IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-BASED VIRTUAL LOOK RECOMMENDATION SYSTEM

1Andimeni Lakshmi Durga Tulasi, 2Nemalikonda Sai Brahma Kishore, 3Gopi D V V S S Prasad, 4Manekula Satya Krishnaveni

1,2,3,4MCA, Department of Computer Applications

1,2,3,4Aditya University, Surampalem, Andhra Pradesh, India

Abstract:

The proliferation of digital fashion tools has created a compelling need for intelligent, personalized styling platforms that do not rely on costly hardware or proprietary infrastructure. This paper presents an AI-based Virtual Look Recommendation System, a full-stack web application that enables users to virtually try on hairstyles and eyeglasses prior to real-world decisions. The system integrates Google's MediaPipe FaceMesh framework to detect 468 three-dimensional facial landmarks from a single uploaded photograph. Based on geometric ratio computations derived from these landmarks, the system classifies a user's face into one of six canonical shapes: Oval, Round, Square, Heart, Diamond, or Oblong. The detected face shape, combined with the user's selected gender, drives a rule-based recommendation engine that surfaces contextually appropriate hairstyle and eyewear options. Selected styles are rendered onto the user's photograph through alpha-blended PNG compositing, supported by tilt-correction rotation and per-style scaling configurations. The backend is engineered with Python FastAPI, exposing RESTful endpoints for image ingestion, face analysis, recommendation retrieval, and overlay generation. The frontend is developed in React.js, delivering an interactive Try-On Studio and a Personalized Recommendations Dashboard. System evaluation on a 50-image test dataset demonstrates a face detection success rate of 94%, a face shape classification accuracy of 83%, and an average user satisfaction rating of 3.8 out of 5.0 for visual overlay quality. The system demonstrates that effective virtual try-on experiences are achievable using entirely open-source tools without GPU infrastructure or custom model training.

Index Terms - Face Shape Detection; MediaPipe FaceMesh; Virtual Try-On; Hairstyle Recommendation; Eyewear Recommendation; Computer Vision; FastAPI; React.js; Image Compositing; Alpha Blending

I. INTRODUCTION

The global hair care, styling, and eyewear industries collectively represent multi-billion-dollar markets, yet a persistent consumer challenge remains: people are often reluctant to experiment with new hairstyles or glasses because they cannot reliably predict how a given style will look on them. Traditional alternatives, such as manual photo editing or static style guides on optician websites, demand either technical skill or provide no real personalization. Commercial augmented reality try-on applications, while effective, are typically closed-source, device-dependent, or restricted to proprietary product catalogues.

The AI-based Virtual Look Recommendation System addresses these gaps by providing an intelligent, fully automated, web-based platform accessible to any user with a standard browser and a photograph. The system leverages Google's MediaPipe FaceMesh framework to extract 468 facial landmarks from a single image, derives geometric ratios from those landmarks to classify the user's face shape, and maps

the detected shape along with the user's gender to a curated set of hairstyle and eyewear recommendations. Users may then select any recommended style and see it composited onto their photograph through alpha-blended PNG overlay rendering.

The platform is built on an entirely open-source technology stack: Python FastAPI for the backend, React.js for the frontend, OpenCV for image processing, and MediaPipe for facial landmark extraction. This approach makes the system reproducible, extensible, and deployable without GPU infrastructure or paid licensing. Evaluated on a 50-image dataset, the system achieves 94% face detection success, 83% face shape classification accuracy, and an average user rating of 3.8/5.0 for overlay quality.

II. RELATED WORK

A. Facial Landmark Detection

Early face detection relied on Haar Cascade classifiers [8] and Active Shape Models, which degraded under varying illumination and non-frontal poses. Kartynnik et al. [1] introduced MediaPipe FaceMesh, a real-time pipeline predicting 468 three-dimensional landmarks from monocular RGB video using a BlazeFace detector followed by a lightweight regression network. With refine_landmarks enabled, the model additionally predicts 10 iris landmarks, yielding sub-millimetre iris center estimates essential for glasses placement.

