University of Mumbai

WeAR - Wear clothes in Augmented Reality

Submitted in partial fulfillment of requirements

For the degree of

Bachelors in Technology

by

Ojas Kapre
Roll No: 1611022
Harsh Patel
Roll No: 1611032
Murtaza Patrawala
Roll No: 1611034
Tanay Raul
Roll No: 1611037

Guide

Prof. Suchita Patil

Department of Computer Engineering
K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch 2016 -2020

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the dissertation report entitled **WeAR - wear clothes in Augmented Reality** submitted by **Ojas Kapre (Roll no 1611022), Harsh Patel (Roll no 1611032), Murtaza Patrawala (Roll no 1611034), Tanay Raul (Roll no 1611037)** at the end of semester VIII of LY B. Tech is a bona fide record for partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Engineering of the University of Mumbai

Guide

Head of the Department

Principal

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this dissertation report entitled **WeAR - wear clothes in Augmented Reality** is bona fide record of project work done by **Ojas Kapre (Roll no 1611022), Harsh Patel (Roll no 1611032), Murtaza Patrawala (Roll no 1611034), Tanay Raul (Roll no 1611037)** during semester VIII.

This project work is submitted at the end of semester VIII in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Engineering of the University of Mumbai.

____

Internal Examiner

____

External Examiner

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

DECLARATION

We declare that this written thesis submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Ojas **Signature of the Student** 1611022 **Roll No.**	Harsh **Signature of the Student** 1611032 **Roll No.**
Murtaza **Signature of the Student** 1611034 **Roll No.**	TANAY **Signature of the Student** 1611037 **Roll No.**

Date:

Place: Mumbai-77

Dedicated to

My family and friends

Key words**:** Virtual Trial Room, AR, Cloth Trail Room

Contents

List of Figures

List of Tables

Chapter 1
Introduction

This chapter presents the overview of the selected project topic. It establishes the scope, context and significance of project being conducted. Various functionalities and features of the product along with the requirements are presented in this chapter

1.1 Problem definition

The problem is to create a mobile application based on AR that can be used by customers to visualize clothes on themselves. This application can be integrated with online shopping websites as an aid to customers who find it difficult to choose the right pattern and size of clothes. All the existing solutions use special hardware for depth sensing and user size and pose estimation. This type of solution cannot be used in an online shopping website as the user may not have the required hardware available with them. The application should be designed in such a way that it needs no external hardware and should be able to run on the user's smartphone with a camera.

1.2 Scope

The project is targeted towards customers who buy clothes online but hesitate to do so in some scenarios where they have some doubts regarding the proper fit of the clothes or how the clothes would actually look on them. In conventional shopping, this problem is eliminated through trial rooms where users can physically try the clothes. Scope of this project is to eliminate these difficulties in the online market by providing the following functionalities to its users..

- A User should be able to visualize the clothes on themselves in real time.
- The app should work on smartphones with minimal hardware and software requirements.
- A User should be able to choose which clothes they wish to try on.
- A User should be able to map a pattern on an existing cloth model.

1.3 Hardware Software Requirements

Hardware Requirements

Processor: ARMv7 CPU with NEON support or Atom CPU

Recommended to have an octa-core CPU which is now quite common in today's smartphone.

Ram :- Minimum 2GB RAM on an Android Smartphone

Software Requirements

Android: API 24+

ARCore Support for smartphone

Chapter 2 Literature Survey

This chapter presents comprehensive overview of all the research papers available till date on this topic. The summary of each of the paper is mentioned in this chapter. Limitations of each methodology is also stated in this chapter. We have identified the where gaps exist in research conducted till date

The idea of virtual trial rooms is not new and many attempts have been made in the past to address the issues that customers face while shopping for clothes. However, with new innovations in the field of Augmented Reality and Machine Learning, the applications became better and more realistic. From simply projecting the static 2D image of clothes, to rendering the 3D cloth model on the user's body in a real time environment, many prototypes have been developed.

Very primitive attempts tried to render a 2D image of cloth on screen. However this was not real time, i.e the rendered cloth image was static and the user had to align himself to the cloth image in order to gain a visual experience of the garment. The attempt was not beneficial as it focuses on the alignment of the user on garment rather than the other way around.

Recent attempts have used the knowledge of Machine Learning for Human Pose Estimation through Convolutional Neural Networks (CNN) so that the previous problem of human aligning to the image is eliminated. The cloth image is automatically mapped to the customer's body based on the estimation of human pose. **'VITON: An Image-based Virtual Try-on Network'** [1], a technical paper published by Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, Larry S. Davis from University of Maryland, College Park in June 2018 is one such attempt which uses state-of-the-art method to transfer any 2d cloth image on a human body with the help of advanced deep learning models namely Encoders-Decoders. The input (i.e Person pose image and 2d cloth model) was passed through an Multi-task Encoder-Decoder model which identified the clothed region and generated a cloth mask.

Working of VITON is as follows:-

1. The input (i.e Person pose image and 2d cloth model) is passed through an Multi-task Encoder-Decoder model which identifies the clothed region and generates a cloth mask. (A cloth mask is basically the shape of the cloth that the user is wearing).
2. Here we take the 2d cloth model and the cloth mask generated in the previous step and pass it through refinement which wraps the 2d cloth model to the shape and texture of the cloth mask. This new 2d cloth model is now simply superimposed on the original human pose to generate the result.

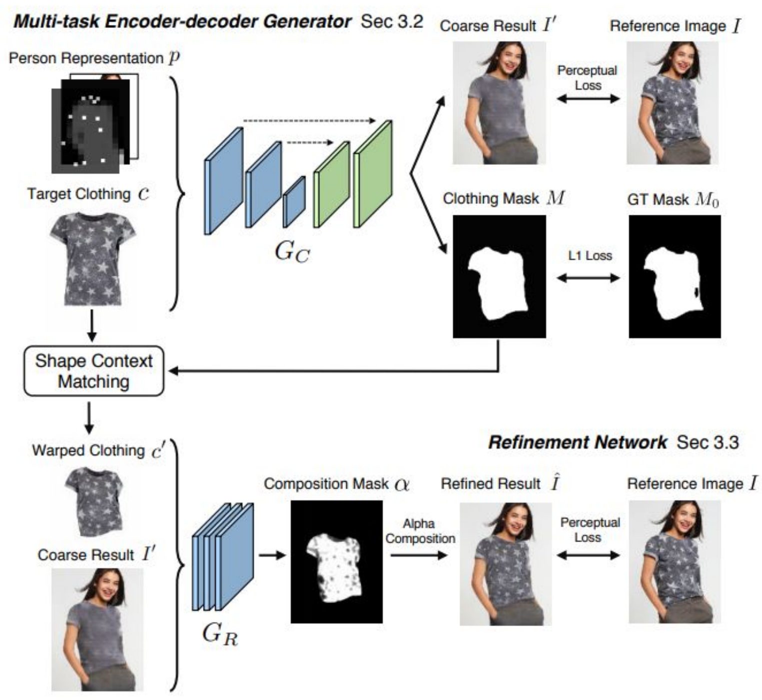


Fig 2.1 - Working of VITON

This approach is time consuming (The mapping is not instant) and thus does not address the issue of mapping an image in real time. Some alternate approaches have addressed this issue by using specialized hardware instead of Machine Learning algorithms for human pose estimation. One such approach is provided in **'Virtual Trial Room'**, a technical paper published by Akshay Shirsat, Samruddhi Sonimindia, Sushmita Patil, Nikita Kotecha and Prof Shweta Koparde, which was published in International Journal of Research in Advent Technology, Vol.7, No.5, May 2019 [2]. This approach used Microsoft's kinect camera for pose detection, opencv for image processing, to map a 3d cloth model on the user in real time.

Kinect provides 25 joints tracked at 30 frames per second. Out of the 25 joints, they used certain joints to calculate the measurements of shirt required to be augmented on the virtual body. To calculate the width of the shirt around shoulders, they used the shoulder left joint coordinates and shoulder right joint coordinates, and their difference gave the width of shirt around the shoulder. To calculate the waist size, the hip left joint coordinates and the hip right joint were used. To calculate rotation, the used angle of line formed by shoulder center joint and hip center joint before and after rotation, their difference gave angle of rotation After this they mapped the cloth models created in Blender by rotation, translation and scaling of the model according to the pose of the user. (Each joint point was transformed from the 3d coordinates on the 3d cloth model to the coordinates of the user using specific rotational, translational, and scaling values)

Scaling:

Position of the left-shoulder is calculated. This real world position is converted to projective.The same is repeated for right-shoulder To calculate x-coordinate, distance between left and right shoulder position is calculated.

