NCE Journal of Science and Engineering (NJSE), Vol. 2, June 2025

3D Reconstruction of Shoes for Augmented Reality

Pratik Shrestha[1] Santosh Giri[1,*] Vishal Pokharel[1] Sujan Kapali[1], and Swikar Gautam
[1]Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Lalitpur, Nepal
[*]Correspondence: Santosh Giri
Manuscript received January 13, 2025; accepted April 2, 2025

***Abstract*—This paper introduces a mobile-based solution that enhances online shoe shopping through 3D modeling and Augmented Reality (AR), leveraging the efficiency of 3D Gaussian Splatting. Addressing the limitations of static 2D images, the framework generates realistic 3D shoe models from 2D images, achieving an average Peak Signal-to-Noise Ratio (PSNR) of 0.32, and enables immersive AR interactions via smartphones. A custom shoe segmentation dataset of 3120 images was created, with the best-performing segmentation model achieving an Intersection over Union (IoU) score of 0.95. This paper demonstrates the potential of 3D modeling and AR to revolutionize online shopping by offering realistic virtual interactions, with applicability across broader fashion categories.**

***Keywords*—Augmented Reality (AR), 3D Modeling, Gaussian Splatting, and Segmentation.**

I. Introduction

THE rapid advancements in 3D modeling and computer graphics have revolutionized various industries, enabling the creation of immersive and realistic Augmented Reality (AR) solutions. These technologies are increasingly utilized in the fashion industry to enhance customer experiences by enabling interaction and visualization of products before purchase. Despite these advancements, traditional online shopping plat- forms remain limited by their reliance on static 2D images, which fail to replicate the exploratory and sensory experience of physical retail environments.

Historically, generating 3D models from 2D images has been a labor-intensive and time-consuming process, requiring significant human intervention. However, recent breakthroughs have accelerated the modeling process while achieving higher levels of precision and realism. Among the approaches to 3D modeling, Photogrammetry, Neural Radiance Fields (NeRF), and Gaussian Splatting have emerged as prominent techniques, each with distinct advantages and limitations.

Photogrammetry, although widely used, encounters challenges such as handling reflective surfaces, dealing with occlusions, and requiring high-quality input images, which limit its practicality. NeRF, a deep learning-based method, generates highly detailed models but is computationally intensive, making real-time applications infeasible. In contrast, Gaussian Splatting offers a significant advantage by enabling faster training, real-time rendering, and easy modification of the generated models. This paper seeks to find techniques to leverage 3D Gaussian Splatting to streamline the creation of realistic 3D models from 2D images, focusing on its application in enhancing online shopping experiences.

II. Literature Review

Recent advancements in deep learning have revolutionized the synthesis and rendering of 3D models from 2D images, presenting efficient alternatives to conventional Photogrammetry techniques [1] [2] [3]. Among these advancements, two prominent methods, Neural Radiance Fields (NeRF) [1] and 3D Gaussian Splatting [3], have garnered considerable attention for their respective strengths and applications.

NeRF, pioneered by Shrinivasan, Mildenhall, and Tancik in 2020, operates by harnessing radiance fields for view synthesis. By employing Multi-Layer Perceptrons (MLPs), this method learns to predict both the volume density and color of a point based on a 5D input—comprising the spatial location of the camera and its viewing direction. Subsequently, NeRF synthesizes views by querying points along marching rays for color and volume density and applies classical volumetric rendering techniques for rendering. This technique excels in producing true-to-life renderings by learning the volumetric representation of a scene from a collection of images. However, its training process demands significant computational resources, making it impractical for real-time applications.

In contrast, 3D Gaussian Splatting, introduced by Kerbel, Kopanas, Leimkuhler, and Drettakis, leverages Gaussians to render views. Firstly, a point cloud is generated from images using the Structure from Motion(SfM). [4] This approach involves the conversion of individual points within a point cloud into Gaussians, and fine-tuning the parameters of these Gaussians through stochastic gradient descent. Notably, this technique offers a substantial leap in speed compared to the original NeRF, enabling real-time rendering capabilities. As a result, it emerges as a promising solution for scenarios requiring instantaneous interactions, making it particularly suitable for applications such as virtual try-on experiences.

