ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Electronics and Telecommunication Engineering (33$^{\text{rd}}$ cycle)

Deep learning for inverse problems in remote sensing: super-resolution and SAR despeckling

By

Andrea Bordone Molini

* * * * * *

Supervisors
Prof. Enrico Magli, Supervisor
Prof. Marco Mellia. Co-supervisor

Doctoral Examination Committee:
Prof. Florence Tupin, Referee, Télécom ParisTech, Paris
Prof. Francesca Bovolo, Referee, Fondazione Bruno Kessler, Trento
Prof. Tatiana Tommasi, Politecnico di Torino, Torino
Prof. Tiziano Bianchi, Politecnico di Torino, Torino
Prof. Giuseppe Scarpa, Università degli studi di Napoli “Federico II”, Napoli

Politecnico di Torino
April 22, 2021

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

....

Andrea Bordone Molini
Turin, April 22, 2021

Summary

In this thesis we present novel deep learning methods to tackle two inverse problems in imaging i.e., super-resolution and denoising. These enhancement tasks are often used as a pre-processing step by many pattern recognition and analysis algorithms as they can leverage an image reconstruction with enriched spatial information and details that eases image understanding, thereby improving their performance.

Recently, convolutional neural networks (CNN) have been successfully applied to many remote sensing problems. However, deep learning techniques for multi-image super-resolution from multitemporal unregistered imagery have received little attention so far. In the first part of this thesis we propose a novel CNN-based technique that exploits both spatial and temporal correlations to combine multiple images. This novel framework integrates the spatial registration task directly inside the CNN, and allows to exploit the representation learning capabilities of the network to enhance registration accuracy. The entire super-resolution process relies on a single CNN with three main stages: shared 2D convolutions to extract high-dimensional features from the input images; a subnetwork proposing registration filters derived from the high-dimensional feature representations; 3D convolutions for slow fusion of the features from multiple images. The whole network is trained end-to-end to recover a single high resolution image from multiple unregistered low resolution images. As opposed to the vast majority of the work in literature that use synthetic datasets, the training procedure is carried out through a set of real-world low resolution observations and the corresponding high resolution image for the same scene, captured from the same platform. This method is the winner of the PROBA-V super-resolution challenge issued by the European Space Agency.

The second contribution of this thesis is a deep learning method to tackle a denoising task in the field of synthetic aperture radar (SAR) remote sensing. Information extraction from SAR images is heavily impaired by speckle noise, hence despeckling is a crucial preliminary step in scene analysis algorithms. The recent success of deep learning envisions a new generation of despeckling techniques that could outperform classical model-based methods. However, current deep learning approaches to despeckling require supervision for training, whereas clean SAR images are impossible to obtain. In the literature, this issue is tackled by resorting to either synthetically speckled optical images, which exhibit different properties with respect to true SAR images, or multi-temporal SAR images, which are difficult to acquire or fuse accurately. In this paper, inspired by recent works on blind-spot denoising networks, we propose a self-supervised Bayesian despeckling method. The proposed method is trained employing only noisy SAR images and can therefore learn features of real SAR images rather than synthetic data. Experiments show that the performance of the proposed approach is very close to the supervised training approach on synthetic data and superior on real data in both quantitative and visual assessments.

Acknowledgements

I wish to express my sincere gratitude to my thesis advisor, Prof. Enrico Magli, for his support, understanding and guidance during all these years. I am extremely grateful for his assistance, suggestions and for giving me the opportunity of joining the IPL group. I wish also to thank Diego Valsesia and Giulia Fracastoro who helped me during these years and allowed me to grow in this field. I also thank them for having the patience of always listening to my concerns.

A huge thanks goes to all my friends and colleagues from the department for their valuable advices and for their continuous support. You have been a precious help, always pushing me to keep a positive attitude.

A special thank goes to Andrea, a former Ph.D student, whose support and friendship during my first year have been vital to get me through some tough times.

I would also like to thank my officemate Nicola for putting up with me and always giving me words of encouragement and advice.

A special thanks goes to Asle. You have always been an essential part of my life with your unconditional support. I am truly lucky to have you as a friend.

Finally, I am also deeply grateful to my wife Roberta. You accepted to live our marriage far apart for many years to allow me to fulfill my dreams. Over these years I have lived some difficult moments in which I have thought to drop everything and come back to you. This thought was actually what gave me the strength to get to the end of this.

To Debby

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Inverse problems are very popular in mathematics and widely present in a large number of applications in engineering. When imaging/sensing technologies are involved we talk about inverse imaging problems such as denoising, deconvolution, deblurring, inpainting, superresolution, and medical image reconstruction. In certain applications this enhancement process is needed because of the low quality observations delivered by the imaging systems during image formation. In computational imaging, solving an inverse problem means exploiting observations produced by a forward process to reconstruct the desirable continuous image x_c. However, in practice, what we are looking for is a discrete sampled approximation of x_c which is denoted by x. The forward model that generates the observations has typically the following forms:

$$y = \mathcal{A}(x) + n \tag{1.1}$$

where $y \in \mathbb{R}^{W \times H}$ is the observed signal (image), $x \in \mathbb{R}^{W' \times H'}$ represents the unknown input image, $\mathcal{A}$ is a possibly non linear forward operator and n models the noise in the observed data. Depending on the problem at hand the forward mapping $\mathcal{A}$ can take different form. More in general, in case the noise is not additive:

$$y = \mathcal{N}(\mathcal{A}(x)) \tag{1.2}$$

where $\mathcal{N}$ is the noise operator, sampling from a noisy distribution.

Among the analytical methods, the adopted approach is to solve an optimization problem to recover the original image x from the observations y as follows:

$$\arg\min_{x} E(\mathcal{A}(x), y)$$

where E is the energy function which measures how well the reconstructed image matches the observations. In almost all practically relevant applications there are many possible solutions given the observations, and the underlying continuous problem is ill-posed. Due to the ill-posedness of these problems a regularized formulation is used to exploit prior knowledge about images in order to promote solutions that match the prior knowledge of x to obtain stable and faithful reconstructions:

$$\arg\min_{x} E(\mathcal{A}(x), y) + R(x)$$

Usually the forward model is explicitly handcrafted through domain knowledge, so E, R and the minimization algorithm need to be designed for each different application. Recently, the computer vision research community has had great success in replacing the explicit modeling of energy function $E(\mathcal{A}(x), y)$ with a parameterized function that directly maps the measurements y to a solution $\hat{x} = f(\theta, y)$. In situations in which $\mathcal{A}$ is not exactly known and/or it cannot be precisely modeled mathematically and/or it consists of the concatenation of multiple operators, there is room for deep-learning techniques to learn from training data the parameter vector θ corresponding to models of $\mathcal{A}$and R; this typically outperforms analytical approaches in terms of reconstruction accuracy. On the other hand, even when the forward model is exactly known, selecting a regularizing function can be difficult when we do not have prior knowledge about the distribution of the original signal x. In imaging literature many priors have been proposed, and they typically correspond to some type of high-pass regularizer $R(x)$ to impose a smoothness constraint, suggesting that most images are naturally smooth with limited high-frequency activity, and therefore it is appropriate to minimize the amount of high-pass energy in the restored image.

1.1 Deep learning in inverse problems

Recent work in deep learning has demonstrated that deep neural networks can leverage large collections of training data to directly compute regularized reconstructions, by performing a supervised inversion of the forward model.

$$\arg\min_{\theta} \sum_{i=1}^{N} E(x^{(i)}, f(\theta, y^{(i)})) + R(\theta)$$

where E is the energy/loss function $E : W' \times H' \to \mathbb{R}^+$, $f(\theta, y)$ is a parameterized mapping function, θ is the set of all possible parameters, $R(\theta)$ is a regularizer $R : \theta \to \mathbb{R}^+$ on the parameters to avoid overfitting the often limited amount of training data. An additional term representing the regularization on $f(\theta, y^{(i)})$ can be used to further promote regularized reconstructions.

Deep networks parameterize $f(\theta, y)$ using several layers of linear operations followed by non-linearities. The free parameters of $f(\theta, y)$ are learned by using large amounts of training data and fitting the parameters to the ground truth data via a large-scale optimization problem.

While data-driven based methods yield powerful representations, they require a training dataset and the training procedure is often difficult. The amount of data to acquire in order to achieve generalization to test data is huge.

In many cases generating training data is straightforward because the forward model we aim to invert is known exactly and easily computable. In denoising, the forward model is the identity and training data are generated by corrupting images with noise; the noisy image then serves as training input and the clean image as the training output. Super-resolution (SR) follows the same pattern, where training pairs are easily generated by downsampling and degrading the high resolution image to generate a low resolution version. The same happens for deblurring.

Often, knowing exactly the sensor imaging model is not feasible and an approximation is used to artificially generate the data set. Many of the works found in the SR literature are based on simulated data, where low-resolution (LR) observations for a specific scene are obtained through a degradation and down-sampling process of the high-resolution (HR) images by assuming a specific sensor imaging model. This is a simplified scenario and a too simple degradation model may not accurately match the real one. The deep network reconstruction ability will be highly dependent on the choice of $\mathcal{A}$ used to create the dataset, since the network may end up learning the inversion of a forward model that does not match the real one.

In this thesis we avoid assuming a specific sensor imaging model by solving a SR problem that employs real images of the same scene for both the low and high resolutions. This enables data-driven methods to learn the inversion of possibly complex degradation models. We will focus on a class of SR problems called multi-image SR (MISR) problems on satellite images. When multiple LR images for a same scene are involved, the forward model gets more complicated embedding also a geometric registration operator. Little work has been done on deep learning MISR methods in the context of remote sensing, which poses specific challenges such as environmental conditions and the complex statistics of remote sensing imagery.

In the last years, deep learning based methods have been proposed to solve MISR problems in context of video super-resolution [75, 20]. Most of these works are composed of two steps: a motion estimation and compensation procedure followed by an upsampling process, heavily relying on the initial motion estimation. In order to reduce the effect of registration errors, we came up with a method to simultaneously estimate the motion parameters and reconstruct the SR image, all within an end-to-end trainable CNN, where the two tasks are optimized jointly. The forward model is entirely learnt during training without relying on any knowledge of the motion parameters.

The second objective of this thesis is again strictly related to the lack of a complete and real training dataset where the observed image y and the ground truth image x are both available. There are some applications that focus on reconstructing from real measurements, while the corresponding ground truth is not known and practically impossible to acquire. Synthetic Aperture Radar (SAR) is a coherent imaging system and as such it strongly suffers from the presence of speckle, a signal dependent granular noise. Speckle noise makes SAR images difficult to interpret, preventing the effectiveness of scene analysis algorithms for, e.g., image segmentation, detection and recognition. Despeckling aims to remove the speckle noise from the observed noisy images. Current deep learning approaches to despeckling require supervision for training, whereas clean SAR images are impossible to obtain. In the literature, this issue is tackled by resorting to either synthetically speckled optical images, which exhibit different properties with respect to unknown underlying clean SAR images. There is an emerging paradigm in deep learning that consists of learning directly from noisy images in a self-supervised fashion. In the second part of this thesis we present a self-supervised Bayesian despeckling framework that enables direct training on real SAR images. Our method bypasses the problem of training a CNN on synthetically-speckled optical images, thus avoiding any domain gap and enabling learning of features from real SAR images.

1.2 Thesis organization

The reminder of this thesis is organized as follows.

In Chapter 2 we give a background about deep neural networks and present an overview of the main deep learning methods for inverse problems in imaging. We introduce in more details the two inverse problems we will focus throughout the thesis and the main motivations.

Chapter 3 describes a novel CNN architecture to solve a MISR problem with real multitemporal and unregistered images called DeepSUM [120].

In Chapter 4, we propose an improvement over DeepSUM [15], by exploiting non locality in the CNN architecture. We improve the feature extraction process of DeepSUM with graph convolutional layers to take into account spatially distant pixels in the computation.

We then move to despeckling in Chapter 5. In this chapter we present a self-supervised Bayesian despeckling method, called Speckle2Void [121], using a new class of convolutional networks for denoising, called blind-spot denoising networks, that does not require ground truth data to train. In Chapter **??** we draw some conclusions and outline open issues.

1.3 Publications

In this section we gather a list of publications representing the outcome of the research carried out during the PhD program.

1. A. B. Molini, D. Valsesia, G. Fracastoro, and E. Magli. “Deep Learning For Super-Resolution Of Unregistered Multi-Temporal Satellite Images.” *2019*

10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS). Sept. 2019, pp. 1–5.

2. A. B. Molini, D. Valsesia, G. Fracastoro, and E. Magli, "DeepSUM: Deep Neural Network for Super-Resolution of Unregistered Multitemporal Images." *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2019.

3. A. B. Molini, D. Valsesia, G. Fracastoro, and E. Magli, "Towards Deep Unsupervised SAR Despeckling with Blind-Spot Convolutional Neural Networks." in *2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Oct 2020.

4. A. B. Molini, D. Valsesia, G. Fracastoro, and E. Magli, "DeepSUM++: Nonlocal Deep Neural Network for Super-Resolution of Unregistered Multitemporal Images." in *2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Oct 2020.

5. A. B. Molini, D. Valsesia, G. Fracastoro, and E. Magli, "Speckle2Void: Deep Self-Supervised SAR Despeckling with Blind-Spot Convolutional Neural Networks." *submitted to* IEEE Transactions on Geoscience and Remote Sensing.

Chapter 2

Background

2.1 Deep learning background

The impact of machine learning in remote sensing image analysis has been of paramount importance, since the first application of such methods in the late 1990s. Conventional machine learning approaches performed the extraction of appropriate hand-crafted features and then applied shallow classification/regression techniques. This went on until a few years ago when deep neural networks started to be employed in the remote sensing domain.

2.1.1 Neural networks

A neural network is composed of a set of neurons with different sets of weights that are used to process an input. Each neuron computes the cross correlation between an input vector and a vector of weights. Multiple neurons constitute the so-called layer of a fully-connected neural network and a series of stacked layers is referred as multilayer perceptron. Fig. 2.1 depicts an example of a fully-connected neural network.

Each layer takes the output of the previous layer and applies an affine transformation followed by a non-linear element-wise function, generating a new representation of the input.

$$h^{l+1} = f_{nl}(W_{l+1}^T h^l + b)$$

where $h^{l+1} \in \mathbb{R}^n$ is the new representation at the output of the layer $l+1$, $h^l \in \mathbb{R}^m$ is the input to layer $l+1$, $W \in \mathbb{R}^{m \times n}$ is the weight matrix, b is the bias vector and f_{nl} is the non-linear function. A series of affine transformations interleaved by non-linearities allows the network to find neat representations for complex, highly folded data manifolds.

The choice of the neural network architecture determines the generic set of possible functions $f(\theta, y)$ parametrized by the values of the weights, and these weights

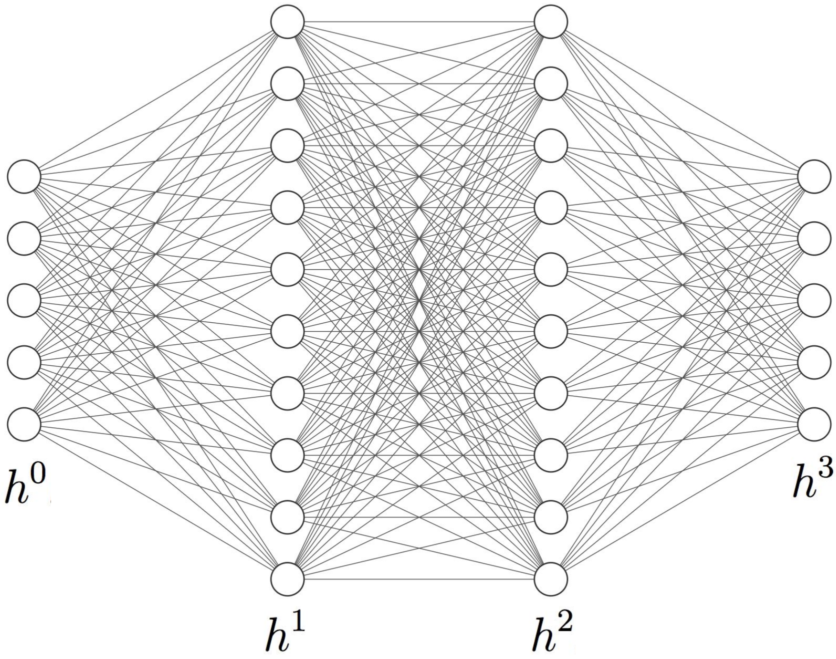

Figure 2.1: Fully-connected neural network.

are the parameters over which the minimization occurs.

2.1.2 Convolutional neural networks

A fully-connected neural network can have a lot of weights, especially when the input is high dimensional, and hence be very computational expensive. Moreover, these networks do not make any assumptions on the properties of the input. However, when dealing with highly structured modalities such as 2D or 3D imagery, some properties of the natural signals can be exploited:

- locality: short-range dependencies capture most of the information;
- stationarity: statistical properties do not change over time/space;
- compositionality: complex features can be created by hierarchically assembling simple and local features.

If data exhibits locality, each neuron needs to be connected to only few local neurons of the previous layer, by dropping connections between far away neurons. When dropping connections, each layer does not work the whole input at once, but the overall architecture will be able to account for the whole input, by stacking more layers allows to exploit compositionality. If data exhibits stationarity, a small set of parameters can be used multiple times across the input by performing a convolution with the weights. After applying sparsity and stationarity the weights of a neural networks represent a convolution kernel and the architecture takes the name of convolutional neural network (CNN). Each convolutional layer is composed by a set of convolution kernels. Each convolution kernel slides across the input signal by a specific step size, called stride, that is often set to 1.

Convolution operation

The dot product operation between the matrix W_l at layer l and the input $h^{(l)} \in \mathbb{R}^m$ in a fully-connected neural network has the following form:

$$\begin{bmatrix} w^l_{11} & w^l_{12} & w^l_{13} & w^l_{14} & \dots & w^l_{1k} & \dots & w^l_{1n} \\ w^l_{21} & w^l_{22} & w^l_{23} & w^l_{24} & \dots & w^l_{2k} & \dots & w^l_{2n} \\ w^l_{31} & w^l_{32} & w^l_{33} & w^l_{34} & \dots & w^l_{3k} & \dots & w^l_{3n} \\ w^l_{41} & w^l_{42} & w^l_{43} & w^l_{44} & \dots & w^l_{4k} & \dots & w^l_{4n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w^l_{m1} & w^l_{m2} & w^l_{m3} & w^l_{m4} & \dots & w^l_{mk} & \dots & w^l_{mn} \end{bmatrix} \begin{bmatrix} h^l_1 \\ h^l_2 \\ h^l_3 \\ h^l_4 \\ \vdots \\ h^l_j \\ \vdots \\ h^l_n \end{bmatrix} = \begin{bmatrix} h^{l+1}_1 \\ h^{l+1}_2 \\ h^{l+1}_3 \\ h^{l+1}_4 \\ \vdots \\ h^{l+1}_j \\ \vdots \\ h^{l+1}_m \end{bmatrix}$$

In a convolution neural network this dot product can be represented with a Toeplitz matrix as follows:

$$\begin{bmatrix} w^{l,k}_1 & w^{l,k}_2 & w^{l,k}_3 & 0 & \dots & 0 & \dots & 0 \\ 0 & w^{l,k}_1 & w^{l,k}_2 & w^{l,k}_3 & \dots & 0 & \dots & 0 \\ 0 & 0 & w^{l,k}_1 & w^{l,k}_2 & w^{l,k}_3 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} h^l_1 \\ h^l_2 \\ h^l_3 \\ h^l_4 \\ \vdots \\ h^l_j \\ \vdots \\ h^l_n \end{bmatrix} = \begin{bmatrix} h^{l+1,k}_1 \\ h^{l+1,k}_2 \\ h^{l+1,k}_3 \\ h^{l+1,k}_4 \\ \vdots \\ h^{l+1,k}_j \\ \vdots \\ h^{l+1,k}_n \end{bmatrix}$$

In this representation of the convolution operation, the kernel size is $3 \times 1 \times f_{in}$, the stride is 1 and $h^l \in \mathbb{R}^{1 \times n \times f_{in}}$ is the 1D input at layer l, where n is the length of the input signal and f_{in} is the number of input features. As we can observe, lots of weights are dropped and the few remaining parameters are re-applied across the entire signal. For this reason, a certain number of weight matrices are applied to the same input signal h^l. In the depicted operation, the k^{th} weight matrix represents the sliding kernel across the input a^l that produces the k^{th} output vector $h^{l+1,k} \in \mathbb{R}^{1 \times n}$. Each output vector captures a particular characteristic of the input and all together they form the output feature map $h^{l+1} \in \mathbb{R}^{1 \times n \times f_{out}}$, where f_{out} is the number of output features. The output feature map $h^{l+1} \in \mathbb{R}^{1 \times n \times f_{out}}$ in the example, has the same length n as the input $h^l \in \mathbb{R}^{1 \times n \times f_{in}}$ since we considered an implicit padding applied to the input signal that depends on the size of the kernel.

For 2D image data, the convolution kernels are 2-dimensional and are typically of spatial size 3 × 3, 5 × 5 and 7 × 7. These kernels cover a small portion of the input at each layer, but as multiple layers are stacked, the spatial extent the network can sense becomes larger. This is called receptive field and it increases layer by layer allowing the network to consider increasingly larger portions of the input. When dealing with imaging inverse problems, the most used kernel is the 3 × 3 one as each layer needs only to focus on local correlations of the immediate neighboring pixels.

Non local convolution operation

The classic convolution operation exploits only local correlations. In some applications involving inverse imaging problems, novel works were proposed to leverage self-similarity among structures in the signal beyond the local neighborhood. A number of non local convolution layers have been proposed in the last years, in order to gather information from similar but spatially distant data patches. The general idea can be formulated through a Toeplitz matrix with 1D signal as follows:

$$\begin{bmatrix} \phi_1^{l,k}(h_1^l, h_1^l) & \phi_2^{l,k}(h_1^l, h_2^l) & \dots & \phi_L^{l,k}(h_1^l, h_L^l) & 0 & \dots & 0 \\ 0 & \phi_1^{l,k}(h_2^l, h_2^l) & \phi_2^{l,k}(h_2^l, h_3^l) & \dots & \phi_L^{l,k}(h_2^l, a_{L+1}^l) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} h_1^l \\ h_2^l \\ h_3^l \\ h_4^l \\ \vdots \\ h_j^l \\ \vdots \\ h_n^l \end{bmatrix} = \begin{bmatrix} h_1^{l+1,k} \\ h_2^{l+1,k} \\ h_3^{l+1,k} \\ h_4^{l+1,k} \\ \vdots \\ h_j^{l+1,k} \\ \vdots \\ h_n^{l+1,k} \end{bmatrix}$$

where $\phi^{l,k}(h_i^l, h_j^l)$ is a function that depends on learnable weights, measuring the correlation between the features at location i and the features at location j, and L is the length of the kernel. In this way, close and distant components in the input signal are combined through weights that are dynamically computed based on the distance between the features. There are two main definitions of non-local convolutional layer: NLRN [105] and graph convolution [165]. In the first definition, $\phi_1^{l,k}(h_i^l, h_i^l)$ depends on a set of weights and all the other $\phi_z^{l,k}(h_i^l, h_j^l)$ on another set of weights, with $z = 2, \dots, L$. In the second definition, the kernel of length L can dynamically extend across the input signal as it is based on the construction of a graph of neighbors. This implies that some of the $\phi_z^{l,k}(h_i^l, h_j^l)$ can be set to zero.

In Chapter 4 we employ the definition of a non-local convolutional layer based on a K-nearest neighbor graph [165] to collect non-local information while solving a super-resolution task. In Chapter 5 NLRN [105] non-local layer is briefly described and used in a despeckling task.

Activation function

The feature map, produced by the convolution operation and the bias, is the result of an affine transformation of the input. A non-linear function is applied to the feature map at each convolutional layer to access a richer hypothesis space. The most popular activations are:

- Sigmoid or logistic function with output ranging from 0 to 1: $\frac{1}{1+e^{-x}}$;
- ReLU function, defined as the positive part of its argument: $\max(0, x)$;
- Leaky ReLU function, allowing a small gradient even when the neuron is not active: $\max(0.1x, x)$.

Batch normalization

A very important component in CNN is the batch normalization layer [67], usually placed between the convolution operation and activation function. It works similarly to the initial normalization operated per channel on the dataset before training. Since the transformation performed by each layer does not guarantee the output to have still zero mean and unit variance, this layer adaptively normalize data by internally maintaining an exponential moving average of the batch-wise mean and variance of the data seen during training. This mitigate the so called Internal Covariate Shift, preventing the activations of intermediate layers to diverge from desirable values and avoiding saturation. The main advantage is to allow a better gradient propagation during optimization, affording deeper networks.

Pooling layers

Pooling layer is another component that makes the receptive field increase. This layer is used to reduce the spatial size of the feature maps and ensure invariance to translation. Two examples are the max pooling and the average pooling layers that are employed to propagate to subsequent layers local non-learned aggregations across the whole feature map. This dimensionality reduction is typically performed in discriminative tasks, where the final output represents a low-dimensional label. When a task involves image generation for reconstruction, like in inverse imaging problems, a spatial reduction of the feature map may be detrimental.

2.1.3 Training and optimization

Cost function

Once we choose a suitable CNN architecture, the training takes place by minimizing an objective function to fit the dataset $(x^{(i)}, y^{(i)})_{i=1}^{N}$ in order to identify the optimal values for the weights θ, where N is the dataset size. Taking this from a statistical viewpoint, training a supervised CNN corresponds to maximizing the conditional likelihood of each training target $x^{(i)}$ given the corresponding input $y^{(i)}$:

$$\max_{\theta} \prod_{i=1}^{N} p(x^{(i)}|y^{(i)}, \theta),$$

Usually all the $y^{(i)}$ are considered independent and identically distributed and the CNN is trained by minimizing the negative log likelihood as follows:

$$\min_{\theta} \sum_{i=1}^{N} -\log(p(x^{(i)}|y^{(i)}, \theta),$$

The approach most commonly employed when using CNN for solving inverse problems in imaging is to consider $x^{(i)}|y^{(i)}$ to be distributed as a multivariate Gaussian $\mathcal{N}(f(y^{(i)}, \theta), \Sigma)$ where $f(y^{(i)}, \theta)$ represents the CNN and Σ represents the covariance matrix of form $\sigma^2 I$. The final cost function is the Euclidian distance between the target $x^{(i)}$ and its estimated mean $f(y^{(i)}, \theta)$:

$$\min_{\theta} \sum_{i=1}^{N} \parallel x^{(i)} - f(y^{(i)}, \theta) \parallel^2, \tag{2.1}$$

The maximum likelihood formulation does not take into account prior information about the desired CNN parameters θ. Maximum a posteriori formulation consists in maximizing the unnormalized posterior:

$$\max_{\theta} \prod_{i=1}^{N} p(x^{(i)}, \theta|y^{(i)}),$$

Considering the negative log of the unnormalized posterior we obtain:

$$\min_{\theta} -\sum_{i=1}^{N} \log(p(x^{(i)}|y^{(i)}, \theta)) + \log(p(\theta)),$$

Depending on the assumption on the probability distribution used to model the parameters, we get a different kind of regularization. Regularization is very important to prevent the CNN to overly specialize to the training examples and to increase its generalization ability on new data. Another method for regularization is dropout [149], where individual neurons of the network are randomly deleted during training. This forces the network to learn alternative paths for predicting the correct output, leading to greater flexibility.

Optimization algorithms

Once designed the right loss function for the dataset at hand, this function is minimized updating the CNN parameters through an optimization algorithm. The most common optimizer is called stochastic gradient descent (SGD). At each step of the training procedure, the gradient of the loss function with respect all the weights is computed on a random batch of the dataset and used to update the model parameters θ. The gradient computation is approximated on a subset of the full dataset because otherwise this process would create speed and memory problems. Amongst the most common variations on gradient descent there are Adam [84], AdaDelta [187] and RMSProp, accounting for gradients from previous iterations.

2.2 Deep learning for inverse problems in imaging

Early deep learning based works to solve inverse problems in imaging made use of fully connected neural networks to learn a mapping from an observation y to its reconstruction $\hat{x}$. Despite the fact that these models were simple, they showed competing performance with respect to the state-of-the-art analytical methods.

