A CONVOLUTIONAL NEURAL NETWORK FOR MOTION-BASED MULTIFRAME SUPER-RESOLUTION USING FUSION OF INTERPOLATED FRAMES

Dissertation

Submitted to

The School of Engineering of the

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Doctor of Philosophy in Engineering

By

Hamed Elwarfalli

Dayton, Ohio

December 2023

A CONVOLUTIONAL NEURAL NETWORK FOR MOTION-BASED MULTIFRAME SUPER-RESOLUTION USING FUSION OF INTERPOLATED FRAMES

Name: Elwarfalli, Hamed

APPROVED BY:

Russell Hardie, Ph.D. Advisory Committee Chairman Professor, Electrical and Computer Engineering	Eric J. Balster, Ph.D. Committee Member Professor and Chair, Electrical and Computer Engineering
John Loomis, Ph.D. Committee Member Professor, Electrical and Computer Engineering	Youssel Raffoul, Ph.D. Committee Member Professor, Mathematics
Robert J. Wilkens, Ph.D., P.E. Associate Dean for Research and Innovation Professor School of Engineering	Margaret F. Pinnell, Ph.D. Interim Dean, School of Engineering

© Copyright by

Hamed Elwarfalli

All rights reserved

2023

ABSTRACT

A CONVOLUTIONAL NEURAL NETWORK FOR MOTION-BASED MULTIFRAME SUPER-RESOLUTION USING FUSION OF INTERPOLATED FRAMES

Name: Elwarfalli, Hamed
University of Dayton

Advisor: Dr. Russell Hardie

We present two novel multiframe image super-resolution (SR) algorithms that employ convolutional neural networks (CNNs) to generate high-resolution (HR) images from multiple low-resolution (LR) input frames. The first algorithm, named Fusion of Interpolated Frames Network (FIFNET), utilizes motion-based multiframe SR by fusing multiple input frames in a single CNN based on Random and Fixed shifts. The second algorithm, called the Exponential weighted Fusion of Interpolated Frames Network (EWF-FIFNET), presents two variations, Externally Exponential Weighted Fusion-FIFNET (EEWF-FIFNET) and Internally Exponential Weighted Fusion-FIFNET (IEWF-FIFNET) based on affine motion. A custom layer called the Exponential Weighted Fusion (EWF) layer is developed to combine input frames using a technique inspired by the fusion interpolation frame SR framework within the IEWF-FIFNET model. The EWF-FIFNET network utilizes a modified Residual Channel Attention Network architecture with residual in residual (RIR) structures.

The proposed algorithms are trained and tested using a realistic observation camera model that incorporates optical and sensor degradation. Affine motion is also incorporated to address a challenging degradation problem. The experimental results show that the proposed algorithms outperform the existing state-of-the-art methods using both simulated

and real camera data. It is noteworthy that the real data is not artificially downsampled or degraded, making the proposed algorithms a promising solution for practical applications. This research contributes significantly to the field of multiframe image SR, particularly in motion-based and exponentially weighted fusion approaches using CNNs.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my dissertation advisor, Dr. Russell Hardie, for their invaluable guidance, support, and patience throughout the research process. Their expertise and encouragement have been instrumental in shaping this dissertation.

I am also grateful to the members of my dissertation committee, for their insightful feedback and constructive criticism, which have greatly contributed to the quality of this work.

I would like to acknowledge the Libyan government and the University of Dayton for providing me with the resources, funding, and facilities necessary for conducting this research. I am especially thankful to the faculty of the Electrical Engineering Department for their support and encouragement.

I would like to extend my sincere thanks to my colleagues and friends, for their intellectual companionship, emotional support, and sense of humor. Their encouragement and camaraderie have made the journey of completing this dissertation much more enjoyable.

Finally, I would like to express my profound gratitude to my family, for their unwavering love, support, and patience throughout my academic journey. Their sacrifices and encouragement have been the foundation of my success.

Thank you all for your support and assistance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Single frame Super-resolution (SR) is another area receiving significant attention in the literature. With the absence of additional spatial samples, single frame methods rely more heavily on exploiting prior statistical information to address the ill-posed inverse problem of interpolating aliased imagery. Notwithstanding the significant challenge, several approaches have demonstrated performance far superior to simple interpolation. Recently, a popular approach to single frame SR has involved the use of convolutional neural networks (CNNs). Here prior information is learned by training the CNNs on large image databases of high-resolution (HR) images. Two examples used here as benchmarks are the Very Deep Super Resolution (VDSR) network [2] and Residual Channel Attention Networks (RCAN) [3]. Other examples of CNN-based single frame SR methods include [4–12].

While numerous CNN-based single frame SR methods may be found in the literature, we are aware of only one that addresses motion-based multiframe SR [13]. The method in [13] uses Residual Networks (ResNets) [14] to separately upsample multiple input images. The upsampled images are then fused and restored outside the CNN to produce the final SR image. The fusion is done with a shift-and-add method [15], and the restoration is done using an evolving imaging model [16]. The SR method in [17] proposes one CNN with inputs from multiple degradations but does not specifically address motion-based SR.

In this dissertation, we present two novel motion-based multiframe SR algorithms. Our methods use one CNN that fuses multiple interpolated input frames and performs restoration to produce an SR output. To the best of our knowledge, these are the first CNN approaches to perform motion-based multiframe SR by internally fusing multiple input frames with one network. One of the big challenges with such an approach is finding a

way to treat interframe motion that will allow a CNN to train consistently, and at the same time, exploit subpixel sampling diversity. We believe we have effectively addressed this by building on the framework of Fusion of Interpolated Frames (FIF) SR proposed by Karch and Hardie [1, 18]. The FIF SR method uses a deterministic subpixel algorithm to fuse multiple interpolated and aligned frames. It then applies a Wiener filter for image restoration.

First, we present a novel motion-based multiframe image super-resolution (SR) algorithm using a convolutional neural network (CNN) that fuses multiple interpolated input frames to produce an SR output. We refer to the proposed CNN and associated preprocessing as the Fusion of Interpolated Frames Network (FIFNET). We believe this is the first such CNN approach in the literature to perform motion-based multiframe SR by fusing multiple input frames in a single network. We study the FIFNET using translational interframe motion with both fixed and random frame shifts. The input to the network is a sequence of interpolated and aligned frames. One key innovation is that we compute subpixel interframe registration information for each interpolated pixel and feed this into the network as additional input channels.

Next, we propose our second multiframe SR algorithm, we transform the FIF SR method using a machine learning approach. We completely replace the fusion and restoration components with a custom-designed CNN architecture. We preprocess input frames by first upsampling and aligning them using interpolation. Using aligned frames as input channels effectively eliminates interframe motion and allows the CNN to train consistently, regardless of the motion. To more fully exploit the subpixel sampling diversity, we also compute subpixel interpolation distances for each pixel and feed these into the CNN as additional input channels. We demonstrate that these subpixel registration channels are critical to

the performance of the network. we extend our success algorithm the Fusion of Interpolated Frames Network (FIFNET), which has been proven to be a powerful technique that can overcome undersampling problems compared to the benchmarks [19]. We propose a CNN model boosts the performance of the Authors' model FIFNET even more, and it overcomes some challenges that have been faced by using the FIFNET model, and we establish the future work of FIFNET that we sign to achieve as we discuss in Chapter V. The architecture of our proposed CNN is an end-to-end SR algorithm, and it uses one CNN that fuses multiple interpolated input frames and performs restoration to produce an SR output. We design a custom exponentially weighted fusion (EWF) custom layer, and it is located at the top (as the first layer) of the proposed CNN model to fuse the input frames. Therefore, we borrow a residual in residual (RIR) structure to form a very deep network of one of the most powerful networks for SISR in the literature Residual Channel Attention Networks (RCAN) [3]. Then, we modify the RIR structure to be employed in multiframe CNN to achieve the deconvolution stage after the fusion process that is done via the custom layer. We refer to the proposed CNN as the Exponential Weighted Fusion of Interpolated Frames Network (EWF-FIFNET). The EWF-FIFNET model presents two architectures of the proposed model which are Externally Exponential Weighted Fusion-FIFNET (EEWF-FIFNET) and Internally Exponential Weighted Fusion-FIFNET (IEWF-FIFNET) based on affine motion. The proposed algorithms incorporate a custom layer called the Exponential Weighted Fusion (EWF) layer that combines input frames using a technique inspired by the fusion interpolation frame SR framework within the IEWF-FIFNET model. EWF-FIFNET network uses a modified Residual Channel Attention Network architecture with residual in residual (RIR) structures. One of the big challenges with such an approach is finding a way to treat interframe motion that will allow a CNN to train consistently, and at the same time, exploit subpixel sampling diversity. For that reason, we design a EWF custom layer

that helps our algorithm to fuse the frames using a deterministic subpixel algorithm to fuse multiple interpolated and aligned frames.

We also employ a camera-specific optical transfer function (OTF) that models diffraction and detector integration for generating training data [20]. We have not seen this kind of theoretical OTF model used previously in the literature with CNN-based SR. Our approach demonstrates the performance of the two proposed methods in a realistic scenario and makes the FIFNET and the EWF-FIFNET models ready to process real camera data. We present a number of experimental results to demonstrate the efficacy of both proposed FIFNET and EWF-FIFNET models on both simulated and real camera data. The real data are acquired by the authors and come directly from a camera and are not artificially downsampled or degraded. The real data include natural scenes and a chirp test pattern that clearly illustrates aliasing reduction from FIFNET and EWF-FIFNET models processing. In the quantitative results with simulated data, we show that EWF-FIFNET performs favorably in comparison to the benchmark methods tested. Based on the novelty and performance analysis results, we believe the EWF-FIFNET method makes an important contribution to the field of multiframe SR. The remainder of this dissertation details the observation model in Chapter II, followed by the introduction of the original fusion interpolation frame (FIF) SR algorithm in Chapter III. The proposed FIFNET and EWF-FIFNET CNN methods are then presented in Chapter IV. Chapter V outlines the experimental results, while Chapter VI concludes the paper.

CHAPTER II

OBSERVATION MODEL

In this chapter, we describe the observation model that was employed to relate an ideal high-resolution (HR) image to a set of low-resolution (LR) observed frames. The observation model was utilized to generate simulated images for both training and testing purposes. Unlike many previous approaches that rely on simple downsampling to generate LR frames, our approach employs a realistic observation model that takes into account the specific camera optics and the focal plane detector array. By doing so, we are able to generate more realistic and accurate LR frames that better approximate real-world imaging conditions. This, in turn, leads to improved performance of the algorithm in restoring degraded images.

The observation model is based on a mathematical framework that captures the physical processes that occur during the imaging process. Specifically, the model accounts for factors such as lens blur, sensor noise, and chromatic aberration, among others. By incorporating these factors into the model, we are able to generate LR frames that more closely resemble those obtained from real cameras. The simulated images generated using the observation model are used for training and testing the algorithm, allowing us to evaluate its performance under a range of imaging conditions. Through this approach, we are able to demonstrate the effectiveness of the algorithm in restoring degraded images in a realistic and practical setting.

Overall, the use of a realistic observation model represents a significant improvement over previous approaches and highlights the importance of accounting for the specific imaging conditions when generating LR frames. This approach allows for more accurate training and testing of the algorithm, leading to improved performance in restoring degraded images. In

this section, we describe the physical observation model that relates a static 2-D ideal scene image to a set of LR observed frames. In order to guarantee a solution to an SR problem via CNN, the degradation process that applies to the training/testing dataset must simulate the real degradation effect on the taken image by the camera. Unlike the previous approaches, where they have used a sample degradation process of downsampling followed by a Gaussian filter for blurring. We use a realistic camera model to simulate the degradation which occurs during taking an image via the camera.

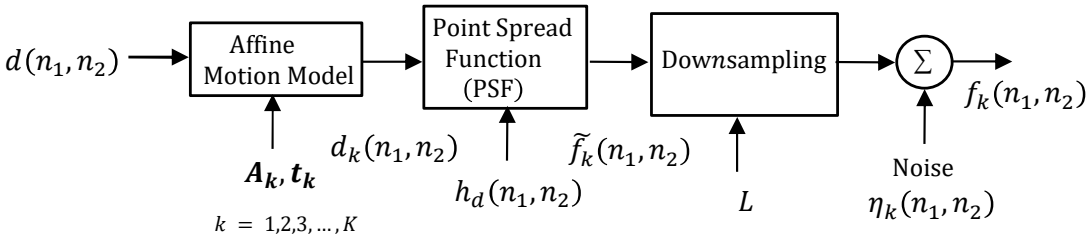


Figure 2.1: Warp-then-blur observation model that performs nonuniform sampling of a single blurred image. This version of the observation model is valid when PSF blurring and the affine motion model commute. The input of the observation model is a single HR image $d(n_1, n_2)$, and its output is a sequence of K LR images $f_k(n_1, n_2)$.

Figure 2.1 shows the block diagram of the warp-then-blur observation model, which is similar to the one presented in references [18, 19, 21, 22]. This model replicates the physical image acquisition process by starting with a pristine HR grayscale image, $d(n_1, n_2)$. We assume this represents a static scene, and using affine warping to model the relative motion between the camera and the scene. The output of the warping process is represented as $d_k(n_1, n_2)$. The next step involves blurring the output using a system Point Spread Function (PSF), $h_d(n_1, n_2)$, which is similar to the model used in Ref. [23]. The blurred image is then sampled below the Nyquist rate and corrupted with additive Gaussian noise,

resulting in a set of LR frames denoted by $f_k(n_1, n_2)$, where $k = 1, 2, \ldots, K$. The warp-then-blur observation model in Fig. 2.1 is often utilized as part of more complex iterative SR methods, such as those presented in references [23, 24], which require a significant amount of computational resources.

2.1 Affine Motion Model and Registration

Affine motion refers to a type of motion model commonly used in computer vision and image processing applications to describe the geometric transformations that occur when an object or scene is viewed from different angles or distances. An affine transformation is a linear mapping that preserves parallel lines, angles, and ratios of distances, but allows for translation, rotation, scaling, shearing, and reflection. Affine transformations are described by a matrix of coefficients, which can be used to transform the coordinates of a point or a set of points in an image. In the context of motion estimation and image registration, affine motion models are particularly useful because they can capture a wide range of motion patterns, including translations, rotations, scaling, and shearing. This makes them well-suited for applications such as video stabilization, object tracking, and 3D reconstruction. Additionally, the mathematical properties of affine transformations make them easy to compute and efficient to apply, making them a popular choice for many real-world applications.

The affine parameters describe the relationship between each frame and a reference frame and can be expressed using equations that depend on the type of motion being simulated. For a 2-D affine model, the affine parameters consist of six elements, which can be represented using a matrix notation. The motion model plays a crucial role in the SR algorithm, as it simulates the relative motion between a 2-D image and a 3-D scene. While SR approaches in the literature have extensively explored the 2-D translational motion

model [25, 26], real-life motion during image processing is often more complex than simple translation. Therefore, we employ a more intricate motion model to replicate the actual motion observed in real-world scenarios. An affine model is a promising option because it can incorporate rotation, shear, and zoom, as well as translation, using 6 parameters. The affine parameters for each frame in relation to a reference frame can be expressed using Eqs. 2.1 and 2.2. The affine matrix $\mathbf{A}_k$ is defined as:

$$\mathbf{A}_k = \begin{bmatrix} A_{1,1}(k) & A_{1,2}(k) \\ A_{2,1}(k) & A_{2,2}(k) \end{bmatrix} \tag{2.1}$$

and

$$\mathbf{t}_k = \begin{bmatrix} t_{n1}(k) & t_{n2}(k) \end{bmatrix}^T, \tag{2.2}$$

where $k = 1, 2, ..., K$, $\mathbf{t}_k$ contains the translational shift parameters, and $\mathbf{A}_k$ contains information on rotation θ, zoom, and shear. Using the model and notation $\mathbf{n} = [n_1, n_2]^T$ and $\tilde{\mathbf{n}}(k) = [\tilde{n}_1(k), \tilde{n}_2(k)]^T$, we have $\tilde{\mathbf{n}}(k) = \mathbf{A}_k\mathbf{n} + \mathbf{t}_k$, such that

$$d_k(\mathbf{n}) = d(\tilde{\mathbf{n}}(k)) = d(\mathbf{A}_k\mathbf{n} + \mathbf{t}_k). \tag{2.3}$$

The Eq. 2.3 can be used to estimate the affine parameters using a gradient-based least-squares algorithm based on the one described in [27, 28]. To estimate the affine parameters using a gradient-based least-squares algorithm, we start with an initial guess for the parameters and iteratively update them to minimize the sum of the squared differences between the observed data and the predicted data. The predicted values are calculated using the current set of parameters, and the error between the predicted and observed data is computed. The gradient of the error with respect to each parameter is then calculated, and each parameter is updated by subtracting its corresponding gradient multiplied by a step size parameter.