The image given to model for 3D reconstruction should have its background removed. There is a wide variety of segmentation models [5]-[9] available for this task. However, YOLOv8 [10], leveraging state-of-the-art progressions in deep learning and computer vision, is commonly employed because of its lightweight and development tools. YOLOv8 is proficient in various vision AI tasks, encompassing detection, segmentation, pose estimation, tracking, and classification.

Training a machine learning model for segmentation tasks typically demands a large, annotated dataset, which is both time-intensive and prone to human error. To address this challenge, generalized large-scale models like SAM [8] can be leveraged to automatically generate segmentation masks, significantly reducing the annotation effort.

Although Gaussian techniques yield state-of-the-art (SOTA) results, existing infrastructure, such as 3D modeling and editing software, as well as platforms offering augmented reality solutions, primarily operate with meshes. Consequently, SuGaR [11] is employed for mesh extraction from 3D Gaussian splatting. Gaussian Splatting accelerates training for rendering intricate scenes but can result in disorganized Gaussians, complicating mesh creation. SuGaR addresses this by aligning Gaussians to the scene's surface, enhancing mesh generation through efficient Poisson reconstruction, which preserves details better than alternatives like Marching Cubes with Neural SDFs.

In 2021, AR-Shoe [12] was introduced, which leverages deep learning methodologies to superimpose a 3D shoe model onto an image of a foot. The model operates by taking foot images as inputs and producing keypoint heatmaps, part affinity fields maps (pafmaps), and segmentation maps of the feet. These outputs are utilized to derive the precise 6 degrees of freedom (6-DoF) pose for the feet, and overlay of 3D shoe which is then occluded in the input image.

Song et al. introduced VTONShoes [13], an advanced AR-based real-time virtual shoe try-on system that achieves precise 6-DoF pose estimation and realistic rendering using dense keypoints, joint keypoint localization, and silhouette segmentation. The system's smooth and stable performance at 25-45 FPS, coupled with the introduction of the Diverse-Shoes dataset, marks a significant advancement in real-time AR virtual try-on technology.

II. Methodology

A. Data Collection

In the 3D Gaussian Splatting Model, we achieve accurate modeling by over-fitting a single object. Consequently, a vast data-set isn't necessary. But, we do need data for segmenta- tion of shoes from the given images. For this purpose, we collected 101 videos, each lasting 30 seconds, from students at Pulchowk Campus. These videos were taken with different cameras, such as Poco X3, Samsung Galaxy S9, Redmi 12, and others.

From each video, we extracted 30 images by evenly dividing the video frames. These images were then used for shoe segmentation to generate 3D models. The segmentation process involved isolating the shoe from the background in each image. These segmented images formed the foundational data for our subsequent 3D modeling.

Fig. 1. Sample images from the dataset.

B. Background Masking

We used Meta's Segment Anything Model (SAM) [8] to generate segmentation maps from images, which provided accurate results. However, the model required over 6GB of GPU memory, making it costly to host on a server due to its high computational demands. To address this, we used SAM to create a dataset and then trained a smaller, more efficient model that significantly reduces computational requirements. The steps we followed are outlined below:

- Dataset Collection: We collected 101 videos of distinct shoes each of length 30 seconds. We collected 30 images from each video by sampling uniformly across the dura- tion of the video. This gave us a total of 3030 images.
- Annotation: We use an automated pipeline to annotate the images. First, we used a dataset [14] from Roboflow to train an YOLO model to detect shoes in images. Then, we used SAM to annotate the images, sending in the bounding box obtained from the YOLO model as prompt.
- Correction of improper annotations: The automated annotation pipeline did not provide correct annotation on all images which arose from inaccuracies in YOLO model while detecting shoes and inaccuracies in segmentation map from SAM. However, such instances were very few in number and hence, did not require much time to correct manually. We manually skimmed through all the annotations correcting the ones with errors to ensure the correctness of whole dataset. Furthermore, there were also some frames which did not have shoes. We removed such images.
- Training: For training, we split the data into train, validation and test set with 80%, 10% and 10% of the total data respectively. We wrote the split script to ensure that images from a single video does not end up in multiple splits which might bump the accuracy simply by overfitting the training samples. We trained the dataset on YOLO v8 and Unet model pre-trained on the COCO128 dataset. The model provided results which is decent enough for the task of 3D reconstruction. The final model size is 6.5 MB which is orders of magnitude smaller than the SAM model which is about 1.5GB.