Burger et al. [19] solved a denoising inverse problem using a fully connected neural network to map noisy images to their cleaned versions, while the authors in [199] learned an end-to-end mapping in the wavelet domain. A specific architecture, called auto-encoder, has been used in denoising and inpainting problems in [175] and in [3] with sparsity. A fully connected neural network was trained on a large dataset of artificially blurred images by Schuler et al. [138] to solve a non-blind deconvolution problem.

2.2.1 CNN-based methods

CNNs are particularly suitable for processing images as they can easily extract the statistics of their input and make use of them to solve inverse problems. There are two major types of CNN-based architectures used for solving inverse problems in imaging. The first type is composed of multiple convolutional layers that produce feature maps of same spatial size as the input image. The other type of CNNs are called encoding-decoding architecture, where the spatial size of the feature maps first decreases and then increases again to match the output image size. When solving a regression inverse problem it is desirable avoid reducing the spatial size as it can have a destructive effect. Some spatial details can be lost during encoder compression and this may lead to a significant loss of detail in the output image. However there are some inverse problems that can use a larger receptive field in order to extract semantic information regarding the input such as optical flow inpainting problem. Most of the works in inverse problem literature make use of the first approach. A five-layer CNN is used in [70] to denoise an image injected with Gaussian noise. The method in [87] solves a compressed sensing inverse problem with a six-layer CNN to map the compressed measurements of an image to its full reconstruction. An off-the-shelf denoiser is employed to remove blocky artifacts and obtain the final reconstructed image. One of the first work in SR task using deep learning was conducted by Dong et al. [34] who used a three-layer CNN that takes an interpolated LR patch as input to produce the corresponding HR version. This network has been used and extended in many subsequent works such as in [75] where the authors applied SR to video. In this case, multiple frames are combined to reconstruct a HR image. These frames are first motion-compensated as a pre-processing step, and then fed to separate CNNs. The individual extracted features corresponding to the input frames are fused together by another CNN acting to gather high frequency information from the different frames in feature space. Hradis et al. [61] trained a deep CNN with up to 15 layers to solve a blind deconvolution and denoising task. They employed a large set of text documents injected with a combination of realistic de-focus and camera shake blur kernels, outperforming the state-of-the-art analytical methods.

The introduction of new learning strategies such as more effective activation functions, batch normalization, weight initialization, architectural choices, allowed for training deeper networks. In [57] a novel architectural design choice was introduced, redefining the convolutional layers as learning residual functions. The residual blocks were crucial to train very deep neural networks. A representation of the traditional residual block is depicted in Fig. 2.2. Residual blocks learn a residual between two or more layers by adding a skip connection from the input of the residual block to its output. The task of learning the full mapping between input and output is much harder than learning the residual. For this reason, residual networks are more stable and easier to train as they are great at countering the vanishing gradient problem.

Ledig et al. [93] proposed a super-resolution generative adversarial network (SR-GAN) composed by a deep residual network (ResNet) with skip-connection and residual blocks. They were the first to couple a residual architecture with adversarial training to form a perceptual loss that combines mean squared error (MSE) loss both at pixel and feature space level. Some works addressing inverse problems started proposing a different type of connection skipping the whole network up to the output layer, instead of skipping some layers. Using this trick turned out to be very successful as it exploits the fact that for many inverse problems in imaging, input and output images share very similar content, making the input an optimal starting point solution.

In the context of image denoising, Zhang et al. [28] proposed to train a residual 17-layer CNN which directly predicts the noise in the observed image. As the noisy input image is connected to the output layer, the network needs only to estimate

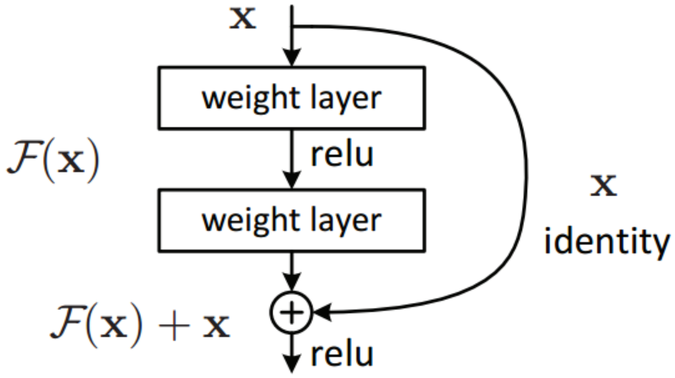

Figure 2.2: Residual block.

the noise to be subtracted to the noisy input image. This prevents the network from having to learn the reconstruction of the image content. The authors in [134] and [81] use this approach to train very deep CNN architectures for SR which learn to predict the missing high-frequency components from the LR patch instead of an entire new mapping function from the LR to the HR patch.

The previously described works use architectures where the spatial dimensions are fixed throughout the whole network. The encoder-decoder CNN is composed of a first network with convolution operations interleaved by downsampling operations and a second network upsampling feature maps back to the input spatial original size. The encoder learns an abstract representation of the input image, which is then used by the upsampling network to produce an output image. As already mentioned, the downsampling operations applied to solve an inverse problem are inappropriate as they could lead to a loss of information, but bring some advantages such as fewer operations performed by the network and a larger receptive field. Pathak et al. [128] showed that their encoder-decoder CNN can learn to reconstruct large missing regions from an input image. Other encoder-decoder CNN based methods solved the loss of information brought by the downsampling operations by inserting symmetric skip connections in the neural network between the lower downsampling convolutional layers of the network and the corresponding upper upsampling convolutional layer, which preserves the relevant details in the input image. This network is called U-Net and was first proposed for biomedical image segmentation in [133]. U-Net architecture became very popular in multiple imaging inverse problems such as denoising [113, 85, 89], super-resolution [62], computed tomography reconstruction [72], image inpainting [178] and optical flow [36] to directly predict optical flow from two input images. Most of the imaging inverse problems, are trained with MSE loss (2.1) as they are formulated as regression problems. MSE loss leads the network to predict the mean of the distribution, resulting in an averaged reconstruction among the multiple similar solutions an ill-posed inverse problem admits. MSE loss is used to achieve higher peak signal noise to the ratio (PSNR), one of the most used metric to assess quantitatively the quality of the estimated image:

$$\mathrm{PSNR} = 20 \log \frac{MAX(x)}{\parallel \hat{x} - x \parallel^2}$$

where x is the ground truth image, $\hat{x}$ is the estimate reconstruction and $MAX(x)$ is maximum possible value of x. Although PSNR is reliable from the content reconstruction fidelity side, it correlates poorly with the human perception of image quality.

Beyond MSE loss

Recent developments in the field of generative models paved the way for a new class of CNN-based generative approaches to inverse problems in imaging, able to approximate the complex density associated with natural image distributions. A generative adversarial network (GAN) [48] is a generative model that consists of two networks trained in competition. The generator tries to learn a mapping between training samples and random noise vectors z, while the discriminator attempts to distinguish between the output of the generator and real data. The generator aims to produce images $\hat{x} = g(z, \theta_{gen})$ that the discriminator cannot distinguish from the real images x. The classical approach used to learn a distribution of data $p(x)$ is MLE. This requires coming up with a parametrized model and fit its parameters. Moreover the chosen model might be too simple to describe the data as sophisticated model are too difficult to optimize or do not have a closed form expression. The GAN essentially replaces the parametric model, with another neural network that plays the role of a discriminator network that indirectly models the probability density of the real data. Many works in literature used the GAN architecture to regularize inverse problems through the adversarial loss, promoting the generated reconstructions to be close to the manifold of the real images. Unlike basic GAN, conditional GAN (cGAN) learns a mapping from the observed image y to the original image x. The generator is conditioned on the observed images instead of the noise, producing a reconstruction $\hat{x} = g(y, \theta_{gen})$. The role of the discriminator does not change in cGAN. In the context of SR, [134] and [93] used cGAN to couple the adversarial loss with the MSE loss. The adversarial loss has the effect of picking a particular mode from the distribution, resulting in much sharper reconstruction and more realistic details. The adversarial loss is not the only addition to the MSE loss. They both complemented it with a perceptual loss. This loss is used to measure the perceptual similarity between the estimated image and the original image, by computing the distance between two images in feature space. The perceptual loss l_{VGG} is usually defined as the Euclidean distance between the higher level feature map F, extracted from the well-known ImageNet pre-trained VGG19 network [146], of the estimated image $\hat{x}$ and the ground truth

x:

$$l_{VGG} = \| F_{VGG}(x) - F_{VGG}(y) \|^2, \tag{2.2}$$

This loss enforces the output images to be semantically and structurally compatible with respect to the original image.

In deblurring context, DeblurGAN method [88] solves a blind motion deblurring problem, using the more stable Wasserstein GAN (WGAN)[7] with the gradient penalty [50], instead of vanilla cGAN, and the perceptual loss in (2.2). In [137] a cGAN was used to solve deblurring on astrophysical images. Similar losses were adopted in the field of inpainting [128], super-resolution [200, 171, 73], denoising [183] and compressed sensing [14].

While the approches based on the adversarial and perceptual losses produce visually pleasing results, they tend to hallucinate information, resulting in lower PSNR scores and less reliable products in the context of remote sensing. The works dealing with remote sensing imagery tends to stick with pixel-level losses.

2.2.2 CNN-based methods in remote sensing

In this section we report briefly the most representative CNN-based inverse problems in remote sensing field.

The deep learning paradigm gained attention due to its natural capability of extracting high-quality features from images. This is particularly important in remote sensing scenarios where images are highly detailed and their statistics can be very complex. Moreover remote sensing imagery presents specific challenges such as the environmental conditions, high-altitude imaging and imaging systems trade-offs that may lead to low-quality observation. The most studied inverse problems in remote sensing are pan-sharpening, super-resolution, deblurring, denoising and fusion. These tasks are very important for remote sensing scene interpretation, and they are used as pre-processing step for several image processing tasks, like feature extraction, detection, segmentation and classification.

Pan-sharpening

Remote sensing imaging systems cannot acquire high resolution both in the spatial and in the spectral domains. The majority of spaceborne remote sensing imaging systems acquire two types of observations, a panchromatic (PAN) component with high spatial and low spectral resolution, and a multispectral (MS) component having low spatial resolution and high spectral resolution. These two sources can be combined to produce HR spatial-spectral observations. The CNN architecture is ideal to capture intra-correlation across different bands of the MS images and the PAN image.

The pan-sharpening approaches exploit mainly two kinds of neural network architectures: encoder-decoder CNNs and CNNs without spatial compression/expansion.

The early CNN-based approaches are inspired by CNN architectures borrowed from the SR context on natural images.

In [117] the authors extended the SRCNN super-resolution architecture [34] to tackle the pan-sharpening. SRCNN is a three-layer CNN working at the target resolution from the beginning. They upsampled the low-resolution spectral bands to match the high-resolution panchromatic band and then stacked all the bands to form the pansharpened input. The SRCNN network is trained by minimizing the MSE loss between the pan-sharpened input and the original high-resolution spectral bands. As the real high-resolution multispectral image, acquired by a multispectral sensor operating at the same spatial resolution of the PAN, is impossible to acquire, the Wald protocol [188] is used to make training possible. The network is trained with downsampled MS image and PAN image as input and the original MS image as ground truth. They used 3 different datasets from the IKONOS, GeoEye-1, and WorldView-2 satellites.

The works [101, 173, 172] proposed residual CNN architectures inspired by the VDSR architecture devised by Kim et al. [81] in SR context. They employed residual connections and skip connections to overcome the vanishing gradient problem when training very deep CNN. These methods achieved promising results, outperforming the CNN-based method in [117]. The authors in [182] trained the well-known ResNet architecture to learn the missing high frequency details directly in the high-pass domain to generalize well to new satellites. A residual connection, acting as a spectral preservation, adds to the upsampled MS bands the high frequency details produced by ResNet. The ℓ_2 loss is used to train the ResNet network. Experimental results demonstrate the superiority of the method proposed in [182] compared to conventional pan-sharpening methods as well as CNN-based method in [117] in the WorldView-3 dataset.

Scarpa et al. [136] extended the baseline proposed in [117] to take into account different changes such as using ℓ_1 loss instead of ℓ_2, using a residual connection and a much deeper CNN network architecture. They performed an ablation study combining these changes in different ways. Moreover they introduced the target-adaptive pansharpening training procedure, where they first trained a network on the available dataset, fine-tuned on a target image until convergence and then fed this image to the fine-tuned model to obtain a pansharpened output. The authors tested this approach on four different datasets, namely IKONOS, GeoEye-1, WorldView-2 and WorldView-3, and compared it with their initial approach [117] as well as with conventional pan-sharpening techniques, obtaining good performance.

Another CNN-based approach for pansharpening is proposed in [140], where the MS bands and the pansharpening band are processed separately. The extracted features, coming from the two streams, are fused to produce the high resolution MS bands. The authors considered PAN and MS (4 bands) observations from QuickBird and Gaofen-1 satellites.

An encoder-decoder network with skip connections is used in [184] to achieve a wider range of spatial information to ensure better extraction of semantic features. They adapted the U-Net architecture [133] to be used in a regression sharpening problem.

In [106] the authors proposed a cGAN based method called PSGAN, composed of a two-stream CNN based generator to first separately process MS bands and the PAN band and then fuse them at feature level, rather than in pixel-level, reducing the spectral distortion. The aforementioned approach has been tested on QuickBird and GaoFen-1, outperforming deep learning based methods [117] and a number of traditional pansharpening methods.

Denoising

The studies on the remote sensing image denoising model and algorithm are very prosperous, especially for multi-spectral, hyper-spectral (HS) and synthetic aperture radar (SAR) data. These three types of data are inevitably corrupted by noise during acquisition and processing.

For HS data due to the sensor instability and atmospheric interference, HS images often suffer from multiple types of noise such as Gaussian noise, stripe noise, impulse noise, dead lines, and mixed noise. Most of the works in HS denoising assume the presence of the Gaussian noise only.

The aim of the method proposed in [176] is to perform denoising while preserving the spectral information. They trained a residual network with batch normalization to learn mapping between the spectral differences of the noisy and the clean HS observations. In the denoising stage, the learned residual network is employed to produce the clean spectral differences. In the meantime, a reference band is selected based on a principal component transformation matrix and fed to the famous pre-trained denoising network (DnCNN) [194] to generate a noise-free single band image. The latter is used as starting point to retrieve the other bands, by combining the clean reference band with the clean spectral differences.

In [185] the authors used a spatial–spectral residual CNN network where 2D convolutional filters enhance the feature extraction ability of the single band, and 3D convolutional filters simultaneously employ spatial–spectral information. Different convolutional kernel sizes are employed to produce multiscale features with different receptive field sizes. The training is carried out on simulated data patches from the Washington DC Mall image obtained by the Hyperspectral Digital Imagery Collection Experiment (HYDICE) airborne sensor. This method outperforms many of traditional non-CNN methods for HS denoising on the Indian Pines and Pavia University datasets.

Zhang et al. [197] addresses the removal of multiple types of noise in HS imagery other than the only Gaussian noise. They employ a multi-scale spatial-spectral CNN to simultaneously collect spatial and spectral information from the spatial and spectral gradients. Both the University of Pavia image obtained by the airborne Reflective Optics System Imaging Spectrometer (ROSIS) sensor and the Washington DC Mall image (HYDICE) were used to train the proposed model.

HSI-DeNet proposed in [23] addresses mixed type noise removal, devising a GAN-based approach with residual learning and dilated convolutions, which directly operates on the 3D data.

The remote sensors capturing HS data are passive acquisition systems. Another class of remote sensing imagery is the one captured by the Synthetic Aperture Radar (SAR) technology. Since SAR images are affected by an intense granular noise, called speckle noise, denoising is of primary importance for subsequent tasks involving image understanding. A more in-depth explanation on SAR image despeckling and its related works is given in Sec. 2.3.2.

Super-resolution

While most of the deep learning SR works are related to traditional natural images, lately CNNs have been exploited for remote sensing imagery. The aim of SR is to increase the spatial and/or spectral resolution of low resolution observations. This can be seen as an intermediate task before solving some downstream tasks. The most widespread CNN architectures are plain multi-layer networks without any downsampling or encoding as reducing spatial accuracy could lead to a loss of information, especially in the context of an upsampling task. Many works dealing with super-resolution in remote sensing field borrowed established network architectures initially devised for natural images such as SRCNN [34], Very Deep Super-Resolution (VDSR) [81] and the Enhanced Deep Super-Resolution (EDSR) [104].

In the first work [103] in SISR for remote sensing the authors re-trained a CNN network drawing from the paradigm of SRCNN, to increase the spatial resolution of MS satellite images from Sentinel-2 satellite, but actually focusing on a single band. In [159], VDSR [81] and the SRCNN [34] architectures are compared on different scale factors using images from SPOT and Pleiades satellites. Being the VDSR a deeper network with residual connection, it shows a significant improvement with respect to SRCNN. Lei et al. [96] proposed LGCNet, a residual network specifically designed to learn multiscale representations of remote sensing data including both local and global priors. Their network is an extension of VDSR, composed of a series of layers for feature extraction and “local-global” concatenation layer to combine multiple feature maps from different layers. LGCNet architecture was evaluated on RGB imagery from the UC Merced dataset, comparing to SRCNN, VDSR and the bicubic baseline.

Another work exploiting multiscale feature extraction is [170]. The authors perform a wavelet decomposition on aerial images to obtain multiple frequency bands and then trained a dedicated SRCNN-like network for each band, to match the wavelet multiscale representations obtain from the high resolution aerial image. For inference, each CNN is responsible for estimating the corresponding representation at the specific scale and the high-resolution image is obtained through wavelet synthesis. This method has been evaluated on the aerial images from RSSCN7 dataset obtaining better results with respect to VDSR architecture [81].

In [111] a CNN takes as input discrete wavelet transformed images and adopts recursive block and residual learning in global and local manners to reconstruct HR wavelet coefficients. The outputs of the network are combined by an inverse discrete wavelet transform to generate the final high spatial resolution image. The evaluation of the method is performed on the airplane images from the NWPU-RESISC45 dataset.

In [55] the authors proposed a generative network with an encoder-decoder architecture to recover a HR estimate in an unsupervised manner. This is an iterative approach where the generator is fed with random noise to produce a HR estimate. The HR estimate is downsampled and used to match the original LR image by minimizing a MSE loss. The HR estimate is progressively refined by iteratively using it as new input.

The GAN framework has recently gained attention for super-resolution of remote sensing images. In [71], a GAN architecture is proposed to jointly recover high frequency details and remove noises and artifacts affecting the edges. The generator network is composed of two sub-networks, aiming to reconstruct an intermediate HR result and to replace its noisy edges with the purified ones. The method uses imagery from Kaggle Open Source Dataset1 and Jilin-1 video satellite.

We presented an overview on common inverse problems tackled through deep learning methods. In the following section we deliver a background on the two inverse problems we focus on in this thesis.

2.3 Inverse problems of this thesis

In this section we report a more thorough explanation of the two inverse problems we deal with in the subsequent chapters of this thesis. Background and motivations are presented for multi-image SR and SAR despeckling problems.

2.3.1 Multi-image SR

SR techniques reconstruct a HR image from one or more LR images. Despite the continuous development of ever more advanced optical devices, for some imaging applications acquiring HR images can be cumbersome or impossible due to theoretical and practical limitations. Super-resolution methods are thus required to improve image resolution beyond the sensor capability. The approaches to image super-resolution can be broadly framed into two main categories: single-image SR (SISR) and MISR. SISR exploits spatial correlation in a single image to recover the HR version. However, the amount of information available in a single image is quite limited as some information has inevitably been lost in the LR image formation process. Certain applications provide multiple LR versions of the same scene from slightly different perspective to be combined by means of MISR techniques, where the reconstruction of high spatial-frequency details leverages the complementary information coming from different observations of the same scene. It is assumed that each input image is a degraded version of an underlying HR scene spoiled by blurring, down-sampling and affine transforms. The multiple LR images for a specific scene must have sub-pixel misalignments with each other. This means that the LR images cannot be obtained from each other by a transformation or resampling process. If the relative shifts between the LR images are integral, the images after motion registration will contain almost the same information. One possible way of obtaining multiple images of a specific scene is through hardware control such as in SPOT-5 satellite system [91]. The imaging mechanism is designed to have sensors able to acquire observations with known sub-pixel displacements like half pixel or multiple "looks" from different angles for the same scene. In this case the sub-pixel displacements are known a priori. For remote sensing problems, multiple images of the same scene can typically be acquired by a spacecraft during multiple orbits, by multiple satellites imaging the same scene at different times, or may be obtained at the same time with different sensors. In these cases the sub-pixel displacements need to be estimated in order to perform the reconstruction.

Historically, the most common methods to solve MISR problems are the regularized ones. To apply a regularized framework, an imaging model has to be defined to describe the various degrading factors of an image acquisition process. In MISR the forward model usually comprises motion, blurring, down-sampling, and noise degradations and it is simulated as follows:

$$y_k = D_k B_k M_k x + n_k \tag{2.3}$$

where x represents the HR unobserved image. M_k, B_k and D_k represent the motion process, the blur matrix and the down-sampling matrix of the k^{th} LR image y_k respectively.

Regularized methods are some of the most effective multi-frame SR reconstruction approaches. Based on the acquisition model stated in (2.3), a minimization problem is solved to reconstruct the HR estimate from a set of warped, blurred, noisy, and downsampled observed images:

$$\arg\min_x E(D_k B_k M_k x, y_k) + R(x)$$

In the past decades, many kinds of regularizers $R(x)$ have been proposed to preserve edge information while removing image noise, such as Tikhonov regularizer [54, 126], Markov random field regularizer [24], total variation (TV) [21, 114, 192] and bilateral total variation (BTV) [40]. In particular, a few works have been proposed for remote sensing applications. Shen et al. [142] proposed a maximum-a-posteriori (MAP) SR method with Huber prior for MODIS images captured in different dates. Another multi-temporal SR method was proposed by Li et al.[99] for Landsat-7 PAN images. Instead, other works [110, 192] proposed SR methods for multi-angle remote sensing captures.

Most of the above SR methods assume a priori knowledge of the motion model, blur kernel and noise level, where both blur identification and image registration are performed as a preprocessing stage before reconstruction. However, there are many applications where knowing the motion parameters and the image degradation process or reliably estimating them can be challenging. For this reason, many studies have been carried out blind SR image reconstruction [125, 56]. These blind methods usually work in two stages, namely (1) image registration from LR images, followed by (2) simultaneous estimation of both the HR image and blurring function. Along the same lines, Zhang et al. [191] also integrated the joint estimation of the blurring function. Moreover, Kato et al. [77] recently proposed a sparse coding method where image registration and sparse coding are performed in a unified framework reducing the image registration error.

Recent work in deep learning has demonstrated that deep neural networks can leverage large collections of training data to directly compute regularized reconstructions, by performing a supervised inversion of the forward model.

The main drawback of supervised deep learning method for solving inverse problems is the need of a large training dataset. Pansharpening, super-resolution, denoising and deblurring methods discussed previously follow a common protocol to generate synthetic training pairs. Focusing on MISR, existing methods create the dataset by assuming a known forward model with known parameters. The LR images are obtained by applying motions, degrading through a blur filter and downsampling the HR images. The model in (2.3) is a simplified version of a real acquisition model that applies much more complex transformations. Generating the training pairs through (2.3) means limiting the deep learning method to learn an inversion function of a overly simplified and non realistic forward model that may lead to an unrepresentative training phase. In the context of remote sensing, one way to solve this is to obtain the LR images and the corresponding HR image of the same scene from different spacecrafts capturing images at different resolutions. This leads to a model that is suboptimal when performing inference on images captures from both spacecrafts.

The Advanced Concepts Team of the European Space Agency has issued a competition [115] to perform MISR for the images acquired by the PROBA-V satellite. The unique feature of this dataset is that both LR and HR images have been acquired by the same spacecraft at multiple times, as opposed to the majority of the previous works where LR images are artificially down-scaled, degraded and shifted versions of an HR image. In this context, developing a successful MISR model hinges on solving important problems such as image registration, invariance to absolute brightness variability, time-varying scene content (e.g., due to the time elapsed between multiple acquisitions), and unreliable data (e.g., due to cloud

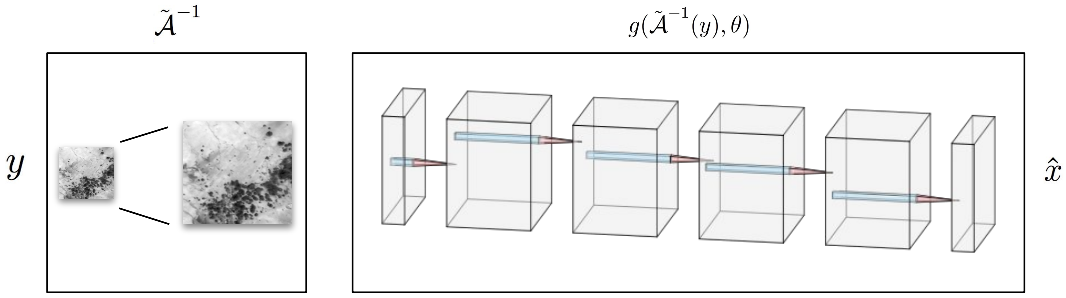
$\tilde{\mathcal{A}}^{-1}$
$g(\tilde{\mathcal{A}}^{-1}(y),\theta)$
y
$\hat{x}$

Figure 2.3: CNN coupled with a bicubic upsampling as the approximate inverse of the forward model in a SISR inverse problem.

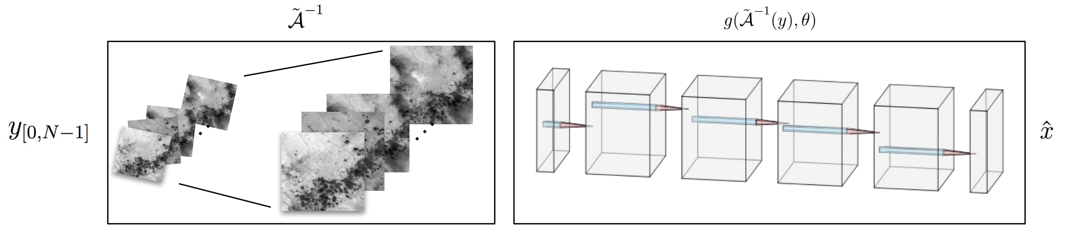
$\tilde{\mathcal{A}}^{-1}$
$g(\tilde{\mathcal{A}}^{-1}(y),\theta)$
$y_{[0,N-1]}$
$\hat{x}$

Figure 2.4: CNN coupled with a bicubic upsampling and a registration process as the approximate inverse of the forward model in a MISR inverse problem.

coverage).

When solving an inverse problem with a deep learning method it is common to use the partial knowledge of the forward model and use an approximate inverse $\tilde{\mathcal{A}}^{-1}$, to first map the observations back to image domain and then train a neural network to refine the resulting images.

$$\hat{x} = f(y,\theta) = g(\tilde{\mathcal{A}}^{-1}(y),\theta),$$

where g is the network mapping the new input $\tilde{\mathcal{A}}^{-1}(y)$ onto the ground truth x.

In super-resolution for instance it is very common to inject domain knowledge of the forward model into the training process and dramatically reducing the complexity of the image reconstruction, by coarsely upsampling the LR images. Most of the deep learning methods in SR problems feed to the network a bicubic interpolation of the LR images as shown in in Fig. 2.3. The task of the network is simply to adjust the higher frequency details and remove artifacts.

In the case of MISR, the registration is also performed as a pre-processing step by fixed filters.

