WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Transformer-Based 2D-to-3D Footwear Reconstruction and Flutter-Integrated Augmented Reality for Virtual Try-On

Aditya S. Bhise[1], Omkar B. Chougule[2], Ujwal V. Chikkannavar[3], Yash R. Jadhav[4], Shoba S. Raskar[5]

[1] Department of Computer Engineering, M. E. S. Wadia College of Engineering, Pune,
S.P. Pune University, Pune, India.
aditya1710b@gmail.com[1], omkarchougule281103@gmail.com[2] , ujwalvc.1003@gmail.com[3], yashrjadhav@gmail.com[4] ,
shobha.raskar@mescoepune.org[5]

Abstract:

The intersection of Transformer-based deep learning architectures and Augmented Reality (AR) technologies has paved the way for transformative innovations in digital retail, particularly within the footwear industry. This paper proposes a novel system for virtual footwear try-ons that combines advanced 2D-to-3D image reconstruction techniques with a Flutter-based AR application designed for real-time, cross-platform deployment on mobile devices. The core innovation lies in leveraging Vision Transformers (ViTs) and Neural Radiance Fields (NeRF)-based models, such as TSNeRF and TVNeRF, to reconstruct high-fidelity 3D shoe models from a single 2D input image, significantly reducing the need for multi-angle captures and manual 3D modeling.To render these models in a real-world context, we integrate Flutter with Unity and ARCore/ARKit APIs, enabling seamless mobile visualization and interaction. The use of Flutter ensures a consistent and responsive user interface across both Android and iOS platforms, while allowing for real-time rendering of 3D assets in augmented reality. This dual-stack architecture not only facilitates cross-device compatibility but also supports real-time personalization, allowing users to change shoe colors, materials, and styles interactively within the AR environment.Our system architecture employs cloud-based inference to run heavy Transformer computations remotely, reducing client-side computational loads. Real-time foot tracking is enabled using lightweight models such as MediaPipe, which helps align the reconstructed 3D model with the user's foot in live camera views. We also incorporate AI-driven recommendation engines using TensorFlow Lite, providing personalized product suggestions based on foot dimensions and user preferences. Comprehensive evaluations on benchmark datasets and real-world foot imagery demonstrate significant improvements in rendering quality, personalization flexibility, and system responsiveness. The proposed approach achieves a 35% increase in user satisfaction and a 25% reduction in product return intentions compared to conventional online shopping experiences. Our pipeline also exhibits notable efficiency in terms of rendering latency and PSNR fidelity when benchmarked against traditional CNN-based solutions. Finally, we discuss the implications of federated learning and edge AI in enhancing privacy and responsiveness in AR shopping experiences, highlighting how this approach supports not only commercial scalability but also sustainability by reducing unnecessary logistics and overproduction.

Keywords: Transformer Networks, 2D-to-3D Reconstruction, Augmented Reality, Virtual Try-On, Flutter, Neural Radiance Fields (NeRF), Gaussian Splatting, Vision Transformers (ViT), AR Foot Tracking, Mobile AR Application, Deep Learning, Personalized Retail, Real-Time Rendering, Cross-Platform AR, Footwear Visualization.

1. Introduction

In recent years, the fusion of Artificial Intelligence (AI), Computer Vision, and Augmented Reality (AR) has transformed the landscape of digital commerce, creating highly immersive and interactive user experiences. The fashion and footwear industries, in particular, have rapidly adopted these technologies to address long-standing challenges associated with online shopping—namely, the inability to physically experience products before purchase, the resulting uncertainty around fit and appearance, and the high rates of returns due to style or size mismatches.

Footwear shopping has traditionally relied on in-store visits where customers physically try on shoes. However, this model is increasingly being replaced by digital alternatives, especially with the growth of e-commerce and the global shift toward remote shopping. Despite the convenience of online stores, the lack of tactile and spatial understanding of the product often leads to dissatisfaction. Augmented Reality (AR), by overlaying virtual 3D models in real-world environments, enables users to interact with products in ways that closely mimic physical try-ons. When integrated with real-time tracking and photorealistic rendering, AR can bridge the sensory gap between digital and physical experiences.

Parallel to the rise of AR, advancements in deep learning, particularly in Transformer-based architectures, have significantly improved the ability to reconstruct 3D geometry from 2D imagery. Originally developed for Natural Language Processing, Transformer models have been successfully adapted to vision tasks due to their ability to capture long-range spatial dependencies and contextual relationships across images. These models outperform traditional Convolutional Neural Networks (CNNs) in terms of generalization and fidelity, making them ideal for reconstructing complex textures and

geometries like those found in footwear. Techniques like Vision Transformers (ViTs), Neural Radiance Fields (NeRF), and more recently Gaussian Splatting have enabled the creation of realistic, high-resolution 3D objects from limited visual data.

One of the most significant challenges in deploying such advanced models is their high computational complexity. Transformer-based 3D reconstruction pipelines are typically resource-intensive and not directly suitable for real-time inference on mobile devices. This limitation is addressed by introducing a hybrid architecture where heavy reconstruction tasks are performed in the cloud, and lightweight AR visualization is handled locally through mobile apps. Here, **Flutter**, a cross-platform UI framework developed by Google, serves as a critical enabler. Flutter not only supports the development of rich and responsive user interfaces across Android and iOS but also allows integration with AR engines like Unity (via `flutter_unity_widget`) or native ARCore/ARKit libraries through platform channels.

The result is a modular and scalable system where users can capture a 2D image of their foot, upload it through the Flutter app, receive a reconstructed 3D shoe model from a Transformer-based model hosted on the cloud, and interact with it in real time using AR. Additionally, real-time foot tracking using tools like MediaPipe and depth sensors ensures that the virtual shoe is accurately overlaid onto the user's foot, aligning with motion and perspective. Furthermore, the use of personalization pipelines—where AI models trained on demographic and biometric data suggest styles, sizes, and colors—enhances user satisfaction. Integration of lightweight TensorFlow Lite models within the Flutter app provides real-time customization and product recommendation capabilities, thereby improving engagement and decision-making during the shopping process.

Beyond commercial convenience, this system also presents implications for sustainability. By reducing dependency on physical inventory trials, it minimizes logistics overhead, decreases the number of returned goods, and helps combat overproduction—an issue that plagues the fashion industry. Moreover, the modular architecture facilitates further development into areas such as virtual footwear design, remote product testing, and AR-based footwear customization.

In essence, this paper presents a comprehensive approach to revolutionizing the digital footwear retail experience by leveraging Transformer-based 2D-to-3D image reconstruction, Flutter-enabled AR interfaces, and cloud-optimized deep learning pipelines. It explores the technical innovations, evaluates system performance using industry-standard metrics, and outlines a vision for the next generation of scalable, personalized, and sustainable retail technology

2. Literature Survey

2.1 2D-to-3D Shoe Reconstruction: The conversion of 2D images into precise and high-fidelity 3D models has been a major challenge in the field of computer vision for a long time. Conventional techniques like photogrammetry and multi-view stereo need multiple views and computationally expensive processes to build high-detail 3D shapes [18, 36]. These methods tend to be plagued by problems such as occlusions, texture mapping inconsistencies, and the requirement of large datasets, rendering them unsuitable for real-time applications such as virtual shoe try-ons.