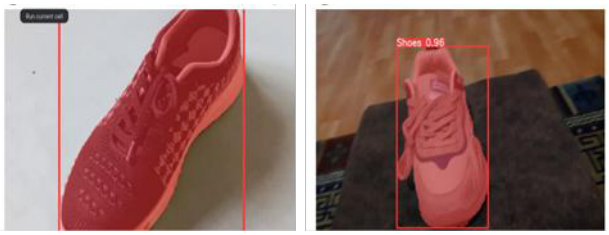

Fig. 2. Prediction on test set from the trained YOLO v8 model.

Fig. 3. Extracted mesh.

This significantly reduces the required computational resources and server costs for hosting the model.

C. Colmap

We used COLMAP, a popular tool for 3D reconstruction, to process our data. It helped us create accurate 3D models from input images by estimating camera poses and generating sparse and dense point clouds. The images were first aligned using COLMAP's feature-matching and structure-from-motion pipeline. After alignment, dense reconstruction was performed to generate detailed 3D points. The output was then used as input for the next stages of our project.

D. Gaussian Splatting Model

We used the implementation provided in the Gaussian Splatting repository with some adjustments. The pre-processing step included cropping each image to its maximum size along both dimensions. The model was then trained for 7000 iterations, which took approximately 10 minutes.

E. Mesh Extraction

We used the implementation from the Sugar repository with some adjustments. The model was trained for 9000 refinement iterations, which was the most time-consuming step. Extracting the mesh took approximately an hour. The final images from the undistortion phase had a black background, resulting in many black faces in the final mesh. To address this, we wrote a script to remove all black vertices and extract the largest connected component, ensuring a cleaner mesh.

Fig. 4. Results in AR.

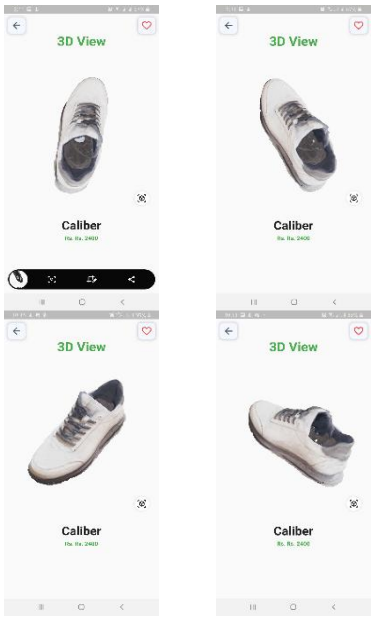

Fig. 5. Interactive 3D Viewer

F. Augmented Reality

To predict the pose of the user's feet in augmented reality (AR), we employed Lens Studio. The process of fitting a shoe onto the user's leg involves the following steps:

- We began by creating a 3D model using Gaussian Splat- ting, then combined the obj, mtl, and png files using Blender.
- Next, we imported the GLB files of both the left and right foot shoes into Lens Studio.

