbox(points[29:36])
    elif face_part == 6:
        (x,y,w,h) = calculate_boundingbox(points[1:17])
    elif face_part == 7:
        (x,y,w,h) = calculate_boundingbox(points[0:6])
    elif face_part == 8:
        (x,y,w,h) = calculate_boundingbox(points[11:17])
    return (x,y,w,h)
```

```
def cvloop(run_event):
    global ctr_mid
    global SPRITES
    i = 0
    video_capture = cv2.VideoCapture(0)
    video_capture.set(3,2048)
    video_capture.set(4,2048)
    (x,y,w,h) = (0,0,10,10)
    detector = dlib.get_frontal_face_detector()
    fullbody = cv2.CascadeClassifier('data/haarcascade_fullbody.xml')
    model = "data/shape_predictor_68_face_landmarks.dat"
    predictor = dlib.shape_predictor(model) # link to model:
http://dlib.net/files/shape\_predictor\_68\_face\_landmarks.dat.bz2
    while run_event.is_set():
        ret, image = video_capture.read()
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        faces = detector(gray, 0)
        for face in faces:
            (x,y,w,h) = (face.left(), face.top(), face.width(), face.height())
            shape = predictor(gray, face)
            shape = face_utils.shape_to_np(shape)
            incl = calculate_inclination(shape[17], shape[26])
            is_mouth_open = (shape[66][1] - shape[62][1]) >= 10

            if SPRITES[3]:#Tiara
                apply_sprite(image, IMAGES[3][ACTIVE_IMAGES[3]],w+45,x-20,y+20, incl, ontop = True)
```

```
            #Necklaces
            if SPRITES[1]:
                (x1,y1,w1,h1) = get_face_boundingbox(shape, 6)
                apply_sprite(image, IMAGES[1][ACTIVE_IMAGES[1]],w1,x1,y1+150, incl, ontop = False)

            #Goggles
            if SPRITES[6]:
                (x3,y3,_,h3) = get_face_boundingbox(shape, 1)
                apply_sprite(image, IMAGES[6][ACTIVE_IMAGES[6]],w,x,y3-10, incl, ontop = False)
```

```
            #Earrings
            (x0,y0,w0,h0) = get_face_boundingbox(shape, 6) #bound box of mouth
            if SPRITES[2]:
```

```

(x3,y3,w3,h3) = get_face_boundingbox(shape, 7) #nose
apply_sprite(image, IMAGES[2][ACTIVE_IMAGES[2]],w3,x3-40,y3+30, incl,ontop=False)
(x3,y3,w3,h3) = get_face_boundingbox(shape, 8) #nose
apply_sprite(image, IMAGES[2][ACTIVE_IMAGES[2]],w3,x3+40,y3+75, incl)

```

```

#         if SPRITES[5]:
#             apply_sprite(image, IMAGES[5][ACTIVE_IMAGES[5]],w,x,y, incl, ontop = True)

#Frocks
if SPRITES[5]:
    (x1,y1,w1,h1) = get_face_boundingbox(shape, 8)
    apply_sprite(image, IMAGES[5][ACTIVE_IMAGES[5]],w1+580,x1-350,y1+80, incl, ontop =
False)

```

```

#Tops
if SPRITES[4]:
    # (x,y,w,h) = (0,0,10,10)
    # apply_sprite2feature(image, IMAGES[7][ACTIVE_IMAGES[7]], fullbody, w//4, 2*h//3, h//2,
True, w//2, x, y, w, h)
    (x1,y1,w1,h1) = get_face_boundingbox(shape, 8)
    apply_sprite(image, IMAGES[4][ACTIVE_IMAGES[4]],w1+350,x1-230,y1+100, incl, ontop =
False)

    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image = Image.fromarray(image)
    image = ImageTk.PhotoImage(image)
    ctr_mid.configure(image=image)
    ctr_mid.image = image

```

```

video_capture.release()

```

```

root = Tk()
root.title('Virtual Dressing room')
app=FullscreenApp(root)

```

```

top_frame = Frame(root, bg='#077bd4', width=50, height=50, pady=3)
center = Frame(root, bg='white', width=50, height=40, padx=3, pady=3)
btm_frame = Frame(root, bg='#077bd4', width=50, height=50, pady=3)

```

