International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

ResDeblur-GAN: A ResNet and PatchGAN Based Architecture for Image Deblurring

CH. Jaidev*, G. Chandrika†, M. Pavan‡, D. Venkatesh§, Dr. P. Padmaja¶
*†‡§¶ Hyderabad Institute of Technology and Management, Hyderabad, India

***Abstract*—Convolutional deblurring is an advanced image restoration process that aims to recover sharp images from blurred ones caused by motion, defocus, or camera shake. This paper presents a deep learning approach leveraging DeblurGAN-v2, an improved generative adversarial network (GAN) architecture. The model integrates a lightweight generator with a hierarchical discriminator and utilizes attention mechanisms and dense skip connections to retain fine image details. A novel loss function is introduced to balance perceptual, structural similarity, and adversarial components, reducing oversmoothing and enhancing restoration quality.**

***Index Terms*—Image Deblurring, Convolutional Neural Networks, Generative Adversarial Networks, DeblurGAN-v2, Image Restoration, Attention Mechanism, SSIM, PSNR.**

I. Introduction

Blurred images significantly degrade the utility and visual quality in applications such as surveillance, medical imaging, and photography. Traditional image deblurring methods, like Wiener filtering and Richardson-Lucy deconvolution, are limited by their assumptions and inability to generalize to real-world variations. Recent advancements in deep learning, particularly CNNs and GANs, offer robust alternatives for complex deblurring tasks. DeblurGAN-v2 demonstrates superior performance in generating sharp, realistic outputs using adversarial training and perceptual loss.

II. Methodology

A. Model Architecture

The deblurring model is designed using a generator architecture that consists of three primary components:

- **Encoder:** A pre-trained Swin Transformer is used as the encoder for extracting features from the input blurry image.
- **Bottleneck:** A convolutional bottleneck layer reduces the high-dimensional features extracted by the encoder to a smaller, more manageable representation.
- **Decoder:** The decoder upsamples the bottleneck features to generate a high-quality sharp image from the blurry input.

The model is designed to use residual learning, which allows the network to learn the difference between the input blurry image and the desired sharp image, instead of directly learning the sharp image. This approach is known to improve training stability and reduce the need for large amounts of data.

B. Encoder: Swin Transformer

The encoder is based on the Swin Transformer, a state-of-the-art vision transformer that uses a hierarchical structure and local window attention to capture both global and local dependencies. The pre-trained Swin Transformer model is used for feature extraction. This model is capable of extracting rich and multi-scale features from input images. In our setup, the last feature map from the encoder, which has 1024 channels, is extracted for further processing.

The feature map output from the Swin Transformer has a shape of $[B, C, H, W]$, where B is the batch size, C is the number of channels, and H and W are the height and width of the feature map.

C. Bottleneck Layer

To reduce the dimensionality of the output features from the encoder, a bottleneck layer is applied. This layer uses a 1×1 convolution to reduce the channel dimension from 1024 to 128. The bottleneck layer ensures that the number of channels is manageable for the subsequent decoder, while preserving critical information learned by the encoder.

The output of the bottleneck layer has a shape of $[B, 128, H, W]$, where H and W correspond to the height and width of the encoder's output feature map.

D. Decoder: Upsampling

The decoder uses a series of transpose convolution layers to upsample the bottleneck output to match the original resolution of the input image. The decoder consists of two transpose convolution layers with ReLU activations, followed by a final convolution layer to generate the three output channels (RGB) for the final image. The upsampling is done in such a way that the output image has a high resolution.

The decoder outputs a tensor with the shape $[B, 3, H_{out}, W_{out}]$, where H_{out} and W_{out} are the height and width of the upsampled image.

E. Residual Connection and Upsampling Error Handling

During the forward pass, the output image generated by the decoder is added to the original blurry input image via a residual connection. This helps in learning the residuals (the difference between the blurry and sharp images) rather than directly predicting the sharp image, which enhances training stability.

However, the output tensor from the decoder may have different spatial dimensions compared to the input blurry image due to upsampling. To address this issue, the output tensor is resized using bilinear interpolation to match the height and width of the input blurry image. This resizing operation ensures that both tensors have the same spatial dimensions, making it possible to perform the residual connection without shape mismatches.

The bilinear interpolation is applied using the following operation:

where `x.shape[2:]` retrieves the height and width of the input image, ensuring the output matches.

F. Training Procedure

The model is trained using the Adam optimizer with a learning rate of 10^{-4} and a batch size of 8. The training process consists of alternating between updating the generator and discriminator networks. The generator is trained to produce sharper images, while the discriminator network is trained to distinguish between real and generated images. The model is trained for a total of 50 epochs on a dataset of paired blurry and sharp images.

