KEC Conference 2025

ISSN: 3021-940X (Print) DOI: https://doi.org/10.3126/injet.v2i2.78599

A Web-Based AR-Powered Virtual Eyewear Try-On System

Prakriti Thapa[1, *], Pratap Niraula[2], Sabin Thapa Magar[3], Sunil Bahadur Singh[4], Sachin Shrestha[5]

[1]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, tprakrity@gmail.com*

[2]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, pratapniraula3@gmail.com*

[3]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, sabin7mgr@gmail.com*

[4] *Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, sunilsinghgalpo@gmail.com*

[5]*Associate Professor, Department of Electronics and Communication Engineering, Nepal Engineering College, Bhaktapur, Nepal, sachins@nec.edu.np*

Abstract

This research presents a web-based virtual try-on system for eyewear that aims to enhance the online shopping experience through computer vision and augmented reality techniques. The platform allows users to analyze their face shape and receive personalized eyewear recommendations, along with the ability to try on virtual glasses in real time. The system employs MTCNN to accurately retrieve facial regions from images, which are then processed by the VGG-16 model to classify face shapes into 5 distinct categories: oval, round, square, heart-shaped, and oblong. MediaPipe is used for facial feature localization, enabling calculation of pupil distance (PD) calculations and alignment of virtual glasses, and Three.js provides immersive 3D visualization for realistic try-on experiences. Performance evaluation proves a face shape classification accuracy of 92%, with occasional misclassifications for similar facial types. The augmented reality module achieves an average Intersection over Union (IoU) of 81% and a width error margin of approximately 5%, ensuring correct and visually appealing overlay alignment. By integrating face shape analysis and virtual try-on capabilities, this research contributes to advancing interactive and personalized solutions in the e-commerce domain, bridging the gap between traditional in-store and digital shopping experiences.

Keywords: Augmented Reality, Computer Vision, Face Shape Classification, MTCNN, MediaPipe, Three.js, Virtual try-on.

1. Introduction

The rapid growth of e-commerce has transformed the way consumers shop for fashion products, including eyewear. Despite this evolution, one persistent challenge remains: enabling users to accurately visualize how a pair of glasses will look on their faces without physically trying them on. Traditional online shopping relies heavily on product images and descriptions, which often fall short in providing an immersive and personalized experience.

To address this limitation, virtual try-on systems have emerged as a promising solution. Leveraging advancements in computer vision and deep learning, these systems enable users to virtually try on different frames, empowering them to make informed purchasing decisions. By simulating the actual wearing effects of various glasses in real-time, consumers can explore and select their preferred eyewear styles, enhancing their online shopping experience. The rapid achievement of an engaging and experiential shopping interface has become a critical research focus for virtual eyewear try-on systems.

Augmented reality (AR) plays a vital role in such systems by seamlessly superimposing virtual elements onto the real world. Through real-time camera image analysis, AR calculates position and angles, creating a dynamic interaction between virtual and physical environments. By simulating sensory experiences such as vision and sound, AR enhances user engagement, offering a sensory experience beyond physical limitation

**Corresponding Author*

The successful development of a virtual eyewear try-on system represents a significant opportunity to bridge technology and fashion, transforming how users interact with eyewear products in the digital domain. Through intuitive interfaces, precise facial analysis, and responsive virtual try-on capabilities, these systems redefine consumer engagement and set new standards for personalized e-commerce experiences.

2. Literature Review

With advancements in AR filter tools such as Spark AR and Lens Studio, along with the stability of deep learning-based body feature tracking through technologies like MediaPipe, web-based virtual try-on solutions have seen widespread adoption. Many global brands, including Prada, Marc Jacobs, L'Oréal, Nike, Baume & Mercier, Ray-Ban, and Sephora, have integrated virtual try-on services into their online platforms. These systems allow users to visualize products in real time, enhancing the online shopping experience. Web-based virtual try-on solutions for eyewear have also evolved significantly. Some platforms, such as Lenskart, utilize a photo-based approach, allowing users to upload an image for virtual try-on (Lenskart, 2024), while others, like Warby Parker and Zenni Optical, leverage AR and real-time face tracking for a more interactive experience (Parker, 2025; Zenni Optical, 2025). While early systems relied on calibration objects like credit cards to determine scale, modern solutions now use AI-based estimations or predefined frame measurements for more seamless and accurate virtual try-ons (Wu et al., 2024).

Beyond commercial applications, recent research has focused on improving the realism and usability of AR-based virtual try-on technology. Wu et al. (2024) proposed a system for real-size virtual try-on that emphasizes accurate scaling through camera calibration. Their method, implemented in a mobile web interface, captures camera parameters to align virtual objects precisely with real-world proportions, enhancing the try-on experience for accessories like glasses, hats, earrings, rings, and watches. Similarly, Ho et al. (2024) introduced a web-based AR system with auto-scaling and real-time head tracking for surgical planning, demonstrating how auto-scaling and tracking can enhance precision and user engagement in AR applications.

