HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Project 2 Report

Image Deblurring on Gopro Dataset using Improved U-net model

Submitted by

Ngo Duy Dat - 20225480

Guided by

Dr. Do Tuan Anh

Hanoi, June 2025

Abstract

This report presents a project on image deblurring utilizing an improved U-Net model [3] applied to the GoPro dataset. The objective is to restore sharp images from blurred inputs caused by camera motion, a common challenge in dynamic scene capture. The project implements the models proposed in the paper, enhancing the traditional U-Net architecture by integrating two-dimensional Haar wavelet transforms for downsampling and upsampling, depth-wise separable convolutions, residual connections, and a DMRFC (Dense Multi-Receptive Field Channel) module. The report evaluates the model's performance using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) on the GoPro dataset, comparing it against traditional deblurring methods and existing deep learning approaches.

Contents

Part I
Introduction

Image deblurring is a fundamental problem in computer vision, essential for applications such as autonomous driving, surveillance, and photography, where clear images are critical. This project implements an improved U-Net model proposed by [3], to address the challenges of motion blur in dynamic scenes. The model incorporates two-dimensional Haar wavelet transforms, depth-wise separable convolutions, residual connections, and a DMRFC module to enhance deblurring performance. The following subsections outline the motivation behind this work, the problem formulation, and the GoPro dataset used for evaluation.

1 Motivation

Motion blur, caused by camera shake or object movement, degrades image quality and hinders downstream tasks like object detection and recognition. Traditional deblurring methods, such as Wiener filtering and Lucy-Richardson deconvolution, struggle with complex, non-uniform blur and are sensitive to noise, resulting in poor restoration quality. Recent deep learning advancements, particularly in convolutional neural networks (CNNs), have shown superior performance by learning complex blur patterns directly from data. The U-Net architecture, known for its effectiveness in image-to-image tasks, provides a robust foundation for deblurring. Inspired by Lian et al. (2022), this project aims to enhance U-Net with wavelet transforms, lightweight convolutions, and multi-scale feature extraction to achieve high-quality deblurring with reduced computational cost, making it suitable for real-world applications.

2 Problem formulation

The image deblurring problem can be formulated as recovering a sharp image S from a blurred image B, where $B = S * k + n$, with k representing the unknown blur kernel and n denoting additive noise. Traditional methods rely on estimating k, which is challenging for non-uniform blur in dynamic scenes. The proposed approach uses an improved U-Net model to learn a mapping $f : B \to S$ directly from paired blurry and sharp images. The objective is to minimize a defined loss function.

3 The Dataset

The GoPro dataset is used for training and evaluating the proposed model. It consists of 3,214 pairs of blurry and sharp images captured at 720p resolution using GoPro cameras, simulating realistic motion blur from camera shake in dynamic scenes. The dataset is divided into 2,103 training pairs and 1,111 testing pairs. To enhance model generalization, data augmentation techniques, including random rotations (90°, 180°, 270°), flips, are applied to the training set. Images are cropped to 256×256 pixels during training to prevent overfitting. The GoPro dataset's diverse blur patterns make it an ideal benchmark for evaluating the proposed model's ability to handle complex motion blur.

Part II
Traditional Methods

Image deblurring has been a longstanding challenge addressed by traditional signal processing techniques, which attempt to reverse the blurring process by estimating the blur kernel and reconstructing the sharp image. This section provides a detailed review of three widely used traditional methods—Wiener filtering, Lucy-Richardson deconvolution, and regularized deconvolution—and summarizes their limitations, to explain why they are insufficient for complex real-world deblurring tasks, such as those evaluated on the GoPro dataset.

1 Wiener Filtering

Wiener filtering is a foundational deblurring technique that operates in the frequency domain to restore a sharp image by minimizing the mean squared error between the estimated and true images. For a blurred image modeled as $B = S * k + n$, where S is the sharp image, k is the blur kernel, and n is additive noise, Wiener filtering computes the estimate $\hat{S}$ using:

$$\hat{S}(u,v) = \frac{H^*(u,v)B(u,v)}{|H(u,v)|^2 + \frac{S_n(u,v)}{S_s(u,v)}}, \tag{1}$$

where $H(u,v)$ is the Fourier transform of the blur kernel, $H^*(u,v)$ is its complex conjugate, and $\frac{S_n(u,v)}{S_s(u,v)}$ represents the noise-to-signal power ratio, often approximated as a constant γ. This method assumes the blur kernel is known or can be estimated, typically from image features or camera motion data. Wiener filtering is computationally efficient due to its use of Fast Fourier Transforms (FFT) and performs well for uniform blur scenarios, such as defocus blur in controlled settings. However, its reliance on accurate kernel estimation makes it vulnerable to errors in dynamic scenes, where motion blur varies spatially. Additionally, the method amplifies noise in regions with low signal strength, leading to artifacts like ringing around edges.