This process is repeated until the error is minimized to some predetermined threshold or a maximum number of iterations is reached. The final values of the affine parameters are the estimated values that minimize the sum of the squared differences between the observed data and the predicted data.

2.2 Point Spread Function Model

Blurring in the observation can have various causes. The most common factors are diffraction caused by optics and spatial integration due to detector elements. However, other sources like optical aberrations, defocus, and atmospheric turbulence can also contribute to blurring. In this context, we adopt the approach presented in [23] and consider modeling diffraction and detector integration. The optical transfer function (OTF) is used for this purpose.

$$H(u,v) = H_{\text{dif}}(u,v)H_{\text{det}}(u,v), \tag{2.4}$$

the variables u and v represent the number of horizontal and vertical cycles per unit of distance. The OTF for diffraction-limited optics is denoted by $H_{\text{dif}}(u,v)$, while the transfer function that models detector integration is denoted by $H_{\text{det}}(u,v)$. Assuming a circular pupil function, the following holds as per [29]:

$$H_{\text{dif}}(\rho) = \begin{cases} \frac{2}{\pi}\left[\cos^{-1}(\rho) - \rho\sqrt{1-\rho^2}\right] & \rho < 1 \\ 0 & \text{else} \end{cases}, \tag{2.5}$$

where $\rho = \sqrt{u^2+v^2}/\rho_c$, the optical cutoff frequency is $\rho_c = 1/(\lambda F)$. It is important to note that $H_{\text{det}}(u,v)$ is the Fourier Transform of the shape of the detector element, as explained

in [20]. The wavelength of light is denoted by λ, and the f-number of the optics is represented by F. The f-number is defined as the ratio of the focal length of the optics to the effective aperture [30]. The f-number F controls the cut-off frequency of the overall OTF, while ρ_c determines the spatial sampling frequency. To avoid aliasing, the sampling process must satisfy the Nyquist sampling theorem condition, which states that $1/p > 2\rho_c = 2/(\lambda F)$. The undersampling factor is defined as $M = 2p/(\lambda F)$ to characterize this condition. The primary motivation for multiframe SR methods that can help overcome aliasing degradation is due to this undersampling issue [31]. Finally, the inverse continuous-space Fourier transform of the OTF in Eq. (2.4) yields the continuous-space system PSF, as indicated by the following equation. (2.6) represents.

$$h(x, y) = \text{ICSFT}\{H(u, v)\}. \tag{2.6}$$

An impulse-invariant equivalent discrete-space PSF, $h_d(n_1, n_2)$, may be found by sampling $h(x, y)$ at the HR sampling rate. This impulse invariant PSF is applied in Fig. 2.1 to produce multiframe blurred image $\tilde{f}_k(n_1, n_2)$ as Eq. (2.7) express.

$$\tilde{f}_k(n_1, n_2) = d_k(n_1, n_2) * h_d(n_1, n_2), \tag{2.7}$$

where $*$ is 2D discrete convolution. The specific camera parameters used in the experimental results in this paper, both simulated and real, are listed in Table 2.1. The camera is an Imaging Source DMK21BU04 visible USB camera with Sony ICX098BL CCD sensor. The camera is equipped with a 5 mm focal length lens set with an f-number of $F = 5.6$. The native undersampling is $M = 3.636$. However, we have obtained good results using an upsampling factor of $L = 3.000$. Thus we define the HR grid with a sample spacing of p/L and downsample the image by $L = 3$ in the observation model. Fig. 2.2 shows a plot of the

overall OTF. Note that the $|H(u,v)|$ OTF curve (shown in red) extends well beyond the native folding frequency (i.e., 1/2 the sampling frequency). This shows that a significant amount of signal frequency content may be aliasing with this sensor. However, minimal signal energy will be passed by this OTF above the $L = 3$ SR folding frequency.

Table 2.1: Optical parameters used for training and testing for real and simulated images.

Parameter	Value
Aperture	$D = 0.893$ mm
Focal length	$l = 5.00$ mm
F-number	$F = 5.60$
Wavelength	$\lambda = 0.550$ μm
Optical cut-off frequency	$\rho_c = 324.68$ cyc/mm
Detector Pitch	$p = 5.6$ μm
Sampling frequency	$1/p = 178.57$ cyc/mm
Undersampling	$M = 3.636$
Upsampling factor	$L = 3.000$

afterward, the frames that have been shifted and blurred, $\tilde{f}_k(n_1, n_2)$, are subjected to downsampling, where the resolution is reduced by a factor of L in both spatial dimensions. Following that, we introduce Gaussian noise that is independently and identically distributed across all frames. Therefore, the resulting frames that are observed can be described as:

$$f_k(n_1, n_2) = \tilde{f}_k(Ln_1, Ln_2) + \eta_k(n_1, n_2), \tag{2.8}$$

we apply this observation process to each individual high-resolution (HR) image in the dataset, resulting in a sequence of K blurred, shifted, and downsampled images denoted as $f_k(n_1, n_2)$. Generating LR images using our real observation model involves simulating the process of how an image is captured or observed as an example as Fig. 2.3 . The observation model accounts for the physical properties of the imaging system, such as the resolution and

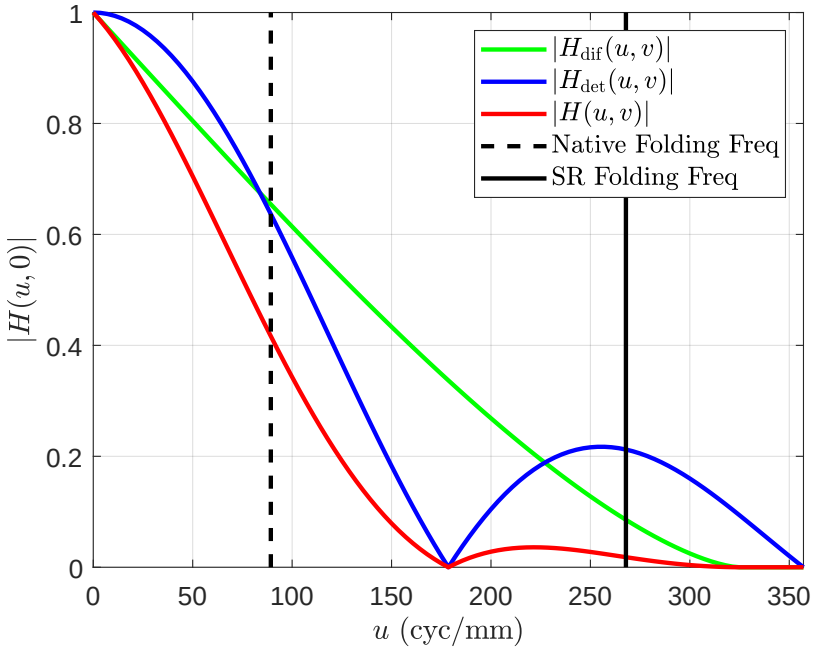


Figure 2.2: Optical transfer function from Eq. (2.4) for the camera model using the parameters from Table 2.1. The SR folding frequency is shown for $L = 3$.

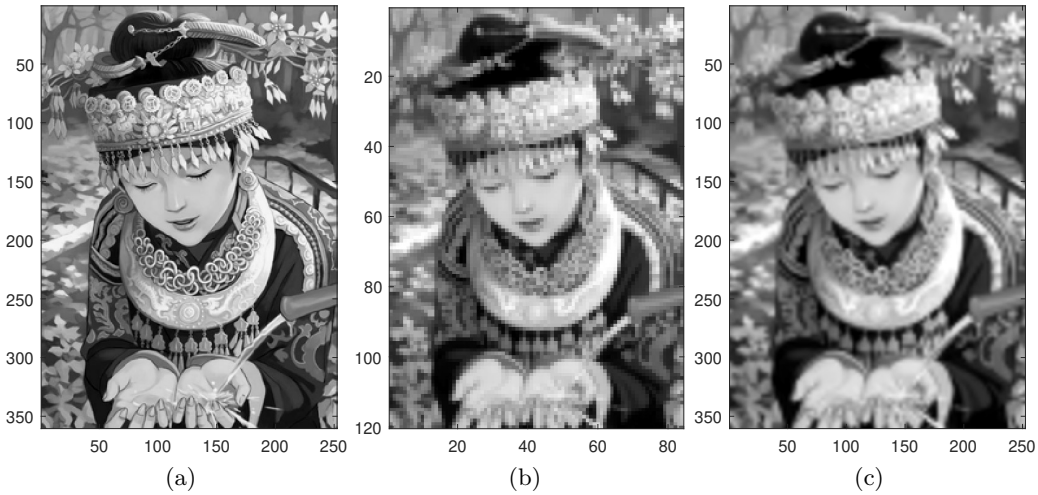


Figure 2.3: The ground truth image and the low-resolution image obtained from the observation model are visualized, with the low-resolution image being interpolated using bicubic interpolation. Specifically, the following three images are shown: (a) ground truth, (b) low-resolution image, and (c) bicubic interpolation.

noise level, to generate a low-resolution version of the original high-resolution image. This technique is commonly used in image processing and computer vision applications, where it is often necessary to work with low-resolution images due to computational constraints or limitations of the imaging hardware. By generating a low-resolution image from an observation model, researchers can apply image processing techniques to enhance or restore the image quality, improving the accuracy and reliability of subsequent analysis.

CHAPTER III

MULTIFRAME SUPER-RESOLUTION

3.1 Multiframe Super-Resolution

Multiframe SR is an image processing technique that aims to enhance the resolution and quality of LR images by utilizing multiple LR frames. By exploiting temporal or spatial redundancy between the frames, a SR image can be reconstructed. This technique has diverse applications such as digital restoration, medical imaging, and surveillance systems. In this chapter, we introduce the original fusion interpolation frame (FIF) method [23], which is a multifram SR approach that has proven to be effective and straightforward. Additionally, we present the preprocessing steps for the dataset used in this method.

3.1.1 Preprocess Dataset

Before applying the muliframe method, the dataset must undergo two primary preprocessing steps. Firstly, the input LR frames denoted as $f_k(n_1, n_2)$ must be individually interpolated and aligned to a common $L\times$ upsampled HR grid. To achieve this, subpixel image registration [32] is utilized to estimate the shifts required for alignment purposes, using the first frame as the reference. The resulting interpolated frames are denoted as $g_k(n_1, n_2)$, where $k = 1, 2, ..., K$. To ensure a good balance of speed and performance, bicubic interpolation is employed for the interpolation process, as it has been shown to be effective in fusion applications [18]. Overall, these preprocessing steps are critical for optimizing the performance of the muliframe method by ensuring that the input frames are accurately aligned and interpolated to a consistent resolution. In the dashed red box in Fig. 3.1, you can see the preprocessing steps.

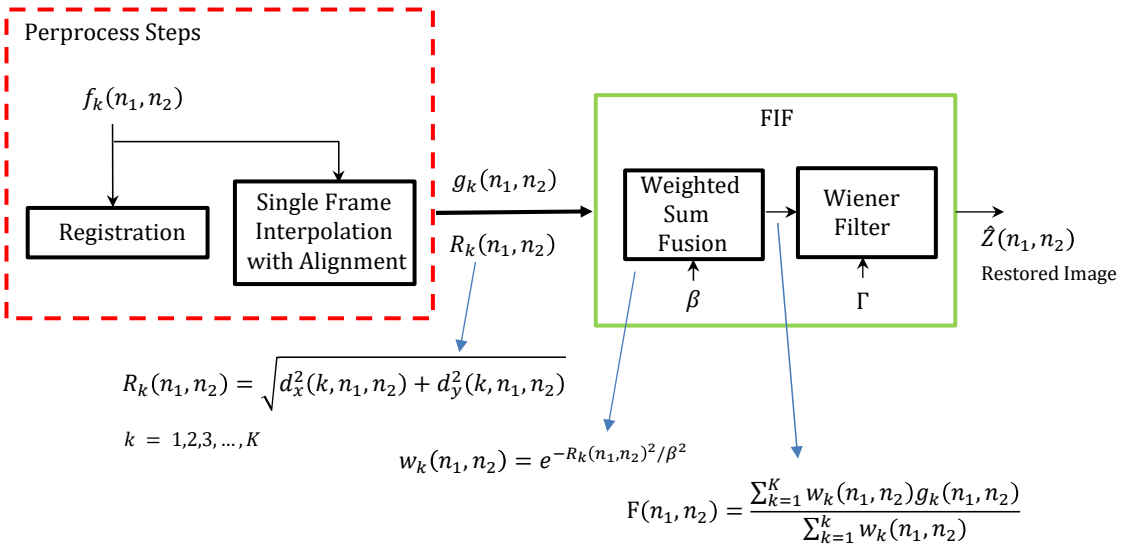


Figure 3.1: The block diagram illustrates the method for multiframe SR including FIF. The original FIF SR method [1]. The interpolated and aligned observed frames are combined with the subpixel registration information to form the input of the FIF SR algorithm.

3.1.2 FIF Super-Resolution

The FIF SR algorithm has proven to be an effective and intuitively simple multiframe method [1,18]. Furthermore, it can be readily adapted to complex scene motion and division of focal plane imaging sensors [1]. We begin by describing this algorithm because our FIFNET approach builds on this framework. The block diagram in Fig. 3.1 shows the FIF SR method. The interpolated frames are then fused on a pixel-wise basis using a weighted sum operation. This is given by

$$F(n_1, n_2) = \frac{\sum_{k=1}^{K} w_k(n_1, n_2) g_k(n_1, n_2)}{\sum_{k=1}^{k} w_k(n_1, n_2)}. \tag{3.1}$$

The weights, $w_k(n_1, n_2)$, are a function of the distance of each interpolated pixel to the nearest original LR pixel. This is expressed as

$$w_k(n_1, n_2) = e^{-R_k(n_1,n_2)^2/\beta^2}, \tag{3.2}$$

where

$$R_k(n_1, n_2) = \sqrt{d_x^2(k, n_1, n_2) + d_y^2(k, n_1, n_2)} \tag{3.3}$$

and $d_x(k, n_1, n_2)$ and $d_y(k, n_1, n_2)$ are the horizontal and vertical distances of interpolated pixel $g_k(n_1, n_2)$ to the nearest LR pixel in the k'th frame. Fig. 3.2 provides a closer view of the FIF process, which takes as input the concatenation of interpolated frames that have been subpixel-shifted and outputs a single SR image. The distance computation defined by Eq. (3.3) is illustrated in Fig. 3.4. The basic idea is that an interpolated pixel that is near an original LR pixel will have less interpolation error and should be given a higher weight. If diverse camera motion is present, each HR interpolated pixel is likely to have an LR pixel nearby in one or more of the input frames. The parameter β controls how sensitive the weights are to the distances in $R_k(n_1, n_2)$. We use $\beta = 0.1$ for all of the results presented in this paper [18]. Note that the interframe registration information in Eq. (3.3) has the same dimensions as the interpolated frames and may be viewed as images. This is illustrated in Fig. 3.3 for $K = 4$ frames with random interframe shifts and $L = 3$.

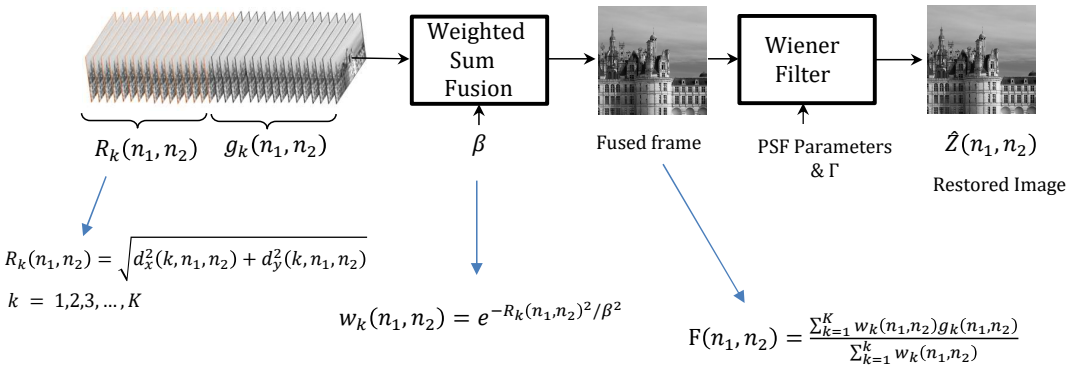


Figure 3.2: In the block diagram, the multiframe SR FIF is presented, where the input of the FIF SR algorithm is composed of the aligned observed frames that are interpolated, and the subpixel registration data.