- x-coordinate=RightShoulderPos.xLeftShoulderPos.x.
- Neck position of skeleton=skel-neck.
- mid.y=((LeftHip.y+RightHip.y)/2)
- y-coordinate= sqrt(sq(skel-neck-mid.y))
- z-coordinate is manually entered.

Translation:

- The centre point of 3d model= jointPos
- This real world position is converted to projective.
- translate(jointPos.x,jointPos.y,60)
- The model gets translated

Rotation:

- To calculate rotation around x axis

- phi=atan(orientation.m12/orientation.m22)
- To calculate rotation around y axis
- theta=-asin(orientation.m02)
- To calculate rotation around z axis
- psi=atan(orientation.m01/orientation.m00)

Fig 2.2 - Output of Virtual Trial Room

Another similar approach was used by Amoli Vani and Dhwani Mehta, students of K.J. Somaiya college of engineering, under the guidance of prof. Suchita Patil in their technical paper **'Virtual Changing Room'** [3]. The approach used in this paper is using kinect sensor for depth and motion detection for human pose tracking, which returns all the parameters for pose detection along with key points marking. The previous approaches used 2D garment images. In this approach, 3D models of garments have been prepared using an open source software named Blender. Blender enables the system to give the stretch, dampness and thickness of the material. It gives realistic experience of creases and folding according to the user's body motions. It also dulls the garments already worn by the user and only highlights the cloth that the user is trying. It provides realistic experience, customer satisfaction of the product and avoids tiresome lines, thus saving customer time.Blender is used to model 3d models of clothes, and Unity is used to map the cloth model on the user's body which was detected by kinect camera. The measurement of the user's chest, height, etc is taken from the user and is then used to estimate the fit of the cloth model in real time.

Size Estimation:

The dress size, small, medium or large is determined using the estimated value of the chest and the height of the user. The measurements of the chest and the height of the user is taken as input as shown. Let the vertical axis be the height and let the horizontal axis be the girth of the chest. On a graph, a S-M-L area is created and their central point is computed.

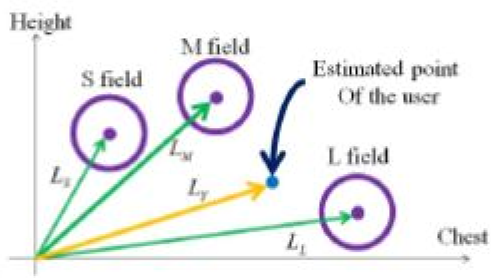


Fig 2.3 - Size estimation graph

The drawback in these approaches is the use of an external specialized hardware, which practically makes it very difficult to integrate in the online shopping domain. One way to eliminate this hardware is to use better and efficient Machine Learning algorithms that can estimate human pose in real time. **Pose Estimation for mobile (Implementation of CPM and Hourglass model using TensorFlow and MobileNetV2)** [6] is an implementation of CPM(Convolutional pose machines) and Stacked Hourglass Network using MobileNetV2 architecture [4] which provides better and faster implementation of high end machine learning models for mobile applications.

Following are the output metrics from this implementation

Model	FLOPs	PCKh	Inference Time
CPM	0.5G	93.78	~60 FPS on Snapdragon 845

Hourglass	0.5G	91.81	~60 FPS on iPhone XS (need more test)

Table 2.1 - Output metrics of pose estimation using Mobilenet v2

A recent attempt at size estimation in 2D images is described in '**Estimating the Object Size from Static 2D Image'** [5]. Here, a solution for estimating the object size from static images using one camera was introduced and related software implementation is comprehensively described. The solution itself is simple, fast and yet easy to implement. The detection of an object in a picture is based on the processing of two images taken in the same environment with and without the object, following the given requirements. Subtraction using RGB color model is used as the first part of the process. Every pixel of the second image is subtracted from a pixel with the same position on the first image. This subtraction removes the background (see Fig. 2.3), as it was identical in both images.By using thresholding image processing technique image is binarized. By using Canny edge detection algorithm, edges are detected.

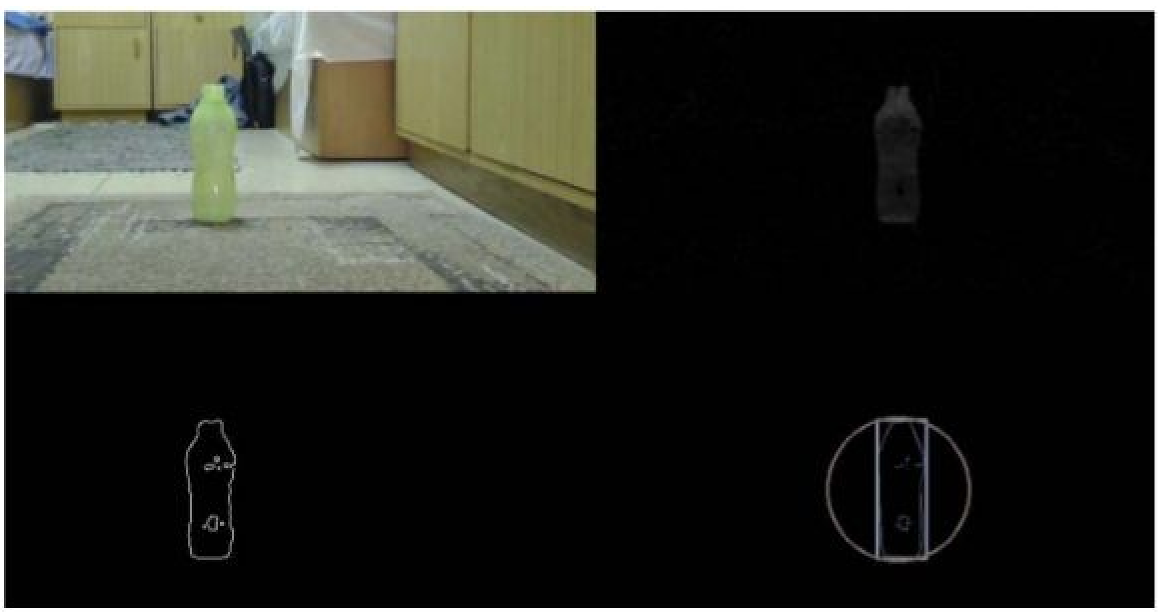


Fig 2.4 - Image processing: original image (the upper-left corner), removed background (the upper-right corner), Canny filter (the bottom-left corner), bounding box of the object (the bottom-right corner)

