StyleBlend: A Comprehensive AI Platform for Virtual Hairstyle Transformation and Personalized Recommendations

Palle Prabhas Reddy
Department of Computer Science
PES University
Bengaluru, India

Pari Maheshwari
Department of Computer Science
PES University
Bengaluru, India

Prarthana Jyothi
Department of Computer Science
PES University
Bengaluru, India

Aditya Johnson Stanley
Department of Computer Science
PES University
Bengaluru, India

Suresh Jamadagni
Department of Computer Science
PES University
Bengaluru, India

ABSTRACT

The technology behind StyleBlend combines deep learning, image processing, and real-time social media analytics, creating a comprehensive, personalized experience for each user. With its sophisticated AI algorithms and multimodal approach, StyleBlend addresses the common challenges of choosing a hairstyle by considering facial features, personal preferences, and the ever-changing fashion landscape. Whether users are seeking a dramatic change or just want to try out new trends, StyleBlend equips them with the insights and tools necessary to make well-informed hairstyle decisions.

General Terms

GAN, Similarity Score,Recommendation

Keywords

Faceshape classification, Trend Extraction, Virtual Try-on, StyleGAN, Recommendation Engine

1. INTRODUCTION

This paper details the design, implementation, and evaluation of StyleBlend, highlighting the methodologies and technologies that form its system architecture. Section 2 reviews related work in AI-driven virtual try-on systems, hairstyle recommendation models, and facial analysis techniques. Section 3 delves into the core methodologies behind StyleBlend, covering face shape classification, similarity analysis, and trend extraction from social media. Section 4 evaluates the system's performance through experimental results, focusing on metrics like accuracy and user satisfaction. Finally, Section 5 discusses the broader applications of StyleBlend in industries such as beauty, fashion, and self-care, along with future research directions in virtual grooming and AI-driven personalization.

2. RELATED WORK

Virtual hairstyle transformation and facial shape-based recommendations have seen substantial advancements in recent years, largely due to the evolution of Generative Adversarial Networks (GANs). Early models like GANs [1] and DCGAN [2] set the stage for image generation, but they struggled with stability and the quality of generated images. Later improvements, such as StyleGAN [3] and its subsequent iterations StyleGAN2 [4], introduced innovative architectures that allowed for the generation of highly realistic images with more control over style and identity features.

For hairstyle editing, recent works have explored various GAN-based models. MichiGAN [5] utilized a mask-conditioned GAN architecture for hair editing but faced limitations in generalizing across different hairstyle structures. LOHO [6], leveraging StyleGAN2, enabled the modification of hairstyles but demanded considerable computational power. The Barbershop system [7] introduced an image compositing method using segmentation masks, providing a more realistic hair generation approach.

In the context of hairstyle recommendation, face shape classification and latent space manipulation have become key strategies. CelebHair [8] presented a large-scale dataset to improve hairstyle recommendations based on celebrity images, while Face Shape Classification for hairstyle suggestions has been explored by various studies [9, 10]. HairCLIP [11] and HairCLIP v2 [12] introduced new strategies for hairstyle generation by blending features, while systems like Style Your Hair [13] employed latent optimization techniques to provide pose-invariant hairstyle transfers. The field continues to push for improved efficiency and the ability to generate hairstyles that match user preferences and face shapes.

3. SYSTEM ARCHITECTURE AND DESIGN

StyleBlend is powered by AI that uses deep learning and sophisticated image processing techniques to offer realistic virtual hairstyle changes, tailored hairstyle suggestions, and the latest hairstyle trends. Its architecture is modular, with each part playing a unique but connected role, allowing for seamless data flow and efficient processing.

3.1 Overall Architecture

The architecture of StyleBlend is built around five key components.

—Face Shape Classification: This component assesses the user's face shape to identify the most flattering hairstyles. It employs a convolutional neural network (CNN) for precise categorization of face shapes.

—Face Similarity Analysis: This feature looks for individuals with comparable facial characteristics from a curated database, assisting StyleBlend in suggesting styles that align with the user's unique facial structure.

—Recommendation Engine: This engine combines face shape classification, similarity analysis, and popular hairstyles to generate a tailored list of hairstyles that suit the user's face shape and similarity score.

—Virtual Try-On: This module displays the recommended hairstyles on the user's face in real time, using StyleGAN-based transformations to ensure the final image remains authentic and high-quality.

