Hybrid GAN-PSO Framework for Image Enhancement

Ayush Mishra , Ayush Suroya , Khushi Verma
Netaji Subhas University of Technology, Delhi
Email: {ayush_mishra, khushi.verma, ayush.suroya}@nsut.ac.in

***Abstract*—This paper presents a hybrid image enhancement framework that integrates a conventional Pix2Pix generative adversarial network (GAN) with Particle Swarm Optimization (PSO) for fine-tuning generator weights. Experiments on the Adobe FiveK dataset demonstrate that the PSO-enhanced GAN achieves faster convergence and superior visual fidelity compared to the baseline. Quantitative metrics (e.g., mean absolute error, structural similarity) and qualitative examples validate the efficacy of the proposed method.**

***Index Terms*—Generative Adversarial Network, Particle Swarm Optimization, Image Enhancement, Adobe FiveK, Pix2Pix**

I. Introduction

A. Introduction to Domain

Image enhancement is crucial for improving visual quality and is widely used in photography, medical imaging, and remote sensing. Traditional methods often struggle to balance detail preservation and noise reduction, motivating the development of deep learning-based approaches. Generative adversarial networks (GANs) have emerged as a powerful tool for image-to-image translation. We present a GAN based on Pix2Pix and a variant with PSO for weight optimization, showing superior convergence and image quality on the Adobe FiveK dataset.

B. Problem Description

Despite advancements in conventional image enhancement techniques, traditional methods often fall short in maintaining a delicate balance between noise reduction and the preservation of fine image details. These techniques, which usually rely on predefined filters or statistical transformations, lack the adaptability to perform consistently across varying lighting conditions, image types, or degradation patterns. As a result, there is a pressing need for more intelligent and flexible solutions.

C. Motivation / Objective

Generative Adversarial Networks (GANs) have emerged as a powerful tool for image enhancement, capable of learning complex mappings between low- and high-quality images. However, their training is unstable and sensitive to hyper-parameters. This work aims to improve GAN training by integrating Particle Swarm Optimization (PSO) to fine-tune generator weights and enhance output quality.

D. Contributions

This paper presents a hybrid image enhancement framework that combines the Pix2Pix GAN architecture with PSO-based optimization. First, a standard Pix2Pix GAN is implemented to handle paired image translation tasks, learning to map low-quality images to their enhanced counterparts. Second, we introduce PSO as an external optimization routine that periodically fine-tunes the generator's weights based on performance on a validation batch, leading to improved convergence and output quality. Finally, we evaluate the model on the Adobe FiveK dataset, a widely used benchmark for image enhancement, and demonstrate that our hybrid approach outperforms the baseline GAN in both quantitative metrics and visual comparisons.

E. Paper Organization

The remainder of this paper is organized as follows. Section II provides a review of related work, highlighting previous applications of GANs and PSO in the context of image enhancement. Section III details the proposed methodology, including both the standard GAN and the PSO-enhanced model. . Section IV presents the results and discusses the performance of the proposed model in comparison to the baseline. Finally, Section V concludes the paper and Section VI outlines directions for future research.

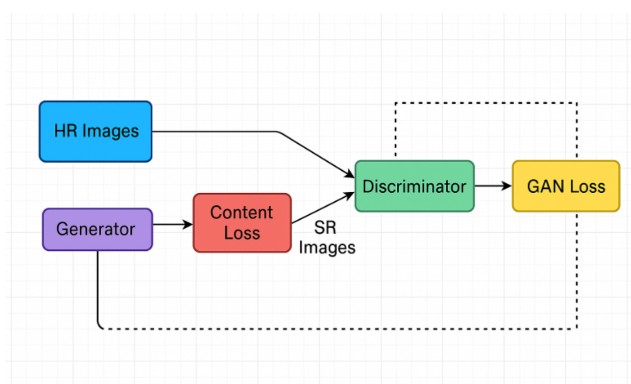

Fig. 1. Simple GAN Model

II. Literature/Related Work

GANs were first introduced by Goodfellow et al. [1] and later adapted for paired image translation in the Pix2Pix framework [2]. Subsequent studies have applied GANs to image enhancement, demonstrating improved noise reduction and detail preservation [3]. Particle Swarm Optimization, a population-based metaheuristic inspired by social 1 behavior in nature, has been widely used for continuous optimization problems [4], with tutorials detailing its efficacy in tuning neural networks [5], [6]. However, few works integrate PSO directly into adversarial training loops for image enhancement.