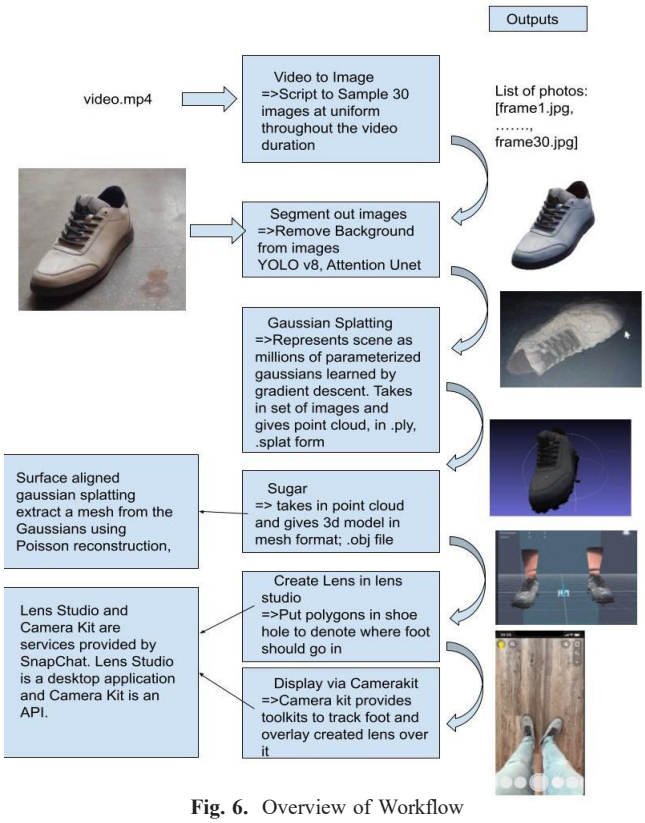

Fig. 6. Overview of Workflow

Using Lens Studio's leg detection template, we aligned the 3D model feet with the user's actual feet.

- To address occlusion, we placed transparent cylinders through the holes of the shoes.
- Finally, we published the lens in Lens Studio and integrated it into our Flutter app.

G. Mobile Application

We integrated individual pieces in a mobile application which allows user to enjoy AR experience in their mobile phones. We utilized flutter along with SnapAR package to develop the app.

H. Overall System Design

The pipeline comprises several integral components working in tandem to create a comprehensive and immersive AR experience. First, the Background Masking element eliminates image backgrounds, facilitating the generation of masked images fed into the model for 3D shoe modeling. Colmap generates a 3D point cloud or mesh representing reconstructed scenes. The Gaussian splatting model refines this representation by using multiple 3D Gaussian distributions to create a smooth point cloud depiction. Sugar converts the obtained point cloud to mesh format. The foot pose estimation model involves key point prediction, pose estimation, and segmentation for occlusion identification, ensuring a realistic representation of the shoe and leg interaction. We utilize Snapchat's Camera Kit to implement this functionality. The project culminates in a mobile application merging these components, enabling users to engage in AR.

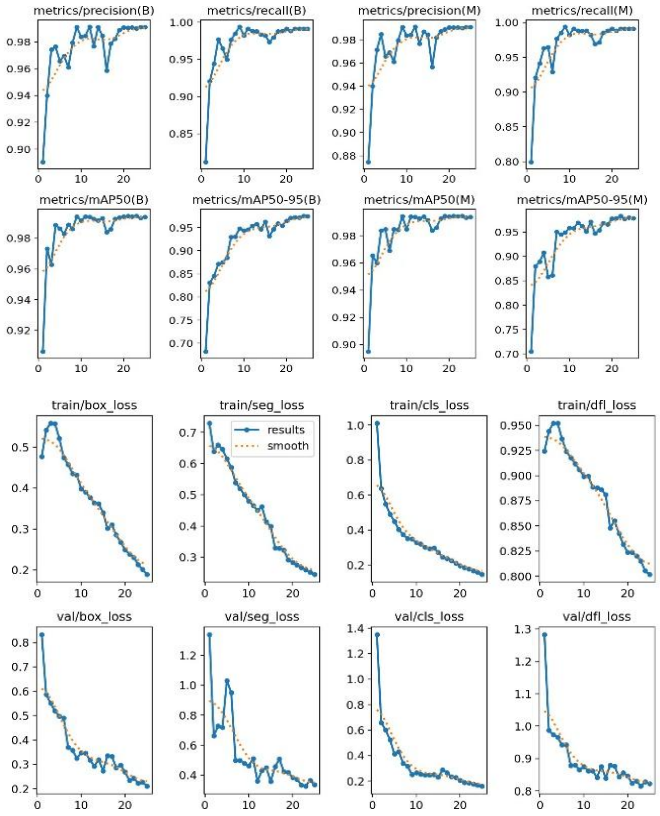

Fig. 7. Training Graph of YOLO V8n

TABLE I: COMPARISON OF SEGMENTATION MODELS (TEST SET)