2 Lucy-Richardson Deconvolution

The Lucy-Richardson deconvolution is an iterative method rooted in maximum likelihood estimation, designed for images corrupted by Poisson noise, common in low-light photography. The algorithm iteratively refines the sharp image estimate using:

$$S_{t+1} = S_t \cdot \left(k * \frac{B}{k * S_t} \right), \tag{2}$$

where S_t is the estimated sharp image at iteration t, B is the blurred image, and k is the blur kernel. The term $\frac{B}{k*S_t}$ represents the ratio of observed to predicted blurred images, and the convolution with k adjusts the estimate to align with the observed data. Starting with an initial guess (e.g., the blurred image itself), the method enhances edges and details over iterations, making it suitable for astronomical or medical imaging with known blur kernels. However, its iterative nature makes it sensitive to noise, which can accumulate and cause ringing artifacts or divergence if too many iterations are performed. The method also requires manual tuning of the iteration count and assumes a spatially invariant blur kernel, limiting its effectiveness for non-uniform motion blur in real-world scenarios.

3 Regularized Deconvolution

Regularized deconvolution methods improve upon basic deconvolution by incorporating prior knowledge about the sharp image to mitigate noise and kernel estimation errors. The approach formulates deblurring as an optimization problem:

$$\hat{S} = \arg\min_{S} \|B - k * S\|^2 + \lambda R(S), \tag{3}$$

where $\|B - k * S\|^2$ is the data fidelity term ensuring the restored image matches the blurred observation, $R(S)$ is a regularization term imposing constraints on S, and λ is a weighting parameter. Common regularization terms include Tikhonov regularization, which promotes smoothness ($R(S) = \|S\|^2$), and total variation (TV) regularization, which preserves edges by minimizing $R(S) = \int |\nabla S|$. TV-based methods are particularly effective for piecewise-smooth images, such as natural scenes with distinct objects. Regularized deconvolution is solved using optimization techniques like gradient descent or conjugate gradient methods, offering robustness to noise compared to Wiener or Lucy-Richardson methods. However, the method still depends on accurate kernel estimation and requires careful selection of λ, which can be computationally expensive to optimize. Additionally, it struggles with complex, spatially varying blur, as the regularization term may oversmooth fine details or fail to capture intricate textures.

4 Weaknesses of Traditional Methods

Despite their contributions, traditional deblurring methods exhibit significant limitations that hinder their performance in complex, real-world scenarios like those in the GoPro dataset:

- **Dependence on Accurate Kernel Estimation**: All methods require a precise blur kernel k, which is challenging to estimate in dynamic scenes with non-uniform motion blur, resulting in artifacts and incomplete deblurring.

- **Noise Amplification**: Wiener filtering and Lucy-Richardson deconvolution are prone to amplifying noise, particularly in low-contrast regions, leading to ringing artifacts and degraded image quality.

- **Inability to Handle Non-Uniform Blur**: These methods assume spatially invariant blur, making them ill-suited for motion blur caused by camera shake or object movement, which varies across the image.

- **Parameter Tuning and Computational Cost**: Lucy-Richardson and regularized deconvolution require manual tuning of parameters (e.g., iteration counts or λ) and can be computationally intensive, especially for high-resolution images.
- **Lack of Semantic Context**: Operating solely on low-level pixel statistics, these methods fail to leverage high-level semantic information, limiting their ability to reconstruct fine details, textures, or object boundaries.

These shortcomings highlight the need for advanced deep learning approaches, such as the improved U-Net model proposed in this project, which leverages data-driven learning to model complex blur patterns, reduce noise sensitivity, and incorporate semantic understanding for superior deblurring performance.

Part III
Improved Unet model

The proposed image deblurring method builds on the U-Net architecture, a popular deep learning model for image-to-image tasks, and enhances it with several modifications proposed by [3]. These improvements include two-dimensional Haar wavelet transforms for downsampling and upsampling, depth-wise separable convolutions, residual connections, and a Dense Multi-Receptive Field Channel (DMRFC) module. This section describes the standard U-Net architecture, details the proposed enhancements, and explains how they improve deblurring performance on the GoPro dataset.

1 Standard U-Net Architecture

The U-Net model, originally developed for medical image segmentation, is well-suited for image deblurring due to its encoder-decoder structure with skip connections. The encoder consists of convolutional layers and max-pooling operations that reduce the spatial dimensions of the input image while extracting high-level features. The decoder uses upsampling and convolutional layers to restore the image to its original size, combining low-level details from the encoder via skip connections. These connections help preserve fine details, such as edges and textures, which are critical for deblurring. For an input blurred image B, the U-Net learns a mapping $f : B \to S$, where S is the sharp image, by minimizing a loss function, typically Mean Squared Error (MSE). However, the standard U-Net struggles with capturing multi-scale features and can be computationally heavy, prompting the need for enhancements.