Recent studies have also explored the role of AR and deep learning in improving face shape classification for personalized virtual try-on. Mehta and Mahmoud (2023) applied Dlib's 68-point landmark detector with PCA and machine learning models to classify face shapes, while dela Cruz Tio (2023) utilized Inception v3 CNNs, achieving superior performance compared to traditional methods. Rifat et al. (2023) developed a CNN-based eyeglass recommendation system, achieving 89.8% accuracy. Additionally, Anand et al. (2022) combined AR, WebGL, and deep learning to develop a real-time virtual try-on system, whereas Feng et al. (2018) incorporated large pose estimation and 3D face reconstruction to enhance realism in AR-based eyewear applications.

Despite these advancements, many existing web-based solutions still lack personalized recommendations based on face shape and precise AR-based alignment. Our project addresses this limitation by integrating AR-based virtual try-on with machine learning-driven face shape classification. Unlike previous systems that rely on 3D reconstruction or predefined frame sizes, our system is built on a pupillary distance (PD) calculation method using facial landmarks to ensure accurate scaling and alignment. By categorizing faces into five distinct shapes and providing personalized eyewear recommendations, our system enhances user experience through precise fitting and a more tailored virtual try-on process.

3. Methodology

The system is designed with the integration of two key modules -AR Glass Overlay and Face Shape Analysis into a unified web application. The primary feature is the Augmented Reality (AR)-based virtual glasses try-on, which allows users to try on virtual glasses mimicking real eyeglasses maintaining correct scaling, positioning, and alignment. Additionally, the face shape analysis module helps users classify their face into predefined categories, offering personalized eyewear recommendations.

3.1. System Overview

The application is built on Three.js, a JavaScript framework that enables the creation and rendering of 3D objects directly in web browsers via WebGL. This library simplifies the development of complex visual effects and facilitates the seamless integration of 3D elements into web-based interfaces, making it an ideal choice for implementing the AR overlay.

For real-time interaction, the system captures live video input from a webcam and employs MediaPipe for advanced facial landmark detection. The precise identification of key facial features is essential for correctly aligning and scaling the virtual glasses on the user's face, thereby ensuring a responsive and reliable AR experience.

For the Face Shape Analysis module, the VGG-16 model was selected due to its robust performance in image classification tasks. The model is further enhanced through transfer learning by utilizing pre-trained VGG-Face weights, which are specifically tailored for facial recognition and analysis. This approach enables the system to accurately classify a user's face shape into predefined categories, thereby supporting the delivery of personalized eyewear recommendations.