In the deep learning MISR literature most of the methods perform the registration step as a pre-processing step and the reconstruction is learned from the dataset by a deep network as depicted in Fig. 2.4. This causes the registration error to be propagated through the learning procedure without never being corrected. In the last years, deep learning based methods have been proposed to solve MISR problems in context of video super-resolution [75, 20]. Most of these works are composed of two steps: a motion estimation and compensation procedure followed by an upsampling process, heavily relying on the prior motion estimation. Recently, Jo et al. [74] presented a novel end-to-end residual CNN to produce a SR image without explicit motion compensation. A CNN is trained to simultaneously solve motion estimation and HR image reconstruction tasks by producing a set of pixel-dependent filters and a residual correction. A similar idea was developed by Tian et al. [154]. However, little work has been done on deep learning MISR methods in the context of remote sensing, which poses specific challenges. Kawulok et al. [78] propose a MISR method that does not fully exploit the benefit of deep learning, restraining their CNN to solve a SISR problem. The fusion of the upsampled LR images is performed by the median shift-and-add method, generating a SR image that is used as initial guess for a classic regularized method. Their method is not end-to-end trainable in a supervised manner and their CNN is trained against LR images obtained by artificially degrading HR images. Inspired by the recent video super-resolution works, we aim to tackle the MISR problem on satellite images by jointly registering the input LR images and reconstructing the SR image, all within an end-to-end trainable CNN, where the two tasks are optimized jointly. In general, incorporating knowledge of the forward model into the reconstruction network makes the learning process easier as long as it is simple enough for the training to compensate for the errors introduced by the approximation of the forward model. In our method we leave as initial reconstruction only the bicubic upsampling, instilling minimal pre-processing in the approximate inverse $\tilde{\mathcal{A}}^{-1}$. Incorporating also the registration approximation of our forward model would lead to errors that the network is not able to correct by construction.

2.3.2 Despeckling

SAR is a coherent imaging system and as such it strongly suffers from the presence of speckle, a signal dependent granular noise. Information extraction from SAR images is heavily impaired by speckle noise, hence despeckling is a crucial preliminary step in scene analysis algorithms.

The most employed imaging forward model in literature is the following:

$$y = \mathcal{A}(x) \cdot n \tag{2.4}$$

where $y \in \mathbb{R}^{W \times H}$ is the observed image, $x \in \mathbb{R}^{W \times H}$ represents the unknown input image, $\mathcal{A}$ is the identity operator and $n \in \mathbb{R}^{W \times H}$ is the spatially multiplicative speckle noise. The model in (2.4) can deal with either intensity or amplitude as well as with single-look or multi-look images. The noise n is usually considered to be an uncorrelated random process and most of the methods in literature make this assumption. Speckle noise in SAR images can actually be considered spatially correlated as the SAR acquisition model tends to correlate it.

Tackling a despeckling task based on this non linear model is hard and in literature researchers introduced some simple manipulations to simplify the forward model in (2.4):

- recasting the multiplicative noise model into a signal-dependent additive noise model:

  $$y = x \cdot n = x + (n-1)x$$

  where $(n-1)x$ represents the signal-dependent noise speckle process;

- applying a homomorphic transformation to the model, by taking the logarithm of the observed image y and obtaining a signal-independent additive noise model:

  $$\log(y) = \log(x) + \log(n)$$

  The homomorphic model introduces a bias in the despeckled image as $E[n] \neq \exp\{E[\log(n)]\}$ and moreover, it changes radically the data dynamics leading to distortions.

The last decades have seen a multitude of SAR image despeckling methods, that can be broadly categorized into four main approaches: spatial-domain methods, wavelet-domain methods, non-local methods and deep learning methods. Filtering-based techniques such as Lee filter [94], Frost filter [44], Kuan filter [86] represent the early attempts to solve SAR despeckling and they operate in spatial domain. Subsequent works in spatial domain aimed to reduce speckle under a non-stationary multiplicative speckle assumption. A popular example is represented by the MAP approaches aiming to give a statistical description to the SAR image. A few MAP-based works have been proposed and the most representative is the Γ-MAP filter [108] that solves the MAP equation modeling both the radar reflectivity and the speckle noise with a Gamma distribution.

Wavelet-based methods proved to be more effective than spatial domain ones, enabling multi-resolution analysis and boosting analysis under non-stationary characteristics. They despeckle SAR images in the transform domain by estimating despeckled coefficients and then by applying the inverse transform to obtain the cleaned SAR image. A first subclass of wavelet based methods solve the despeckling problem with a homomorphic approach, consisting in applying a logarithmic transform of the data to convert the multiplicative noise into an additive one. The works in [51, 45] applied the traditional wavelet shrinkage based on hard- and soft-thresholding with an empirical selection of the threshold. Further wavelet-based methods [2, 148, 12, 1] introduce prior knowledge about the log-transformed reflectance in the wavelet domain, employing a MAP estimator. Most of the wavelet-based homomorphic approaches do not compensate for the bias in the reconstructed images resulting from the mean of the log-transform speckle. To cope with this problem, a non-homomorphic approach has been considered by some works [63, 118, 41, 5] in the wavelet domain, dealing with a signal-dependent speckle whose distribution parameters are harder to be estimated.

In general, both spatial domain and wavelet domain techniques yield limited detail preservation with the introduction of severe artifacts. The amount of information provided by a local window is quite limited and the need of incorporating more information from the neighborhood led to the proliferation of non-local methods. The pioneering work in this field is represented by the non-local means (NLM) filter [26] that performs a weighted average of all pixels in the image and the weights depend on their similarity with respect to the target pixel. The weights are defined by computing the Euclidean distance between a surrounding patch centered at a neighboring pixel and a local patch centered at the target pixel. In [30], the Probabilistic Patch-Based (PPB) algorithm has been proposed to adapt the non-local means approach to SAR despeckling. The authors devised a patch similarity measure that generalizes to the case of multiplicative, non-Gaussian speckle.

In [31], the authors proposed another extension of NLM for despeckling, called NL-SAR, to deal with arbitrary SAR modalities (SAR, polarimetric SAR, interferometric SAR) and any number of looks. They proposed a unified non local framework where several non local estimations are performed and the best one is locally selected to ensure adaptivity to local structures. Moreover, in order to ensure robustness to noise correlation, similarities are weighted using kernels learned from a homogeneous region.

NLM inspired a number of extensions in the Gaussian noise context such as the Block-Matching 3D (BM3D) algorithm [29], a combination of non-local approach, wavelet domain shrinkage and Wiener filtering in a two-step process.

One of the most popular SAR despeckling algorithm is the SAR version of BM3D [29] (SAR-BM3D) that follows the same BM3D phases with an adaptation to the SAR statistics in the grouping phase where the same PPB similarity measure is used. Moreover the hard-thresholding and Wiener filtering, suitable in the Gaussian noise context, are replaced with an LMMSE estimator (based on an additive signal-dependent noise model).

The success of deep learning on many tasks involving image processing has suggested that the powerful learning capabilities of CNNs could be exploited for SAR despeckling and a few works have started addressing the problem. Such methods use a supervised training approach where the network weights are optimized by minimizing a distance metric between noisy inputs and clean targets. As for most of imaging inverse problems tackled with a supervised deep learning approach, the retrieval of training pairs is difficult. For pansharpening and super-resolution inverse problems the training pairs are generated by treating the original data as ground truth and their downsampled version as low resolution obtained through a simplified forward model. In denoising and deblurring the approach is the same.

Optical clean images are considered as ground truth and synthetically degraded, in this case, through a fully known forward model to create the noisy versions.

In SAR context the clean underlying image is impossible to collect. One might be able to gather multi-temporal SAR images of the same scene and average them to get an approximated ground truth, but this would pose other challenges as acquisition of multi-temporal data, scene registration and robustness to temporal variations can be challenging, leading to a sub-optimal rejection of speckle. Similarly to the aforementioned inverse problems, the researchers solve despeckling by resorting to synthetic datasets where optical images are used as ground truth and their artificially speckled version as noisy inputs. This creates a domain gap between the features of synthetic training data and those of real SAR images, possibly leading to the presence of artifacts or poor preservation of radiometric features when despeckling real SAR images.

Chierchia et al. [25] proposed SAR-CNN, which applies a DnCNN-like [194] supervised denoising approach to SAR data. They exploit the homomorphic approach to deal with multiplicative noise model and use a new similarity measure for speckle noise distribution as loss function rather than the usual Euclidean distance. Clean data for training are obtained by averaging multitemporal SAR images. Wang et al. [169] proposed a residual CNN (ID-CNN) trained on synthetic SAR images, to directly estimate the noise in the original domain, and, hence, the despeckled image is obtained by dividing the noisy image by the estimated noise. Training is once again supervised using synthetically speckled optical images and carried out with the Euclidean distance and a total variation regularization as loss function. Several subsequent deep learning works [168, 198, 49, 100, 193, 92] proposed slight variations on the topic by introducing different architectures and losses, but all under the supervised training umbrella using synthetically speckled SAR images. In [168] the authors proposed IDGAN, a deep learning SAR despeckling method based on a generative adversarial network (GAN) and trained using a weighted combination of Euclidean loss, perceptual loss and adversarial loss. In [49], a dilated densely connected network (SAR-DDCN) trained with Euclidian distance, was proposed to enlarge the receptive field and to improve feature propagation and reuse. A combination of hybrid dilated convolutions and both spatial and channel attention modules through a residual architecture called HDRANet was proposed in [100], to further improve the feature extraction capability. More recently, Cozzolino et al. [27] proposed a method that combines the classical non-local means method with the power of CNN, where NLM weights are assigned by a CNN with non local layers.

Until now, the power of CNN has not been fully exploited yet, since most of the works in literature make use of synthetic SAR images. Inspired by the recent blind-spot CNN denoising works, we tackle SAR despeckling with a self-supervised Bayesian framework relying on blind-spot CNN. This is a modified version of the classical CNN, which reconstructs each clean pixel exclusively from its neighboring

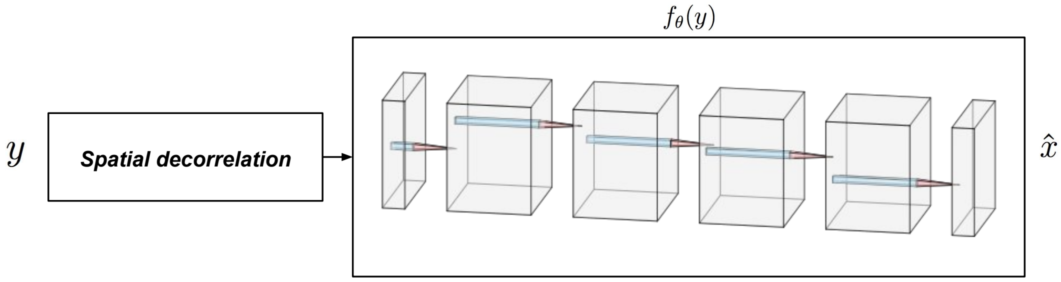

Figure 2.5: The CNN is trained to learn the inversion of the SAR acquisition system. A spatial decorrelator [90] is employed to whiten the speckle noise.

pixels. However, the blind-spot CNN requires to assume spatially uncorrelated noise in order to be properly trained and it uses a training dataset composed by the only noisy SAR image per scene $(y^{(i)})_{i=1}^{N}$ where N is the number of scene in the dataset. The noisy SAR image is used both as noisy input and ground truth. To this end, we preprocess the noisy SAR images with a decorrelation procedure [90]. Fig. 2.5 shows the general architecture of our method, where we do not incorporate any domain knowledge of the forward model into the training process. The network is responsible for the inversion of the full forward model.

Chapter 3

DeepSUM: Deep neural network for Super-resolution of Unregistered Multitemporal images

Deep learning methods have been proved highly successful in the SISR problem but little work has been done for the MISR problem with remote sensing data. In particular, we aim to develop a deep learning technique to solve a MISR problem with multitemporal unregistered imagery, that requires to handle some important problems such as image registration, invariance to absolute brightness variability, time-varying scene content (e.g., due to the time elapsed between multiple acquisitions), and unreliable data (e.g., due to cloud coverage).

In this chapter we present a deep learning architecture addressing MISR applied to a novel dataset provided by the European Space Agency's Advanced Concepts Team in the context of a challenge [80]. The goal of the challenge is to super-resolve images from the PROBA-V satellite. The method presented in this chapter won the challenge by achieving the highest fidelity on the reconstructed images. The unique feature of this dataset is that both LR and HR images have been acquired by the same spacecraft, as opposed to previous works where LR images are artificially down-scaled, degraded and shifted versions of an HR image. In this case, the forward model becomes much more complicated than in (2.3) as it should describe complicated factors found in real scenarios. In this work we do not need to approximate the sensor imaging model as a CNN is exploited to invert it by relying on a large dataset.

Our main contribution is DeepSUM, a novel CNN-based architecture to combine multiple unregistered images from the same scene exploiting both spatial and temporal correlations. Our method includes image registration inside the CNN architecture, as a subnetwork named RegNet, which dynamically computes custom filters and applies them to higher dimensional image representations. This is in contrast with the vast majority of deep-learning MISR methods in literature [79] that compensate for the motion as a preprocessing step. This approach allows the registration task to leverage the feature learning capabilities of the network in order to be more accurate and resilient to scene variations, and it also optimizes it in an end-to-end fashion for the final goal of reconstructing a single HR image. The proposed method is blind to the image degradation model as it does not require to explicitly model the blur kernel or the noise statistics, and it is robust to temporal variations in the scene as well as occlusions due to cloud coverage. The only assumption of our model is the translational nature of the shift among LR images.

The remainder of this chapter is organized as follows. Section 3.1 provides details on the novel PROBA-V dataset. Sections 3.2 and 3.3 detail the proposed framework and the training procedure. Section 3.4 contains results and performance evaluation.

3.1 The PROBA-V SR dataset

At present, it is difficult to find a dataset collecting both a set of real-world LR observations and the corresponding HR image for the same scene, as captured from the same platform. Many of the works found in the SR literature are based on simulated data, where LR observations for a specific scene are obtained through a degradation and down-sampling process of the HR images by assuming a sensor imaging model. This is a simplified scenario as it either assumes a non-blind problem, i.e., the degradation model can be characterized to some extent, or has the limitation that a too simple degradation model may not accurately match the real one, especially when in presence of temporal variations in the scene content.

The Advanced Concepts Team of the European Space Agency has issued a competition [115] to perform MISR for the images acquired by the PROBA-V satellite. PROBA-V is an Earth observation satellite designed to map land cover and vegetation growth across the entire globe. It was launched in 2013 into a Sun-synchronous orbit at an altitude of 820km. Its payload provides an almost global coverage with 300m LR images and 100m HR images. However, the HR images are acquired with a higher revisit time, roughly one every 5 days, instead of one per day. The dataset gathers satellite data from 74 regions located around the world from the PROBA-V mission. Images are provided as level 2A products composed of radiometrically and geometrically corrected Top-of-Atmosphere reflectance in Plate Carrée projection for the RED and NIR spectral bands. The size of the collected images is 128×128 and 384×384 for the LR and HR data respectively. The images have a single channel with a bit-depth of 14 bits. Each data point consists of one HR image and several LR images (ranging from a minimum of 9 to a maximum of 30) from the

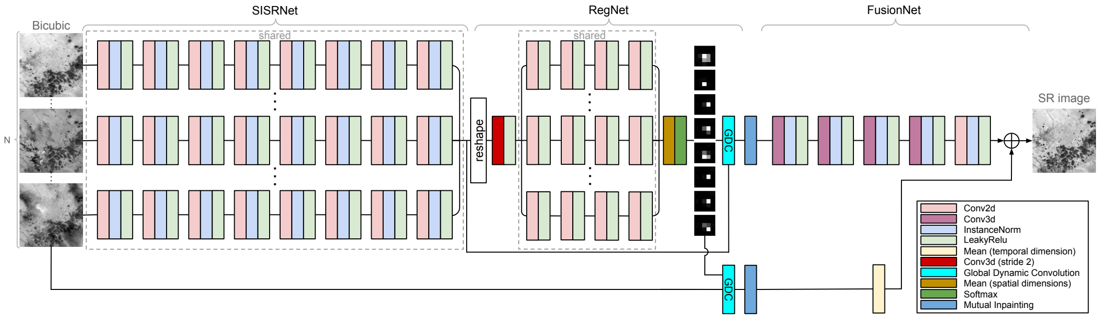

Figure 3.1: DeepSUM network. The N input bicubic-upsampled and registered images are independently processed by a SISRNet subnetwork, and their features used by the RegNet to compute registration filters to register the feature maps of the N images to each other. The FusionNet subnetwork merges the features of the images to produce a residual image. The residual image is then added element-wise to the average of the registered input to obtain the SR image.

same scene. In total, the dataset contains 1160 scenes, 566 are from NIR spectral band and 594 are from RED band. The images of a specific scene are captured at multiple times over a maximum period of 30 days. Weather and changes in the landscape pose a limitation in the similarity of the images. Clouds, cloud shadows, ice, water, missing regions, presence of agricultural activities and, in general, human activity are the main sources of inconsistency across these images, thus posing a major challenge for any image fusion method. Moreover, each image comes with a mask, indicating which pixels in the image can be reliably used for reconstruction (e.g., they are not covered by clouds). The geometric disparity among the images can be considered as translational only. Subpixel shifts in the content of the LR images do occur and are indeed important for the MISR task.

The unique nature of this dataset (with real LR and HR images captured by the same platform at multiple times) makes for an interesting case study for SR techniques, enabling data-driven methods such as CNNs to learn the inversion of possibly complex degradation models and the best feature fusion strategy to handle temporal variations.

3.2 Proposed architecture

Even though the LR images roughly represent the same scene as the HR image, the described PROBA-V dataset makes the SR task quite complicated, by posing a bunch of additional challenges:

- the LR images are not registered with each other;
- the LR images and the HR image are not registered;

- the brightness of the HR image may be different from that of any LR image;
- the scene changes over multiple acquisitions;
- LR and HR images may be covered by different clouds and cloud shadows patterns or affected by corrupted pixels.

To tackle this problem we propose to employ a supervised deep learning approach to learn a mapping function f from $y_{[0,N-1]}$, representing the N LR images, to the HR image x, aiming to invert the unknown degradation model $\mathcal{A}$ defined in Eq. (1.1):

$$\hat{x} = f(y_{[0,N-1]}, \theta),$$

where θ represents the model parameters, f represents the mapping function from LR to HR and $\hat{x}$ the super-resolved image. $y_{[0,N-1]}$ and x are represented as real-valued tensors with shape $N \times H \times W \times C$ and $1 \times rH \times rW \times C$ respectively, where H and W are the height and the width of the input LR frames, C is the number of channels and r is the scale factor. The proposed CNN learns a mapping between bicubic interpolation and the ground truth using a residual connection. By using bicubic interpolation we incorporate knowledge of $\mathcal{A}$ into the reconstruction network. As approximate inverse of $\mathcal{A}$, bicubic interpolation is the most used as it helps the learning process and it is simple enough to allow the training to recover from the errors introduced by the approximation of the forward model $\tilde{A}^{-1}$. Our actual CNN is represented by $g(\theta, y)$:

$$\hat{x} = g(\tilde{\mathcal{A}}^{-1}(y_{[0,N-1]}), \theta) + h(\tilde{\mathcal{A}}^{-1}(y_{[0,N-1]})),$$

where $\tilde{A}^{-1}$ represents the preprocessing step where the LR images are bicubically interpolated to the desired size and then fed into a CNN $g(\theta, y)$ composed of three main building blocks, while $h(y)$ represents the registration and fusion applied to the residual connection. An overview of the network is shown in Fig. 3.1.

The first block, called SISRNet, is a feature extractor that can be seen as a SISR network without the output projection to a single channel. Each of the N input images is processed independently by a sequence of 2D convolutional layers. The convolutional filters are shared along the temporal dimension, i.e., all the N interpolated LR (ILR) images $I^{\text{ILR}}_{[0,N-1]} = \tilde{\mathcal{A}}^{-1}(y_{[0,N-1]})$ go through the same set of filters.

The second network block, called RegNet, aims at estimating a set of filters to register the N higher dimensional image representations produced by the SISRNet block to each other at integer-pixel precision (notice that the network is working at the same spatial resolution as the HR image, so integer shifts correspond to sub-pixel shifts in the LR data). RegNet has been devised to align $N-1$ instances with respect to the first, taken as reference, by operating purely translational shifts.

Therefore, the output is a set of $N-1$ 2D filters to be applied spatially to each feature map of the $N-1$ inputs.

Finally, the third block, called FusionNet, merges the registered image representations in the feature space in a "slow" fashion, i.e., by exploiting a sequence of 3D convolutional operations with small kernels. The output is a single super-resolved image.

In the following, we are going to describe each individual block more in detail.

3.2.1 SISRNet Architecture

The goal of SISRNet is to exploit spatial correlations to improve upon the initial bicubic interpolation. In doing so, the network learns to extract visual features that can be conveniently exploited by the subsequent network blocks. SISRNet has multiple 2D convolutional layers whose weights are shared among the N input images, effectively processing each of them independently. Each convolutional layer is followed by Instance Normalization [160]. Instance normalization is used in place of Batch normalization [67] to make the network training as independent as possible of the contrast and brightness differences among the input images.

3.2.2 RegNet Architecture

RegNet is composed of two sub-blocks: a CNN, and a global dynamic convolutional layer (GDC). The CNN processes the higher dimensional image representations $Z^{\text{ILR}}_{[0,N-1]}$ generated by SISRNet block and outputs a set of $N-1$ filters $G_{[1,N-1]}$. Each filter G_i is subsequently applied in the spatial dimensions to each of the channels of Z^{ILR}_i by the GDC layer by means of a 2D convolution in order to register each feature map of Z^{ILR}_i with respect to the reference one Z^{ILR}_0. The filters $G_{[1,N-1]}$ have a fixed support equal to $K \times K$ that upper bounds the maximum possible translational shift correction to $\lfloor K/2 \rfloor$. Notice that there is an implicit assumption that all feature maps of an image require the same shift to be registered with the reference, so that the computed filter is shared channel-wise. The registered feature maps $Z^{\text{IRLR}}_{[0,N-1]}$ of the N images are thus obtained as:

$$G_i = f_{\text{RegNet}}(Z^{\text{ILR}}_{[0,N-1]}, \theta_{\text{RegNet}}), \quad i = 1, \ldots, N-1$$

$$Z^{\text{IRLR}}_i = \begin{cases} Z^{\text{ILR}}_i, & i = 0 \\ G_i * Z^{\text{ILR}}_i, & i = 1, \ldots, N-1 \end{cases},$$

being $*$ the 2D convolution operator. The same filters are also applied to the input ILR images to register them in the residual connection:

$$I^{\text{IRLR}}_i = \begin{cases} I^{\text{ILR}}_i, & i = 0 \\ G_i * I^{\text{ILR}}_i, & i = 1, \ldots, N-1 \end{cases},$$

The novelty of this network is twofold: firstly the filters are dynamically computed for each input image, and secondly it makes use of the features to compute the per-image optimal registration instead of performing it in image space, like most of motion estimation algorithms do. This allows to leverage the powerful feature space of the network to boost the registration performance by making it robust to scene variations. In addition, it is fully differentiable so that the whole architecture can be trained end-to-end.

More in detail, the operations performed by RegNet are depicted in Fig. 3.2. SISRNet outputs a tensor Z^{ILR} with shape $N \times rH \times rW \times F$, where F is the number of features, that is reshaped before being fed to RegNet. The features of the first image Z_0^{ILR} are chosen as a reference and a new tensor of size $2(N-1) \times rH \times rW \times F$ is built by replicating the reference Z_0^{ILR} $N-1$ times and interleaving each replica with the other $(N-1)$ image representations $Z_{[1,N-1]}^{\text{ILR}}$. This sequence of paired reference/unregistered features is then processed by convolutional layers to produce the filters. RegNet has a first 3D convolutional layer and a series of shared 2D convolutional layers. The first layer is the key component of registration and it processes the $2(N-1)$ image representations in pairs by using a stride equal to 2 along the temporal dimension and filters of shape $2 \times 3 \times 3$. This operation allows to correlate the features of each Z_i^{ILR} with respect to the ones of the reference Z_0^{ILR} and compute the shift. Notice that this processing in pairs is necessary to avoid any ordering ambiguity and let the network understand that the output is relative to the reference. After this 3D convolutional layer the output tensor has shape $(N-1) \times rH \times rW \times F$.

This tensor passes through a series of 2D convolutional layers with shared weights along the temporal dimension. The last RegNet 2D convolutional layer applies a number of kernels corresponding to the spatial size of the dynamic filters $K \times K$, obtaining a tensor with shape $(N-1) \times rH \times rW \times K^2$. Each value over the spatial dimensions can be seen as a local estimate of the desired shift based on the local image representation. Since there is a global translational shift by assumption, the values are averaged over the spatial dimensions to obtain a tensor with shape $(N-1) \times 1 \times 1 \times K^2$.

Finally, this tensor is passed through a softmax layer, so that the values over the last dimension (K^2) add up to 1. The softmax layer promotes a spiked filter with most elements set to zero [17]. The final tensor represents the $(N-1)$ dynamic filters with shape $K \times K$ to be used to register the $(N-1)$ image representations with the GDC operation, as in Fig. 3.3.

3.2.3 Mutual Inpainting

The registered and interpolated feature maps $Z_{[0,N-1]}^{\text{IRLR}}$ have regions with unreliable values due to cloud coverage, shadows, corrupted pixels and so on. A per-pixel boolean mask is assumed to be available as side information, with the purpose of

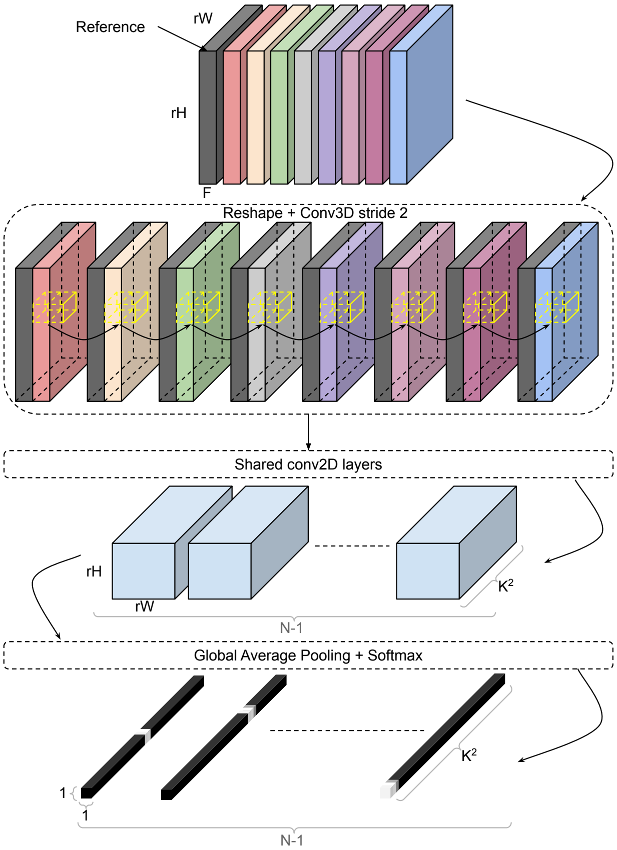

Figure 3.2: Visual depiction of the RegNet operations to generate the dynamic registration filters from the image features produced by SISRnet.

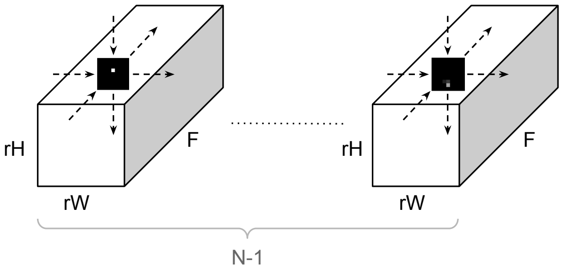

Figure 3.3: GDC: convolution between the dynamic filters and the image representations to align them with respect to the reference.

mapping pixels that can be reliably used for the fusion task. An example on how to obtain such mask is to run a cloud detection algorithm on the image to segment areas with clouds. This is very important because areas occluded by clouds do not provide any useful information. In order to prevent FusionNet from combining feature maps from multiple images where some have unreliable intensities, we fill the masked areas with values from the feature maps of other images. The regions with missing or unreliable values in each feature map of each image are filled with values taken from the corresponding feature map of other images having reliable values in those regions, if any are available. In the case none of the images has feature maps with reliable values, we keep those unreliable regions as they are. Since after RegNet the masks are not aligned with the corresponding image representations, we shift the masks by an integral shift as close as possible to subpixel shift computed and operated by RegNet. This procedure is performed on both the residual image representations $Z^{\mathrm{IRLR}}_{[0,N-1]}$ and the registered input images $I^{\mathrm{IRLR}}_{[0,N-1]}$ right before averaging them.