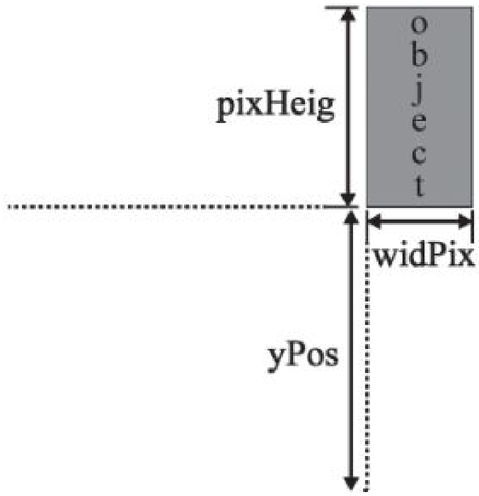


Fig 2.5 - Position of the object in the display window

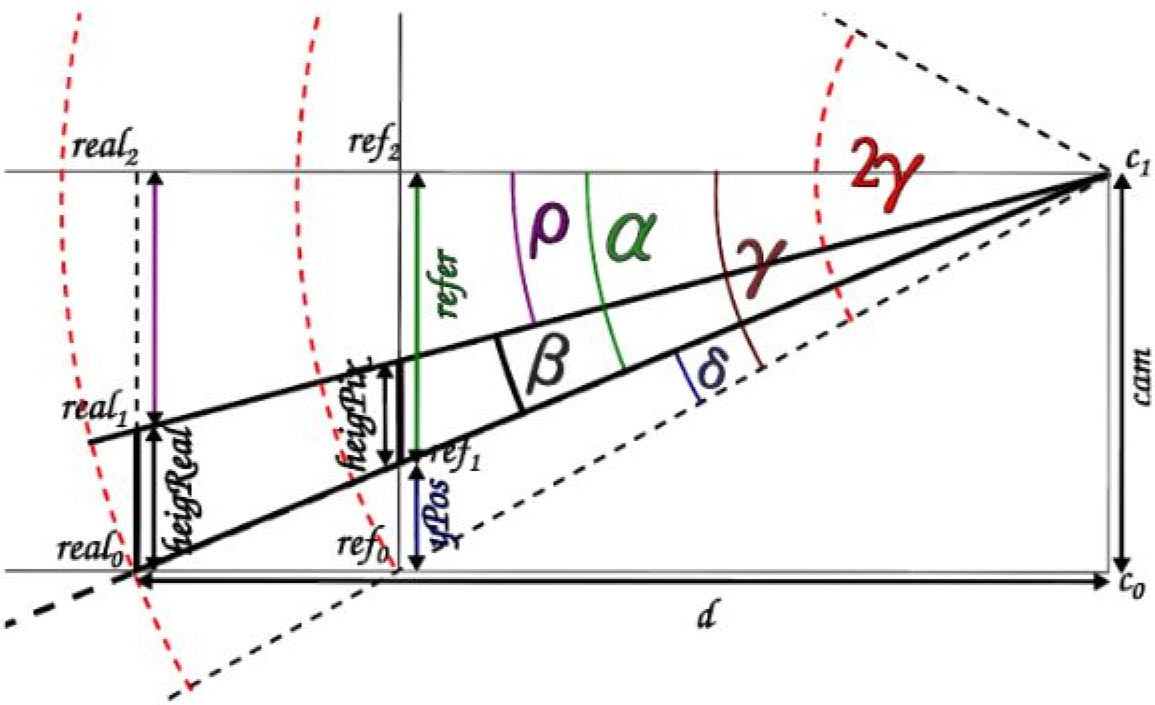


Fig 2.6- Important angles and distances

γ - half of vertical view angle

heighReal - real height of the object

d - The physical horizontal distance between the measured object and the camera

cam - height from which the camera shoots the scene

Estimating object width:

$$widPix = \alpha * yPos + \beta$$

$$\alpha = m_1 * \frac{cam}{wid\,\mathrm{Re}al} - b_1$$

$$\beta = m_2 * \frac{cam}{wid\,\mathrm{Re}al} - b_2$$

$$estWidth = cam * \frac{widPix + b_1 * yPos + b_2}{m_1 * yPos + m_2}$$

and m1 = -1.028, m2 = 0.06892, b2 = 203.7 are constants obtained from the analyzed data

Estimating object height:

$$referH = m_1 * yPos + m_2 * cam + b_2$$

$$estHeig = cam + k * \tan(\rho) * d$$

$$\rho = \frac{|pixHeig - referH|}{referH} * \alpha$$

$$\alpha = \gamma - \delta$$

$$\delta = \frac{yPos}{yPos + referHeig} * \gamma$$

The equations were created according to the basic trigonometric rules and sets of tests are described in this paper. Based on these tests, while using various objects, different heights and different distances of objects from the camera, it was found that the deviation of the measurement is smaller than 10%.

Chapter 3 Project Design

This chapter presents proposed system model of the product to be implemented. After the system design, the software management plan and the software requirements document is presented in this chapter. Use cases, activities to be performed, classes of the system and sequence of the events, etc are supported using the UML diagrams

Proposed System model

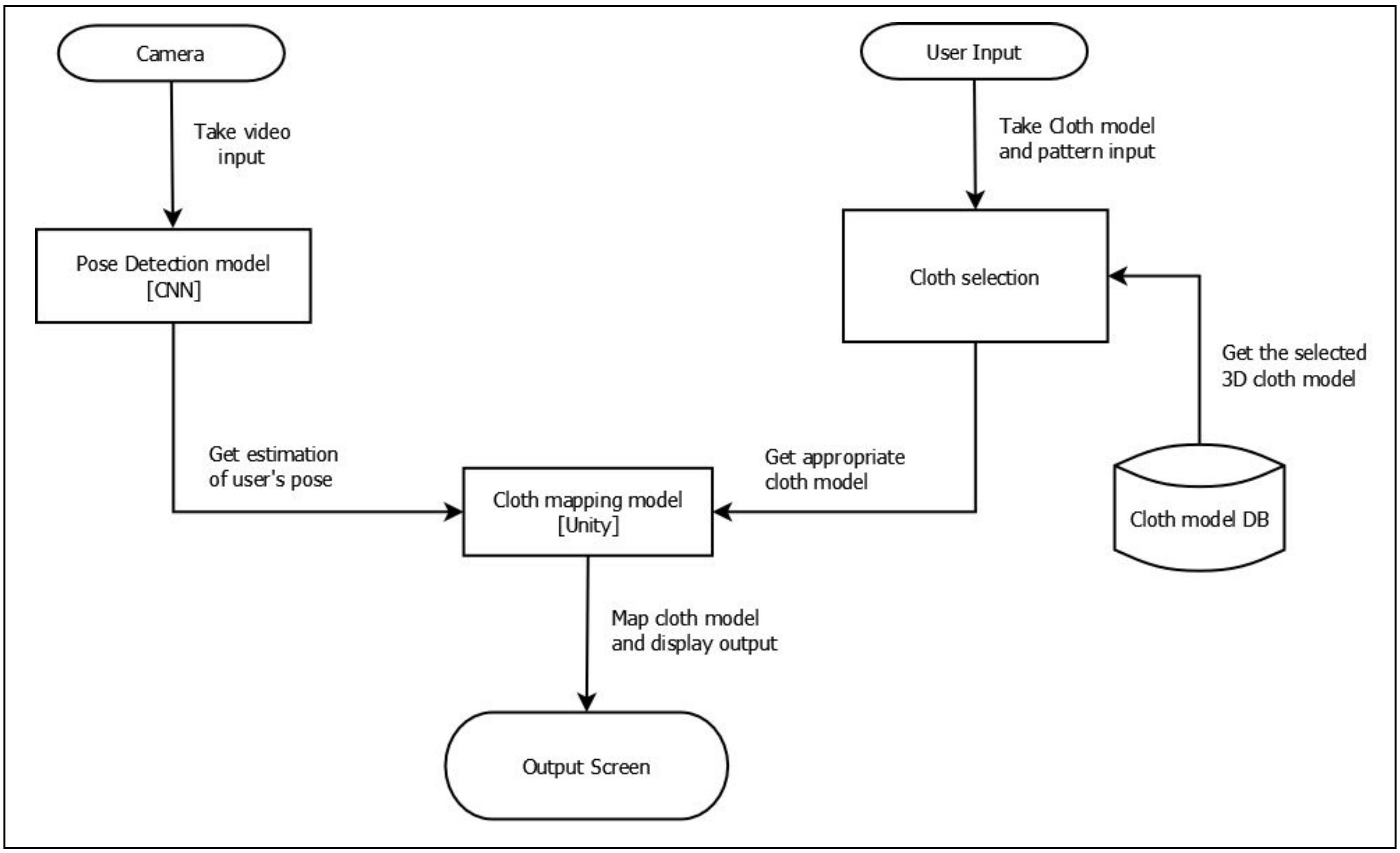


Fig 3.1 - Proposed System Model

Software Project Management Plan

Feasibility analysis

The technology and tools used in this project are Blender, Unity and Android which are open source tools. Each of these have good online documentation which results in faster solving of technical problems and so technical skills required to use these tools are moderate. As a result of this project is technically as well as legally feasible. Processing real time videos will incur some cost and there will also be a considerable hosting cost. But the development cost will be minimum because of the use of open source and free tools. This project is also economically feasible.

Lifecycle model

Because of a smaller development team of four, Agile model is selected for the software development lifecycle. Agile models will also allow pair programming. It is the combination of incremental and iterative model where each iteration consists of the following steps -