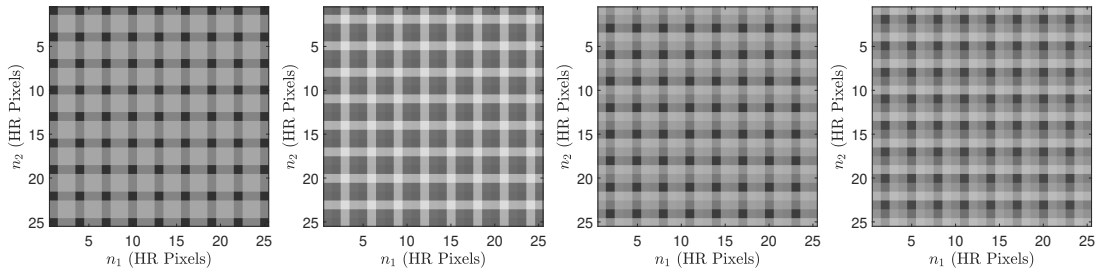


Figure 3.3: Visualization of $R_k(n_1, n_2)$ for a 25×25 patch size with different shifts and $L = 3$. HR pixel shifts are (a) $\mathbf{s}_1 = [0.00, 0.00]^T$, (b) $\mathbf{s}_2 = [0.56, 02.54]^T$, (c) $\mathbf{s}_3 = [-2.03, 4.86]^T$, and (d) $\mathbf{s}_4 = [-2.09, 1.25]^T$.

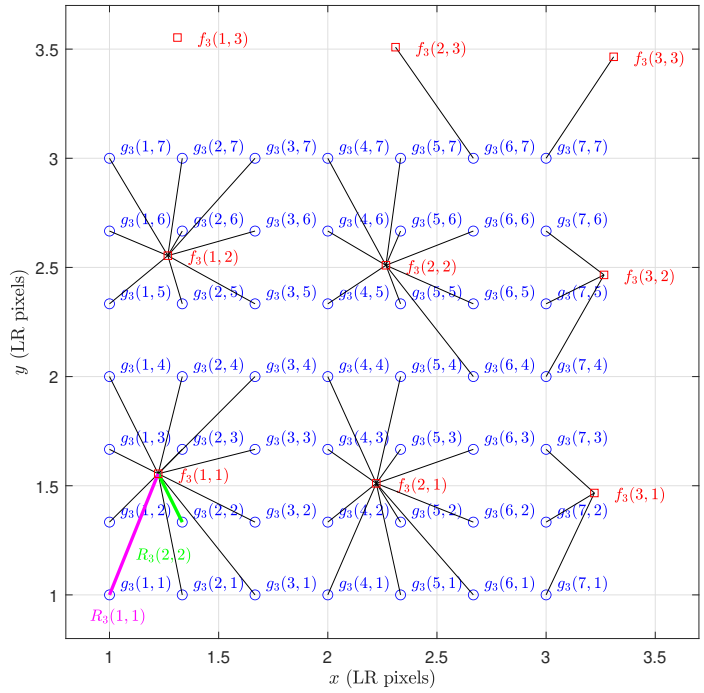
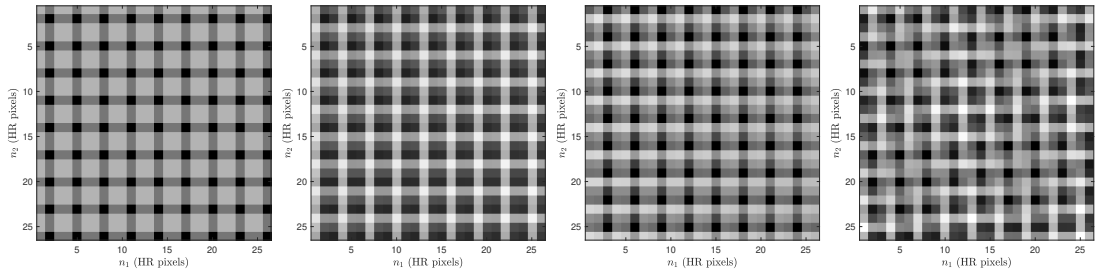


Figure 3.4: Spatial sampling grid shown in LR pixel spacings. The pixel positions of interpolated frame $g_3(n_1, n_2)$ are shown as blue circles for $L = 3$. The corresponding LR frame samples, $f_3(n_1, n_2)$, are shown as red squares for a shift and rotation of $\mathbf{s}_3 = [0.18, 0.6]^T$, and $\theta = -2.527$ LR pixels. The subpixel distances, $R_3(n_1, n_2)$, are shown as black lines. An example of a large distance value is shown in green, and a small one is shown in magenta.



(a) (b) (c) (d)

Figure 3.5: Visualization of $R_k(n_1, n_2)$ for a 25×25 patch size with different shifts and $L = 3$. HR pixel shifts and rotation are (a) $\mathbf{s}_1 = [0.00, 0.00]^T$, $\theta_1 = 0$, (b) $\mathbf{s}_2 = [0.52, -0.40]^T$, $\theta_2 = -0.37$, (c) $\mathbf{s}_3 = [0.93, -1.08]^T$, $\theta_3 = -0.48$, and (d) $\mathbf{s}_4 = [0.83, 1.28]^T$, $\theta_4 = -4.46$.

Note that the darker pixels correspond to interpolated values with smaller interpolation distances, and presumably more accurate values. Figs. 3.3 (a), and 3.5 (a) corresponds to the interpolated reference frame with no shift. Thus, every L'th pixel starting from pixel $n_1 = n_2 = 1$ lines up exactly with an input pixel and has an interpolation distance of 0. Fig. 3.5 depicts the case of $K = 4$ frames with affine motion and $L = 3$, demonstrating the illustration of this scenario.

The final step of the FIF SR method is deconvolution of the PSF blur from the fused image. In the original method, this is accomplished with a Wiener filter [1,18]. The Wiener filter uses the parameter Γ as the constant noise-to-signal power spectral density ratio. In all of our results in this research , we search for and use the optimum Γ when reporting FIF SR quantitative performance results.

CHAPTER IV

DEEP LEARNING-BASED MULTIFRAME SUPER-RESOLUTION

In this chapter, we present our machine learning approaches, which are FIFNET based on Random and Fixed shifts and Externally Exponential Weighted Fusion-FIFNET (EEWF-FIFNET), and Internally Exponential Weighted Fusion-FIFNET (IEWF-FIFNET) based on affine motion. Our aim is to significantly improve the original FIF SR method. We have achieved this by completely replacing the deterministic fusion and separate Wiener filter restoration with a single custom-designed CNN. The use of a CNN offers several benefits. Firstly, the non-linear nature of CNNs has proven to be effective in handling aliasing in SISR, and we believe it offers similar advantages in the multiframe case. Additionally, the large number of degrees of freedom in a CNN has the potential to outperform a Wiener filter. Lastly, our proposed CNN models achieve joint image fusion and restoration.

4.1 FIFNET Model

In this section we describe the proposed FIFNET SR method. To the best of our knowledge, there are no published algorithms that perform motion-based multiframe SR where the image fusion takes place within a single CNN. One possible reason for this may be the challenge of how to deal with the interframe motion in a manner that both exploits the motion and allows a CNN to train consistently. It is our hypothesis that the proposed FIFNET method can effectively address this challenge by employing interpolation and registration preprocessing steps similar to those in the original FIF SR method. By providing upsampled and aligned frames as multichannel inputs to a CNN, the interframe motion is effectively removed. This provides a consistent input for the CNN to train on. To more fully exploit the subpixel sampling diversity, we propose providing the subpixel distance

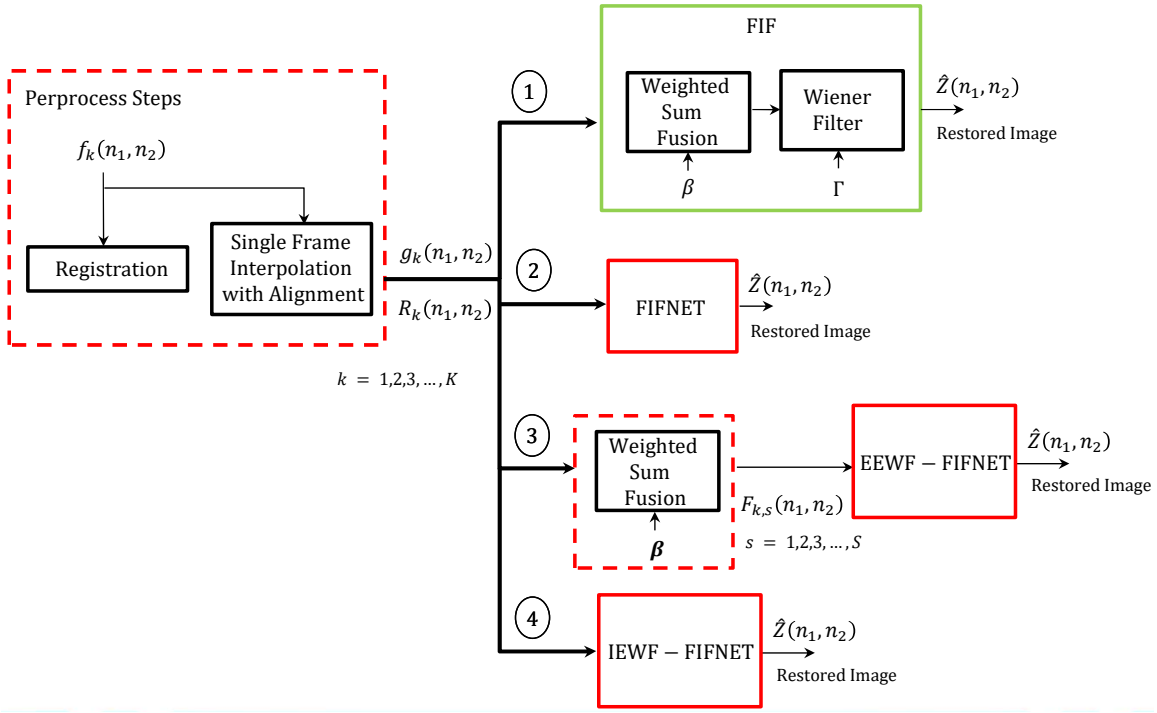


Figure 4.1: The block diagram illustrates four different methods for multiframe SR including FIF, FIFNET, EEWF-FIFNET, and IEWF-FIFNET. The original FIF SR method [1] is represented in the top branch labeled as (1). Each of the proposed models, namely FIFNET, EEWF-FIFNET, and IEWF-FIFNET, replaces the deterministic fusion and restoration blocks with a novel machine learning approach utilizing a specifically designed CNN architecture, which is shown in the bottom branch labeled as (2), (3), and (4) respectively.

arrays defined in Eq. (3.3) as additional input channels. This results in the processing steps depicted in Fig. 4.1 following Path (2).

Note that the joint $L\times$ upsampling and subpixel alignment preprocessing is important. Alignment with upsampling preserves the fidelity of those interpolated pixels near the original samples. If alignment is performed at the native LR sensor resolution, the fidelity of all of the interpolated pixels is lost. Furthermore, with the proposed approach we are able to track the relative fidelity of all of the interpolated pixels using Eq. (3.3). Our experimental

results show that a significant performance gain is achieved as a result of including this critical information.

With the FIFNET machine learning approach, we believe we have significantly transformed the original FIF SR method. In particular, we have completely replaced the deterministic fusion and separate Wiener filter restoration with a single custom-designed CNN. We believe there are several benefits to using a CNN here. First, the non-linear nature of CNNs has proven to be helpful in dealing with aliasing in single image SR. We believe it offers similar benefits in the multiframe case. Also, the large number of degrees of freedom in a CNN offers the potential to be more potent than a Wiener filter in dealing with the non-Gaussian and non-stationary nature of most images. Finally, because the image fusion and restoration are being accomplished jointly in the FIFNET, we can expect some synergy between these two functions.

4.1.1 Preprocess Dataset

The FIFNET algorithm follows the same preprocessing steps as the FIF SR algorithm detailed in chapter III. The dataset is subjected to two primary preprocessing steps. The first step involves individually interpolating and aligning the input LR frames denoted as $f_k(n_1, n_2)$ to a common $L\times$ upsampled HR grid using subpixel image registration [32]. The first frame serves as the reference for estimating the required shifts for alignment purposes. The resulting interpolated frames are denoted as $g_k(n_1, n_2)$, where $k = 1, 2, ..., K$. Bicubic interpolation is employed for the interpolation process to achieve a balance between speed and performance, as it has been proven effective in fusion applications [18]. These preprocessing steps are crucial for optimizing the performance of the muliframe method by

ensuring accurate alignment and interpolation of the input frames to a consistent resolution. The dashed red box in Fig. 4.1 illustrates the preprocessing steps.

4.1.2 FIFNET Architecture

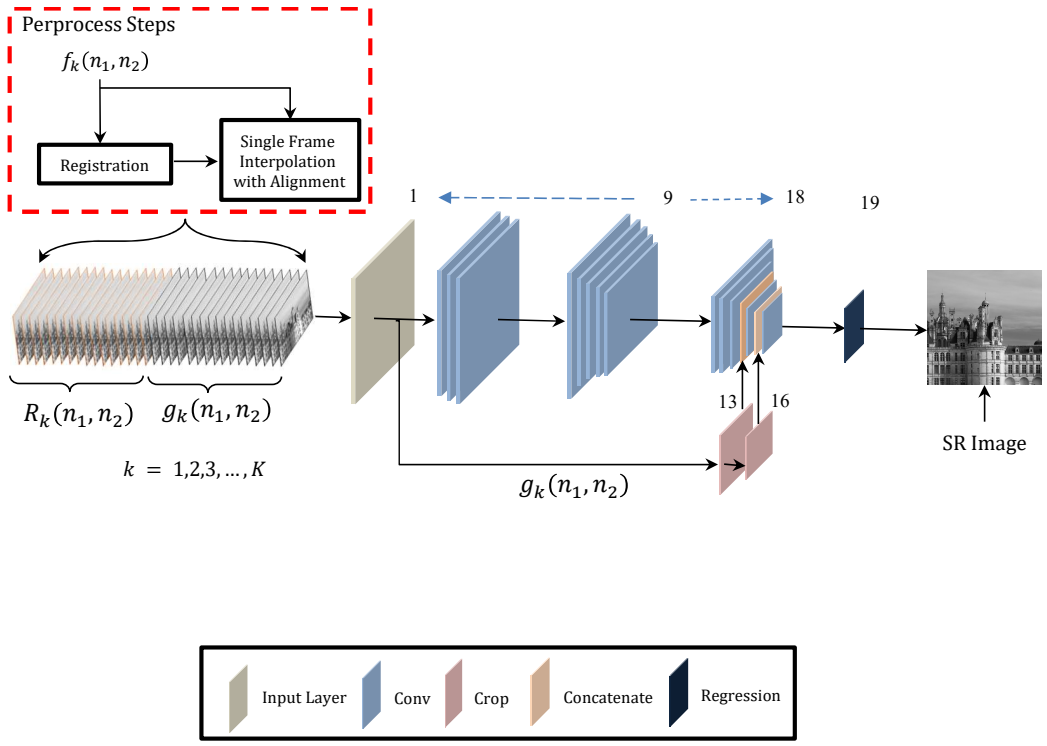


Figure 4.2: FIFNET architecture. The interpolated and aligned observed frames are combined with the subpixel registration information to form the input channels. A description of the individual layers is provided in Table 4.1. The output of the FIFNET is a single estimated SR image.

The FIFNET CNN architecture is shown in Fig. 4.2 and each of the layers is defined in Table 4.1. The network is implemented here in MATLAB using the Deep Learning Toolbox™. The input is composed of the upsampled and aligned interpolated frames, $g_k(n_1, n_2)$, and the corresponding subpixel registration arrays, $R_k(n_1, n_2)$, for $k = 1, 2, ...K$. The output of the FIFNET network is one SR image estimate.

Table 4.1: Description of the FIFNET layers. Note that the spatial dimensions of the activations are for training using 61×61 image patches. The trained network may be applied to any size testing image.