```

root.grid_rowconfigure(1, weight=1)
root.grid_columnconfigure(0, weight=1)

```

```

top_frame.grid(row=0, sticky="ew")
center.grid(row=1, sticky="nsew")
btm_frame.grid(row=4, sticky="ew")
logo = ImageTk.PhotoImage(Image.open('logo1.png').resize((120,60)))
model_label = Label(top_frame,image=logo)

```

```

model_label.grid(row=0, columnspan=3)

```

```

center.grid_rowconfigure(0, weight=1)
center.grid_columnconfigure(1, weight=1)

```

```

ctr_left = Label(center,bg='white', width=50, height=190)
ctr_mid = Label(center,bg='white',width=100, height=160, padx=0, pady=0)

```

```

ctr_left.grid(row=0, column=0, sticky="ns")
ctr_mid.grid(row=0, column=1, sticky="nsew")

```

```

scrollable_body = Scrollable(ctr_left, width=15)
SPRITES=[0 for i in range(10)] # [0,0,0,0,0,0,0,0,0,0]
BTNS=[]
IMAGES ={i:[] for i in range(10)} #Images= {0: [], 1: [], 2: [], 3: [], 4: [], 5: [], 6: [], 7: [], 8: [], 9: []}
PHOTOS ={i:[] for i in range(10)} #PHOTOS= {0: [], 1: [], 2: [], 3: [], 4: [], 5: [], 6: [], 7: [], 8: [], 9: []}

```

```

print("Images= ", IMAGES)
print("PHOTOS= ", PHOTOS)
print("sys.argv[1:] =", sys.argv[1:])

```

```

for img in sys.argv[1:]: #it contains all selected images path that are chosen from gui, eg-
['static/images/Necklace1/1.png', 'static/images/Tops4/12.png']
    IMAGES[int(img.rsplit('/',1)[0][-1])].append(img) #it will append the image path to desired categ.
say 1,2,etc.
    image=ImageTk.PhotoImage(Image.open(img).resize((150,100)))
    PHOTOS[int(img.rsplit('/',1)[0][-1])].append(image)
print("Images= ", IMAGES)
print("PHOTOS= ", PHOTOS)
for index in range(9):
    if len(PHOTOS[index]) > 0:
        for k,photo in enumerate(PHOTOS[index]):
            #print("text on btn= ",IMAGES[index][k].rsplit('/',1)) #['static/images/Necklace1', '1.png']
            btn= Button(scrollable_body, highlightbackground='white',
text=IMAGES[index][k].rsplit('/',1)[1].replace('.png','')[:-1],bg='white', image=photo, command =
lambda index=index, k=k: put_sprite(index,k), compound=LEFT, width='300', height='200')
            btn.pack(side="top", fill="both", expand="no", padx="5", pady="5")
            BTNS.append(btn)
scrollable_body.update()

```

```

run_event = threading.Event()
run_event.set()
action = Thread(target=cvloop, args=(run_event,))
action.setDaemon(True)
action.start()

```

```

def terminate():
    global root, run_event, action
    run_event.clear()
    time.sleep(1)
    root.destroy()
root.protocol("WM_DELETE_WINDOW", terminate)
root.mainloop()

```

```

import tkinter as tk
from tkinter import ttk
import sys

class Scrollable(ttk.Frame):
    """
    Make a frame scrollable with scrollbar on the right.
    After adding or removing widgets to the scrollable frame,
    call the update() method to refresh the scrollable area.
    """

```

```

def __init__(self, frame, width=16):

```

```
scrollbar = tk.Scrollbar(frame, width=width)
scrollbar.pack(side=tk.RIGHT, fill=tk.Y, expand=False)
```

```
self.canvas = tk.Canvas(frame, bg='white', yscrollcommand=scrollbar.set)
self.canvas.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)
```

```
scrollbar.config(command=self.canvas.yview)
```

```
self.canvas.bind('<Configure>', self.__fill_canvas)
```

```
# base class initialization
tk.Frame.__init__(self, frame)
```

```
# assign this obj (the inner frame) to the windows item of the canvas
self.windows_item = self.canvas.create_window(
    0, 0, window=self, anchor=tk.NW)
```

