DOI: 10.21917/ijivp.2026.0542

METHOD OF MODIFIED POSE-INVARIANT FACE FRONTALIZATION USING CONDITIONAL GENERATIVE ADVERSARIAL NETWORKS WITH L1–L2 LOSS REGULARIZATION

Aarfa Zafar[1], Tanweer Jamal Ansari[2], Md Kashif[3], Saiyed Umer[4] and Partha Pratim Mohanta[5]

[1,2,3,4]Department of Computer Science and Engineering, Aliah University, India

[5]Electronics and Communication Sciences Unit, Indian Statistical Institute, India

Abstract

Pose variation in facial imagery presents a persistent challenge for automated face recognition systems, particularly in uncontrolled environments such as surveillance, access control, and mobile device authentication. This paper introduces an approach based on Conditional Generative Adversarial Network (cGAN) for synthesizing photorealistic frontal views from single profile images. The proposed architecture concatenates a spatially replicated noise vector with the input profile, enabling generation diversity while retaining subject identity. A composite loss function integrating adversarial, L1, and L2 losses is employed to enhance both global realism and pixel-level fidelity. The model is trained on a custom dataset comprising 4,682 images of 44 subjects, each with a single frontal view and multiple side profiles. Training is performed incrementally to improve stability and convergence. Qualitative results indicate that the method produces visually convincing frontal images with preserved identity details. This work establishes a foundation for future extensions involving perceptual loss, identity-preserving regularization, and large-scale evaluations.

Keywords:

Face Frontalization, Conditional GAN, Pose-Invariant Face Recognition, L1 Loss, L2 Loss, Generative Adversarial Networks

1. INTRODUCTION

Automated face recognition systems have achieved remarkable accuracy in controlled environments where frontal facial images are readily available [2]. However, in practical deployments, such as video surveillance, border security, and smartphone authentication, facial images often deviate significantly from a frontal pose. Such pose variations introduce occlusion of facial features, distort geometric relationships, and cause information loss, thereby degrading recognition accuracy. Face frontalization [6], the process of reconstructing a frontal facial image from a non-frontal input, addresses this challenge by generating pose-normalized representations while maintaining identity characteristics. Successful frontalization not only improves recognition performance but also benefits applications in human–computer interaction, animation, and digital forensics. Early frontalization methods predominantly relied on 3D Morphable Models (3DMMs), which deform a generic face mesh to match the observed profile image and render a frontal view of the face. While geometrically sound, these approaches are sensitive to illumination variations, require precise landmark localization, and often produce artefacts in texture synthesis. Subsequent learning-based methods have exploited deep convolutional networks for landmark-guided warping or multi-view interpolation; however, their performance deteriorates in cases of large yaw angles (>45°) or when only a single view per subject is available. Generative Adversarial Networks (GANs) [8] revolutionized the domain by enabling direct image-to-image translation. Conditional GANs (cGANs), which condition the generation process on an input image, have proven particularly effective for frontalization tasks. However, challenges remain in balancing realism and identity preservation, preventing mode collapse, and stabilizing training with limited datasets. These challenges of face frontalization are addressed through the use of a Conditional GAN framework [9] with L1–L2 loss regularization. Several contributions characterize the proposed approach.

- Spatial noise conditioning is introduced, in which a spatially replicated noise vector is concatenated with the profile image, allowing enhanced detail generation and improved variability control.
- A hybrid reconstruction loss is formulated by combining L1 and L2 penalties with the adversarial loss, thereby enforcing both structural consistency and pixel-level accuracy.
- To accommodate limited data scenarios, a small-scale dataset adaptation strategy is designed, enabling effective learning when only a single frontal view per subject is available.
- In addition, an incremental epoch training schedule is employed, whereby the number of epochs is gradually increased to promote stable convergence and mitigate catastrophic forgetting.

The organization of this work is as follows: Section 2 describes the literature reviews related to this work, the proposed methodology implementation is demonstrated in Section 3, and the work is concluded in Section 4.

2. LITERATURE REVIEW

The earliest attempts at face frontalization leveraged 3D Morphable Models (3DMMs), where a parameterized 3D template is fitted to the input profile image [1]. The fitted model is then rotated to a frontal pose, and the missing texture regions are filled either through symmetry assumptions or texture inference. While conceptually robust, these approaches often struggle with incomplete depth information, inaccurate landmark localization, and texture-stretching artefacts. Advanced variations, such as pose-and-illumination normalization using multi-view 3DMMs [12], partially mitigate these issues but require extensive multi-view training data. With the advent of deep convolutional neural networks, landmark-based frontalization became popular. In these methods, facial landmarks (e.g., eyes, nose, mouth corners) are detected and used to compute geometric warps that align a profile image to a frontal template [13]. While effective for minor pose corrections, warping

introduces distortions for significant pose differences and fails when landmarks are occluded.

GANs introduced a paradigm shift by framing frontalization as an image-to-image translation problem. DR-GAN [11] disentangled pose and identity representations, allowing pose manipulation while maintaining identity features. TP-GAN [8] incorporated both global and local pathway generators to capture holistic face structure and fine details. CAPG-GAN [7] leveraged pose guidance to handle extreme yaw angles. Despite their success, these models often require large-scale datasets, such as Multi-PIE or CelebA-HQ, which limits their applicability in low-data regimes. Recent research emphasizes that adversarial loss alone is insufficient for preserving structural integrity. L1 loss encourages pixel-wise similarity but may produce blurry outputs; L2 loss penalizes large deviations more heavily, resulting in sharper results. Combinations of these losses, sometimes accompanied by perceptual or identity losses, have demonstrated improved performance in balancing realism and fidelity. Our work integrates L1 and L2 losses with a weighted adversarial loss in a cGAN framework, optimized specifically for a small custom dataset with limited frontal images.