Layer	Type	Spatial kernel	Activations
1	Input	-	$61 \times 61 \times 2K$
2	2D Conv	1×1	$61 \times 61 \times 64$
3	2D Conv	1×1	$61 \times 61 \times 64$
4	2D Conv	1×1	$61 \times 61 \times 64$
5	2D Conv	3×3	$59 \times 59 \times 64$
6	2D Conv	3×3	$57 \times 57 \times 64$
7	2D Conv	3×3	$55 \times 55 \times 64$
8	2D Conv	3×3	$53 \times 53 \times 64$
9	2D Conv	3×3	$51 \times 51 \times 64$
10	2D Conv	5×5	$47 \times 47 \times 64$
11	2D Conv	5×5	$43 \times 43 \times 64$
12	2D Conv	5×5	$39 \times 39 \times 64$
13	2D Crop	-	$39 \times 39 \times K$
14	Concatenate	-	$39 \times 39 \times (64+K)$
15	2D Conv	5×5	$35 \times 35 \times (64+K)$
16	2D Crop	-	$35 \times 35 \times K$
17	Concatenate	-	$35 \times 35 \times (64+2K)$
18	2D Conv	5×5	$31 \times 31 \times 1$
19	Regression	-	-

The network is trained with image patches of size 61×61. Given K frames, and combining these with the corresponding registration arrays, the input to the network is an array of size $61 \times 61 \times 2K$. We do not employ any internal padding with the convolutions in the network. Thus, the spatial dimensions of the activations get smaller with each of the spatial convolution layers, as shown in Table 4.1. This is also illustrated in Fig. 4.2 with the reduced size layer blocks. Finally, the output in Layer 18 is a $31 \times 31 \times 1$ image patch estimate. This output is compared with the corresponding target patch for regression error computation in what we list as Layer 19 in Table 4.1.

We implement the FIFNET using a total of 13 convolution layers. The convolution layers have been divided into three groups, with a different spatial kernel size for each

group. The first group contains Layers 3-5 and performs 2D convolution using a 1×1 spatial kernel that spans all channels. We employ the 1×1 spatial kernel size for this first group so as to exclusively produce channel fusion of $g_k(n_1, n_2)$ and $R_k(n_1, n_2)$, rather than spatial processing. This "fusion-first" approach is in keeping with the original FIF SR framework and keeps the CNN network complexity low. Layers 6-10 make up the next group that begins the spatial processing using a 3×3 kernel size. The third and final group of convolution layers uses a 5×5 kernel size. With the exception of the final convolution layer (i.e., Layer 19), all convolution layers employ 64 filters to produce 64 output channels or feature maps and employ a rectified linear unit (ReLU). The final convolutional layer employs only one filter and has no ReLU.

Note in Fig. 4.2 that we concatenate the original interpolated frames (without the registration channels) to the processed feature maps in Layers 15 and 18. To accommodate this concatenation, Layers 14 and 17 crop the input to match the sizes of the reduced activation dimensions in Layers 15 and 18. The interpolated frames tend to provide the low spatial frequency content for the final estimate. This allows the main spatial processing pipeline to have the arguably simpler task of generating high spatial frequency "corrections", rather than the full SR image. The final layers fuse the correction information with the interpolated frames to produce the final output. We have found that inclusion of the registration channels at the input, and concatenation of the interpolated frames late in the network, are both crucial to network performance.

While the network training uses patches, the trained network can be applied to any size test image. The convolutions in each layer are simply extended across an arbitrary size test image. During testing we pre-pad the borders of the interpolated input images (and registration arrays) by 15 pixels on all sides. This allows the FIFNET output size

to match the interpolated input size. We use symmetric reflection padding to minimize discontinuities. The main benefit of the no-padding CNN architecture is that it reduces the network size to allow for faster training. In some cases this also provides improved performance. The improved performance may be the result of avoiding data extrapolation error during training. By selecting patches away from the border, no extrapolated data is ever used in the network during training. In contrast, many other methods employ zero padding within the network. This effectively introduces artificial discontinuities to every patch in the training data.

4.1.3 FIFNET Training

The objective function for the FIFNET regression network is the mean squared error between the true HR image d and the SR estimate $\hat{z}(n_1, n_2)$. As mentioned above, this error is computed over a 31×31 patch. All network training and hyper-parameter optimization has been done using the publicly available DIV2K HR training dataset [33, 34]. This database consists of 800 24-bit RGB images with 2K resolution. All of the input images have been converted to grayscale with a floating point dynamic range of $0 - 1$. The Gaussian noise in the observation model is set to $\sigma_\eta = 0.001$ for the simulated training and testing data. Training is done using stochastic gradient descent with momentum optimization. Sixty four different random patches are extracted from each training image. We use a mini-batch size of 64 so that each iteration corresponds to one training image. All training is done with a total of 100 epochs. We set the initial learning rate to 0.1 and reduced this by a factor of 0.1 after every 10 epochs.

Network training is achieved here using a Windows workstation with Intel Core i9-9980XE Processor using 3.0 GHz clock speed. It is equipped two NVIDIA TITAN RTX

Graphics Processing Units (GPU). Only one GPU is used for training a single network. Training one FIFNET for $K = 10$ takes 5.56 hours using the parameters in Table 4.1. If the network is reconfigured to use zero padding with each convolution layer, and retain the 61×61 input size, this larger network takes 6.78 hours to train (an increase of 21.95%). For $K = 1$, training the standard FIFNET takes 2.36 hours. Using padding, it takes 3.81 hours (an increase of 61.44%). In addition to the significant speedup in training, the no-padding architecture gives a modest boost in average PSNR of approximately 0.2 dB in testing for $K = 10$.

4.2 EWF-FIFNET Model

In this section, we provide a comprehensive explanation of the Exponential Weighted Fusion-FIFNET (EWF-FIFNET) model, which consists of two distinct architectures. The first architecture, shown in Fig. 4.1 Path (3), is the Externally Exponential Weighted Fusion-FIFNET (EEWF-FIFNET) model. The frames are fused externally as a preprocessing step, and the resulting fused frames serve as input for the EEWF-FIFNET model. The second architecture, which we propose to improve the EWF-FIFNET model, is the Internally Exponential Weighted Fusion-FIFNET (IEWF-FIFNET) model, as depicted in Fig. 4.1 Path (4). The IEWF-FIFNET model achieves fusion and restoration of the SR image in a single CNN model. To achieve internal fusion within the IEWF-FIFNET model, we incorporate a custom layer called the Exponential Weighted Fusion (EWF) layer. The EWF layer is a critical component of our approach as it enables the combination of features extracted by multiple CNN models. We assign different weights to each model based on its performance on the training data, allowing the model to rely more heavily on better-performing models. This approach has proven highly effective in achieving superior performance on

benchmark datasets. Both architectures feature a deep SR model residual in residual (RIR) blocks from the RCAN model [3].

4.2.1 Preprocess Dataset

The EWF-FIFNET model comprises two different architectures, which will be covered in the following section. The EEWF-FIFNET and IEWF-FIFNET models are shown in path (3) and (4), respectively, in Fig. 4.1. In the EEWF-FIFNET model, an additional preprocess step called weighted sum fusion is introduced, which produces $F_{k,s}(n_1, n_2)$ as the input to the EEWF-FIFNET model. $F_{k,s}(n_1, n_2)$ is created by fusing the interpolated frames $g_k(n_1, n_2)$ using subpixel interframe registration information $R_k(n_1, n_2)$, resulting in fused frames $F_{k,s}(n_1, n_2)$ with the same dimensions as the ground truth image. The length of the $\boldsymbol{\beta}$ vector, denoted by S, determines the weight sensitivity to the distances in $g_k(n_1, n_2)$, where $k = 1, 2, ..., K$, as explained in Section 3.1.2. In contrast, the IEWF-FIFNET model fuses the frames internally and follows the same two-step process as the FIF algorithm and the FIFNET model. These preprocessing steps are critical for optimizing the performance of the muliframe method by ensuring accurate alignment and interpolation of the input frames to a consistent resolution. The dashed red box in Fig. 4.1 depicts the preprocessing steps.

4.2.2 EEWF-FIFNET Model Architecture

By utilizing the FIFNET model, we have successfully made significant improvements to the original FIF SR method. Our approach consisted of replacing the deterministic fusion and separate Wiener filter restoration with a custom-designed CNN. The promising results produced by the FIFNET model have motivated us to push the boundaries and

enhance FIFNET further. As a result, we have developed the EEWF-FIFNET model, which extends the successful FIFNET SR model, as demonstrated in Fig. 4.3. To achieve this, we have adapted the RCAN model [3], a state-of-the-art CNN architecture designed for SISR problems, and utilized a deep network structure solely comprised of residual in residual (RIR) blocks to effectively restore and deconvolve imagery, resulting in a high-quality SR image estimate. Our implementation of this network was carried out using the Python framework.

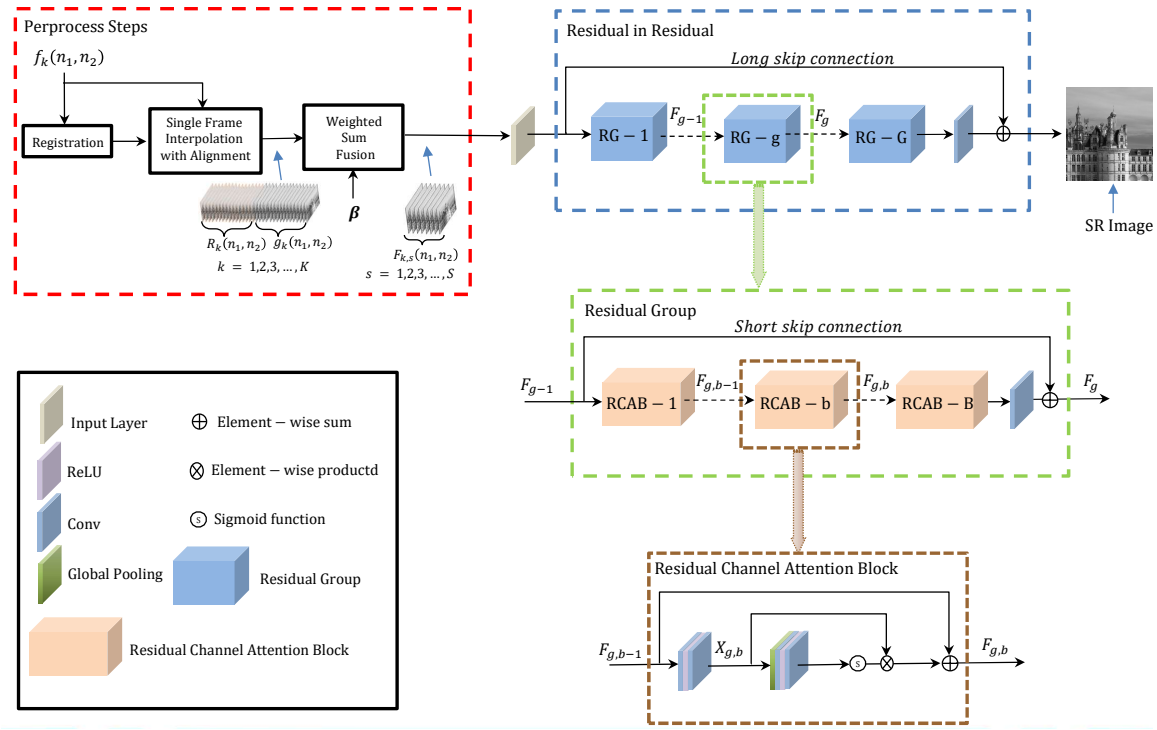


Figure 4.3: The architecture of the EEWF-FIFNET model is depicted in the figure, where the preprocessing steps are enclosed in a red dashed box. These steps involve fusing the interpolated and aligned observed frames with subpixel registration information to produce the input channels of the EEWF-FIFNET model, denoted as $F_{k,s}(n_1, n_2)$. The EEWF-FIFNET model utilized a deep network structure solely comprised of residual in residual (RIR) blocks.

The EEWF-FIFNET model network begins with an input layer represented by C_{IN} in Eq. 4.1, where $F_{k,s}(n_1, n_2)$ is the input to this layer, and S is the length of the $\boldsymbol{\beta}$ vector. $F_{k,s}(n_1, n_2)$ is produced by fusing the interpolated frames $g_k(n_1, n_2)$ using subpixel interframe registration information $R_k(n_1, n_2)$, resulting in fused frames $F_{k,s}(n_1, n_2)$ with the same dimensions as the ground truth image. The length of the $\boldsymbol{\beta}$ vector is denoted by S. As explained in 3.1.2, β determines the weight sensitivity to the distances in $g_k(n_1, n_2)$. In the original FIF method, a single value of β is used to fuse the frames, generating one fused frame that is then filtered with a Wiener filter. However, since the optimal value of β is unknown, we use a vector of $\boldsymbol{\beta}$ to increase the probability of obtaining the best value of β. This process results in multiple fused frames $F_{k,s}(n_1, n_2)$ that are used as input to the EEWF-FIFNET model. The fundamental idea behind this approach is that an interpolated pixel in close proximity to an original LR pixel possesses a lower interpolation error and should be assigned a higher weight. When diverse camera motion is present, each HR interpolated pixel is likely to have an LR pixel nearby in one or more of the input frames. The β vector is defined as $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_S]$.

$$F_s = C_{IN}(F_{k,s}), \tag{4.1}$$

next, the output of input layer fused C_{IN} which is F_s are fed to RIR producing a deep feature extraction F_{DF}. The RIR body consists of G "residual groups" followed by a "long skip connection" (LSC). Denoting the g^{th} residual group as $C_g(1 \leqslant g \leqslant G)$. Each RG further contains B residual channel attention blocks (RCAB) with short skip connection (SSC) as shown in Fig. 4.3, we can formulate the RIR body as

$$F_{DF} = C_{RIR}(F_s), \tag{4.2}$$

since the input dimension of our model is the same as the feature map and the ground truth image, the final step is to restore the image using a regression layer.

$$I_{SR} = C_{REC}(F_{DF}) = C_{EEWF-FIFNET}(F_{k,s}). \tag{4.3}$$

The reconstruction layer is denoted as $C_{REC}(\cdot)$ and the function of our EEWF-FIFNET model is denoted as $C_{EEWF-FIFNET}(\cdot)$. A channel attention mechanism is implemented in the network, as claimed in [3], to emphasize more useful features and increase discriminability. This mechanism allows for different attention to be applied to each channel-wise feature, determining the inter-dependencies among feature channels, as shown in Fig. 4.3. There are two main concerns to consider: Firstly, information in the LR space contains abundant low-frequency and valuable high-frequency components, where the low-frequency parts are usually more flat, and the high-frequency components are typically regions with edges, texture, and other details. Secondly, each filter in the $Conv$ layer operates within a local receptive field, limiting the network's ability to exploit contextual information beyond the local region.

4.2.3 IEWF-FIFNET Model Architecture

The Internally Exponential Weighted Fusion-FIFNET (IEWF-FIFNET) model is an end-to-end super-resolution (SR) approach, as depicted in Fig. 4.4. Its input is a set of upsampled and aligned interpolated frames, $g_k(n_1, n_2)$, along with their corresponding subpixel registration arrays, $R_k(n_1, n_2)$, for $k = 1, 2, ...K$. The model outputs a single high-quality SR image estimate. Our objective in designing the IEWF-FIFNET model is to achieve fusion and deconvolution within a single CNN architecture. The EWF layer

performs the fusion stage and is placed as the first layer after the input layer of the IEWF-FIFNET model, as shown in Fig. 4.4. To deconvolve or restore the imagery and generate the final SR image estimate, we employ a very deep network based on the residual in residual (RIR) structure of the RCAN model [3], which is originally designed for single image super-resolution (SISR) problems.