Model	Parameters	Epochs	IOU (threshold = 0.5)
YOLOv8n		100	0.9494
YOLOv8m	27.3M	25	0.9577
Unet (Resnet Backbone)	32M	25	0.9548

I. Evaluation and Testing

The quality of the final augmented output largely depends on the quality of 3D model generated; we have to ensure that the initial output is as realistic as possible. We used peak signal-to-noise ratio to evaluate the performance of our method.

$$PSNR = 10log_{10}\left(\frac{MAX^2}{MSE}\right) \qquad (1)$$

where MAX indicates the maximum possible pixel value of the image or video (e.g., 255 for an 8-bit grayscale image). MSE is the mean squared error between the original and the reconstructed image. The mean squared error (MSE) is calculated as follows:

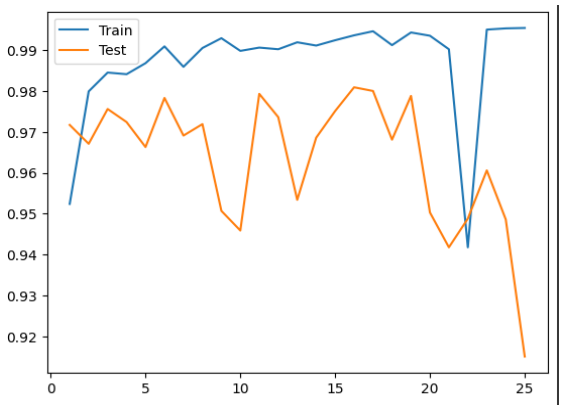

Fig. 8. Training Graph of Attention Unet (IoU).

$$MSE = \frac{1}{N \times N}\sum_{i=1}^{N}\sum_{j=1}^{M}\big(Original(i,j) - Reconstructed(i,j)\big)^2 \quad (2)$$

where N is the number of rows in the image or video and M is the number of columns in the image or video. For the segmentation task, we evaluate our models over IOU.

$$IOU = \frac{Area\ of\ Intersection}{Area\ of Union} = \frac{TP}{TP + FP + FN} \quad (3)$$

True Positive (TP): Instances where the model correctly predicts the presence of a positive class
False Positive (FP): Instances where the model incorrectly predicts the presence of a positive class
False Negative (FN): Instances where the model fails to predict the presence of a positive class
The quality of the final augmented output largely depends on the quality of 3D model generated; we have to ensure that the initial output is as realistic as possible. We used peak signal-to-noise ratio to evaluate the performance of our method.
The final qualitative evaluation of our application was done by testing the system on various input conditions. A robust system can handle various conditions of different foot poses and can realize a realistic AR effect in practical scenes.

III. Results and Discussion

The following are the results we obtained grouped by system components:

A. Background Masking

The data collection phase resulted in 3,000 images. Pre-processing was done as explained in the Implementation section. The following are the results we obtained after training different models. We trained the dataset on YOLO v8 and the Unet model pre-trained on the COCO128 dataset. The model provided results which is decent enough for the task of 3D reconstruction. The final model size is 6.5 MB which is orders of magnitude smaller than the SAM model which is about 1.5GB. This significantly reduces

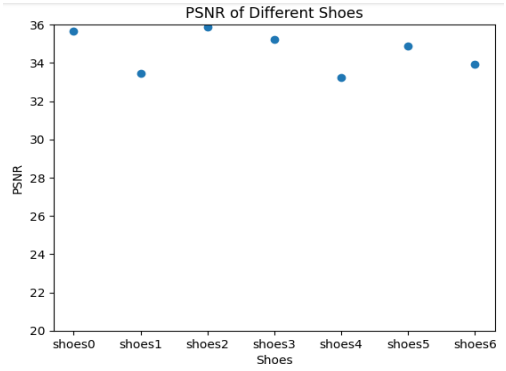

Fig. 9. PSNR of different shoes.