2 Improved U-net Architecture

The improved U-Net model follows a U-shaped architecture with an encoder, a bottleneck, and a decoder, as shown in Figure 1. The encoder extracts hierarchical features from the input blurred image, reducing spatial dimensions while increasing feature depth. The decoder reconstructs the sharp image by upsampling these features, combining them with low-level details from the encoder via skip connections. The input is a blurred RGB image of size $256 \times 256 \times 3$, and the output is a sharp image of the same size. Unlike the standard U-Net, which uses max-pooling and transposed convolutions, this model employs Haar wavelet transforms for spatial transformations, depth-wise separable convolutions for efficiency, residual connections for stable training, and a DMRFC module at the bottleneck to capture multi-scale features.

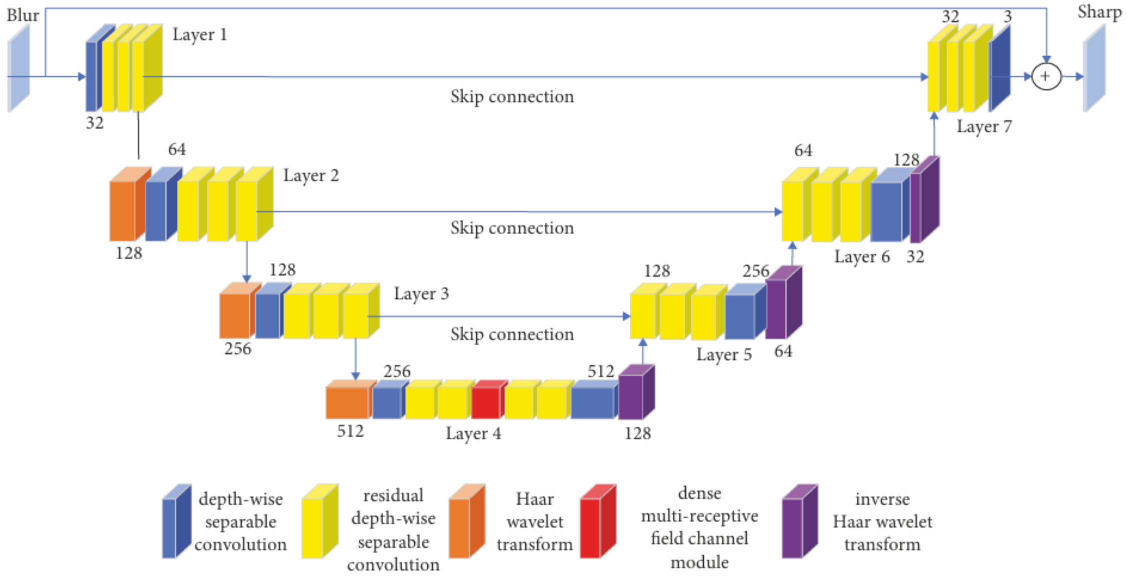


Figure 1 . Improved U-net model architecture

2.1 Encoder

The encoder consists of four stages, each reducing the spatial dimensions by half while doubling the number of feature channels. Each stage includes:

- **Haar Wavelet Downsampling**: A two-dimensional Haar wavelet transform decomposes the input into four sub-bands (low-frequency LL, and high-frequency LH, HL, HH). The LL sub-band, containing the approximation, is used for downsampling, reducing the resolution (e.g., from 256×256 to 128×128).

- **Depth-Wise Separable Convolutions**: Two depth-wise separable convolution layers extract features, each followed by batch normalization and ReLU activation. These layers reduce the parameter count compared to standard convolutions, maintaining efficiency.

- **Residual Connections**: A residual connection adds the input to the output of the convolution layers, aiding gradient flow and preserving low-level features.

The encoder processes the input image through these stages, producing feature maps of sizes $128 \times 128 \times 64$, $64 \times 64 \times 128$, $32 \times 32 \times 256$, and $16 \times 16 \times 512$.

2.2 Bottleneck

The bottleneck, located at the deepest layer, processes the most compact feature map ($16 \times 16 \times 512$) using the DMRFC module. This module employs parallel depth-wise separable convolutions with kernel sizes of 3×3, 5×5, and 7×7 to capture features at different scales. The outputs are densely connected, meaning each convolution's output is fed into subsequent layers, and then concatenated. A 1×1 convolution reduces the channel dimensions, producing a rich feature map that encodes both local and global context. This enhances the model's ability to handle non-uniform motion blur in dynamic scenes.

2.3 Decoder

The decoder mirrors the encoder with four stages, each upsampling the feature map to double its spatial dimensions while halving the number of channels. Each stage includes:

- **Inverse Haar Wavelet Upsampling**: The inverse Haar wavelet transform reconstructs the feature map using the low-frequency and high-frequency sub-bands, increasing resolution (e.g., from 16×16 to 32×32).
- **Skip Connections**: Features from the corresponding encoder stage are concatenated with the upsampled features, preserving low-level details like edges and textures.
- **Depth-Wise Separable Convolutions**: Two depth-wise separable convolution layers, with batch normalization and ReLU activation, refine the features.
- **Residual Connections**: A residual connection adds the input to the convolution output, stabilizing training.