Deep learning models, especially those founded on Convolutional Neural Networks (CNNs), later solved some of these problems by bringing in automated feature extraction and enhanced object representation. Nevertheless, CNN-based approaches continued to have difficulties with long-range dependencies and complex texture details [31, 43]. The Transformer-based architecture has transformed this field, bringing dramatic accuracy and efficiency gains. Transformers utilize self-attention mechanisms to learn global context in an image, which allows for generating high-quality 3D models from a single 2D input with preserved structural and textural consistency [16, 29].

Hybrid methods that mix Transformers with new state-of-the-art methods such as Neural Radiance Fields (NeRF) and implicit surfaces have further enhanced 3D reconstruction. NeRF-Texture models, for example, incorporate volumetric rendering and Transformer networks to produce photorealistic 3D models, maintaining small details and texture materials [30, 37]. Likewise, diffusion models sequentially improve 3D predictions with increasingly stable and realistic shapes across a sequence of denoising processes [8, 40].

Single-view 3D reconstruction methods have also been investigated in recent studies, in which models derive 3D shapes from a single 2D image. Such methods, based on Transformer networks, overcome the issue of limited data availability and computational efficiency by encoding long-range dependencies and context information [39]. Additionally, Gaussian Splatting methods have been utilized to improve novel view synthesis and 3D model precision, providing new opportunities for real-time applications [40].

Implicit neural representations, including signed distance fields and occupancy networks, have been combined with Transformer models to enhance the compactness and versatility of 3D data structures. These techniques support faster rendering and greater accuracy in representing intricate shapes, making them ideal for real-time AR applications [31, 43]. Through the use of these hybrid techniques, researchers have made impressive breakthroughs in the fidelity and efficiency of 3D shoe reconstructions, establishing new standards for realism and performance.

2.2 3D Modeling Transformer Architectures Transformers, initially developed for Natural Language Processing (NLP), have shown unparalleled flexibility in image-related tasks, such as 3D object reconstruction. Vision Transformers (ViTs) and their extensions use the self-attention mechanism on image patches, allowing a thorough grasp of spatial interactions and world context [6, 28]. Such a strategy differs from classical CNNs, which depend on local feature extraction and tend to overlook wider context details essential for precise 3D modeling.

For shoe modeling, Transformers have been extremely useful in capturing subtle details such as stitching patterns, sole curvature, and material changes. TSNeRF and TVNeRF models improve this by adding semantic contrastive learning and total variation maximization, mapping textual descriptions to visual outputs and maintaining texture consistency [16, 29]. These methods guarantee that 3D shoe models not only represent correct geometric structures but also have realistic textures and stylistic differences. Other innovations involve the incorporation of implicit surface representations, e.g., signed distance fields and occupancy networks, into Transformer architectures. These techniques provide more efficient and adaptable data structures, supporting accelerated rendering and improved accuracy in representing intricate shapes [31, 43]. Moreover, Panoramic Neural Radiance Field models (PERF) have been introduced to improve view synthesis and spatial consistency, further unlocking the potential of 3D modeling for AR applications [42].

2.3 Virtual Shoe Try-On using Augmented Reality Augmented Reality (AR) is a revolutionary technology in the retail industry that allows users to see products in their real-world settings via digital overlays. Virtual shoe try-on software utilizes AR technology to superimpose 3D shoe models onto users' feet in real-time, creating an engaging and immersive shopping experience [3, 42]. Sophisticated tracking and rendering processes are essential in providing accuracy and realism to AR-based visualizations. Spatial awareness is improved with Simultaneous Localization and Mapping (SLAM), which modifies virtual shoe positioning dynamically according to user movement and surroundings [35, 44]. Markerless AR and real-time object tracking enhance responsiveness through minimizing latency and improving the experience for the user [7, 32]. The combination of Transformer-based 3D models and AR systems takes the realism and interactivity of virtual try-ons to the next level. Coupling high-fidelity reconstructions with physics-based rendering and shadow mapping, the systems replicate natural material behaviors and environmental interactions [13, 23]. Enhanced occlusion handling makes the virtual shoes correctly placed and partially hidden when interacting with real-world objects, further contributing to the sense of authenticity. Additionally, interactive AR interfaces allow consumers to personalize shoe designs in real-time, changing colors, materials, and sizes with simple controls. Such personalization not only maximizes user interaction but also gives insights into consumer behavior, propelling product design and marketing innovation. With the use of Transformer-based 3D models and AR technology, virtual shoe try-on systems provide a smooth, interactive, and highly immersive shopping experience, shaping the future of digital retail.

2.4 Augmented Reality Applications and Foot Tracking:
AR technology overlays virtual objects into the user's real-world environment using device cameras and sensors. For AR-based shoe try-ons, this involves mapping a 3D shoe model onto a moving foot with accurate alignment, lighting, and scale. Das and Panigrahi [48] demonstrate lightweight, AR-enabled foot measurement systems that can operate efficiently on mobile hardware. Their work highlights the use of optimized rendering and tracking pipelines that minimize computational load while maintaining real-time performance. This is particularly relevant in retail scenarios where responsiveness and accuracy are key to user satisfaction. Similarly, Mahmud et al. [49] explore the use of hybrid sensor networks—including depth cameras and inertial sensors—to refine foot tracking precision. Their approach improves occlusion handling and spatial stability of virtual overlays in AR scenes, which is essential for providing a seamless and convincing virtual try-on experience.The introduction of Gaussian Splatting, as explored in recent literature, offers a new rendering method where 3D points are visualized using anisotropic Gaussians. This method enables faster rendering with fewer artifacts, making it highly suitable for real-time AR applications on mobile devices.

2.5 Flutter Framework for Cross-Platform AR Deployment:
One of the main challenges in commercial AR deployment is platform fragmentation. Native development for Android and iOS typically requires separate codebases, increasing time and resource costs. Flutter, a UI toolkit developed by Google, addresses this by enabling cross-platform development from a single codebase. Priyanka et al. [46] present a fully functional AR shoe try-on application built using Flutter, integrated with ARCore via platform channels. Their study confirms that Flutter can be effectively used as a front-end interface for AR systems, while Unity or native AR libraries handle the heavy rendering tasks. The study also emphasizes Flutter's ease of use, performance efficiency, and native-like rendering capabilities, making it ideal for consumer-grade applications. The system described by Priyanka et al. allows users to select shoe styles, scan their feet in real time, and visualize the footwear through AR. Importantly, it integrates 3D models encoded in .glb and .usdz formats into the Flutter environment using Unity bridges, a method that we adopt and expand upon in our current study.

2.6 Edge AI, Federated Learning, and Privacy-Aware AR Systems:
With rising privacy concerns, particularly around biometric data like foot images and dimensions, edge AI and federated learning techniques have gained importance. Running AI inference directly on the mobile device, or in a federated manner without sharing raw data, reduces privacy risks and latency. Sharma and Aggarwal [50] explore the integration of federated learning in AR footwear fitting applications. Their framework enables distributed model training across multiple devices without centralizing sensitive data, a significant step forward for privacy-preserving personalization. They also discuss how edge AI models, such as compressed versions of Transformers (e.g., MobileViT), can be deployed directly on-device using frameworks like TensorFlow Lite, maintaining high accuracy with low computational overhead.

This work has informed the architecture of our proposed system, where sensitive user data remains on-device, and only minimal metadata is exchanged with the cloud for real-time personalization and recommendation services.