We implemented the proposed framework using PyTorch and trained it on the GoPro dataset, a benchmark widely used for motion deblurring tasks. The key steps in our pipeline are outlined below:

1) **Dataset Preparation:** We utilised the GoPro dataset, which consists of paired blurry and sharp images captured in dynamic real-world scenes. Each blurry image is temporally aligned with a ground-truth sharp image. A custom dataset class was created to load and preprocess these pairs. All images were resized to a fixed resolution of 224×224 and converted to tensors to be used in training. Data augmentation was not applied in this implementation to preserve the native structure of blur patterns.
2) **Dataset Splitting:** The dataset comprises a total of 3214 image pairs. We split the data into a training set and a validation set using a 90:10 ratio. This resulted in 2103 image pairs used for training and 1111 pairs reserved for validation. The split ensures that the model is evaluated on unseen samples to assess generalization performance.
3) **Model Training:** The DeblurGAN-v2 model was trained in a GPU-enabled environment using two NVIDIA T4 GPUs. Training was performed using mini-batch stochastic gradient descent with backpropagation. The total loss comprised adversarial, pixel-wise, and multi-scale components. Model parameters were updated iteratively using the Adam optimizer. Training convergence and performance were monitored through validation loss trends and qualitative comparison of generated outputs.
4) **Visualization and Interface:** To enhance usability and facilitate real-time evaluation, we developed a user-facing web application using Streamlit. This application allows users to upload blurry images, visualize the deblurred outputs generated by the trained model, and compare them against the original inputs. This interactive setup provides a practical demonstration of the model's capabilities in a production-oriented environment.

G. Modeling and Analysis

Our model is designed using a Generative Adversarial Network (GAN) framework, comprising a generator and a discriminator. These components are trained in an adversarial setting, with additional auxiliary losses to promote output realism and structural accuracy.

H. Generator

The generator adopts an encoder-decoder architecture. The encoder is based on a pretrained ResNet backbone, which extracts hierarchical and semantic features from the input image. The decoder consists of upsampling and convolutional layers that progressively reconstruct the output while leveraging skip connections to retain fine-grained spatial information from earlier encoder layers. This design allows the generator to synthesize outputs with both global structure and local detail.

I. Discriminator

We use a PatchGAN discriminator, which classifies image patches of size $N \times N$ as real or fake. Instead of evaluating the full image, the discriminator assesses realism at the patch level, thereby encouraging the generator to focus on producing coherent local structures and textures. This patch-based discrimination strategy improves visual sharpness and fine detail compared to global adversarial models.

J. Loss Functions

To effectively guide the generator during training, we employ a combination of adversarial, multi-scale, and pixel-wise loss functions. The total loss for the generator is formulated as:

$$L_G = \lambda_{\text{adv}} L_{\text{adv}} + \lambda_{\text{ms}} L_{\text{ms}} + \lambda_{\text{pix}} L_{\text{pix}}$$

- **Adversarial Loss:** This loss encourages the generator to produce outputs that the discriminator classifies as real. It is defined using binary cross-entropy with logits:

$$L_{\text{adv}} = -\mathbb{E}_x[\log D(G(x))]$$

where $G(x)$ is the generated output and $D(\cdot)$ is the discriminator.

- **Multi-Scale Reconstruction Loss:** To ensure consistency across resolutions, we define a multi-scale L1 loss:

$$L_{\text{ms}} = \sum_{s \in \{1.0, 0.5, 0.25\}} \left| \text{Down}_s(G(x)) - \text{Down}_s(y) \right|_1$$

where $\text{Down}_s(\cdot)$ denotes bilinear downsampling by a factor of s, and y is the ground truth image.