1. Planning
2. Requirement analysis
3. Design
4. Development
5. Unit Testing
6. Deployment

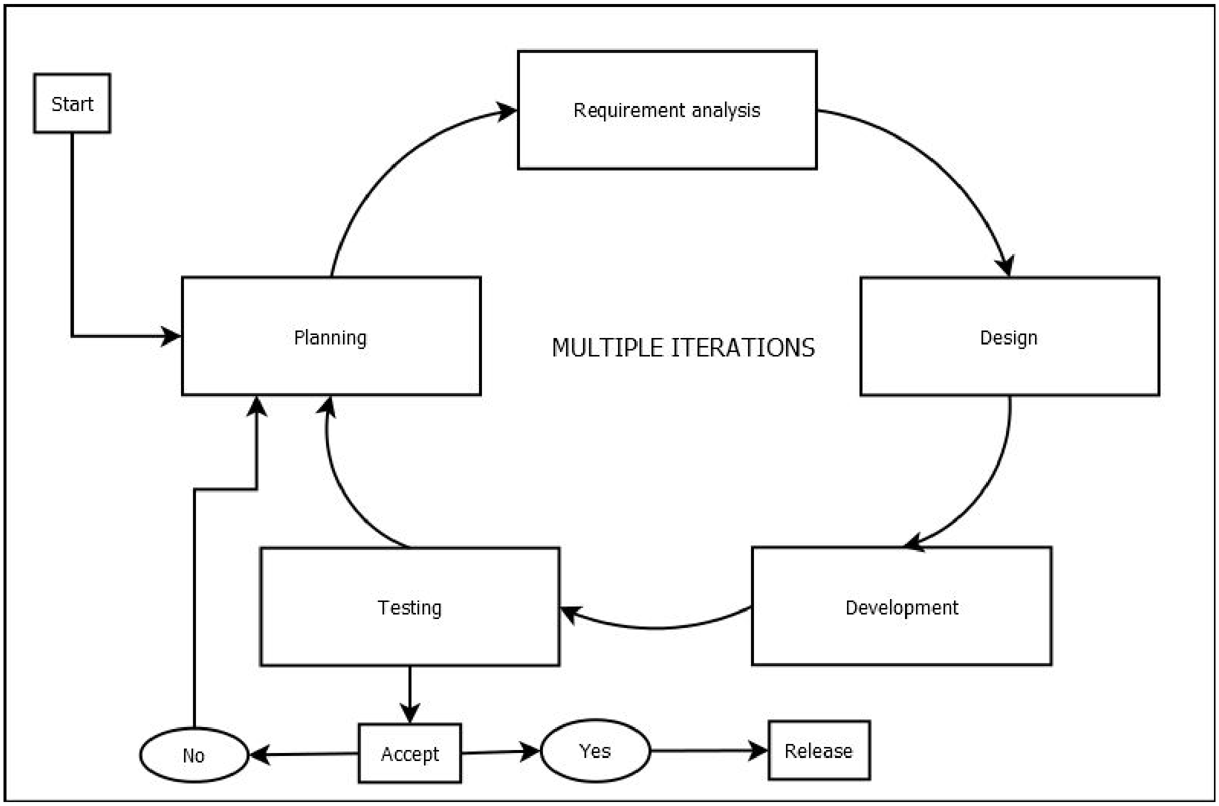


Fig 3.2 - Agile Lifecycle Model

Project schedule

ACTIVITIES	TIME-FRAME
Problem and Scope Definition	15-07-2019 to 29-07-2019
Requirement and feasibility analysis	29-07-2019 to 12-08-2019
Literature survey and research	12-08-2019 to 26-08-2019
Design and Planning	26-08-2019 to 09-09-2019
Design and Planning (UML Diagrams)	09-09-2019 to 23-09-2019
Selection of algorithms	23-09-2019 to 07-10-2019
Implementation Activity to be performed - Pose estimation	07-10-2019 to 21-10-2019
Implementation Activity to be performed - Size estimation	21-10-2019 to 04-11-2019

Unit testing of above modules	04-11-2019 to 18-11-2019
Development (Sprint 1)	06-01-2020 to 20-01-2020
Development (Sprint 2) Activity to be performed - 3D clothes model creation	20-01-2020 to 03-02-2020
Development (Sprint 3) Activity to be performed - Mapping of clothes on users	03-02-2020 to 17-02-2020
Development (Sprint 4) Activity to be performed - User interface	17-02-2020 to 02-03-2020
Unit testing of above modules	02-03-2020 to 16-03-2020
Integration of various modules	16-03-2020 to 30-03-2020
Stress testing of entire application	30-03-2020 to 13-04-2020
Analysing the results achieved	13-04-2020 to 27-04-2020
Deployment	27-04-2020 to 11-05-2020

Table 3.2 - Project Schedule

Milestones

- Realtime pose detection
- Mapping 3D clothes model on user
- Integration of various modules
- Stress testing of integrated application

Cost estimation

From the user's/client's point of view, as no specialized hardware is required, no cost will be incurred. Only hardware required is a smartphone camera which is readily available.

From developers point of view, the tools and technology used are open source and free of cost.

Since the business model of this project is to provide services to online clothes retailers, only hosting costs will be incurred.

Task and Responsibility matrix

	Ojas	Harsh	Tanay	Murtaza
Problem and Scope Definition	X	X	X	X
Requirement and feasibility analysis	X	X	X	X
Literature survey and research	X	X	X	X
Design and Planning	X	X	X	X
Design and Planning (UML Diagrams)	X	X	X	X
Selection of algorithms	X	X	X	X
Implementation Activity to be performed - Pose estimation			X	X
Implementation Activity to be performed - Size estimation	X	X		
Unit testing of above modules	X	X	X	X
Development (Sprint 1) Activity to be performed - UI development	X	X		
Development (Sprint 2) Activity to be performed - 3D clothes model creation	X	X		
Development (Sprint 3) Activity to be performed - Mapping of clothes on users			X	X
Development (Sprint 4)			X	X
Unit testing of above modules	X	X	X	X
Integration of various modules	X	X	X	X
Stress testing of entire application	X	X	X	X
Analysing the results achieved	X	X	X	X
Deployment	X	X	X	X

Table 3.3 - Task and responsibility matrix

Software Requirements Specification Plan

System Features

User related product functions are as follows-

- Pose estimation - Estimate the user’s movement which will help in real time mapping of clothes
- Mapping of clothes - The 3D model of clothes will be mapped on user’s skeleton we get from pose estimation
- Create new cloth pattern - This functions will be used only by a designer where they can create new pattern which will be mapped on cloth model

Operating Environment

Tools and technology used area as follows:

- Unity Real-Time Development Platform - for mapping clothes on user model
- Blender Open Source 3D computer graphics software - to create 3D clothes model
- Android OS - for application development

Operating system - Windows 10

Database - SQLite

User Documentation

At each step, the users will be guided about how to use the application. For example, in size and pose estimation step, the user will be prompted about how to place the camera for effective estimation.

User Interfaces

The user interface will be an android application. The first interface will be for pose and size estimation. Then there will be an interface for users to select different types of clothes that are available in the database. Also, designers will have an interface where they can upload own designed clothes pattern

Software Interfaces

To interact with the database, we will use the SQLite module in python for loading different cloth patterns.

Performance Requirements

- Lag-Free Experience

- Accurate Mapping

Proper alignment of various predefined points of 3D cloth models with actual body parts

Safety Requirements

There will be no harm or damage incurred from this product

Security Requirements

- Data acquired from the user should be private and.
- Photos/Video Frames that might be captured during mapping should be private.

Database Requirements

There will be a database which will store designer names, cloth type (full t-shirt, half sleeves, etc), cloth color, cloth pattern, gender, clothing size, etc.

Software Quality Attributes

The important quality attributes for the user are reliability and correctness. The user size estimation should be very accurate. Also the pose estimation and mapping of the clothes model should be done in real time for better user experience

Business Rules

The business model will be to provide services for online clothes retailers like flipkart, amazon, myntra, etc. This product will also help fashion designers to design and try different new clothes patterns.

Software Design Document (All applicable UML diagrams)