The final decoder stage produces a feature map of size $256 \times 256 \times 3$, which is passed through a 1×1 convolution with a sigmoid activation to generate the sharp image.

2.4 Loss Function

The model is trained to minimize a combined loss function of Mean Squared Error (MSE) and Structural Similarity Index (SSIM), defined as:

$$L_{\text{total}} = L_{\text{MSE}} + w_1 L_{\text{SSIM}}, \tag{4}$$

where $L_{\text{MSE}} = \|R - S\|^2$, $L_{\text{SSIM}} = 1 - \text{SSIM}(R, S)$, R is the restored image, S is the ground-truth sharp image, and $w_1 = 0.001$. This loss ensures pixel-level accuracy and structural fidelity, optimizing the model for high-quality deblurring on the GoPro dataset.

3 Haar Wavelet Transforms

The Haar wavelet transform is a cornerstone of the model, replacing conventional downsampling and upsampling techniques with a multi-resolution decomposition approach. This method breaks down an image into four sub-bands—LL (low-frequency), LH (horizontal high-frequency), HL

(vertical high-frequency), and HH (diagonal high-frequency)—enabling the model to capture and reconstruct details at various scales. The following sections explain its mechanism, why it is effective, and the detailed process of decomposition and reconstruction.

3.1 Mechanism of Haar Wavelet Transform

The Haar wavelet transform operates on the principle of wavelet analysis, using the simplest wavelet basis—step functions—to decompose signals into different frequency bands. Its mechanism involves a two-step process applied in two dimensions (rows and columns), leveraging the Haar mother wavelet, defined as:

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 0.5, \\ -1 & 0.5 \leq t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

The scaling function (father wavelet) is a constant value over the interval, used for averaging. This simplicity makes the Haar transform computationally efficient and suitable for real-time applications.

- One-Dimensional Decomposition: For a sequence of two adjacent pixels $[x_1, x_2]$, the Haar transform computes:

$$LL = \frac{x_1 + x_2}{2}, \quad LH = \frac{x_1 - x_2}{2}.$$

Here, LL represents the average (low-frequency component), capturing the overall trend, while LH represents the difference (high-frequency component), highlighting variations such as edges. This process is repeated across all pairs in a row or column.

- Two-Dimensional Decomposition: The one-dimensional transform is first applied to all rows of the input image I (size $N \times N$), producing intermediate low-frequency and high-frequency sub-bands. Then, it is applied to all columns of these intermediates, yielding the four sub-bands:

$$I \rightarrow \{LL, LH, HL, HH\}.$$

For an $N \times N$ image, each sub-band is $\frac{N}{2} \times \frac{N}{2}$, and LL is used for downsampling in the encoder, reducing resolution while preserving essential information.

3.1.1 Decomposition Process:

The decomposition is implemented as a filter bank operation. For a 2×2 block of pixels $I = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}$, the process is:

- **Row-wise Transform**: Apply the Haar transform to each row:

$$[x_{11}, x_{12}] \rightarrow \left[\frac{x_{11} + x_{12}}{2}, \frac{x_{11} - x_{12}}{2}\right],$$

$$[x_{21}, x_{22}] \rightarrow \left[\frac{x_{21} + x_{22}}{2}, \frac{x_{21} - x_{22}}{2}\right].$$

  This yields an intermediate matrix.

- **Column-wise Transform**: Apply the Haar transform to the columns of the intermediate matrix:
$$\left[\frac{x_{11}+x_{12}}{2}, \frac{x_{21}+x_{22}}{2}\right] \to \left[\frac{(x_{11}+x_{12})+(x_{21}+x_{22})}{4}, \frac{(x_{11}+x_{12})-(x_{21}+x_{22})}{4}\right],$$
and similarly for the difference terms, producing LL, LH, HL, and HH. This recursive application across the entire image generates the four sub-bands, with LL serving as the downsampled input for the next encoder layer.

3.1.2 Reconstruction Process

The inverse Haar wavelet transform reconstructs the original image I from the four sub-bands during upsampling in the decoder. This process reverses the decomposition, ensuring all details are restored:

- **Inverse Column Transform:** For each pair of LL and LH (or HL and HH) in the column direction:
$$LL + LH \to x_1, \quad LL - LH \to x_2,$$
where x_1 and x_2 are the reconstructed pixel values. This step recovers the intermediate column-wise representation.

- **Inverse Row Transform**: Apply the inverse transform to the rows of the intermediate results using all four sub-bands:
$$\frac{LL_{\text{row}} + LH_{\text{row}}}{2} \to x_1, \quad \frac{LL_{\text{row}} - LH_{\text{row}}}{2} \to x_2,$$
and similarly for HL and HH. This step restores the full $N \times N$ image by combining the low-frequency structure with high-frequency details.