3.2.4 FusionNet Architecture

The N registered outputs $Z^{\mathrm{IRLR}}_{[0,N-1]}$ are progressively fused by the FusionNet subnetwork. FusionNet is composed of $\lfloor N/2 \rfloor$ 3D convolutional layers where convolution is performed without any padding in the temporal dimension, so that the temporal depth eventually reduces to 1. This architecture implements a "slow" fusion process in the feature space, which allows the network to learn the best space to decouple image features that are relevant to the fusion from irrelevant variations and to construct the best function to exploit spatio-temporal correlations [20]. Finally, the proposed architecture employs a global input-output residual connection. The network estimates only the high frequency details necessary to correct the bicubically-upsampled input. This is an established technique for image restoration problems using deep learning [194], including SISR. However, with respect to SISR, our proposed network is a many to one mapping, so the residual is actually added element-wise to a basic merge of $I^{\mathrm{IRLR}}_{[0,N-1]}$ in the form of their average. Notice that registration of the input images is performed before averaging by means of the same filters produced by RegNet. Hence, the output is computed as follows:

$$\bar{I}^{\mathrm{IRLR}} = \frac{1}{N} \sum_{i \in [0,N-1]} I_i^{\mathrm{IRLR}},$$
$$\hat{x} = \bar{I}^{\mathrm{IRLR}} + R.$$

being R the residual estimated by the CNN.

3.2.5 Loss Function

Model parameters are optimized by minimizing a loss function computed as a modified version of the Euclidean distance between the SR image and the HR target. Minimizing the Euclidean distance is optimal in terms of the mean-squared error metric. Some deep learning works on SISR attempted to use an adversarial loss [93]. While this approach produces visually pleasing results, it tends to hallucinate information, resulting in lower MSE scores and less reliable products in the context of remote sensing; hence, the adversarial approach has not been followed in the present work. As we mentioned in Sec. 3.1, since the PROBA-V satellite does not capture LR images and HR images of a specific ground scene simultaneously, there are discrepancies coming from different weather conditions, changes in the landscape and variable absolute brightness due to the large interval between scene acquisitions. The LR images could be quite different from one another and from the corresponding HR image as well. For this reason, we must make the training objective as invariant as possible to such conditions. In particular, in order to build invariance to absolute brightness differences between $\hat{x}$ and x, the modified loss function equalizes the intensities of the SR and HR images so that the average pixel brightness is the same on both images. Moreover, since $\hat{x}$ and x could be shifted, the loss embeds a shift correction. $\hat{x}$ is cropped at the center by d pixels, i.e., as many pixels as the maximum expected shift. Then all possible patches $x_{u,v}$ of size $(rH-d)\times(rW-d)$ for vertical and horizontal shifts u, v are extracted from the target x. All possible Euclidean distances are computed and the minimum one is taken as loss to optimize. In summary, our loss is as follows:

$$L = \min_{u,v\in[0,2d]} \|x_{u,v} - (\hat{x}_{\text{crop}} + b)\|^2, \tag{3.1}$$

where $\hat{x}_{\text{crop}}$ is the cropped version of $\hat{x}$ and b represents the brightness correction:

$$b = \frac{1}{(rW-d)(rH-d)} \sum_{X,Y} \left(x_{u,v} - \hat{x}_{\text{crop}}\right).$$

The loss is computed by utilizing only the HR image pixels that are marked as reliable by the mask provided with the dataset and the SR image pixels for which at least one out of N LR images were clear. The reason for this is that a cloud in the HR image can never be predicted from terrain data in the IRLR images, so its pixels should not contribute to the loss function. Viceversa, it is also impossible to predict HR terrain if all the IRLR images have concealed regions.

3.3 Training process

3.3.1 Pre-training

Training the whole network end-to-end from scratch is hard due to several local minima that do not make SISRNet, RegNet and FusionNet work as expected. For example, the gradients computed during training do not sharply discriminate the RegNet task to generate registration filters from the high-resolution feature learning of SISRNet.

In order to solve this issue, it is possible to pretrain each block to handle its specific subtask, and then combine all the blocks to be fine-tuned in an end-to-end fashion.

SISRNet pre-training

As mentioned in Sec. 3.2, SISRNet aims to independently super-resolve each of the N input images, while providing useful higher dimensional image representations. SISRNet is pretrained by setting up a pure SISR problem (i.e., a single input image) where an additional projection layer is added at the end, in order to turn the high-dimensional feature space into a single-channel image. SISRNet with the final projection layer is trained with the same objective function of the final training, where the single image reconstruction is compared with the only HR image available for the scene. The rationale behind this is to make SISRNet exploit spatial correlations as much as possible to generate the best image features for the SISR task. Once the pretraining procedure is completed, the final layer is removed and a dataset of feature maps of the input training images is generated to pretrain RegNet.

RegNet pre-training

The purpose of pre-training RegNet is learning to generate registration filters, i.e., filters that shift the feature maps of the $N-1$ input images with respect to the reference input. This operation would be quite challenging to learn if the whole network was trained end-to-end, so its pretraining is crucial for the overall network performance. RegNet is pre-trained by casting registration as a multi-class classification problem. Each dynamic registration filter generated by the network is viewed as a probability distribution over the possible shifts with the objective of estimating the correct shift. The number of classes is K^2 since the filter size is $K \times K$. In case of an ideal shift of an integer number of pixels, the predicted filter should be a delta function centered at the desired shift.

The input data to be used for the pretraining of RegNet are the feature maps produced by the pretrained SISRNet for the images in the training set. As described in Sec. 3.2.2, the input to RegNet are N feature maps from images of the same scene. These feature maps are then synthetically shifted with respect to the first one by a random integer amount of pixels. The purpose is to create a balanced dataset where all possible K^2 classes (shifts) are seen by the network. The desired output is a filter with all zeros except for a one in the position corresponding to the chosen shift. A cross-entropy loss between the softmax output and the true filter is used to learn the RegNet weights.

3.3.2 Final training

The proposed network is finally trained as a whole, end-to-end for the MISR task. FusionNet is trained from scratch while SISRNet and RegNet weights are initialized from the pretraining procedures. The concurrent optimization of all the network blocks allows SISRNet to finetune the image representations to facilitate the RegNet task that in turn finds the best registration to boost the efficiency of FusionNet.

3.3.3 Testing phase

The network architecture presented in the previous sections has been designed to deal with a fixed number N of LR images for a given scene. However, it might happen that more than N images are available and exploiting them could further boost the SR reconstruction performance. Therefore, during testing, one can perform multiple forward passes by using multiple subsets of the available images. Each subset will produce a different SR estimate and, in the end, all SR estimates are averaged. Notice that the estimates should be registered to each other so it is advisable to always use the same LR image as the reference in the network (e.g., one could choose the image with fewer masked pixels). One method to produce useful subsets when more than N LR images are available is to sort them by increasing number of masked pixels and then use a sliding window over N images to compute SR estimates. It must be remarked that the SR estimate quality degrades with increasing number of masked pixels. Also, the estimates are clearly not independent if some images are reused multiple times, but we found consistent gains on our test set, nevertheless.

Defining the optimal function to merge SR estimates or making the network independent of the number of input images could be studied in future research.

3.4 Experimental results and discussions

In this section we perform an experimental evaluation of DeepSUM, comparing it with several alternative approaches. Code and pretrained models are available online[1]. We first perform an ablation study to highlight the contribution given by RegNet to the overall network performance. Then, we assess the performance of alternative approaches.

3.4.1 Experimental setting

In the following experiments, we employ both the NIR and RED band datasets described in Sec. 3.1. We use 396 scenes for training and 170 for testing from the NIR band dataset and 415 for training and 176 for testing from the RED band dataset. Expanding the training set with more scenes should further improve performance as more variability can be captured by our model. Since DeepSUM is devised to work with a fixed size temporal dimension, we train the network using

[1] https://github.com/diegovalsesia/deepsum

the minimum number of images available for each scene, i.e., $N = 9$ images. When more images are available we select the 9 clearest images according to the masks. As a preprocessing step, all LR images are clipped to $2^{14} - 1$ since corrupted pixels with large values occur in the LR images throughout the PROBA-V dataset.

After the bicubic interpolation, each scene is a data-cube of size $9 \times 384 \times 384$, from which we extract a dataset with patches of size $9 \times 96 \times 96$. 100 random patches are extracted from each scene, resulting in a total of 38400 samples. The patches are extracted considering the available pixel masks: a patch is accepted only if at least 9 scene images are at least 70% clear and the HR image in the same coordinates is at least 85% clear. The amount of unreliable pixels is relaxed to keep as much information as possible from the original images at the cost of training with sub-optimal patches. Separate networks are trained for RED and NIR. The proposed network is trained for around 3000 epochs with a batch size of 8 for both RED and NIR.

The Adam optimization algorithm [84] is employed for training, with momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The learning rate λ is initialized to 5×10^{-6} for the whole network. We employ the Tensorflow framework to train the proposed network on a PC with 64-GB RAM, an Intel Xeon E5-2609 v3 CPU, and an Nvidia 1080Ti GPU. The exact number of network layers is shown in Fig. 3.1 and the number of filters is 64 everywhere except for the RegNet's first layer, which has 128 filters. In order to mitigate border effects, we use reflection padding in all 2D convolutions. Each layer in the network is followed by Leaky ReLU non-linearity, except for the last layer. Each layer in SISRnet and FusionNet is followed by an Instance Norm layer. Instance normalization [160] is used in place of Batch normalization layer to make the network training as independent as possible of the contrast and brightness differences among the input images. Finally, since the network produces a residual estimate R, we normalize $\bar{I}^{\mathrm{IRLR}}$ and x so that their difference gives a unit variance residual R, thus avoiding any scaling to be performed by the last layer of the network and improving convergence speed.

3.4.2 Quantitative results

The evaluation metric that we consider is a modified version of the PSNR (mPSNR), from which we derived the loss function described in Sec. 3.2.5

$$\mathrm{mPSNR} = \max_{u,v \in [0,6]} 20 \log \frac{2^{16} - 1}{\| x_{u,v} - (x_{\mathrm{crop}} + b) \|^2}. \tag{3.2}$$

The mPSNR computation is meant only for pixels that are not concealed both in the target HR image and in the reconstructed image. Similarly to the loss function during training, this metric has been devised to cope with the high sensitivity of the PSNR to biases in brightness and with the relative translation that the reconstructed image might have with respect to the target HR image. In this case the maximum mPSNR over all possible shifts is considered for evaluation. Note that, by design of the dataset, the maximum shift in the horizontal and vertical directions is equal to 6 pixels.

We remark that this metric was also used to evaluate submissions to the ESA challenge, where the score was computed as a ratio between the mPSNR of the submission and that of the baseline approach, average over all the held-out test set.

Ablation study

First, we want to assess the effectiveness of the sliding window procedure described in Sec.3.3.3 to account for more than 9 images for a given scene. Fig. 3.4 shows the mPSNR as function of the number of SR estimates used for computing the average. Notice that the mPSNR quickly saturates due to the lower quality of the images in the dataset (e.g., too many masked pixels). Nevertheless, averaging allow to achieve an mPSNR gain up to 0.3 dB over a single SR estimate on the NIR data and up to 0.2 dB on the RED data. All the following results have been obtained with a sliding factor equal to 5.

Then, we want to verify the effectiveness of the RegNet component of DeepSUM with respect to external registration of the images by means of cross correlation. This test should highlight the advantage of exploiting the feature space of the end-to-end trained network for the registration task. Hence, we compare two versions of our network:

- full network (SISRNet+RegNet+FusionNet);
- network without the RegNet block (SISRNet+FusionNet). We keep the registration filters but they are fixed to be a delta centered at the integer shift determined by maximum cross correlation on the ILR input images.

The full network outperforms the one without RegNet by 0.16 dB and 0.13 dB for the NIR and RED test sets, respectively, as shown in Table 3.1. This is a significant margin and it is due to the fact that an inaccurate registration can be an important source of error for the SR reconstruction.

On the other hand, the full network, being trainable end-to-end, is able to exploit the feature space produced by SISRNet to provide a more accurate registration and help FusionNet to perform the feature merging task. We remark that the full network and the reduced network have been trained independently.

Comparison to State-of-the-Art

We compare the proposed MISR technique to a number of alternatives based on deep learning and model-based methods:

1. single image bicubic interpolation with least masked image (Bicubic);

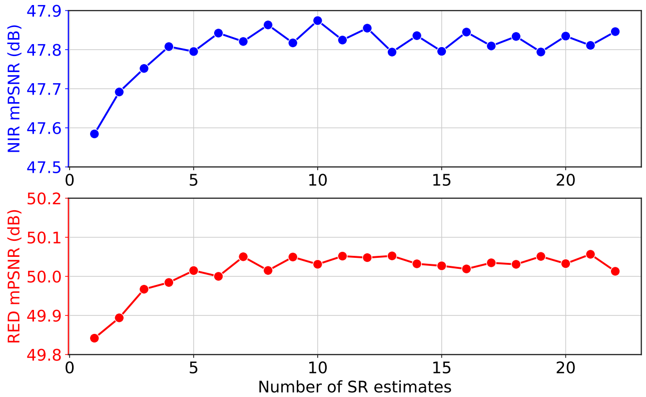

Figure 3.4: Effect of testing sliding window to deal with more than 9 LR images.

Table 3.1: Average mPSNR (dB) and SSIM - RegNet Performance.

	Proposed without RegNet	**Proposed with RegNet**
NIR	47.68 / 0.98519	**47.84 / 0.98578**
RED	49.87 / 0.99038	**50.00 / 0.99075**

2. averaged bicubic interpolated and registered images (Bicubic+Mean);

3. CNN-based SISR with least masked image;

4. CNN-based SISR method shared across multiple images followed by registration and averaging (SISR+Mean);

5. IBP [68];

6. BTV [40];

7. deep learning method based on simultaneous motion compensation and interpolation developed for video (dynamic upsampling filters (DUF) network) [74].

Table 3.2 reports the results of the comparison. It can be noticed that the proposed method outperforms all the other methods.

For all these methods, we followed the same procedure for the data preparation: bicubic interpolation and registration by phase correlation algorithm, except for DUF that computes its own registration. For MISR methods we averaged the 5 SR estimates produced by the sliding window method to ensure a fair comparison with the proposed technique.

Our IBP implementation takes as input an initial guess corresponding to our Bicubic+Mean baseline and the precomputed shifts related to the LR images using phase correlation algorithm. At each step, the LR images are estimated through

Table 3.2: Average mPSNR (db) and SSIM.

	Bicubic	Bicubic+Mean	IBP [68]	BTV [40]	SISR	SISR+Mean	DUF [74]	**DeepSUM**
NIR	45.05/0.97654	45.69/0.97782	45.96/0.97960	45.93/0.97942	45.56/0.97938	46.41/0.98166	47.06/0.98417	**47.84/0.98578**
RED	47.61/0.98474	47.91/0.98507	48.21/0.98648	48.12/0.98606	48.20/0.98704	48.71/0.98787	49.36/0.98948	**50.00/0.99075**

the forward (HR to LR) imaging model and the error with respect to the actual LR images is back projected to the current SR image. We can observe that IBP improves over the Bicubic+Mean baseline but its performance is ultimately limited by its inability to deal with a complex and unknown degradation model. BTV implementation takes the same initial guess and precomputed shifts as in IBP with the difference that at each iteration the cost function to minimize is a L1 norm plus the bilateral regularization term. BTV shows comparable performance with respect to IBP. BTV is slightly worse due to the L1 norm data fidelity that tends to be more robust to outliers but suboptimal with respect to the mPSNR metric. The deep learning models show marked improvements over the Bicubic+Mean baseline. We consider two deep learning baselines (SISR only and SISR+Mean) that use the SISRNet architecture with the addition of a final layer projecting from the feature space to the image space, a residual connection from the (IRLR) bicubic image(s) and an increased number of parameters to roughly match the number of parameters of the full proposed architecture in order to ensure a fair comparison. The SISR+Mean result has been obtained by averaging 9 SISR images. Notice that SISR+Mean does not train the network by showing the averaged image to the loss function; it just uses the pretrained SISR network on multiple images and averages its outputs. The reason behind this choice is to provide a reference result to reader who might be interested in taking a state-of-the-art off-the-shelf SISR model, apply it to multiple images and then avserage the results. The comparison between SISR+Mean and the SISR only method is meant to highlight the large gain brought by exploiting both the spatial and temporal correlations, even if the LR images of a specific scene are taken under different conditions and might be wildly different from one another in terms of contrast, brightness and landscape due to temporal variations. Also, notice that SISR only is unable to improve over the simple Bicubic+Mean MISR on the NIR data. Instead, the comparison between DeepSUM and the SISR+Mean method shows the improvement brought by the introduction of FusionNet, which can exploit the slow fusion via 3D convolutions to find the best way to merge the image representations.

Another method chosen for comparison is the recent DUF network [74]. This is one of the current state-of-the-art methods for video super-resolution. DUF network processes N frames in order to compute local pixel-dependent dynamic filters that are later applied on the central frame to increase its resolution and compensate motion. The network has a residual branch estimating a residual image to increase sharpness of the final SR image. The DUF network has been trained from scratch, maintaining the original structure and roughly the same number of learnable parameters with respect to our method for fair comparison. The only difference lies in using the loss function stated in Sec. 3.2.5 instead of the one used in the original paper (Huber loss). Moreover, we always considered the first one among the 9 input LR images as central frame. The performance is worse than our proposed method and we can deduce that it highly depends on the LR input image taken to apply the dynamic local filters. We cannot know in advance which is the LR image that is closer to the HR image due to change in brightness, landscape, weather, and clouds. Involving all the LR images for HR estimation is crucial to somehow average the differences across them and try to include as much information as possible in the final SR estimate.

PROBA-V challenge winning score

For completeness, we report the score achieved by DeepSUM on the unreleased test set of the PROBA-V challenge. DeepSUM achieved a score equal to 0.9474466476281652, computed as the average ratio between the mPSNR of ESA's baseline and the mPSNR of the submitted images, over both RED and NIR data in the held-out test set.

3.4.3 Qualitative results

We present a set of qualitative comparisons on the RED and NIR images of our PROBA-V test set.

First of all, Figs. 3.5 and 3.8 show the multitemporal variability among the LR images and between the LR set and the HR target for the NIR and RED bands, respectively.

Figs. 3.6 and 3.9 show a visual comparison between the SR images reconstructed by the various methods for the NIR and RED bands, respectively. It can be noticed that our proposed method produces visually more detailed images, recovering finer texture and sharper edges. In order to help visualization, Figs. 3.7 and 3.10 report the absolute difference between the HR target and the SR reconstructions for the various methods after registration and compensation for absolute brightness variations (as in the mPSNR computation).

3.5 Importance of the feature extractor

It is important to notice that SISRNet component is built with fully convolutional local layers and improving the feature extraction ability would reflect into the registration and fusion performance. In the following chapter, we investigate the possibility of integrating non-local features in the network, e.g., by using graph-convolutional architectures [164], a kind of convolution that draws from ideas in graph signal processing [144, 165].

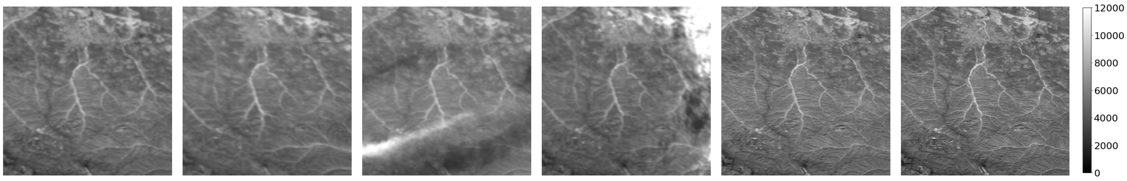

Figure 3.5: NIR band images (imgset0708). Left to right: 4 LR images, SR image reconstructed by DeepSUM and HR image.

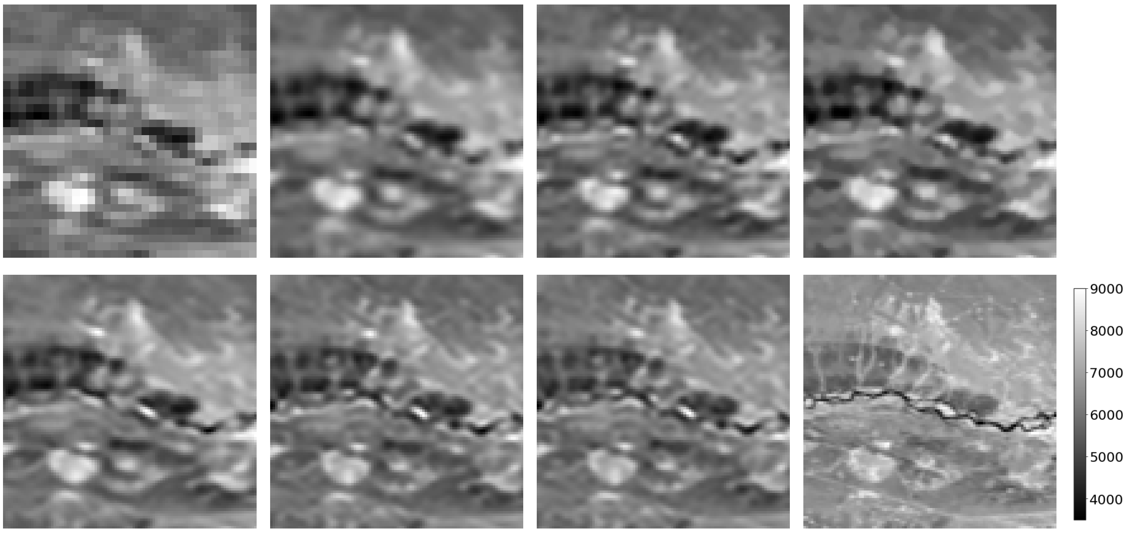

Figure 3.6: NIR band images (imgset0792). Top-Left to bottom-right: one among the LR images, Bicubic+Mean (47.71 dB / 0.98736), IBP (48.46 dB / 0.98919), BTV(48.12 dB / 0.98866), DUF (48.93 dB / 0.99028), proposed method without RegNet (50.71 dB / 0.99303), DeepSUM (50.82 dB / 0.99331), HR image.

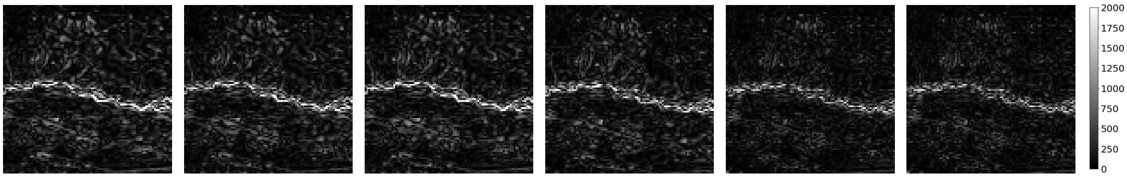

Figure 3.7: Absolute difference between SR image and HR image (NIR band). Left to right: Bicubic+Mean, IBP, BTV, DUF, proposed method without RegNet, DeepSUM.

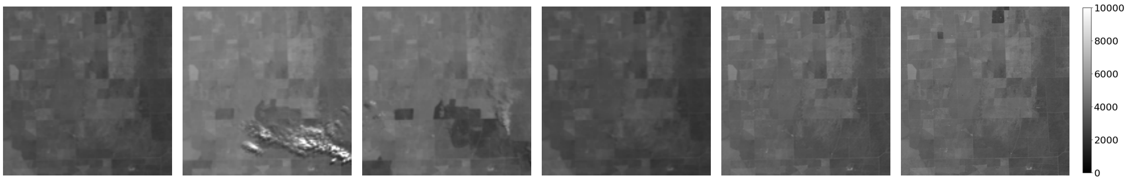

Figure 3.8: RED band images (imgset0103). Left to right: 4 LR images, SR image reconstructed by DeepSUM and HR image.

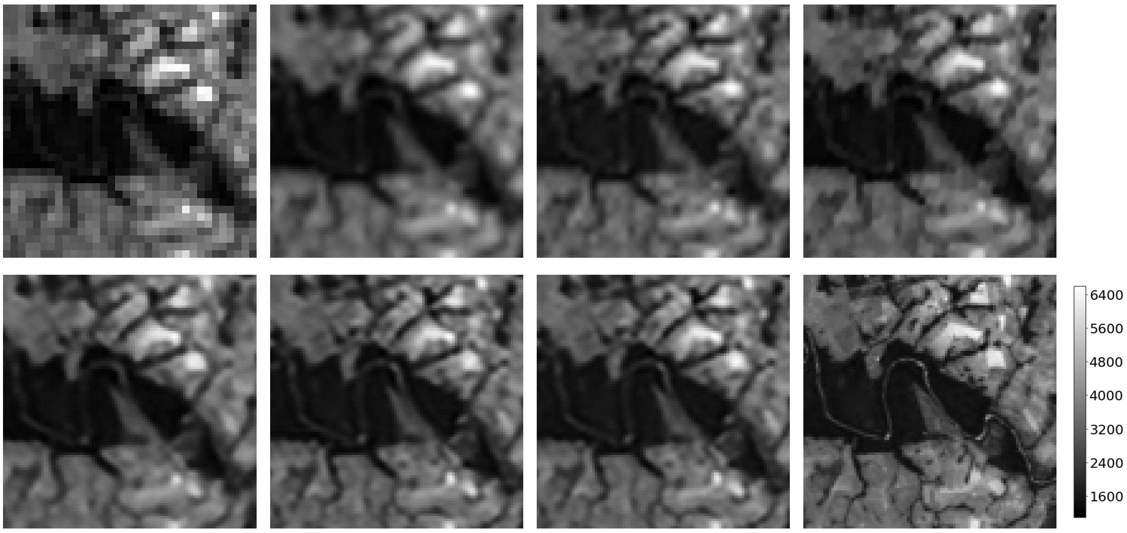

Figure 3.9: RED band images (imgset0184). Top-Left to bottom-right: one among the LR images, Bicubic+Mean (46.32 dB / 0.97897), IBP (46.52 dB / 0.97965), BTV (46.53 dB / 0.97983), DUF (47.64 dB / 0.98468), proposed method without RegNet (49.55 dB / 0.98886), DeepSUM (49.89 dB / 0.99041), HR image.

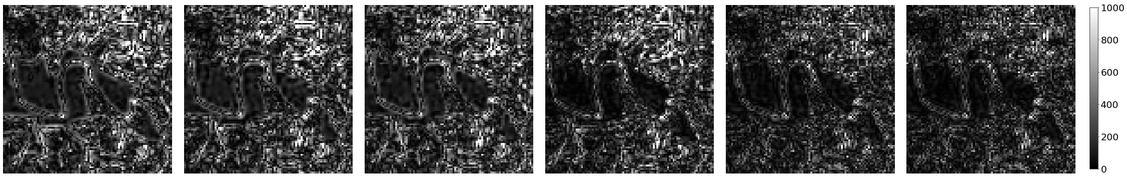

Figure 3.10: Absolute difference between SR image and HR image (RED band). Left to right: Bicubic+Mean, IBP, BTV, DUF, proposed method without RegNet, DeepSUM.

Chapter 4

DeepSUM++: Non-local Deep neural network for Super-resolution of Unregistered Multitemporal images

As winning method of the PROBA-V SR challenge [80], DeepSUM has proven to be very effective to enhance and combine satellite multitemporal images. One of the main flaw of the SR method described in the previous chapter is the feature extraction network. DeepSUM basically inverts blindly the whole forward model, solving both registration and enhancement of multiple unregistered images from the same scene through an end-to-end trainable CNN in a way that is robust to content variations. The registration and fusion tasks rely on the ability of feature extraction network (SISRNet) to generate meaningful features. The feature extraction network can be seen as a bottleneck for the subsequent tasks.

In this chapter we present an evolution of DeepSUM, showing how incorporating non-local information in a CNN allows to exploit self-similar patterns that provide enhanced regularization of the super-resolution problem.

In MISR literature, non-local techniques have also been introduced by Protter et al. in [132, 131], based on the non-local means filter [18] to improve the effectiveness of MISR methods. The idea of these methods is to exploit non-local structural similarity across spatially distant patches within an image. Other non local MISR techniques focus on improving regularization of the HR image reconstruction directly using non local priors.