The IEWF-FIFNET internally fusion network begins with the EWF layer C_{EWF} as shown Eq. 4.4. The input of the EWF layer is Rg_k , where Rg_k is the concatenation of $g_k(n_1, n_2)$, and $R_k(n_1, n_2)$, and its output $F_{k,s}$ is S fused frames with the same dimension of the ground truth image.

$$F_{k,s} = C_{EWF}(Rg_k), \tag{4.4}$$

next, the fused frames $F_{k,s}$ are fed to RIR producing a deep feature extraction F_{DF}. The RIR body consists of G "residual groups" followed by a "long skip connection" (LSC). Denoting the g^{th} residual group as $C_g(1 \leqslant g \leqslant G)$. Each RG further contains B residual channel attention blocks (RCAB) with short skip connection (SSC) as shown in Fig. 4.4, we can formulate the RIR body as

$$F_{DF} = C_{RIR}(F_{k,s}). \tag{4.5}$$

The last step is restore the image.

$$I_{SR} = C_{REC}(F_{DF}) = C_{IEWF-FIFNET}(Rg_k), \tag{4.6}$$

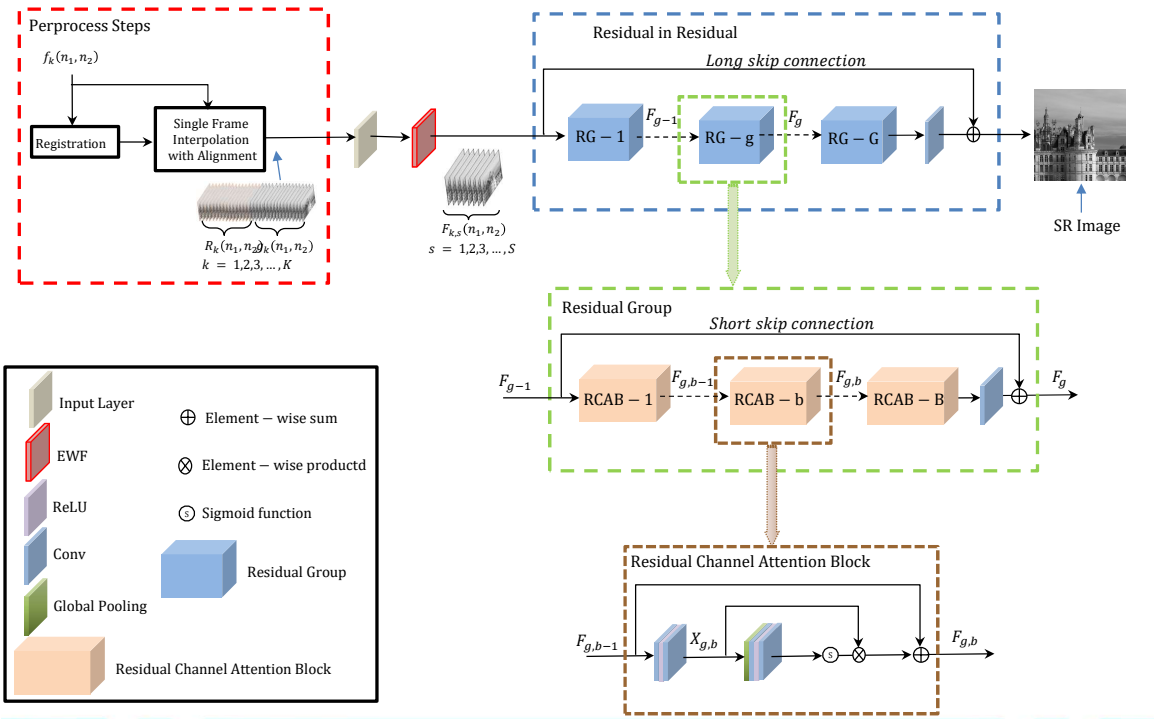


Figure 4.4: The architecture of the IEWF-FIFNET model consisted entirely of residual in residual (RIR) blocks arranged in a deep network structure as shown in the figure, where the preprocessing steps are enclosed by a red dashed box. The input of the model consists of upsampled and aligned interpolated frames, $g_k(n_1, n_2)$, and their corresponding subpixel registration arrays, $R_k(n_1, n_2)$. Our custom EWF layer, depicted by the red layer followed by the input layer, is incorporated into the model. The output of the model is a single SR image.

here, $C_{REC}(\cdot)$ and $C_{IEWF-FIFNET}(\cdot)$ refer to the reconstruction layer and the function of our IEWF-FIFNET model, respectively. The network backbone in both the EEWF-FIFNET and IEWF-FIFNET models is the same, which is the Residual-in-Residual (RIR) C_{RIR} structure.

4.2.4 Exponential Weighted Fusion (EWF) Layer

We have developed a custom layer called the Exponential Weighted Fusion (EWF) layer that combines input frames using a technique inspired by the FIF SR framework [18]. This

layer is located as the first layer after the input layer in our EWF-FIFNET model, as depicted in Fig. 4.4. The EWF layer merges interpolated frames $g_k(n_1, n_2)$ by utilizing subpixel interframe registration data $R_k(n_1, n_2)$, resulting in fused frames $F_{k,s}(n_1, n_2)$ as illustrated in Fig. 4.5. The length of the $\boldsymbol{\beta}$ vector is denoted by S. In our earlier approach (EEWF-FIFNET) model, the sensitivity of the weights to the distances in $R_k(n_1, n_2)$ was regulated by the parameter $\boldsymbol{\beta}$, but its optimal value was unknown. Thus, in the IEWF-FIFNET model, We have enabled $\boldsymbol{\beta}$ to be a trainable parameter, allowing it to update itself during the training of the IEWF-FIFNET model in order to obtain the optimal $\boldsymbol{\beta}$ value. To increase the likelihood of obtaining different good $\boldsymbol{\beta}$ values for the fusion process, we employ a vector of $\boldsymbol{\beta}$ with a length of S. The β vector is expressed as $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_S]$.

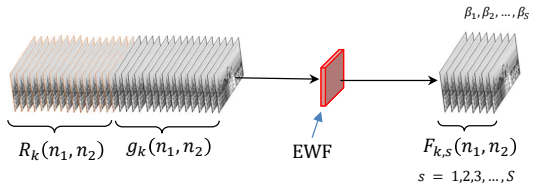


Figure 4.5: The EWF Layer is a custom layer used to fuse interpolated frames using subpixel interframe registration information to produce high-quality fused frames. EWF Layer fuses the interpolated frames $g_k(n_1, n_2)$ using subpixel interframe registration information $R_k(n_1, n_2)$ with different vaules of $\boldsymbol{\beta}$ to produce fused frames $F_{k,s}(n_1, n_2)$.

The EWF's code sequence is pseudocode in Algorithm. 1 illustrates the following of the code. The input of the EWF layer is composed of the upsampled and aligned interpolated frames, $g_k(n_1, n_2)$, and the corresponding subpixel registration arrays interpolated aligned frames $R_k(n_1, n_2)$. The fusion process is as follows. First, we initiate a $\boldsymbol{\beta}$ vector with a length of S using random numbers. Then, we compute the weights, $w_k(n_1, n_2)$, which are expressed as Eq. 4.7. The weights are a function of the distance of each interpolated pixel to the nearest original LR pixel.

Algorithm 1 The code sequence description of Exponentially weighted Fusion (EWF) layer

1: $\boldsymbol{\beta} \leftarrow$ Generate a random single array length of S
2: **for** $s = 1, 2, \ldots, S$ **do**
3: Compute weights, $w_{k,s}(n_1, n_2) = e^{(-R_k(n_1,n_2)^2/\boldsymbol{\beta}_s^2)}$
4: Fuse the interpolated frames, $FIF = \sum_{k=1}^{K} g_k(n_1, n_2) \cdot w_{k,s}(n_1, n_2)$
5: Compute advantage of fused frames, $F_{k,s}(n_1, n_2) = FIF./\sum_{k=1}^{K} w_{k,s}(n_1, n_2)$
6: Stack of S advantage of fused frames $F_S(n_1, n_2)$ as an input of the next stage of the network
7: Optimize/Update $\boldsymbol{\beta}$
8: $\boldsymbol{\beta}_{old} \leftarrow \boldsymbol{\beta}$
9: **goto** step 2

$$w_{k,s}(n_1, n_2) = e^{-R_k(n_1,n_2)^2/\beta_s^2}. \tag{4.7}$$

Next, the interpolated frames are then fused on a pixel-wise basis using a weighted sum operation. The element-wise right division is performed between matrices FIF and the sum of $w_k(n_1, n_2)$ where $k = 1, 2, 3, \ldots, K$ to compute advantage of fused frames This is given by

$$F_{k,s}(n_1, n_2) = \frac{\sum_{k=1}^{K} w_{k,s}(n_1, n_2) \cdot g_k(n_1, n_2)}{\sum_{k=1}^{k} w_{k,s}(n_1, n_2)}. \tag{4.8}$$

Note that each β value produces a different single fused frame $F(n_1, n_2)$, we repeat the process of computing the weights $w_{k,s}(n_1, n_2)$ following with frames fusion to obtain, $F(n_1, n_2)$, S times. As a result, the output of EWF layer is a stack of S fused frames $F_{k,s}(n_1, n_2)$ as Fig. 4.5 shows.

4.2.5 EWF-FIFNET Training

We trained our EWF-FIFNET networks using the default settings of the original RCAN network, which included $G = 10$ residual groups and $B = 20$ residual blocks. During training, we used a patch size of 48×48 pixels with a batch size of 16 patches and the Adam optimizer with a learning rate of $1e^{-4}$. After conducting several experiments, we selected a length vector of beta equal to $S = 7$. For the EEWF-FIFNET model, we fixed the values of $\boldsymbol{\beta}$ at $[0.9, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01]$ after trying different values. For the IEEWF-FIFNET model, we set the values of beta to be learnable parameters and randomly initialized the beta vector with a random vector of length 7. Both EWF-FIFNET models underwent 120 epochs, corresponding to 1600 passes through the data, and we decreased the learning rate by a factor of 0.5 at steps 120 and 160. To train the networks, we used a Windows workstation with an Intel Core i9-9980XE Processor clocked at 3.0 GHz and two NVIDIA TITAN RTX Graphics Processing Units (GPUs). We utilized both GPUs to simultaneously train multiple experiments of the EWF-FIFNET network, which required approximately three days to complete.

CHAPTER V

RESULTS AND DISCUSSION

The Results and Discussion chapter of our dissertation presents a comprehensive analysis and interpretation of the data collected from our study of FIFNET's efficacy. In this section, we provide an overview of the experimental results obtained to demonstrate the effectiveness of FIFNET.

To evaluate FIFNET's performance, we utilized a range of experimental approaches, including simulated degradation of imagery for quantitative analysis and real camera data for subjective analysis in a practical application. First, we conducted a quantitative analysis using imagery with simulated degradation to assess FIFNET's performance. The results of this analysis are presented in detail, including statistical metrics and graphical representations, to illustrate the effectiveness of FIFNET in restoring degraded images.

Next, we processed real camera data with no artificial degradation for subjective analysis in a real application. This approach enabled us to assess the practical effectiveness of FIFNET in a real-world scenario, and the results of this analysis are presented in the following section of this chapter. In this section, we discuss the subjective assessment of the restored images and compare them to the original images.

Overall, our experimental results provide compelling evidence of FIFNET's efficacy in restoring degraded images. The quantitative analysis demonstrates the algorithm's ability to restore images with a high degree of accuracy, while the subjective analysis highlights the algorithm's practical effectiveness in real-world applications. These results underscore the importance of FIFNET in enhancing the quality of digital images and its potential for a range of applications in various fields.

5.1 FIFNET Experimental Results

In this section we present a number of experimental results to demonstrate the efficacy of FIFNET. First, we use imagery with simulated degradation to provide a quantitative performance analysis. Next, we process real camera data with no artificial degradation for subjective analysis in a real application.

5.1.1 Simulated Data

The imagery used for testing come from three publicly available databases. These are the DIV2K Validation dataset [33] (100 images), the Set14 dataset [33] (14 images), and the BSDS100 dataset [35] (100 images). None of the images contained in these databases have been used in training. We employ two performance metrics, Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) [36].

We consider two motion model scenarios. The first is what we call the Fixed Shifts (FS) scenario. Here, the set of interframe shifts for the testing data are known *a priori* and are fixed. All training images are artificially degraded using the observation model with these same fixed shifts. The FS scenario might occur in practice when the best possible result is desired for one test sequence and network training time is not a major consideration. The second case is what we call the Random Shifts (RS) scenario. Here each training image gets a different random shift with the hope of making FIFNET robust to shift pattern. Using this approach, one trained network can be applied regardless of the shift pattern in the testing sequence. The RS FIFNET could then be used to process video by employing a temporal moving window of frames. In addition, we evaluate FIFNET with and without

using $R_k(n_1, n_1)$ as part of the input channels. We do this to study the importance of the subpixel registration information channels.

The average PSNR results for the DIV2K Validation dataset as a function of the number of input frames is provided in Fig. 5.1. We use $L = 3$ and the noise standard deviation is $\sigma_\eta = 0.001$. A curve for each of the scenarios described in the previous paragraph is included. We also include three single frame methods as benchmarks and we show these with dashed lines. These are single frame bicubic interpolation, VDSR [2], and RCAN [3]. The VDSR and RCAN networks are trained here using the same data and degradation model as those for FIFNET. We also include comparison results for two additional multiframe SR methods. These are the original FIF SR [1] and the Adaptive Wiener Filter (AWF) SR method [24, 37]. For the AWF method we apply a window size of 9×9 and a one-step correlation value of $\rho = 0.7$.

Figure 5.1 shows that in this experiment the FIFNET outperforms the benchmark methods for $K > 1$. For $K = 1$, the single-frame RCAN method provides the best results. Also, note that increasing the number of input frames significantly improves the FIFNET performance. As expected, the FS scenario yields higher PSNR than the RS scenario. However, the FS network is only intended for the one set of shifts it was trained on. Another important thing to observe in Fig. 5.1 is the boost in PSNR that results from including the registration distance images as additional input channels (see "with R" in the legend). This clearly demonstrates the importance of providing subpixel registration information to the network. We have also observed that not concatenating the input frames at the end of the network leads to a drop in PSNR of approximately 1.7 dB at $K = 10$.

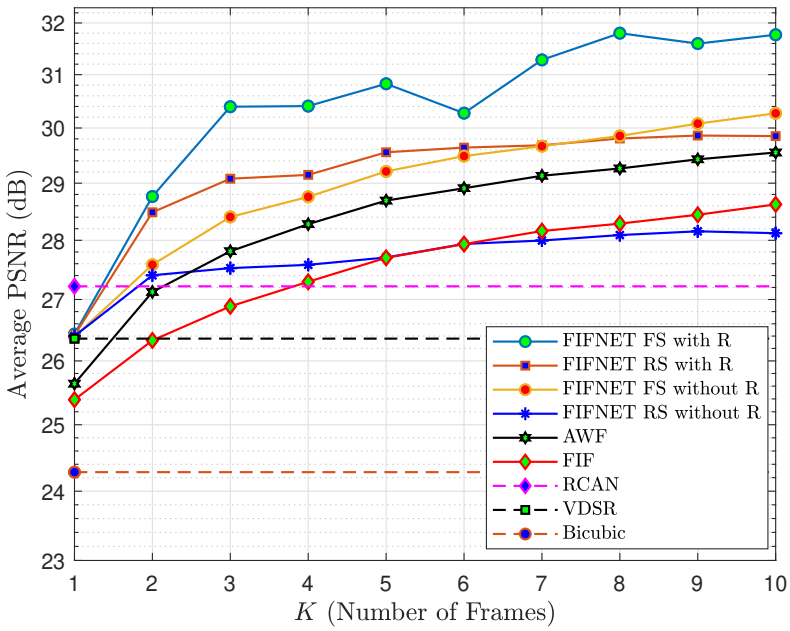


Figure 5.1: Quantitative performance comparison using simulated data from the DIV2K Validation dataset for $L = 3$ and $\sigma_\eta = 0.001$. The average PSNR is shown as a function of the number of input frames for the methods shown in the legend.

Additional quantitative results are provided in Table 5.1. These results show PSNR and SSIM for all three test databases. In these experiments the best quantitative results are obtained using the FIFNET with FS scenario and $K = 8$ or $K = 10$. This is followed by the FIFNET with RS scenario.

In addition to the quantitative results, a number of processed images are provided for subjective evaluation in Figs. 5.2-5.4. In each of these figures the truth image is shown in (a), the truth region of interest (ROI) is shown in (b), and the various processed ROIs are shown in (c)-(h). Note that the images in the top rows of these figures are for single frame methods, and the bottom rows are for multiframe methods using $K = 10$ frames. The error metric values associated with the images are listed in the captions.

Table 5.1: Average PSNR(dB)/SSIM for $K = 1, 4, 8$, and 10 using different methods and three different datasets: DIV2K Validation, Set14, and BSDS100. The **bold** numbers indicate the best performance for the corresponding metric and dataset category.

Dataset	K	Bicubic	VDSR	RCAN	FIF	AWF	FIFNET RS	FIFNET FS
DIV2K Validation [33]	1	24.28/0.720	26.36/0.797	27.07/0.831	25.38/0.777	25.64/0.787	26.41/0.811	26.43/0.812
	4	-	-	-	27.18/0.848	28.33/0.879	29.36/0.900	30.41/0.914
	8	-	-	-	28.10/0.875	29.57/0.907	29.91/0.910	**31.80**/0.930
	10	-	-	-	28.13/0.875	29.84/0.912	30.01/0.912	31.77/**0.932**
Set14 [33]	1	24.91/0.719	27.33/0.783	27.86/0.816	26.11/0.771	26.38/0.778	27.43/0.803	27.45/0.803
	4	-	-	-	27.85/0.837	29.03/0.865	30.14/0.884	31.22/0.899
	8	-	-	-	28.77/0.862	30.26/0.893	30.71/0.895	**32.36**/0.915
	10	-	-	-	28.80/0.863	30.54/0.898	30.77/0.897	32.32/**0.916**
BSDS100 [35]	1	25.08/0.690	26.74/0.747	27.39/0.783	25.98/0.743	26.17/0.752	26.84/0.770	26.86/0.771
	4	-	-	-	27.54/0.812	28.48/0.845	29.37/0.868	30.30/0.886
	8	-	-	-	28.32/0.840	29.59/0.878	29.88/0.880	**31.63**/0.909
	10	-	-	-	28.35/0.841	29.83/0.884	29.99/0.884	31.60/**0.911**

The Car image in Fig. 5.2 gives a good illustration of typical results. Note that bicubic interpolation image in Fig. 5.2 (c) shows significant blurring as well as aliasing artifacts in the form of jagged edges along the borders of the license plate. The RCAN single-frame method in Fig. 5.2 (d) provides significant sharpening and does a good job reconstructing the edges of the license plate. However, the ill-posed nature of the inverse problem is exposed on the numbers "6" and "8" here. The multiframe methods are able to recover the numbers reasonably well. However, the original FIF SR method in Fig. 5.2 (e) has artifacts on the license plate perimeter. The AWF SR method in Fig. 5.2 (f) appears better on the edges and numbers. However, the FIFNET results in Figs. 5.2 (g) and (h) appear to provide the sharpest images with minimal aliasing artifacts. Note that the FIFNET FS result in Fig. 5.2 (h) is 2.63 dB higher than AWF SR. Similar observations can be made in Figs. 5.3 and 5.4. It is interesting to note the dark thin wire in Fig. 5.3 that runs from the top of the light post to the right middle edge of the image. We believe that this detail is most prominently restored in Fig. 5.3 (h) using FIFNET FS.