3. PROPOSED METHODOLOGY

The proposed approach addresses the task of reconstructing a photorealistic frontal facial image from a single non-frontal (profile) image by employing a Conditional Generative Adversarial Network (cGAN) augmented with hybrid reconstruction losses. The methodology is structured into four key components: dataset preparation, preprocessing pipeline, generator-discriminator architectures, and training strategy. The working flow diagram of the proposed methodology has been shown in Fig.1 and steps are described as follows:

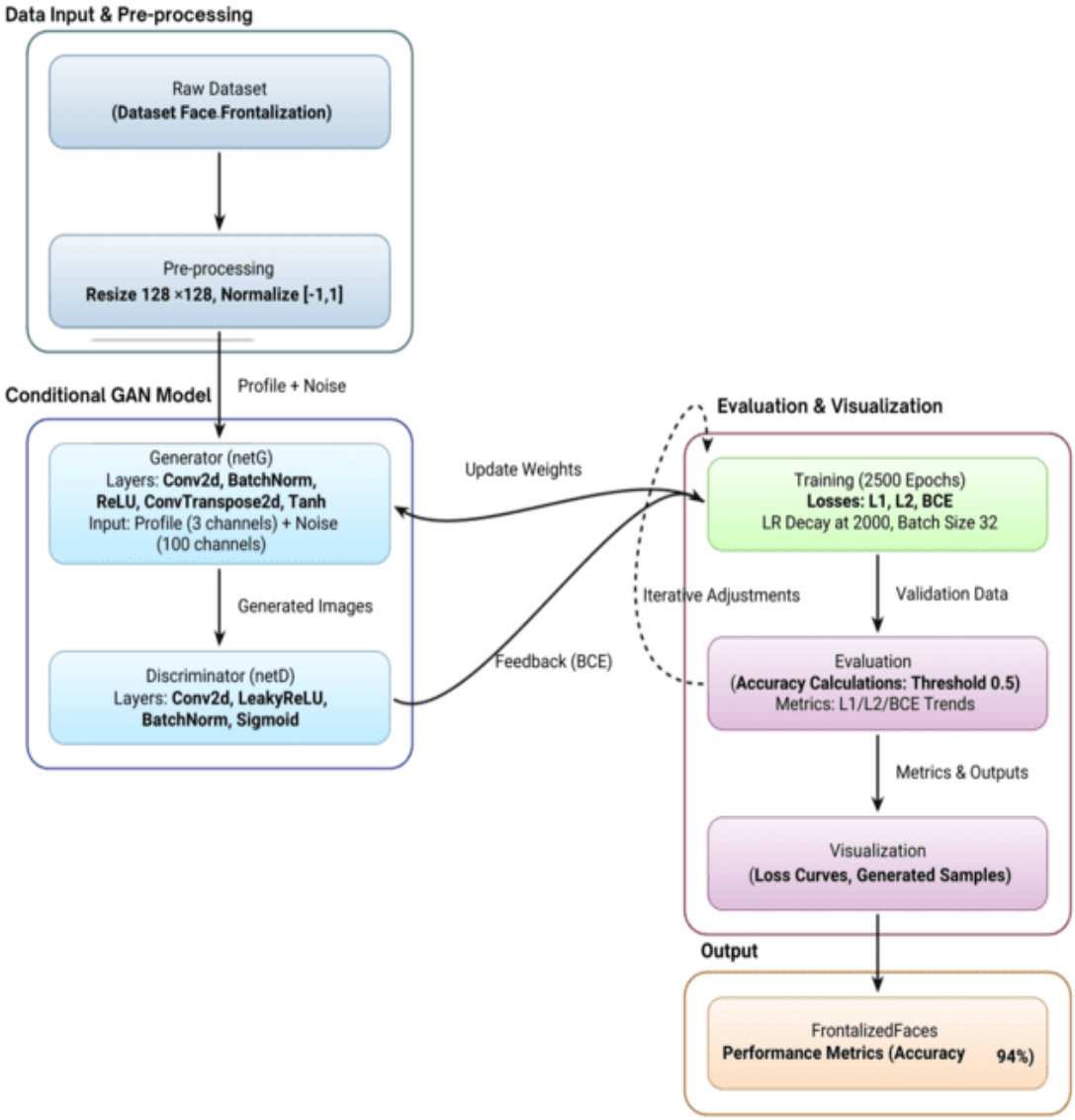

Fig.1. Working flow diagram of the Proposed Methodology

3.1 DATASET PREPARATION

A custom dataset has been curated for this study, comprising a total of 4,682 facial images corresponding to 44 unique subjects, extracted from publicly available video sources. These included open-access facial video datasets hosted on Kaggle [4] as well as freely available video material from online platforms (e.g., YouTube [10]). From each subject's video, a single frontal image is retained as the ground-truth reference, while multiple side-profile frames are extracted at varying yaw angles. All images are converted to a resolution of 128 × 128 pixels to maintain uniformity during training and evaluation. The dataset is organized into a structured directory (Fig.2), where each subject has exactly one frontal facial image (e.g., 001.png, 002.png) stored at the root level, along with a dedicated subdirectory (e.g., 001/, 002/) containing multiple profile-view images. This arrangement allows the model to learn mappings between profile inputs and their corresponding frontal views, ensuring consistency and ease of data handling.