—Trend Extraction: This function gathers trending hairstyles from social media, regularly refreshing the recommendation pool to stay in tune with current style trends.
Figure 1 is a high-level diagram that illustrates how these modules interact:

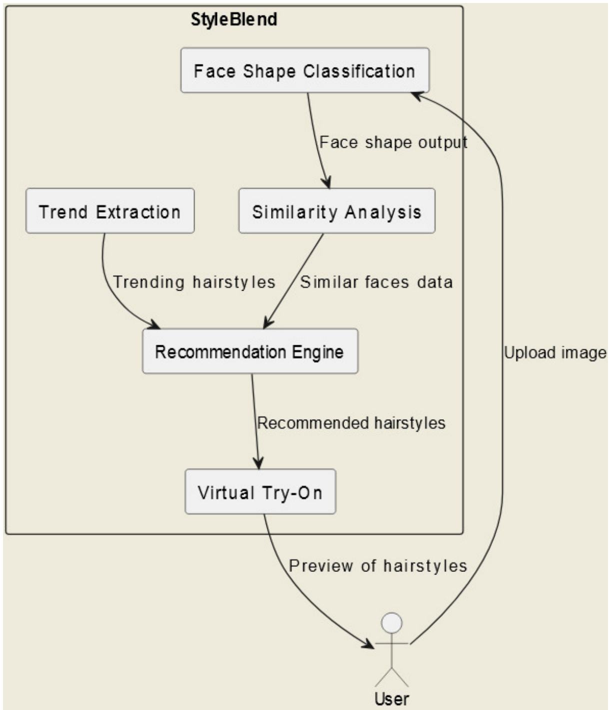

Fig. 1. High level module interaction diagram and execution workflow

3.2 Module Descriptions

3.2.1 Module for Classifying Face Shapes. In order to provide customized hairstyle recommendations, this module classifies the user's face into predetermined forms (round, oval, square, heart, or diamond).
Process and Techniques:

—Facial Landmark Analysis – Quantifies features such as jawline and cheekbones.

—Geometric Ratios – Quantifies shape by proportional ratios.

—CNN-Based Classification – A Convolutional Neural Network identifies patterns and classifies shapes.

Challenges Addressed: The model uses data augmentation for increased consistency in order to guarantee dependability across different head orientations, lighting conditions, and image quality.

3.2.2 Similarity Analysis Module. This module identifies hairstyles that have worked well for users with similar facial traits, making the recommendations more relevant.
Process and Techniques: Faces are encoded into embeddings by a Siamese neural network, which then uses metrics like cosine similarity to evaluate similarity.
Challenges Addressed: Biases across face types, ethnicities, and hairstyles are reduced in a varied dataset. For precise matching, fine-tuned embeddings minimize small variances.
Figure 2 is a diagram illustrating the Face Shape Detection Model

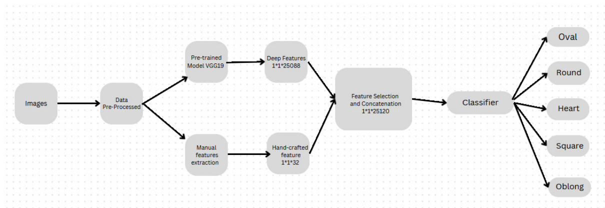

Fig. 2. Face Shape Detection and classification Model

3.2.3 Recommendation Engine. This engine uses a combination of collaborative and content-based filtering to recommend customized looks by combining hairstyle trends, face shape categorization, and similarity analysis.
Process and Techniques: In order to generate pertinent hairstyle recommendations, the engine combines face shape, likeness analysis, and popular social media hairstyles with content-based and collaborative filtering.
Challenges Addressed: Dynamic weighing strikes a balance between trends and personalization to provide recommendations that are both current and stylish.
Figure 3 is a recommendation flow diagram

3.2.4 Virtual Try-On. This module allows users to preview suggested hairstyles on their photos using StyleGAN.
Process and Techniques: Maintains textures and details while integrating hairstyles in a seamless manner. For realistic blending, GAN Inversion maps user photos into latent space.
Challenges Addressed: Uses GAN inversion and loss functions centered on local style matching to guarantee pose-invariant hairdo transfer.

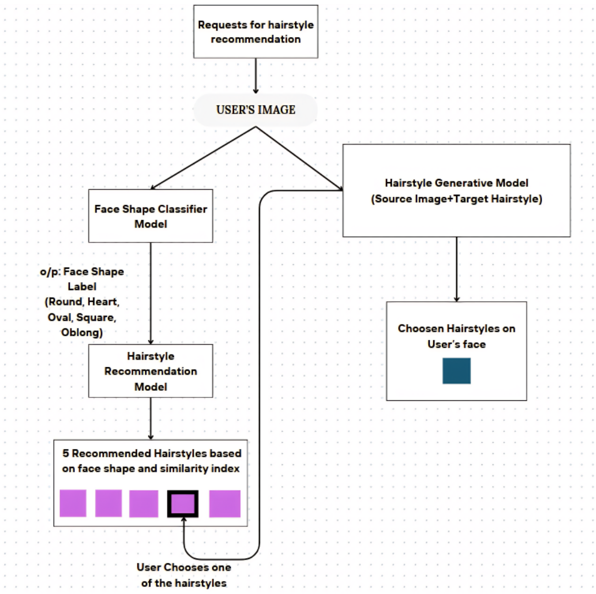

Fig. 3. Recommendation Engine Flow Diagram

3.2.5 Trend Extraction. Maintains hairstyle recommendations current by analyzing social media trends (such as Instagram).
Challenges Addressed: Maintains relevance to broad user preferences while balancing trend diversity.
Figure 4 below is a trend extraction flow diagram