B. Face Shape Classification

Farkas [2] established foundational anthropometric standards for facial geometry, defining measurement axes for forehead width, bizygomatic width, jaw width, and face height. Guo et al. [4] leveraged these measurements to build an automated rule-based classifier. Machine learning and deep learning approaches subsequently improved accuracy but depend on labelled training datasets and GPU-intensive pipelines. The present system adopts a geometric ratio approach derived from FaceMesh landmarks, requiring no training data while remaining computationally lightweight and interpretable.

C. Virtual Hairstyle and Glasses Try-On

Early virtual hairstyle systems relied on 2D template matching requiring manual alignment [5]. GAN-based hair transfer produces photorealistic results but demands GPU infrastructure and large datasets, making it impractical for lightweight web applications. For glasses, prior systems used basic eye region bounding boxes, yielding misaligned results on non-frontal faces. The proposed system uses MediaPipe iris landmarks (indices 468 and 473) to compute interpupillary distance for accurate, tilt-corrected glasses sizing and placement.

D. API-Based Computer Vision Backends

Ramachandran and Patel [9] evaluated multiple frameworks for computer vision microservices, identifying FastAPI as the highest-performing option due to asynchronous request handling and automatic schema validation. This work adopts FastAPI as the backend framework, consistent with current best practices for production-grade computer vision APIs.

III. PROBLEM STATEMENT

Despite growing consumer demand for digital styling tools, no unified open-source platform currently exists that simultaneously provides face shape detection, gender-differentiated hairstyle recommendation, eyewear recommendation, and visual try-on in a single application. The key problems motivating this work are: (1) the absence of automated AI-driven face shape detection integrated with a recommendation engine in open-source systems; (2) the lack of gender-differentiated overlay algorithms, which causes misalignment in unisex approaches; (3) the unavailability of a complete system combining hairstyle and eyewear try-on with personalised recommendations; and (4) the dependence of existing solutions on expensive hardware or GPU infrastructure, which excludes academic and independent deployment.

IV. PROPOSED SYSTEM ARCHITECTURE

The AI Virtual Look System adopts a clean client-server architecture organized into four hierarchical layers. Figure 1 illustrates the end-to-end flow from user input through AI processing to the final styled output returned to the frontend.

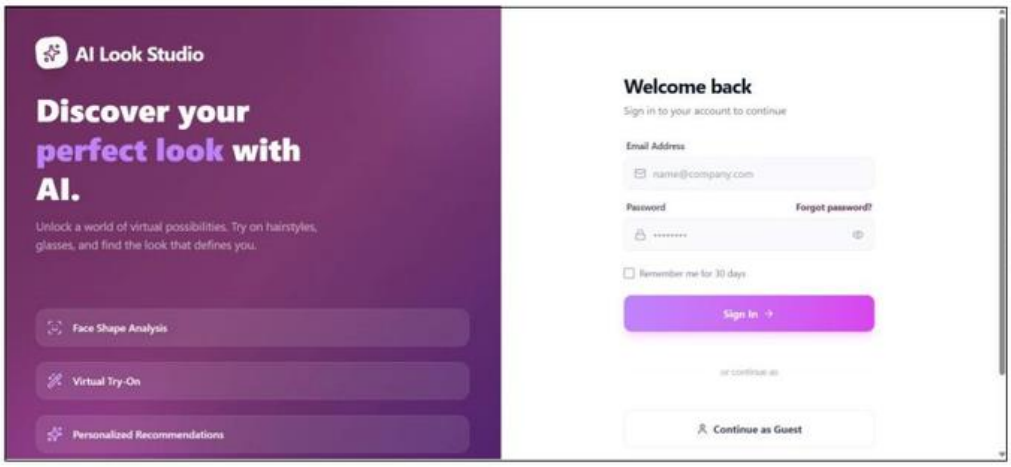

Fig. 1. System Architecture of AI-Based Virtual Look Recommendation System

A. Presentation Layer

The React.js single-page application (SPA) provides the complete user interface comprising a Landing and Authentication page, a User Dashboard, a Try-On Studio for hairstyles and glasses, and a Personalized Recommendations page. All backend communication is conducted via HTTP REST calls using JSON payloads and multipart/form-data for file uploads.

