



## Augmented Reality Clothes Fitting Web App Using Three.js & Posenet

<sup>1</sup>Rushikesh Kore, <sup>2</sup>Hansel D'silva, <sup>3</sup>Aaditya Shivapurkar, <sup>4</sup>Reena Kharat

<sup>1,2,3</sup> BE Student, <sup>4</sup>Associate Professor

<sup>1,2,3,4</sup> Department of Computer Engineering

<sup>1,2,3,4</sup> Pimpri Chinchwad College of Engineering, Pune, India

**Abstract:** The covid-19 pandemic has prompted a great deal of interest in online shopping for clothes. We have explored different methodologies for virtually replicating the experience of a clothes fitting room using Augmented Reality. We have developed a web-application that allows potential customers of fashion ecommerce to virtually try-on a particular outfit of their choosing via input from their webcam. Our application breaks down this problem into two processes: pose-estimation via the PoseNet 2.0 tensorflow.js library, and 3D model animation using the three.js library. We have briefly explained these processes along with suitable illustrations. This research paper provides a brief idea about virtual cloth fitting using the technology known as augmented reality.

**Index Terms - Augmented Reality, PoseNet, ResNet, Javascript.**

### I. INTRODUCTION

Fitting clothes virtually is a very fascinating concept. It allows the user to see themselves wearing virtual clothes of their choice via their camera device. The latest advancements in web technologies have enabled such an endeavor to become feasible with the help of augmented reality. A recent survey paper<sup>[1]</sup> explored various methodologies for the purpose of virtual fitting-room. This paper extends upon those foundations for the web-app 3D-based implementation of the same. This problem is broken down into two processes: Pose Estimation<sup>[12]</sup> & Cloth Rendering<sup>[13]</sup>. The pose estimation is implemented using a convolution neural network called "Posenet". A Convolutional Neural Network is a Deep Learning algorithm that can take in an image, assign learnable weights and biases to various aspects/objects in the image, and distinguish one from the other. Posenet is an algorithm which detects body and estimates body position in real time. Posenet is available in the tensorflow library. TensorFlow is an open-source machine learning library that can be used for both research and development. It is a framework to create deep learning models. Next process is to overlay virtual clothes on the body in real time. We have used a javascript library called **three.js**<sup>[8]</sup> for loading and overlapping of virtual clothes. Three.js is a javascript library which is used to load 3D assets or content in a web browser. Last step is to update the position of clothes with respect to the body. These three processes are briefly described in methodology.

### II. METHODOLOGY

#### 1. Pose Estimation:

This module talks about the first step of this application i.e. Pose Estimation. Pose estimation refers to using computer vision for detecting human beings in photographs and videos, for example as determining where someone's knee appears in a photograph. The pose estimation algorithm can calculate the location of main body joints in a particular photograph or a real-time footage. This section of the project is the most important part & it heavily relies on PoseNet which is an Artificial Neural Network based on ResNet-50 which is pre-trained convolutional neural network programmed for image recognition.

#### 2. ResNet – 50:

ResNet-50<sup>[10]</sup> is a residual network with a large number of nodes. It's a convolutional neural network & number "50" in its name refers to the number of layers it has. Vanishing gradients are a problem for deep neural networks. Resnet uses skip connections to relay information through layers, enabling data scientists to create more complex networks. The network is able to understand global features using skip connection. Since details can be lost in CNN due to max pooling, this can improve the image's identification of smaller artifacts.

### 3. PoseNet 2.0:

Many pose estimation algorithms, that are open source, require additional hardware except PoseNet 2.0v<sup>[11]</sup>. Pose 2.0 is a new & improved version with improved accuracy and added ability to catch limbs in all sorts of strange positions, which was not always the case with the PoseNet 1.0 model since it was based on MobileNet. Added weight quantization allows for the download of much smaller models with minimal effect on accuracy.

PoseNet 2.0 implemented on TensorFlow.js can work on any regular phone or desktop that has internet & descent webcam. It takes RGB input from the camera & feeds it into a convolution neural network. The algorithm then calculates the following:

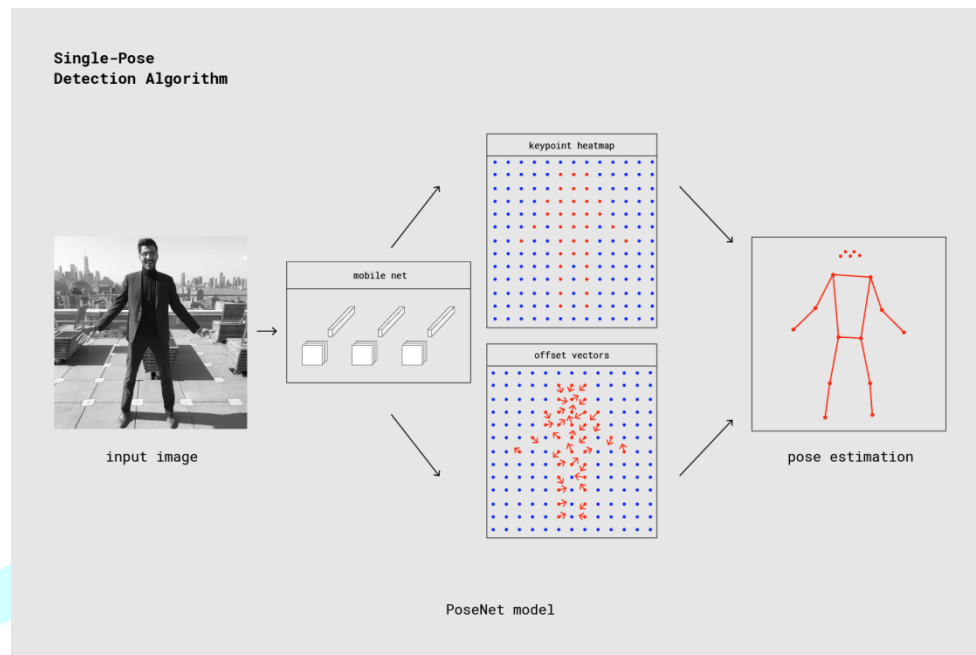


Figure 1: Working of Posenet Algorithm

1. List of Keypoints: Key Points includes a position and a confidence key.
2. Pose Confidence Score: Score ranges from 0 to 1.
3. Confidence Key: This ranges from 0 to 1 and describes how accurate a particular keypoint is.
4. Position: 2D X & Y coordinates in the original image where the keypoint is recognized.

Along with the RGB input footage, posenet also takes following things:

1. Image Scale Factor: Value ranges between 0.2 to 1.0 & defaults to 0.5. Scale factor refers to scaling the media before feeding it into. 0.5 is original scale, value lower than 0.5
2. Lowers the scale & value greater than 0.5 increases it.
3. Horizontal Flip: Value is either 1 or 0 & default is 0. Flips the media horizontally before feeding it in.
4. Output Stride: Value can be 32, 16 or 8 & defaults to 8. This parameter influences the height & width of layers in a neural network which can affect accuracy & speed. Higher value means faster speed but the accuracy is lowered & lower values means slower speed but higher accuracy.

### 4. 3D Clothes Model Animation - three.js:

We have chosen the three.js<sup>[2]</sup> library for our real-time clothes animation. We wanted something that wasn't too performance-intensive<sup>[3]</sup> to fulfill our requirements for a fluid and seamless clothes animation and overlay. At the same time, we needed it to be functional enough to allow for point-based reverse kinematics. The three.js javascript library was a good fit for our use-case.

three.js is an easy-to-use, lightweight, cross-browser, general-purpose 3D library Javascript 3D library. It makes 3D animation inside a browser<sup>[4]</sup> very accessible without the need for high-end hardware.

We have used three.js to:

1. Start a new WebGL Renderer
2. Start a new three.js scene which represents the 3D environment.
3. Set the axis angle of the joints of the clothes model using the coordinates output by PoseNet.
4. Smoothly rotate the joints as per the set axis angle using three.js' quaternion object.

The above steps are performed continuously in realtime to give the user a seamless experience of clothes overlaying.

### 5. Updating Clothes Position:

This section talks about updating 3d-clothes position with respect to the change in body position.

Using PoseNet, body pose estimation has been estimated in real time. After that we loaded 3D-clothes using a javascript library called three.js. In the next step, we are going to update clothes' position and movement with respect to the body's position and movement.

In the first step, we divide the cloth model into different parts just like the posenet algorithm divided body parts into 17 different key points. For example, cloth model A is divided into following points.

Hips, Spine, Chest, Upper Chest, Upper left Arm, Upper Right Arm, Lower Left Arm, Lower Right Arm, Upper Right Leg, Lower Right Leg, Left foot, Right foot.

In order to update clothes position there are two that need to be taken into consideration.

- 1) Updating of position of the cloth parts.
- 2) Updating of angle of rotation of cloth joints.

### 6. Updating of position of cloth parts :

Person moves in the left-right direction and forward-backward direction. The three.js has a 3D coordinate system which has (X, Y, Z) directions. So the position of parts of clothes has been updated according to (X, Y, Z) position.

Resnet gives output of each keypoint in X and Y position. This output of keypoint is taken as input for position clothes. So that position of clothes is updated. Cloth movement changes according to change in movement of the body.