III. Methodology

A. Approach

Two models are developed for image enhancement. The first is a baseline Pix2Pix-based Generative Adversarial Network (GAN), which learns to translate input images into enhanced versions using paired training data. The second model enhances this framework by integrating Particle Swarm Optimization (PSO) to fine-tune the generator's weights at regular intervals, improving both convergence and image quality.

B. Normal GAN Model

1) Overview: The proposed framework builds upon the Pix2Pix architecture [?], comprising two adversarial components: a generator (G) and a discriminator (D). The generator G receives a raw input image x and learns to produce an enhanced version G(x) that approximates the corresponding ground truth y. Concurrently, the discriminator D evaluates image pairs, distinguishing real pairs (x, y) from fake pairs (x, G(x)).

2) Generator Architecture: The generator adopts an encoder–decoder layout with skip connections, structured as follows:

- **Downsampling:** Four successive 2D convolutional layers (kernel size 4×4, stride 2) halve the spatial resolution at each stage while doubling the number of feature channels. Batch normalization is applied after all but the first convolution, and LeakyReLU activations (negative slope 0.2) introduce nonlinearity.
- **Upsampling:** Four transposed convolutional layers restore spatial dimensions, each featuring batch normalization, ReLU activations, and, in the middle two layers, a dropout rate of 0.5 to mitigate overfitting. The final transposed convolution employs a tanh activation, scaling output values to $[-1, 1]$
- **Skip connections:**Feature maps from each downsampling layer are concatenated with the corresponding upsampling layer input to preserve high-frequency spatial details lost during encoding.

3) Discriminator Architecture: The discriminator follows the PatchGAN paradigm [2], classifying image patches as real or fake:

- **Input:** The discriminator accepts the channel-wise concatenation of the raw input x and either the ground truth y or the generated image $G(x)$.
- **Convolutional layers:** A sequence of five 2D convolutional blocks (kernel size 4×4, stride 2), each followed by batch normalization (except the first) and LeakyReLU activations, progressively reduces spatial dimensions.
- **Output:** A final convolutional layer without activation yields an $M \times N$ feature map, where each element represents the authenticity score of a local image patch.

4) Loss Functions and Training Procedure: The training objective comprises adversarial and reconstruction losses. The generator loss L_G is defined as:

$$L_G = E_{x,y}[-\log D(x, G(x))] + \lambda E_{x,y}[\|y - G(x)\|_1] \quad (1)$$

where $\lambda = 100$ balances the L1 penalty. The discriminator loss L_D is:

$$L_D = E_{x,y}[-\log D(x, y)] + E_x[-\log(1 - D(x, G(x)))] \quad (2)$$

Both networks are optimized using the Adam optimizer with a learning rate of 2×10^{-4}, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. During each training iteration, G generates an image, D evaluates both real and fake pairs, and gradients are alternately backpropagated to update the parameters of G and D.

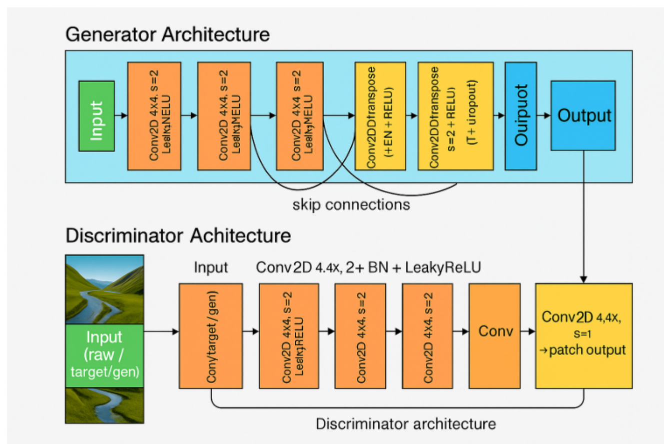

Fig. 2. GAN Model

C. Proposed GAN Model with PSO Integration

1) Overview: The enhanced model builds upon the Pix2Pix framework and integrates Particle Swarm Optimization (PSO) to improve generator performance. The architecture comprises two core components:

- **Generator:** Converts a raw input image into an enhanced version using a U-Net-style architecture with skip connections.
- **Discriminator:** Assesses the realism of the generated images by distinguishing between real (ground truth) and generated image pairs.

In addition to standard adversarial training, a PSO routine is introduced to fine-tune the generator's weights based on a fixed validation batch. This hybrid approach leverages both gradient-based updates and swarm-based optimization to improve image quality and convergence stability.