Use Case Diagram

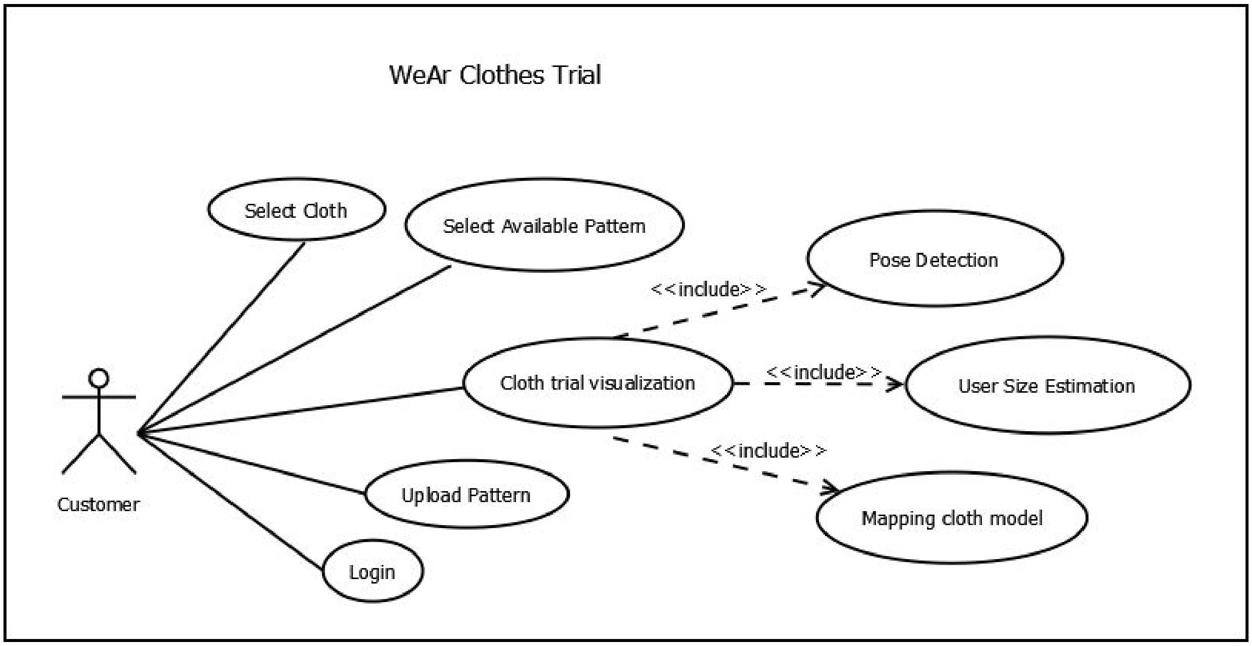


Fig 3.3 - Use Case Diagram

Sequence Diagram

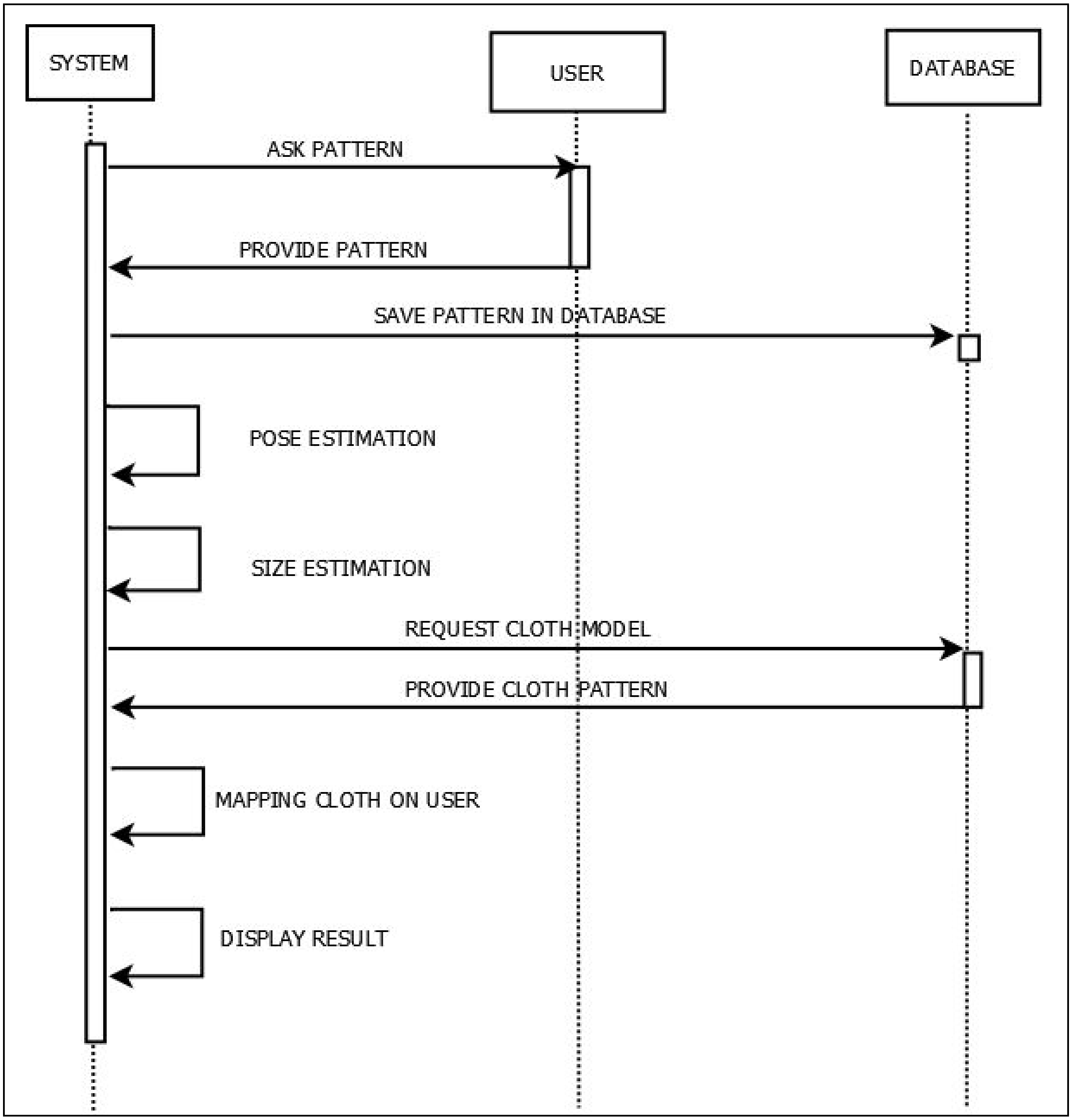


Fig 3.4 - Sequence Diagram

Class Diagram

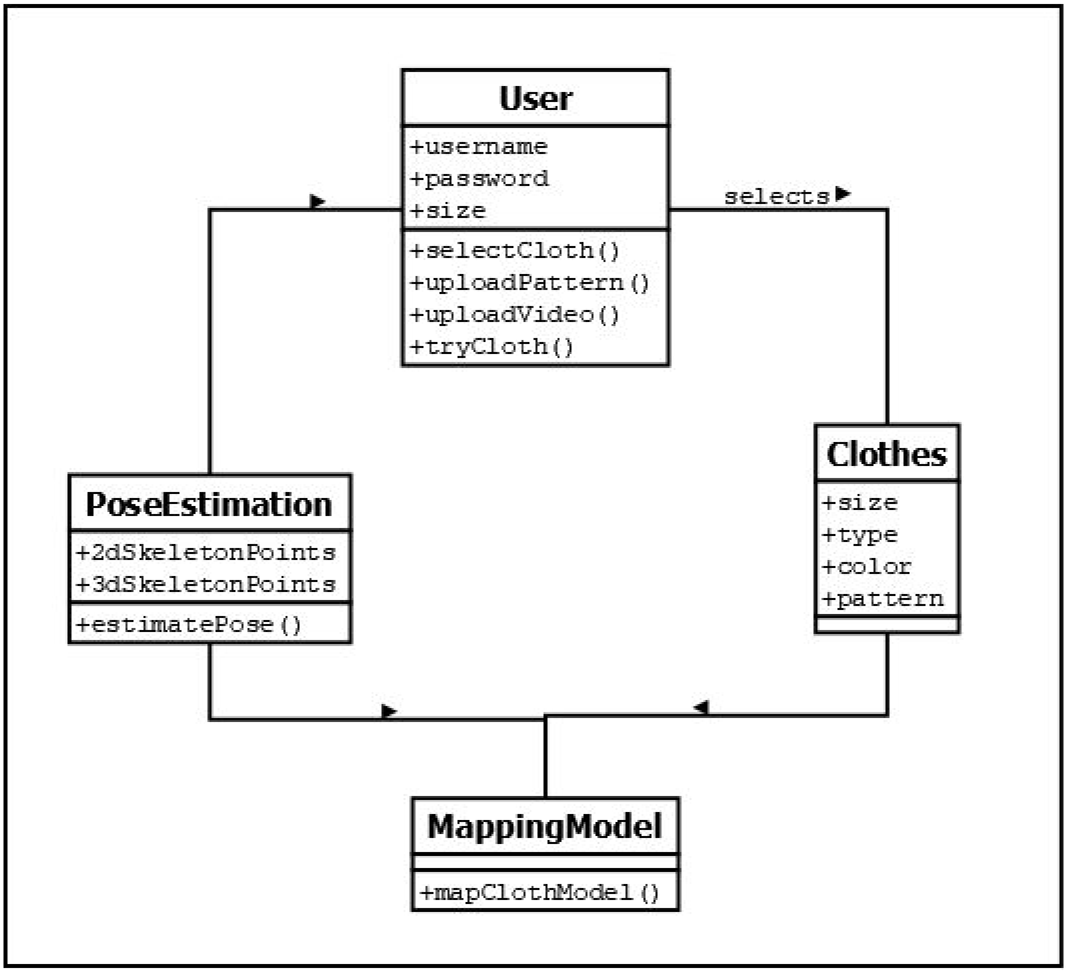


Fig 3.5 - Class Diagram

Activity Diagram

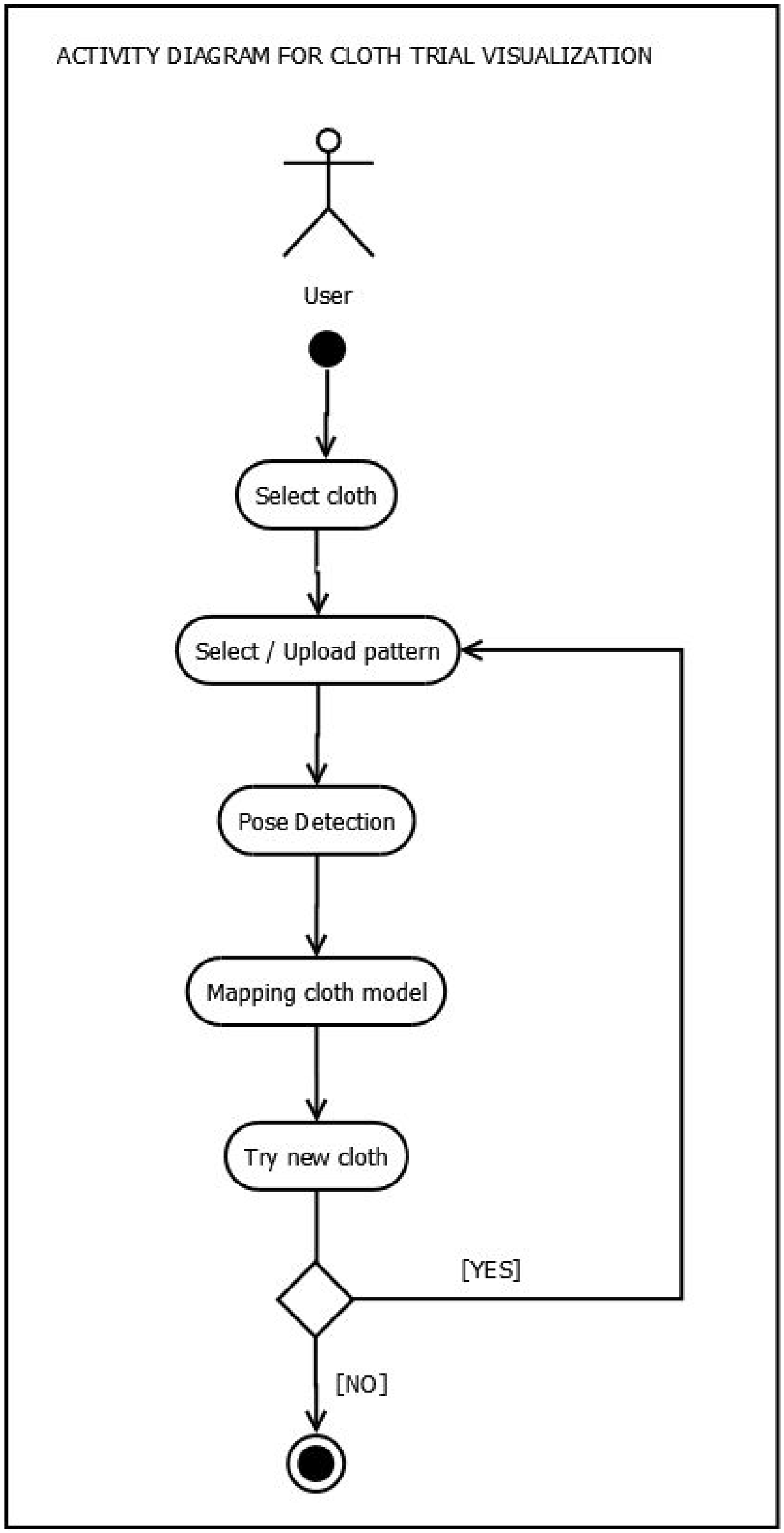


Fig 3.6 - Activity Diagram

Chapter 4 Implementation and experimentation of Prototype model

This chapter presents the results which were inferred after experimenting, implementing and testing of various modules during the designing and development phase.

Proposed system model implementation:-

- **Server-side implementation n flow**

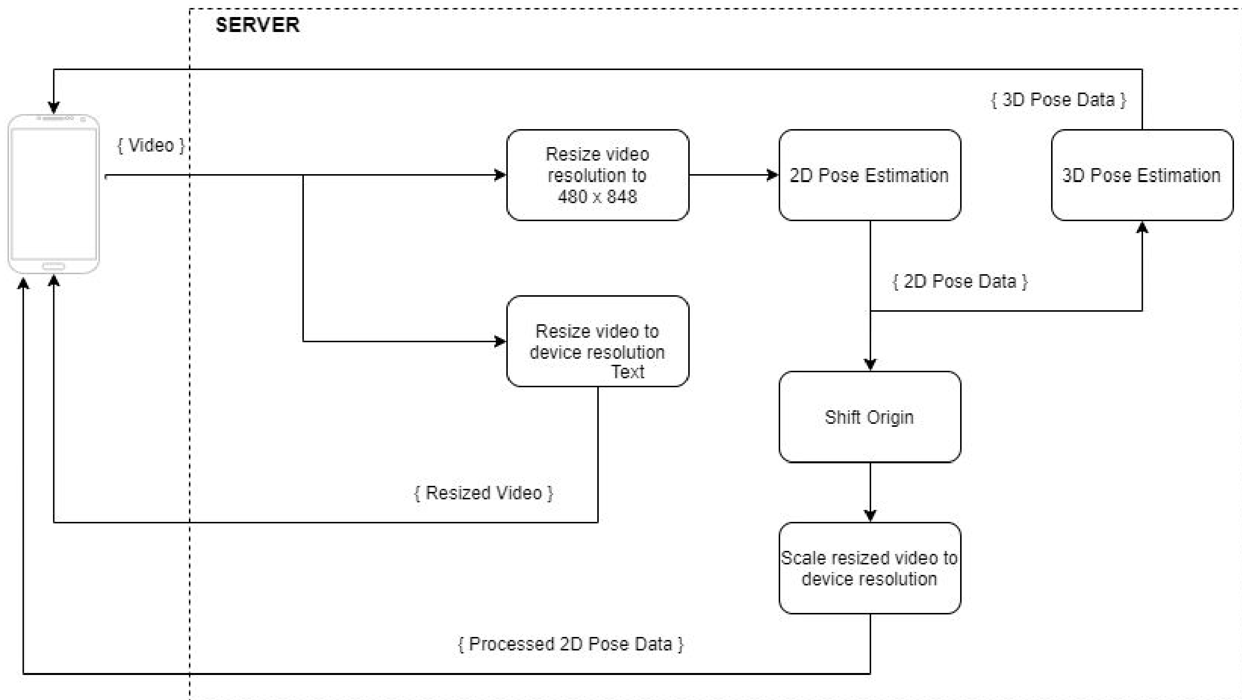


Fig 4.1 - Server Side Model

Working:-

The user uploads the video to Server.

On receiving the uploaded video, the server starts 2 jobs in parallel

1. In the first job, it resizes the video as per the device resolution. This is done using the OpenCV library in python.
2. The second job will involve running various pose estimation models.

a. Initially, it will resize the video to a fixed 480 x 848 resolution. This is an optimization step for our pose estimation model since a lower resolution video will take less time to process.

b. The rescaled video is sent to the 2d pose estimation model. We have used the following open source 2d pose estimation model in our project (https://github.com/ildoonet/tf-pose-estimation).

The output of 2d pose estimation is a list of (x,y) joint pixel coordinates.

Since we had rescaled the video to a fixed resolution(480 x 848) before, the 2d pose estimation points are currently within that resolution. In the next step, we will need to scale those points to the device video resolutions. To do so, we make use of the Min-Max normalization method.