### 7. Updating of angle of rotation of cloth joints:

Here, we have to consider only rotation joints. Here we considered joints of the left arm, right arm, left leg, right leg and waist. Angles are calculated using quaternion which is used to rotate angles in 3 dimensional spaces. Based on outputs of position of X and Y. These are taken as input to calculate angles using quaternion functions. Each angle is calculated as below shown in figure.

$$q_x = v_x \sin \frac{\theta}{2.0}$$

$$q_y = v_y \sin \frac{\theta}{2.0}$$

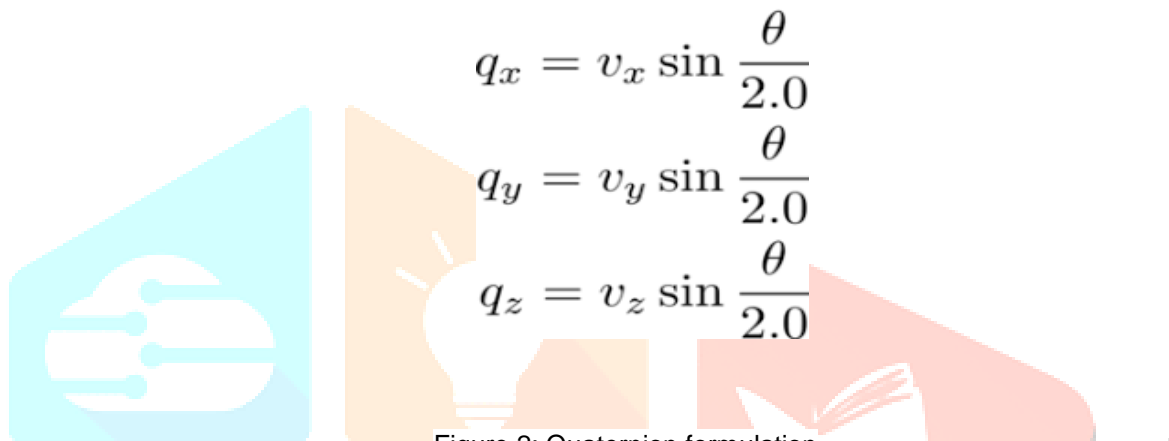
$$q_z = v_z \sin \frac{\theta}{2.0}$$


Figure 2: Quaternion formulation

Here  $Q_x$  is angled along the x-axis.  $Q_y$  is angled along the y-axis and  $Q_z$  is angled along the z axis.  $V_x$  is the position of x-axis.  $V_y$  is the position of the y-axis.  $V_z$  is the position of z-axis. So using these techniques position and movement of clothes is being estimated and updated at runtime. As a final output the cloth follows the trajectory of the body in runtime.

### III. VRM & VRoid Studio:

The various 3D models for different clothes to be overlaid upon the user are a very significant aspect of our application. We wanted a simple and reusable way to quickly design and integrate the 3D models into the system as per the user's requirements. Also, we did not prefer to manually rig every individual model as that would prove to be quite tedious. VRM, along with VRoid studio was a good solution that fulfilled our needs.

VRM is a special 3D model file format specifically created for handling humanoid models. It is based on glTF2 which is open-source and free-to-use. It is intended for VR applications which have an extensive use of human avatars. Unlike other conventional 3D file formats like FBX and OBJ, VRM comes with many features related to avatar designing and animation. All VRM models are designed as per a standard skeleton design meaning they are already rigged and ready to animate.

Although VRM supports complete human models, we customized each VRM model for our clothes-fitting purpose by omitting the exposed skin layers and only leaving the clothes layers. This was done quite easily using the VRoid Studio tool. VRoid Studio<sup>[5]</sup> is a free GUI application to easily create our own custom human models.

It comes with many useful features including altering the size and shape of the face and body feature, changing the color and overall design of the clothes, adding and deleting layers related to a particular section of the model, and many more. VRoid greatly simplified our process of obtaining unique 3D clothes models. Its powerful features allowed us to generate a decent number of ready-to-use VRM models in less time which would have proven tedious and error-prone with conventional 3D file formats.