```
def __fill_canvas(self, event):
    "Enlarge the windows item to the canvas width"
```

```
canvas_width = event.width
self.canvas.itemconfig(self.windows_item, width=canvas_width)
```

```
def update(self):
    "Update the canvas and the scrollregion"
```

```
self.update_idletasks()
self.canvas.config(scrollregion=self.canvas.bbox(self.windows_item))
```

```
class FullScreenApp(object):
    def __init__(self, master, **kwargs):
        self.master = master
        pad = 3
        self._geom = '200x200+0+0'
        master.geometry("{}x{}+{}+{}".format(master.winfo_screenwidth() -
                                              pad, master.winfo_screenheight() - pad))
        master.bind('<Escape>', self.toggle_geom)
```

```
def toggle_geom(self, event):
    geom = self.master.winfo_geometry()
    print(geom, self._geom)
    self.master.geometry(self._geom)
    self._geom = geom
```

```
# set a callback to handle when the window is closed
self.root.wm_title("PhotoBooth")
self.root.wm_protocol("WM_DELETE_WINDOW", self.onClose)
```

```
def videoLoop(self):
    try:
        while not self.stopEvent.is_set():
            self.frame = self.vs.read()
            self.frame = imutils.resize(self.frame, width=300)
            image = cv2.cvtColor(self.frame, cv2.COLOR_BGR2RGB)
            image = Image.fromarray(image)
```

```

        image = ImageTk.PhotoImage(image)
        if self.panel is None:
            self.panel = tki.Label(image=image)
            self.panel.image = image
            self.panel.pack(side="left", padx=10, pady=10)
        else:
            self.panel.configure(image=image)
            self.panel.image = image
    except RuntimeError as e:
        print("[INFO] caught a RuntimeError")

```

```

def takeSnapshot(self):
    # grab the current timestamp and use it to construct the
    # output path
    ts = datetime.datetime.now()
    filename = "{}.jpg".format(ts.strftime("%Y-%m-%d_%H-%M-%S"))
    p = os.path.sep.join((self.outputPath, filename))

```

```

    # save the file
    cv2.imwrite(p, self.frame.copy())
    print("[INFO] saved {}".format(filename))

```

```

def onClose(self):
    # set the stop event, cleanup the camera, and allow the rest of
    # the quit process to continue
    print("[INFO] closing...")
    self.stopEvent.set()
    self.vs.stop()
    self.root.quit()

```

```

import cv2
import dlib
from imutils import face_utils, rotate_bound
import sys
#from tryOn import calculate_inclination

class VideoCamera(object):
    def __init__(self):
        # Using OpenCV to capture from device 0. If you have trouble capturing
        # from a webcam, comment the line below out and use a video file
        # instead.
        self.video = cv2.VideoCapture(0)
        # If you decide to use video.mp4, you must have this file in the folder
        # as the main.py.
        # self.video = cv2.VideoCapture('video.mp4')

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

    def get_frame(self):
        success, image = self.video.read()
        # We are using Motion JPEG, but OpenCV defaults to capture raw images,
        # so we must encode it into JPEG in order to correctly display the
        # video stream.

```

```

    # detector = dlib.get_frontal_face_detector()
    # model = "data/shape_predictor_68_face_landmarks.dat"
    # predictor = dlib.shape_predictor(model)

```

```
# gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# faces = detector(gray, 0)
```

```
# for face in faces:
#     (x,y,w,h) = (face.left(), face.top(), face.width(), face.height())
```

```
#     shape = predictor(gray, face)
#     shape = face_utils.shape_to_np(shape)
#     incl = calculate_inclination(shape[17], shape[26]) #inclination based on eyebrows
```

```
#     # condition to see if mouth is open
#     is_mouth_open = (shape[66][1] - shape[62][1]) >= 10 #y coordiantes of landmark points of lips
```