Recently, non local-based deep learning techniques have spread in various fields

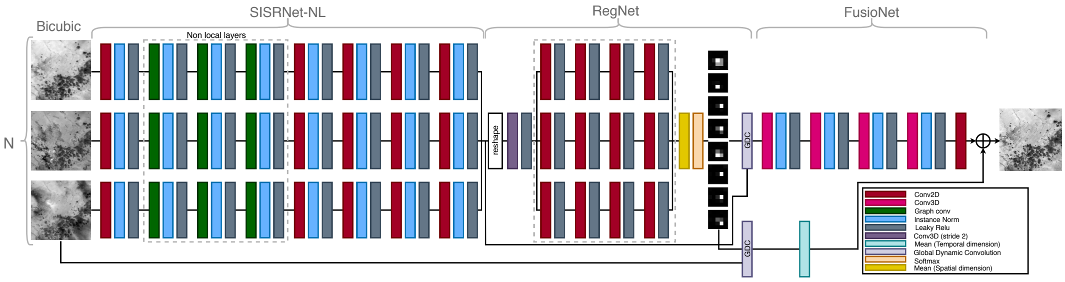

Figure 4.1: DeepSUM++ architecture. Graph-convolutional layers are used in SISRNet-NL.

of research, due to their ability to make use of more information taking into account the peculiar characteristics of the image to be recovered. Some works in the denoising literature [105, 163] attempt various approaches to define CNNs able to combine both spatially-neighboring as well as distant pixels. The graph convolution in [163] adopts a non-local convolution operator where a similarity graph is constructed connecting pixels whose feature vectors are close to each other.

In this following section, we introduce graph-convolutional layers in the DeepSUM architecture, in order to improve its learning capability and generate non-local feature maps in the hidden layers to solve a MISR problem on remote sensing imagery. The resulting DeepSUM++ architecture shows that the performance of DeepSUM is greatly enhanced by the non-local operations, improving upon the state of the art.

4.1 Proposed method

As in the previous chapter, we want to solve a MISR problem in a setting with multitemporal LR image acquisitions. This is characterized by variations in scene content over multiple acquisitions due to weather or human activities. Moreover, the absolute image brightness may vary among LR images as well as between the reference HR image and the LR set. Finally, the LR images are not registered with each other, and we assume that the geometric disparity is only translational.

DeepSUM addressed this problem with a CNN composed of three main building blocks: feature extraction (SISRNet), registration in feature space (RegNet) and fusion to obtain a single HR reconstruction (FusionNet). All blocks are based on classical 2D or 3D convolutions, so only local information is exploited to obtain the final HR reconstruction. DeepSUM is optimized in an end-to-end fashion allowing the registration and fusion task to leverage the feature learning capabilities of SISRNet that is in fact a crucial component.

DeepSUM++ builds upon the DeepSUM architecture and introduces non-local operations in the SISRNet block, which allows SISRNet to compute more powerful high-dimensional image representations that considerably improve the quality of the subsequent tasks by relying on more informative features. An overview of the network is shown in Fig. 4.1.

The first block, called SISRNet-NL (non local), is a simple SISR network without the output projection to a single channel, where N bicubically interpolated LR (ILR) images are processed independently. The weights are shared along the temporal dimension, i.e., all the N ILR images go through the same operators. Overall, it acts as a feature extractor exploiting both local and non-local spatial correlations to improve upon the initial bicubic interpolation. In order to exploit non-local spatial correlation, traditional convolution is augmented with a graph convolution operator, which adds to the receptive field of a pixel a predefined number of non-local pixels whose feature vectors are close to the feature vector of the current pixel.

More formally, the graph-convolutional layer takes as input the feature vectors $\mathbf{H}^l \in \mathbb{R}^{F^l \times rHrW}$, i.e., the high-dimensional representations of the image pixels at layer l, and the adjacency matrix of a graph connecting image pixels. The graph structure is constructed as a K-nearest neighbor graph where each pixel is connected to the K pixels whose feature vectors are closest in terms of Euclidean distance, within a search window. The 8 local neighbors are excluded from this search as they already contribute to the local convolution. The graph-convolutional layer is composed of two components both taking as input the feature vectors $\mathbf{H}^l$. A classic 2D convolution aggregates the local neighbors through 3×3 filters and an edge-conditioned convolution (ECC) [145, 163] aggregates the feature vectors of the non-local pixels. For each pixel i, the ECC computes the output feature vector $\mathbf{H}_i^{l+1,\mathrm{NL}} \in \mathbb{R}^{F^{l+1}}$ as follows:

$$\mathbf{H}_i^{l+1,\mathrm{NL}} = \sum_{j \in \mathcal{N}_i^l} \gamma^{l,j \rightarrow i} \frac{\mathcal{F}_{w^l}^l(\mathbf{d}^{l,j \rightarrow i})\mathbf{H}_j^l}{|\mathcal{N}_i^l|} = \sum_{j \in \mathcal{N}_i^l} \gamma^{l,j \rightarrow i} \frac{\Theta^{l,j \rightarrow i}\mathbf{H}_j^l}{|\mathcal{N}_i^l|}$$

where $\mathcal{F}_{w^l}^l : \mathbb{R}^{F^l} \rightarrow \mathbb{R}^{F^{l+1} \times F^l}$ is a fully-connected network, parameterized by w^l, that takes as input the differences between feature vectors $\mathbf{d}^{l,j \rightarrow i} = \mathbf{H}_j^l - \mathbf{H}_i^l$ and outputs a weight matrix, and $\mathcal{N}_i^l$ is the set of non-local neighbors of pixel i. $\gamma^{l,j \rightarrow i}$ is a non-learnable scalar edge-attention term to underweight the edges between pixels with distant feature vectors for training stabilization. This term is computed as:

$$\gamma^{l,j \rightarrow i} = \exp(-\|\mathbf{d}^{l,j \rightarrow i}\|_2^2 / \delta)$$

where δ is a hyperparameter.

Finally, the local and the non-local contributions are averaged to estimate the output feature vector:

$$\mathbf{H}_i^{l+1} = \frac{\mathbf{H}_i^{l+1,\mathrm{NL}} + \mathbf{H}_i^{l+1,\mathrm{L}}}{2} + \mathbf{b}^l$$

where $\mathbf{H}_i^{l+1,\mathrm{L}}$ is the 2D convolution output and b^l is a bias. We refer the reader to [163] for more details on the advantages of this definition of graph convolution. Depending on the desired computational complexity one may want to use graph convolution for all the layers in SISRNet or just for a subset.

The second network block, called RegNet, dynamically estimates a set of 2D filters using the N higher dimensional image representations produced by the SISRNet-NL block to register them to each other. Handling registration within an end-to-end trainable network enables the generation of adaptive filters for disparity compensation.

The third block, called FusionNet, merges the registered image representations in the feature space in a "slow" fashion, i.e., by exploiting a sequence of 3D convolutional operations with small kernels. This block allows the network to learn the best space to decouple image features that are relevant to the fusion from irrelevant features and to construct the best function to exploit spatio-temporal correlations. The output of this block is a single super-resolved image.

Finally, DeepSUM++ employs the same global input-output residual connection handled as in DeepSUM, as well as the same modified Euclidean loss function in Eq. (3.1).

Table 4.1: Average mPSNR (dB) and SSIM.

	Bicubic+Mean	IBP [68]	BTV [40]	SISR+Mean	DUF [74]	DeepSUM [120]	**DeepSUM++**
NIR	45.69/0.97782	45.96/0.97960	45.93/0.97942	46.41/0.98166	47.06/0.98417	47.84/0.98578	**47.93/0.98620**
RED	47.91/0.98507	48.21/0.98648	48.12/0.98606	48.71/0.98787	49.36/0.98948	50.00/0.99075	**50.08/0.99118**

4.2 Experimental results and discussions

To validate DeepSUM++ we compare its performance with that of DeepSUM [120] and the same set of MISR methods used in the previous chapter to assess DeepSUM performance: averaged bicubic interpolated and registered images (Bicubic+Mean), CNN-based SISR method shared across multiple images followed by registration and averaging (SISR+Mean), IBP [68], BTV [40] and dynamic upsampling filters (DUF) network [74]. For all these methods, we followed the same procedure for the data preparation explained in section 3.4.2: bicubic interpolation and registration by phase correlation algorithm, except for DUF and DeepSUM that compute their own registration.

4.2.1 Experimental setting and training process

In the following experiments, we performed the same pre-processing and data preparation steps as for DeepSUM for both NIR and RED band images. The images used to construct the two datasets have been captured by the PROBA-V satellite

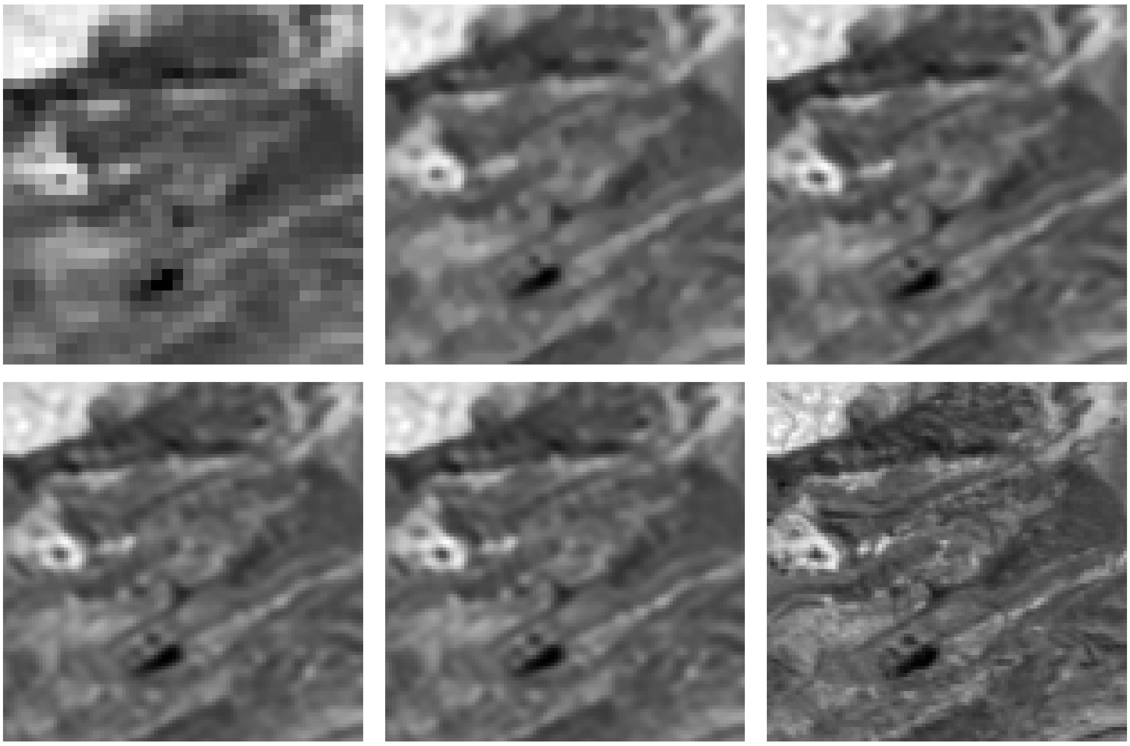

Figure 4.2: NIR band images (imgset1144). Top-left to bottom-right: one among the LR images, BTV(47.37 dB / 0.98284), DUF (48.02 dB / 0.98620), DeepSUM (48.74 dB / 0.98844), DeepSUM++ (49.46 dB / 0.98943), HR image.

and are part the ESA challenge dataset [80]. The size of the collected images is 128 × 128 and 384 × 384 for the LR and HR data respectively.

Before training DeepSUM++ as a whole, SISRNet-NL is pretrained by setting up a pure SISR problem (single input image) where an additional projection layer is added at the end, in order to turn the high-dimensional feature space into a single-channel image. SISRNet-NL with the final projection layer is trained with the same loss function in Eq. (3.1). DeepSUM++ is trained for 300 epochs with a batch size of 4, with separate models for NIR and RED. SISRNet-NL is initialized from the pretraining while FusionNet and RegNet weights are initialized from the final DeepSUM model.

We train DeepSUM++ using the minimum number of images available for each scene, i.e., $N = 9$ images.

The exact number of network layers is shown in Fig. 4.1 and the number of features is 64 everywhere except for the RegNet's first layer, which has 128 filters. Three graph convolutional layers are used in SISRNet-NL.

4.2.2 Quantitative and qualitative results

The evaluation metric that we consider is the mPSNR metric (3.2) described in Sec. 3.4.2. The validation has been performed over the same NIR and RED test sets used for DeepSUM [120], using the sliding window procedure to use 13 images per scene in the testing process. Table 4.1 reports the results of the comparison. It can be noticed that the proposed method outperforms all the other methods. In particular, DeepSUM++ outperforms DeepSUM by 0.09 dB for NIR and 0.08 dB for RED band. To ensure a fair comparison we have retrained DeepSUM following the same procedure used for DeepSUM++ for the same number of iterations.

These quantitative results are accompanied by a qualitative comparison in Fig. 4.2. It can be noticed that our proposed method produces visually more detailed images, recovering finer texture and sharper edges.

Chapter 5

Speckle2Void: Deep Self-Supervised SAR Despeckling with Blind-Spot Convolutional Neural Networks

In the second part of this thesis we present a work dealing with remote sensing images captured by radar technology satellites instead of optical satellites. We continue to focus on the dataset availability issue as in the previous works, noting that in SAR remote sensing applications the clean images are impossible to acquire. We will solve a denoising inverse problem known as SAR despeckling. Much like super-resolution is performed as a pre-processing step to enhance the outcome of downstream tasks such as classification or object detection, also despeckling is a crucial preliminary step in scene analysis algorithms as information extraction from SAR images is heavily impaired by speckle noise.

The recent success of deep learning envisions a new generation of despeckling techniques that could outperform classical model-based methods.

As mentioned in chapter 2, current deep learning approaches to despeckling require supervision for training, whereas clean SAR images are impossible to obtain. In the literature, this issue is tackled by resorting to either synthetically speckled optical images, which exhibit different properties with respect to true SAR images, or multi-temporal SAR images, which are difficult to acquire or fuse accurately.

During the last year, significant advances have been made on deep learning approaches to denoising proving, under certain assumptions, to be a valid alternative when it is not possible to have access to clean images. Despite these methods do not require ground-truth, they show that it is possible to reach performance close to that of fully-supervised methods. These new self-supervised denoising methods have been developed on natural images, but it is quite clear that extending them to the SAR context is appealing, as significant speckle noise is always present in SAR acquisitions. Noise2Noise [95] proposed to use pairs of images with the same content but independent noise realizations. The main drawback of this method is the difficulty of accessing multiple versions of the same scene with independently drawn noise realizations. Yuan et al. [186] presented a despeckling method based on the idea of Noise2Noise [95], but still simulating speckle on a dataset based on ImageNet. Ma et al. [112] devised a method based on the Noise2Noise scheme, requiring multi-temporal SAR images to train the network. They coped with the possible temporal variations by introducing a similarity measure in order to weight the contribution of each pixel pair in the loss.

Noise2Void [85] and Noise2Self [10] further relax the constraints on the dataset, requiring only a single noisy version of the training images, by introducing the concept of blind-spot networks. Assuming spatially uncorrelated noise, and excluding the center pixel from the receptive field of the network, the network learns to predict the value of the center pixel from its receptive field by minimizing the ℓ_2 distance between the prediction and the noisy value. The network is prevented from learning the identity mapping because the pixel to be predicted is removed from the receptive field. Notice that this is also the reason for the uncorrelated noise assumption. The blind-spot scheme used in Noise2Void [85] is carried out by a simple masking method that hides a small subset of noisy pixels at a time, processing the entire image to learn to reconstruct a small amount of pixels. Laine et al. [89] devised a novel blind-spot CNN architecture capable of processing the entire image at once, increasing the efficiency. They also introduced a Bayesian framework to include noise models and priors on the conditional distribution of the blind spot given the receptive field.

Inspired by these works, in this chapter we present Speckle2Void (S2V), a self-supervised Bayesian despeckling framework that enables direct training on real SAR images. Our method bypasses the problem of training a CNN on synthetically-speckled optical images, thus avoiding any domain gap and enabling learning of features from real SAR images. It also avoids the inherent difficulty in constructing multitemporal datasets, as done in [25]. Our main contributions can be summarized as follows:

- we formulate a Bayesian model to characterize the speckle and the prior distribution of pixels in the clean SAR image, conditioned on their neighborhoods;
- we propose an improved version of the blind-spot CNN architecture in [89] and a regularized training procedure with a variable blind-spot shape in order to account for the autocorrelation of the speckle process;

- we present two versions of Speckle2Void: a local version with classical convolutional layers and a non local version to incorporate information from both spatially-neighboring as well as distant pixels to exploit self-similarity, albeit at higher computational complexity;
- we achieve remarkable despeckling performance, showing how our self-supervised approach is better than model-based techniques, close to the deep learning methods requiring supervised training on synthetic images and superior to them on real SAR data.

A preliminary version of this work appeared in [16], showing the basic principles of the proposed approach. This work significantly expands the treatment with improvements on network modeling, on the loss function and on the training procedure. In particular, it solves the problem of the residual granularity in the despeckled images in [16], by showing the importance of properly decorrelating the speckle process and carefully designing the blind-spot shape.

5.1 Self-supervised denoising CNN: background from a probabilistic perspective

CNN denoising methods estimate the clean image by learning a function that takes each noisy pixel and combines its value with the local neighboring pixel values (receptive field) by means of multiple convolutional layers interleaved with non-linearities. Taking this from a statistical inference perspective, a CNN is a point estimator of $p(x_i|y_i, \Omega_{y_i})$, where x_i is the i^{th} clean pixel, y_i is the i^{th} noisy pixel and Ω_{y_i} represents the receptive field composed of the noisy neighboring pixels, excluding y_i itself. Noise2Void and Noise2Self predict the clean pixel x_i by relying solely on the neighboring pixels and using y_i as a noisy target. By doing so, the CNN learns to produce an estimate of $\mathbb{E}_{x_i}[x_i|\Omega_{y_i}]$, using the ℓ_2 loss when in presence of Gaussian noise. The drawback of these methods is that the value of the noisy pixel y_i is never used to compute the clean estimate.

The Bayesian framework devised by Laine et al. [89] explicitly introduces the noise model $p(y_i|x_i)$ and conditional pixel prior given the receptive field $p(x_i|\Omega_{y_i})$ as follows:

$$p(x_i|y_i, \Omega_{y_i}) \propto p(y_i|x_i)p(x_i|\Omega_{y_i}).$$

The role of the CNN is to predict the parameters of the chosen prior $p(x_i|\Omega_{y_i})$. The denoised pixel is then obtained as the posterior mean (MMSE estimate), i.e., it seeks to find $\mathbb{E}_{x_i}[x_i|y_i, \Omega_{y_i}]$. Under the assumption that the noise is pixel-wise i.i.d., the CNN is trained so that the data likelihood $p(y_i|\Omega_{y_i})$ for each pixel is maximized. The main difficulty involved with this technique is the definition of a suitable prior distribution that, when combined with the noise model, allows for

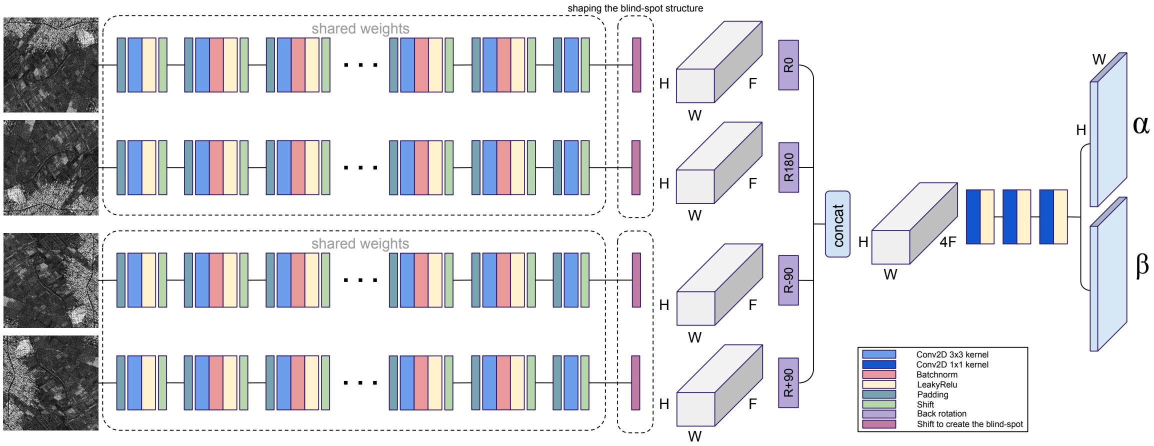

Figure 5.1: Speckle2Void takes as input four rotated versions of an image. Each branch processes a specific rotation to compute the receptive field in a specific direction. Subsequently, the four half-plane receptive fields are shifted to achieve the desired blind-spot shape, rotated back and concatenated. As last, a series of 2D convolutions with kernel 1x1 are used to fuse the four receptive fields and generate the parameters of the inverse gamma for each pixel.

closed-form posterior and likelihood distributions. We also remark that while imposing a handcrafted distribution as $p(x_i|\Omega_{y_i})$ may seem very limiting, it is actually not since i) that is the *conditional* distribution given the receptive field rather than the raw pixel distribution, and ii) its parameters are predicted by a powerful CNN on a pixel-by-pixel basis.

5.2 Proposed method

Following the notation in Sec. 5.1, this section presents the Bayesian model we adopt for SAR despeckling, the training procedure and the blind-spot architecture. A summary is shown in Figs. 5.1 and 5.2.

5.2.1 Model

We consider the multiplicative SAR speckle noise model: $y_i = n_i x_i$, where x represents the unobserved clean image in intensity format and n the spatially uncorrelated multiplicative speckle. Concerning noise modeling, one common assumption is that it follows a Gamma distribution with unit mean and variance $1/L$ for an L-look image and has the following probability density function:

$$p(n) = \frac{1}{\Gamma(L)} L^L n^{L-1} e^{-Ln}$$

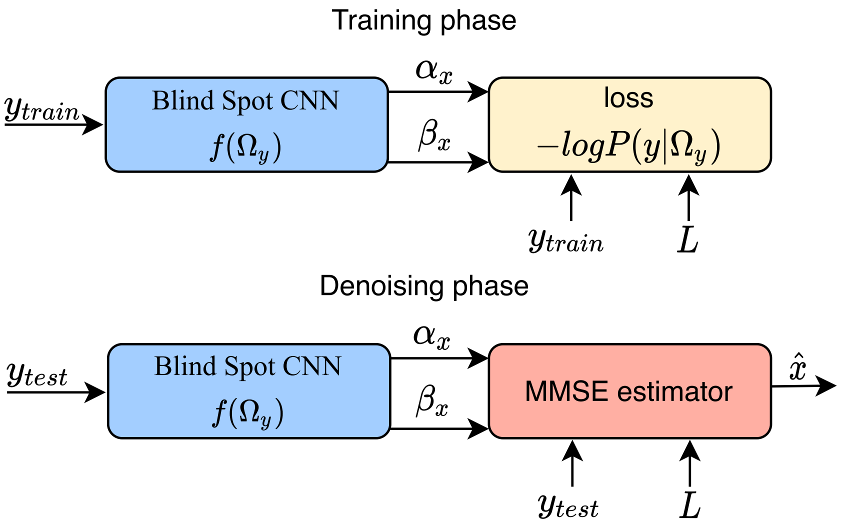

Figure 5.2: Scheme depicting the training and the testing phases. During training phase the blind-spot network is trained to minimize the negative log of the noisy data likelihood to estimate α_{x_i} and β_{x_i} for each pixel. In testing phase, the MMSE estimator generates the final clean image, combining together the parameters of the pixel prior, the noisy pixel and the parameter of noise distribution.

where $\Gamma(.)$ denotes the Gamma function and $n \geq 0$, $L \geq 1$. The aim of despeckling is to estimate intensity backscatter x from the observed intensity return y.

We model the conditional prior distribution given the receptive field as an inverse Gamma distribution with shape α_{x_i} and scale β_{x_i}:

$$p(x_i|\Omega_{y_i}) = \text{inv}\Gamma(\alpha_{x_i}, \beta_{x_i}),$$

where α_{x_i} and β_{x_i} depend on Ω_{y_i}, since they are the outputs of the CNN at pixel i. Assuming the noise to be Gamma-distributed, i.e., $n_i \sim \Gamma(L, L)$ being both the scale and rate parameters equal to L, then by the scaling property of the Gamma distribution, we obtain that $y_i|x_i \sim \Gamma(L, \frac{L}{x_i})$. We can now write the unnormalized posterior distribution as:

$$\begin{aligned} p(x_i|y_i, \Omega_{y_i}) &\propto p(y_i|x_i)p(x_i|\Omega_{y_i}), \\ p(x_i|y_i, \Omega_{y_i}) &\propto \frac{1}{\Gamma(L)}\left(\frac{L}{x_i}\right)^L y_i^{L-1} e^{-\frac{L}{x_i}y_i} \frac{\beta_{x_i}^{\alpha_{x_i}}}{\Gamma(\alpha_{x_i})} \frac{e^{-\frac{\beta_{x_i}}{x_i}}}{x^{\alpha_{x_i}+1}}, \\ &\propto \frac{e^{-\frac{Ly_i+\beta_{x_i}}{x_i}}}{x^{\alpha_{x_i}+L+1}} \end{aligned}$$

For the chosen prior and noise models, the posterior distribution has still the form of an inverse Gamma with shape $L + \alpha_{x_i}$ and scale $\beta_{x_i} + Ly_i$:

$$p(x_i|y_i, \Omega_{y_i}) = \mathrm{inv}\Gamma(L + \alpha_{x_i}, \beta_{x_i} + Ly_i). \tag{5.1}$$

The chosen prior distribution and noise model allow to conveniently obtain the marginal distribution of the noisy training data $p(y_i|\Omega_{y_i})$ in close form by solving the following integral:

$$p(y_i|\Omega_{y_i}) = \int p(y_i|x_i)p(x_i|\Omega_{y_i})dx_i \tag{5.2}$$

The probability density obtained by solving this integral is known as G_I^0, and has the following expression:

$$p(y_i|\Omega_{y_i}) = G_I^0 = \frac{L^L y_i^{L-1}}{\beta_{x_i}^{-\alpha_{x_i}} \mathrm{Beta}(L, \alpha_{x_i})(\beta_{x_i} + Ly_i)^{L+\alpha_{x_i}}}, \tag{5.3}$$

According to [43], the G_I^0 distribution is a very general model, that is particularly suitable to model the observed intensity return y of SAR images and able to accommodate different types of areas: from extremely heterogeneous scenes such as urban areas, to extremely homogeneous scenes such as deforested area as α_{x_i} and β_{x_i} become larger.

5.2.2 Training

The training procedure learns the weights of the blind-spot CNN. The blind-spot CNN processes the noisy image to produce the estimates for parameters α_{x_i} and β_{x_i} of the inverse gamma distribution $p(x_i|\Omega_{y_i})$ used as prior. It is trained to minimize the negative log likelihood $p(y_i|\Omega_{y_i})$ for each pixel, so that the estimates of α_{x_i} and β_{x_i} fit the noisy observations.

As stated at the beginning of this chapter, training a blind-spot network requires noise to be spatially uncorrelated, so that the CNN is prevented from exploiting the latent correlation to reproduce the noise in the blind spot. While many works assume that SAR speckle is uncorrelated, the SAR acquisition and focusing system has a point spread function (PSF) that correlates the data. To cope with this, we apply a pre-processing whitening procedure, such as the one proposed by Lapini et al. [90] to decorrelate the speckle. In [90], the authors use the complex SAR data after focusing to estimate the PSF of the system and approximately invert it, achieving the desired decorrelation and showing that this step boosts the performance of any despeckling algorithm relying on the uncorrelated speckle assumption. The point targets present in the SAR image, due to man-made features or edges, are detected and filtered before the decorrelation procedure and subsequently placed back, in order to preserve them. This whitening step is especially critical in the proposed approach due to the high capacity of neural networks to overfit even random patterns.