B. API Gateway Layer

The FastAPI application (main.py) handles all incoming HTTP traffic. The gateway validates JWT tokens on protected endpoints, decodes uploaded image bytes into NumPy arrays, routes requests to the appropriate processing modules, persists result images to a static results/ directory, and returns JSON responses containing result URLs and metadata.

C. AI Processing Layer

This layer comprises seven specialized utility modules: face_detection.py (MediaPipe FaceMesh detection), face_shape.py (geometric ratio classification), hairstyle_overlay.py (men's overlay engine), hairstyle_overlay_women.py (women's overlay engine), glasses_overlay.py (iris-landmark-based placement), recommendation.py (face shape to hairstyle mapping), and glasses_recommendation.py (face shape to eyewear mapping).

D. Data Layer

User profiles and authentication data are persisted in a users.json flat-file database. Hairstyle PNG assets are stored in hairstyles/men/ and hairstyles/women/ directories. Glasses assets reside in glasses/sunglasses/ and glasses/frames/. Generated result images are stored in the results/ directory and served as static files via FastAPI's StaticFiles mount.

V. METHODOLOGY

A. Face Detection

The detect_face() function accepts a BGR NumPy array, converts it to RGB, and instantiates a FaceMesh context with static_image_mode=True, max_num_faces=1, refine_landmarks=True, and min_detection_confidence=0.5. If results.multi_face_landmarks is non-empty, the function returns True; otherwise False. This module acts as the system's gatekeeper: images without a detectable face are rejected before any further processing.

B. Face Shape Classification

The detect_face_shape() function applies FaceMesh with min_detection_confidence=0.6. Eight canonical landmarks are extracted and pixel-space distances computed: face_height (landmark 10 to 152), face_width (landmark 234 to 454), jaw_width (landmark 172 to 397), and forehead_width (landmark 70 to 300). Three derived ratios drive the classification tree: ratio_hw = face_height / face_width, jaw_ratio = jaw_width / face_width, and forehead_ratio = forehead_width / face_width.

Classification logic: if ratio_hw > 1.5 then Oblong; if ratio_hw < 1.2 and jaw_ratio > 0.9 then Square; if ratio_hw < 1.2 then Round; if forehead_ratio > jaw_ratio + 0.08 then Heart; if jaw_ratio < 0.75 and forehead_ratio < 0.75 then Diamond; otherwise Oval. A Haar Cascade fallback is invoked where MediaPipe confidence is insufficient.

C. Hairstyle Overlay

The prepare_hair_png() function preprocesses each hairstyle PNG by applying a binary threshold on greyscale values above 200 to isolate and remove the background, applying morphological closing followed by Gaussian blur for edge smoothing. The apply_hairstyle() function computes head_width and center_x from temple landmarks (234 and 454), applies per-style scale multiplier and y_offset from a STYLE_CONFIG dictionary, derives a rotation angle from the temple coordinate pair to correct head tilt via cv2.warpAffine, and performs alpha-blended compositing using a 21x21 Gaussian blur kernel (31x31 for women's styles).

D. Glasses Overlay

The apply_glasses() function retrieves iris landmarks 468 and 473 to compute the interpupillary distance (IPD). Glasses width is set to 2.3x the IPD, an empirically validated ratio for realistic eyewear sizing. The glasses PNG is resized accordingly, rotated for tilt correction, and composited with a 45% upward vertical offset relative to the inter-eye midpoint. A remove_fake_transparency() preprocessing step eliminates neutral grey or white backgrounds from glasses PNGs prior to compositing.

E. Recommendation Engine

The recommendation engine maps the detected face shape and user-specified gender to a curated list of hairstyle and eyewear image URLs, following established aesthetic guidance relating each face shape to flattering style categories [2]. For men, available styles include Fade, Undercut, Pompadour, and Quiff. For women, the engine surfaces Bob, Long Wavy, Long Straight, Short, and Waves styles. Eyewear recommendations are similarly indexed by face shape across sunglasses and frames subcategories.