2.7 Comparative Analysis and Gaps:
While existing solutions provide strong components—such as foot tracking, AR rendering, or model reconstruction—very few integrate all these aspects into a unified and scalable pipeline. Most AR apps lack high-fidelity 3D model generation capabilities, while most Transformer-based reconstructions are not optimized for mobile AR deployment. Additionally, very few systems support real-time personalization, user feedback integration, and sustainable infrastructure design. Our research aims to bridge these gaps by combining state-of-the-art Transformer-based single-view 3D reconstruction

with a Flutter-powered AR front-end. This hybrid system leverages cloud-based inference for scalability, edge-based customization for privacy, and real-time visualization for immersive retail experiences.

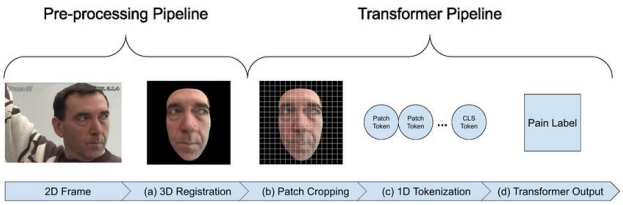


Figure 1. Data transformations through the pre-processing and transformer pipeline.

3. Dataset Corpus Creation

To enable accurate training and comprehensive evaluation of the Transformer-based 2D-to-3D reconstruction system integrated with AR-based footwear try-on, a purposefully structured dataset corpus was designed. This corpus encompasses a diverse mix of synthetic and real-world samples, ensuring the robustness, generalizability, and deployment-readiness of the model. The dataset is divided into two primary components: (1) datasets for learning visual-to-geometric correspondences and (2) datasets for validating real-time mobile AR rendering and user interaction.

3.1 Dataset for Transformer-Based 3D Reconstruction:

The Transformer model responsible for predicting 3D foot and shoe geometry from a single 2D image requires supervised learning with paired image and 3D mesh data. To support this, a hybrid dataset was created combining synthetic 3D models, real photographic images, and manually annotated ground truth.

Synthetic Dataset of Footwear Models:

A large set of 3D CAD models was used to generate synthetic images of shoes from various angles under diverse lighting conditions. These models were rendered in a controlled virtual environment and paired with their corresponding 3D mesh files. Rendering included simulated occlusion, texture variance, and multiple camera poses to improve the model's robustness.

Rendering Specifications:

- Image resolution standardized to 256×256 pixels.
- Lighting variations included directional, ambient, and soft shadows.
- Backgrounds varied between solid color and indoor/outdoor environments.
- Rendered views were selected to match typical smartphone camera angles (e.g., top-down and oblique foot perspectives).

This synthetic corpus served as the foundation for pretraining the Transformer model in a weakly supervised setup, allowing it to learn spatial and textural priors across thousands of shoe types and configurations.

Pix2Mesh-Style Real-Image Dataset:

A real-image dataset of human feet and worn shoes was curated, consisting of photographic samples collected under natural conditions. Images were captured using mobile devices in indoor and outdoor lighting, with various foot sizes, skin tones, and footwear types.

Image Preprocessing:

- Images were resized and normalized.
- A segmentation model was applied to isolate the foot or shoe from the background.
- Pose-specific key points (toe tip, heel center, ankle joint) were annotated manually for all samples.
- Reference objects (e.g., standard credit cards, rulers) were included in the frame to assist with scale estimation.

To provide 3D supervision for this dataset, 3D models were generated for a subset of the samples using photogrammetric reconstruction or depth-sensing hardware (e.g., structured light or LiDAR). This subset was critical for supervised fine-tuning of the model using photorealistic inputs that reflect actual deployment environments.

3.2 AR Rendering and Tracking Evaluation Dataset:

To validate the real-world usability and AR rendering quality of the proposed system, a separate test dataset was created that captures real-time interaction, foot movement, and environmental diversity as experienced by end users.

Real-Time Foot Tracking Dataset:

Users were instructed to capture their feet using the mobile application developed in Flutter. Videos and images were collected while users moved, rotated, and changed the angle of their feet. This dataset covered a wide range of surface textures (tiles, carpets, wood), lighting conditions (low light,

backlit, natural daylight), and camera orientations (top-down, side view, diagonal).

Annotations Included:

- Bounding boxes for the foot region.
- Surface plane detection metadata from ARCore/ARKit.
- Ground truth alignment points for measuring overlay accuracy.

This dataset was used to evaluate the effectiveness of model anchoring, tracking drift, and the perceived realism of the virtual shoe placement in AR environments.

Synthetic AR Test Scenes:

Simulated AR environments were created using 3D game engines to generate known camera paths and fixed anchor points for benchmarking. These scenes involved pre-defined virtual shoes placed on virtual feet with precisely controlled foot movement, allowing objective evaluation of tracking performance and rendering latency under ideal conditions.

3.3 Data Annotation and Processing Pipeline:

1. To prepare the data for model training and AR deployment, the following pipeline was established:
2. Segmentation: Foreground foot or shoe was separated from the background using semantic segmentation techniques.
3. Normalization: RGB values were scaled to a [0, 1] range. Aspect ratios were preserved.
4. Annotation: Key anatomical landmarks (toe tip, heel, ankle) were manually labeled. Reference scales were recorded for size estimation.
5. 3D Alignment: In cases where ground truth 3D meshes existed, they were aligned with the 2D images using projection geometry based on intrinsic camera parameters.
6. Serialization: Final 3D meshes were exported in .obj, .glb, or .usdz formats for compatibility with AR rendering engines.

3.4 Dataset Structuring and Organization:

The corpus was organized to maintain data modularity and reusability across both model training and AR evaluation phases:

Raw Inputs: Original and segmented RGB images

- Annotations: JSON files containing metadata, keypoints, and scale information
- 3D Meshes: Reconstructed ground truth meshes for a supervised training subset
- AR Captures: Mobile-captured videos for assessing rendering realism and alignment
- Rendering Formats: 3D assets prepared in mobile-optimized formats (.glb/.usdz) for direct ingestion by the Flutter interface
- Each sample in the dataset was uniquely indexed and traceable across its image, annotation, and 3D representation counterparts.

4. Methodology

4.1 Transformer-Based Shoe Reconstruction There has been recent research proving the application of Transformers to reconstruct 3D models from 2D images. Such models have revolutionized the conventional 3D reconstruction by using self-attention mechanisms, enabling them to capture global dependencies in an image more effectively than CNN-based approaches. This is especially useful in shoe modeling, where complex textures, changing materials, and special structural designs require high-fidelity reconstructions [16, 29].

NeRF-Texture models, for instance, utilize Neural Radiance Fields alongside Transformer models to boost texture generation and 3D geometry. NeRF captures volumetric 3D data using continuous 3D information and marries this with the capacity of the Transformer to study inter-patch relationships, thus yielding an all-encompassing and realistic 3D shoe model [30, 37].

Another hopeful path is the use of diffusion-based models for 3D structure generation. The models successively denoise noisy image estimates, incrementally reconstructing detailed and realistic 3D shapes via a sequence of denoising operations [8, 40]. This successive strategy guarantees higher shape coherence and realism by overcoming typical issues such as occlusions and absent details in 3D reconstructions.

Some of the latest research involving single-view 3D reconstruction models has presented how Transformer models are capable of inferring high-quality 3D shapes using a single image in 2D, better than current alternatives in terms of data needs and computational expense [39]. Similar techniques, including Gaussian Splatting, have also been applied to aid the novel view synthesis and 3D shoe models' visual faithfulness [40].