However, perfect decorrelation is in practice impossible and the residual correlation could limit the performance of the blind-spot CNN. For this reason, we modify the basic design of the blind-spot CNN by Laine et al. [89], and introduce a variable-sized blind spot. If noise correlation cannot be removed by other means, one could consider the width of the autocorrelation function of the noise and set a blind spot that is wide enough to cover the peak of the autocorrelation. This ensures that the receptive field contains a negligible amount of information for the reproduction of the noise component of the pixel to be estimated. However, this inevitably reduces the amount of information that can be exploited by the CNN, as the content of the immediate neighbors of a pixel is the most similar to that of the pixel itself. Therefore, a larger blind spot trades off more effective noise suppression with a less accurate (appearing as blurry) prediction.

To achieve a finer control about this trade-off, we devise a regularized training procedure that allows to tune the degree of reliance of the CNN on the immediate neighbors, leading to an improvement of the high frequency details in the denoised image, while still suppressing most of the noise correlation. During training, we randomly alternate, with predefinied probabilities, a 1 × 1 blind spot and a larger blind spot that can have arbitrary shape to match the noise autocorrelation. This mechanism allows the network weights to learn how to partially exploit the neighboring pixels belonging to the larger blind-spot but at the same time not to rely too much on them, in order to prevent from overfitting the noise components. During testing, a 1 × 1 blind spot is used, thus only excluding the center pixel, and exploiting the closest neighbors. Due to their weak training, these neighbors allow to recover some high frequency image content, which is the stronger signal present, while not being able to exploit the weaker correlations in the noise. We refer the reader to Sec. 5.3.4 for the details on the chosen parameter settings and the specific SAR dataset used for training.

5.2.3 Testing

In testing, the blind-spot CNN processes the noisy SAR image to estimate α_{x_i} and β_{x_i} for each pixel. The despeckled image is then obtained through the MMSE estimator, i.e., the expected value of the posterior distribution in Eq. (5.1), as:

$$\hat{x}_i = \mathbb{E}[x_i|y_i, \Omega_{y_i}] = \frac{\beta_{x_i} + Ly_i}{L + \alpha_{x_i} - 1}.$$

Notice that this estimator combines both the per-pixel prior estimated by the CNN and the noisy observation.

5.2.4 Loss function

As mentioned in Sec. 5.2.2, the blind-spot CNN is trained by minimizing the negative log likelihood of the noisy observations, based on the estimated parameters α_{x_i} and β_{x_i} of the prior. Moreover, we incorporate a total variation (TV) component, computed over the posterior image, to further promote smoothness. Our final loss function is as follows:

$$l = -\sum_i \log p(y_i|\Omega_{y_i}) + \lambda_{TV} TV(\hat{x})$$

where $p(y_i|\Omega_{y_i})$ is defined in Eq. (5.3), the TV term is the anisotropic version of the total variation $TV(\hat{x}) = \sum_{i,j} |\hat{x}_{i+1,j} - \hat{x}_{i,j}| + |\hat{x}_{i,j+1} - \hat{x}_{i,j}|$ and λ_{TV} is a hyperparameter to tune the desired degree of smoothness.

5.2.5 Blind-spot architecture

The rationale behind the blind-spot network is to introduce a pixel-sized hole in the receptive field, in order to prevent the network from learning the identity mapping. Our model is built upon the architecture by Laine et al.[89], who designed a CNN architecture to naturally account for the blind spot in the receptive field, thus increasing training efficiency. They cleverly implemented shift and padding operations on the feature maps at each layer, in order to limit the receptive field to grow in a specific direction, excluding the center pixel from the computation. Their architecture is composed of four different CNNs, each responsible of limiting the receptive field to extend in a single direction by means of shift and padding operations on the feature maps at each layer. The four subnetworks produce four limited receptive fields that extend strictly above, below, leftward and rightward of the target pixel. In order to reduce the number of trainable parameters, they feed four rotated versions of each input image to a single network that computes the receptive field in a specific direction. The four limited receptive fields are finally combined through a series of 2D convolutions with 1×1 filters, ensuring no further expansion of the receptive field. To perform this particular computation, classical 2D convolutional layers are used but their receptive field is limited to grow in a direction by shifting the feature map in the opposite direction by an offset of $\lfloor k/2 \rfloor$ pixels, where $k \times k$ is the kernel size, before performing the convolution operation. At the end of the network, each of the four limited receptive fields still contains the center row/column, so the center pixel as well. To exclude it, the feature maps are shifted by one pixel before combining them.

An overview of the blind-spot network used by Speckle2Void is shown in Fig. 5.1. Speckle2Void modifies the basic architecture by Laine et al.[89] described above to allow more flexibility in shaping the blind-spot. In principle, if the final shift applied to each of the four directional receptive fields was different from one another, we would be able to control the size of the blind spot in each direction.

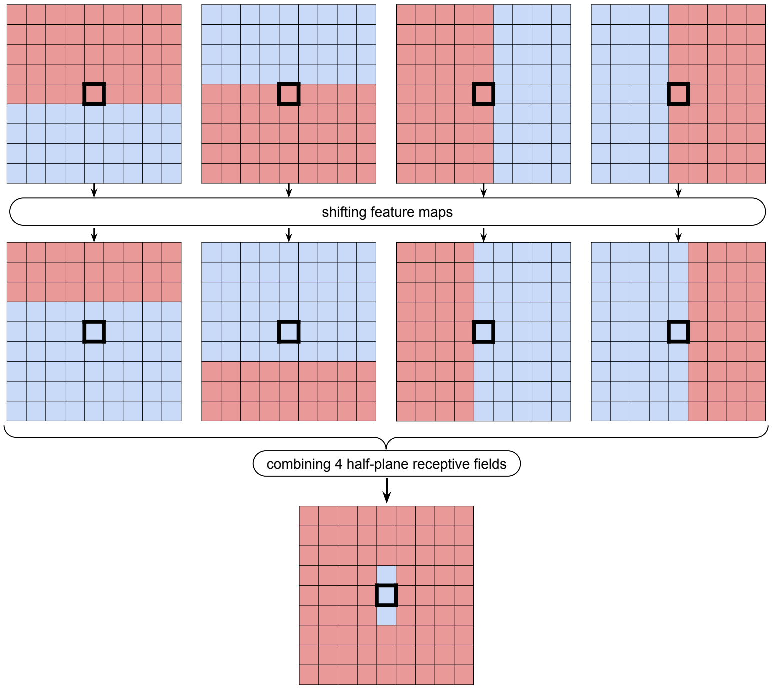

Figure 5.3: Visual depiction of the operations performed by the blind-spot network to constrain the receptive field related to the center pixel to exclude the center pixel itself and two pixels in the vertical direction. The first row represents, in pink color, the four limited receptive fields extending in four directions. As the center pixel is still included in the receptive fields, each feature map is shifted in the opposite direction with respect to the growing direction of the receptive field. This shifting operation allows the pink pixels in the second row to be the new receptive fields associated to the center pixel. The shift is 1 in azimuth direction and 2 in the range one. The last row represents the final receptive field, related to the center pixel, as the result of a combination of the four receptive fields depicted in the second row.

In SAR images, the azimuth and range directions may exhibit different statistical properties, including the residual noise autocorrelation. We therefore account for that by only sharing weights between the two branches processing the receptive field oriented as the azimuth or range directions, instead of sharing them for all four branches as in [89]. Furthermore, as shown in Fig. 5.3, Speckle2Void can apply one shift in the azimuth direction and a different shift in the range one.

5.2.6 Non local convolutional layer and its adaptation to blind-spot networks

The blind-spot CNN used by Speckle2Void also comes in two versions. The "local" version of Speckle2Void is composed by a series of classic 2D convolutional layers, each followed by Batch normalization [67] and a Leaky-ReLU non-linearity. The "non local" version adds several non local layers, as defined in [105]. Such layers introduce a dynamic weighted function of the feature vectors that help retrieving more information from a wider image context. While the "local" version of Speckle2Void employs classical 2D convolutions, so only local information is exploited at each layer, non local layers leverage non local structural similarity across spatially distant patches within an image, enabling the CNN to combine both spatially-neighboring as well as distant pixels. In particular, non local self-similarity can be effective in recovering the information hidden by the blind spot, without encountering the problem of noise correlation as it is drawn from spatially-distant areas. However, exploiting non-locality incurs a significant penalty in terms of computational cost.

The non local module proposed by NLRN [105] uses a soft block matching approach and applies the Euclidean distance with linearly embedded Gaussian kernel as distance metric. The rational behind this module is to perform a weighted combination of all the feature vectors in a patch (search window) to compute the new feature vector at its center, where the used weights dynamically depend on the similarity between the center feature vector and all the others within the patch, and repeat it for each feature vector in the feature map. This non local layer is designed to work in a traditional CNN architecture, and requires introducing a masking technique to adapt it to the blind-spot architecture used by Speckle2Void. In [105], the linear embeddings are defined as follows:

$$\Phi(\mathbf{H}^l_{ij}) = \phi(\mathbf{H}^l_{ij}, \mathbf{H}^l_{p_{ij}}) = \exp\{\theta(\mathbf{H}^l_{ij})\psi(\mathbf{H}^l_{p_{ij}}))\}, \forall i, j,$$
$$\theta(\mathbf{H}^l_{ij}) = \mathbf{H}^l_{ij}W_\theta, \psi(\mathbf{H}^l_{p_{ij}}) = \mathbf{H}^l_{p_{ij}}W_\psi, G(\mathbf{H}^l_{ij}) = \mathbf{H}^l_{p_{ij}}W_g, \forall i, j.$$

$\Phi(\mathbf{H}^l_{ij})$ represents the distance metric to encode the non local correlation between the feature vector in position i, j and each neighbours in the patch $\mathbf{H}^l_{p_{ij}}$ at layer l. $\Phi(\mathbf{H}^l_{ij})$ has shape $1 \times q \times q$ where $q \times q$ denotes the spatial size of the neighbour patch centered at pixel i, j. $\theta(\mathbf{H}_{ij})$ represents the embedding associated to the feature vector in position i, j with shape $1 \times l$ where l is the number of features. $\psi(\mathbf{H}_{p_{ij}})$ represents the embeddings associated to each feature vector in the neighbour patch p centered at i, j with shape $q \times q \times m$ where m is the number of features. The transformation weights W_θ, W_ψ, W_g used to compute the embeddings have shape $m \times l$, $m \times l$, $m \times m$ respectively, and are trainable weights. We add a masking operation to the non local layer proposed in [105] and the final formulation is obtained as:

$$\mathbf{H}^{l+1}_{ij} = \frac{1}{\delta'(\mathbf{H}^l_{ij})}(M_i \odot \exp\{\mathbf{H}^l_{ij}W_\theta W^T_\psi \mathbf{H}^{lT}_{p_{ij}})\})\mathbf{H}^l_{p_{ij}}W_g, \forall i, j,$$

where $\delta'(\mathbf{H}^l_{ij}) = \sum_{p_{ij}} M_i \odot \phi(\mathbf{H}^l_{ij}, \mathbf{H}^l_{p_{ij}})$ is the normalization factor, $\mathbf{H}^{l+1}_{ij}$ is the output feature vector at spatial location i, j and M_i is a mask, associated to row i,

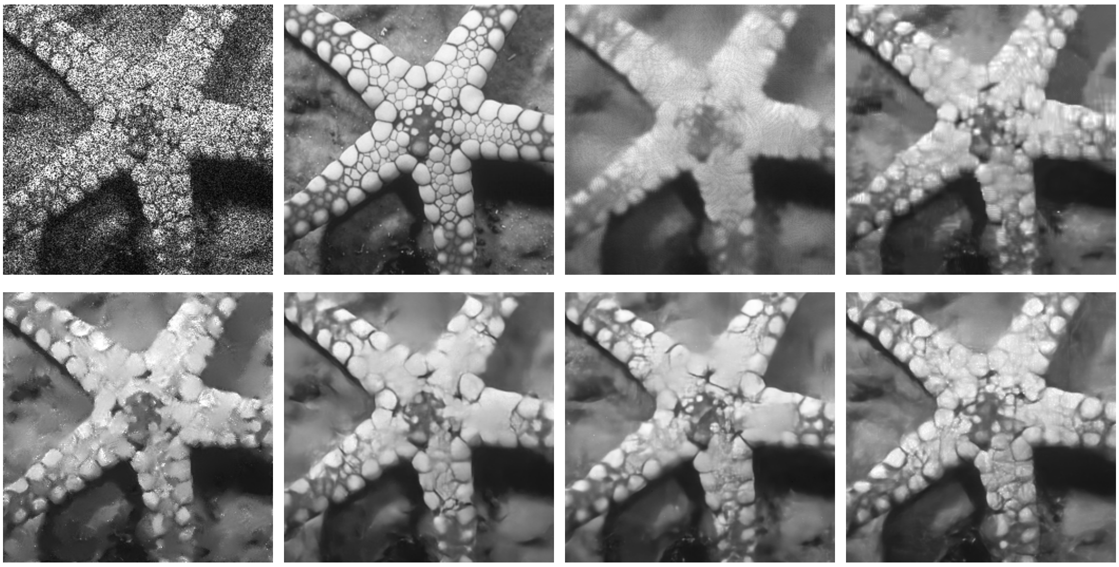

Figure 5.4: Synthetic images: Noisy, Clean, PPB (21.13 dB), SAR-BM3D (22.71 dB), NL-SAR (21.89 dB), CNN-based baseline (23.37 dB), ID-CNN (23.42 dB), synthetic Speckle2Void (23.32 dB).

aiming to get rid of the contribution of specific feature vectors in the computation of the new feature vector $\mathbf{H}_{ij}^{l+1}$ at layer $l+1$. Considering the receptive field extending upwards, all the pixels in a specific row i are associated with a mask M_i which has weight 1 in row i and all the rows above, and 0 everywhere else. This allows to disregard all Euclidian distances with respect to feature vectors that are not contained in the receptive field extending upwards. The construction of the mask M_i is not influenced by the shape of the blind-spot structure. The blind-spot shaping always happens right after the four receptive fields are computed, by shifting each of the four feature maps according to the desired final shape, as in the "local" version.

Table 5.1: Synthetic images - PSNR (dB).

Image	**PPB** [30]	**SAR-BM3D** [127]	**NL-SAR** [31]	**Baseline CNN**	**ID-CNN** [169]	**S2V**	**S2V+TV**	**S2V+NL**
Cameraman	23.02	24.76	24.37	26.26	25.83	25.90	25.90	25.85
House	25.51	27.55	25.75	28.17	28.32	27.96	27.94	28.08
Peppers	23.85	24.92	23.62	26.30	26.26	25.99	26.02	26.09
Starfish	21.13	22.71	21.84	23.39	23.42	23.32	23.31	23.50
Butterfly	22.76	24.48	23.82	25.96	26.09	25.82	25.80	25.98
Airplane	21.22	22.71	21.83	23.78	23.90	23.67	23.65	23.61
Parrot	21.88	24.17	24.13	25.91	25.85	25.44	25.45	25.46
Lena	26.64	27.85	26.80	28.66	28.71	28.54	28.58	28.44
Barbara	24.08	25.37	23.13	24.30	24.38	24.36	24.31	24.74
Boat	24.22	25.43	24.55	26.06	26.00	26.02	25.57	25.88
Average	*23.43*	*24.99*	*23.98*	*25.88*	*25.88*	*25.70*	*25.69*	*25.76*

5.3 Experimental results and discussions

In this section, we evaluate the performance of Speckle2Void, both quantitatively and qualitatively. First, we compare our method with several state-of-the-art methods on a synthetic dataset, where the availability of ground truth images allows to compute objective performance metrics, and then on a real-world SAR dataset, relying on several established no-reference performance metrics and visual results. We also test the proposed method against a benchmarking dataset, composed of a set of simulated canonical images, to highlight its behavior in all the major types of regions found in SAR images. Moreover, we perform an ablation study to show the impact of various design choices on the despeckling performance.

5.3.1 Quality assessment criteria

The evaluation reference metric used to assess quantitative results on synthetic SAR images corrupted by simulated speckle is the PSNR. This allows to understand the denoising capability of our self-supervised method when compared with traditional methods and CNN-based ones with supervised training. In the second set of experiments, conducted on real SAR images, we compare the various despeckling methods by relying on some no-reference performance metrics such as equivalent number of looks (ENL), moments of the ratio image (μ_r, σ_r), quality index $\mathcal{M}$ [32] and the ratio image structuredness RIS [166]. The ENL is estimated over apparently homogeneous areas in the image and is defined as the ratio of the squared average intensity to the variance. Computing the ENL on the noisy SAR image provides an approximate estimate of its nominal number of looks. Moments of the ratio image μ_r and σ_r measure how close the obtained ratio image is to the statistics of pure speckle ($\mu_r = 1$, $\sigma_r = 1$ are desirable for a single-look image). The previous metrics lack in conveying information about the detail preservation capability of a filter and the visual inspection of the ratio image would provide an indication of the remaining structure of what ideally should be pure speckle with no visible pattern. To avoid the subjectiveness of the visual interpretation of ratio images, Gomez et al. [32] designed the quality index $\mathcal{M}$. This index evaluates the goodness of a filter by integrating two measures together: a first-order component measuring the deviation from ideal ENL and from ideal speckle mean over n automatically selected textureless areas and a second-order component measuring the remaining geometrical content within the ratio image through the homogeneity textural descriptor proposed by Haralick et al. [52]. Ideally, $\mathcal{M}$ should tend to zero. RIS [166] is a metric closely related to the second-order component of $\mathcal{M}$, allowing to evaluate solely the remaining geometrical content within the ratio image. Similarly to Gomez et al. [32], it employes the homogeneity textural descriptor proposed by Haralick et al. [52] to measure the similarity among neighbouring pixels. RIS is zero when the ratio image consists of independent identically distributed speckle samples.

5.3.2 Reference methods

The following state-of-the-art references are compared with our method on both optical and SAR datasets:

1. PPB [30];
2. SAR-BM3D [127];
3. NL-SAR [31];
4. CNN baseline with the improved loss defined in [25];
5. ID-CNN [169].

These methods have been chosen for their popularity and diffusion in the SAR community. For PPB [30], SAR-BM3D [127] and NL-SAR [31] methods, we selected parameters as suggested in the original papers. As a CNN baseline we used the well-known network architecture proposed in [194], employing a homomorphic approach and the loss proposed in [25] that better adapts to deal with the speckle noise distribution. ID-CNN has been implemented from scratch following the indications in the original paper for what concerns the CNN architecture and the hyperparameters. Notice that both CNNs follow a supervised training approach with synthetically speckled natural images. We remark that we do not directly compare with the results in SAR-CNN [25] or the more recent work in [27] as they use multitemporal data, which would make the setting unfair with respect to the single observation of a scene in our case. In addition, the dataset used in those works is not publicly available.

As described in Sec. 5.2, Speckle2Void employs four branches where the horizontal and the vertical directions are processed separately with a different set of parameters, as shown in Fig. 5.1. The first part of the architecture consists of 17 blocks composed of 2D convolution with 3×3 kernels with 64 filters each, batch normalization and Leaky ReLU nonlinearity. After that, the branches are merged with a series of three 1×1 convolutions. The non local version of our method maintains the same general structure with an addition of 5 non local layers, one every 3 local layers. The same architecture is used in both the experiments with the only difference that in the case of synthetic images the blind-spot shape is 1×1, since the injected speckle is pixel-wise i.i.d and therefore there is no need to use an enlarged blind-spot. Instead, in the real SAR case the blind-spot shape is variable across training.

For both experiments, the Adam optimization algorithm [84] is employed, with momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We use the Tensorflow framework to train the proposed network on a PC with 64 GB RAM, an AMD Threadripper 1920X, and an Nvidia 1080Ti GPU.

Table 5.2: Performance metrics on 5 real TerraSAR-X test images.

Metric	**Image**	**PPB** [30]	**SAR-BM3D** [127]	**NL-SAR** [31]	**CNN baseline**	**ID-CNN** [169]	**S2V**	**S2V NL**
ENL ↑	1	82	46.2	77.3	52.9	76.5	**88.5**	86.5
	2	78.6	49.1	60.6	48.7	69.9	**89.9**	81.8
	3	76.9	58.1	59.4	52.5	73.1	84.0	**86.0**
	4	54.2	40.4	45.0	37.6	46.2	**54.7**	53.1
	5	**22.9**	16.2	16.8	14.6	16.6	18.9	17.5
μ_r ↑	1	0.887	0.919	0.921	0.963	0.943	0.966	**0.970**
	2	0.925	0.938	0.933	**0.969**	0.964	0.966	0.967
	3	0.926	0.941	0.936	**0.974**	0.969	0.968	0.968
	4	0.933	0.942	0.936	0.974	0.976	0.962	**0.977**
	5	0.853	0.894	0.902	**0.950**	0.918	0.947	0.946
σ_r ↑	1	**0.847**	0.627	0.692	0.726	0.745	0.803	0.800
	2	**0.886**	0.674	0.734	0.740	0.803	0.829	0.817
	3	**0.874**	0.684	0.739	0.756	0.817	0.816	0.814
	4	**0.876**	0.688	0.746	0.755	0.846	0.823	0.837
	5	**0.891**	0.549	0.621	0.683	0.664	0.748	0.736
$\mathcal{M}$ [32] ↓	1	24.4	16.5	13.8	11.9	14.6	7.72	**6.71**
	2	10.1	11.6	15.4	11.6	9.12	9.11	**8.04**
	3	9.82	11.3	13.0	11.3	6.93	6.24	**5.44**
	4	10.6	10.5	16.9	12.3	9.7	8.07	**7.74**
	5	14.4	14.3	11.7	9.76	10.4	8.91	**7.9**
RIS [166] ↓	1	0.402	0.186	0.098	0.145	0.242	0.0929	**0.0817**
	2	0.114	0.0765	0.111	0.0925	0.112	0.0918	**0.075**
	3	0.114	0.0782	0.076	0.113	0.0643	0.0396	**0.0257**
	4	0.0962	**0.0392**	0.129	0.127	0.106	0.0873	0.0804
	5	0.159	0.114	0.0643	0.0566	0.130	0.0708	**0.0547**

5.3.3 Synthetic dataset

In this experiment we use natural images to construct a synthetic SAR-like dataset. Pairs of noisy and clean images are built by generating i.i.d. speckle to simulate a single-look intensity image ($L = 1$).

During training, patches are extracted from 450 different images of the Berkeley Segmentation Dataset (BSD) [116]. The network has been trained for about 400 epochs with a batch size of 16 and learning rate equal to 10^{-5}. All the CNN-based methods have been trained with the same synthetic dataset. Table 5.1 shows performance results on a set of well-known testing images in terms of PSNR. It can be seen that all the CNN-based methods outperform the non local traditional methods by a significant margin. Despite ID-CNN employs the suboptimal ℓ_2 loss, the TV regularizer helps smoothing out the artifacts, showing approximately the same result as the CNN baseline. It can be noticed that our self-supervised method outperforms PPB, SAR-BM3D and NL-SAR. Moreover, it is interesting to notice that while the proposed approach does not use the clean data for training, it achieves comparable results with respect to the supervised ID-CNN and CNN-based baseline methods. This happens for the non local version and TV version as well. We can observe a slight improvement when non-locality is employed. Even if we analyze the performance from a qualitative perspective, as done in Fig. 5.4, we observe the same behaviour. Despite the absence of the true clean images during training, our method produces images as visually pleasing as those produced by the CNN-based reference approaches with comparable edge-preservation capabilities. This is a significant result because it shows that, in theory, we do not need supervised training to achieve the outstanding despeckling results obtained by CNN-based methods.

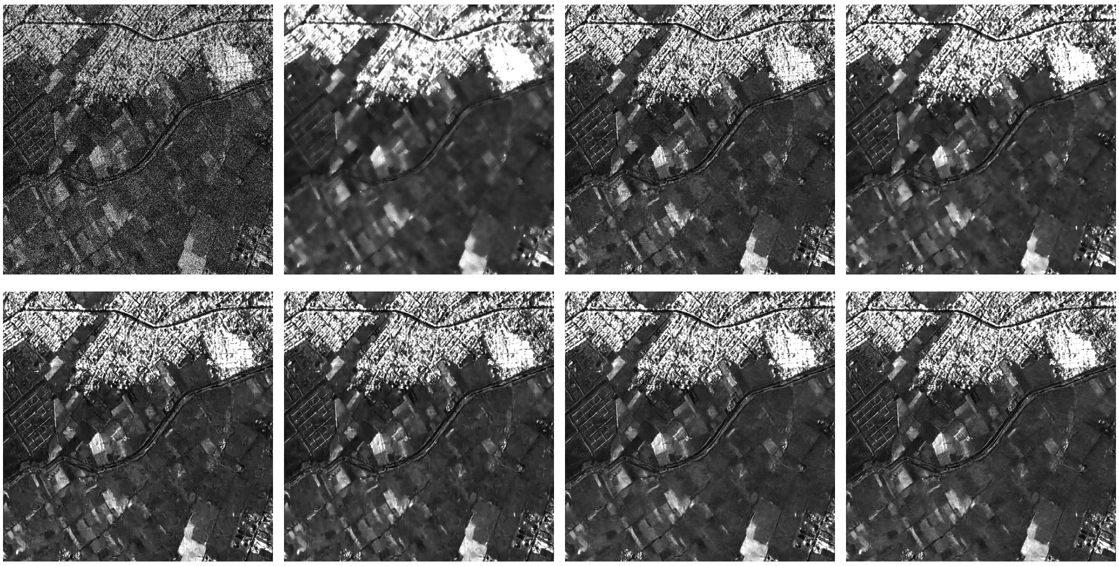

Figure 5.5: TerraSAR-X image 1. Top-Left to bottom-right: Noisy, PPB, SAR-BM3D, NL-SAR, CNN-based baseline, ID-CNN, Speckle2Void, Speckle2Void+NL.

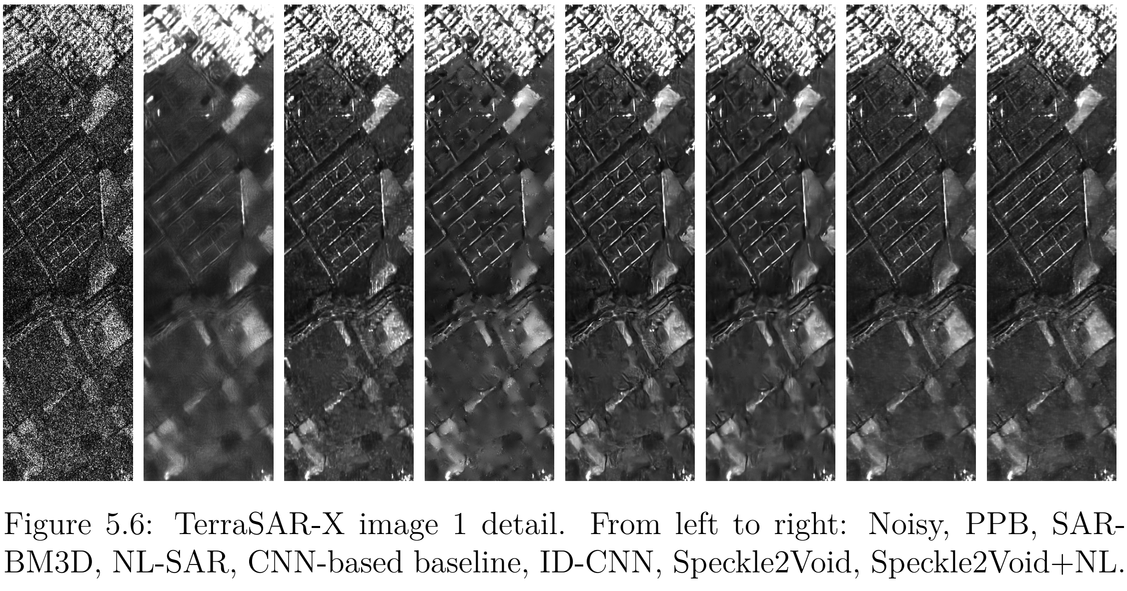

Figure 5.6: TerraSAR-X image 1 detail. From left to right: Noisy, PPB, SAR-BM3D, NL-SAR, CNN-based baseline, ID-CNN, Speckle2Void, Speckle2Void+NL.