In a final experiment with the simulated data, we have performed a study of the processing time for the various methods using the 100 image DIV2K Validation dataset. The average testing time per image is plotted versus K in Fig. 5.5. The reported times include

Figure 5.2: Results for image "0879" in the DIV2K validation dataset. The PSNR(dB)/SSIM values are (b) 23.99/0.804, (c) 25.96/0.863, (d) 26.44/0.878, (e) 28.55/0.923, (f) 29.00/0.928, (g) 29.71/0.935, and (h) 31.63/0.938. The noise has a standard deviation of $\sigma_\eta = 0.001$ and $K = 10$ frames are used in (e)-(h).

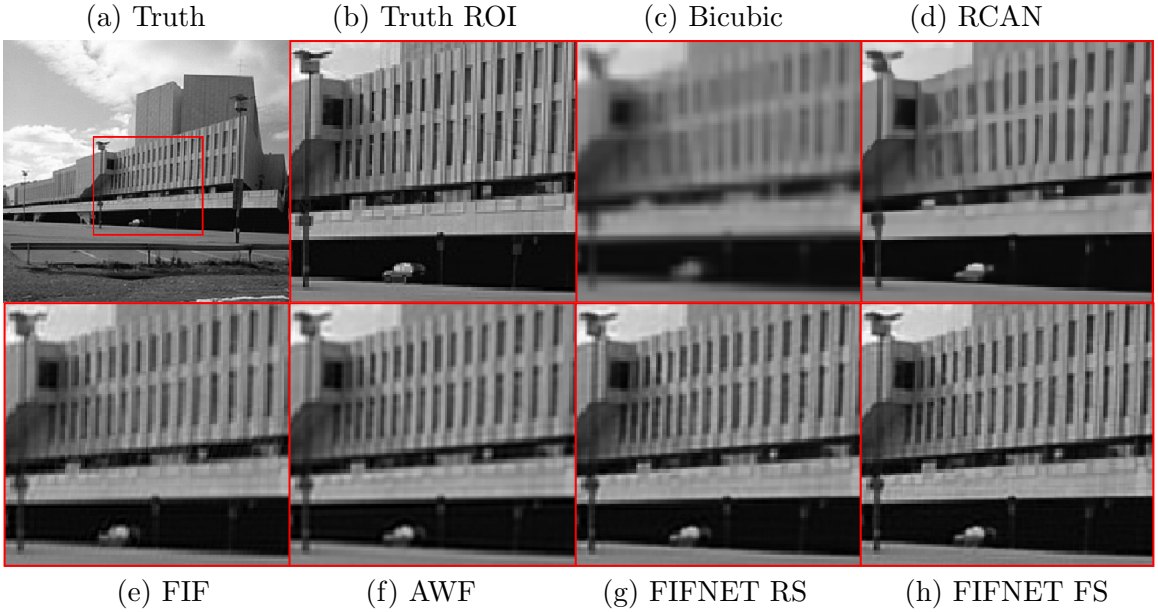


Figure 5.3: Results for image "078" in the BSD100 dataset. The PSNR(dB)/SSIM values are (b) 23.46/0.665, (c) 25.23/0.737, (d) 26.74/0.803, (e) 27.50/0.854, (f) 27.80/0.862, (g) 28.21/0.867, and (h) 30.34/0.908. The noise has a standard deviation of $\sigma_\eta = 0.001$ and $K = 10$ frames are used in (e)-(h).

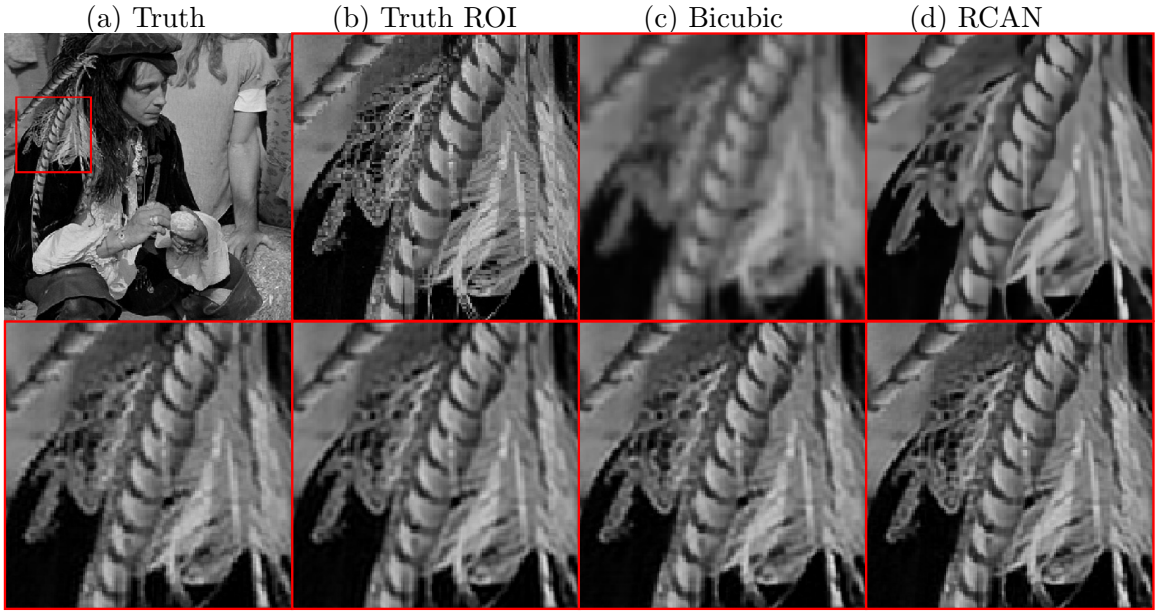


Figure 5.4: Results for the image "Man" in the Set14 dataset. The PSNR(dB)/SSIM values are (c) 24.74/0.684, (d) 27.09/0.784, (e) 28.36/.853, (f) 29.11/0.872, (g) 29.24/0.867, and (h) 30.88/0.899. The noise has a standard deviation of $\sigma_\eta = 0.001$ and $K = 10$ frames are used in (e)-(h).

the time for all of the operations required by the stated method, such as registration and interpolation preprocessing for FIFNET. The single-frame results (i.e., VDSR and RCAN) are shown with dashed lines. The methods are implemented here on a laptop computer with Core i7-7700HQ Processor using 2.8GHz clock speed and NVIDIA GTX 1070 GPU. RCAN is implemented in Python and all of the other methods are implemented in MATLAB. As expected, the multiframe methods have longer processing times as more input frames are included. The AWF method is the fastest of the multiframe methods here. However, note that the computational complexity of AWF is significantly higher for interframe motion other than translational [21]. For FIFNET, the type of motion does not impact the processing time of the CNN. With regard to padding, one can see in Fig. 5.5

that FIFNET without padding is slightly slower than FIFNET with padding during testing. We attribute this to the extra crop layers, shown in Fig. 5.5, for the no-padding FIFNET architecture. Thus, there is a small tradeoff with regard to testing time versus training time and performance. We attribute the relatively long processing times for RCAN to the large network size. Note that all of the CNN-based methods can be accelerated with a specialized hardware implementation.

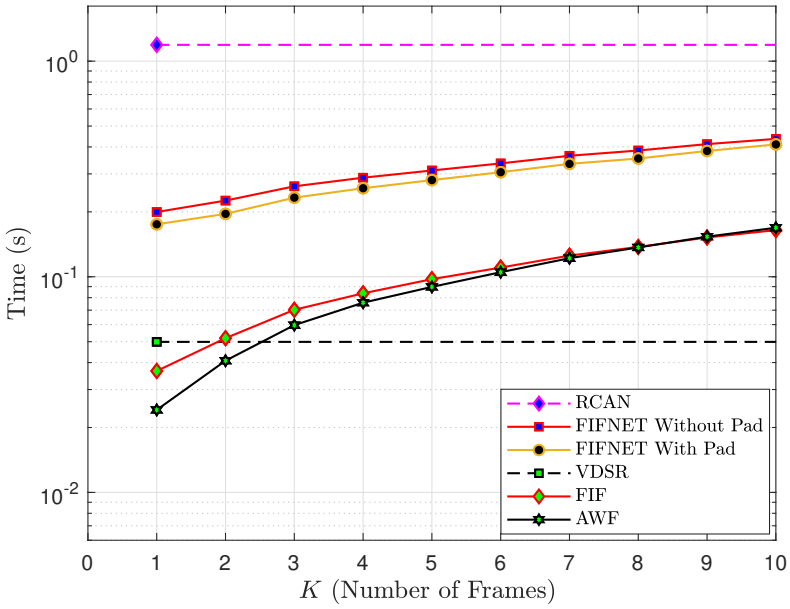


Figure 5.5: Average testing time per image for FIFNET and several benchmark method using simulated data from the DIV2K Validation dataset.

5.1.2 Real Camera Data

The true test of any SR algorithm is to process real camera data that have not been artificially degraded. In this section we present several results for such a scenario using image data acquired by the authors. Because the data are not artificially degraded, there are no corresponding ground truth images. The results presented here are for subjective

(a) Bicubic (b) Bicubic ROI (c) RCAN

(d) FIF (e) AWF (f) FIFNET RS

Figure 5.6: Image results for the real camera data of a brick house. The images shown are (a) bicubic, (b) bicubic ROI, (c) RCAN, (d) FIF SR, (e) AWF and (f) FIFNET RS. The multiframe methods use $K = 10$ frames for (d)-(f).

(a) Bicubic (b) Bicubic ROI (c) RCAN

(d) FIF (e) AWF (f) FIFNET FS

Figure 5.7: Image results for the real camera data of a bookshelf. The images shown are (a) bicubic, (b) bicubic ROI, (c) RCAN, (d) FIF SR, (e) AWF and (f) FIFNET FS. The multiframe methods use $K = 10$ frames for (d)-(f).

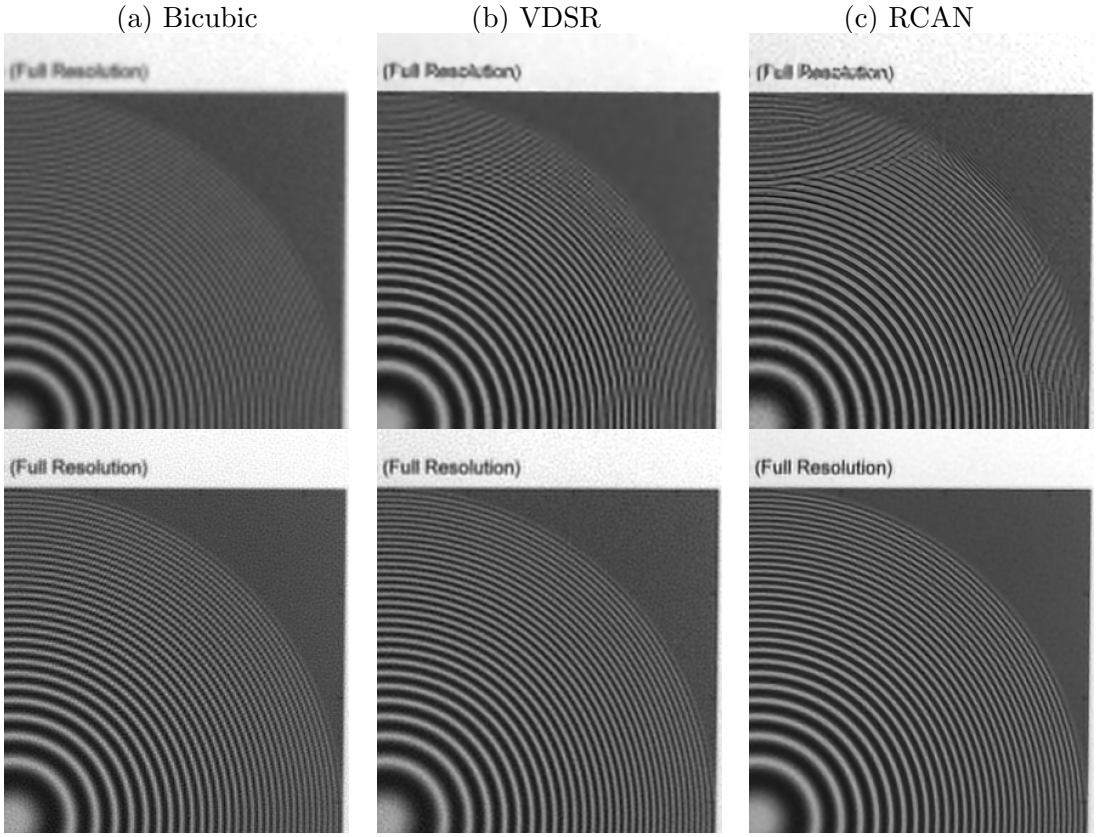


Figure 5.8: Image results for the real camera data of a chirp pattern scene. The images shown are (a) bicubic, (b) VDSR, (c) RCAN, (d) FIF SR, (e) AWF, and (f) FIFNET FS. The multiframe methods use $K = 10$ for (d)-(f).

evaluation purposes. We have chosen familiar scene content and a well-defined test pattern to facilitate the evaluation.

Since the observation model presented in Section II is based on a realistic camera OTF, our trained network can be applied directly to data from that camera. The interframe motion is created here by manually panning and tilting the camera on a tripod during video acquisition that was at a rate of 30 frame per second. We present results for three

distinct datasets in Figs. 5.6-5.8. For each of the datasets, the multiframe methods use $K = 10$ frames with $L = 3$.

Figure 5.6 shows results for an outdoor scene of a brick house. Note that the single-frame methods in Figs. 5.6 (b) and (c) have difficulty restoring the texture of the brick pattern. On the other hand, all of the multiframe methods in Figs. 5.6 (d) - (f) appear to restore the brick pattern fairly well. The FIFNET RS result trained using $\sigma_\eta = 0.001$ is shown in Fig. 5.6 (f). It appears to have a more sharpness than FIF SR, and more noise suppression than AWF.

Results for an indoor bookshelf dataset are shown in Fig. 5.7. Here the aliasing pattern is evident on the lettering in the bicubic interpolation image in Fig. 5.7 (b). The RCAN method appears to do a good job reducing the aliasing artifacts on the large letters, but has more difficulty with the smaller lettering. The FIFNET FS result in Fig. 5.7 (f) appears to provide better sharpness, compared with AWF and FIF, and more noise suppression. Training was done here using $\sigma_\eta = 0.01$ due to the lower indoor lighting level and corresponding decreased signal-to-noise ratio.

Finally, a circularly-symmetric chirp pattern image is shown in Fig. 5.8. This test image has been selected to clearly illustrate the aliasing reduction capabilities of the methods being tested. Training has also been done here using $\sigma_\eta = 0.01$. Aliasing in the form of a Moiré pattern is visible on the high frequency components of the chirp in the bicubic interpolation image in Fig. 5.8 (a). The aliasing causes the concentric ring pattern to appear inverted on the top and on the right in the image. The single frame restoration methods shown in Figs. 5.8 (b) and (c) improve contrast, but are unable to correctly decode the true chirp structure from the undersampled imagery. For a single-frame method, RCAN does do a

good job on most of the chirp pattern. Note that the multiframe methods shown in Figs. 5.8 (d)-(f) correctly restore all of the rings of the chirp and improve the readability of the title text. As with the bookshelf data, FIFNET FS in Fig. 5.8 (f) appears to show the best sharpness and more noise reduction than AWF and FIF.