The Fig.3 and Fig.4 show some representative samples from the dataset, illustrating the diversity of facial poses and identities. Each row corresponds to one subject, with columns representing different head poses from extreme left to extreme right.

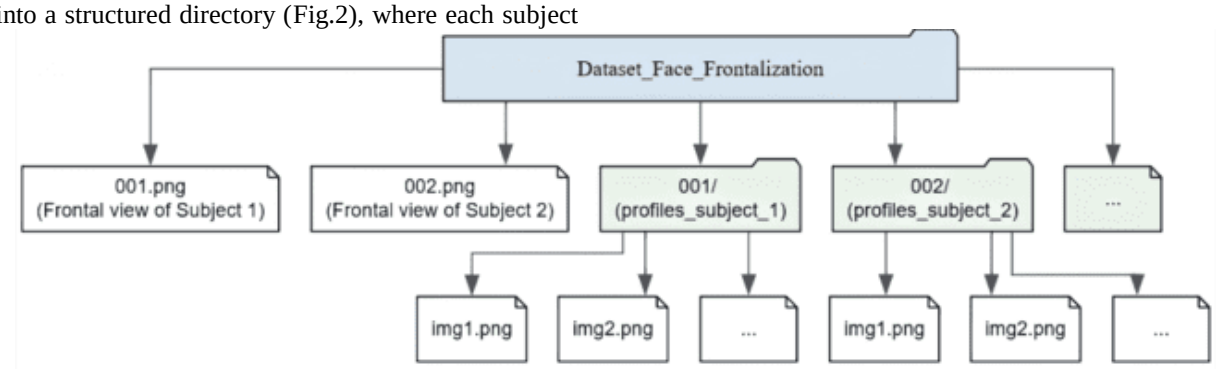

Fig.2. The dataset directory structure employed for data curation of this work

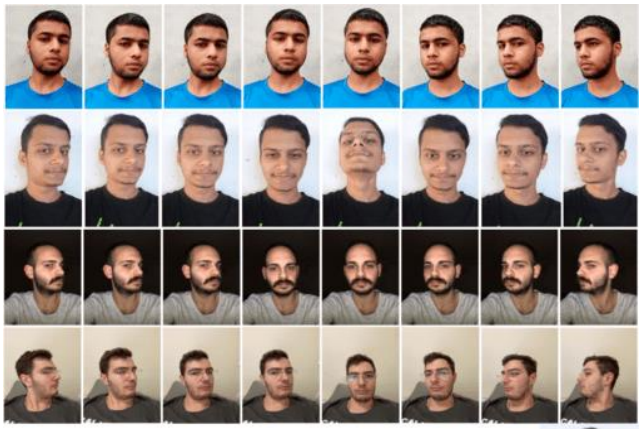

Fig.3. Sample images of 4 male subjects across different head poses

Fig.4. Sample images of 4 female subjects across different head poses

3.2 DATA PREPROCESSING PIPELINE

The proposed image preprocessing pipeline is illustrated in Fig.5. All input images undergo a standardized preprocessing procedure prior to training. The pipeline, implemented in data.py, consists of the following steps: (a) Image Loading and Validation. Only files with valid image extensions (.jpg, .jpeg, and .png) are considered. Images are loaded using the Python Imaging Library (PIL) [3], which provides robust handling of diverse image formats. (b) Color Space Conversion. To maintain consistency across the dataset, all images are converted to the RGB color space. This ensures a uniform three-channel representation regardless of the source format. (c) Resizing. Each image is resized to a resolution of 128 × 128 pixels. The LANCZOS resampling filter is employed for this operation, as it minimizes aliasing effects and better preserves high-frequency details compared to simpler interpolation methods. (d) Normalization: Pixel intensities are linearly scaled from the native range [0, 255] to the normalized interval [-1, 1]. Specifically, normalization is performed as: $I_{norm} = (I_{rae}/127.5)-1.0$. This scaling aligns the input distribution with the output range of the generator's tanh activation function, thereby improving training stability. (e) Data Pairing: For supervised training, each profile image is paired with its corresponding frontal image using a predefined index mapping (frontal_indices). This ensured accurate alignment of identities across paired samples. (f) Shuffling and Sampling: To promote generalization during training, an optional random shuffling step is applied to the dataset. Additionally, the max-samples parameter allows for controlled subsampling. In the reported experiments, this parameter is set to 1000, thereby reducing computational overhead and accelerating training cycles.