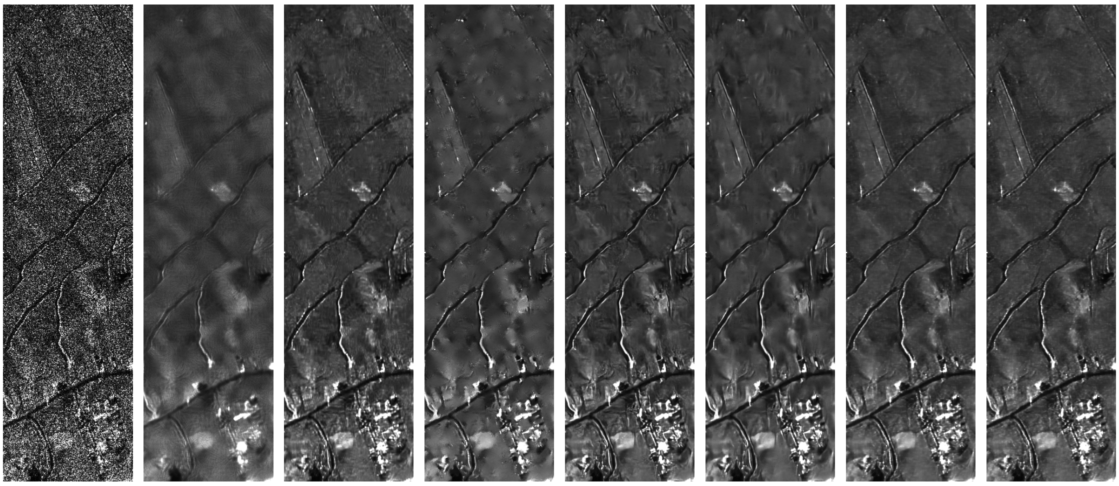

Figure 5.7: TerraSAR-X image 2 detail. From left to right: Noisy, PPB, SAR-BM3D, NL-SAR, CNN-based baseline, ID-CNN, Speckle2Void, Speckle2Void+NL.

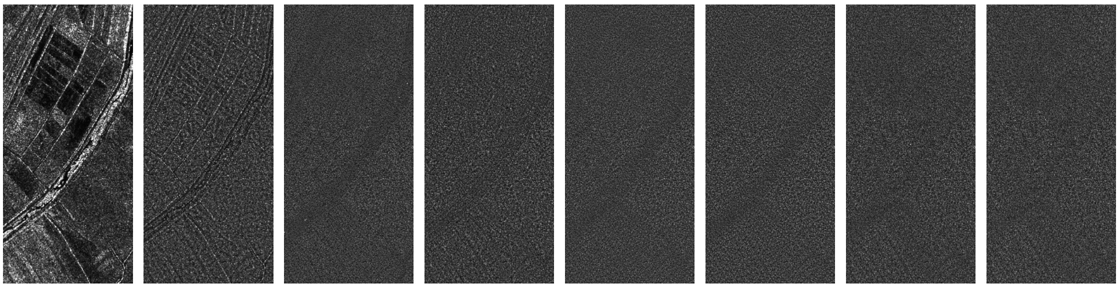

Figure 5.8: TerraSAR-X image 4 detail. From left to right: Noisy and ratio images (PPB, SAR-BM3D, NL-SAR, CNN-based baseline, ID-CNN, Speckle2Void, Speckle2Void+NL).

5.3.4 TerraSAR-X dataset

In this experiment we employ single-look TerraSAR-X images[1]. Notice that optimal results are obtained by training a model that is specific to a given SAR platform (e.g., TerraSAR-X in our example). We suggest retraining from random initialization to optimize the model for a different platform. This should not be an issue since we only require a modest number of noisy images and we also do not need careful curation of multitemporal data.

As mentioned in Sec. 5.2.2, both training and testing images are pre-processed through the blind speckle decorrelator in [90] to whiten them. To ensure fairness, the whitening procedure is applied to the images for all the tested methods.

[1] https://tpm-ds.eo.esa.int/oads/access/collection/TerraSAR-X/tree

During training, 64×64 patches are extracted from 30000 whitened SAR images of size 256×256. The network has been trained for 300000 iterations with a batch size of 16 and an initial learning rate of 10^{-4} multiplied by 0.1 at 150000 iterations. In this case, in addition to two versions (L/NL) of the proposed method used for the synthetic images, we add the TV regularizer to the loss with a λ_{TV} of 5×10^{-5} and we apply the regularized training procedure described in Sec. 5.2.2, carefully choosing the blind-spot shape. By empirical observation we found non-negligible residual noise correlation in the vertical direction after the whitening stage, so we adapted the structure of the blind spot accordingly. The regularized training alternates between a 3×1 and 1×1 shape with probabilities 0.9 and 0.1, respectively. This allows us to take into account the wider vertical autocorrelation of the speckle. In the ablation study presented in Sec. 5.3.6 we also show the results obtained when only a 1×1 blind spot is used.

Table 5.2 and Figs. 5.5,5.6,5.7 show the results obtained on a set of 1000×1000 test images[2], that were not included in the training set. Speckle2Void outperforms all other methods for almost all testing images in terms of ENL, showing a better speckle suppression capability on smooth areas. The non local version of Speckle2Void scores a slightly lower ENL with respect to the local version as it recovers finer details, generating an additional texture over the apparently homogeneous areas as shown in Fig. 5.6. The metric μ_r is very close to the desired statistic of the ratio image for all the considered methods, in particular for the CNN-based ones. The reference method PPB [30] provides the best result in terms of σ_r showing a strong speckle suppression, but a very poor detail preservation capability as confirmed by the qualitative comparison in Figs. 5.6 and 5.7. Despite SAR-BM3D [127] provides worse results in terms of σ_r with respect to PPB[30], it produces images with higher fidelity and finer details, as can be observed both visually in Fig. 5.5 and quantitatively with the RIS [166]. However, several areas in the SAR-BM3D image still present artifacts like streaks or unrealistic texture. NL-SAR [31] shows a stronger speckle suppression than SAR-BM3D [127], providing better results in terms of ENL and σ_r.

Overall, the CNN-based methods show a greater speckle suppression than SAR-BM3D [127] and PPB [30]. However, both the CNN baseline and ID-CNN [169] tend to oversmooth and produce cartoon-like edges. The test image in Fig. 5.5 presents strong artifacts, making the recovered details look quite unrealistic. This is due to the domain gap between natural images and real SAR images and it represents a strong argument against supervised training with synthetically speckled images. On the contrary, Speckle2Void does not hallucinate artifacts over homogeneous regions and produces higher quality images with respect to any other reference method, with much more realistic details in regions with man-made structures and sharp

[2] High-resolution visualization: `https://diegovalsesia.github.io/speckle2void`

edges. This is confirmed qualitatively by a visual inspection of the cleaned image in Fig. 5.5, 5.6, 5.7. Instead, Fig. 5.8 shows the image obtained as the ratio between the noisy and despeckled images. Ideally, no structure should be evident in the ratio image. Also in this case, we can observe the capability of Speckle2Void to remove the speckle effectively, with a minimal amount of visible patterns. The outstanding visual quality of Speckle2Void demonstrates the effectiveness of both direct training on real SAR images and of the adopted regularized training procedure to tackle the residual local noise correlation structure.

Moreover, if we compare the two versions of the proposed method, we can notice that adding the non local layers provides a marginal improvement in the preservation of the details, yielding lower values for $\mathcal{M}$ [32] and RIS [166]. The drawback of the non local version of Speckle2Void is its higher computational overhead, leading to a much longer training and inference time.

5.3.5 Benchmarking dataset

The presented quantitative assessment relies on no-reference metrics as the lack of clean images prevents from using full-reference measures. In [33] the authors introduce a standard benchmark for the objective assessment of SAR despeckling techniques. The use of this framework enriches our quantitative assessment on no-reference metrics by evaluating the behaviour of the compared methods on a set of canonical scenes, generated through physical SAR simulation. Five different scenes have been simulated to assess specific features of the despeckling methods:

- homogeneous scene (water, bare soils, and vegetated areas) to focus on speckle suppression ability;
- texture scene to specifically evaluate the scene feature preservation on a nonflat terrain;
- scene with edges (roads, rivers, and region boundaries) to evaluate the preservation of contours;
- scene with isolated point target to assess the amount of radiometric distortion;
- scene with urban areas to assess the preservation of man-made structures.

In [33] the authors also propose to use a set of reference and no-reference measures associated to each test image. Table 5.3 shows that the proposed methods achieve comparable results for most of the test images and in some cases outperform the other methods. We remark that Speckle2Void is optimized on the real TerraSAR-X dataset, which present different statistics with respect to the simulated SAR images considered in the benchmark, such as a different residual noise correlation.This leads us to believe that the despeckling action of the proposed method is actually slightly sub-optimal when evaluated on the simulated SAR test images rather than on TerraSAR-X images.

Table 5.3: Measures for simulated SAR test images.

Image	**Metric**	**PPB** [30]	**SAR-BM3D** [127]	**NL-SAR** [31]	**CNN baseline**	**ID-CNN** [169]	**S2V**	**S2V NL**
Homogeneous	MoI ↑	0.997	0.978	1.000	0.991	0.978	0.987	0.988
	MoR ↑	0.960	0.979	0.972	0.979	0.995	1.01	0.989
	VoR ↑	0.820	0.814	0.837	0.844	0.903	0.898	0.88
	ENL ↑	127.68	102.44	104.52	125.69	122.94	120.48	112.96
	DG ↑	20.29	19.40	19.46	20.2	20.04	20.03	19.8
Texture	MoI ↑	0.998	0.968	0.915	0.931	0.836	0.867	0.846
	MoR ↑	0.911	0.833	0.857	0.807	0.893	0.847	0.808
	VoR ↑	0.560	0.415	0.485	0.475	0.766	0.848	0.822
	C_x (2.40)	2.71	2.43	2.31	2.25	2.29	2.24	2.21
	DG ↑	3.68	5.32	4.83	4.25	3.77	3.5	3.45
Squares	ES (up) ↓	0.07	0.036	0.07	0.026	0.033	0.057	0.058
	ES (down) ↓	0.209	0.113	0.198	0.0825	0.0873	0.138	0.158
	FOM ↑	0.837	0.847	0.799	0.818	0.82	0.825	0.834
Corner	C_{NN} ↑	3.75	7.39	5.67	7.8	7.77	7.79	7.79
	C_{BG} ↑	32.69	35.45	33.75	36.53	36.51	36.55	36.54
Building	C_{DR} ↑	64.90	65.91	64.44	65.92	65.98	65.91	65.9
	BS ↓	3.13	1.46	6.827	0.3082	0.2612	0.272	0.4031

Homogeneous case

This test case represents a flat surface. The performance is evaluated using the following metrics: the mean value of the filtered image (MoI), that should be preserved after despeckling; the mean and the variance of the ratio image (MoR and VoR) that should match the pure speckle statistics; the ENL and the despeckling gain (DG), which measure the speckle reduction factor on a logarithmic scale by exploiting the available clean reference. All the compared methods do not introduce any notable distortion on the mean. However, the two version of Speckle2Void present the mean indicators that are overall the closest to 1. In addition, the VoR indicates that the proposed methods are the ones that more strongly suppress speckle. The DG metric shows comparable performance for all the compared methods. The latter measure is slightly biased by the fact that the reference image is not really clean.

Texture case (Digital Elevation Model)

The texture image represents an artificial canonical fractal DEM. The performance is evaluated measuring MoI, MoR, VoR, DG and the coefficient of variation $C_{\hat{x}}$, i.e., the ratio between the estimated standard deviation and the mean of the image. The latter metric measures the texture preservation. The two means show slightly worse performance for the proposed methods with respect to the references, denoting a slight radiometric distortion. All the reference techniques present a small value of VoR, showing the challenge of speckle removal in case of a highly textured image. The VoR values of the two proposed methods are the closest to 1.

The coefficient of variation C_x should match the theoretical one computed on the clean image, which corresponds to 2.40. The two versions of Speckle2Void present a comparable $C_{\hat{x}}$ with respect to the other CNN-based methods. DG shows similar results for all the compared methods, showing a good speckle suppression even for this challenging image.

Edges (Squares)

This test case represents a flat surface divided in 4 regions with different intensity levels, creating straight contours aligned to the range and azimuth coordinates as shown in Fig. 5.9. The performance is evaluated through the measure of edge smearing (ES), which gives an indication of the edge degradation and the smoothing action applied by the despeckling methods, and an indirect measure called Pratt's FOM, which quantifies the ability of an automatic edge detection algorithm to recognize the edges in the clean estimate. Table 5.3 reports the ES measures for the two vertical edges, characterized by lower (up) and higher (down) contrast, along with the FOM for the detected edges. Lower ES values indicate less smearing. The worst results comes from the methods producing the blurriest edges such as PPB [30] and NL-SAR [31]. However, this metric does not give a complete insight about the edge preservation and it is quite unreliable. FOM represents the best measure to evaluate edge preservation by quantifying their recognition through a detector algorithm. The FOM values in Table 5.3 should be higher than the FOM resulting from the noisy image (0.792) and as close as possible to the one resulting from the clean reference image (0.993). The two proposed methods present FOM values that are higher than the ones produced by the supervised CNN-based methods and consistent with the best results, provided by PPB [30] and SAR-BM3D [127].

Isolated point target case (Corner)

The corner image represents a point target produced by a corner reflector at the center of a flat scene. The performance is evaluated through two intensity contrast measures in logarithmic scale, quantifying the preservation of the point target with respect to the average intensity in the surrounding region (C_{NN}) and the average intensity of the whole background (C_{BG}). All the CNN-based methods in Table 5.3 perform prior classification as they have been trained without the point targets. In testing, a thresholding procedure is performed to remove the point targets prior to filtering and to copy them back right after. Overall, CNN-based techniques tend to present the highest values for these two metrics.

Urban area case (Building)

The building image represents an isolated building over a homogeneous flat surface. The intense double reflection line resulting from the multiple scattering mechanisms should be preserved by the despeckling technique. The performance is evaluated employing a building smearing measure BS and an intensity contrast measure C_{DR} in logarithmic scale. C_{DR} quantifies the preservation of the double reflection segment with respect to the average intensity of the background. This is another case where the CNN-based methods better preserve the radiometric features of the building, presenting a BS closer to zero and a higher C_{DR}.

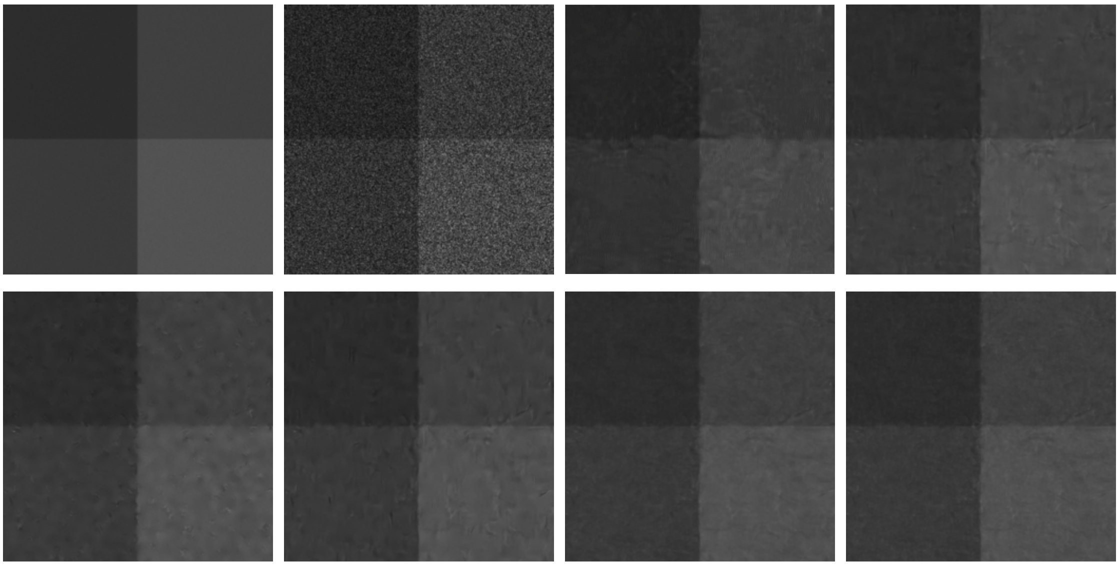

Figure 5.9: *Squares* benchmark image. Top-Left to bottom-right: Clean, Noisy, SAR-BM3D, NL-SAR, CNN-based baseline, ID-CNN, Speckle2Void, Speckle2Void+NL.

5.3.6 Ablation study

In the following study, we want to assess the benefits of some of the features proposed for Speckle2Void.

Original vs whitened

First, we show the importance of the pixel-wise noise independence condition when training a blind-spot network. To assess it, we train Spleckle2Void with two different datasets. One dataset is composed of real single-look complex images as they are provided by the focusing algorithm for the TerraSAR-X satellite, while the other dataset is composed of the same real SAR images but pre-processed by the decorrelator defined in [90]. For both datasets we use a 1×1 blind-spot shape, including solely the center pixel during the entire training. To better highlight the effect of the whitening procedure, we do not add the TV regularization in

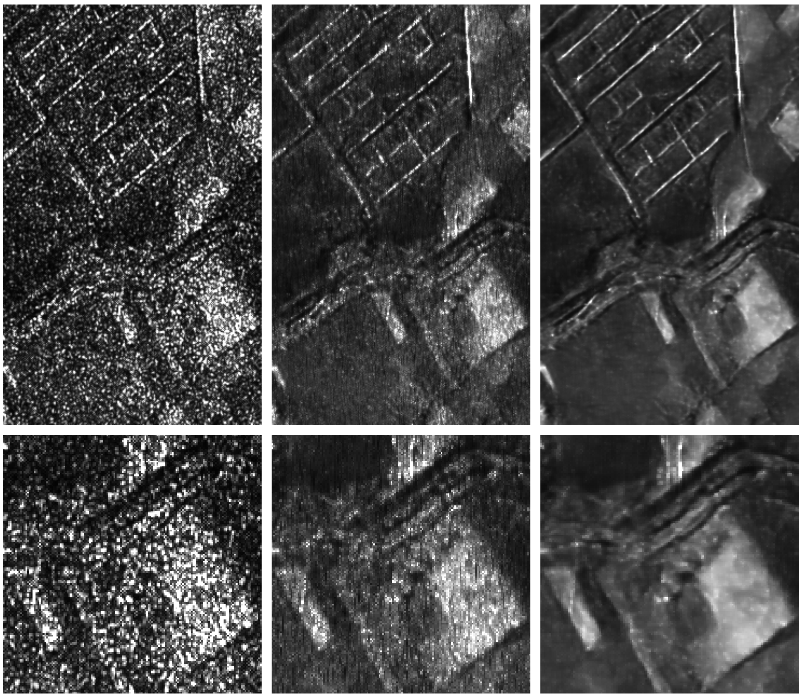

Figure 5.10: From left to right: cleaned image resulting from the training with the original TerraSAR-X dataset (ENL 1.28), cleaned image resulting from the training with the whitened TerraSAR-X dataset (ENL 14.5) and Speckle2Void (ENL 88.5).

the loss. Fig. 5.10 shows the two resulting cleaned images together with the one obtained by the full Speckle2Void method (whitening+variable blind spot). The visual difference between the left image and the middle one shows that the decorrelator improves drastically the qualitative performance, since barely any denoising is performed in the first image.

Enlarging the blind-spot

In our regularized training procedure we vary the shape of the blind-spot to account for the residual noise correlation that persists even after the whitening procedure. To better understand the effect of enlarging the size of the blind-spot structure, we compare Speckle2Void trained with the canonical 1×1 blind-spot shape against a 3×3 shape. Notice that, in this experiment, the latter uses the 3×3 blind-spot in testing as well, differently from the regularization procedure explained in 5.2.2 which always uses a 1×1 blind spot in testing. Moreover, to better highlight the effect of the shape of the blind-spot, we do not add the TV regularization in the loss. Fig. 5.11 shows a visual comparison between the two methods. The left image is the result produced by the network with blind-spot of shape 1×1. We can notice sharper edges and more details with respect to the

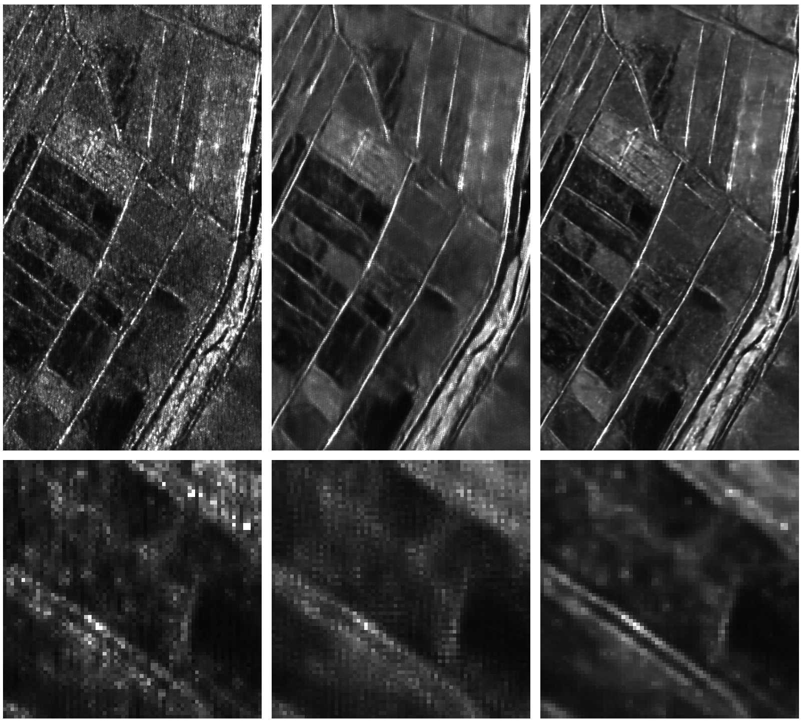

Figure 5.11: From left to right: network with 1 × 1 blind-spot, network with 3 × 3 blind-spot, Speckle2Void.

middle image produced by the network with blind-spot of shape 3 × 3, which looks more blurry. However, we also see more residual noise in the image on the left. Enlarging the shape of blind-spot structure leads to a more effective speckle noise reduction as the network uses surrounding pixels that are less correlated with center pixel. A downside of expanding the blind-spot is to reduce the amount of relevant information for the network to estimate the center pixel, resulting in a smoother image with a loss of high frequency details, failing to preserve the original edges. In the image on the right we report the result of Speckle2Void, showing that the proposed method is able to achieve stronger speckle suppression with an impressive preservation of details.

Table 5.4 provides a quantitative comparison using the benchmark dataset proposed in [33]. For the homogeneous case, Speckle2Void provides a stronger speckle suppression than the network with a blind-spot of shape 1 × 1 or with shape 3 × 3. The latter method presents a despeckling gain (DG) very close to the one of Speckle2Void and much higher than the one produced by the network with blind-spot of shape 1 × 1. This suggests the ability of the 3 × 3 blind-spot to disregard the strong noise correlation of the immediate neighboring pixels with respect to the center pixel, when producing the clean estimate. For the same reason, the network with blind-spot of shape 3 × 3 provides the best despeckling suppression ability

Table 5.4: Blind-spot size. Measures for simulated SAR test images.

Image	Metric	1x1	3x3	Speckle2Void
Homogeneous	MoI ↑	0.977	1.000	0.988
	MoR ↑	1.000	0.976	0.989
	VoR ↑	0.874	0.861	0.88
	ENL ↑	20.05	103.09	112.96
	DG ↑	13.00	19.43	19.8
Texture	MoI ↑	1.020	0.987	0.846
	MoR ↑	0.834	0.838	0.808
	VoR ↑	0.963	0.719	0.822
	C_x (2.40)	2.45	2.43	2.21
	DG ↑	3.34	4.03	3.45
Squares	ES (up) ↓	0.064	0.074	0.058
	ES (down) ↓	0.145	0.171	0.158
	FOM ↑	0.783	0.795	0.834
Corner	C_{NN} ↑	7.77	7.77	7.79
	C_{BG} ↑	36.61	35.51	36.54
Building	C_{DR} ↑	65.92	65.86	65.9
	BS ↓	0.4394	0.4159	0.4031

in the DEM test case. The FOM metric for the squares case shows that a bigger blind-spot allows a better edge detection even in the presence of blurrier contours. Speckle2Void adds to the filtered image the necessary high frequency details to help the downstream detector algorithm. For the corner and building cases, the results of the three methods are comparable, since the radiometric preservation of the point targets strongly depends on the prior classification step that is the same in all the three methods.

Effect of the TV regularizer

Speckle2Void employs TV in the loss as an additional spatial regularizer. We aim to understand its impact by comparing Speckle2Void with a version trained without TV. Fig. 5.12 shows the resulting cleaned images, revealing the reduced amount of artifacts and smoother flat areas when the TV regularization is employed.

Prior vs posterior

The Bayesian framework, exploited in our method, makes use of the noisy SAR image to obtain the despeckled version by computing the expected value of the posterior distribution. The blind-spot CNN produces the parameters of the prior distribution. If we compute its expected value we obtain the prior despeckled image.

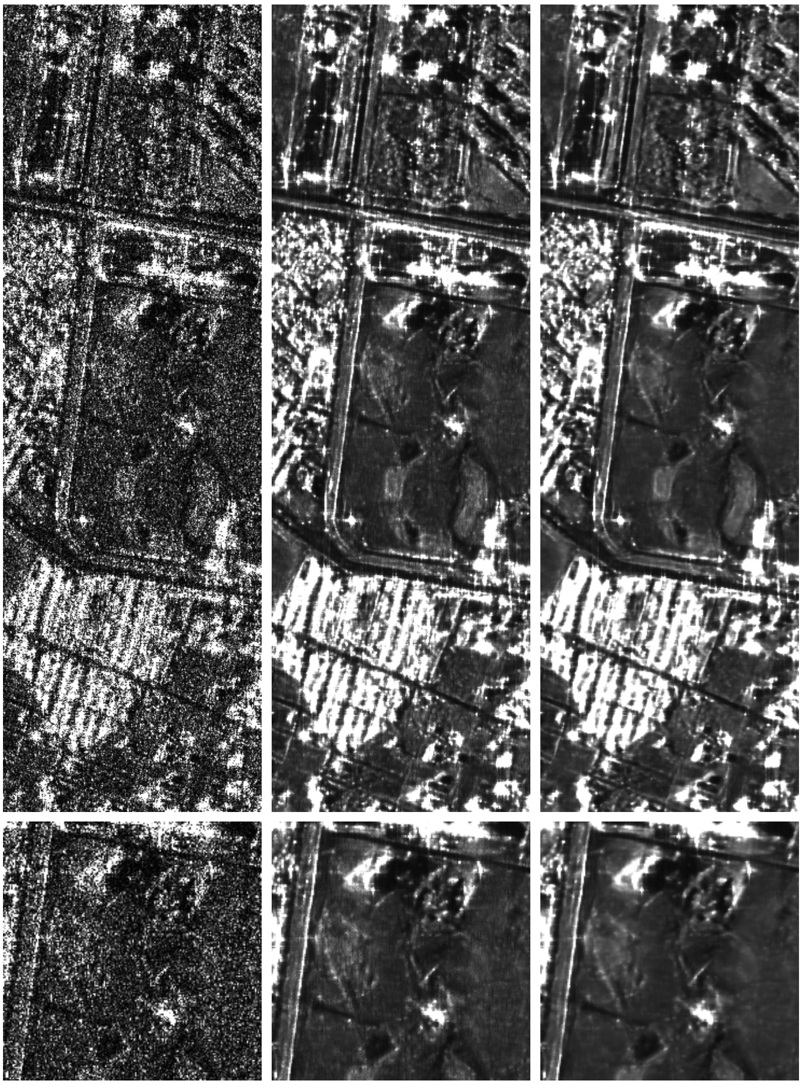

Figure 5.12: From left to right: Noisy, Speckle2Void w/o TV and Speckle2Void.

In Fig. 5.13, the prior and the posterior images highlight the great qualitative improvement brought by the use of the noisy observations in the estimation of the cleaned image with the posterior mean. The prior image shows fuzzy edges and a disturbing granular pattern that makes the posterior image visually preferable.

Table 5.5: Training time and Runtime comparisons.