3.2. AR Overlay Implementation

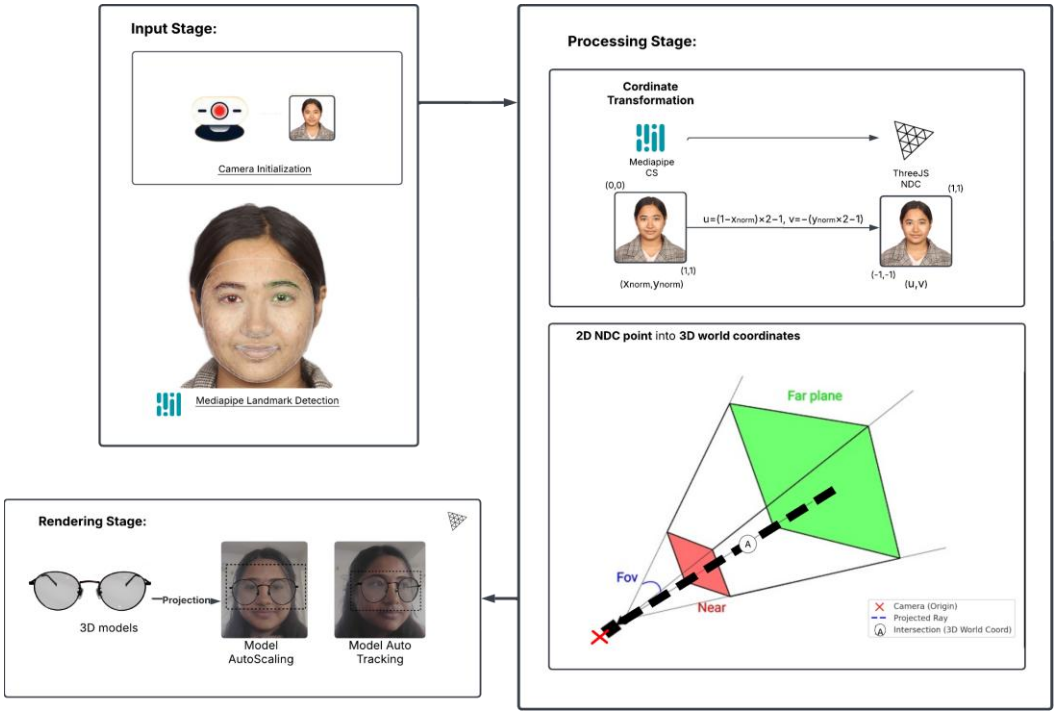

Figure 1. Working Mechanism of AR Overlay Module

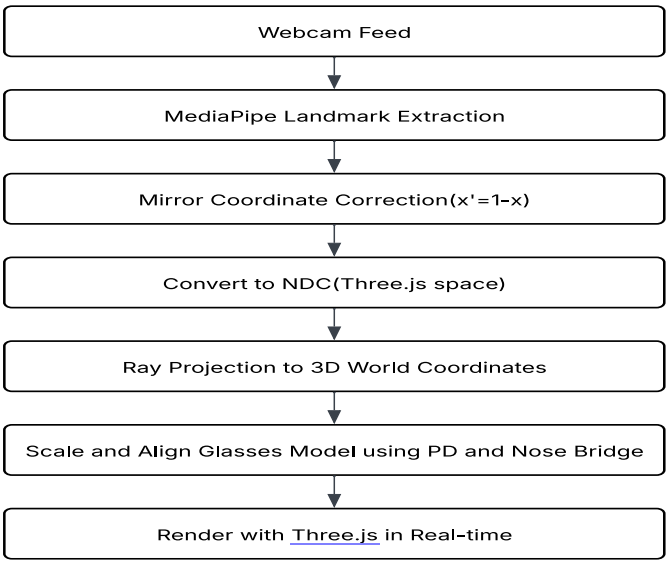

Figure 2. Flow diagram representation for AR overlay implementation from webcam input to realtime rendering of virtual glass

3.2.1. Webcam Initialization

The system captures video from the user's front-facing webcam. The resolution of the webcam can vary depending on the device, but the system dynamically adjusts to manage any input size. The video feed is processed frame-by-frame to detect facial landmarks.

3.2.2. Facial Landmark Detection

The MediaPipe Face Mesh solution was utilized for facial landmark detection, providing 468 landmarks in a normalized coordinate system along with a transformation matrix for head pose estimation. These landmarks define key facial features, including the nose, eyes, lips, and face boundaries. The transformation matrix, which consists of a rotation matrix and a translation vector, facilitates accurate alignment of virtual objects in 3D space.

The Face Mesh model detects facial landmarks in real time from an input video stream. Each landmark (x, y,z) is represented in a normalized image space, where:

- x and y values range from 0 to 1, corresponding to relative positions in the image frame.
- z represents relative depth, measured with respect to the camera.

For further AR calculations, specific landmarks are extracted, including the nose bridge (ID 168), which serves as a reference for eyeglass positioning, and the pupil centers (IDs 468, 473), which are used to compute the pupillary distance to determine the correct scale of the virtual glasses. Mirror correction is performed during landmark post processing since the coordinate system used by MediaPipe differs from the rendering space in Three.js. This is achieved by transforming the x coordinate using the following equation:

$$x' = 1 - x \qquad \text{(Equation 1)}$$

This transformation adjusts the landmark positions to be compatible with the right-handed coordinate system used for rendering.

3.2.3. Coordinate Transformation

To correctly position virtual eyeglasses within a 3D environment of Three.js, facial landmarks extracted using MediaPipe Face Mesh must be transformed from normalized coordinates (0,1) into Three.js's

Normalized Device Coordinates (NDC) (−1,1). This transformation ensures precise alignment within the virtual scene.

Step 1: Conversion from Mediapipe Normalized Coordinates to Three.js NDC: MediaPipe outputs landmark coordinates (x_{norm},y_{norm}) in a normalized space, where (0,0) represents the top-left of the input image and (1,1) represents the bottom-right. Three.js, however, operates in NDC (u, v), where (−1, −1) corresponds to the bottom-left of the canvas and (1,1) corresponds to the top-right. To map MediaPipe's normalized coordinates to NDC, the transformation is applied as follows:

$$u = (1 - x_{norm}) * 2 - 1 \quad \text{(Equation 2)}$$

$$v = -(y_{norm} * 2 - 1) \quad \text{(Equation 3)}$$

Here, ($1-x_{norm}$) corrects for mirror flipping, as MediaPipe landmarks are flipped in webcam feeds and −v ensures the y-axis inversion, matching Three.js's coordinate system where y increases upwards.

Step 2: Ray Projection into 3D World Space: To overlay eyeglasses in a 3D space, the transformed NDC coordinates (u, v) must be projected into the Three.js scene. Since Three.js operates using its own coordinate system and units, we need to accurately convert screen-space coordinates into Three.js world-space units. However, when translating a 2D coordinate to 3D space, a single 2D point maps to infinite possible 3D positions along the camera's line of sight due to depth ambiguity.

