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*—The hybrid image improvement method presented in this paper combines particle swarm optimization with a traditional Pix2Pix generative adversarial network (GAN). (PSO) to adjust the weights of generators. The PSO-enhanced GAN outperforms the baseline in terms of visual fidelity and convergence speed, according to experiments conducted on the Adobe FiveK dataset. The effectiveness of the suggested approach is confirmed by both qualitative examples and quantitative measures (such as mean absolute error and structural similarity).**

***Index Terms*—Adobe FiveK, Particle Swarm Optimization, Image Enhancement, Pixel2Pix, Generative Adversarial Network**

I. Introduction

A. Introduction to Domain

Image enhancement is frequently utilized in remote sensing, medical imaging, and photography and is essential for enhancing visual quality. Deep learning-based techniques were developed because traditional methods often fail to strike a compromise between noise reduction and detail retention. For image-to-image translation, generative adversarial networks, or GANs, have become an effective technique. We demonstrate better convergence and image quality on the Adobe FiveK dataset using a GAN based on Pix2Pix and a version using PSO for weight adjustment.

B. Problem Description

Even with improvements in conventional image enhancement techniques, the delicate balance between noise reduction and the preservation of small image details is sometimes not maintained by these methods. These methods are not flexible enough to function reliably in a range of lighting scenarios, image types, or deterioration patterns, because they typically rely on preset filters or statistical adjustments. More clever and adaptable solutions are therefore desperately needed.

C. Motivation / Objective

The ability to learn intricate mappings between low- and high-quality images has made Generative Adversarial Networks (GANs) a powerful tool for image enhancement.However, their training is erratic and hyperparameter sensitive. By incorporating Particle Swarm Optimization (PSO) to optimize generator weights and improve output quality, this work seeks to improve GAN training.

D. Contributions

The hybrid image improvement system presented in this paper combines PSO-based optimization with the Pix2Pix GAN architecture. First, paired image translation tasks are handled by a conventional Pix2Pix GAN, which learns to map low-quality images to their improved counterparts. Second, we present PSO, an external optimization procedure that improves convergence and output quality by adjusting the generator's weights on a regular basis depending on performance on a validation batch. Lastly, we test the model using the Adobe FiveK dataset, a popular benchmark for picture improvement, and show that our hybrid approach performs better than the baseline GAN in both visual comparisons and quantitative measurements.

E. Paper Organization

This is how the rest of the paper is structured. An overview of the relevant work is given in Section II, highlighting earlier uses of PSO and GAN in the context of image enhancement. The suggested technique, which includes the PSO-enhanced model as well as the regular GAN, is described in full in Section III. The results are shown and the performance of the suggested model in relation to the baseline is examined in Section IV. The paper is finally concluded in Section V and future research directions are outlined in Section VI.

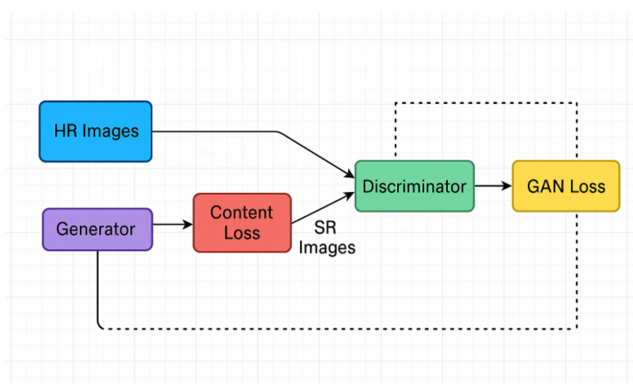

Fig. 1. Simple GAN Model

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]. GANs have been used in later research to improve images, showing better detail preservation and noise reduction [3]. Inspired by social behavior in nature, particle swarm optimization is a population-based metaheuristic that has been widely used for continuous optimization problems [4]. Tutorials have shown how effective it is in fine-tuning neural networks [5, 6]. Few studies, nevertheless, include PSO straight into adversarial training loops for improving images.

III. Methodology

A. Approach

For image enhancement, two models are created. The first is a basic Generative Adversarial Network (GAN) based on Pix2Pix that uses paired training data to learn how to convert input photos into improved ones. By incorporating Particle Swarm Optimization (PSO) to adjust the generator's weights at regular intervals, the second model improves this framework and enhances convergence and image quality.