3.2 Why Haar Wavelet Transform Works

The Haar wavelet transform is effective for image deblurring because it provides a simple yet powerful way to analyze an image at multiple resolution levels, which is critical for handling the varied blur patterns in dynamic scenes like those in the GoPro dataset. Unlike max-pooling, which discards much of the high-frequency information, the Haar transform preserves both low-frequency (structural) and high-frequency (detail) components. This preservation is key for deblurring, as motion blur often distorts edges and textures—high-frequency features—that need to be recovered. Additionally, its ability to reconstruct the original image from all sub-bands during upsampling ensures minimal loss of information, making it superior for tasks requiring fine detail restoration. This multi-resolution capability aligns with the U-Net's need to process features hierarchically, enhancing the model's ability to learn complex blur mappings directly from data.

Benefits in the Model: The Haar wavelet transform's ability to separate and reconstruct multi-resolution features makes it ideal for deblurring. It reduces information loss compared to max-pooling by retaining high-frequency details, which are essential for recovering blurred edges in the GoPro dataset. Its computational efficiency, due to the simple averaging and differencing operations, complements the model's depth-wise separable convolutions, while its multi-scale nature supports the DMRFC module's feature extraction.

4 Depth-Wise Separable Convolutions

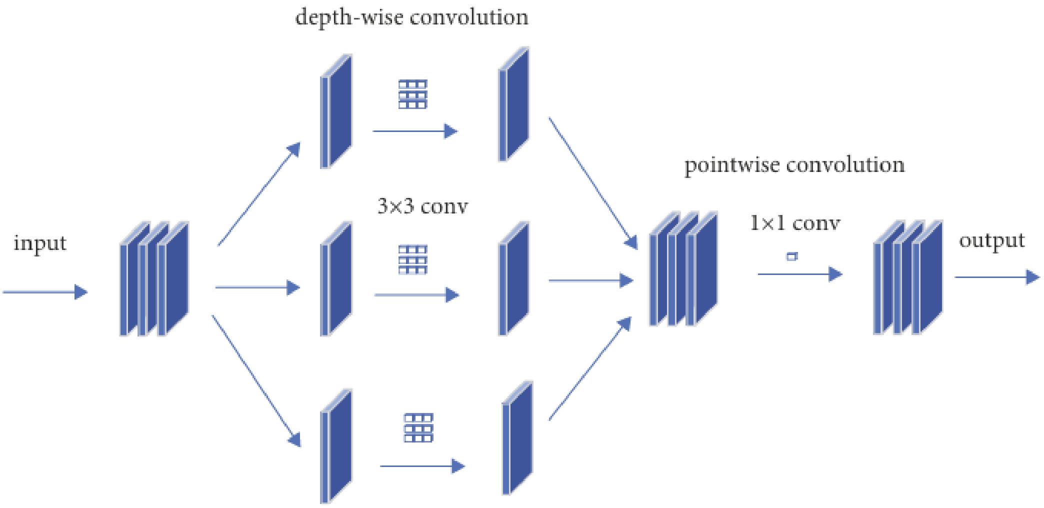


Figure 2 . Depth-wise separablec onvolution structure

To reduce computational complexity, the model replaces standard convolutional layers with depth-wise separable convolutions. A standard convolution applies filters across all input channels simultaneously, requiring many parameters. In contrast, depth-wise separable convolution splits the process into two steps: a depth-wise convolution applies a single filter to each input channel, and a point-wise convolution (1×1 kernel) combines the outputs. This reduces the number of parameters and computations while maintaining feature extraction capability. For a convolution with C_{in} input channels, C_{out} output channels, and kernel size K, the parameter count is reduced from $K^2 \cdot C_{\text{in}} \cdot C_{\text{out}}$ to $K^2 \cdot C_{\text{in}} + C_{\text{in}} \cdot C_{\text{out}}$, making the model lighter and faster, ideal for real-time applications.

5 Residual Depth-Wise Separable Convolution

The residual depth-wise separable convolution uses two sequential depth-wise separable convolution layers. Each layer applies a depth-wise convolution (filtering each input channel separately) followed by a point-wise convolution (combining channels with a 1×1 kernel), reducing the number of parameters compared to standard convolutions. A skip connection then adds the input x to the output of these layers, computed as:

$$y = F(x) + x, \tag{5}$$

where $F(x)$ represents the result of the two depth-wise separable convolutions, and y is the final output. This addition stabilizes training by allowing gradients to flow directly through the skip connection, mitigating the vanishing gradient problem.