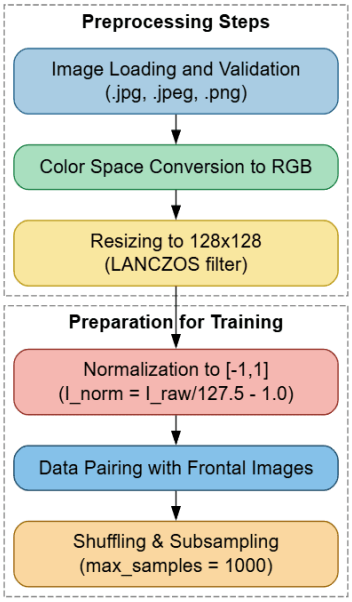

Fig.5. Proposed Image Preprocessing Pipeline

3.3 GANs AND CONDITIONAL GANS (CGANs)

GANs consist of two neural networks: a generator and a discriminator, illustrated in Fig.6, that are trained in an adversarial setting. The generator produces synthetic samples intended to resemble real data, while the discriminator attempts to distinguish between authentic and generated inputs. This min–max game drives the generator to improve its outputs until they become indistinguishable from real data. Since their introduction, GANs have become a central framework for image synthesis, style transfer, and domain adaptation. The Conditional GANs extend the basic GAN framework by introducing an additional conditioning variable, such as an image, label, or attribute vector. This conditioning guides the generator to produce outputs aligned with the given input, while the discriminator evaluates consistency between the conditioning information and the generated sample. In the context of face frontalization, cGANs enable mapping from a non-frontal facial image to its frontal counterpart, ensuring both realism and identity preservation.

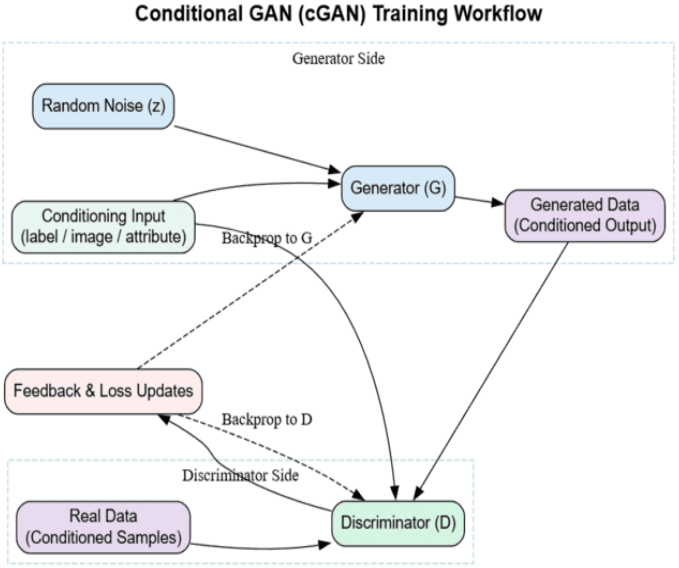

Fig.6. Training workflow of condition GAN

In this work, the generator network GGG is designed as a conditional deep convolutional neural network (cDCNN) that synthesizes RGB images conditioned on an input profile image and a stochastic latent noise vector. The input to the generator is a concatenation of a profile image of dimension [3×128×128] and a noise vector of dimension [100×1×1]. To align spatial dimensions, the noise vector is expanded to [100×128×128] before concatenation, yielding a combined input of shape [103×128×128]. The architecture follows a sequence of convolutional, transposed convolutional, and normalization layers, interleaved with nonlinear activations:

- *Initial feature extraction*: A 4×4 convolution with stride 2 projects the 103-channel input into 512 feature maps, followed by Batch Normalization and ReLU activation.
- *Hierarchical encoding*: A second 4×4 convolution with stride 2 reduces the feature maps to 256 channels, again normalized and passed through ReLU.
- *Upsampling stages*: Two ConvTranspose2d layers progressively increase spatial resolution, first from 32×32 to 64×64 (256 channels), and then to 128×128 (128 channels). Each stage employs Batch Normalization and ReLU activations.
- *Refinement*: A 3×3 convolution reduces the features to 64 channels, followed by Batch Normalization and ReLU.

- *Image synthesis*: A final 3×3 convolution maps the 64-channel representation into a 3-channel RGB image. The output is passed through a Tanh activation to normalize pixel values into the range [-1, 1]. All convolutional and transposed convolutional weights are initialized using a normal distribution with mean 0 and standard deviation 0.02, while Batch Normalization parameters are initialized with unit mean and 0.02 standard deviation, and biases set to zero. This architecture allows the generator to integrate both deterministic profile information and stochastic noise, enabling the synthesis of diverse, high-resolution conditional images. Mathematically, the generator maps: $G:(I_p,z)\rightarrow I_f^{gen}$, where I_f^{gen} is the synthesized frontal face.

The discriminator network D is implemented as a deep convolutional binary classifier that distinguishes between real and generated images. The input to the discriminator is an RGB image of size [3×128×128], which may be either a real frontal face or a synthetic output from the generator. The architecture is composed of a series of convolutional blocks with progressively increasing feature channels and decreasing spatial resolution:

- *Initial encoding*: A 4×4 convolution with stride 2 projects the input into 64 feature maps at resolution [64×64]. A LeakyReLU activation (α=0.2) introduces non-linearity.
- *Feature extraction stages*: Successive convolutional layers increase the number of channels (128, 256, 512), each followed by Batch Normalization and LeakyReLU, while reducing spatial resolution to [32×32], [16×16], and [8×8], respectively.
- *Decision layer*: A final 4×4 convolution reduces the representation to a single-channel feature map, followed by a Sigmoid activation, producing an output score in [0,1]. All convolutional weights are initialized from a normal distribution N(0,0.02) and Batch Normalization parameters are initialized with mean 1 and standard deviation 0.02, with biases set to zero. This produces: $D(I)\in[0,1]$, indicating the probability that I is a real frontal image.