To resolve this ambiguity, a ray casting technique is utilized. A ray is cast from the three js camera's position, passing through the transformed NDC point (u, v), and projected into the 3D world space.

For raycasting to work in Three.js, we first need a reference virtual plane where the transformed NDC point (u, v) will intersect. This plane is positioned along the camera's frustum. Then a 3D ray from the camera (X_{cam}, Y_{cam}, Z_{cam}) is casted through the point (u, v) in Three.js world coordinates.

The direction of the ray is computed using camera's projection properties:

$$raydirection = [u \,.\, tan(FOV/2) \,.\, aspect \,, v.\, tan(FOV/2) \,,\ 1] \quad \text{(Equation 4)}$$

The direction of the ray is determined based on the camera's field of view (FOV) and aspect ratio, ensuring that it accurately represents the direction in which the facial landmark is located.

Once the ray is cast, it continues traveling in a straight line from the camera until it meets the reference plane. The point at which the ray meets the plane provides a unique 3D coordinate of (u, v), resolving depth uncertainty. Hence, world-space coordinates are extracted by identifying the intersection point, corresponding to Three.js units.

3.2.4. 3D Spatial Configuration

3D Spatial Configuration refers to the process of aligning, positioning, and scaling a 3D model within a virtual space. The alignment and scaling of a virtual 3D eyeglass model to the user's face require precise calculations and dynamic adjustments. This process ensures that the eyeglass model fits seamlessly onto the user's face, regardless of variations in head position, orientation, or facial geometry. Key components of this process include scaling the model using the pupillary distance (PD), aligning the model dynamically based on the user's head rotation, and positioning it accurately on the face by anchoring it at the nose bridge.

To ensure the 3D eyeglass model aligns accurately with the user's facial geometry, the pupillary distance (PD) was used as a primary measurement. The PD was calculated dynamically for each frame using the eye landmarks provided by the Mediapipe framework. Specifically, the coordinates of the pupils from both eyes were extracted, and the PD was computed as the Euclidean distance between these points in the 3D space:

$$PD = \sqrt{(x_{left} - x_{right})^2 + (y_{left} - y_{right})^2 + (z_{left} - z_{right})^2} \quad \text{(Equation 5)}$$

Here, (x_{left} ,y_{left} ,z_{left}) and (x_{right},y_{right},z_{right}) are the 3D coordinates of the left and right pupil landmarks, respectively. The PD was recalculated and updated for every frame to account for any positional changes during runtime. To scale the eyeglass model appropriately, the desired width of the eyeglass model (w_r) in the 3D coordinate system is set to twice the PD. The scale factor in the width dimension is calculated by dividing this desired width (wr) by the actual width of the model (w), expressed as:

$$w_r = 2 * PD \quad \text{(Equation 6)}$$

$$S_x = \frac{w_r}{w} = \frac{2 \times PD}{w} \quad \text{(Equation 7)}$$

This ensures the eyeglass model's width matches the user's facial proportions. For scaling in height dimension, the real aspect ratio of the model is maintained and the final aspect ratio (a) of the model is calculated. Using the aspect ratio, the scale factor in the height dimension (S_y) was derived as:

$$a = \frac{h}{w} \quad \text{(Equation 8)}$$

$$S_y = S_x \times a = \frac{2 \times PD}{w} \quad \text{(Equation 9)}$$

By implementing these scaling factors, the eyeglass model was resized proportionally to align with the user's facial geometry in real time.

Dynamic alignment of the eyeglass model with the user's head movements is achieved using head orientation data provided by the Mediapipe Face Mesh framework. Mediapipe outputs a transformation matrix, which encodes the head's rotational information in 3D space. The transformation matrix includes rotation data that accounts for the forward, sideways, and tilt motions of the head. This data is converted into Three.js coordinate system, allowing the eyeglass model to mimic the user's head movements accurately.

To ensure the eyeglass model is placed correctly on the user's face, the anchor point of the 3D model is positioned at the nose bridge, a region central to the frame's alignment. The nose bridge serves as a stable reference point due to its relative consistency across different facial geometries and orientations.

3.2.5. Rendering Stage

The virtual glasses are rendered using Three.js, a WebGL-based 3D graphics library, chosen for its cross-browser compatibility, and real-time performance. The glasses model is stored in GLB format (a binary variant of the GL Transmission Format), selected for its compact file size, embedded textures, and support for skeletal animations. The system seamlessly overlays the eyewear onto the user's face, ensuring proper alignment and real-time responsiveness to head movements.