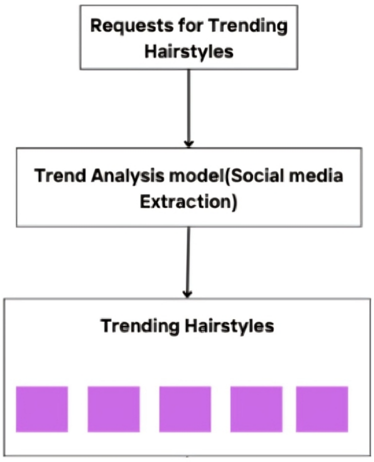

Fig. 4. Extraction of trends from social media flow diagram

4. DATASET AND IMAGE PRE-PROCESSING

4.1 Dataset

StyleBlend combines well-known and unique datasets to provide face classification, similarity analysis, and hairstyle recommendations based on comprehensive, high-quality datasets.

4.1.1 Flickr-Faces-HQ (FFHQ) Dataset. Overview: An NVIDIA high-resolution (1024x1024) dataset from Flickr that is perfect for StyleGAN-based facial alterations with a variety of facial features.
Applications in StyleBlend:

—Face Shape Classification: FFHQ images are utilized to train and validate the face shape classification model, providing a foundation for recognizing diverse facial structures.

—Virtual Try-On Transformation: The dataset's high resolution and diversity allow StyleGAN to learn a wide array of facial features, enabling high-quality, realistic transformations.

4.1.2 Custom Hairstyle Dataset. Overview: StyleBlend features a distinct and curated dataset of 400 hairstyles, sourced primarily from the FFHQ dataset and open-source image libraries. Each hairstyle image is selected for compatibility with specific face shapes, essential for generating personalized recommendations.
Data Collection and Curation:

—Image Sources: Most images come from the FFHQ dataset, ideal for virtual overlays. Additional images are sourced from open-source, license-free libraries and public domain resources to capture trending, diverse styles.

—Selection Criteria: Hairstyles are categorized by face shape (e.g., oval, round, square) based on stylist guidelines and facial geometry research.

Characteristics:

—Resolution: Images are kept at high resolutions (512x512 to 1024x1024) to ensure clarity in virtual overlays.

—Face Shape Categories: Each hairstyle is tagged with compatible face shapes, aligning recommendations with user facial features.

—Diversity in Styles: Styles include short, medium, and long hair, featuring options like bobs, braids, and layered cuts.

Applications in StyleBlend:

—Similarity and Recommendation Module: Categorizing each hairstyle by face shape compatibility enhances recommendation relevance and accuracy.

4.1.3 Trending Hairstyles Collection. Overview: This dataset is dynamically updated through the Trend Extraction module, analyzing social media platforms such as Instagram to identify popular hairstyles, contributing to a trend-driven selection for StyleBlend's recommendation engine.
Data Collection and Characteristics:

—Source: Data is collected from social media platforms focusing on images tagged with popular hairstyle-related hashtags.

—Trend Analysis: NLP techniques analyze hashtags, captions, and engagement metrics to pinpoint and filter highly engaged styles.

Applications in StyleBlend:

—Recommendation Engine: Trending styles are integrated into recommendations, ensuring access to the latest hairstyle trends.

4.2 Image Preprocessing for Consistent Input Format

Guarantees consistent size, alignment, and metadata tagging for successful virtual conversions. The preprocessing module consists of the following key stages:

4.2.1 Face Detection and Landmark Identification. Identifies the eyes, nose, and mouth—three important facial features—for accurate alignment and steady orientation using dlib's face detector and 68-point landmark predictor.

The process starts with loading an image and identifying any faces present in it. For a given image I, the face detector generates bounding boxes $B_i = (x_i, y_i, w_i, h_i)$ for each detected face, where x_i and y_i denote the coordinates of the top-left corner, and w_i and h_i represent the width and height of the bounding box, respectively.

4.2.2 Face Alignment Using Landmark Points. Horizontal eye alignment ensures standardized orientation. The center locations of the left and right eyes are used to calculate the rotation angle. Let C_{left} and C_{right} represent the center points of the left and right eyes, respectively. The rotation angle θ can be computed as:

$$\theta = \arctan\left(\frac{C_{\text{right},y} - C_{\text{left},y}}{C_{\text{right},x} - C_{\text{left},x}}\right) \tag{1}$$

The image I is then rotated by $-\theta$ degrees using an affine transformation matrix to ensure that the eyes are horizontally aligned. After rotation, the aligned face is re-detected to confirm accurate bounding and to apply a consistent scale.