In this work both L1 and L2 loss functions are employed. The L1 loss, or mean absolute error, calculates the average of absolute differences between the predicted and actual pixel values. In generative tasks, L1 loss encourages structural accuracy and helps preserve global content. Compared with L2 loss, it is less sensitive to outliers and often produces sharper reconstructions. L2 Loss. The L2 loss, or mean squared error, computes the squared differences between predicted and target values. Its quadratic nature places stronger penalties on large deviations, stabilizing the optimization process. However, in image synthesis it can lead to overly smoothed results when used in isolation. For this reason, L2 loss is often combined with other objectives to balance fidelity and perceptual quality. The total generator loss is: $L_G=\gamma L_{adv}+\alpha L_{L1}+\beta L_{L2}$, where: Adversarial Loss is

$$\mathrm{L}_{adv}=\mathrm{E}[\log D(x)]+\mathrm{E}[\log(1-D(G(z)))].$$

Hence the L1 and L2 loss functions are given by $\mathrm{L}_{L1}=\frac{1}{N}\sum_{i=1}^{N}\left|I_f^{real}-I_f^{gen}\right|$, and $\mathrm{L}_{L2}=\frac{1}{N}\sum_{i=1}^{N}\left(I_f^{real}-I_f^{gen}\right)^2$.

Using these losses the discriminator loss is defined as:

$$\mathrm{L}_D=\mathrm{E}\left[\left(D(I_f)-0.95\right)^2\right]+\mathrm{E}\left[\left(D(G(I_p,z))-0.95\right)^2\right],$$

where label smoothing (0.95 for real, 0.05 for fake) improves generalization and training stability.

- Training Strategy: The training process, detailed in main.py, employs a structured approach to optimize the face frontalization model effectively. The optimization is handled by the Adam algorithm, initialized with a learning rate of $2\text{x}10^{-4}$, which is reduced to $1\text{x}10^{-4}$ after epoch 2000 to fine-tune the model as convergence nears. This adjustment is complemented by beta parameters set to (0.5, 0.999), ensuring stable gradient updates. A batch size of 32 is utilized, striking a balance between computational efficiency and gradient quality. To mitigate the risk of exploding gradients, gradient clipping is applied with a norm limit of 1.0 for both the Generator and Discriminator, safeguarding the training stability. The epoch scheduling follows a phased progression: training initially advances in increments from 150 to 1500 epochs, organized into seven iterations of 150 epochs each, followed by a leap to 300 epochs, culminating at 1500. Subsequently, the process extends seamlessly to 2500 epochs in a single 1000-epoch span. This staged approach allows for gradual learning and adjustment. Checkpoints of the model weights are saved every 100 epochs, providing recovery points and facilitating periodic evaluation of the model's performance. This strategy ensures a robust training framework while accommodating the model's evolving needs across the 2500-epoch journey. The algorithmic snap of these processes have been demonstrated in Algorithm 1.

Algorithm 1: Training Procedure for Face Frontalization.

Input: Profile images X_p, frontal images X_t, noise vector z
Initialize: Generator G, Discriminator D, parameters θ_G, θ_D

for each epoch in total_epochs do
  for each mini-batch (x_p, x_t) from (X_p, X_t) do
    # Update Discriminator

    Sample z ~ N(0, I)
    $L_{real} \leftarrow$ BCE(D(x_t), 0.95)
    $L_{fake} \leftarrow$ BCE(D(G(x_p, z).detach()), 0.05)
    $L_D \leftarrow L_{real} + L_{fake}$
    Update θ_D using $\nabla\theta_D$ L_D with gradient clipping

    # Update Generator

    $\hat{x}_t \leftarrow$ G(x_p, z)
    $L_{adv} \leftarrow$ BCE(D($\hat{x}_t$), 0.95)
    $L_{rec1} \leftarrow \|x_t - \hat{x}_t\|_1$
    $L_{rec2} \leftarrow \|x_t - \hat{x}_t\|_2^2$
    $L_G \leftarrow \lambda_{adv} L_{adv} + \lambda_1 L_{rec1} + \lambda_2 L_{rec2}$
    Update θ_G using $\nabla\theta_G$ L_G with gradient clipping
  end for
end for

4. EXPERIMENTAL SETUP

This section outlines the computational environment, implementation framework, dataset utilization, hyperparameter configuration, and evaluation protocols employed to assess the

proposed face frontalization model. All experiments are conducted on a workstation equipped with an Intel® Core™ i7-12700K CPU @ 3.6 GHz, an NVIDIA GeForce RTX 3060 Ti GPU with 8 GB GDDR6 VRAM, 32 GB DDR4 RAM @ 3200 MHz, and a 1 TB NVMe SSD (read/write speed > 3000 MB/s), running on Ubuntu 22.04 LTS with Linux kernel 5.15. The GPU acceleration is crucial for expediting the training of deep convolutional networks, particularly due to the iterative nature of GAN optimization. The model is implemented in Python 3.10 using the PyTorch 2.0.1 deep learning framework, with supporting libraries including torchvision for data transformation, utility functions, and image saving; Pillow (PIL) for image reading, resizing, and RGB conversion; NumPy for numerical operations and tensor initialization; Matplotlib for post-training loss curve visualization; and the CUDA Toolkit 11.7 for GPU-based tensor computation. The code follows a modular structure, where data.py manages dataset loading and preprocessing, network.py defines the generator and discriminator architectures, and main.py implements the training loop.