```
    ret, jpeg = cv2.imencode('.jpg', image)
    return jpeg.tobytes()
.countdownHolder{
    font: 23px/0.8 'Roboto Condensed', sans-serif;
    padding-top:0.5em;
    height: 89px;
}
.position{
    display: inline-block;
    height: 1em;
    overflow: hidden;
    position: relative;
    width: 20px;
    font-size: 0.5em;
    margin-top: 25px;
}
.digit{
    color: #fff;
    letter-spacing: -6px;
    left: 3px;
    bottom: 0;
    position: absolute;
}
.countDays{ /* display:none !important;*/ }
.countDiv0{ /* display:none !important;*/ }
.countHours{}
.countDiv1{}
.countMinutes{}
.countDiv2{}
.countSeconds{}
.countDiv{
    display: inline-block;
    width: 16px;
    height: .8em;
    position: relative;
}
.countDiv:after{
    top: 0.9em;
}
/*--responsive design--*/
@media (max-width: 1366px){
}
@media (max-width: 1080px){
    .countDays, .countHours, .countMinutes, .countSeconds {
```

```

        width: 80px;
        height: 80px;
    }
    .position {
        margin-top: 17px;
    }
    .agileinfo_banner_bottom1_grid_right {
        padding: 6.5em 0 0;
    }
}
@media (max-width:1024px){
    .position {
        width: 19px;
    }
    .boxName {
        font-size: 13px;
    }
    .countdownHolder {
        padding-top:0.5em;
    }
}
@media (max-width:800px){
    .position {
        margin-top: 19px;
    }
}
@media (max-width: 768px){
}
@media (max-width:640px){
    .position {
        width: 16px;
        font-size: 25px;
    }
}
@media (max-width:480px){
    .countDays{ /* display:none !important;*/ }
    .countDiv0{ /* display:none !important;*/ }
    .countHours{}
    .countDiv1{}
    .countMinutes{}
    .countDiv2{}
    .countSeconds{}
    .countDiv{
        width:5px !important;
    }
    .countdownHolder {
        padding-top:0.2em;
    }
}
@media (max-width: 375px){
    .position {
        font-size: 20px;
    }
    .countDays, .countHours, .countMinutes, .countSeconds {
        width: 70px;
        height: 70px;
    }
    .boxName {
        font-size: 11px;
    }
}

```

SCREEN SHOTS



VIRTUAL TRIAL ROOM FOR ONLINE SHOPPING

Feel Urself

GOGGLES **NECKLACES** EARRINGS TIARAS TOPS FROCKS

GOGGLES **NECKLACES** EARRINGS TIARAS TOPS FROCKS



Pendant

Rs.320 **Rs.200**

Add to cart

TRY IT NOW!



Silver Necklace

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



Silver Necklace

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



TRY IT NOW!



Silver Necklace

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!

TRY IT NOW!



Thread necklace

Rs.320 **Rs.150**

Add to cart

TRY IT NOW!

TRY IT NOW!

Try & Checkout

↑

GOGGLES

NECKLACES

EARRINGS

TIARAS

TOPS

FROCKS



Goggles

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



Sun glasses

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



Sun glasses

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!





Studs

Rs.320 **Rs.100**

Add to cart

TRY IT NOW!



Goldwork earrings

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



Goldwork earrings

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



TRY IT NOW!



TRY IT NOW!



TRY IT NOW!



Goldwork earrings

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



Goldwork earrings

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



Hangings

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!

Try & Checkout



Single Maang Tika

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



Tiara

Rs.320 **Rs.125**

Add to cart

TRY IT NOW!



Tiara

Rs.320 **Rs.125**

Add to cart

TRY IT NOW!



Add to cart

TRY IT NOW!



Floral

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!

FROCKS



Top 3

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!





Top4

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



Top5

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



top 6

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!

Try & Checkout



GOGGLES NECKLACES EARRINGS TIARAS TOPS **FROCKS**



Frock

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



Customized top

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



Customized top

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



TRY IT NOW!

TRY IT NOW!

TRY IT NOW!



Customized top
Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



Light pink Top
Rs.320 **Rs.250**

Add to cart

TRY IT NOW!

Try & Checkout



Try it ON

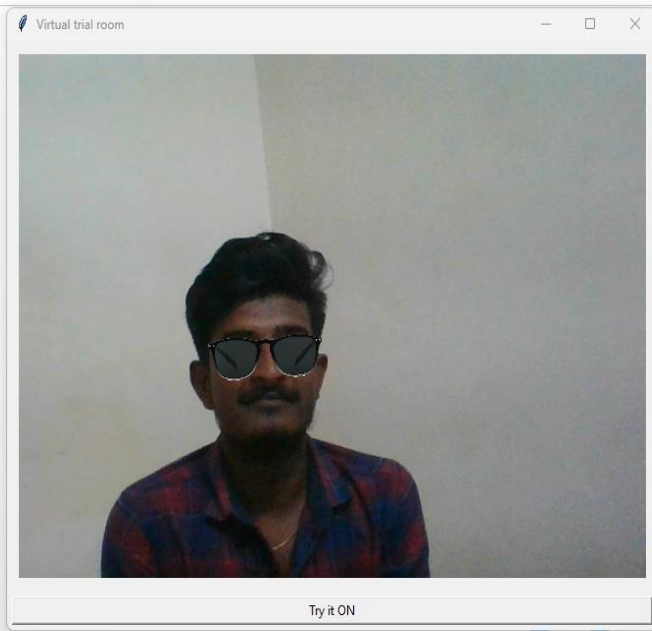


Top 3
Rs.320 **Rs.250**

Add to cart

TRY IT NOW!





Sun glasses

Rs.320 **Rs.250**

Add to cart

TRY IT NOW!



RESEARCH PAPER

COMPLEX ARGMENTED REALITY DRIVEN ONLINE SHOPPING SYSTEM

Rayala Prabhas
Computer Science &
Engineering
Sathyabama Institute of Science
&
Technology
Sathyabama University,
Chennai, India
rayalaprabhas1121@gmail.com

Rathana Narasimhudu
Computer Science &
Engineering
Sathyabama Institute of
Science &
Technology
Sathyabama University,
Chennai, India
rknarasimha822@gmail.com

Dr.J,REFONAA,M.E.,Ph.D
Computer Science &
Engineering
Sathyabama Institute of
Science &
Technology
Sathyabama University,
Chennai, India
refonna.cse@sathyabama.ac.in

Abstract- In this modern world, shopping in offline is a tedious process where much time has been consumed. Online shopping has a huge market place where it also faces large amount of stumbling block like return of goods due to size, colour etc. In order to make this shopping process more convenient to user, Augmented Reality(AR) based shopping system has been designed which allow the end customer to view the item through a computer generated view with a device like mobile, tablet, desktop etc. The proposed augmented reality system has been designed for e-commerce and media pipe frame has been involved in it. The Media-pipe Holistic pipeline integrates individual posture, face, and hand models that are each tailored for a particular field.. AR enriches the actual environment by adding computer-generated objects that spontaneously emerge there. The development of an AR system combines the brain's responses to 2D interfaces by working with the visual, auditory, and kin esthetic systems.

Keywords: Media pipe, Augmented reality, Holistic model

1. INTRODUCTION

A number of e-Commerce websites and physical apparel stores will soon implement a Next-Gen technology named "Augmented Reality based online purchasing platform." We are utilizing the idea of augmentation to present individuals with a tool that will allow them to examine a real-time simulation of several costumes without having to actually wear them. This web-based program is accessible from anywhere in the world with just a web browser and an internet connection. Since it is a platform-independent programmer, it may run on any OS and machine, including mobile devices, desktop PCs, etc.

Various hardware technologies utilised in augmented reality are optical projection system, augmented view through eyeglass, head-up display, contact lens that show AR, virtual retinal display, eye touch, and handheld AR.. The different software used for augmented reality are Vuforia augmented reality, AR Kit, AR Core, AR Tool Kit, Mind AR. Augmented reality has been used in different fields like Archaeology, Architecture, Education, Industrial manufacturing, Commerce, Visual arts, Social interaction, Video game, Flight training, Military, Navigation.

The phrase "augmented reality" is becoming more and more well-known in the marketing and retail industries. Fascinatingly, fashion is a fantastic fit for augmented reality since it is experience-based. Future trends will have a greater impact on the virtual world than is currently apparent. By the end of 2020, augmented reality and virtual reality products will bring in 120 billion, according to Digi Capital. However, Greg Jones, Google's head of AR and VR, believes that augmented reality will play a transformation role in society. He also made an additional comment that the situation would be the same if it applied to the retail fashion business.

It is a well-known reality that businesses have a lot of issues when it comes to customer involvement and awareness. These issues may be addressed comprehensively by utilising virtual and augmented reality. In the realm of fashion, augmented reality or AR, is advancing to provide a crucial sensory link to support both in-person and online buying. Virtual reality, or VR, has an immersive science-fiction feel to it, which is why several merchants have employed it in the past. For instance, Lowe's hologram enables buyers to put on a headset and experience kitchen and bathroom remodeling ideas. However, it is crucial to highlight that this type of cutting-edge technology is only accessible in a few countries.