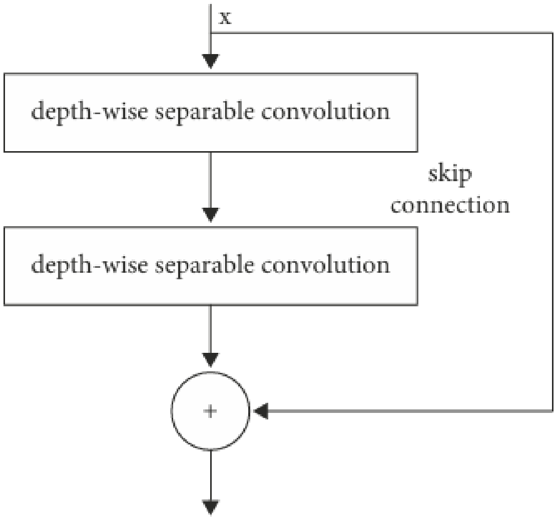

x
depth-wise separable convolution
skip connection
depth-wise separable convolution
+

Figure 3 . Structure of the residual depth-wise separable convolution

6 Dense Multi-Receptive Field Channel (DMRFC) Module

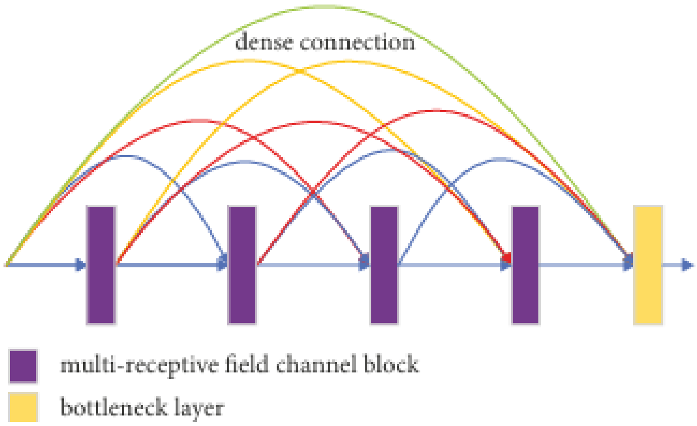


Figure 4 . The DMRFC module

The DMRFC module is a novel addition to capture multi-scale features, addressing the challenge of varying blur patterns in dynamic scenes. The DMRFC module i is composed of four multireceptive field channel blocks and a bottleneck layer.The semantic features of the image are extracted through multireceptive field channel blocks. These features are densely connected, meaning each block's output is fed into subsequent blocks, promoting feature reuse. The outputs are then concatenated and processed by a 1×1 convolution to reduce channel dimensions, forming a compact feature map. This design allows the model to capture both local and global context, improving its ability to handle non-uniform motion blur.

Multi-receptive field channel (MRFC) block: Each Multi-Receptive Field Block uses four parallel feature extraction branches, all employing 3×3 convolution kernels but with different extensional rates of 1, 3, 5, and 7. These varying rates expand the receptive field of each branch, allowing the block to capture features at different scales—ranging from fine local details to broader contextual information. The connection operation then merges the feature maps from these four branches, combining their outputs into a unified representation. This multi-scale approach helps the model handle the complex, non-uniform blur patterns found in dynamic scenes.

Additionally, the block incorporates a channel attention module adapted from CBAM (Convolutional Block Attention Module). It combines average and maximum pooling features, which enhances the network's nonlinear representation.

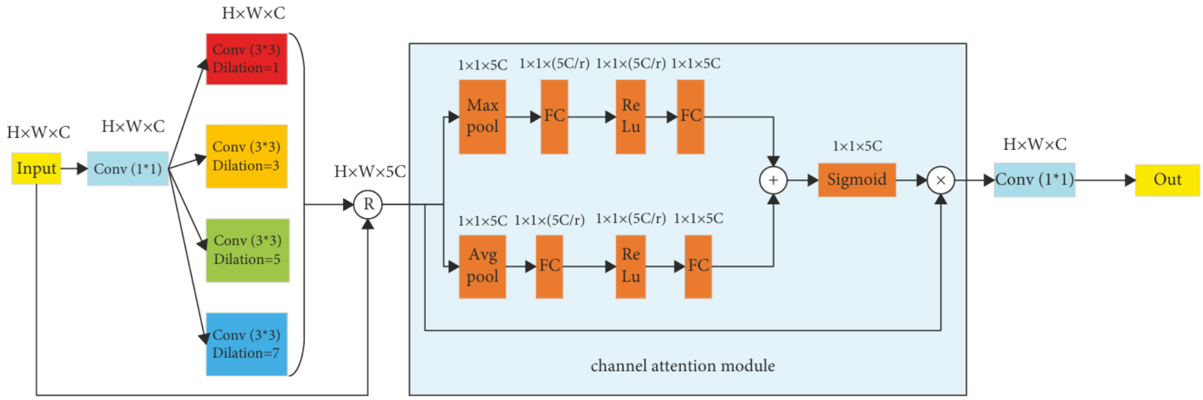


Figure 5 . Multireceptive channel block

Part IV

Result

1 Training Setup

To prevent overfitting, data augmentation is applied: the images were cropped to 256×256 pixels to manage computational load and prevent overfitting. Data augmentation techniques, such as random rotations (90°, 180°, 270°), horizontal and vertical flips.