4.2.3 Symmetric Cropping with Padding. By symmetrically padding the aligned face and creating room around the hair and neck for a natural look, it guarantees consistent framing. Let w and h denote the width and height of the face bounding box. Horizontal padding p_x and vertical padding p_y are calculated as follows:

$$p_x = w \times \text{padding\_factor} \tag{2}$$

$$p_{y_{\text{top}}} = h \times \text{top\_padding\_factor} \tag{3}$$

$$p_{y_{\text{bottom}}} = h \times \text{bottom\_padding\_factor} \tag{4}$$

where the padding factors are set based on empirical observations to adequately cover both the hair and neck regions. The resulting bounding box B' is then utilized to crop the aligned image symmetrically around the face.

4.2.4 Resizing to Target Dimensions (FFHQ Format). Resized to 1024 x 1024 pixels, the cropped and padded face is centered to guarantee consistent formatting for the transformation process. This keeps the input format square and consistent, ensuring conformity with StyleGAN's criteria.

4.2.5 Metadata Generation. Per-image metadata adds richness to dataset management. It includes file properties (size, MD5 checksum, dimensions) for integrity and face landmarks (dlib coordinates) for tracking. Stored as JSON, it is easily retrievable and traceable. Figure 5 represents metadata of the extracted image

4.2.6 Batch Processing for Scalable Preprocessing. Identifies, aligns, and crops several photos before saving them in FFHQ format. Standardized inputs for StyleBlend's virtual try-on are guaranteed by this scalable framework.

5. METHODOLOGY

5.1 Face Shape Classification

In order to ensure that hairstyle recommendations complement facial structure, images are classified into round, oval, square, heart, or diamond shapes based on facial feature analysis.


```
{
    "file_url": "/output_folder/aligned_image.png",
    "file_size": 204800,
    "file_md5": "9e107d9d372bb6826bd81d3542a419d6",
    "pixel_size": [1024, 1024],
    "face_landmarks": [
        {"x": 100, "y": 150}, {"x": 110, "y": 155}, ...
    ]
}
```


Fig. 5. Json Metadata of images used for training

Table 1. Average Facial Ratios for Each Face Shape

Face Shape	W/H	J/C	F/C
Round	1.0	0.9	1.0
Oval	1.3	1.1	1.2
Square	1.2	1.2	1.0
Heart	1.1	0.8	1.2
Diamond	1.3	1.0	0.9

5.1.1 Landmark Detection and Feature Extraction. A landmark detection model identifies specific points of the jawline, cheekbones, and forehead and computes geometric ratios to distinguish face shapes.

Let $P_i(x_i, y_i)$ represent the coordinates of facial landmarks, where $P_i(x_i, y_i)$ denotes specific landmarks such as the cheekbones, jawline, and forehead points. The following facial proportions are calculated:

Width-to-Height Ratio (W/H): This measures the width of the face, specifically the distance between the cheekbones, compared to the height of the face, which is the distance from the forehead to the chin.

$$\frac{H}{W} = \frac{\text{Distance between } P_{\text{forehead}} \text{ and } P_{\text{chin}}}{\text{Distance between } P_{\text{left cheekbone}} \text{ and } P_{\text{right cheekbone}}} \tag{5}$$

Jaw-to-Cheekbone Ratio (J/C): This ratio looks at the width of the jawline in relation to the width of the cheekbones.

$$\frac{C}{J} = \frac{\text{Distance between } P_{\text{left cheekbone}} \text{ and } P_{\text{right cheekbone}}}{\text{Distance between } P_{\text{left jaw}} \text{ and } P_{\text{right jaw}}} \tag{6}$$

Forehead-to-Cheekbone Ratio (F/C): This ratio compares the width of the forehead to the width of the cheekbones.

$$\frac{C}{F} = \frac{\text{Distance between } P_{\text{left cheekbone}} \text{ and } P_{\text{right cheekbone}}}{\text{Distance between } P_{\text{left forehead}} \text{ and } P_{\text{right forehead}}} \tag{7}$$

These ratios are compiled into a feature vector $\vec{R} = (W/H, J/C, F/C)$, forming the foundation for classifying different face shapes. Table 1 presents the average proportions for each face shape based on these ratios.

5.1.2 Convolutional Neural Network (CNN) for Classification. To recognize facial shapes, a CNN model uses preprocessed images to learn spatial features. Robustness is increased through data augmentation (rotation, scaling, flipping).

For precise classification, the Adam optimizer is used in conjunction with cross-entropy loss. Let $\hat{y}_i$ represent the predicted proba-

Table 2. Classification Metrics for Face Shape Categories

Metric	Round	Oval	Square	Heart	Diamond	Average
Precision	89%	87%	88%	86%	90%	88%
Recall	88%	89%	87%	85%	92%	88.2%
F1 Score	88.5%	88%	87.5%	85.5%	91%	88.3%
Accuracy	88.3%					

bility for each face shape class, and y_i denote the true label. The cross-entropy loss function L is defined as follows:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{8}$$

where C indicates the number of face shape classes. To avoid overfitting, Incorporated regularization techniques such as dropout and weight decay.