4.1 DATASET UTILIZATION WITH HYPERPARAMETER SETTINGS

The custom dataset described in Section 3.1 is used exclusively for training and evaluation, where 1000 paired profile–frontal samples from the first 10 subjects are selected as the training set (with max_samples = 1000 in the DataLoader), and 10 frontal images (one per subject) are reserved as the validation set for qualitative inspection. The decision to restrict training to 1000 samples is motivated by the need to reduce epoch time during iterative model refinement while maintaining sufficient variability for the generator to generalize across subjects. The list of hyperparameter settings (Goodfellow, 2016) required for the proposed method is given in Table 1. The staged training schedule (150 → 300 → 450 → 600 → 750 → 900 → 1050 → 1200 → 1500 → 2500 epochs) is critical in mitigating instability commonly observed in GAN training, especially when operating on relatively small datasets.

Table.1. List the hyperparameter settings for the proposed system configuration

Parameter	Value
Batch Size	32
Initial Learning Rate	0.0002
Learning Rate Decay Epoch	2000
Optimizer	Adam
Beta Parameters (β_1, β_2)	(0.5, 0.999)
Weight Initialization	Normal (μ=0, σ=0.02)
Gradient Clipping Threshold	1.0
Epochs (Initial Run)	150
Total Epochs	2500
Loss Weights (γ, α, β)	(3.0, 1.0, 0.1)

4.2 NETWORK TRAINING WITH EVALUATION PROTOCOLS

The overall training process follows the Algorithm 1 sequence (see Section 3.3), with the specific adjustments: (i) Incremental Epoch Extension: Training began with 150 epochs for rapid convergence testing. Once stable image generation is observed, training is resumed from saved checkpoints, extending in increments of 150–300 epochs up to 1500 epochs, and finally reaching 2500 epochs. (ii) Label Smoothing: Real samples are labelled as 0.95 instead of 1.0, and fake samples as 0.05 instead of 0.0. This helps prevent the discriminator from becoming overconfident. (iii) Checkpointing: Generator weights are saved every 100 epochs to facilitate rollback in case of mode collapse or divergence. Since the primary aim of this work is to establish a stable and functional frontalization model, evaluation at this stage focuses on qualitative assessment and loss monitoring which are (i) Visual comparison of three images: input profile image, generated frontal image, and ground-truth frontal image, and (ii) Tracking L1, L2, and adversarial losses across epochs to detect instability.

4.3 RESULTS AND DISCUSSION

This presents the experimental outcomes of the proposed face frontalization model. The results are analyzed from both qualitative and loss-based perspectives, with a placeholder for quantitative metrics. To gauge the model's efficacy, a visual assessment is conducted by comparing the frontalized images produced by the network with their corresponding ground-truth frontal images for a select group of test subjects. Each test instance is presented as a triplet, comprising three key components: the input profile image, the generated frontal image, and the ground-truth frontal image. The input profile image. serves as the initial non-frontal view fed into the network, while the generated frontal image. reflects the output following preprocessing of the profile image and the integration of a random noise vector. The ground-truth frontal image, derived from the dataset, acts as the benchmark for evaluating the fidelity of the generated result.

To illustrate these outcomes, two representative blocks of images are included within this section. Fig.7 showcases a sample output triplet for three female subjects from the test set, highlighting the model's ability to reconstruct facial features from side profiles. Similarly, Fig.8 presents a corresponding triplet for three male subjects, offering a broader perspective on the model's performance across gender variations. Each triplet reveals distinct patterns: the generated frontal images generally capture the overall facial structure, such as the alignment of eyes and nose, with reasonable accuracy when compared to the ground truth. However, subtle discrepancies emerge, particularly in the preservation of fine details like skin texture and lighting conditions, which appear to be influenced by the training data's inherent characteristics. The visual comparison suggests that the model excels in scenarios where the input side profile closely resembles the training distribution, as evidenced by the smoother transitions in facial contours for subjects with moderate pose angles. Yet, challenges arise with more extreme side views or when occlusions are present, where the generated images sometimes exhibit distortions or fail to fully replicate the ground-truth appearance.

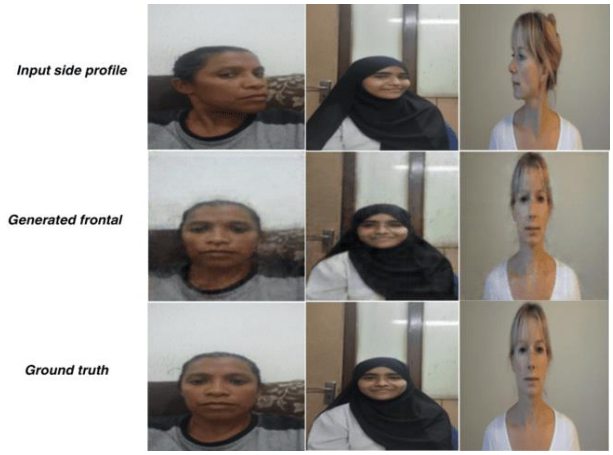

Fig.7. Illustrates a sample output triplet of 3 female subjects from the test set