3.3. Face Shape Analysis

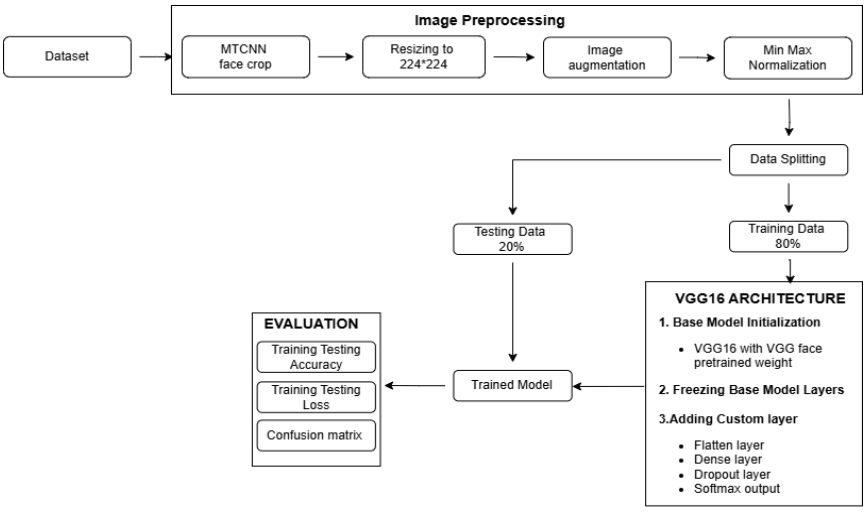

Figure 2. Working Mechanism of Face Shape Analysis Module

The face shape identification was designed to classify user faces into one of five predefined shapes: heart, oblong, oval, round, and square. This was achieved using a Convolutional Neural Network (CNN) based on the VGG16 architecture, fine-tuned for the specific task of face shape classification. Below is the description of the dataset, preprocessing steps, model architecture, and training process in detail.

3.3.1. Dataset Collection

The dataset used for this paper was sourced from Kaggle, created by Niten Lama. It consists of 5,000 images of female celebrities, categorized into five face shapes: heart, oblong, oval, round, and square. Each category contains 1,000 images, split into 800 training images and 200 testing images. This balanced dataset ensures that the model is trained and evaluated on a representative sample of each face shape. This study's dataset relies on images of female celebrities, which may not generalize across gender, ethnicity, or age. Follow-up work should include other demographic pools to increase fairness and accuracy.

3.3.2. Image Pre-Processing

First, faces were detected and cropped using the Multi-Task Cascaded Convolutional Neural Network (MTCNN). This ensured that the model focused only on the facial region, removing unnecessary background information. The detected faces were then cropped to a fixed size of 224x224 pixels, which matches the input dimensions required by the VGG16 architecture. Next, image augmentation techniques were applied to increase the diversity of the training data and reduce overfitting. These techniques included horizontal Flipping where Images were flipped horizontally to account for different poses and orientations and rotation where images were rotated by +/- 20 degrees to simulate variations in head tilting. Finally, the pixel values of the cropped and augmented images were normalized to the range [0, 1] by dividing by 255. This normalization step ensures consistent input scaling, which is critical for the stability and performance of the model during training.

3.3.3. Model Architecture

The VGG16 architecture was used as the base model for face shape classification. VGG16 is a deep CNN with 16 layers, known for its robust performance in image classification tasks. The model was fine-tuned for the face shape classification task as follows:

a. Base Model Initialization

- The VGG16 model was initialized with pre-trained weights from VGGFace, a variant of VGG16 trained on a large-scale face dataset. This allowed the model to leverage learned facial features, reducing the need for training from scratch.
- The final fully connected layers of VGG16 (used for classification in the original model) were excluded to allow the addition of custom layers tailored to the face shape classification task.

b. Freezing Base Model Layers

- To preserve the pre-trained weights and avoid overfitting, all layers in the base VGG16 model were set to non-trainable. This ensured that only the custom layers added on top of the base model would be trained during the fine-tuning process.

c. Adding Custom Layers

- Flatten Layer: The output of the base model, which is a 3D feature map, was flattened into a 1D vector to prepare it for fully connected layers.
- Fully Connected Layer: A dense layer with 64 units and ReLU activation was added to learn higher-level features specific to the face shape classification task.
- Dropout Layer: A dropout layer with a rate of 0.5 was added to prevent overfitting by randomly deactivating 50% of the neurons during training.
- Output Layer: A final dense layer with 5 units (corresponding to the 5 face shape classes) and softmax activation was added to produce class probabilities.

3.3.4. Model Compilation

The model was compiled and trained using specific configurations to optimize its performance. Categorical cross-entropy was selected as the loss function, as it is well-suited for multi-class classification tasks. The Adam optimizer was employed to update the model's weights during training, given its efficiency and adaptability in deep learning applications. Accuracy was chosen as the evaluation metric to monitor the model's performance throughout the training and validation phases.