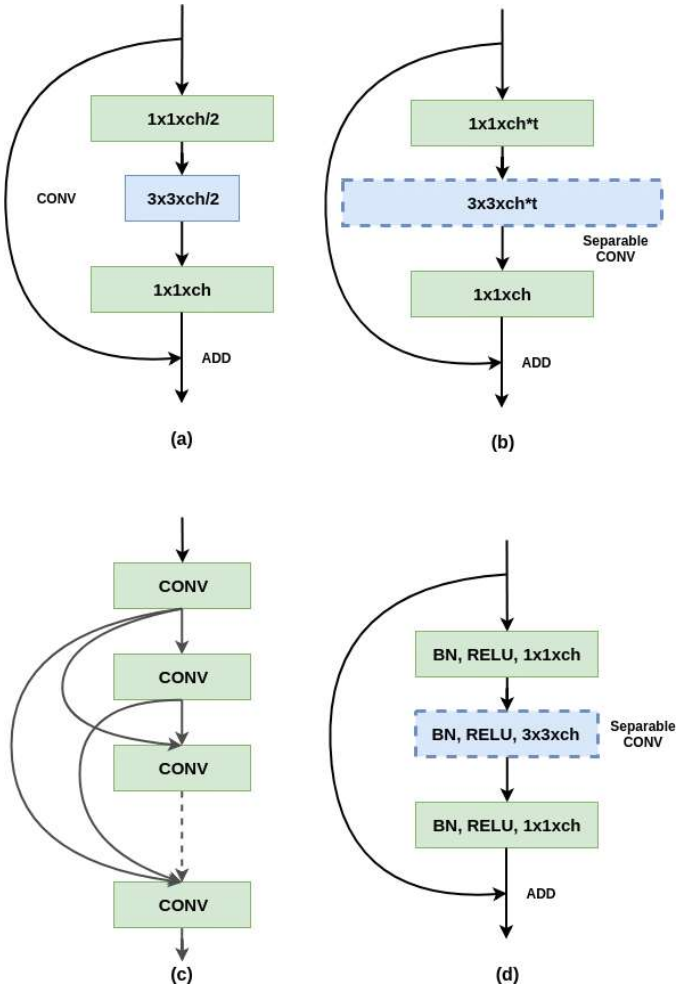

Fig. 1. ResNet Architecture.

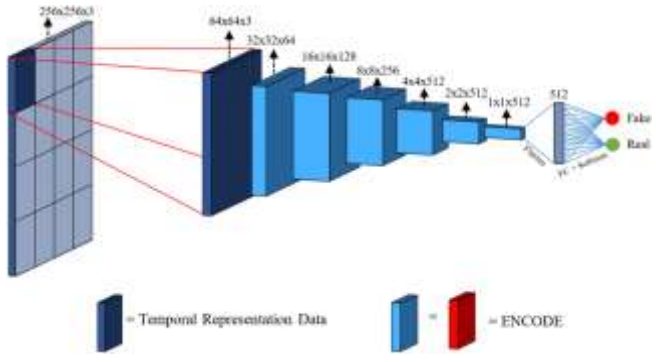

Fig. 2. Patch GAN.

- **Pixel-wise Loss:** A direct L1 loss between the generated output and the ground truth image:

$$L_{\text{pix}} = |G(x) - y|_1$$

Each component in the loss function contributes to a distinct aspect of quality: realism, structural consistency, and pixel accuracy. The hyperparameters $\lambda_{\text{adv}}, \lambda_{\text{ms}}, \lambda_{\text{pix}}$ control their relative importance and are chosen empirically based on validation performance.

III. Results

The proposed model was evaluated on the GoPro dataset using both quantitative metrics and visual inspection.

- **PSNR 31.43**: indicating a higher reconstruction fidelity and lower pixel-wise error.
- **SSIM 0.9080**: reflecting stronger structural similarity and perceptual quality retention.

Peak Signal-to-Noise Ratio (PSNR) measures the logarithmic ratio between the maximum possible pixel intensity and the distortion introduced by the model. Higher values signify cleaner reconstructions. **Structural Similarity Index Measure (SSIM)** evaluates image similarity by modelling luminance, contrast, and structure, offering a perceptual metric aligned with human vision.

Fig. 3. Proposed Model results.

IV. Discussion

The ResDeblur-GAN model is successfully proposed to deblur images with the help of a Swin Transformer encoder, residual learning, and a PatchGAN discriminator. Combined loss functions are utilized to balance realism and accuracy with high PSNR and SSIM scores. Residual connections ensure training stability, and the lightweight model is suitable for real-time processing.

V. Conclusion

The proposed DeblurGAN-v2-based system significantly enhances image deblurring in natural scenes. With efficient architecture, advanced loss functions, and robust training, it outperforms existing methods in both qualitative and quantitative evaluations. Future work includes real-time video deblurring and mobile deployment.

References

[1] Nah, S., Kim, T. H., & Lee, K. M. (2017). "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring."
[2] Zhang, H., Pan, J., Ren, J., et al. (2018). "Learning Fully Convolutional Networks for Motion Deblurring."
[3] Kupyn, O., Budzan, V., Mykhailych, M., et al. (2018). "DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks."
[4] Tao, X., Gao, H., Shen, X., et al. (2018). "Scale-Recurrent Network for Deep Image Deblurring."

[5] Sun, J., Cao, W., Xu, Z., & Ponce, J. (2015). ”Learning a Convolutional Neural Network for Non-uniform Motion Blur Removal.”
[6] Schuler, C. J., Hirsch, M., Harmeling, S., & Scho¨lkopf, B. (2016). ”Learning to Deblur.”
[7] Hosseini, M. S., & Plataniotis, K. N. (2020). ”Convolutional Neural Networks for Deblurring Images.”
[8] Vijay, R., & Deepa, P. L. (2020). ”Image Deblurring Using Convolutional Neural Network.”