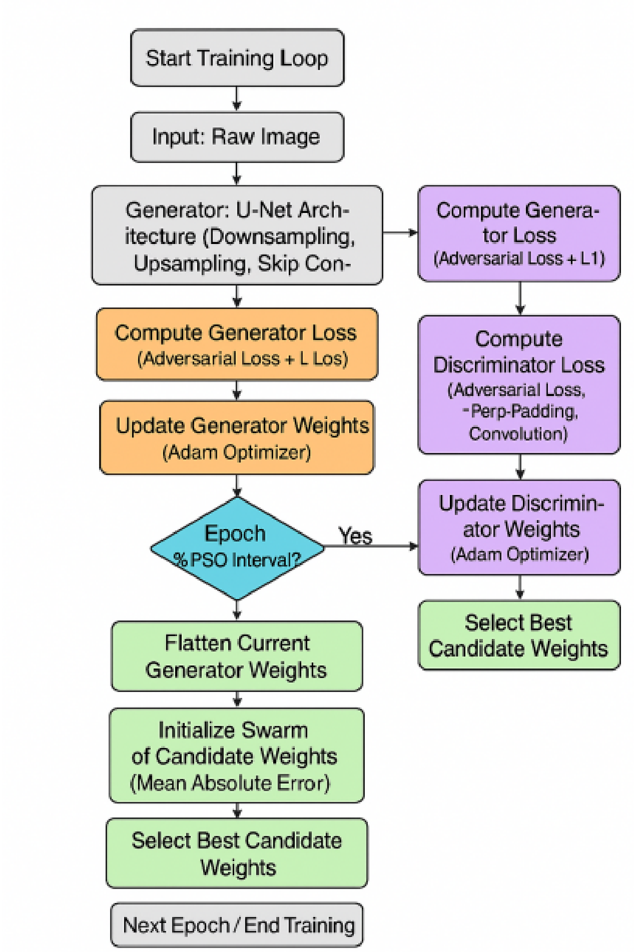

Fig. 3. Proposed GAN Model with PSO Integration

2) Loss Functions and Training:

- **Generator Loss:** Consists of:
  - *Adversarial Loss:* Binary cross-entropy loss encouraging the generator to produce outputs indistinguishable from real images.
  - *L1 Loss:* A pixel-wise loss enforcing structural similarity between generated and ground truth images, scaled by a weight factor $\lambda = 100$.

  The total generator loss is the sum of adversarial and L1 losses.
- **Discriminator Loss:** Comprises:
  - *Real Loss:* When real image pairs are correctly identified.
  - *Fake Loss:* When generated image pairs are identified as fake.

  The total discriminator loss is computed using binary cross-entropy.
- **Optimization:** Both the generator and discriminator are optimized using the Adam optimizer. The learning rate and momentum parameters are fine-tuned via PSO hypertuning.
- **Training Procedure with PSO:**
  - Standard training is conducted via mini-batch gradient descent.
  - At fixed intervals (e.g., every 2 epochs), a PSO-based fine-tuning step is performed on the generator's weights.

3) PSO-Based Fine-Tuning: Every two training epochs, the following steps are performed:

1) **Swarm Initialization:** Flatten the generator's weights into particle positions.
2) **Fitness Evaluation:** Compute mean absolute error (MAE) on a fixed validation batch.
3) **Velocity and Position Update:** Update particles using PSO equations based on personal and global bests.
4) **Parameter Injection:** Replace the generator's weights with those of the best-performing particle.

This hybrid approach combines PSO's global search capability with gradient descent's local refinement to achieve improved image quality and training stability.

D. Hardware/Software

- **Python** – The primary programming language used for implementing the GAN and PSO modules.
- **TensorFlow/Keras** – Deep learning frameworks used to build, train, and evaluate the GAN models.
- **OpenCV** – Utilized for image preprocessing tasks such as resizing, normalization, and visualization of outputs.
- **Kaggle Notebook** – Provided access to free P100 GPU resources, enabling faster model training and experimentation.
- **NumPy & Pandas** – Used for efficient data handling, numerical operations, and managing validation batches for PSO evaluation.
- **Matplotlib** – Used for plotting training curves, loss trends, and visual comparison of image enhancement results.

IV. Results and Discussion

A. Experimental Setup

Dataset: We use the Adobe FiveK dataset [?], which contains 5,000 raw images with expert retouches captured under diverse lighting and scene conditions. All images were resized to 256×256 and normalized to the range $[-1, 1]$.

Environment: Experiments were conducted on Kaggle using a P100 GPU.