5.2 EWF-FIFNET Experimental Results

This section demonstrates the performance evaluation of the IEWF-FINET model and the EEWF-FIFNET model using both simulated and real data, and presents the obtained results. Initially, we perform a quantitative analysis of performance using imagery that has simulated degradation using the degradation model in chapter II. Then, we process real camera data with no artificial degradation to conduct a subjective analysis in a real-world application. Finally, we compare the performance of both models against chosen benchmarks and classify them as good comparators.

5.2.1 Simulated Data

Three publicly available databases are used to test the imagery, namely the DIV2K Validation dataset [33] (100 images), the Set14 dataset [33] (14 images), and the BSDS100 dataset [35] (100 images), with none of the images being used in training. The performance of the method is evaluated using two metrics, namely Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) [36]. To provide context, two single frame methods are used as benchmarks and are represented by dashed lines: single frame bicubic interpolation, and RCAN [3]. The RCAN network is trained using the same data and degradation model as that of EWF-FIFNET. Additionally, comparison results are included for an additional multiframe SR method, namely the original FIF SR [23].

To simulate the degradation process in our study, we used an observation model that includes various sources of degradation, including noise and blur. In addition, we incorporated affine motion into the model to simulate the effects of object motion in real-world image acquisition systems. We applied affine motion to each image in the training and testing sets using a transformation matrix that captures translation, rotation, scaling, and shearing parameters. The transformation matrix was randomly generated for each image to simulate different types of motion. To apply the affine motion, we used an affine transformation function in MATLAB that applies the transformation matrix to the image as shown above in the equations 2.1, and 2.2. We then added noise and blur to the transformed image to simulate the effects of other sources of degradation. By incorporating affine motion into the observation model, we were able to generate a more realistic and diverse set of degraded images for our study. This allowed us to evaluate the performance of our image processing algorithm under a range of motion conditions and develop a more robust algorithm that can handle motion artifacts in real-world scenarios.

The graph in Fig.5.9 shows the average PSNR results for the DIV2K Validation dataset, based on the number of input frames used. The value of L is set to 3, and the noise standard deviation is $\sigma_\eta = 0.001$. The graph includes curves for each scenario described earlier, as well as benchmarks for three single frame methods, shown with dashed lines. These benchmarks include single frame bicubic interpolation and RCAN, which is trained using the same data and degradation model as FIFNET and EWF-FIENET. Additionally, two other multiframe SR methods are compared, including the original FIF SR.

The results show that in this experiment, the EWF-FIFNET outperforms the benchmark methods for $K > 1$, with similar performance to the single-frame RCAN method for $K = 1$. Furthermore, increasing the number of input frames improves the EWF-FIFNET's perfor-

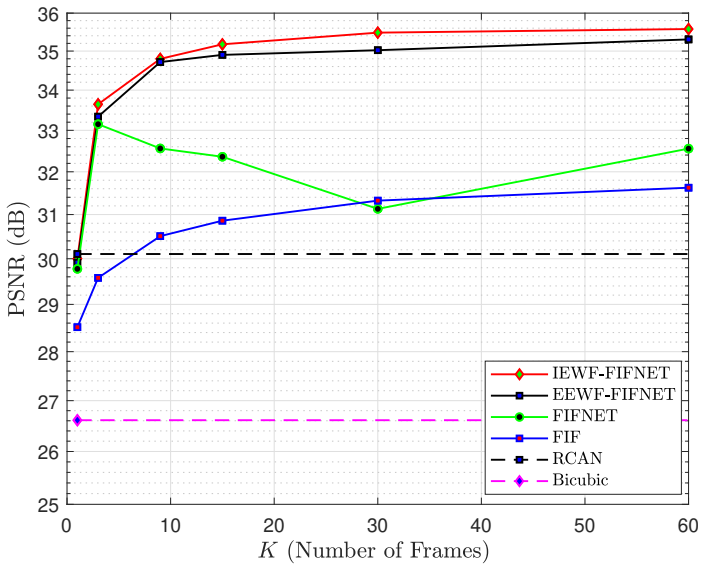
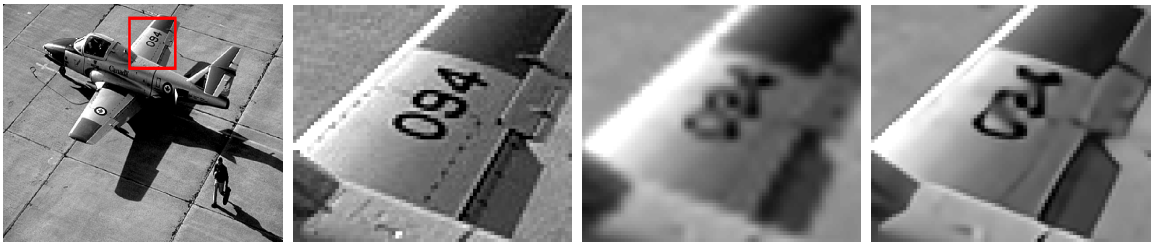
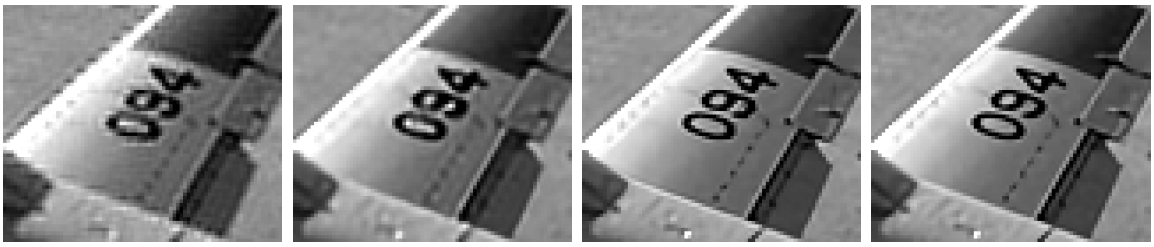


Figure 5.9: Quantitative performance comparison using simulated data from the DIV2K Validation dataset for $L = 3$ and $\sigma_\eta = 0.001$. The average PSNR is shown as a function of the number of input frames for the methods shown in the legend.

mance significantly. This is demonstrated by the smooth increase in the PSNR performance curve of the EWF-FIFNET model in Fig. 5.9 as the number of input frames K increases. Table 5.2 presents additional quantitative results, including PSNR and SSIM values, for all three test databases. The results demonstrate that the EWF-FIFNET model with K = 60 yields the best quantitative performance across all three databases. Figs. 5.10-5.13 provide several processed images for subjective evaluation, in addition to the quantitative results. Each figure includes the truth image in (a), the truth region of interest (ROI) in (b), and the processed ROIs in (c)-(d). It is important to note that the images in the top rows of these figures are for single frame methods, while the bottom rows are for multiframe methods with K = 60 frames. The captions of the figures provide the error metric values associated with the images.



(a) Truth (b) Truth ROI (c) Bicubic (d) RCAN



(e) FIF (f) FIFNET (g) EEWF-FIFNET (h) IEWF-FIFNET

Figure 5.10: Results for image "071" in the BSDS100 dataset. The PSNR(dB)/SSIM values are (c) 26.755/0.726, (d) 30.895/0.839, (e) 31535/0.876, (f) 33.183/0.886, (g) 36.892/0.941, and (h) 36.949/0.932. The noise has a standard deviation of $\sigma_\eta = 0.001$ and $K = 60$ frames are used in (e)-(h).

(a) Truth (b) Truth ROI (c) Bicubic (d) RCAN

(e) FIF (f) FIFNET (g) EEWF-FIFNET (h) IEWF-FIFNET

Figure 5.11: Results for image "91" in the DIV2K dataset. The PSNR(dB)/SSIM values are (c) 24.445/0.753, (d) 28.279/0.878, (e) 27.993/0.878, (f) 30.776/0.928, and (g) 36.532/0.971, and (h). The noise has a standard deviation of $\sigma_\eta = 0.001$ and $K = 60$ frames are used in (e)-(h).

Table 5.2: Average PSNR(dB)/SSIM for $K = 1, 3, 9, 15, 30$ and 60 using different methods and three different datasets: BSDS100, Set14, and DIV2K Validation. The **bold** numbers indicate the best performance for the corresponding metric and dataset category, **Crop = 0**.

Dataset	K	PSNR(dB)/SSIM					
		Bicubic	RCAN	FIF	FIFNET	EWF-FIFNET	
						Externally Fusion	Internally Fusion
BSDS100	1	23.553/0.692	26.611/0.783	25.956/0.742	26.315/0.776	26.428/0.781	26.464/0.782
	3	-	-	27.010/0.796	29.406/0.877	29.824/0.887	30.008/0.888
	9	-	-	28.069/0.840	30.194/0.898	31.624/0.922	31.799/0.924
	15	-	-	28.480/0.855	29.940/0.896	32.003/0.929	32.451/0.933
	30	-	-	28.961/0.870	30.212/0.898	32.791/0.939	32.927/0.940
	60	-	-	29.328/0.881	29.811/0.888	**33.369/0.947**	33.346/0.948
Set14	1	24.314/0.720	27.610/0.815	25.971/0.770	27.381/0.809	27.440/0.814	27.694/0.815
	3	-	-	27.006/0.821	30.540/0.892	31.250/0.900	31.406/0.902
	9	-	-	28.192/0.859	31.131/0.909	32.951/0.928	33.236/0.930
	15	-	-	28.600/0.871	30.970/0.99	33.346/0.934	33.731/0.936
	30	-	-	29.068/0.883	31.013/0.909	33.959/0.940	34.133/0.942
	60	-	-	29.519/0.894	30.906/0.904	**34.615/0.947**	34.522/0.948
DIV2K validation	1	26.611/0.790	30.105/0.867	28.513/0.829	29.775/0.860	29.9124/0.865	30.000/0.866
	3	-	-	29.577/0.862	33.151/0.921	33.337/0.923	33.644/0.928
	9	-	-	30.507/0.887	32.559/0.905	34.716/0.945	34.798/0.941
	15	-	-	30.859/0.894	32.359/0.907	34.901/0.945	35.176/0.944
	30	-	-	31.322/0.902	31.131/0.892	35.022/0.947	35.482/0.949
	60	-	-	31.624/0.907	32.555/0.915	35.304/0.951	**35.577/0.953**

5.2.2 Real Camera Data

Using real-world images for testing super-resolution models provides a more realistic evaluation of their performance, as opposed to synthetic images that may not fully capture the nuances of real-world scenarios. It also allows for a more comprehensive analysis of the model's performance across various scenarios, including different lighting conditions, textures, and artifacts that occur in real-world image acquisition. In this subsection, we present several results that showcase the algorithm's efficacy under such circumstances, using image data that was acquired by the authors. Because the data has not been artificially degraded, there are no corresponding ground truth images available for comparison. The results presented here are solely for subjective evaluation purposes, and to facilitate this process, we have selected familiar scene content and a well-defined test pattern. Since the

(a) Truth (b) Truth ROI (c) Bicubic (d) RCAN

(e) FIF (f) FIFNET (g) EEWF-FIFNET (h) IEWF-FIFNET

Figure 5.12: Results for image "10" in the Set14 dataset. The PSNR(dB)/SSIM values are (c) 25.025/0.688, (d) 27.532/0.796, (e) 29.165/0.873, (f) 30.255/0.886, (g) 33.147/0.934 and (h) 33.168/0.934. The noise has a standard deviation of $\sigma_\eta = 0.001$ and $K = 60$ frames are used in (e)-(h).

observation model presented in Chapter II is based on a realistic camera OTF, the trained network can be applied directly to data from that camera. The interframe motion was generated by manually panning and tilting the camera on a tripod during image acquisition, which was carried out at a rate of 30 frames per second. We present the results for three distinct datasets in Figs. 5.15-5.18. For each dataset, the multiframe methods use K=60 frames with L=3, as a patch size. Overall, the presented results showcase the potential of the algorithm when processing real camera data without artificial degradation.

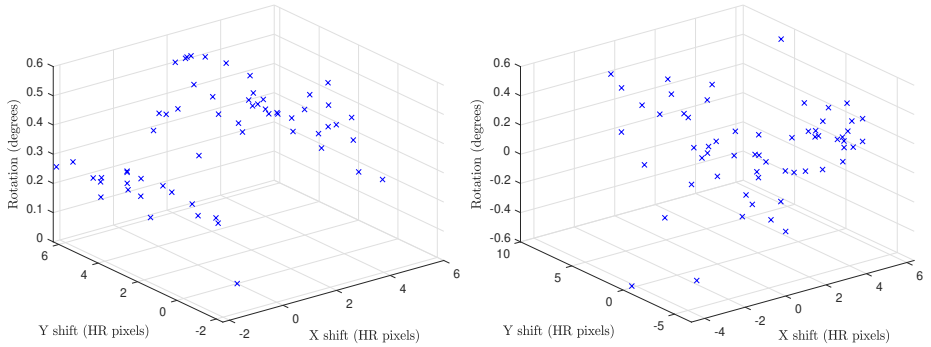
(a) Truth (b) Truth ROI (c) Bicubic (d) RCAN

(e) FIF (f) FIFNET (g) EEWF-FIFNET (h) IEWF-FIFNET

Figure 5.13: Results for image "92" in the DSBS100 dataset. The PSNR(dB)/SSIM values are (c) 25.025/0.688, (d) 27.532/0.796, (e) 27.165/0.793, (f) 29.318.255/0.889, (g) 34.932/0.960 and (h) 34.753/0.960. The noise has a standard deviation of $\sigma_\eta = 0.001$ and $K = 60$ frames are used in (e)-(h).

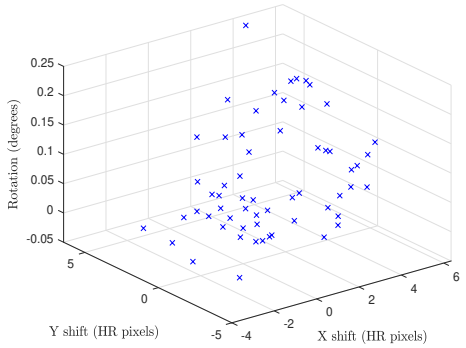
Fig. 5.15 show the results of applying various super-resolution (SR) methods to an outdoor image of a brick house. Notably, the single-frame methods in Fig. 5.15(b),(c) struggle to restore the complex texture of the brick pattern. In contrast, all of the multi-frame methods in Fig. 5.15(d)-(f) appear to restore the brick pattern quite effectively. Of these methods, the EWF-FIFNET result is shown in Fig. 5.15(f) and seems to produce sharper edges than FIF SR and better noise suppression than FIFNET. Furthermore, we utilize indoor sensing to validate our model's performance. Fig. 5.16 illustrates the application of SR models on a bookshelf dataset captured indoors. In Fig. 5.16(b), the bicubic interpolation image depicts an aliasing pattern on the lettering. Although the RCAN method managed to decrease the aliasing artifacts on larger letters, it struggled with smaller lettering. On the

other hand, the EWF-FIFNET outcome shown in Fig. 5.16(f) offers better sharpness than FIF and FIFNET, along with superior noise suppression. In real-world scenarios, the noise in the images may differ depending on whether they are captured indoors or outdoors. The image in Fig. 5.18 depicts a circularly-symmetric chirp pattern that is specifically chosen to demonstrate the ability of the tested methods to reduce aliasing. The image is trained using $\sigma_\eta = 0.01$. As there is no actual ground truth available for real images, a photograph taken using an iPhone Pro Max 14 are used as a reference for the scenes, as depicted in Figs. 5.18(a), and 5.17(a). The bicubic interpolation image in Fig. 5.18(b) shows aliasing in the form of a Moiré pattern on the high frequency components of the chirp, causing the concentric ring pattern to appear inverted on the top and right of the image. The single-frame restoration methods displayed in Figs. 5.18(b) and (c) enhance contrast, but are unable to accurately reconstruct the true chirp structure from the undersampled imagery. RCAN performs relatively well in reconstructing most of the chirp pattern for a single-frame method. It is worth noting that the multiframe methods demonstrated in Figs. 5.18 (d)-(f) are capable of properly restoring all the rings of the chirp and improving the legibility of the title text. EWF-FIFNET in Fig. 5.18(f) demonstrates the best sharpness and greater noise reduction compared to FIF and EWF-FIFNET, as is also observed in the line pair data in Fig.5.17(f). In conclusion, the chirp pattern image serves as a suitable example to demonstrate the superiority of multiframe methods over single-frame methods in reducing aliasing and restoring the details of complex patterns. To address this issue, we train the model using different σ_η values for every 800 sub-images of 6921 sub-images in the training dataset. We start with $\sigma_\eta = 0.001$, and for every 800 sub-images, it is updated by $\sigma_\eta = \sigma_\eta \times 1.5$. The last used σ_η in our dataset is $\sigma_\eta = 0.025$, making our model more robust in handling unknown noise ratios in real data.