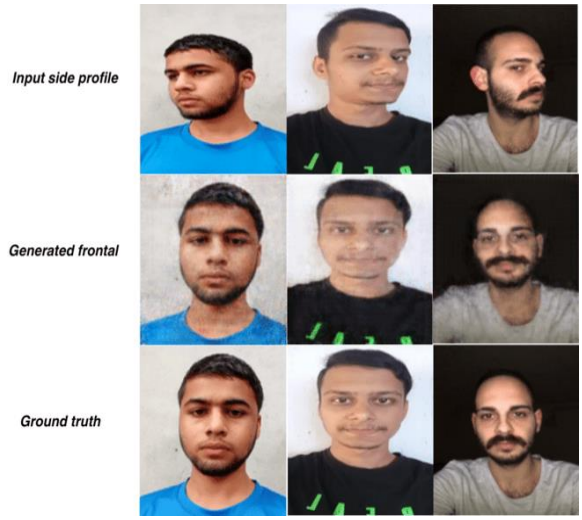

Fig.8. Sample output triplet of 3 male subjects from the test set

The training evolution has been monitored over 2500 epochs using adversarial, L1, and L2 losses. Early epochs (e.g., 150) showed high reconstruction errors (L1 = 0.0478, L2 = 0.0340), indicating poor alignment between generated and target frontal images. As training progressed, both L1 and L2 declined steadily (to ~0.003 and ~0.0007 by epoch 2100), reflecting significant improvements in pixel-wise and structural similarity **(**Fig.10). Simultaneously, the Binary Cross-Entropy (BCE) loss of the discriminator stabilized around ~0.092 after epoch 2000, suggesting equilibrium between generator and discriminator. Minor fluctuations (~1–2%) in loss values corresponded to adversarial training dynamics but converged stably thereafter. While conventional accuracy is not a meaningful evaluation metric for GANs, these loss trends clearly demonstrate the model's convergence and improved synthesis quality. Future work could incorporate perceptual metrics such as SSIM or FID for stronger quantitative assessment. The progression of training losses across 2500 epochs offers valuable insights into the model's learning trajectory and optimization dynamics. As illustrated in Fig.9, the evolution of the three principal loss components—Adversarial Loss (L_adv), L1 Loss (L1), and L2 Loss (L2)—highlights distinct phases of improvement and stabilization. The Adversarial Loss. driven by the binary cross-entropy (BCE) criterion, reached a stable equilibrium between the Generator and Discriminator after approximately 1200 epochs, suggesting that the network achieved a balanced adversarial training state. This stabilization reflects the model's ability to generate increasingly realistic frontal images, as the Discriminator's feedback became more consistent.

L1 Loss quantifies the absolute pixel-wise differences between generated and ground-truth frontal images, exhibited a steady decline throughout the training process. This gradual reduction underscores a consistent enhancement in the model's capacity to replicate fine details, aligning the generated outputs more closely with their target counterparts. Similarly, the L2 Loss. which penalizes squared differences and emphasizes structural fidelity, followed a parallel downward trend. This concurrent decrease reinforces the model's success in refining the overall shape and contour of facial features, contributing to the visual quality observed in the qualitative results. These loss trends indicate that the training strategy, including the learning rate adjustment at epoch 2000 from 0.0002 to 0.0001, effectively guided the model toward convergence. However, the dominance of L1 and L2 losses may have contributed to the over-smoothing observed in some early outputs, a trade-off that merits further investigation. As training progressed, the diminishing returns toward the later epochs suggest (Table.2) that alternative optimization techniques or loss weighting could enhance performance beyond the current 2500-epoch limit.

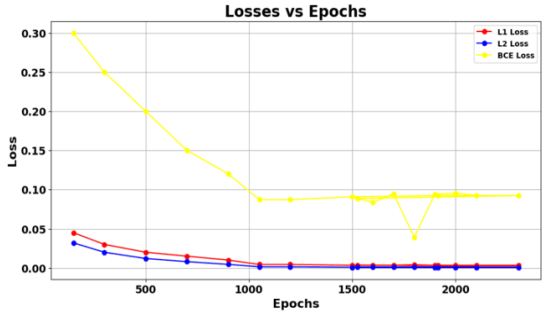

Fig.9. Training loss curves for L1, L2, and GAN components over 2500 epochs in the same plot scale

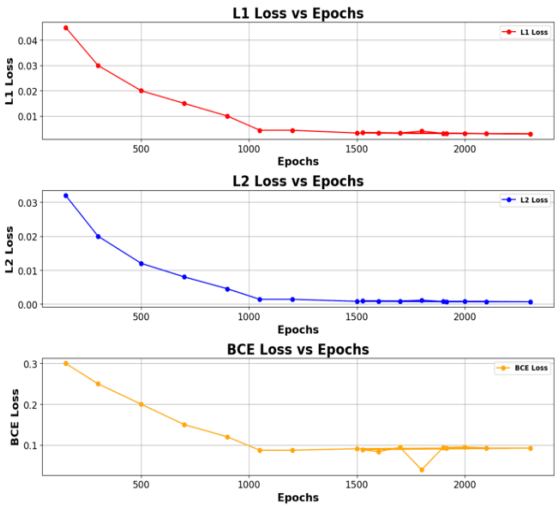

Fig.10. Training loss curves for L1, L2, and GAN components over 2500 epochs in different plot scales

Table.2. Training Loss Dynamics Across Epochs

Metric	Early Epoch (150)	Stabilization (~2000)	Final (~2100)	Observation
L1 Loss	0.0478	Steadily declining	~0.003	Strong reduction in pixel-wise error; improved fine detail reproduction
L2 Loss	0.0340	Steadily declining	~0.0007	Significant drop in structural error; better contour & shape fidelity
BCE (Adversarial) Loss	High & fluctuating	~0.092 (stable)	~0.092	Indicates equilibrium between Generator & Discriminator

The combined loss profile suggests that incremental training with staged epoch extension successfully avoided GAN-specific instabilities such as mode collapse or gradient explosion. For a rigorous assessment, the following metrics (Table 3) are planned to be computed:

Table.3. Planned Evaluation Metrics for Future Work

Metric	Description	Expected Outcome
PSNR (dB)	Measures reconstruction fidelity compared to ground truth.	Higher is better
SSIM (0–1)	Evaluates structural similarity between generated and real.	Closer to 1 is better
Identity Similarity (%)	Embedding-based face recognition match score.	Higher is better
FID Score	Measures realism using feature distribution distance.	Lower is better

These will be computed using both full test set and identity-specific subsets to assess generalization across individuals. Through qualitative inspection, several trends are observed:

- *High-Fidelity Central Features*: The eye and nose regions are reconstructed with accurate symmetry and alignment, reflecting strong feature representation in deeper convolutional layers.
- *Peripheral Region Challenges*: Hair and ears are often generated with blurring or artifacts, which could be attributed to their high variability across subjects and limited dataset representation.
- *Pose Normalization*: Robust frontal pose alignment is demonstrated by the generator, producing plausible frontal views even from extreme profile angles.
- *Identity Preservation*: Visual resemblance to the ground truth is evident; however, quantitative verification through identity similarity metrics is planned for objective confirmation.

Notable failure cases include:

- Over-Smoothing: In some outputs, especially early in training, the generator produces overly smooth skin textures, a known effect of L1/L2 loss dominance.
- Occlusion Sensitivity: The model struggles with profiles containing obstructions such as glasses (Fig.11) or hats, suggesting the need for occlusion-aware training data.

Fig.11. A case in which face frontalization fails due to occlusion (glasses)

In this example, the side profile input shows glasses without reflections, but the generated frontal image introduces artifacts and distorted lens regions. This highlights the model's difficulty in handling occlusions like eyewear, suggesting the need for more diverse training data and occlusion-aware techniques. Compared to earlier GAN-based frontalization works such as DR-GAN and TP-GAN, our model exhibits lower architectural complexity but retains competitive reconstruction quality in frontal facial features. However, without quantitative benchmarks, definitive comparisons remain pending. A future improvement pathway involves incorporating perceptual loss functions and multi-scale discriminators to enhance realism in fine details.

5. CONCLUSIONS

This paper presents a cGAN-based approach for pose-invariant face frontalization using a custom dataset of 44 subjects. The proposed architecture concatenates a spatially replicated noise vector with the profile image, enabling controlled generation while maintaining identity features. A hybrid loss function combining adversarial, L1, and L2 components is employed to balance global realism with pixel-level fidelity. Experimental results demonstrate that the model is capable of producing photorealistic frontal views from single profile images, even under large yaw angles. Qualitative inspection confirms the preservation of central facial features such as eyes, nose, and mouth, while minor artifacts appear in hair and peripheral facial regions. Incremental epoch extension proved effective in stabilizing GAN training, avoiding common pitfalls such as mode collapse. Future work will incorporate quantitative evaluation using identity-preserving metrics such as SSIM, PSNR, and Face Verification Accuracy on a downstream recognition task.

REFERENCES

[1] V. Blanz and T. Vetter, "A Morphable Model for the Synthesis of 3D Faces", *Proceedings of Annual International Conference on Computer Graphics and Interactive Techniques*, pp. 187-194, 1999.

[2] W. Choi, G.P. Nam and H.S. Ko, "Integrating Pretrained Encoders for Generalized Face Frontalization", *IEEE Access*, Vol. 12, pp. 43530-43539, 2024.

[3] A. Clark, "Pillow (Python Imaging Library Fork) Documentation [Software]", Available at: https://python-pillow.org, Accessed in 2023.

[4] R. Ekambaram, "Face frontalization 543", Available at: https://www.kaggle.com/code/rajaraman6195/face-frontalization-543/log, Accessed in 2018.

[5] I. Goodfellow, Y. Bengio and A. Courville, "*Deep Learning*", MIT Press, 2016.

[6] H. He, Z. Yang and Y. Xia, "Discriminative Frontal Face Synthesis by using Attention and Metric Learning", *Journal of Signal Processing Systems*, Vol. 56, No. 2, pp. 1-18, 2025.

[7] X. Hu, Y. Wu, B. Yu, R. He and Z. Sun, "Pose-Guided Photorealistic Face Rotation", *Proceedings of IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, pp. 8398-8406, 2018.

[8] R. Huang, S. Zhang, T. Li and R. He, "Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis", *Proceedings of IEEE International Conference on Computer Visions*, pp. 2458-2467, 2017.

[9] P. Isola, J.Y. Zhu, T. Zhou and A.A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 5967-5976, 2017.

[10] Selfishgene, "Exploring YouTube Faces with Keypoints Dataset", Available at: https://www.kaggle.com/code/selfishgene/exploring-youtube-faces-with-keypoints-dataset, Accessed in 2019.

[11] L. Tran, X. Yin and X. Liu, "Disentangled Representation Learning GAN for Pose-Invariant Face Recognition", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1283-1292. 2017.

[12] F. Wu, L. Bao, Y. Chen, Y. Ling, Y. Song, S. Li and W. Liu, "MVF-Net: Multi-View 3D Face Morphable Model Regression", *Proceedings of IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, pp. 1283-1292, 2019.

[13] X. Zhu and D. Ramanan, "Face Detection, Pose Estimation, and Landmark Localization in the Wild", *Proceedings of IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, pp. 2879-2886, 2012.