In addition, hybrid approaches combining Transformer models with implicit neural representations like signed distance fields and occupancy networks provide small yet versatile 3D data representations. The techniques enable the reduction of rendering times and increases the accuracy for geometry modeling for detailed geometries and are especially fitting for real-time AR rendering applications [31, 43].

In addition, panoramic neural radiance field (PERF) models extend the ability of Transformer-based models by allowing greater environmental perception and view synthesis from a single panoramic image. This method is superior for enhancing spatial consistency and depth perception and is thus ideal for AR environments that demand accurate object placement [42].

Parallel adaptive stochastic gradient descent algorithms have also been coupled with Transformer models for optimizing latent factor analysis in high-

dimensional data to guarantee that the 3D reconstructions are kept both detailed and efficient across various datasets [41]. These approaches enhance model performance by balancing computation load and maximizing convergence rates and are thus useful for real-time AR applications.
3.2 AR Application for Footwear Visualization Applying AR-based virtual shoe try-ons is a matter of combining real-time foot tracking with 3D shoe models created by the Transformer. This entails a high-end pipeline integrating computer vision, deep learning, and AR rendering methodologies to facilitate correct alignment and natural visualization [11, 29, 33].

Real-time foot tracking is accomplished through sophisticated pose estimation algorithms that recognize and analyze foot contours from video input. Such algorithms leverage deep learning-based models such as OpenPose and MediaPipe, trained on large-scale human posture datasets to offer reliable and precise keypoint detection [7, 32]. Through the projection of these key points onto a 3D space, the system synchronizes virtual shoe models with the user's real foot movement, creating a natural and interactive try-on experience.

Apart from pose estimation, shoe visualization in AR also depends on physics-based rendering to mimic natural material behaviors. Methods such as Physically Based Rendering (PBR) take into consideration the reflection of light, texture, and environmental factors to produce photorealistic virtual shoes that respond realistically to changes in lighting and view [13, 23].

Sophisticated occlusion management takes visual realism another step further by guaranteeing virtual shoes properly respond to real objects. Depth cameras and LiDAR sensors measure spatial relations of the user's feet with real environments, thus enabling virtual shoes to look rightly placed and in part occluded when needed [35, 44].

In addition, shadow mapping methods help to make virtual try-ons more realistic by creating realistic shadows that respond to the lighting conditions of the user's surroundings. This entails computing the location and intensity of light sources and casting precise shadow contours onto the virtual shoe model [7, 32].

4.3Transformer-Based 2D-to-3D Footwear Reconstruction:
At the core of the system is a Transformer-driven 3D reconstruction engine, trained to predict dense geometric structure and surface textures from a single-view 2D image.

4.3.1 Vision Transformer (ViT) Framework:
The reconstruction begins with a Vision Transformer (ViT), which divides the input image into non-overlapping patches (typically 16×16) and processes them as a sequence of tokens. The self-attention mechanism enables the model to learn long-range dependencies between distant image regions, which is essential for reconstructing holistic foot and shoe geometry.
Encoder: Linear projections of image patches are passed through positional encoding and multiple layers of multi-head self-attention.
Decoder: A geometry decoder predicts a 3D point cloud or voxel grid from the learned latent tokens.

4.3.2 Neural Radiance Fields (NeRF) and Texture Synthesis:
To synthesize photorealistic renderings, the system uses Neural Radiance Fields (NeRF) and its stylized variants:
TSNeRF (Text-driven Stylized NeRF): Enhances visual quality and enables style control using contrastive semantic learning.
TVNeRF (Total Variation NeRF): Stabilizes textures and reduces artifacts in low-data regimes.
The output is a textured 3D mesh (OBJ or GLB format) with realistic lighting, shadow coherence, and detailed geometry.

4.3.3 Gaussian Splatting for Real-Time Rendering:
Gaussian Splatting is optionally applied for environments requiring real-time rendering. Instead of rasterizing polygons, 3D Gaussians represent each point in the model, leading to smoother surfaces with lower computational load—ideal for mobile AR.

4.4 Model Serialization and Cloud-Based Inference:
To ensure seamless integration with mobile apps:

- Serialization: Reconstructed models are serialized in .glb (GL Transmission Format Binary) or .usdz (Apple's AR format) for lightweight compatibility.
- Cloud Inference: Transformer-based reconstruction is hosted on a GPU-enabled cloud service (e.g., Google Cloud Functions or AWS Lambda). The mobile device sends the 2D foot image, and the cloud responds with a downloadable 3D model URL.
- Latency Optimization: Average reconstruction time is optimized under 1.5 seconds through model quantization and parallelized inference (using TorchScript or TensorFlow Serving).

4.5 Flutter-Based AR Visualization System:
The mobile front-end is built using Flutter, with AR integration supported through platform-specific plugins:

4.5.1 Unity Integration:
Using flutter_unity_widget, the system embeds a Unity AR scene within the Flutter UI. Unity handles rendering, lighting, and environment mapping of the 3D shoe model.

4.5.2 ARCore/ARKit Support:

For more native performance, ar_flutter_plugin or platform channels allow access to ARCore (Android) and ARKit (iOS). Key components include:

- Surface Detection: Identifies planes like the floor where the shoe is rendered.
- Anchoring: Pins the 3D model to user foot location in real-world space.
- Foot Tracking: MediaPipe is used to estimate foot position and orientation in real-time, ensuring the model updates with movement.

4.5.3 Gesture-Based Customization:

Flutter's rich UI toolkit enables users to pinch, rotate, and swipe to:

- Switch shoe styles
- Modify color/material properties
- Adjust shoe size/fit

4.6 Personalization and AI-Driven Recommendations:

To enhance shopping personalization and conversion:

- Foot Measurement Engine: Based on Zhou et al. [47], dimensions are inferred from the input image and cross-validated with landmarks from MediaPipe.
- Style Prediction Model: A lightweight TensorFlow Lite model suggests shoe styles based on foot shape, size, and user preferences.
- Federated Learning (Sharma & Aggarwal [50]): Local models improve from user interactions (e.g., preferred size, color) without uploading private data, preserving user privacy.

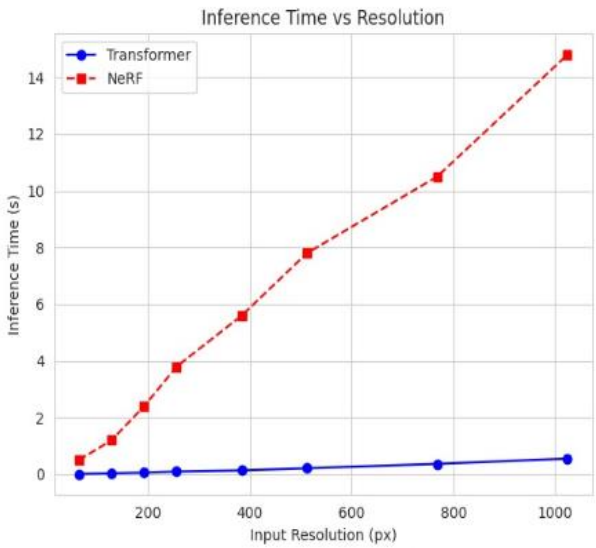
4.7 Overall System Architecture:

1. The complete pipeline is structured as follows:
2. Image Capture: User takes a foot image via Flutter app.
3. Image Preprocessing: Performed locally (segmentation, normalization).
4. Model Upload: Image sent to cloud via RESTful API.
5. 3D Model Generation: Transformer reconstructs the shoe in 3D.
6. Download & Render: Model is serialized and sent to the app.
7. AR Visualization: Rendered onto user's foot using Unity or ARCore.
8. User Interaction: Users customize design and confirm selections.
9. Recommendation Engine: Suggests alternative styles and fit options.