c. The 2d pose estimation output from the previous step given as an input to the 3d pose estimation model. We have used the following open source 3d pose estimation model in our project. https://github.com/ArashHosseini/3d-pose-baseline/

The 3d pose estimation model outputs (x,y,z) joint points.

3. Finally, the 2d pose estimation point from step 2.b, 3d pose estimation points from step 2.c and the rescaled video from step 1 is sent back to the user's device.

- **Client-side implementation**
  - **Mapping Operation**

    The main objective of mapping operation is to superimpose the 3D cloth model on the user's body.

    The **input** to this operation are:-

    - 2d Pose Estimation Points
    - 3d Pose Estimation Points
    - Selected 3d cloth model
    - User Video

The **output** from this operation will be our final product i.e a user's video superimposed with the selected 3d cloth model.

- **Issues/Difficulties and our Solution:-**

  Dealing with 3 Coordinate systems.

  One of the main difficulties we faced was to handle the 3 conversions between 3 totally different coordinate systems. The following is the list of those coordinate systems.

  1. 2d Pose Estimation Coordinate System.

     These were pixel-based (x,y) coordinates. neck
  2. 3d Pose Estimation Coordinate System.

     (x,y,z) coordinate points. neck
  3. Unity Coordinate System.

  The **Solution** we came up with to deal with these Issue is as follows:-

  Conversion from 3d pose estimation points to Unity coordinates.

  This was achieved with the help of Camera.ScreenToWorldPoint function in Unity.