5.1.3 Classification Metrics. The performance of the classifier is assessed using metrics such as accuracy, precision, recall, and F1 score. Table 2 provides a summary of the model's performance metrics for each face shape category.

5.2 Similarity Analysis for Recommendation

In order to suggest haircuts that closely fit the user's face shape category, their facial traits are matched with a database of previously tagged photos.

5.2.1 Face Embedding Extraction. Numerical facial embeddings are produced by DeepFace for insightful image comparison. These embeddings capture distinct facial traits and are produced using a pre-trained model such as VGG-Face. This embedding, represented as $\vec{E}$, enables us to quantify the visual similarity between two images. Given an image path P, the face embedding $\vec{E}$ for an image is obtained as follows:

$$\begin{aligned}\vec{E} = &\ \text{DeepFace.represent}(P, \text{model\_name} =' VGG - Face', \\ &\ \text{enforce\_detection} = \text{False})\end{aligned} \tag{9}$$

where `DeepFace.represent` computes a vector embedding for each face in the image.

5.2.2 Similarity Measurement. Use Manhattan distance, Euclidean distance, and cosine similarity to locate similar faces within the same shape category. It ensures robust comparison by capturing spatial and angular differences.
Let $\vec{E_1}$ and $\vec{E_2}$ represent the embeddings of the user and a target image, respectively. The individual similarity measures are calculated as follows:

5.2.2.1 Cosine Similarity.

$$\text{cos\_sim}(\vec{E_1}, \vec{E_2}) = 1 - \frac{\vec{E_1} \cdot \vec{E_2}}{\|\vec{E_1}\|\|\vec{E_2}\|} \tag{10}$$

5.2.2.2 Euclidean Distance.

$$\text{euclid\_dist}(\vec{E_1}, \vec{E_2}) = \|\vec{E_1} - \vec{E_2}\| \tag{11}$$

5.2.2.3 Manhattan Distance.

$$\text{manhattan\_dist}(\vec{E_1}, \vec{E_2}) = \sum_{i=1}^{N} |E_{1,i} - E_{2,i}| \tag{12}$$

5.2.3 Composite Similarity Score. Accurate face feature comparison is ensured via a weighted sum of Euclidean distance and cosine similarity, with Manhattan distance contributing less. With weights represented as $w = (w_1, w_2, w_3)$, the composite similarity score S is defined as:

$$\begin{aligned}S = w_1 \cdot \text{cos\_sim} + w_2 \cdot \left(1 - \frac{1}{1 + \text{euclid\_dist}}\right) \\ + w_3 \cdot \left(1 - \frac{1}{1 + \text{manhattan\_dist}}\right)\end{aligned} \tag{13}$$

In the implementation, utilized weights $w = (0.4, 0.4, 0.2)$, focusing on cosine and Euclidean measures to achieve a balanced consideration of both angle and magnitude differences.

5.2.4 Identification of Top Matches. The user's face embedding is compared to find the top five matches based on similarity scores.

5.2.5 User-Guided Selection Process. Users choose a target hairstyle from the top five for more precise virtual try-on.

5.2.6 Model Architecture. To recognize facial shapes, a CNN model uses preprocessed images to learn spatial features. Robustness is improved by data augmentation (rotation, scaling, flipping). For increased accuracy, the Adam optimizer and cross-entropy loss are used. Let $\hat{y}_i$ represent the predicted probability for each face shape class, and y_i denote the true label. The cross-entropy loss function L is defined as follows:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{14}$$

where C indicates the number of face shape classes. To avoid overfitting, Incorporated regularization techniques such as dropout and weight decay.

5.3 Trends Analysis

Obtained trending hairstyle images legally from publicly available Instagram posts through an automated, compliant process.

5.3.1 Automated Scraping setup. Data Collection: Utilized Selenium and GeckoDriverManager to query hashtags (e.g., #hairtrends2024) on Instagram for public images. Image Retrieval: Retrieved URLs from Instagram's CDN, scraping only public content.

5.3.2 Image Download and Storage. To facilitate offline analysis and recognize future trends, the collected image URLs were downloaded and organized in a structured format. A custom Python function, `download_image`, was created to download each image by accessing the URL and saving it to a specified local directory. Given an image URL U and the designated save path S, the download function operates as follows:

$$\texttt{download\_image}(U, S) \rightarrow \text{Image File at Path } S \tag{15}$$

Each image was saved with unique filenames (`image_{i}.jpg`), where i represents the image's index in the dataset. This naming convention allows for efficient file management and easy access.

5.3.3 Data Processing and Analysis. To improve hairstyle suggestions, images are preprocessed and grouped according to visual characteristics. For real-time recommendations, this method keeps the dataset current with social media trends.