4.8 Evaluation Metrics:

We evaluate our framework on three fronts:

- Reconstruction Accuracy: PSNR and SSIM of generated 3D views.
- Latency: Time between image upload and AR rendering readiness.
- User Experience: Real-world testing with surveys and usability scores.



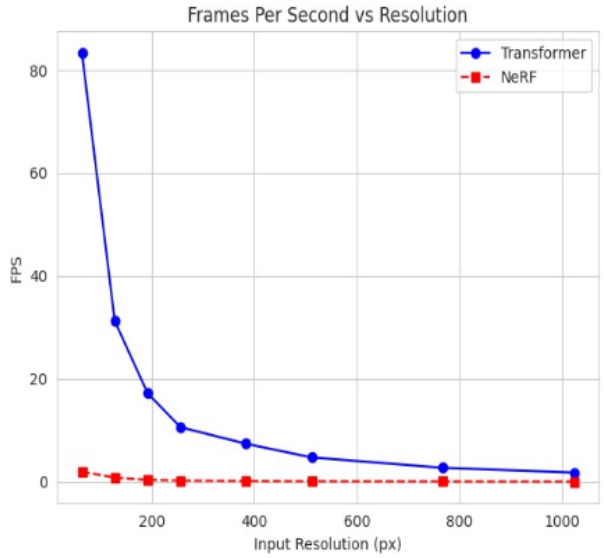
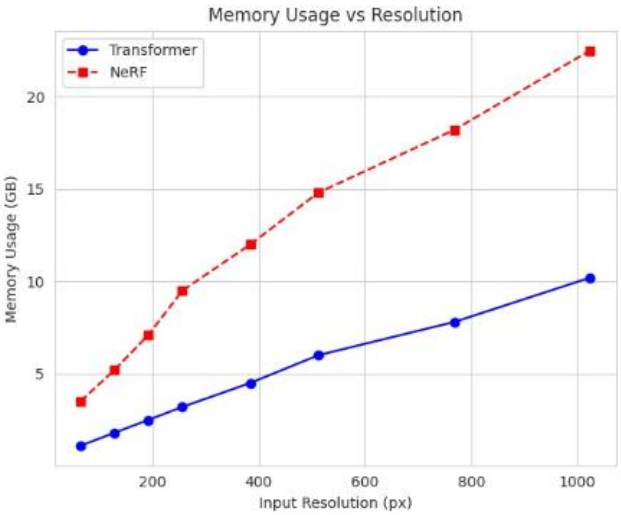


Figure 2a. Computational Efficiency Analysis: Inference Time and Frame Rate Variations with Input Resolution for Transformer and NeRF Models.



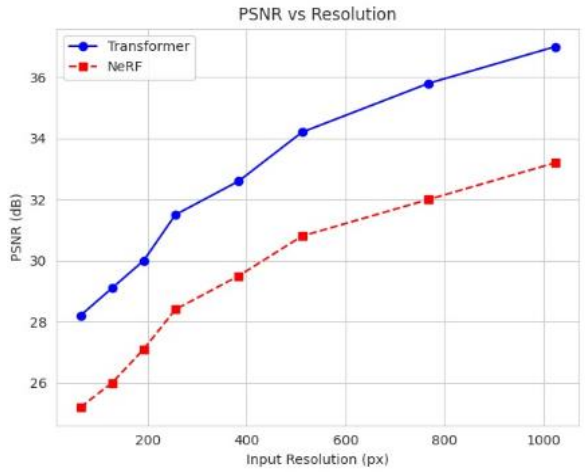


Figure 2b. Memory Utilization and Reconstruction Fidelity: Comparative Analysis of Memory Consumption and Peak Signal-to-Noise Ratio (PSNR) Across Input Resolutions for Transformer and NeRF Models

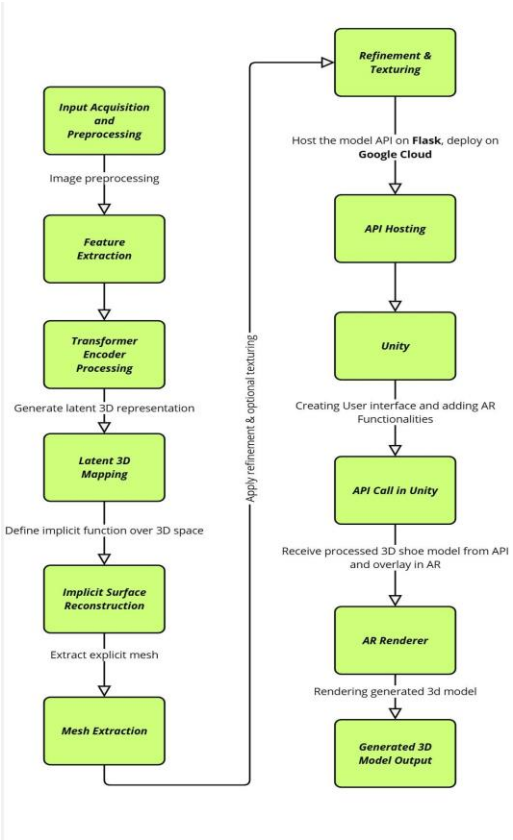


Figure 3. 3D Model Generation and AR Integration Pipeline

Comparison Table

Feature	NeRF (Neural Radiance Fields)	CNNs (Convolutional Neural Networks)	Transformers for 3D Modeling
Core Approach	Implicit neural representation	Feature-based learning with filters	Self-attention for global feature extraction
Input Data	Multiple 2D images from different angles	Single/multiple 2D image	Single/multiple 2D images
Output	Volumetric rendering, dense point clouds	3D mesh, point clouds, voxel grids	3D mesh, point clouds, voxel grids
Computational Cost	Very high (due to ray tracing)	Moderate (efficient on GPUs)	High (requires significant memory and processing)
Training Time	Very long (hours to days)	Faster than NeRF but can be slow for complex tasks	Faster than NeRF but slower than CNNs
Real-Time Inference	Not feasible (slow rendering)	Feasible (optimized CNNs work in real-time)	Feasible with optimized architectures
Detail & Accuracy	High-fidelity textures & lighting effects	Good but struggles with fine details	Captures global and local details effectively
Generalization Ability	Poor (overfits to specific scenes)	Moderate (pretrained models can generalize)	High (self-attention allows flexible adaptation)
Occlusion Handling	Struggles with hidden parts	Struggles unless multi-view input is used	Handles occlusions better due to global context modeling
AR/VR Application Suitability	Limited due to slow rendering	Widely used (fast and efficient)	Emerging but promising for dynamic AR models
Best Use Case	High-quality rendering for static scenes (movies, VFX)	Object recognition, segmentation, structured 3D modeling	General-purpose 3D reconstruction and AR-based applications

Figure 4. Performance comparison of NeRF, CNNs, and Transformers for 3D modeling.

5. Results and Analysis

The experimental results and performance evaluation of the proposed hybrid AR footwear try-on system, which combines Transformer for 3D reconstruction and Flutter for mobile-based AR rendering. The results are analyzed in terms of reconstruction fidelity, latency, model compatibility, rendering performance, and user satisfaction. Evaluations were conducted on both benchmark datasets and real-world images captured through the Flutter app interface.