Training Configuration: The models were trained for 100 epochs with a batch size of 16. The Adam optimizer was used with a learning rate of 2×10^{-4}, $\beta_1 = 0.5$, and $\beta_2 = 0.999$.

PSO Configuration: The PSO algorithm used a swarm size of 30, inertia weight of 0.7, and cognitive/social coefficients of 1.4 each.

B. Comparison/Result Graphs

The results indicate that PSO-GAN outperforms the normal GAN by providing a more stable training process with lower generator loss and balanced adversarial learning.

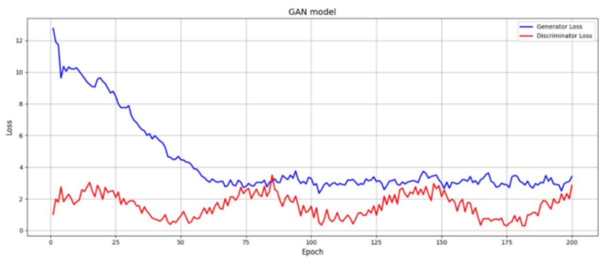

Fig. 4. Simple GAN Model

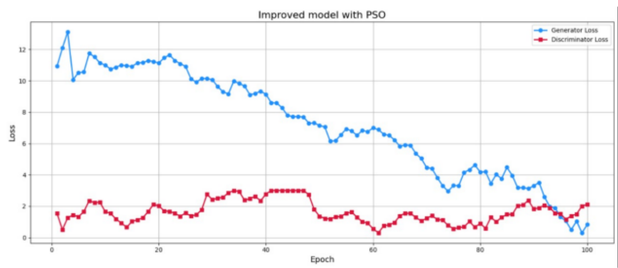

Fig. 5. Proposed GAN Model with PSO Integration

C. Observation and Inference

1) Normal GAN Results:

- **Generator Loss (Blue Curve):**
  - Starts at a high value (around 12) and sharply decreases during the first 50 epochs.
  - Stabilizes between 2 and 3 in the later stages of training.
  - Indicates early convergence and sustained generator learning performance.
- **Discriminator Loss (Red Curve):**
  - Exhibits fluctuations within a range of approximately 1 to 3.
  - Shows continuous adaptation to generator improvements.
  - Maintains balance without signs of collapse or overfitting.

2) Proposed GAN Model with PSO Integration:

- **Generator Loss (Blue Curve):**
  - Initial loss starts above 10 with significant fluctuations.
  - Gradual decline is observed from epoch 20 onwards.
  - Final generator loss approaches values below 2.
  - Indicates improved generator performance in producing realistic data over time.
- **Discriminator Loss (Red Curve):**
  - Loss oscillates within a stable range of 1 to 2.5.
  - Fluctuations represent the discriminator adapting to the evolving generator.
  - Suggests a well-maintained balance in the adversarial training process.

V. Conclusion

We show that integrating PSO with GAN improves convergence and image quality for enhancement tasks. Future work includes adaptive PSO, unpaired data (CycleGAN), and edge deployment.

VI. Future Work

- **Adaptive PSO Scheduling:** Introduce dynamic triggering of PSO fine-tuning based on training performance, rather than fixed intervals.
- **Support for Unpaired Datasets:** Extend the model to work with unpaired datasets by integrating architectures like CycleGAN.
- **Real-Time Deployment:** Optimize the model for deployment on edge devices or mobile platforms using model compression and inference acceleration techniques.
- **Multi-Objective Optimization:** Incorporate objectives such as perceptual quality and edge preservation alongside pixel-level loss for more balanced enhancement.
- **User-Guided Enhancement:** Enable feedback-driven optimization, allowing user preferences to guide enhancement quality.
- **Cross-Domain Generalization:** Assess the model's robustness by applying it to varied image domains such as medical imaging and satellite photography.

References

[1] B. Xu et al., "Image enhancement algorithm based on GAN neural network," *IEEE Access*, vol. 10, pp. 36766–36777, 2022.
[2] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[3] A. Creswell et al., "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.
[4] J. Kennedy, "Swarm intelligence," in *Handbook of Nature-Inspired Computing*, Springer, 2006, pp. 187–219.
[5] F. Marini and B. Walczak, "Particle swarm optimization (PSO): A tutorial," *Chemometrics*, vol. 149, pp. 153–165, 2015.
[6] M. Abdel-Basset et al., "Metaheuristic algorithms: A comprehensive review," in *Comp. Intelligence for Multimedia Big Data*, 2018.