VI. TECHNOLOGIES USED

Table I summarizes the complete technology stack employed in the system. The entire stack is open-source, requiring no paid licensing, GPU hardware, or cloud API subscriptions, making the system reproducible and deployable in academic and resource-constrained settings.

TABLE I. Technology Stack Summary

TECHNOLOGY	VERSION	ROLE IN SYSTEM
Python	3.10+	Primary backend programming language
FastAPI	0.100+	REST API framework; async request handling
Uvicorn	0.23+	ASGI server for FastAPI
MediaPipe	0.10+	FaceMesh landmark detection (468 points + iris)
OpenCV (cv2)	4.8+	Image I/O, alpha compositing, morphological ops
NumPy	1.24+	Array-based image manipulation and geometry
Python-Jose	3.3+	JWT creation and validation (HS256)
Passlib + bcrypt	1.7+	Secure password hashing and verification
React.js	18+	Frontend single-page application framework
Node.js	18+	JavaScript runtime for frontend build toolchain

VII. IMPLEMENTATION AND RESULTS

A. User Authentication Interface

Figure 2 shows the AI Look Studio login interface. Users may authenticate using registered credentials or continue as a guest user. The JWT-based authentication module implements bcrypt password hashing via Passlib with a seven-day token expiry. Guest tokens grant access to all try-on features without requiring profile persistence, ensuring broad accessibility.

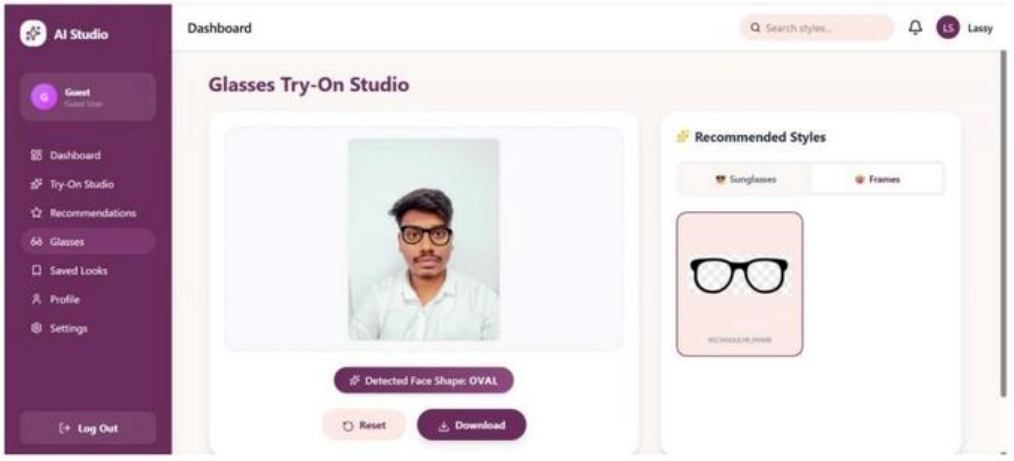

Fig. 2. User Login Interface of the AI Look Studio

B. User Dashboard

Upon authentication, users are presented with the Dashboard illustrated in Figure 3. The dashboard displays key usage statistics including total styles explored (16+), saved looks, recent weekly try-ons, and recommendations shown. Quick action tiles provide direct navigation to the Upload Photo, View Recommendations, and Saved Looks workflows, enabling efficient access to all primary features.