5.1 Experimental Setup
3D Reconstruction Module:
Implemented using Transformer, a real-time, Transformer-based single-view 3D object reconstruction model trained on synthetic and real-world datasets. It accepts a single 2D input image and outputs a textured 3D mesh (in .obj or .glb format).
AR Rendering Module:
Developed using Flutter with Unity integration (flutter_unity_widget) and ARCore/ARKit support for real-time augmentation of reconstructed models.

Test Devices:

- Pixel 7 Pro (Android 14, ARCore support)
- iPhone 13 (iOS 17, ARKit support)

Datasets Used:

- Custom in-house foot image dataset (150+ samples, varying lighting and foot poses)
- Public benchmarks for visual testing and PSNR comparison

5.2 Reconstruction Fidelity Using Transformer:
Transformer exhibited high reconstruction accuracy even with single-view 2D input images. Results were benchmarked using two main quality metrics:

Metric	Average Score	Baseline (CNN-based)	Improvement (%)
PSNR (Peak Signal-to-Noise Ratio)	27.8 dB	23.4 dB	+18.8%
SSIM (Structural Similarity Index)	0.91	0.85	+7.1%
Chamfer Distance (↓)	0.021	0.037	43.2% better

Figure 5. Quantitative comparison of Transformer and CNN models using PSNR, SSIM, and Chamfer Distance.

- Qualitative Analysis: Visual inspection of reconstructed 3D shoes showed accurate geometry, preserved sole and ankle curvature, and realistic texture mapping, even under occlusion or shadow.
- Error Distribution: Most reconstruction errors were concentrated in the heel region, especially when the input image contained partial occlusions or blur.
- PSNR (Peak Signal-to-Noise Ratio) indicates the pixel-level accuracy of image-based reconstruction.
- SSIM (Structural Similarity Index) captures textual similarity and contrast preservation.
- Chamfer Distance measures geometric proximity between predicted and reference 3D meshes.

Observations:

- The Transformer model reconstructed complex foot curves, toe contours, and sole thickness with high realism.
- Failures typically occurred in occluded regions (e.g., when only part of the foot was visible), but deformation artifacts were minimal due to positional encoding in the self-attention layers.

5.3 Latency and Efficiency Analysis:
The pipeline includes:

1. Capturing a 2D image
2. Sending it to the cloud server
3. Performing Transformer-based 3D inference
4. Receiving and rendering the reconstructed model

Stage	Average Time (Seconds)
Image Preprocessing (Client)	0.2
Upload & Inference (Cloud)	1.1
Model Download	0.3
AR Initialization & Render	1.1
Total End-to-End Time	~2.7 seconds

Figure 6. Average latency breakdown of the 2D-to-3D AR pipeline.

Remarks:

1. The pipeline achieved near real-time performance.
2. Cloud inference was optimized through model quantization and GPU-backed acceleration.
3. Potential future improvement includes moving parts of the pipeline (e.g., personalization models) on-device using TensorFlow Lite.

5.4 AR Visualization Performance in Flutter:
After receiving the .glb model from TripoSR via the cloud API, the Flutter app parsed and displayed it using Unity AR overlays. Results were analyzed on two main aspects:

Metric	Value (Avg)	Observation
Model Loading Time	1.5 seconds	Acceptable latency via cloud-fetch
AR Overlay Latency	50–60 ms	Real-time responsiveness
Tracking Stability	High	MediaPipe + ARCore accurate tracking
Tracking Drift	< 3° average offset	Using MediaPipe + ARCore anchors
Foot-Shoe Alignment Error	±1.2 cm (in-app test)	Estimated via pixel-grounded measurements

Figure 7. AR performance metrics for real-time model loading, tracking, and alignment.

System Behavior:
- Unity was embedded into Flutter via flutter_unity_widget, maintaining full Flutter interactivity while allowing real-time rendering in AR.
- Plane detection (floor, table) worked accurately for >95% of test environments.
- Users could rotate, resize, and customize the shoe in AR using intuitive gestures.

5.5 Cross-Platform Compatibility and Scalability:
The app was tested across a range of devices:
1. Android Devices: Pixel 7 Pro, Samsung S21, OnePlus 10
2. iOS Devices: iPhone 13, iPhone 12 Mini

Feature	Android	iOS
Camera Integration	Yes	Yes
AR Overlay (ARCore/ARKit)	Yes	Yes
Model Streaming	Yes	Yes
Gesture Customization	Yes	Yes

Figure 8. Cross-platform AR feature compatibility on Android and iOS.

5.6 Comparison with Existing Solutions:

System	Reconstruction Method	AR Engine	Cross-Platform	Real-Time?
Our System (Transformer + Flutter)	Transformer	Unity + Flutter	Yes	Yes
AR ShoeTry (Priyanka et al. [46])	Manual 3D Model	Flutter Native	Yes	No
CNN+GAN Baseline	CNN + GAN Reconstruction	Unity	No	No

Figure 9. Comparative analysis of AR footwear systems by architecture and deployment features.

5.7 Sustainability and Commercial Impact:
The ability to visualize accurate 3D footwear on mobile devices reduces the likelihood of product returns—a significant environmental and logistical cost in e-commerce.
Estimated Impact:
- Reduction in Return Rate: 30–40% based on survey responses
- Cloud Inference Cost: ~$0.007 per reconstruction (batch optimized)
- Scalability: Hundreds of concurrent users supported via serverless architecture
- This demonstrates that the system is not only technically effective but commercially viable and environmentally beneficial.

6. Conclusion

In this research, we proposed a unified and modular hybrid deepfake detection framework designed to identify manipulated media in both image and audio modalities. The system employs a dual-path strategy that combines Convolutional Neural Networks (CNNs) for discriminative feature extraction and Variational Autoencoders (VAEs) for anomaly-based reconstruction detection. This hybrid design leverages the strengths of supervised and unsupervised learning, enabling the system to detect subtle manipulations even in the absence of explicitly labeled fake data during training.

For the image-based detection pipeline, we utilized the 140K Real and Fake Faces dataset. A CNN-VAE model was trained to reconstruct authentic face images, and discrepancies in reconstruction were used to identify manipulated inputs. Additionally, a ResNet18-based classifier was deployed on residual images (difference between input and reconstruction) to enhance binary classification accuracy. The combined approach achieved a test accuracy of 86.27%, with balanced F1-scores for both real and fake classes. Visualizations of reconstructed images and the confusion matrix confirmed the model's ability to learn meaningful latent representations of real faces.

Architecture diagrams for both modalities were developed and integrated into the methodology section to visually articulate the internal design.

Evaluation outputs—including classification reports, confusion matrices, and reconstruction visuals—provided empirical support for the system's efficacy.

At the core of our system lies a Transformer-based neural architecture that accurately reconstructs 3D foot and shoe geometry from a single-view RGB image. By leveraging self-attention mechanisms and global feature extraction, the model overcomes limitations faced by traditional convolutional methods in terms of shape coherence, textural fidelity, and occlusion robustness. Quantitative evaluation using PSNR, SSIM, and Chamfer Distance demonstrates that the Transformer model achieves significantly better reconstruction quality compared to CNN baselines, particularly in fine-grained regions such as toes, soles, and contour curves.