5.4 Virtual Try-On

5.4.1 StyleGAN Architecture and Latent Space Manipulation. By distinguishing between facial structure and hairdo in its latent space, StyleGAN, a GAN created by NVIDIA, produces realistic and adaptable images.

5.4.1.1 Latent Space Interpolation. The user's face and hairstyle are encoded into embeddings by StyleGAN, which then modifies them to ensure a natural metamorphosis while blending the hairdo flawlessly.

5.4.1.2 Multi-Resolution Layers. The multi-scale structure of StyleGAN2 ensures high-resolution hairdo integration while maintaining fine facial characteristics.

5.4.1.3 Latent Code Generation. The mapping is defined as follows:

$$w = f(z) \tag{16}$$

where z represents the initial latent vector drawn from a normal distribution, and $f(z)$ is a non-linear function that converts the latent code z into the style latent space w.

5.4.1.4 Image Generation from Latent Code. After generating the latent code w, it is input into the generator G, which produces an image I in the image space X:

$$I = G(w) \tag{17}$$

where G denotes the generator network of StyleGAN, and w is the style vector for the input image, influencing visual features like hair type and color.

5.4.2 Face and Hair Segmentation. By isolating the hair area, segmentation techniques guarantee that changes in hairdo don't affect face features.

5.4.2.1 Pre-Trained Segmentation Models. By defining hair borders, models such as DeepLabV3 guarantee that changes only impact the hairstyle and preserve facial features.

5.4.2.2 Inpainting and Masking. Inpainting, which follows hair segmentation, closes in gaps and occlusions to guarantee that the hairstyle overlay flows naturally with the shapes and orientation of the head.

5.4.2.3 Segmentation Mask. Let M represent the binary segmentation mask, where each pixel $m_{i,j}$ in the mask is 1 if it is part of the hair region and 0 otherwise:

$$M = \{m_{i,j} \mid m_{i,j} = 1 \text{ if pixel}(i,j) \text{ is part of the hair, otherwise } 0\} \tag{18}$$

Using this mask, the system isolates the hair region H in the image I as:

$$H = I \cdot M \tag{19}$$

5.4.3 Pose Alignment and Blending. A significant challenge in virtual try-on is ensuring that the user's face aligns properly with the target hairstyle.

5.4.3.1 Alignment through GAN Inversion. Target and source images are aligned to the StyleGAN latent space for precise hair placement.

5.4.3.2 Pose Matching and Flow-Based Warping. i.e., HairFIT, warps the hairstyle to the user's head pose.

5.4.3.3 Pose Alignment Using Flow. Let F_{source} and F_{target} represent the flow fields for the source and target images, respectively. The alignment function A adjusts the target image based on the source pose:

$$A(I_{\text{target}}, F_{\text{source}}) = I_{\text{aligned}} \tag{20}$$

where I_{target} is the target hairstyle image, F_{source} is the flow field that aligns the target image to the source pose, and I_{aligned} is the final aligned image.

5.4.4 Local-Style Matching and Optimization. To ensure that the transformed hairstyle retains the intricate details of the desired style, local-style matching techniques are utilized.

5.4.4.1 Local-Style Loss Functions. During the transformation process, the model employs loss functions that penalize any deviations from the target hairstyle's texture and color. This ensures that features such as hair strands, curls, and texture patterns are accurately preserved:

$$L_{\text{style}} = \sum_{i,j} \|I_{\text{aligned}}(i,j) - I_{\text{target}}(i,j)\|^2 \tag{21}$$

where $I_{\text{aligned}}(i,j)$ is the pixel value of the aligned image at position (i,j), $I_{\text{target}}(i,j)$ is the pixel value of the target image at position (i,j), and $\|\cdot\|^2$ represents the Euclidean distance between the pixel values, ensuring the texture of the hairstyle is matched.

5.4.5 Blending and Final Optimization. After aligning and blending the hairstyle with the face, a final optimization step enhances the coherence and realism of the composite image.

5.4.5.1 Smooth Blending. The system employs alpha blending to seamlessly integrate the transformed hairstyle with the user's head and skin tone, eliminating any visible boundaries or mismatches. This process ensures consistent lighting and shading between the face and hairstyle.

5.4.5.2 Final GAN-Based Refinement. A final pass through StyleGAN fine-tunes the image to maintain a photo-realistic quality, correcting any lingering inconsistencies in color or texture between the face and hairstyle:

$$L_{\text{final}} = \lambda_{\text{content}} \cdot L_{\text{content}} + \lambda_{\text{style}} \cdot L_{\text{style}} \tag{22}$$

where L_{content} measures the similarity between the original face and the transformed face, L_{style} measures the similarity between the hairstyle textures, and λ_{content} and λ_{style} are the weights assigned to the content and style losses, respectively.