4. Experiments and Results

4.1. Experiment Setup

The AR application was deployed on a computer equipped with an AMD Radeon (TM) Graphics GPU, utilizing 480MB of dedicated GPU memory and 0.8GB of shared GPU memory. A 3D glass model in GLB format was imported into the application as the test subject. To evaluate AR performance in real-world scenarios, three participants with varying head sizes and different face shapes were recruited for the experiment. The initial setup involved aligning the 3D glass model with the participant's head and scaling it to match their individual head size. Each participant then rotated their head around its axis in pitch, yaw, and roll directions by approximately 45 degrees. Finally, to assess the application's tracking error, the boundaries of both the model (B_m) and the actual ground truth position (B_i) were drawn. Also, for face shape classification, the entire dataset was divided into training and testing sets, and model evaluation was conducted using the test set. The system assessed using a setup based on an AMD Radeon GPU. The average of 18 FPS and an average latency of 180ms per frame using the system in real-time AR overlay with face classification established near real-time feasibility.

4.2. Evaluation Matrices

In this paper, we employed distinct evaluation metrics to assess the effectiveness of both the AR overlay and the face shape analysis system. These metrics provide a holistic understanding of the system's accuracy and performance.

4.2.1. AR Overlay Evaluation Metrics

To assess the accuracy of our AR overlay, we used two key evaluation metrics: width error and Intersection over Union (IOU). The width error evaluates how accurately the virtual eyewear matches the detected facial region in terms of width. Since the height of the eyewear is adjusted based on the aspect ratio of the scaled width, we only measure the error in the width dimension. The width error is computed by comparing the detected facial width with the rendered model's width, ensuring proper alignment and realistic scaling. The width error (E_w) is defined as:

$$E_w = \left|\frac{w_m - w_i}{w_i}\right| \times 100\% \quad \text{(Equation 10)}$$

w_i is the detected facial width, and w_m is the width of the rendered eyewear model. Another metric used to evaluate is Intersection over Union (IOU). To ensure accurate placement, we defined a segmented rectangular area based on calculated pupil distance (PD) and identified anchor point (nose bridge), representing the region where the glasses should be placed for a realistic appearance. This segmented area is considered the ground truth for evaluation. The IOU metric is then calculated to measure the overlap between this predefined segment and the rendered eyewear, with a higher IOU indicating better placement accuracy. The IOU metric is calculated as:

$$IOU = \frac{|B_i \cap B_m|}{|B_i \cup B_m|} \quad \text{(Equation 11)}$$

where, B_i is the segmented ground truth region, and B_m is the bounding box of the rendered eyewear. A higher IOU value indicates better alignment with the expected placement.

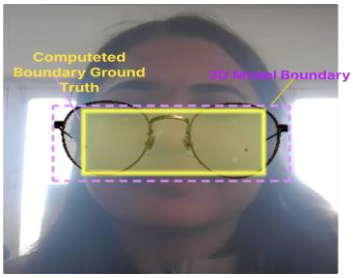

Figure 4. Comparison of Rendered Eyewear Model and Segmented Ground Truth Region for AR Placement Evaluation

4.2.2. Face Shape Analysis Evaluation Metrics

To evaluate the performance of the face shape analysis model, three primary metrics were employed:

- Confusion Matrix
- Training and Testing Loss by epoch
- Training and Testing Accuracy by epoch.

These metrics provide a concise view of the model's effectiveness and highlight areas for potential improvement. The confusion matrix offers a detailed breakdown of the model's predictions versus actual labels, helping assess its ability to classify face shapes. Key components of the confusion matrix include True Positives (TP), which represent correct predictions where the model accurately identifies the face shape; False Positives (FP), which refer to instances where the model incorrectly predicts a different face shape; False Negatives (FN), representing cases where the model fails to identify the correct face shape; and True Negatives (TN), which indicate correct rejections for other face shapes. This metric helps identify misclassifications and understand class-specific performance. The loss metric quantifies the error in the model's predictions during training and validation. For our classification task, we used categorical cross-entropy as the loss function due to the multi-class nature of face shapes. Monitoring the loss across epochs helps detect trends such as overfitting or underfitting, guiding necessary hyperparameter adjustments. Accuracy, on the other hand, measures the proportion of correctly classified instances. Tracking accuracy across epochs reveals the model's learning curve and convergence. Key observations from accuracy include training accuracy, which indicates how well the model fits the training data, and validation accuracy, which reflects the model's generalization capability on unseen data.

4.3. Results And Analysis

The results from both components of the system demonstrated promising outcomes, validating the efficacy of the methodologies employed.