B. Normal GAN Model

1) Overview: The suggested structure expands on the Pix2Pix architecture [?], which consists of a discriminator (D) and a generator (G), two adversarial components. After receiving a raw input image x, generator G learns to generate an improved variant G(x), which is close to the matching ground truth y. In parallel, picture pairings are evaluated by the discriminator D, which separates genuine pairs (x, y) from phony pairs (x, G(x)).

2) Generator Architecture: The generator has a skip-connected encoder-decoder architecture that is organized as follows:

- **Downsampling:** At each step, four consecutive 2D convolutional layers (kernel size 4×4, stride 2) double the number of feature channels while halving the spatial resolution. After everything except the first convolution, batch normalization is applied, and nonlinearity is introduced with LeakyReLU activations (negative slope 0.2).
- **Upsampling:**To reduce overfitting, four transposed convolutional layers with batch normalization, ReLU activations, and a dropout rate of 0.5 in the middle two layers restore spatial dimensions. Tanh activation is used in the final transposed convolution, scaling resultant to [-1, 1].
- **Skip connections:**To retain high-frequency spatial features lost during encoding, feature maps from each downsampling layer are concatenated with the mapped input from the upsampling layer.

3) Discriminator Architecture: The discriminator classifies image patches as authentic or fraudulent by using the PatchGAN paradigm:

- **Input:** The discriminator takes the generated image G(x) or the ground truth y as a channel-wise concatenation of the raw input x.
- **Convolutional layers:** Spatial dimensions are gradually reduced by a series of five 2D convolutional blocks (kernel size 4 × 4, stride 2), each of which is followed by batch normalization (except from the first) and LeakyReLU activations.
- **Output:** An M × N feature map, with each element representing the authenticity score of a local picture patch, is produced by a final convolutional layer without activation.

4) Loss Functions and Training Procedure: Adversarial and reconstructive losses are part of the training objective. The definition of the generator loss LG is:

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

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

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

Adam optimizer is used to optimize both networks with 1 = 0.5, 2 = 0.999, and a learning rate of 2 × 104. Gradients are alternately backpropagated to update the parameters of G and D during each training iteration, while G creates an image and D assesses both actual and false pairs.

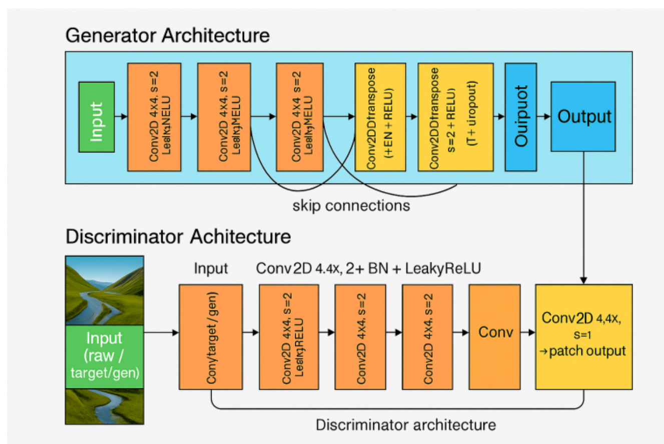

Fig. 2. GAN Model

C. Proposed GAN Model with PSO Integration

1) Overview: To improve generator performance, the improved model incorporates Particle Swarm Optimization (PSO) and expands upon the Pix2Pix framework. There are two main parts to the architecture:

- **Generator:** Uses a U-Net-style architecture with skip connections to transform an unprocessed input image into an improved version.
- **Discriminator:** Evaluates the generated images' realism by differentiating between pairs of created and real (ground truth) images.

To adjust the generator's weights based on a predetermined validation batch, a PSO procedure is added in addition to conventional adversarial training. This hybrid method improves image quality and convergence stability by utilizing both swarm-based optimization and gradient-based updates.