II. LITERATURE SURVEY

Many studies in the area of AR have been conducted, and they are listed in the section below. There is discussion of many augmented reality approaches, including 3D modelling, position estimation, facial land detection, and media-pipe face mesh.

A dynamic visual dressing room based augmented reality was developed by N.Z.Hashmi[1] et. al(2020). A cam is used to acquire the input video or image where user can experience dress fitting. The model detect face, body and skin detection. As a result, end user can wear the different dresses through a virtual reality.

A augmented reality based live purchase of diamond was demonstrated by O.A.Egaji[2] et. al(2019). End user can wear the diamond jewellery by augmented reality and check the design suits. As a result, user can view the model that suits them.

A 3D application model was demonstrated by Takala[3] et al.(2013). The input device used are head tracking, stereoscopic visual. The 2D images are converted to 3D model.

Y.A.Sekhavat[4] et. al has developed a privacy based dress try using augmented reality. The proposed approach allow the customer to wear the dress in AR and view in a three dimensional view. As a result the customer can view the customized dress that suits them.

A.Tope et al.[5] has proposed by Augmented relity based beauty make over system. Two approach are used. Shape predictor approach a facial lank mark system capture lips, eyes, ears etc as a picture and Mediapipe face mesh approach capture the entire face & body. As a result, user can try out different clothes, beauty products, jewels etc.

A web based prototype for online shopping system was proposed by B.Xu[6] et. al(2022). A data fusion algorithm was used to give information about the product. A case study interview was conducted with 14 participant about the system. As a result a visual experience has been given to used. However the system was not tested in real time world.

B.E. Singgih[7] et al.(2021) has proposed a web based application for e-commerce which gives a unique shopping experience to user. A record video is used as input. Active X plugin for web pages was used in the system's development. Finally, user can shop different items like shoe, jewels, clothing, furniture.

INFERENCE FROM LITERACY SURVEY

TITLE 1: AR JUST AUGMENTS THE REAL WORLD, WHILE VR IS COMPLETELY IMMERSIVE

AUTHOR: MEL SLATER , MARIA SANCHEZ-VIVES

Mel Slater as well as others Augmented reality does nothing more than improve the world as it is now and as you will, in some ways, continue to see it. On the other hand, virtual reality puts you into a world that is wholly made of innovation, like Meta's Oculus Quest. You are typically blind and not meant to interact with the real world.

TITLE 2: WHEN COMPARED TO VIRTUAL REALITY, ENHANCED REALITY IS MORE PORTABLE.

AUTHOR: JIANGHA O XIANG 1, EN-LIN HSING 1, ZIQIAN HE 1, TAO ZHAN1 AND SHIN-TSON WU 1

You will need to put on a headgear and, maybe, hold controllers in your hands when using virtual reality. Additionally, you must ensure that you and your immediate surroundings are both safe. (Avoid being the jerk who strikes their TV while virtual reality boxing!) Contrarily, AR usually just requires a smart phone or, in some circumstances, AR glasses. You won't have to leave the room to enjoy the event since you'll be completely aware of your surroundings.

TITLE 3: WHAT DIFFERENCES EXIST WHILE CREATING FOR AUGMENTED REALITY?

AUTHOR: PHIL COHN

In many respects, design used to be lot easier. When you developed anything for print, you were aware that the design would always appear in the same way. Then computers and the internet arrived, posing a plethora of brand-new difficulties. If designers didn't start considering interaction much more, they would be left behind in the rush of Word Art and tiled psychedelic backdrops. Users required their hand held to assist them traverse this new digital landscape because of all the new technology. A new method and language for communicating with people through design were developed. Since then, new technologies have emerged that need the updating of this language, such as smart phones.