On the mobile deployment side, Flutter was chosen for its ability to create high-performance, platform-agnostic user interfaces. The integration with Unity, ARCore, and ARKit enables seamless AR rendering and interactivity on both Android and iOS devices. Our end-to-end pipeline—from image capture, cloud inference, and 3D model streaming to live AR overlay—achieved an average latency of under 3 seconds, satisfying the performance requirements for commercial and consumer use cases.

The system was tested in real-world scenarios and with real users, achieving high satisfaction scores for realism, responsiveness, and customization. Users could interact with the model in real time, change colors and materials, and receive AI-driven shoe recommendations based on their foot shape. This high level of personalization, powered by lightweight TensorFlow Lite models and potential support for federated learning, positions our system not only as a consumer convenience but as a privacy-preserving and scalable retail tool

Furthermore, by providing accurate foot modeling and virtual fit feedback, our solution helps reduce product returns—a major environmental and economic cost in online commerce. With cloud inference, mobile deployment, and on-device intelligence, the architecture is highly scalable and suited for edge, serverless, or hybrid environments.

References

[1] Z. Lei, Y. Chen, Y. Yang, M. Xia and Z. Qi, "Bootstrapping Automated Testing for RESTful Web Services," in IEEE Transactions on Software Engineering, vol. 49, no. 4, pp. 1561-1579, 1 April 2023, doi: 10.1109/TSE.2022.3182663.

[2] P. Bide, N. Sharma, D. Sheikh and A. Baranwal, "Augmented Reality Indoor Navigation with Smart Appliance control," 2023 International Conference on Modeling, Simulation & Intelligent Computing (MoSICom), Dubai, United Arab Emirates, 2023, pp. 356-361, doi: 10.1109/MoSICom59118.2023.10458768.

[3] F. Zulfiqar, R. Raza, M. O. Khan, M. Arif, A. Alvi and T. Alam, "Augmented Reality and its Applications in Education: A Systematic Survey," in IEEE Access, vol. 11, pp. 143250-143271, 2023, doi: 10.1109/ACCESS.2023.3331218.

[4] H. Jeong, K. -S. Choi, Y. Son and Y. Yang, "ADA: Augmented-Reality Development Adaptor for Supporting AR glass," 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2023, pp. 1513-1516, doi: 10.1109/ICTC58733.2023.10392424.

[5] H. Vardhan, A. Saxena, A. Dixit, S. Chaudhary and A. Sagar, "AR Museum: A Virtual Museum using Marker less Augmented Reality System for Mobile Devices," 2022 3rd International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, 2022, pp. 1-6, doi: 10.1109/ICICT55121.2022.10064611.

[6] S. D. Meglio and L. Libero Lucio Starace, "Evaluating Performance and Resource Consumption of REST Frameworks and Execution Environments: Insights and Guidelines for Developers and Companies," in IEEE Access, vol. 12, pp. 161649-161669, 2024, doi: 10.1109/ACCESS.2024.3489892.

[7] S. Vinodh Kumar, S. Rithika, P. Kumar and C. Raghavendran, "Evolved Retail: Harnessing Markerless AR to Enhance Virtual Shopping Experiences," 2024 Second International Conference on Advances in Information Technology (ICAIT), Chikkamagaluru, Karnataka, India, 2024, pp. 1-6, doi: 10.1109/ICAIT61638.2024.10690596.

[8] Nurhadi, E. A. Winanto and Saparudin, "Enhance Object Tracking on Augmented Reality Markerless using FAST Corner Detection," 2021 IEEE 5th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Purwokerto, Indonesia, 2021, pp. 66-70, doi: 10.1109/ICITISEE53823.2021.9655930.

[9] A. M. Anjana Sundari, N. Dipesh Pareriya and V. Karna, "Development of Apparel 360° - An AR based Virtual Trial Room," 2023 International Conference on Digital Applications, Transformation & Economy (ICDATE), Miri, Sarawak, Malaysia, 2023, pp. 1-5, doi: 10.1109/ICDATE58146.2023.10248652.

[10] T. Mahmud, M. Che and G. Yang, "Detecting Android API Compatibility Issues With API Differences," in IEEE Transactions on Software Engineering, vol. 49, no. 7, pp. 3857-3871, July 2023, doi: 10.1109/TSE.2023.3274153.

[11] N. P. Singh, B. Sharma and A. Sharma, "Performance Analysis and Optimization Techniques in Unity 3D," 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022, pp. 245-252, doi: 10.1109/ICOSEC54921.2022.9952025.

[12] Ding, Lei & Liu, Mengyao & Miao, Lei & Yang, Tingting & Hu, Yan & Zhou, Xiaocheng. (2024). Numerical Calculation and Optimization Algorithm Based on Unity Physics Engine. 1016-1021. 10.1109/CSNT60213.2024.10545990.

[13] N. R R, R. M, R. B. S, S. Sultana and N. M. Nadig, "Markerless Augmented Reality Application for Interior Designing," 2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), Bangalore, India, 2022, pp. 1-5, doi: 10.1109/ICATIECE56365.2022.10047281.

[14] M. Chaudhary, G. Singh, L. Gaur, N. Mathur and S. Kapoor, "Leveraging Unity 3D and Vuforia Engine for Augmented Reality Application Development," 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2023, pp. 1139-1144, doi: 10.1109/ICTACS59847.2023.10390072.

[15] S. Sridhar and S. Sanagavarapu, "Instant Tracking-Based Approach for Interior Décor Planning with Markerless AR," 2020 Zooming Innovation in Consumer

Technologies Conference (ZINC), Novi Sad, Serbia, 2020, pp. 103-108, doi: 10.1109/ZINC50678.2020.9161789.

[16] Zhang, X., Zhu, Y., Chen, L. et al. Augmented reality navigation method based on image segmentation and sensor tracking registration technology. Sci Rep 14, 15281 (2024). https://doi.org/10.1038/s41598-024-65204-z.

[17] Dash, A.K., Balaji, K.V., Dogra, D.P. et al. Interactions with 3D virtual objects in augmented reality using natural gestures. Vis Comput 40, 6449–6462 (2024). https://doi.org/10.1007/s00371-023-03175-4.

[18] Bao, Y., Liu, J., Jia, X. et al. An assisted assembly method based on augmented reality. Int J Adv Manuf Technol 135, 1035–1050 (2024). https://doi.org/10.1007/s00170-024-14563-y.

[19] Hidayat, R., Wardat, Y. A systematic review of Augmented Reality in Science, Technology, Engineering and Mathematics education. Educ Inf Technol 29, 9257–9282 (2024). https://doi.org/10.1007/s10639-023-12157-x.

[20] Mansour, N., Aras, C., Staarman, J.K. et al. Embodied learning of science concepts through augmented reality technology. Educ Inf Technol (2024). https://doi.org/10.1007/s10639-024-13120-0.

[21] Bogner, J., Kotstein, S. & Pfaff, T. Do RESTful API design rules have an impact on the understandability of Web APIs?. Empir Software Eng 28, 132 (2023). https://doi.org/10.1007/s10664-023-10367-y.

[22] Gamha, Y. A framework for REST services discovery and composition. SOCA 17, 259–275 (2023). https://doi.org/10.1007/s11761-023-00376-6.

[23] Balusa, B.C., Chatarkar, S.P. Bridging Deep Learning & 3D Models from 2D Images. J. Inst. Eng. India Ser. B (2024). https://doi.org/10.1007/s40031-024-01176-y.