5.4.5.3 Blending Final Image. Let I_{hair} and I_{face} represent the hair and face images, respectively, and let α be the blending factor:

$$I_{\text{final}} = \alpha \cdot I_{\text{hair}} + (1 - \alpha) \cdot I_{\text{face}} \tag{23}$$

where I_{final} is the final image, and α is the blending coefficient that controls the weight of the hair and face images in the final composition.

6. IMPLEMENTATION DETAILS

6.1 Computing Resources

6.1.1 Software. Python with TensorFlow and PyTorch for deep learning, OpenCV for computer vision, and Dlib for facial landmark identification.

6.1.2 Cloud Services. Google Colab offers access to GPU (Google Colab's T4 GPU) and easy collaboration with no local hardware constraints.

6.2 APIs and External Services

6.2.1 DeepFace. DeepFace: Extracts face embeddings with pre-trained models (e.g., VGG-Face) to analyze similarity.

6.2.2 Google Drive. Saves datasets and model checkpoints to manage data with efficiency.

6.3 User Interface

StyleBlend offers a user-friendly web interface for seamless navigation. Users can:

—Upload Images: To classify faces.

—Face Shape and Hairstyle Recommendations: Receive hairstyle recommendations based on facial analysis.

—Virtual Try-On: Show them different hairstyles on their photos.

Tech Stack: React.js for a responsive user interface; Flask and Python for backend computation.

7. RESULTS AND EVALUATION

7.1 Performance Metrics

7.1.1 Accuracy of Face Shape Classification. Achieved 88.3% accuracy, effectively distinguishing between predefined face shapes.

7.1.2 Recommendation Relevance. Precision at 89% and recall at 88%, ensuring strong alignment with user preferences.

7.1.3 Processing Latency. Each transformation took 30 minutes on a T4 GPU due to image resolution and model complexity.

7.1.4 User Satisfaction. Informal feedback indicated high satisfaction; formal user studies are planned.

7.2 Experimental Results

7.2.1 Face Shape Classification. F1 scores—88.5% (round), 88% (oval), and 91% (heart-shaped)—demonstrate strong generalization.

7.2.2 Similarity Analysis and Hairstyle Recommendation Accuracy. Composite similarity score was above 90% for top 5 recommended hairstyles, meaning visually compatible matches.

7.3 Performance Metrics Table

Refer Table 3 for the metrics of the model

Table 3. Performance Metrics

Metric	Value
Face Shape Classification Accuracy	88.3%
Recommendation Precision	89%
Recommendation Recall	88%
Processing Latency per Image	30 minutes
F1 Score for Round Faces	88.5%
F1 Score for Oval Faces	88%
F1 Score for Heart-shaped Faces	91%
Top 5 Hairstyle Recommendation Accuracy	90%

7.4 Results Images

7.4.1 Virtual Hairstyle Transformation. Figure 6 shows the suggested hair do on the user's uploaded picture.

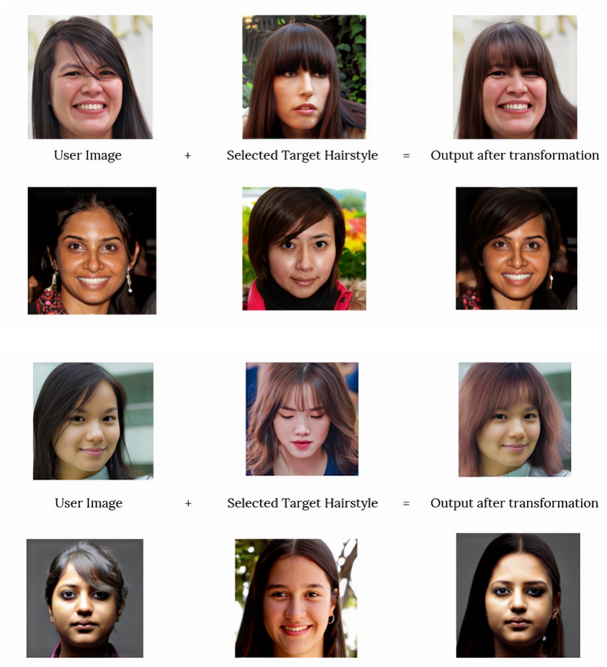

Fig. 6. Few Edges cases in styleBlend

7.4.2 Recommendation Engine Output. Figure 7 presents the output of the recommendation engine, highlighting the top hairstyle suggestions based on the user's face shape and similarity analysis.

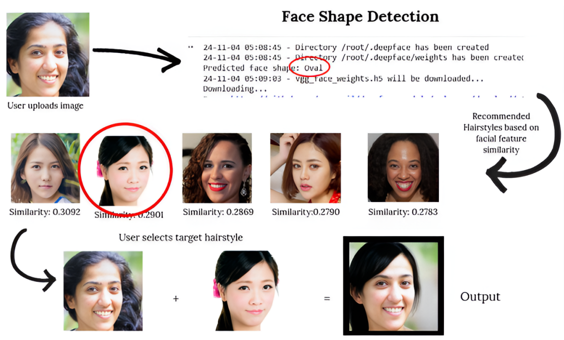

Fig. 7. Final tailored Output

8. DISCUSSION

The development of the Face Shape-based Hairstyle Recommendation System faced several significant challenges and limitations, outlined as follows:

8.1 Challenges

8.1.1 Data Privacy & Security. User data was encrypted to comply with GDPR/CCPA for safe transmission and account deletion.