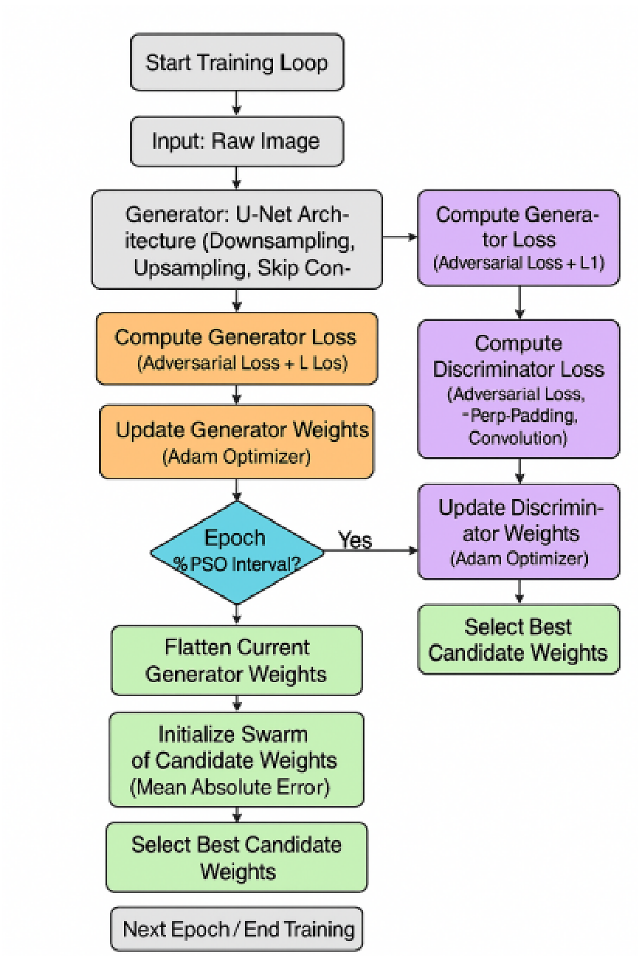

Fig. 3. Proposed GAN Model with PSO Integration

2) Loss Functions and Training:

- **Generator Loss:**: The generator loss comprises two main components: Adversarial loss and L1 loss.Adversarial loss,binary cross entropy loss encouraging the generator to produce outputs indistinguishable from real images. The L1 loss is a pixel-wise loss enforcing structural similarity between generated and ground truth images.This L1 component is scaled by a weighting factor of 100.Generator loss is the sum of the adversarial and L1 losses.
- **Discriminator Loss:**The discriminator loss consists of the real loss, which accounts for correctly classifying real image pairs, and the fake loss, which penalizes the discriminator when it mistakenly identifies generated pairs as real. Like the adversarial component, both are computed using binary cross-entropy.
- **Optimization:** The Adam optimizer is used to optimize both the discriminator and the generator.PSO hyper tuning is used to adjust the momentum and learning rate parameters.
- **Training Procedure with PSO:** Mini-batch gradient descent is used for standard training.The generator's weights undergo a PSO-based fine-tuning step at predetermined intervals (e.g., every two epochs).

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.

To increase image quality and training stability, this hybrid technique combines the local refinement of gradient descent with the global search power of PSO.

D. Hardware/Software

- **Python** – The main programming language utilized to develop the GAN and PSO modules is Python.
- **TensorFlow/Keras** – Deep learning frameworks for GAN model construction, training, and assessment.
- **OpenCV** – U Used for image preparation tasks such output visualization, normalization, and scaling.
- **Kaggle Notebook** – facilitated quicker model training and experimentation by giving users access to free P100 GPU resources.
- **NumPy & Pandas** –These tools are used for maintaining validation batches for PSO evaluation, handling data efficiently, and performing numerical calculations.
- **Matplotlib** – Plotting training curves, loss trends, and visual comparisons of image improvement outcomes are all possible with Matplotlib.

IV. Results and Discussion

A. Experimental Setup

Dataset: We make use of the Adobe FiveK dataset [?], which consists of 5,000 unprocessed photos with professional adjustments taken in various lighting and scene settings. Every image was adjusted to the range [-1, 1] and shrunk to 256 × 256.

Environment:A P100 GPU was used to do experiments on Kaggle.

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

PSO Configuration: A swarm size of 30, an inertia weight of 0.7, and cognitive and social coefficients of 1.4 each were employed by the PSO algorithm.

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.