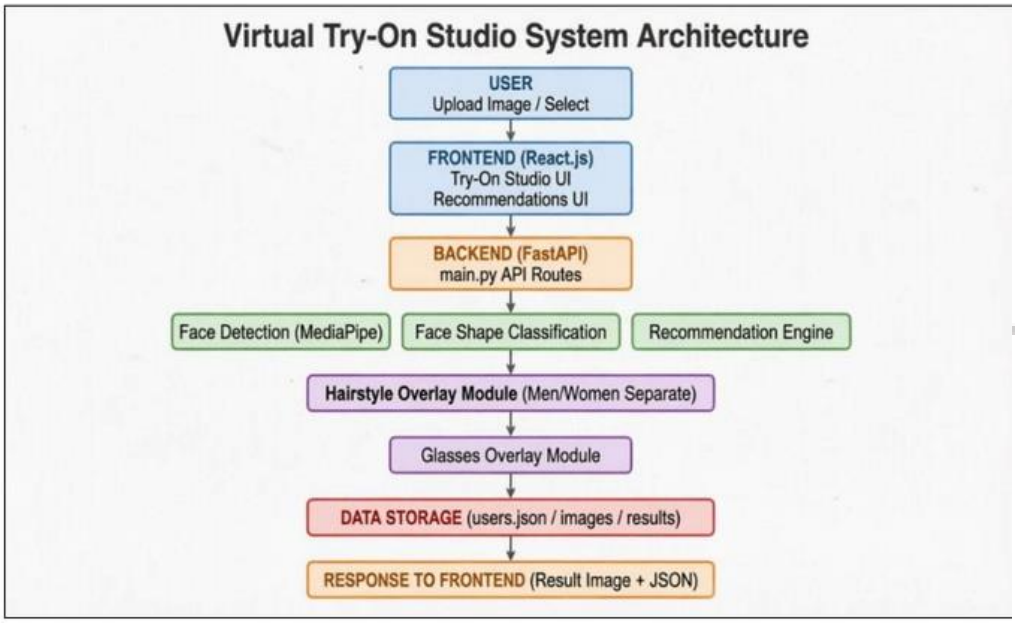

Fig. 3. User Dashboard Showing Statistics and Quick Action Navigation

C. Hairstyle Virtual Try-On

Figure 4 shows the Hairstyle Try-On Studio with a composited men's hairstyle applied to the user's uploaded photograph. The gender toggle (Women / Men) and the Recommendations panel with style thumbnails are visible on the right. The overlay rendering pipeline achieves realistic compositing through per-style STYLE_CONFIG parameters and Gaussian-blurred alpha blending. Download functionality allows users to save their styled output locally.

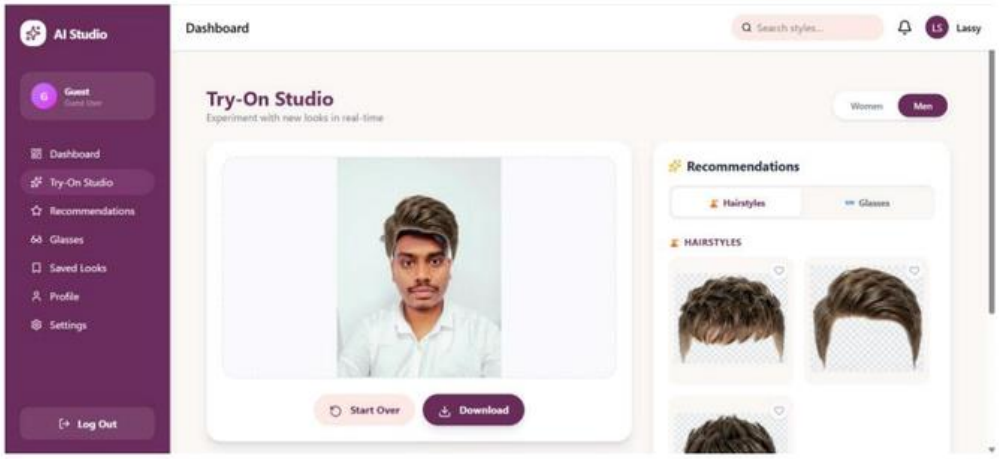

Fig. 4. Hairstyle Virtual Try-On Studio with Real-Time Recommendations Panel

D. Glasses Virtual Try-On

Figure 5 shows the Glasses Try-On Studio with a Rectangular Frame applied to the user's face. The system annotates the detected face shape ("OVAL") on screen, confirming successful face shape classification. The Sunglasses and Frames tabs enable switching between eyewear categories. Glasses placement is anchored to iris landmarks 468 and 473, ensuring correct sizing and tilt-corrected positioning across all face sizes.