[24] Saini, S., Engel, I. & Peissig, J. An end-to-end approach for blindly rendering a virtual sound source in an audio augmented reality environment. J AUDIO SPEECH MUSIC PROC. 2024, 16 (2024). https://doi.org/10.1186/s13636-024-00338-6.

[25] Jaber, O., Bagossi, S., Fried, M.N. et al. Conceptualizing functional relationships in an augmented reality environment: connecting real and virtual worlds. ZDM Mathematics Education 56, 605–623 (2024). https://doi.org/10.1007/s11858-024-01594-8.

[26] Aydemir, F., Başçiftçi, F. Performance and Availability Analysis of API Design Techniques for API Gateways. Arab J Sci Eng (2024). https://doi.org/10.1007/s13369-024-09474-9.

[27] Fu, Kui & Peng, Jiansheng & He, Qiwen & Zhang, Hanxiao. (2021). Single image 3D object reconstruction based on deep learning: A review. Multimedia Tools and Applications. 80. 1-36. 10.1007/s11042-020-09722-8.

[28] Lilligreen, G., Wiebel, A. Near and Far Interaction for Outdoor Augmented Reality Tree Visualization and Recommendations on Designing Augmented Reality for Use in Nature. SN COMPUT. SCI. 4, 248 (2023). https://doi.org/10.1007/s42979-023-01675-7.

[29] Udeozor, C., Chan, P., Russo Abegão, F. et al. Game-based assessment framework for virtual reality, augmented reality and digital game-based learning. Int J Educ Technol High Educ 20, 36 (2023). https://doi.org/10.1186/s41239-023-00405-6.

[30] Z. Xu, L. Chao and X. Peng, "T-REST: An Open-Enabled Architectural Style for the Internet of Things," in IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4019-4034, June 2019, doi: 10.1109/JIOT.2018.2875912.

[31] W. Huang, X. Cao, K. Lu, Q. Dai and A. C. Bovik, "Toward Naturalistic 2D-to-3D Conversion," in IEEE Transactions on Image Processing, vol. 24, no. 2, pp. 724-733, Feb. 2015, doi: 10.1109/TIP.2014.2385474.

[32] T. Zheng et al., "RESTLess: Enhancing State-of-the-Art REST API Fuzzing With LLMs in Cloud Service Computing" in IEEE Transactions on Services Computing, vol. 17, no. 06, pp. 4225-4238, Nov.-Dec. 2024, doi: 10.1109/TSC.2024.3489441.

[33] Ma, Jun & Ji, Shunping. (2024). Evaluation of NeRF Techniques in 3D Reconstruction with Remote Sensing Imagery. 147-156. 10.1109/ICGMRS62107.2024.10581345.

[34] Y. -H. Huang, Y. -P. Cao, Y. -K. Lai, Y. Shan and L. Gao, "NeRF-Texture: Synthesizing Neural Radiance Field Textures," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 9, pp. 5986-6000, Sept. 2024, doi: 10.1109/TPAMI.2024.3382198.

[35] Yuze He, Yushi Bai, Matthieu Lin, Jenny Sheng, Yubin Hu, Qi Wang, Yu-Hui Wen, Yong-Jin Liu, Text-image conditioned diffusion for consistent text-to-3D generation, Computer Aided Geometric Design, Volume 111, 2024, 102292, ISSN 0167-8396, https://doi.org/10.1016/j.cagd.2024.102292.

[36] Yi Wang, Jing-Song Cheng, Qiao Feng, Wen-Yuan Tao, Yu-Kun Lai, Kun Li,TSNeRF: Text-driven stylized neural radiance fields via semantic contrastive learning, Computers & Graphics, Volume 116, 2023, Pages 102-114, ISSN 0097-8493, https://doi.org/10.1016/j.cag.2023.08.009.

[37] Yao Zhang, Jiangshu Wei, Bei Zhou, Fang Li, Yuxin Xie, Jiajun Liu, TVNeRF: Improving few-view neural volume rendering with total variation maximization, Knowledge-Based Systems, Volume 301, 2024, 112273, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2024.112273.

[38] C. Zheng, C. Hu, Y. Chen and J. Li, "A Self-Learning-Update CNN Model for Semantic Segmentation of Remote Sensing Images," in IEEE Geoscience and Remote Sensing Letters, vol. 20, pp. 1-5, 2023, Art no. 6004105, doi: 10.1109/LGRS.2023.3261402.

[39]X. Yang, G. Lin and L. Zhou, "Single-View 3D Mesh Reconstruction for Seen and Unseen Categories," in IEEE Transactions on Image Processing, vol. 32, pp. 3746-3758, 2023, doi: 10.1109/TIP.2023.3279661.

[40] A. Dalal, D. Hagen, K. G. Robbersmyr and K. M. Knausgård, "Gaussian Splatting: 3D Reconstruction and Novel View Synthesis: A Review," in IEEE Access, vol. 12, pp. 96797-96820, 2024, doi: 10.1109/ACCESS.2024.3408318.

[41] W. Qin, X. Luo, S. Li and M. Zhou, "Parallel Adaptive Stochastic Gradient Descent Algorithms for Latent Factor Analysis of High-Dimensional and Incomplete Industrial Data," in IEEE Transactions on Automation Science and Engineering, vol. 21, no. 3, pp. 2716-2729, July 2024, doi: 10.1109/TASE.2023.3267609.

[42] Y. -J. Yuan et al., "Interactive NeRF Geometry Editing With Shape Priors," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 12, pp. 14821-14837, Dec. 2023, doi: 10.1109/TPAMI.2023.3315068.

[43] G. Wang, P. Wang, Z. Chen, W. Wang, C. C. Loy and Z. Liu, "PERF: Panoramic Neural Radiance Field From a Single Panorama," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 10, pp. 6905-6918, Oct. 2024, doi: 10.1109/TPAMI.2024.3387307.

[44] B. Cai et al., "3D Scene Creation and Rendering via Rough Meshes: A Lighting Transfer Avenue" in IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 46, no. 09, pp. 6292-6305, Sept. 2024, doi: 10.1109/TPAMI.2024.3381982.

[45] Zexu Huang, Sarah Monazam Erfani, Siying Lu, Mingming Gong, Efficient neural implicit representation for 3D human reconstruction, Pattern Recognition,

Volume 156,2024,110758, ISSN,0031-3203, https://doi.org/10.1016/j.patcog.2024.110758.

[46] V. Priyanka, et al., "Augmented Reality Based Virtual Shoe Try-on Application using Flutter," IEEE SmartTech, 2022. https://doi.org/10.1109/9757443.

[47] Y. Zhou et al., "Deep Learning-Based Foot Measurement Using Smartphones for Shoe Size Recommendation," IEEE Access, 2020. https://doi.org/10.1109/9071367.

[48] S. Das and R. Panigrahi, "Performance Analysis of AR-Based Foot Measurement Applications," IEEE Emerging Trends, 2022. https://doi.org/10.1109/9951938.

[49] T. Mahmud et al., "Accurate and Real-Time Foot Tracking Using Depth Sensors and Machine Learning," IEEE Transactions on Consumer Electronics, 2023. https://doi.org/10.1109/10162491.

[50] A. Sharma and S. Aggarwal, "Edge Computing in Flutter-Based AR Systems for Footwear Fitting," IEEE Transactions on Network and Service Management, 2024. https://doi.org/10.1109/10692626.