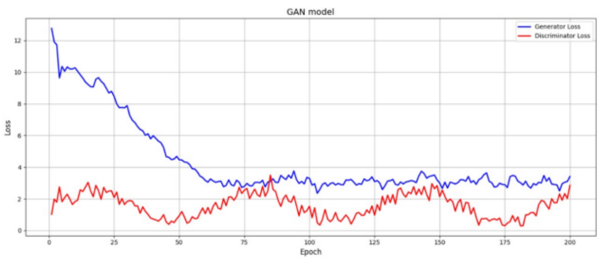

Fig. 4. Simple GAN Model

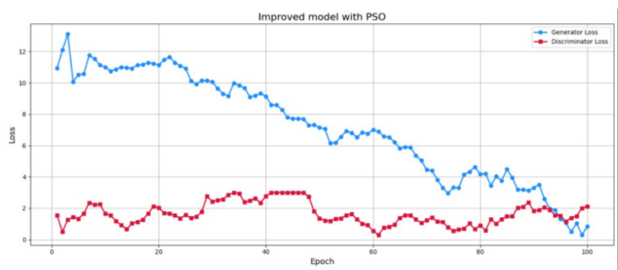

Fig. 5. Proposed GAN Model with PSO Integration

C. Observation and Inference

1) Normal GAN Results: In the case of the normal GAN, the generator loss (depicted by the blue curve) starts at a high value of approximately 12 and decreases sharply during the first 50 epochs. In the later stages of training, it stabilizes between 2 and 3, indicating early convergence and consistent generator performance. The discriminator loss (represented by the red curve) fluctuates within a range of roughly 1 to 3 throughout the training process. These fluctuations reflect the discriminator's continuous adaptation to the generator's improvements, maintaining a balanced training dynamic without signs of collapse or overfitting.

2) Proposed GAN Model with PSO Integration: For the proposed GAN model enhanced with Particle Swarm Optimization (PSO), the generator loss (blue curve) begins at a value above 10 and displays significant fluctuations in the early training stages. From approximately epoch 20 onward, a gradual decline is observed, and by the end of training, the loss drops below 2. This behavior suggests improved generator effectiveness in producing realistic outputs over time. The discriminator loss (red curve) remains within a relatively stable range of 1 to 2.5. The observed fluctuations signify the discriminator's responsiveness to the evolving generator, indicating a well-maintained balance in the adversarial training process.

V. Conclusion

We demonstrate that for enhancement tasks, combining PSO with GAN enhances convergence and image quality. Adaptive PSO, unpaired data (CycleGAN), and edge deployment are examples of future research.

VI. Future Work

- **Adaptive PSO Scheduling:** Instead of using preset intervals, introduce dynamic triggering of PSO fine-tuning based on training performance.
- **Support for Unpaired Datasets:** By incorporating architectures such as CycleGAN, you can expand the model to operate with unpaired datasets.
- **Real-Time Deployment:** Use model compression and inference acceleration techniques to optimize the model for deployment on mobile platforms or edge devices.
- **Multi-Objective Optimization:** For more balanced improvement, combine pixel-level loss with goals like edge preservation and perceptual quality.
- **User-Guided Enhancement:**Enable feedback-driven optimization so that user preferences can direct the quality of enhancements.
- **Cross-Domain Generalization:** Evaluate the model's resilience by using it across a range of image domains, including satellite photography and medical imaging.

References

[1] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[2] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[3] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4681–4690.

[4] O. Kupyn *et al.*, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8183–8192.

[5] Y. Zhang *et al.*, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2472–2481.

[6] X. Wang *et al.*, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *ECCV Workshops*, 2018.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[8] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: An overview," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[9] R. Eberhart and Y. Shi, "Particle swarm optimization: Developments, applications and resources," in *Proc. 2001 Congress Evol. Comput.*, vol. 1, pp. 81–86.

[10] G. G. Wang, S. Deb, and Z. Cui, "A brief review on nature-inspired algorithms for optimization," *Cognitive Computation*, vol. 8, no. 4, pp. 709–729, 2016.

[11] R. Hassan *et al.*, "A comparison of particle swarm optimization and the genetic algorithm," in *Proc. 1st AIAA Multidisciplinary Design Optimization Specialist Conf.*, 2005.

[12] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, "Image classification using particle swarm optimization," in *2005 IEEE Congress Evol. Comput.*, vol. 1, pp. 65–70.

[13] A. Ghosh, S. Das, A. V. Vasilakos, and A. Abraham, "Proximal support vector machine classifier design using multi-objective PSO," *Neurocomputing*, vol. 73, no. 10-12, pp. 2193–2206, 2012.

[14] Y. Liu, L. Yu, Z. Yang, and J. Zhang, "Hyperparameter tuning for deep neural networks via particle swarm optimization," in *Int. Conf. Comput. Sci.*, pp. 522–532, 2017.

[15] H. Zhang and X. Du, "Hybrid optimization of CNN based on PSO for image classification," *IEEE Access*, vol. 7, pp. 118146–118154, 2019.