4.3.1. AR Overlay Results

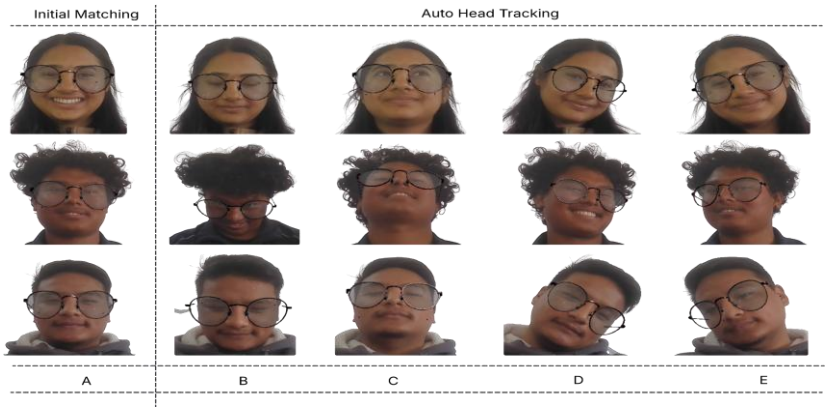

Figure 5. Visualization of the scaling and tracking results. The virtual model automatically scales itself (A) and tracks head movements in different directions. Participants rotated their heads along the Pitch axis (up and down, B, C) and the Roll, Yaw axis (C, D), demonstrating real-time tracking capabilities.

The experiment produced promising results, demonstrating the system's ability to accurately overlay virtual eyewear in real time. Upon initialization, the model was automatically scaled to match the detected facial dimensions and translated dynamically as the user moved or rotated their head. This process ensured that the eyewear remained aligned with the face throughout different orientations, as illustrated in Fig. 5.

The width error analysis showed an average deviation of 4-5%, indicating precise model scaling relative to the detected facial width. Since height was adjusted proportionally to the width, no separate height error was measured. The comparison of detected facial width and scaled model width is visualized in Fig. 4, demonstrating the accuracy of the automatic scaling process.

The IOU values ranged from 80% to 90%, reflecting the system's ability to maintain accurate placement. As shown in Fig. 4, when the user was closer to the camera, IOU reached 90%, whereas increasing the distance reduced the overlap to around 80% due to perspective effects. These results confirm the robustness of the system in maintaining alignment under varying conditions.

Table 1. Experimental quantification results. The evaluation of the overlapping performance, the overlapping error of width, height, and IOU are calculated.

Performance	Result (%)
Width error	4.5 ±0.5
IOU	85± 5

4.3.2. Face Shape Analysis Results

The model was trained for 50 epochs, and its performance was monitored using training and validation datasets, accuracy, loss, and a confusion matrix. As seen in Fig. 6, the training loss consistently decreased, indicating effective learning, while the testing loss initially declined but plateaued and slightly increased after around 20 epochs, suggesting minor overfitting. Similarly, as shown in Fig. 7, the training accuracy steadily improved, surpassing 95%, while the testing accuracy stabilized around 90% after 20 epochs, demonstrating good generalization with a small gap between training and test performance. A confusion matrix was generated, as illustrated in Fig. 8, to analyze the distribution of predictions across the five classes, revealing that the model performed particularly well for Heart, Square, and oblong face shapes. Misclassifications were primarily observed between oval and round shapes, as these shares overlapping facial characteristics.