  Preprocessing Step for enhancing 3d joint Points

  Following are the list of steps carried out in sequence to enhance the 3d joint points for mapping:-

  1. For the First Frame make all joint coordinates equal to torso. Save this joint coordinates in a variable(hereafter referred to as prevFramePoints).
  2. Now for each Frame(let's call this currFrame), calculate the difference in Z coordinate with the previous frame for each joint point(let's call this diffZ).
  3. Calculate the new Z position for each joint point using this formula

     newZ = prevFramePoints.Z + diffZ
  4. Keep the X and Y coordinate the same and update the new Z coordinate for each joint point.

So the new Joint Point for the current Frame will be

newJointPoint = (currFrame.X, currFrame.Y, newZ)

5. Store currFrame in prevFramePoints variable.

Scaling the 3d cloth model

The scale of the 3d cloth model can be handled by the following steps:-

1. First, in Blender, modify the clothes
   a. shoulder-length(neck to shoulder length),
   b. elbow-length(shoulder to elbow length)
   c. full torso-length(neck to waist).

   Note that this length will be the actual length of the cloth in the real world. Keep the dimensions in Meters.
2. Then set the scale of the 3d cloth model to 1.00 in Blender.

Asynchronous Video Processing

Another problem we faced was that the user had to wait till the video was processed at the server side and return back to the user. For this we implemented a push notification service where users will periodically ask the server for the processed video. Once the video is ready the user will receive notification about it. Also, users need not upload the same video again and again. The output of pose estimation is saved in local device storage so that the user can try clothes any time without waiting again for video processing at the server

Inclusion of Any additional details as suggested by Project Guide/during progress seminar

Cloth Model wasn't rotating towards Right

(Suggested in Progress Seminar of Sem 8)

After debugging we found out that this issue was a result of inaccurate results of 3d pose estimation model. The Z coordinates of joint points were not accurate and as a result our mapping module was not working and the Cloth model was appearing as tilted left.

The solution we came up with to handle this problem was to add a preprocessing step for the result of 3d pose estimation model. This preprocessing step include tasks like aligning the z coordinate of torso and neck, etc

4.3 Software Testing (Software testing reports at various levels)

4.3.1. Introduction

This chapter documents and tracks the necessary information required to effectively define the approach to be used in the testing of the project's product. The Test Plan document is created during the Planning Phase of the project. Its intended audience is the project manager, project team, and testing team.

4.3.1.1 Test Approach

Proactive approach: Proactive approach involves anticipating possible bugs while the modules are being made and resolving them as when they are uncovered. So the integration product is easier to debug thereby reducing the number of variables.

4.3.2. Test Plan

4.3.2.1 Features and Modules to be tested

1) 2d pose estimation module
2) 3d pose estimation module
3) Video resolution conversion module
4) Estimated Pose scaling scripts
5) Selecting multiple patterns feature
6) Uploading a new pattern feature
7) Uploading a new video feature
8) Asynchronous Video Processing Module
9) Authentication module

4.3.2.2 Testing Method and Tools Used

Some modules were tested using an Automated Testing tool named ZAPTEST, while the rest of the modules and features were manually tested.

4.3.3 Test Cases

4.3.3.1 Test Case 1

Description: Testing 2d pose estimation model

Valid Input: .mp4 video file of the user that is uploaded to the server

Expected Output: The pose estimation points for each of the frame in the video is saved on the server in a folder called json_files/video_name

Actual Output: As expected

Test Result: Pass

4.3.3.2 Test Case 2

Description: Testing 3d pose estimation model

Valid Input: The output of the 2d pose estimation model, i.e., all the files in the folder 'json_files/video_name'

Expected Output: Two files are generated as output in 'json_files' folder named '{video_name}_2d.json' and '{video_name}_3d.json'

Actual Output: As expected

Test Result: Pass

4.3.3.3 Test Case 3

Description: Testing video resolution conversion

Valid Input: .mp4 video file of the user that is uploaded to the server

Expected Output: The video resolution is converted to 480x848 and saved to the folder 'resized_videos'

Actual Output: As expected

Test Result: Pass

4.3.3.3 Test Case 4

Description: Testing processing of the output of 3d pose estimation - Shift of origin and scaling of the 3d pose estimation output

Valid Input: Output generated by the 3d pose estimation model - 'json_files/{video_name}_2d.json' and 'json_files/{video_name}_3d.json'

Expected Output: The output of the 3d pose estimation model is scaled and shifted properly and saved in the same files.

Actual Output: As expected

Test Result: Pass

4.3.3.5 Test Case 5

Description: Testing App feature to change patterns on cloth model

Valid Input: Any one of the available patterns should be selected

Expected Output: The selected pattern is properly mapped to the demo cloth model

Actual Output: As expected

Test Result: Pass

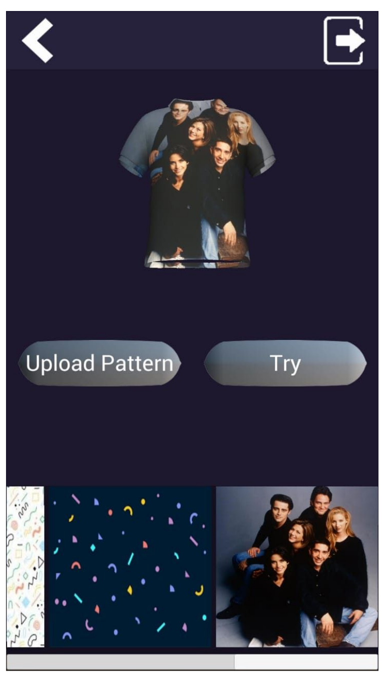


Fig 4.2 - Test Case-5 Output

4.3.3.6 Test Case 6

Description: Testing App feature to upload new patterns

Valid Input: JPG pattern image file

Expected Output: Pattern successfully uploaded and pattern thumbnail should be visible in the pattern selection Page for valid inputs and error message displayed for invalid input.

Actual Output: As expected

Test Result: Pass

Fig 4.3 - Test Case-6 Output

4.3.3.7 Test Case 7

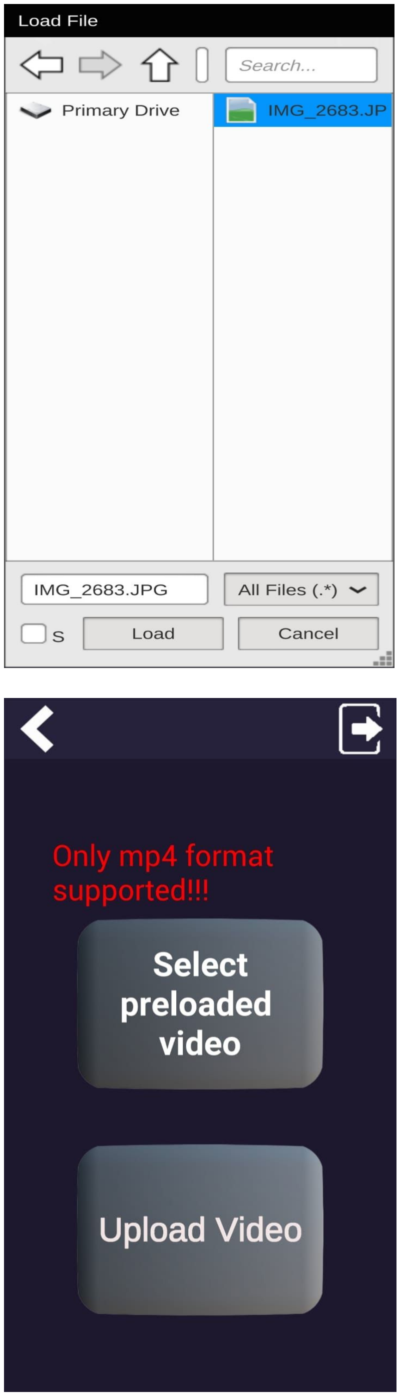
Description: Testing App feature to upload a new video

Valid Input: An video file in MP4 format

Expected Output: Processing Video Dialog Box should appear

Actual Output:

If Invalid Video File is Selected	If Valid Video File is selected



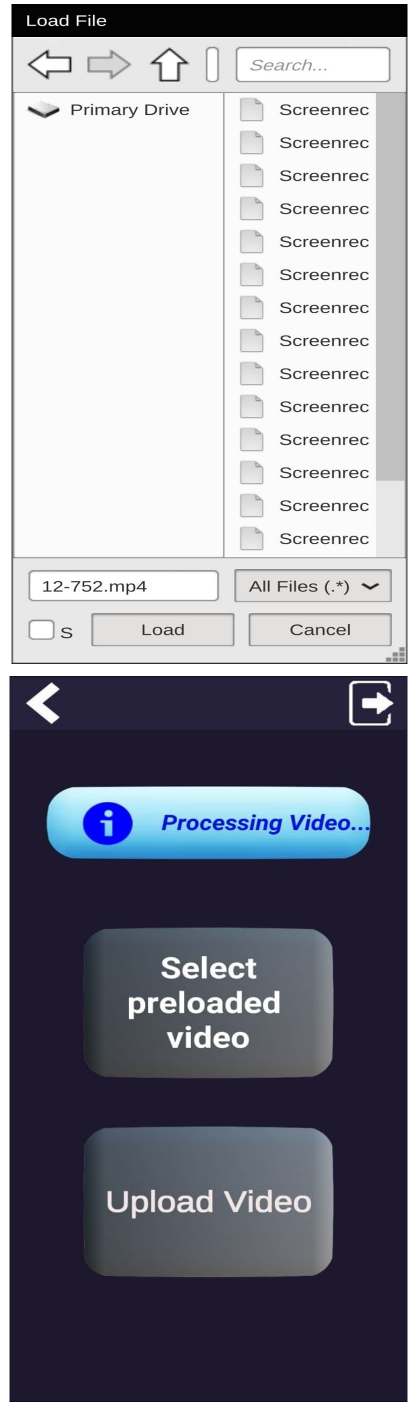


Fig 4.4 - Test Case-7 Output

Test Result: Pass

4.3.3.9 Test Case 8

Description: Testing of Login Page

Valid Input: Registered User's Username and Password.