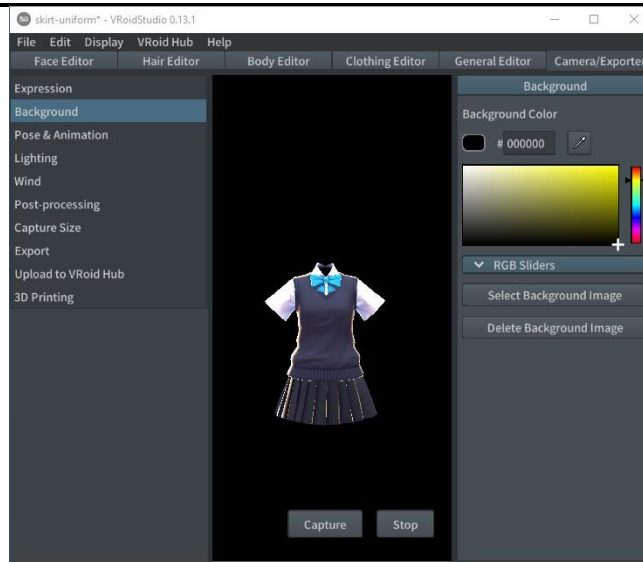


Figure 3: Clothes model preparation using VRoid Studio

#### IV. Conclusion and Future Scope:

Highly interactive and cross-platform clothes-fitting web-application can be developed using free and open source Javascript libraries. The PoseNet library for tensorflow.js provides an accurate and efficient pose-estimation CNN to obtain the position and pose of the user. The three.js library can then be used to animate a clothes-only 3D VRM model to match the coordinates of PoseNet. We have tested an early functional prototype with a responsive fashion ecommerce frontend to considerable success.

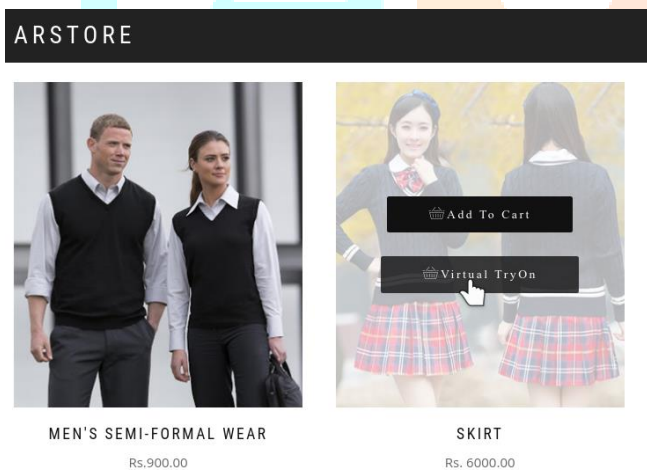


Figure 4: Fashion ecommerce frontend for virtual try-on



Figure 5: Clothes-overlaying using PoseNet 2.0

We seek to further improve our application by adopting cloud-based solutions to enhance the frame rate and improve accessibility for phones and low-end devices. We are also trying to solve depth-related problems which sometimes cause misalignment between the user's body and the overlaid outfit. Finally, we are working on improving the realism of the 3D models so that it blends seamlessly into the user's indoor environment.

## V. References

- [1] Hansel D'Silva, Aaditya Shivapurkar, Rushikesh Kore - "Research Paper - Virtual Clothes Fitting Application based on Augmented Reality for Online Retailers", Journal of Critical Reviews, 2020.
- [2] Baruah, Rakesh. "Creating an Augmented Reality Website with Three.js and the WebXR API." AR and VR Using the WebXR API. Apress, Berkeley, CA, 2021. 217-252.
- [3] Johansson, Julia. "Performance and Ease of Use in 3D on the Web: Comparing Babylon.js with Three.js." (2021).
- [4] Stanton, Matthew, et al. "Dual-mode user interfaces for web based interactive 3d virtual environments using three.js." IFIP Conference on Human-Computer Interaction. Springer, Cham, 2017.
- [5] Ito, Akihiro. "VRoid Studio | VRoid." VRoid Studio, J-STAGE, 2020, [vroid.com/en/studio](http://vroid.com/en/studio)
- [6] Roman Bajireanu, Joao A.R. Pereira, Ricardo J.M. Veiga, Joao D.P. Sardo, Pedro J.S. Cardoso, Roberto Lam, Joao M.F. Rodrigues - "Mobile Human Shape Superimposition: an initial approach using OpenPose", International Journal of Computers, 2018.
- [7] Yoones A. Sekhavat - "Privacy-Preserving Cloth Try-On Using Mobile Augmented Reality". IEEE TRANSACTIONS ON MULTIMEDIA, 3 May 2018.
- [8] Francisco Pereira, Catarina Silva and Mário Alves - "Virtual Fitting Room Augmented Reality Techniques for e-Commerce", Computer Science and Communication Research Centre, 2011.
- [9] M. Samir Abou El-Seoud - "An Android Augmented Reality Application for Retail Fashion Shopping", British University in Egypt-BUE, Cairo, Egypt.2019.
- [10] Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun - "Deep Residual Learning for Image Recognition", Microsoft Research
- [11] Alex Kendall Matthew Grimes - "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization", University of Cambridge agk34, mkg30, rc10001 @cam.ac.uk Roberto Cipolla
- [12] Toshev, Alexander, and Christian Szegedy. "DeepPose: Human Pose Estimation via Deep Neural Networks." 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, doi:10.1109/cvpr.2014.214.
- [13] "Soft Skin Simulation for Wearable Haptic Rendering." *Fundamentals of Wearable Computers and Augmented Reality*, 2015, pp. 570-601., doi:10.1201/b18703-26.