The training process employed the Adam optimizer with a learning rate of 0.001, which gradually decreased until 1e-5. A batch size of 4 was used, and the model was trained for over 1000 epochs. The loss function is the combination of Mean Squared Error (MSE) and Structural Similarity Index (SSIM) loss, defined as:

$$L_{\text{total}} = L_{\text{MSE}} + w_1 L_{\text{SSIM}}, \tag{6}$$

where $L_{\text{MSE}} = \|R - S\|^2$, $L_{\text{SSIM}} = 1 - \text{SSIM}(R, S)$, R is the restored image, and $w_1 = 0.001$. This formulation aims to restore sharp images while preserving structural details and minimizing computational complexity.

2 Evaluation

The performance of the improved U-Net model was evaluated using quantitative metrics and visual inspection on the GoPro test set. Two key metrics, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), were employed to assess the quality of the deblurred images.

2.1 Peak Signal-to-Noise Ratio (PSNR)

PSNR [1] measures the ratio between the maximum possible power of a signal and the power of corrupting noise, expressed in decibels (dB). It is calculated as:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}_I^2}{\text{MSE}}\right), \tag{7}$$

where MAX_I is the maximum pixel value of the image (e.g., 255 for 8-bit images), and MSE is the mean squared error between the restored image and the ground-truth sharp image. A higher PSNR indicates better image quality, with values typically ranging from 20 to 40 dB for deblurring tasks. PSNR is sensitive to pixel-level differences, making it a good indicator of noise and distortion reduction.

2.2 Structural Similarity Index (SSIM)

SSIM [4] evaluates the similarity between two images based on luminance, contrast, and structural information, providing a value between -1 and 1 (with 1 indicating perfect similarity). It is computed as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{8}$$

where μ_x and μ_y are the mean intensities, σ_x^2 and σ_y^2 are the variances, σ_{xy} is the covariance of the two images, and c_1 and c_2 are small constants to stabilize the division. SSIM is more aligned with human perception, focusing on structural preservation rather than just pixel differences, making it complementary to PSNR.

2.3 Result

2.3.1 Quantiative result

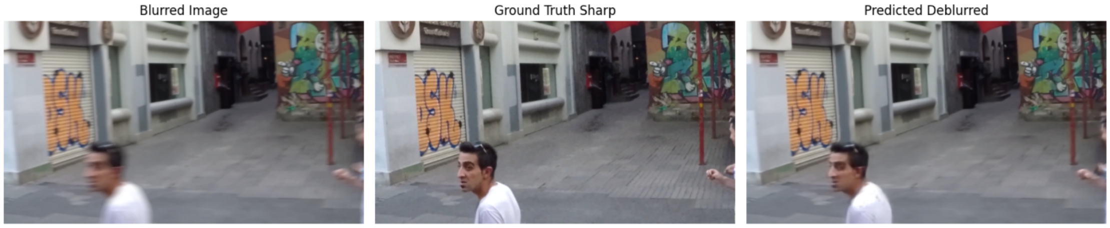
As shown in Table 1, our proposed model achieves a PSNR of 28.85 dB and an SSIM of 0.8688 on the test dataset. We compare these results against two prominent methods: Deblur-GAN V2 [2], a well-established GAN-based approach, and Restormer [5], the current state-of-the-art Transformer-based model. The results indicate that our model delivers a competitive performance, though it falls slightly below Deblur-GAN V2 (29.55 dB / 0.932 SSIM) and falls far below state-of-the-art model like Restormer. This demonstrates that our architecture is effective and operates within the performance range of strong GAN-based solutions.

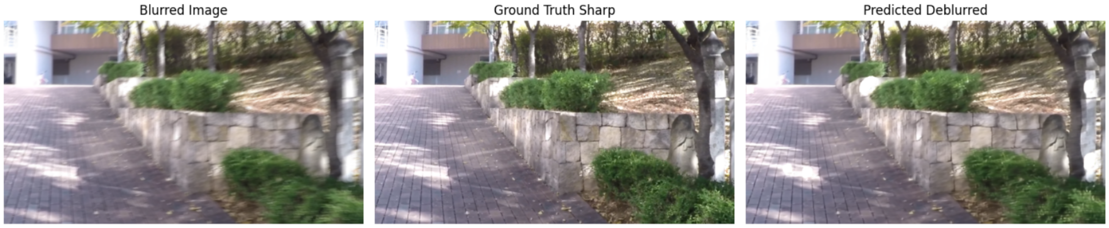
	Our model	Deblur-GAN V2	Restormer
PSNR ↑	28.85	29.55	32.92
SSIM ↑	0.8688	0.932	0.961