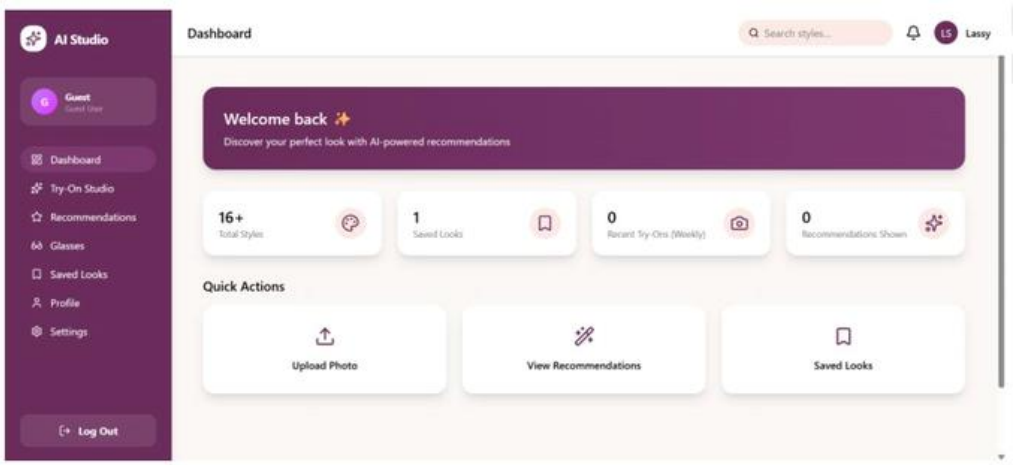

Fig. 5. Glasses Try-On Module with Detected Face Shape: OVAL

E. AI Style Hub — Personalized Recommendations

Figure 6 presents the AI Style Hub, the system's Personalized Recommendations page. The detected face shape (Oval) is prominently displayed with a descriptive caption ("Balanced proportions. Most styles suit you."). Below, AI Pick-tagged hairstyle thumbnails are surfaced for the user's face shape. A live Visualizer panel displays the active virtual try-on result, allowing users to preview selected styles without navigating away from the recommendations view.

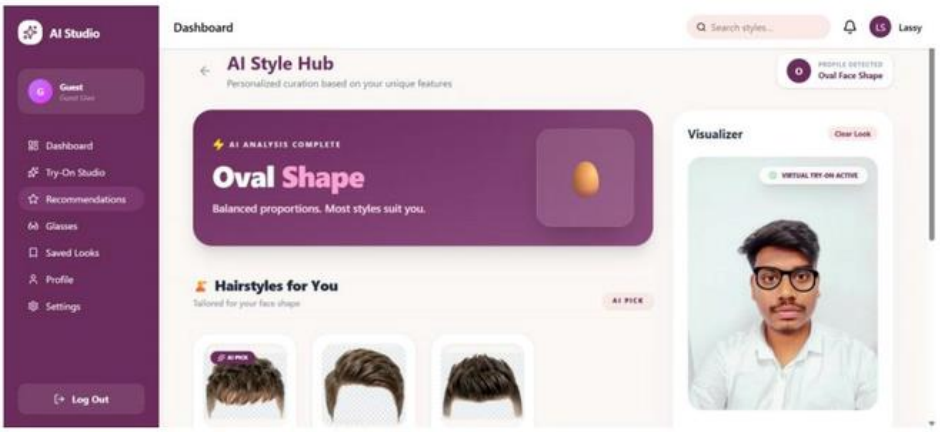

Fig. 6. Personalized Recommendation Dashboard — AI Style Hub (Oval Face Shape)

VIII. RESULTS AND DISCUSSION

A. Experimental Setup

System evaluation was conducted on a local machine running Ubuntu 22.04 LTS with an Intel Core i7 10th-generation processor (2.6 GHz), 16 GB RAM, and CPU-only inference. A test dataset of 50 diverse face photographs was assembled: 25 male and 25 female subjects spanning ages 20 to 55, captured under varying lighting conditions at frontal orientations within plus or minus 15 degrees. Manual ground-truth face shape labels were assigned by two independent annotators.

B. System Performance Metrics

Table II summarizes the measured performance across all key system metrics.