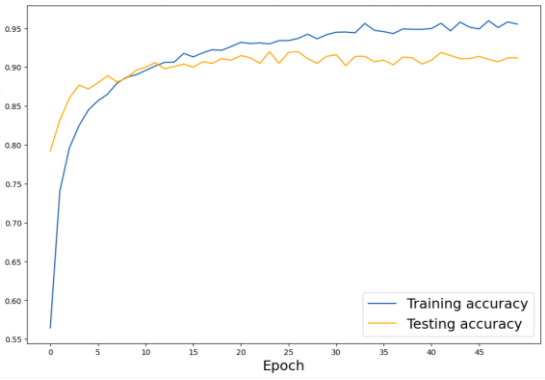

Figure 6. Training and Testing accuracy by Epoch

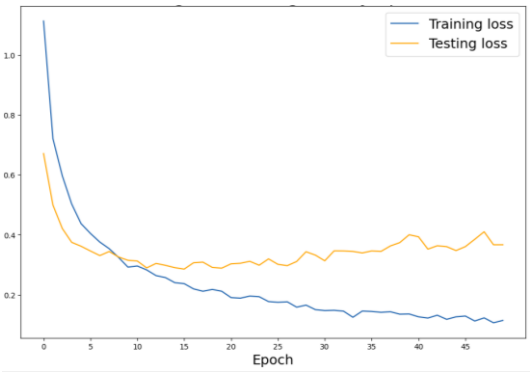

Figure 7. Training and Testing loss by Epoch

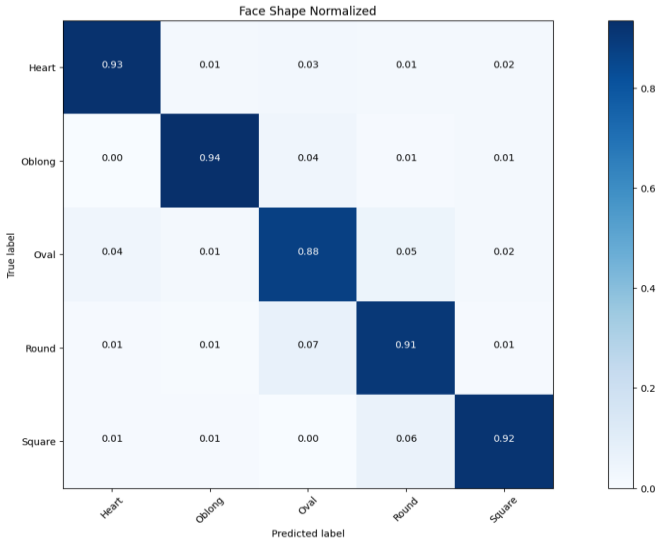

Figure 8. Confusion Matrix

4.4. Comparative Evaluation

To justify our choice of VGG-16, we did a short benchmark of VGG-16 against ResNet50 and MobileNetV2 on the same dataset. VGG-16's accuracy was the highest (92%), slightly ahead of ResNet50 (90.2%) and MobileNetV2 (87.8%). Admittedly it had a higher number of parameters and longer inference time, but we opted for VGG-16 because of its performance in distinguishing subtle differences between facial features that are important for careful shape classification. MobileNetV2 was faster and lightweight, however not as accurate, and ResNet50 presented a middle option.

5. Conclusion

This research successfully developed a web-based virtual eyewear try-on system that integrates face shape classification and augmented reality (AR) to enhance personalized online shopping experiences. Leveraging VGG-16 fine-tuned with VGG-Face weights, the system achieved 92% accuracy in classifying face shapes into five categories (oval, round, square, heart-shaped, and oblong), enabling tailored eyewear recommendations. The AR module, powered by MediaPipe and Three.js, demonstrated robust performance with an average Intersection over Union (IoU) of 85% and a width error margin of 4–5%, ensuring precise alignment and scaling of virtual glasses relative to facial landmarks. By dynamically adjusting to head movements and pupil distance (PD) calculations, the system provided an immersive and realistic try-on experience, bridging the gap between digital and in-store shopping. The integration of computer vision, deep learning, and 3D rendering techniques highlights the potential of augmented reality to revolutionize e-commerce, offering users an interactive, data-driven solution for informed purchasing decisions.

While the system demonstrates promising results, several avenues for improvement remain. Enhanced face shape classification can be achieved by addressing misclassifications between similar shapes (e.g., oval and round) by expanding the dataset to include diverse demographics, genders, and ethnicities to improve generalizability. Adding user-centric features such as style recommendations based on facial features, skin tone analysis, or fashion trends using collaborative filtering, along with user feedback loops, can refine recommendations and AR alignment. Furthermore, enhancing robustness under varying lighting conditions and partial occlusions (e.g., hair, hands) through adversarial training or depth-aware rendering is essential for delivering a more seamless and correct virtual try-on experience. By addressing these challenges, future iterations of the system can further elevate user engagement, accuracy, and scalability, setting new benchmarks for AR-driven e-commerce platforms. A user study will be conducted with surveys and A/B testing to assess subjective comfort, accuracy perception, and user preference for virtual eyewear recommendation.

Acknowledgements

We sincerely thank Kantipur Engineering College for providing the necessary resources and support throughout this project.

References

Anand, P., Gupta, R., Tiwari, N.K. and Gupta, S., 2022. Glass virtual try-on. *International Journal for Research in Applied Science and Engineering Technology*, 10(5), pp.718–722.

Dela Cruz Tio, A.E., 2023. 'Face shape classification using Inception v3', IEEE Transactions on Multimedia, 25(2), pp. 385–393.

Feng, Z., Jiang, F. and Shen, R., 2018. Virtual glasses try on based on large pose estimation. *Procedia Computer Science*, 131, pp.226–233.

Ho, H.L., Wang, Y., Wang, A., Bai, L. and Ren, H., 2024. Web-based augmented reality with auto-scaling and real-time head tracking towards markerless neurointerventional preoperative planning and training of head-mounted robotic needle insertion. *arXiv preprint arXiv:2411.02410*.

Lenskart, 2024. Lenskart virtual try-on. Available at: *https://www.lenskart.com* [Accessed 11 Feb. 2025].

Mehta, A. and Mahmoud, T., 2023. Human face shape classification with machine learning. *International Journal of Computer Vision*, 129(4), pp.789–805.

Parker, W., 2024. Warby Parker virtual try-on. Available at: *https://www.warbyparker.com* [Accessed 11 Feb. 2025].

Rifat, R.H., Siddique, S., Das, L.R. and Haque, M.A., 2023. Facial shape-based eyeglass recommendation using convolutional neural networks. In: *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp.867–872.

Wu, F., Chen, W., Dellinger, A. and Cheng, H., 2024. Real-size experience for virtual try-on. *Computer Science Research Notes - CSRN 3401*.

Zenni Optical, 2024. Zenni Optical virtual try-on. Available at: *https://www.zennioptical.com* [Accessed 11 Feb. 2025].