Expected Output: Redirect to Video selection Page

Actual Output:

For Invalid Login

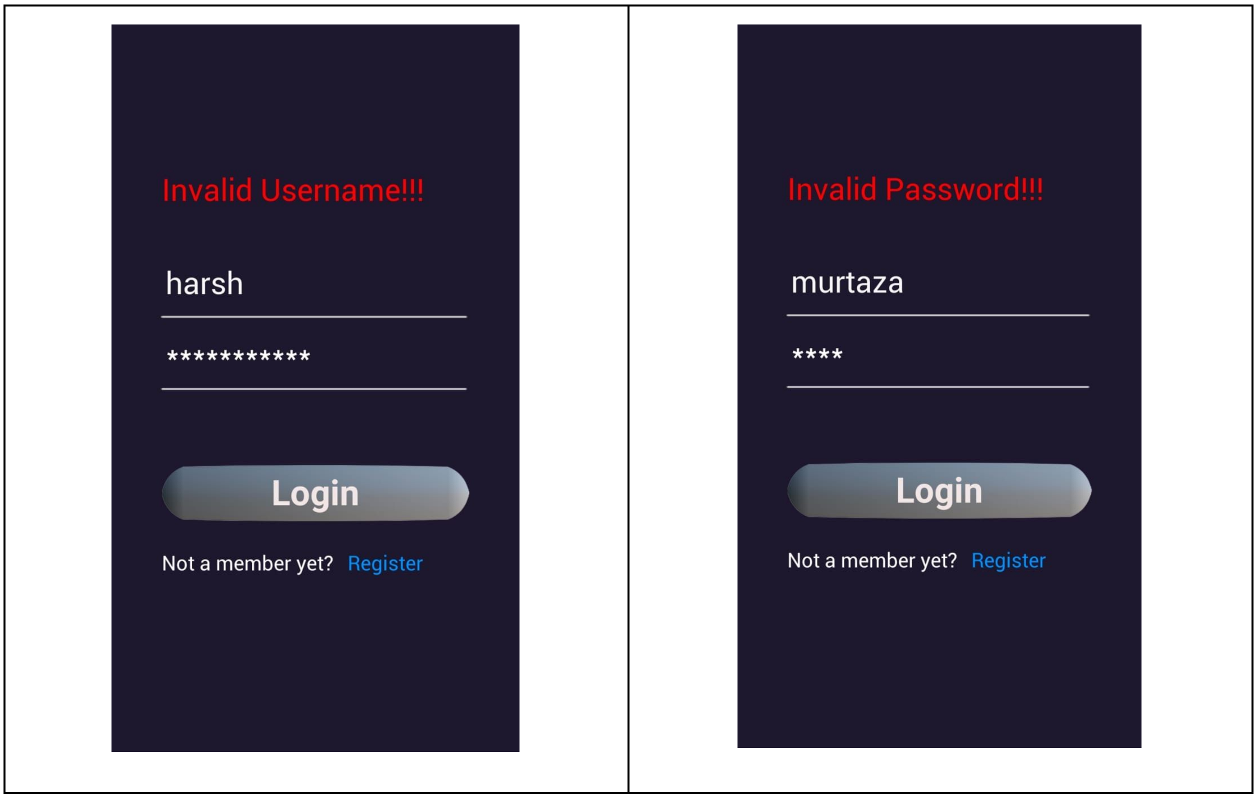


Fig 4.4 - Test Case-8 Output

Test Result: Pass

4.4 Experimental results and its analysis

App Demo Screenshots:-

Fig 4.5 - App Demo Screenshot 1

Fig 4.6 - App Demo Screenshot 2

Fig 4.7 - App Demo Screenshot 3

Analysis:-

2d pose Estimation model(Openpose) analysis:- The research paper[8] stated that the accuracy of this model is around 90-95%. We tested this model on some real world videos of our project and found that the accuracy is acceptable.

3d Pose Estimation model analysis:- The research paper of this project states that the accuracy of this model is about 45.5mm. The metric is an average error in millimetres between the ground truth and our prediction across all joints and cameras, after alignment of the root (central hip) joint.

However after debugging the 3d joint points in Unity, we found out that the accuracy of this model on our project dataset is very less. Due to low accuracy our mapping algorithm wasn't functioning as expected as a result the mapping visualization wasn't proper. We solved this error by adding a preprocessing step of 3d joint points before giving it as an input to the mapping module. (For more info refer to the implementation section above).

Processing analysis:-

Processing time for Video with 300 frames

Machine spec:-

Intel i5 core processor (8 multithreaded processors)

8GB RAM

Model	Time
2d pose estimation	8 minutes
3d pose estimation	2 minutes
	Total → 10 minutes

Table 4.1 - Processing Time Analysis

Chapter 5

Conclusions and further work

This report has presented an augmented reality application in which users can select and try on virtual clothes. These clothes are rendered on a screen over the image of the user. The presented application is an improvement over similar existing augmented reality applications in that it offers the same functionality without any specialized hardware requirement. The tasks carried out in this semester were project implementation and software testing. Currently we have built an application for only Android OS, however this application can be easily extended for other OS like IOS, since most of the Unity Code can be reused.

5.1 Further Work:

- Enhancing the accuracy of 3d pose estimation model..
- Adding support for more cloth models like Shirt, Pants, Hats, etc

5.2 Acknowledgement

A project is the creative work of many minds. A proper synchronization between individuals is a must for any project to be successfully completed. It is only due to the complete dedication of students that is combined with the guidelines of the college professors that any task can be completed.

We are thankful to our guide and mentor Prof. Suchita Patil who provided expertise that greatly assisted in the learning and successful implementation of the Project 'WeAR'. We would like to thank her for constantly motivating and pushing us to work harder, which made this project into reality. We also express our appreciation to Prof. Bharathi HN for sharing invaluable and priceless insights with us during the course of developing the project and several project seminars. We are also immensely grateful to our institute for providing us with the infrastructure support.

Bibliography

[1] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu and Larry S. Davis, "VITON: An Image-based Virtual Try-on Network". [https://arxiv.org/abs/1711.08447]

[2] Akshay Shirsat, Samruddhi Sonimindia, Sushmita Patil, Nikita Kotecha, Prof Shweta Koparde, "Virtual Trial Room", International Journal of Research in Advent Technology (IJRAT), VOLUME-7 ISSUE-5, MAY 2019, pp. 182-185. [https://doi.org/10.32622/ijrat.75201976]

[3] Amoli Vani, Dhwani Mehta and Prof. Suchita Patil, "Virtual Changing Room". K. J. Somaiya College of Engineering.

[4] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov and Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks". IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. [https://arxiv.org/abs/1801.04381]

[5] Ondrej Kainz, František Jakab, Matúš W. Horečný and Dávid Cymbalák, "Estimating the Object Size from Static 2D Image". 2015 International Conference and Workshop on Computing and Communication (IEMCON). [https://ieeexplore.ieee.org/document/7344423]

[6] Pose Estimation for mobile (Implementation of CPM and Hourglass model using TensorFlow and MobileNetV2) [https://github.com/edvardHua/PoseEstimationForMobile]

[7] Alejandro Newell, Kaiyu Yang, and Jia Deng, "Stacked Hourglass Networks for Human Pose Estimation". [https://arxiv.org/pdf/1603.06937.pdf]

[8] OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Field. [https://arxiv.org/abs/1812.08008]

[9] A simple yet effective baseline for 3d human pose estimation. [https://arxiv.org/pdf/1705.03098.pdf]