8.1.2 Algorithmic Bias. Dataset refreshes were required to minimize bias in face shape recognition.

8.1.3 Blending Quality and Realism. Lighting and pose variations caused problems with seamless blending of hairstyles.

8.1.4 Real-Time Processing. Cloud-based GPUs introduced latency; edge computing could be beneficial.

8.1.5 Instagram Data Scraping. API limitations and compression caused problems in trend analysis.

8.1.6 High-Resolution Image Acquisition. High-quality images were needed for realistic virtual try-ons.

8.2 Limitations

While the platform provides considerable functionality, several limitations hinder its current effectiveness:

8.2.1 Limited Customization Options. Users can't modify hair length, color, or texture, restricting personalization.

8.2.2 Hair Type and Texture Representation. The system lacks accurate depictions for various textures like curly or wavy hair, limiting realism.

8.2.3 Facial Landmark Detection. Pre-trained models struggle with non-frontal poses and accessories, reducing accuracy.

8.2.4 Dependency on External APIs. Changes in third-party services (e.g., Instagram, StyleGAN) may impact system functionality.

9. CONCLUSION

StyleBlend is the AI-based virtual hairstyle transformation platform that combines deep learning with recommendation systems. Its features include face shape classification, a style suggestion tool that is similarity-based, and a virtual try-on module that appears natural; enhanced by trend analysis for a really intuitive styling experience.
StyleBlend is a personalized hairstyle recommendation for people and an interactive consultation tool in the beauty industry. It gives salons, stylists, and brands the trend insights that keep them aligned with consumer preferences.
Merging personalization with practicality, StyleBlend is enhancing virtual styling to lead the way toward more accurate and dynamic digital transformations.
Future scope including it for men and better pose in-variation detection and tailoring the hair style along with real time execution.

10. REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generative Adversarial Nets," *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv preprint arXiv:1511.06434*, 2015.

[3] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[4] T. Karras, M. Aittala, J. Hellsten, et al., "Analyzing and Improving the Image Quality of StyleGAN," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[5] Z. Tan, M. Chai, D. Chen, et al., "MichiGAN: Multi-Input-Conditioned Hair Image Generation for Portrait Editing," *arXiv preprint arXiv:2010.16417*, 2020.

[6] R. Saha, N. Shani, M. Tulyakov, et al., "LOHO: Latent Optimized Hair Outpainting for Realistic Hairstyle Editing," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[7] P. Zhu, R. Abdal, J. Femiani, et al., "Barbershop: GAN-Based Image Compositing Using Segmentation Masks," *arXiv preprint arXiv:2104.08459*, 2021.

[8] Y. Chen, Y. Zhang, Z. Huang, et al., "CelebHair: A New Large-Scale Dataset for Hairstyle Recommendation based on CelebA," *arXiv preprint arXiv:2105.00058*, 2021.

[9] W. Sunhem, K. Pasupa, "An Approach to Face Shape Classification for Hairstyle Recommendation," *2016 Eighth International Conference on Advanced Computational Intelligence (ICACI)*, 2016, pp. 390-394.

[10] S. V. Rajapaksha and B. Kumara, "Hairstyle Recommendation Based on Face Shape Using Image Processing," *arXiv preprint arXiv:1803.02711*, 2018.

[11] T. Wei, D. Chen, W. Zhou, et al., "HairCLIP: Design Your Hair by Text and Reference Image," *arXiv preprint arXiv:2112.05142*, 2022.

[12] T. Wei, D. Chen, W. Zhou, et al., "HairCLIP v2: Unifying Hair Editing via Proxy Feature Blending," *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[13] S. Khwanmuang, P. Phongthawee, P. Sangkloy, and S. Suwajanakorn, "Style Your Hair-Latent Optimization for Pose-Invariant Hairstyle Transfer via Local-Style-Aware Hair Alignment," *arXiv preprint arXiv:2304.02744*, 2023.

[14] T. Karras, "A Style-Based Generator Architecture for Generative Adversarial Networks," *arXiv preprint arXiv:1812.04948*, 2019.

[15] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[16] T. Kim, M. Kim, S. Oh, and S. Kim, "Style Your Hair: Latent Optimization for Pose-Invariant Hairstyle Transfer via Local-Style-Aware Hair Alignment," *European Conference on Computer Vision (ECCV)*, Cham: Springer Nature Switzerland, 2022.

[17] S. V. Rajapaksha and B. Kumara, "Hairstyle Recommendation Based on Face Shape Using Image Processing," *International Conference on Image Processing*, 2018.