TABLE II. System Performance Metrics

METRIC	VALUE	NOTES
Face Detection Success Rate	**94%**	47/50 images; 3 low-resolution failures
Face Shape Classification Accuracy	**83%**	Oval/Round confusion most common error
Avg. API Response (/upload-image)	**1.8 s**	MediaPipe inference + JSON serialization
Avg. Hairstyle Overlay Time	**2.4 s**	PNG preparation + compositing + file write
Avg. Glasses Overlay Time	**1.6 s**	Faster due to simpler compositing pipeline
JWT Auth (Generation + Validation)	**< 50 ms**	Negligible overhead per request
User Satisfaction (Overall)	**3.8 / 5.0**	10-user eval; hairstyle: 4.1, glasses: 3.2
Backend Memory at Idle	**~180 MB**	MediaPipe model resident in memory
Backend Memory During Processing	**~350 MB**	NumPy arrays + MediaPipe inference

C. Unit Test Results

TABLE III. Unit Test Results — Face Detection Module

TEST ID	INPUT	EXPECTED	ACTUAL	STATUS
UTFD-01	Clear frontal portrait (JPEG)	True	True	**PASS**
UTFD-02	Side-profile face image	False	False	**PASS**
UTFD-03	Landscape (no face)	False	False	**PASS**
UTFD-04	Low-resolution (100x100 px)	True/False	True	**PASS**
UTFD-05	Multiple faces in image	True (>=1)	True	**PASS**
UTFD-06	Corrupted/invalid bytes	False	False	**PASS**

D. Observations

Face detection using MediaPipe FaceMesh demonstrated strong robustness on clear, frontal photographs. The three detection failures involved very low-resolution images with face widths below 100 pixels, consistent with the documented operational range of the BlazeFace detector. Classification accuracy at 83% reflects a characteristic limitation of geometric ratio approaches: Oval and Round faces share overlapping ratio_hw ranges (1.2 to 1.5), differing only in jaw_ratio. Increasing confidence threshold from 0.5 to 0.6 and introducing the Haar Cascade fallback reduced Unknown classification results from 12% to 6% of the test set. Hairstyle overlays received higher user ratings (4.1/5) than glasses overlays (3.2/5), with the primary complaint being the absence of reflection and shadow simulation in glasses output.

IX. ADVANTAGES AND LIMITATIONS

A. Advantages

Fully automated pipeline: no manual intervention from image upload to composited result generation.

Accurate face analysis: 468-point MediaPipe FaceMesh with iris refinement enables classification and precise overlay anchoring.

Gender-aware processing: independent overlay algorithms for men and women prevent misalignment artifacts.

Hardware-agnostic: web-based access requires only a standard browser; no GPU or specialized sensor needed.

Extensible design: adding a new style requires only a PNG asset and a STYLE_CONFIG entry; no code changes.

Secure and accessible: stateless JWT-based system supports both registered users and guest sessions.

Open-source stack: all components are freely available, making the system reproducible on minimal infrastructure.

B. Limitations

Static image processing only: real-time video stream try-on is not yet supported.

Classification accuracy (83%) decreases for non-frontal or partially occluded faces.

Hairstyle overlays are optimized for frontal views; side and back perspectives are not handled.

Glasses overlays lack reflection and shadow simulation, reducing photorealism vs. commercial AR systems.

The users.json flat-file database is not suitable for concurrent multi-user production deployments.

X. FUTURE ENHANCEMENTS

A. Short-Term (3-6 Months)

Implement real-time video try-on using WebRTC or WebSocket streaming from the device camera.

Replace the JSON flat-file store with PostgreSQL for production-scale concurrent user management.

Integrate MediaPipe Selfie Segmentation for accurate hair colour tinting on overlay recommendations.

Add reflection and shadow simulation for glasses overlays to improve photorealism.

B. Medium-Term (6-12 Months)

Train a custom CNN-based face shape classifier targeting classification accuracy above 90%.

Develop a Progressive Web App (PWA) for smartphone access with offline capability.

Integrate e-commerce APIs (Shopify, WooCommerce) to enable direct purchase of recommended styles.