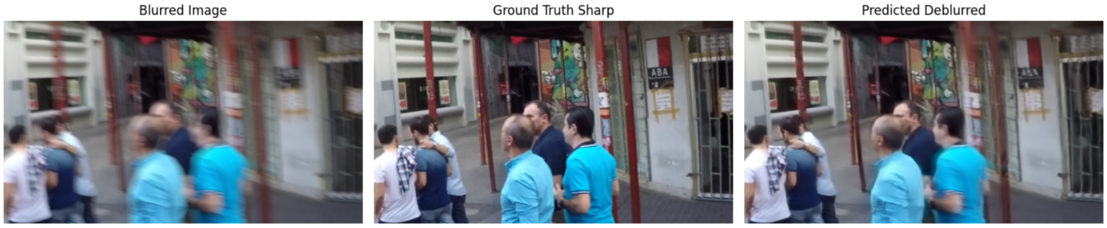
Table 1 . Comparison of Deblurring Models Performance.

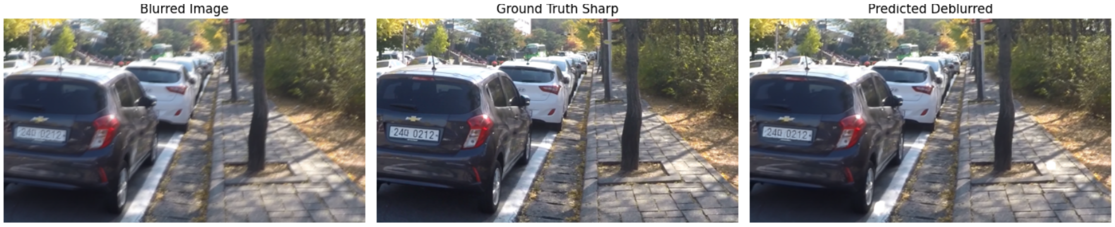
2.3.2 Visual result

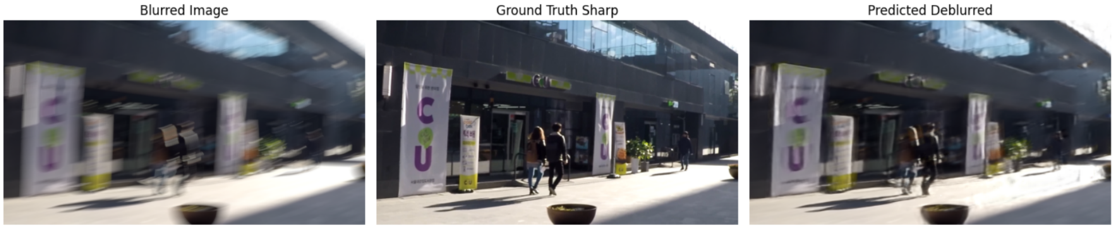
The visual results are fairly good and demonstrate the robustness of our deblurring model. It consistently handles a variety of challenging conditions, from fast-moving subjects and complex crowd scenes to subtle camera shake. However, the images generated still lack sharp details, especially on human faces where the model restores the general shape and color of the faces but produces a smoother texture compared to the ground truth. This trade-off is a well-known challenge in image restoration and is often attributed to the optimization objective. Our model, which is primarily guided by a pixel-wise loss (such as L1 or L2), learns to produce a perceptually safer, ”averaged” solution that minimizes pixel error but at the expense of high-frequency details.











Part V
Conclusion

1 Conclusion

This project has successfully implemented an improved Unet model for image deblurring task on Gopro Dataset. However, the results, with an average Peak Signal-to-Noise Ratio (PSNR) of 28.85 dB and Structural Similarity Index (SSIM) of 0.8688, indicate that the performance is still mediocre, not yet competitive with state-of-the-art deblurring models. This suggests that, despite the successful implementation, further refinements are needed to match the effectiveness of advanced techniques in the field.

2 Further work

This project provides a solid foundation for our research in image deblurring task. For future research, severals avenues should be explored:

- Refinements on the improved Unet model: further optimizing the hyperparameters, training for more epochs.
- Experiments with transformer-based models: leverage ViT for image deblurring task to achieve state of the art result on Gopro dataset.

The insights gained from this project provide a valuable roadmap for future research in the rapidly evolving field of image deblurring technology.

References

[1] Fernando A Fardo, Victor H Conforto, Francisco C de Oliveira, and Paulo S Rodrigues. A formal evaluation of psnr as quality measurement parameter for image segmentation algorithms. *arXiv preprint arXiv:1605.07116*, 2016.

[2] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8878–8887, 2019.

[3] Zuozheng Lian, Haizhen Wang, and Qianjun Zhang. An image deblurring method using improved u-net model. *Mobile Information Systems*, 2022(1):6394788, 2022.

[4] Jim Nilsson and Tomas Akenine-Möller. Understanding ssim. *arXiv preprint arXiv:2006.13846*, 2020.

[5] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022.