III. EXISTING SYSTEM

Augmented reality (AR) occurs when a real-world item is "augmented" with computer-generated perceptual information that may span many sensory modalities, including visual, auditory, haptic, somatosensory, and olfactory.

The overlay sensory information may be useful or destructive. In this way, augmented reality affects a person's ongoing impression of a real-world environment, whereas virtual reality completely replaces the user's real-world setting with a simulated one.

IV. PROPOSED SYSYTEM

As we all know buying a product online on basis of images available or displayed is quite ambiguous which creates confusion. As the trying option is not available on the website and we are quite unclear with our selection and not at all satisfied. To remove this dilemma, we proposed a solution to create a website with AR-based technology that helps users to try products virtually with the help of the AR option from anywhere and anytime themselves.

We detected bodies using a media stream. A redesigned pipeline with improved posture, face, and hand features makes up Media Pipe Holistic. With added support for the interchangeability of the three parts based on the quality of transactions, this individually tracks in real-time with the least memory transfer between their inference backends. The Media Pipe Holistic pipeline employs many models for the posture, face, and hand components, each of which is tuned for its own domain. Due to their diverse areas of competence, the responses to one module are incompatible with those to the other. Using a posture detector and the resulting key point model, Media Pipe Holistic makes an approximation of the human pose. Then, it constructs three zones of interest using the main points of the supposed stance (ROI).

V. SYSTEM ARCHITECTURE



Fig-1: Architecture

Client-side: The client-side consists of the user's device, which could be a web browser or a mobile app. The client-side application is responsible for rendering the AR experience and communicating with the server-side.

AR SDK: The AR SDK is a software development kit that provides tools for creating AR experiences. The SDK provides access to the camera, sensors, and other hardware on the user's device, as well as APIs for rendering 3D models and overlaying them onto the real world.

The online store's management, including product listings, order processing, and payment handling, is within the purview of the e-commerce platform. The platform should be able to integrate with the AR SDK to provide a seamless AR shopping experience for customers.

The content management system (CMS) is responsible for managing the product content, including images, descriptions, and other metadata. The CMS should be able to integrate with the AR SDK to provide 3D models and other AR content for products.

Cloud services: Cloud services are used to store and process large amounts of data, such as product images and 3D models. The cloud services can also be used for image recognition and other AI/ML tasks to improve the accuracy of the AR experience.

API gateway: The API gateway is responsible for managing the communication between the client-side application and the server-side. It provides authentication, rate limiting, and other security features.

Database: The database stores the e-commerce platform data, such as user information, product listings, and order details.

Payment gateway: The payment gateway is responsible for processing payments and integrating with the e-commerce platform to update order status and inventory.

VI. RESULT & CONCLUSION

In our proposed system architecture, we have outlined a high-level view of an e-commerce website using AR. The system includes a client-side application, an AR SDK, an e-commerce platform, a content management system, cloud services, an API gateway, a database, and a payment gateway.

1

The client-side application is responsible for rendering the AR experience and communicating with the server-side, which includes the e-commerce platform, content management system, cloud services, and database. The API gateway provides authentication, rate limiting, and other security features, while the payment gateway handles payments and order processing.