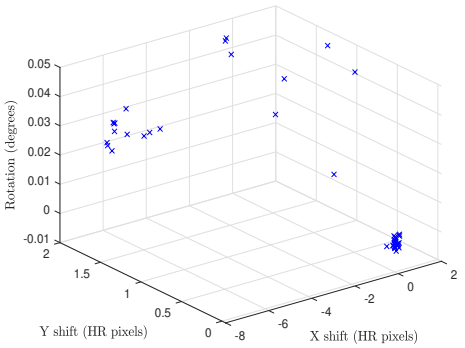


(a) Affine Motion on the chirp

(b) Affine motion on the line pair



(c) Affine motion on the brick house



(d) Affine motion on the bookshelf

Figure 5.14: Display estimated affine motion in 60 frames of each real data that are used in the multiframes methods as following images: (a) chirp image, (b) line pair image, (c) brick house image, and (d) bookshelf image.

In order to display the estimated affine motion in 60 frames of each real data used in the multiframes methods as shown in 5.14, we examine four different images: the chirp image, the line pair image, the brick house image, and the bookshelf image. Affine motion refers to a transformation of an image that preserves parallelism and ratios of distances. By estimating the affine motion of each frame in a sequence, we can understand how the image is moving and changing over time. In each of the four images, we will analyze 60 frames of real data and use the multiframes methods to estimate the affine motion. The resulting motion estimates will be displayed to provide insight into the movement and transformations of the images over time.

(a) Bicubic

(b) Bicubic ROI

(c) RCAN

(d) FIF

(e) FIFNET

(f) EEWF-FIFNET

(g) IEWF-FIFNET

Figure 5.15: Image results for the real camera data of a brick house. The images shown are (a) bicubic, (b) bicubic ROI, (c) RCAN, (d) FIF SR, (e) FIFNET, and (f) EWF-FIFNET. The multiframe methods use $K = 60$ frames for (d)-(f).

(a) Bicubic

(b) Bicubic ROI

(c) RCAN

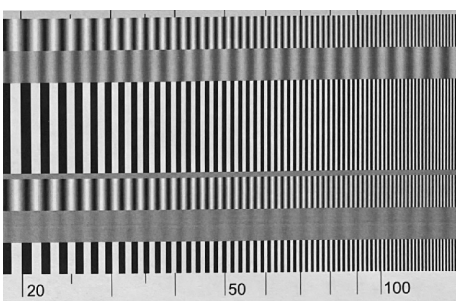
(d) FIF

(e) FIFNET

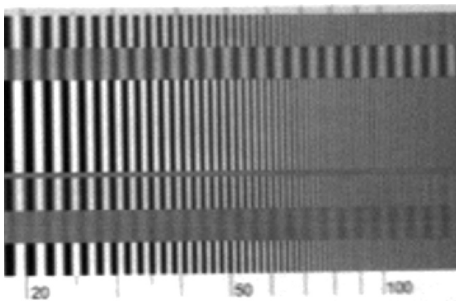
(f) EEEWF-FIFNET

(g) IEEWF-FIFNET

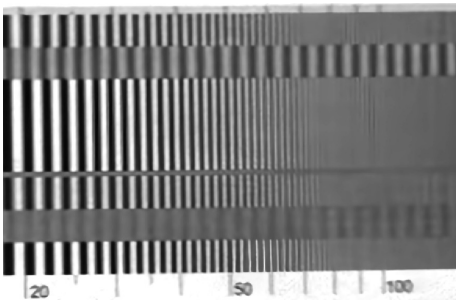
Figure 5.16: Image results for the real camera data of a bookshelf. The images shown are (a) bicubic, (b) bicubic ROI, (c) RCAN, (d) FIF SR, (e) FIFNET (f) IEWF-FIFNET, and (g) EEWF-FIFNET. The multiframe methods use $K = 60$ frames for (d)-(g).



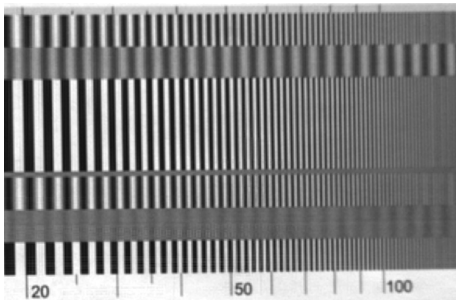
(a) iPhone Pro Max 14



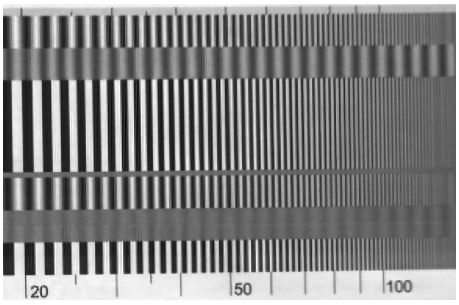
(b) Bicubic



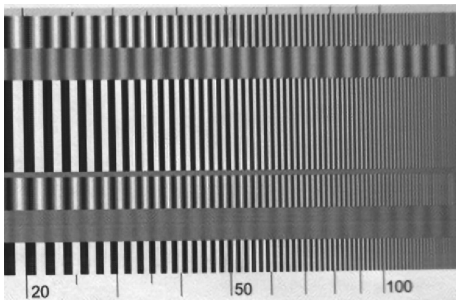
(c) RCAN



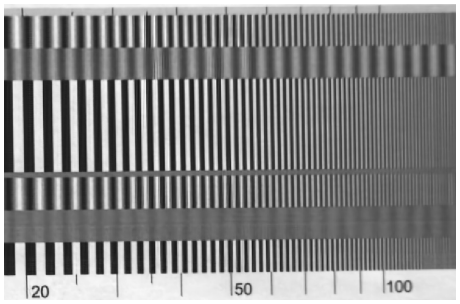
(d) FIF



(e) FIFNET

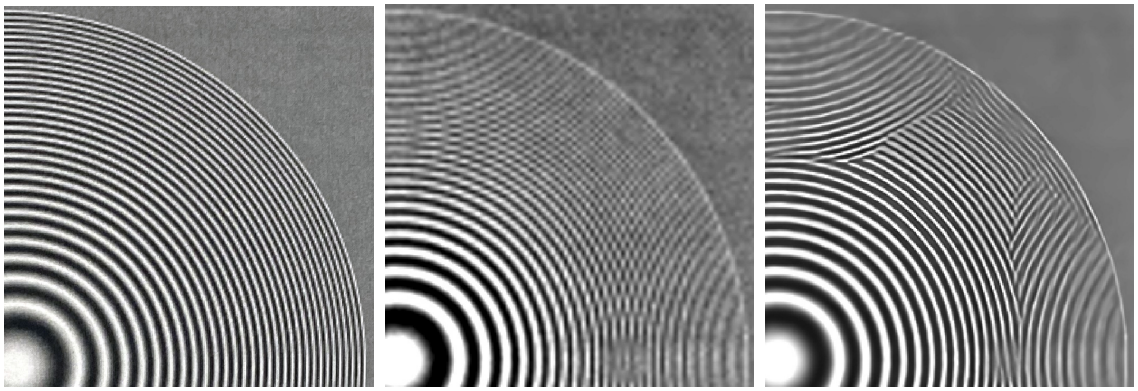


(f) EEWF-FIFNET

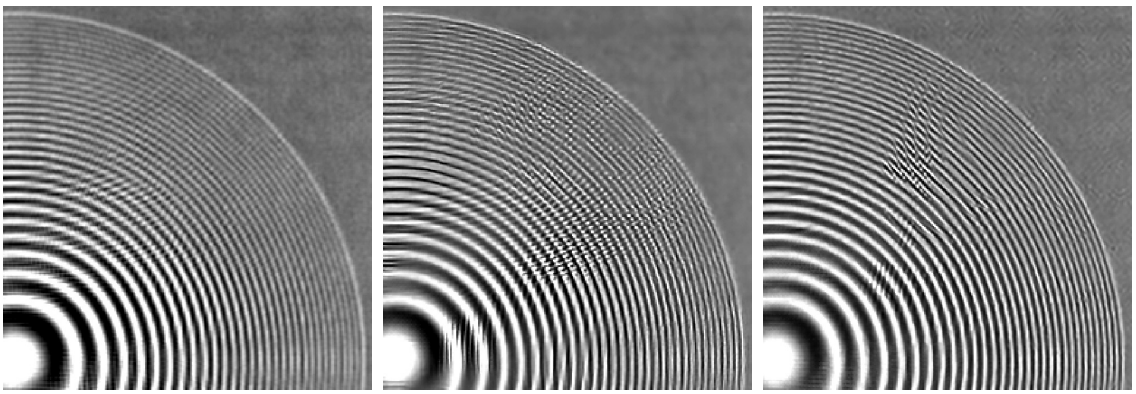


(g) IEWF-FIFNET

Figure 5.17: Image results for the real camera data of a line pair. The images shown are (a) iPhone Pro Max 14, (b) bicubic, (c) RCAN, (d) FIF SR, (e) FIFNET (f) IEWF-FIFNET, and (g) EEWF-FIFNET. The multiframe methods use $K = 60$ frames for (d)-(g).



(a) iPhone Pro Max 14 (b) Bicubic (c) RCAN



(d) FIF (e) FIFNET (f) EEWF-FIFNET

(g) IEWF-FIFNET

Figure 5.18: Image results for the real camera data of a chirp pattern scene. The images shown are (a) iPhone Pro Max 14, (b) bicubic, (c) RCAN, (d) FIF SR, (e) FIFNET, (f) IEWF-FIFNET, and (g) EEWF-FIFNET. The multiframe methods use $K = 60$ frames for (d)-(g).

CHAPTER VI

CONCLUSIONS

Our proposed approach, FIFNET, is a motion-based multiframe CNN SR method that fuses and restores multiple interpolated input frames using a single CNN architecture. This approach is the first of its kind to perform fusion of input images within one CNN. The inclusion of subpixel registration information as auxiliary information channels that accompany the input frames, as described in Section 3.1.1, is a key innovation that makes this method effective. The quantitative results in Section 5.1 demonstrate a significant boost in performance achieved by including these extra channels.

Compared with the VDSR architecture, FIFNET concatenates the input frames near the end of the convolution layers, which boosts performance and allows for fewer total layers. Another notable architectural innovation is the elimination of zero padding inside the network to reduce the network size for training and prevent extrapolation error from potentially contaminating the training process.

Our experimental results demonstrate that FIFNET provides the best performance among the methods tested for multiple frames (i.e., $K > 1$) and exhibits a consistent performance advantage over the original FIF SR method. This is attributed to the highly flexible non-linear processing of the CNN and the synergy achieved by jointly performing fusion and restoration. Subjective results with both simulated and real data are consistent with the quantitative results.

We also present the Exponential Weighted Externally Fusion (EEWF-FIFNET) and the Exponential Weighted Internally Fusion (IEWF-FIFNET) models that extend the FIFNET model. The EEWF-FIFNET model addresses the limitations of the FIFNET model when

dealing with a large number of frames ($K > 10$), resulting in improved overall performance. The IEWF-FIFNET model introduces a custom convolution layer called the Exponential Weighted Fusion (EWF) layer, which applies exponential weights to each frame based on their temporal distance from the current frame being processed, and combines the weighted frames using a convolution operation to produce fused frames. This process captures temporal dependencies between frames and improves the network's performance on image recognition tasks.

Our experimental results show that both the EEWF-FIFNET and IEWF-FIFNET models achieve the best performance among the tested methods when multiple frames are used ($i.e., K > 1$). The subjective results from simulated data agree with the quantitative results, and our models outperform the benchmark in terms of both PSNR and visual quality. We also evaluate our models using real data captured directly from the camera and compare their performance against the benchmark.

Notably, we train FIFNET, EEWF-FIFNET, and IEWF-FIFNET using a realistic observation model that accounts for camera optics and the corresponding detector array. This allows us to more realistically evaluate the methods on simulated data and apply our trained models directly to real camera data. Overall, our approach provides a significant improvement over previous methods and offers a more realistic and efficient solution to the problem of image super-resolution.

BIBLIOGRAPHY

[1] B. K. Karch and R. C. Hardie, "Robust super-resolution by fusion of interpolated frames for color and grayscale images," *Frontiers in Physics*, vol. 3, p. 28, 2015.

[2] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1646–1654.

[3] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 07 2018, pp. 286–301.

[4] Z. Ma, R. Liao, X. Tao, L. Xu, J. Jia, and E. Wu, "Handling motion blur in multi-frame super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5224–5232.

[5] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[6] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[7] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 624–632.

[8] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

[9] M. S. Sajjadi, B. Scholkopf, and M. Hirsch, "Enhancenet: Single image super-resolution through automated texture synthesis," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4491–4500.

[10] A. Bulat and G. Tzimiropoulos, "Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 109–117.

[11] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 252–268.

[12] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, "A fully progressive approach to single-image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 864–873.

[13] M. Kawulok, P. Benecki, S. Piechaczek, K. Hrynczenko, D. Kostrzewa, and J. Nalepa, "Deep learning for multiple-image super-resolution," *arXiv preprint arXiv:1903.00440*, 2019.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[15] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Robust shift and add approach to superresolution," in *Applications of Digital Image Processing XXVI*, A. G. Tescher, Ed., vol. 5203, International Society for Optics and Photonics. SPIE, 2003, pp. 121 – 130.

[16] M. Kawulok, P. Benecki, D. Kostrzewa, and L. Skonieczny, "Evolving imaging model for super-resolution reconstruction," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '18. New York, NY, USA: ACM, 2018, pp. 284–285.

[17] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3262–3271.

[18] R. C. Hardie, M. Rucci, B. K. Karch, A. J. Dapore, D. R. Droege, and J. C. French, "Fusion of interpolated frames superresolution in the presence of atmospheric optical turbulence," *Optical Engineering*, vol. 58, no. 8, pp. 1 – 16, 2019.

[19] H. Elwarfalli and R. C. Hardie, "Fifnet: a convolutional neural network for motion-based multiframe super-resolution using fusion of interpolated frames," *Computer Vision and Image Understanding*, vol. 202, p. 103097, 2021.

[20] R. C. Hardie, D. R. Droege, A. J. Dapore, and M. E. Greiner, "Impact of detector-element active-area shape and fill factor on super-resolution," *Frontiers in Physics*, vol. 3, p. 31, 2015.

[21] R. C. Hardie, K. J. Barnard, and R. Ordonez, "Fast super-resolution with affine motion using an adaptive Wiener filter and its application to airborne imaging," *Opt. Express*, vol. 19, no. 27, pp. 26 208–26 231, Dec 2011.

[22] D. Flaute, R. C. Hardie, and H. Elwarfalli, "Resampling and super-resolution of hexagonally sampled images using deep learning," *Optical Engineering*, vol. 60, no. 10, p. 103105, 2021.

[23] R. Hardie, "A fast image super-resolution algorithm using an adaptive wiener filter," *IEEE Transactions on Image Processing*, vol. 16, pp. 2953–2964, 2007.

[24] R. C. Hardie, "A fast super-resolution algorithm using an adaptive wiener filter," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2953–2964, Dec 2007.

[25] H. E. Sankaran, A. Gotchev, and K. Egiazarian, "Efficient super-resolution reconstruction for translational motion using a near least squares resampling method," in *2006 International Conference on Image Processing*, 2006, pp. 1745–1748.

[26] F. Kara and C. Vural, "Exact image super-resolution for pure translational motion and shift-invariant blur," *International Journal of Computer and Information Engineering*, vol. 2, no. 5, pp. 1370–1373, 2008.

[27] R. C. Hardie, K. J. Barnard, and R. Ordonez, "Fast super-resolution with affine motion using an adaptive wiener filter and its application to airborne imaging," *Opt. Express*, vol. 19, no. 27, pp. 26 208–26 231, Dec 2011. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-19-27-26208

[28] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI'81: 7th international joint conference on Artificial intelligence*, vol. 2, 1981, pp. 674–679.

[29] J. Goodman, *Introduction to Fourier Optics*, 3rd ed. Roberts and Company Publishers, Dec. 2004.

[30] G. D. Boreman, *Basic electro-optics for electrical engineering*. SPIE Press, 1998, vol. 31.

[31] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, May 2003.

[32] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence, Vancouver*, Aug. 1981.

[33] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 126–135.

[34] R. Timofte, S. Gu, J. Wu, L. Van Gool, L. Zhang, M.-H. Yang, M. Haris *et al.*, "Ntire 2018 challenge on single image super-resolution: Methods and results," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018, pp. 965–96 511.

[35] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, pp. 898–916, 2010.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, pp. 600–612, 2004.

[37] P. Milanfar, *Super-Resolution Imaging*. CRC Press, 2017.