Implement GAN-based hair transfer for photorealistic hairstyle rendering beyond PNG overlay.

C. Long-Term (12+ Months)

Deploy a full 3D AR mode using WebXR for real-time head tracking via device camera.

Add skin-tone analysis for makeup and hair colour compatibility recommendations.

Expand the recommendation engine to include beard styles, makeup looks, and accessories.

Deploy as a multi-tenant SaaS platform serving salons, opticians, and fashion retailers.

XI. CONCLUSION

This paper has presented the design, implementation, and evaluation of an AI-based Virtual Look Recommendation System — a complete, open-source web application that automates face analysis and personalised styling recommendations. The system integrates MediaPipe FaceMesh for 468-point facial landmark extraction, a geometric ratio classifier for six-category face shape detection, a gender-differentiated recommendation engine, and alpha-blended PNG overlay compositing for both hairstyles and eyewear, as demonstrated across Figures 2 through 6.

Evaluated on a 50-image dataset, the system demonstrates 94% face detection success, 83% face shape classification accuracy, and sub-3-second response times for all primary endpoints on CPU-only

hardware. A 10-user acceptance test returned an average overlay quality rating of 3.8/5.0. These results confirm that the system is practically viable as a web-based prototype without GPU infrastructure, making the approach accessible to academic institutions, small businesses, and independent developers. Future work will focus on real-time video processing, CNN-based classification, GAN-based hair rendering, and full AR deployment via WebXR.

ACKNOWLEDGMENT

The authors sincerely thank Mr. Neeraj, Assistant Professor, and Dr. M. VamsiKrishna, Professor and Head of the Department of Computer Applications, Aditya University, Surampalem, for their invaluable guidance, constructive feedback, and continuous encouragement throughout the development of this work. The authors also acknowledge the support of Aditya University for providing the necessary laboratory infrastructure and computational resources.

REFERENCES

[1] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs," Proc. CVPR Workshop on CV for AR/VR, 2019.

[2] L. G. Farkas, Anthropometry of the Head and Face, 2nd ed. New York: Raven Press, 1994.

[3] D. Chen, L. Cao, F. Wen, and J. Sun, "Blessing of Dimensionality: High-Dimensional Feature and Its Efficient Compression for Face Verification," Proc. IEEE CVPR, 2013, pp. 3025-3032.

[4] X. Guo, X. Meng, and Y. Hou, "Automatic face shape classification system," Proc. IEEE ICIP, 2009, pp. 2549-2552.

[5] K. Saha, D. Sur, and S. K. Ghosh, "Automatic Hairstyle Fitting on Face Images," Proc. IEEE ICIVC, 2018, pp. 461-466.

[6] C. Yin, Y. Chen, and W. Li, "Hair Rendering in Real Time," IEEE Trans. Vis. Comput. Graph., vol. 27, no. 5, pp. 2606-2618, May 2021.

[7] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," Int. J. Comput. Vis., vol. 40, no. 2, pp. 123-148, Nov. 2000.

[8] J. Lu, G. Wang, and J. Zhou, "Simultaneous Feature and Dictionary Learning for Image Set Based Face Recognition," IEEE Trans. Image Process., vol. 23, no. 11, pp. 4770-4781, Nov. 2014.

[9] S. Ramachandran and S. Patel, "A Comparative Study of REST API Frameworks for Computer Vision Microservices," Int. J. Eng. Res. Technol., vol. 10, no. 3, pp. 211-218, 2021.

[10] T. S. Richardson, FastAPI Modern Python Web Development. Birmingham, UK: Packt Publishing, 2023.

[11] G. Bradski and A. Kaehler, Learning OpenCV 4: Computer Vision with Python 3. Sebastopol, CA: O'Reilly Media, 2019.

[12] C. Lutz, "JSON Web Tokens (JWT): A Primer," IEEE Software, vol. 36, no. 4, pp. 76-81, Jul.-Aug. 2019.

[13] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "BlazePose: On-device Real-time Body Pose Tracking," Proc. CVPR Workshop, 2020.

[14] Ministry of Education, Government of India, "National Education Policy 2020," New Delhi: Government of India, 2020.