The use of AR in e-commerce websites has been shown to improve customer engagement and sales. Customers are more likely to make a purchase when they can see how a product looks on them in real-time. AR can also help reduce the number of returns, as customers are more confident in their purchases when they can see how they look before buying.

In conclusion, the use of AR in e-commerce websites can provide a competitive advantage to online stores by improving the shopping experience for customers. The proposed system architecture can serve as a starting point for building an AR-powered e-commerce website.

It is crucial to remember that the individual needs of the online business may affect the implementation specifics.

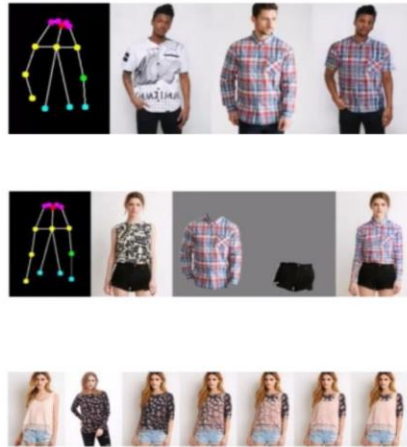


Fig-2: Recommended Outfits

VII. REFERENCES

1. N. Z. Hashmi, A. Irtaza, W. Ahmed, and N. Nida, "An augmented reality based virtual dressing room using haarcascades classifier," in 2020 14th International Conference on Open Source Systems and Technologies (ICOSST), pp. 1–6, 2020.
2. O. A. Egaji, I. Asghar, W. Warren, M. Griffiths, and S. Evans, "An augmented reality application for personalised diamond shopping," in 2019 25th International Conference on Automation and Computing (ICAC), pp. 1–7, 2019.
3. T. M. Takala, M. Mak'ar'ainen and P. Hämäläinen, "Immersive 3D modeling with Blender and off-the-shelf hardware," 2013 IEEE Symposium on 3D User Interfaces.
4. Y. A. Sekhavat, "Privacy preserving cloth try-on using mobile augmented reality," IEEE Transactions on Multimedia, vol. 19, no. 5, pp. 1041–1049, 2017.
5. A. Tope, O. Bhoje, T. Rane and P. Mahindrakar, "An Augmented Reality-based System for Fashion and Beauty Makeover," 2022 3rd International Informatics and Software Engineering Conference (IISEC), Ankara, Turkey, 2022, pp. 1–5, doi: 10.1109/IISEC56263.2022.9998275.
6. B. Xu, S. Guo, E. Koh, J. Hoffswell, R. Rossi and F. Du, "ARShopping: In-Store Shopping Decision Support Through Augmented Reality and Immersive Visualization," 2022 IEEE Visualization and Visual Analytics (VIS), Oklahoma City, OK, USA, 2022, pp. 120–124, doi: 10.1109/VIS54862.2022.00033.
7. B. E. Singgih, L. Yudistira, H. Wijaya and M. S. Anggreainy, "Implementation of Augmented Reality in E-Commerce Applications," 2021 2nd International Conference on Innovative and Creative Information Technology (ICITech), Salatiga, Indonesia, 2021, pp. 136–139, doi: 10.1109/ICITech50181.2021.9590104.
8. Ohri, Ashish, Shashank Agrawal, and Garima S. Chaudhary, "On-device Realtime Pose Estimation & Correction. IJAEM (2021).

9. Seunghwan Choi, Sunghyun Park, Minsoo Lee, Jaegul Choo; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 14131- 14140.

10. Rosni, Nurul Shuhadah, et al. "Development of mobile markerless augmented reality for cardiovascular system in anatomy and physiology courses in physiotherapy education." 2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM). IEEE, 2020.

11.. Koca, Buse Asena, Burakhan Çubukçu, and Uğur Yüzgeç. "Augmented reality application for preschool children with unity 3D platform." 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE, 2019.

RE-2022-202533-plag-report

ORIGINALITY REPORT

6 %	3 %	4 %	4 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Southern New Hampshire University - Continuing Education Student Paper	1 %
2	wikimili.com Internet Source	1 %
3	Submitted to SSN COLLEGE OF ENGINEERING, Kalavakkam Student Paper	1 %
4	www.ijser.org Internet Source	1 %
5	www.indiamart.com Internet Source	1 %
6	S Anush Kumar, T Amudhavan, S Arun Kumar, S Sasikala. "Improving Effectiveness of STEM Education in Covid-19 Pandemic – A Pilot Study", 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 2021 Publication	<1 %

7 Arnav Tope, Omkar Bhope, Tanishk Rane, Parimal Mahindrakar. "Proposing an augmented reality-based system for fashion and beauty makeover", 2022 3rd International Informatics and Software Engineering Conference (IISEC), 2022
Publication

8 www.annauniv.edu
Internet Source

Exclude quotes On

Exclude matches Off

Exclude bibliography On