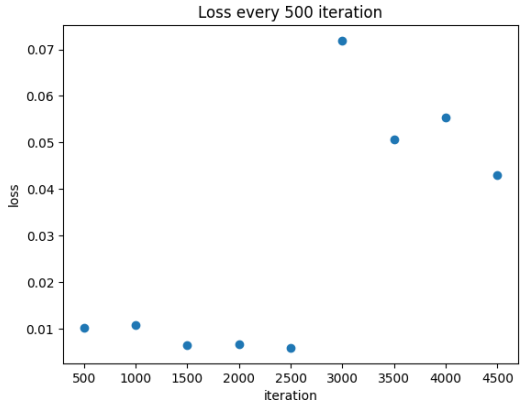

Fig. 10. Loss every 500 iterations.

the required computational resources and server costs for hosting the model. In all the models IoU spiked during the initial epoch and saturated quickly. The training loss kept on decreasing, but the training was stopped after validation loss started to saturate. Among the models, YOLOv8m performed the best on the test set. However, it should be noted that each model was trained for an hour. Results may vary if training is left to progress for more time.

B. Gaussian Splatting Model

We drew random samples from our dataset and evaluated the Gaussian Splatting model. We obtained an average PSNR of 34.

C. Mesh Extraction

The result of Gaussian splatting was used to extract the mesh using Sugar. Following is the trajectory of loss we obtained for a shoe: Initially, the loss decreased, but after some iterations, it abruptly increased. For optimal results, the best model should be saved.

V. Conclusion

This work presents a robust system for 3D shoe modeling and AR integration, demonstrating the potential of Gaussian Splatting for efficient and accurate 3D reconstruction. By addressing limitations in data processing and model complexity, the proposed framework achieves realistic rendering suitable for mobile applications. Future efforts will focus on enhancing real-time rendering capabilities, improving

occlusion handling, and extending the pipeline to other product categories for broader application in the fashion industry.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *CoRR*, vol. abs/2003.08934, 2020. [Online]. Available: https://arxiv.org/abs/2003.08934

[2] T. Mu¨ller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[3] B. Kerbl, G. Kopanas, T. Leimkuhler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Journals*, vol. 42, no. 4, pp. 1–14, 2023.

[4] J. L. Scho¨nberger and J.-M. Frahm, "Structure-from-motion revisited," *IEEE*, 2016.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," CoRR, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[6] O. Oktay, J. Schlemper, L. L. Folgoc, M. C. H. Lee, M. P. Heinrich, K. Misawa, K. Mori, S. G. McDonagh, N. Y. Hammerla, B. Kainz,B. Glocker, and D. Rueckert, "Attention u-net: Learning where to look for the pancreas," CoRR, vol. abs/1804.03999, 2018. [Online]. Available: http://arxiv.org/abs/1804.03999

[7] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, Wasserthal, G. Ko¨hler, T. Norajitra, S. J. Wirkert, and K. H. Maier-Hein, "nnu-net: Self-adapting framework for u-net-based medical image segmentation," CoRR, vol. abs/1809.10486, 2018. [Online]. Available: http://arxiv.org/abs/1809.10486

[8] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dolla´r, and R. Girshick, "Segment anything," 2023.

[9] X. Zhou, R. Girdhar, A. Joulin, P. Kra¨henbu¨hl, and Misra, "Detecting twenty-thousand classes using image-level supervision," CoRR, vol. abs/2201.02605, 2022. [Online]. Available: https://arxiv.org/abs/2201.02605Ultralytics. (2024) Yolov8. [Online]. Available: https://github.com/ultralytics/ultralytics

[10] A. Gue´don and V. Lepetit, "Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering," arXiv preprint arXiv:2311.12775, 2023.

[11] S. An, G. Che, J. Guo, H. Zhu, J. Ye, F. Zhou, Z. Zhu, D. Wei, A. Liu, and W. Zhang, "Arshoe: Real-time augmented reality shoe try-on system on smartphones," CoRR, vol. abs/2108.10515, 2021. [Online]. Available: https://arxiv.org/abs/2108.10515

[12] W. Song, Y. Gong, and Y. Wang, "Vtonshoes: Virtual try-on of shoes in augmented reality on a mobile device," in 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2022, pp. 234–242. Boston university cdr2g - shoes segmentation. [Online]. Available: https://universe.roboflow.com/boston-university-cdr2g/shoes-seg