Image	**PPB** [30]	**SAR-BM3D** [127]	**NL-SAR** [31]	**Baseline CNN**	**ID-CNN** [169]	**S2V**	**S2V+NL**
Training	-	-	0.8645 s (100x100)	3 days 2 h	7 h	1 day 3 h	6 days 19h
Inference (1000x1000)	27.54 s	223.51 s	23.39 s	0.587 s	0.1627 s	1.26 s	432.41 s

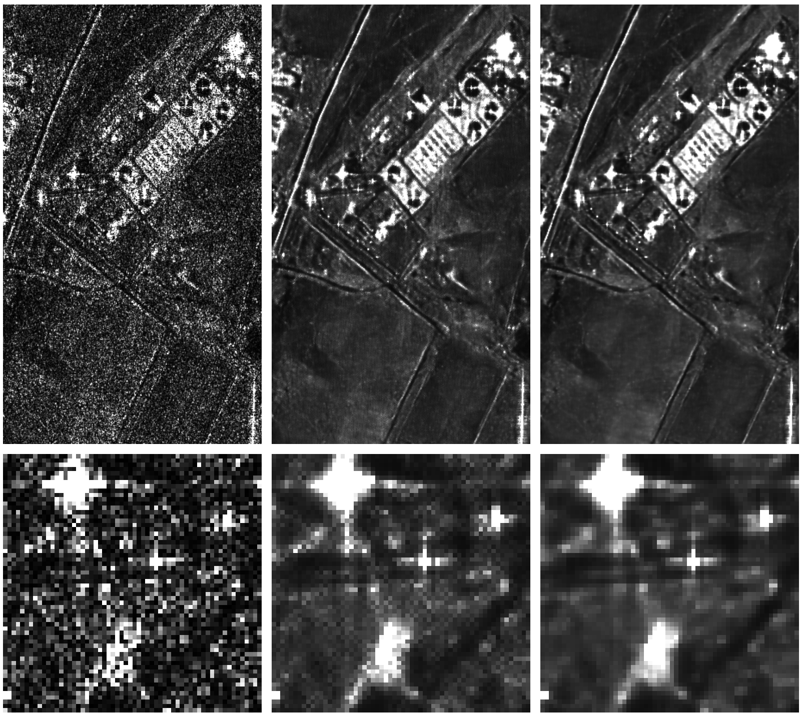

Figure 5.13: From left to right: Noisy, Speckle2Void (Prior mean image), Speckle2Void (Posterior mean image).

5.3.7 Transferability to Sentinel-1

In this section we present a result to show the performance of the Speckle2Void model trained on TerraSAR-X data when applied to Sentinel-1 single look images. Fig. 5.14 shows a qualitative result while the caption reports quantitative metrics. It is interesting to notice that Speckle2Void provides excellent performance, both qualitatively by showing strong speckle suppression while maintaining several details of the scene, and quantitatively according to the metrics presented in the previous sections. A more detailed study on how to train optimally on Sentinel-1, either by finetuning a pretrained model or from scratch, is out of the scope of this thesis, but it would be an interesting future developement, especially in the context of studying how well self-supervised representations transfer across platforms.

5.3.8 Training time and runtime comparisons

The training and inference run-times for all the methods considered in the experimental evaluation are shown in Table 5.5. The experiments have been performed on a PC with 64-GB RAM, an AMD Threadripper 1920X CPU, and an Nvidia 1080Ti GPU. All the CNN-based methods have been trained using the Tensorflow framework. The CNN-based methods have the lowest inference times except for

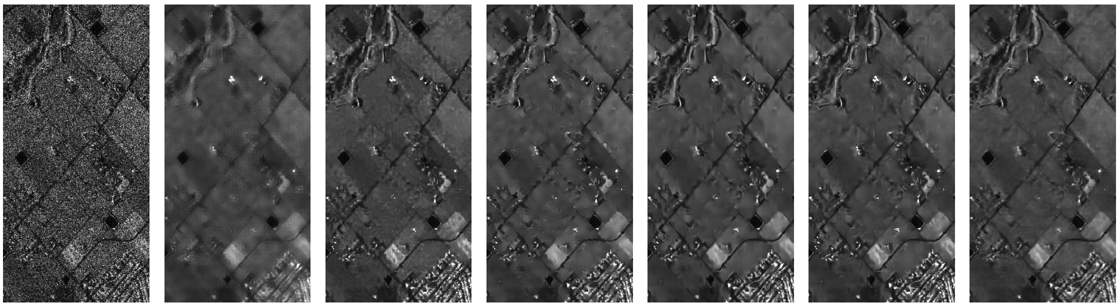

Figure 5.14: Sentinel-1 image detail. From left to right: Noisy, PPB (ENL = 141, $\mu_r = 0.926$, $\sigma_r = 0.89$, $\mathcal{M} = 9.17$, RIS = 0.1032), SARBM3D (ENL = 245, $\mu_r = 0.954$, $\sigma_r = 0.787$, $\mathcal{M} = 4.8$, RIS = 0.0227), NL-SAR (ENL = 150, $\mu_r = 0.944$, $\sigma_r = 0.778$, $\mathcal{M} = 9.7$, RIS = 0.0080), CNN baseline (ENL = **384**, $\mu_r = 0.979$, $\sigma_r = 0.900$, $\mathcal{M} = 3.27$, RIS = 0.0128), ID-CNN (ENL = 259, $\mu_r = 0.968$, $\sigma_r = 0.867$, $\mathcal{M} = 3.22$, RIS = 0.0102), Speckle2Void (ENL = 299, $\mu_r = \mathbf{0.981}$, $\sigma_r = \mathbf{0.939}$, $\mathcal{M} = \mathbf{2.70}$, RIS = **0.0016**)

the nonlocal version of Speckle2Void. This version is more expensive due to the non local layers, which have to compute dynamic aggregation weights for all the pixels in a search window. Moreover, due to GPU memory constraints, the nonlocal version of Speckle2Void processes SAR images in multiple smaller patches, resulting in a longer inference time to reconstruct the entire clean image. The local version of Speckle2Void takes, on average, 1.26 seconds to process a 1000 × 1000 image, which is slightly higher than the inference times of the baseline CNN and ID-CNN models because it has to process the same image four times to compute the four half-plane receptive fields. However, it is significantly lower than the inference times of model-based methods. The training times affect only the CNN-based methods and span from some hours to several days.

Chapter 6

Conclusions

In this thesis we explored the application of deep learning paradigm to two imaging inverse problems with a particular attention to data availability. More specifically, we avoided the use of synthetic datasets when training the networks.

Firstly, we have introduced DeepSUM, one of the first CNN architectures to deal with super-resolution from multitemporal remote sensing images. We showed that the proposed deep learning framework can successfully deal with complex degradation and temporal variation models and provide state-of-the-art performance, resulting as the best method in the PROBA-V SR challenge. We also demonstrated that non local information can be successfully exploited by neural networks to enhance the reconstruction quality of multitemporal remote sensing images in a MISR problem.

The second inverse imaging problem we addressed is despeckling. In Chapter 5 we have presented Speckle2Void, a self-supervised Bayesian denoising framework for despeckling. The main obstacle in applying classical supervised deep learning methods to despeckling is represented by the vast content disparity between speckle injected natural images and real SAR images, often resulting in unfaithful cleaned images. Speckle2Void exploits a customized version of the blind-spot convolutional networks where the receptive field is constrained to exclude a variable amount of pixels throughout training to account for the correlation structure of the noise, introducing one of the first deep learning despeckling method purely based on real single-look complex SAR images. Speckle2Void is able to learn to produce excellent images with faithful details and no visible residual speckle noise.

6.1 Open problems

In this thesis, we have described a new blind super-resolution method trained on a dataset with real LR and HR images, a non local extension and a novel self-supervised despeckling method requiring no ground truth. However, there are some research directions worth to be investigated that may lead to interesting research problems. In the following we state some future works which may be interesting to develop to improve the current methods.

- *DeepSUM architecture*: in Chapter 3 we presented a novel method for super-resolution coupling the reconstruction and the registration problems solved with an end-to-end trainable CNN. Chapter 4 proposes an improvement in the feature extraction network. The assumption on the registration in both works is that the misalignment among the LR images is of translational nature. The RegNet architecture is specifically devised to handle this type of geometric disparity. A possible advancement could be designing a different architecture to account for non-rigid transformations in order to generalize to multiple datasets where more complex misalignments are present among LR images.

- *Number of LR images*: DeepSUM architecture was designed to take a fixed number of LR images. In order to handle a variable number of LR images, the method could be extended to be independent of the number of input images.

- *Speckle noise decorrelation procedure*: in Chapter 5 the presented method copes with the spatially correlated speckle using a decorrelation procedure [90] as a preprocessing and a regularized training procedure. In the ideal setting the correlated speckle noise would be handled by the network together with the actual denoising task. A method based on the current definition of blind-spot networks, needs to be trained on SAR images with decorrelated noise as it is performed in a self-supervised fashion. During training, the network is pushed to use noise correlations between the neighboring pixels and the center pixel to generate the noisy SAR image in output. Moreover enlarging the blind-spot is a sub-optimal solution as it reduces the amount of information the network can exploit to clean the center pixel. Plugging a decorrelation mechanism into the network to naturally taking care of the pixel-wise noise correlation would require an extra term in the loss working in a contrastive way with respect to the reconstruction term or or directly redefining the reconstruction term. When working on our method (Speckle2Void), we tried to add a regularization term to the loss to impose a flat spectrum of the noise ratio image, achieving unsatisfactory results. Probably the regularization term should have a direct effect on the way the network uses the noise samples from neighboring pixels.

Bibliography

[1] A. Achim, E. E. Kuruoglu, and J. Zerubia. "SAR image filtering based on the heavy-tailed Rayleigh model". In: *IEEE Transactions on Image Processing* 15.9 (2006), pp. 2686–2693.

[2] A. Achim, P. Tsakalides, and A. Bezerianos. "SAR image denoising via Bayesian wavelet shrinkage based on heavy-tailed modeling". In: *IEEE Transactions on Geoscience and Remote Sensing* 41.8 (2003), pp. 1773–1784.

[3] Forest Agostinelli, Michael R Anderson, and Honglak Lee. "Adaptive Multi-Column Deep Neural Networks with Application to Robust Image Denoising". In: *Advances in Neural Information Processing Systems*. Ed. by C. J. C. Burges et al. Vol. 26. Curran Associates, Inc., 2013, pp. 1493–1501.

[4] F. Argenti and L. Alparone. "Speckle removal from SAR images in the undecimated wavelet domain". In: *IEEE Transactions on Geoscience and Remote Sensing* 40.11 (Nov. 2002), pp. 2363–2374. DOI: `10.1109/TGRS.2002.805083`.

[5] F. Argenti, T. Bianchi, and L. Alparone. "Multiresolution MAP Despeckling of SAR Images Based on Locally Adaptive Generalized Gaussian pdf Modeling". In: *IEEE Transactions on Image Processing* 15.11 (2006), pp. 3385–3399.

[6] F. Argenti et al. "A Tutorial on Speckle Reduction in Synthetic Aperture Radar Images". In: *IEEE Geoscience and Remote Sensing Magazine* 1.3 (2013), pp. 6–35.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein Generative Adversarial Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, June 2017, pp. 214–223. URL: `http://proceedings.mlr.press/v70/arjovsky17a.html`.

[8] Gilles. Aubert and Jean-François. Aujol. "A Variational Approach to Removing Multiplicative Noise". In: *SIAM Journal on Applied Mathematics* 68.4 (2008), pp. 925–946. DOI: `10.1137/060671814`. eprint: `https://doi.org/10.1137/060671814`. URL: `https://doi.org/10.1137/060671814`.

[9] S. D. Babacan, R. Molina, and A. K. Katsaggelos. "Total variation super resolution using a variational approach". In: *IEEE International Conference on Image Processing (ICIP)*. Oct. 2008, pp. 641–644.

[10] Joshua Batson and Loic Royer. "Noise2Self: Blind denoising by self-supervision". In: 2019.

[11] Marco Bevilacqua et al. "Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding". In: *British Machine Vision Conference (BMVC)*. 2012.

[12] M. I. H. Bhuiyan, M. O. Ahmad, and M. N. S. Swamy. "Spatially Adaptive Wavelet-Based Method Using the Cauchy Prior for Denoising the SAR Images". In: *IEEE Transactions on Circuits and Systems for Video Technology* 17.4 (2007), pp. 500–507.

[13] J. M. Bioucas-Dias and M. A. T. Figueiredo. "Multiplicative Noise Removal Using Variable Splitting and Constrained Optimization". In: *IEEE Transactions on Image Processing* 19.7 (July 2010), pp. 1720–1730. DOI: `10.1109/TIP.2010.2045029`.

[14] Ashish Bora et al. "Compressed Sensing Using Generative Models". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, pp. 537–546.

[15] Andrea Bordone Molini et al. "DeepSUM++: Non-local Deep Neural Network for Super-Resolution of Unregistered Multitemporal Images". In: *2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2020.

[16] Andrea Bordone Molini et al. "Towards Deep Unsupervised SAR Despeckling with Blind-Spot Convolutional Neural Networks". In: *2020 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2020.

[17] Bert De Brabandere et al. "Dynamic Filter Networks". In: *Proceedings, International Conference on Neural Information Processing Systems (NIPS)*. 2016.

[18] A. Buades, B. Coll, and J. -. Morel. "A non-local algorithm for image denoising". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. June 2005, 60–65 vol. 2. DOI: `10.1109/CVPR.2005.38`.

[19] H. C. Burger, C. J. Schuler, and S. Harmeling. "Image denoising: Can plain neural networks compete with BM3D?" In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 2392–2399. DOI: 10.1109/CVPR.2012.6247952.

[20] Jose Caballero et al. "Real-Time Video Super-Resolution with Spatio-Temporal Networks and Motion Compensation". In: *CoRR* abs/1611.05250 (2016). arXiv: 1611.05250.

[21] T. F. Chan and Chiu-Kwong Wong. "Total variation blind deconvolution". In: *IEEE Transactions on Image Processing (TIP)* 7.3 (Mar. 1998), pp. 370–375.

[22] Tak-Ming Chan et al. "Neighbor embedding based super-resolution algorithm through edge detection and feature selection". In: *Pattern Recognition Letters* 30.5 (2009), pp. 494–502.

[23] Y. Chang et al. "HSI-DeNet: Hyperspectral Image Restoration via Convolutional Neural Network". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.2 (2019), pp. 667–682. DOI: 10.1109/TGRS.2018.2859203.

[24] J. Chen, J. Nunez-Yanez, and A. Achim. "Video Super-Resolution Using Generalized Gaussian Markov Random Fields". In: *IEEE Signal Processing Letters (SPL* 19.2 (Feb. 2012), pp. 63–66.

[25] G. Chierchia et al. "SAR image despeckling through convolutional neural networks". In: *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 2017, pp. 5438–5441.

[26] Bartomeu Coll and Jean-Michel Morel. "A Review of Image Denoising Algorithms, with a New One". In: *SIAM Journal on Multiscale Modeling and Simulation* 4 (Jan. 2005). DOI: 10.1137/040616024.

[27] Davide Cozzolino et al. "Nonlocal CNN SAR Image Despeckling". In: *Remote Sensing* 12 (Mar. 2020), p. 1006. DOI: 10.3390/rs12061006.

[28] Zhen Cui et al. "Deep Network Cascade for Image Super-resolution". In: *European Conference on Computer Vision (ECCV)*. 2014.

[29] K. Dabov et al. "Image denoising by sparse 3-D transform-domain collaborative filtering". In: *IEEE Transactions on Image Processing* 16.8 (Aug. 2007), pp. 2080–2095. DOI: 10.1109/TIP.2007.901238.

[30] C. Deledalle, L. Denis, and F. Tupin. "Iterative Weighted Maximum Likelihood Denoising With Probabilistic Patch-Based Weights". In: *IEEE Transactions on Image Processing* 18.12 (Dec. 2009), pp. 2661–2672. DOI: 10.1109/TIP.2009.2029593.

[31] C. Deledalle et al. “NL-SAR: A Unified Nonlocal Framework for Resolution-Preserving (Pol)(In)SAR Denoising”. In: *IEEE Transactions on Geoscience and Remote Sensing* 53.4 (2015), pp. 2021–2038. DOI: 10.1109/TGRS.2014.2352555.

[32] Luís Gómez Déniz, Raydonal Ospina, and Alejandro C. Frery. “Unassisted Quantitative Evaluation of Despeckling Filters”. In: *Remote. Sens.* 9 (2017), p. 389.

[33] G. Di Martino et al. “Benchmarking Framework for SAR Despeckling”. In: *IEEE Transactions on Geoscience and Remote Sensing* 52.3 (2014), pp. 1596–1615. DOI: 10.1109/TGRS.2013.2252907.

[34] Chao Dong et al. “Learning a Deep Convolutional Network for Image Super-Resolution”. In: *European Conference on Computer Vision (ECCV)*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 184–199.

[35] W. Dong et al. “Nonlocally Centralized Sparse Representation for Image Restoration”. In: *IEEE Transactions on Image Processing (TIP)* 22.4 (Apr. 2013), pp. 1620–1630.

[36] A. Dosovitskiy et al. “FlowNet: Learning Optical Flow with Convolutional Networks”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 2758–2766. DOI: 10.1109/ICCV.2015.316.

[37] M. Elad and Y. Hel-Or. “A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur”. In: *IEEE Transactions on Image Processing (TIP)* 10.8 (Aug. 2001), pp. 1187–1193.

[38] Michael Elad and Arie Feuer. “Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images”. In: *IEEE Transactions on Image Processing (TIP)* 6 (Feb. 1997), pp. 1646–58.

[39] Chong Fan et al. “Projections onto Convex Sets Super-Resolution Reconstruction Based on Point Spread Function Estimation of Low-Resolution Remote Sensing Images”. In: *Sensors (Basel, Switzerland)* 17 (Feb. 2017).

[40] S. Farsiu et al. “Fast and robust multiframe super resolution”. In: *IEEE Transactions on Image Processing (TIP)* 13.10 (Oct. 2004), pp. 1327–1344.

[41] S. Foucher, G. B. Benie, and J. -. Boucher. “Multiscale MAP filtering of SAR images”. In: *IEEE Transactions on Image Processing* 10.1 (2001), pp. 49–60.

[42] W. T. Freeman, T. R. Jones, and E. C. Pasztor. “Example-based super-resolution”. In: *IEEE Computer Graphics and Applications* 22.2 (Mar. 2002), pp. 56–65.

[43] A. C. Frery et al. “A model for extremely heterogeneous clutter”. In: *IEEE Transactions on Geoscience and Remote Sensing* 35.3 (May 1997), pp. 648–659. DOI: 10.1109/36.581981.

[44] V. S. Frost et al. “A Model for Radar Images and Its Application to Adaptive Digital Filtering of Multiplicative Noise”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-4.2 (Mar. 1982), pp. 157–166. DOI: 10.1109/TPAMI.1982.4767223.

[45] Langis Gagnon and Alexandre Jouan. “Speckle filtering of SAR images: a comparative study between complex-wavelet-based and standard filters”. In: *Wavelet Applications in Signal and Image Processing V*. Ed. by Akram Aldroubi, Andrew F. Laine, and Michael A. Unser. Vol. 3169. International Society for Optics and Photonics. SPIE, 1997, pp. 80–91. DOI: 10.1117/12.279681. URL: https://doi.org/10.1117/12.279681.

[46] X. Gao et al. “Image Super-Resolution With Sparse Neighbor Embedding”. In: *IEEE Transactions on Image Processing (TIP)* 21.7 (July 2012), pp. 3194–3205.

[47] L. A. Gatys, A. S. Ecker, and M. Bethge. “Image Style Transfer Using Convolutional Neural Networks”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 2414–2423. DOI: 10.1109/CVPR.2016.265.

[48] Ian Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014, pp. 2672–2680. URL: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[49] Yunchuan Gui, Lei Xue, and Xiuhe Li. “SAR image despeckling using a dilated densely connected network”. In: *Remote Sensing Letters* 9 (Sept. 2018), pp. 857–866. DOI: 10.1080/2150704X.2018.1492170.

[50] Ishaan Gulrajani et al. “Improved Training of Wasserstein GANs”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017, pp. 5767–5777. URL: https://proceedings.neurips.cc/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf.

[51] H. Guo et al. “Wavelet based speckle reduction with application to SAR based ATD/R”. In: *Proceedings of 1st International Conference on Image Processing*. Vol. 1. 1994, 75–79 vol.1.

[52] R. M. Haralick, K. Shanmugam, and I. Dinstein. “Textural Features for Image Classification”. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-3.6 (1973), pp. 610–621.

[53] R. C. Hardie, K. J. Barnard, and E. E. Armstrong. “Joint MAP registration and high-resolution image estimation using a sequence of undersampled images”. In: *IEEE Transactions on Image Processing (TIP)* 6.12 (Dec. 1997), pp. 1621–1633.

[54] Russell C. Hardie et al. “High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system”. In: *Optical Engineering* 37.1 (1998), pp. 247 - 260 –14.

[55] J. M. Haut et al. “A New Deep Generative Network for Unsupervised Remote Sensing Single-Image Super-Resolution”. In: *IEEE Transactions on Geoscience and Remote Sensing* 56.11 (2018), pp. 6792–6810. DOI: `10.1109/TGRS.2018.2843525`.

[56] Hu He and Lisimachos Paul Kondi. “A regularization framework for joint blur estimation and super-resolution of video sequences”. In: *IEEE International Conference on Image Processing (ICIP)* 3 (2005), pp. III–329.

[57] K. He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: `10.1109/CVPR.2016.90`.

[58] L. He, H. Qi, and R. Zaretzki. “Beta Process Joint Dictionary Learning for Coupled Feature Spaces with Application to Single Image Super-Resolution”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2013, pp. 345–352.

[59] Yu He et al. “A soft MAP framework for blind super-resolution image reconstruction”. In: *Image and Vision Computing* 27 (Mar. 2009), pp. 364–373.

[60] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. “Super-resolution through neighbor embedding”. In: *Proceedings, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. June 2004, pp. I–I.

[61] Michal Hradiš et al. “Convolutional Neural Networks for Direct Text Deblurring”. In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by Mark W. Jones Xianghua Xie and Gary K. L. Tam. BMVA Press, Sept. 2015, pp. 6.1–6.13. DOI: `10.5244/C.29.6`. URL: `https://dx.doi.org/10.5244/C.29.6`.

[62] X. Hu et al. “RUNet: A Robust UNet Architecture for Image Super-Resolution”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pp. 505–507. DOI: `10.1109/CVPRW.2019.00073`.

[63] Hua Xie, L. E. Pierce, and F. T. Ulaby. “Despeckling SAR images using a low-complexity wavelet denoising process”. In: *IEEE International Geoscience and Remote Sensing Symposium*. Vol. 1. 2002, 321–324 vol.1.

[64] Hua Xie, L. E. Pierce, and F. T. Ulaby. “SAR speckle reduction using wavelet denoising and Markov random field modeling”. In: *IEEE Transactions on Geoscience and Remote Sensing* 40.10 (Oct. 2002), pp. 2196–2212. DOI: `10.1109/TGRS.2002.802473`.

[65] Jun-Jie Huang and Wan-Chi Siu. “Learning Hierarchical Decision Trees for Single Image Super-Resolution”. In: *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)* 27 (Dec. 2015), pp. 937–950.

[66] K. Hung and W. Siu. “Single image super-resolution using iterative Wiener filter”. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2012, pp. 1269–1272.

[67] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167* (2015).

[68] Michal Irani and Shmuel Peleg. “Improving resolution by image registration”. In: *Graphical Models and Image Processing (CVGIP)* 53.3 (1991), pp. 231–239.

[69] Michal Irani and Shmuel Peleg. “Super resolution from image Sequences”. In: vol. ii. July 1990, 115–120 vol.2.

[70] Viren Jain and Sebastian Seung. “Natural Image Denoising with Convolutional Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by D. Koller et al. Vol. 21. Curran Associates, Inc., 2009, pp. 769–776. URL: `https://proceedings.neurips.cc/paper/2008/file/c16a5320fa475530d9583c34fd`
`Paper.pdf`.

[71] K. Jiang et al. “Edge-Enhanced GAN for Remote Sensing Image Superresolution”. In: *IEEE Transactions on Geoscience and Remote Sensing* 57.8 (2019), pp. 5799–5812. DOI: `10.1109/TGRS.2019.2902431`.

[72] K. H. Jin et al. “Deep Convolutional Neural Network for Inverse Problems in Imaging”. In: *IEEE Transactions on Image Processing* 26.9 (2017), pp. 4509–4522. DOI: `10.1109/TIP.2017.2713099`.

[73] Y. Jo, S. Yang, and S. J. Kim. “Investigating Loss Functions for Extreme Super-Resolution”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 1705–1712. DOI: `10.1109/CVPRW50498.2020.00220`.

[74] Younghyun Jo et al. “Deep Video Super-Resolution Network Using Dynamic Upsampling Filters Without Explicit Motion Compensation”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[75] A. Kappeler et al. “Video Super-Resolution With Convolutional Neural Networks”. In: *IEEE Transactions on Computational Imaging (TCI)* 2.2 (June 2016), pp. 109–122.

[76] Andrej Karpathy et al. “Large-scale Video Classification with Convolutional Neural Networks”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[77] Toshiyuki Kato, Hideitsu Hino, and Noboru Murata. “Double Sparsity for Multi-frame Super Resolution”. In: *Neurocomputing* 240.C (May 2017), pp. 115–126.

[78] Michal Kawulok et al. “Deep learning for fast super-resolution reconstruction from multiple images”. In: vol. 10996. 2019.

[79] Michal Kawulok et al. “Deep Learning for Multiple-Image Super-Resolution”. In: *arXiv preprint arXiv:1903.00440* (2019).

[80] Kelvins - ESA’s Advanced Concepts. *PROBA-V Super Resolution*. `https://kelvins.esa.int/proba-v-super-resolution`.

[81] J. Kim, J. K. Lee, and K. M. Lee. “Accurate Image Super-Resolution Using Very Deep Convolutional Networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 1646–1654.

[82] J. Kim, J. K. Lee, and K. M. Lee. “Deeply-Recursive Convolutional Network for Image Super-Resolution”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 1637–1645.

[83] K. I. Kim and Y. Kwon. “Single-Image Super-Resolution Using Sparse Regression and Natural Image Prior”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 32.6 (June 2010), pp. 1127–1133.

[84] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[85] A. Krull, T. Buchholz, and F. Jug. “Noise2Void - Learning Denoising From Single Noisy Images”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2124–2132. DOI: `10.1109/CVPR.2019.00223`.

[86] D. Kuan et al. “Adaptive restoration of images with speckle”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35.3 (Mar. 1987), pp. 373–383. DOI: `10.1109/TASSP.1987.1165131`.

[87] K. Kulkarni et al. “ReconNet: Non-Iterative Reconstruction of Images from Compressively Sensed Measurements”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 449–458. DOI: `10.1109/CVPR.2016.55`.

[88] O. Kupyn et al. “DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8183–8192. DOI: `10.1109/CVPR.2018.00854`.

[89] S. Laine et al. “High-quality self-supervised deep image denoising”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 6968–6978.

[90] A. Lapini et al. “Blind speckle decorrelation for SAR image despeckling”. In: *IEEE Transactions on Geoscience and Remote Sensing* 52.2 (Feb. 2014), pp. 1044–1058. DOI: `10.1109/TGRS.2013.2246838`.

[91] C. Latry and B. Rouge. “Super resolution: quincunx sampling and fusion processing”. In: *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477)*. Vol. 1. 2003, 315–317 vol.1. DOI: `10.1109/IGARSS.2003.1293761`.

[92] Francesco Lattari et al. “Deep Learning for SAR Image Despeckling”. In: *Remote Sensing* 11 (June 2019), p. 1532. DOI: `10.3390/rs11131532`.

[93] Christian Ledig 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 (CVPR)*. 2017, pp. 4681–4690.

[94] Jong-Sen Lee. “Speckle analysis and smoothing of synthetic aperture radar images”. In: *Computer Graphics and Image Processing* 17.1 (1981), pp. 24–32. DOI: `https://doi.org/10.1016/S0146-664X(81)80005-6`. URL: `http://www.sciencedirect.com/science/article/pii/S0146664X81800056`.

[95] J. Lehtinen et al. “Noise2Noise: Learning Image Restoration without Clean Data”. In: *Proceedings of the 35th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR, 2018, pp. 2965–2974.

[96] S. Lei, Z. Shi, and Z. Zou. “Super-Resolution for Remote Sensing Images via Local–Global Combined Network”. In: *IEEE Geoscience and Remote Sensing Letters* 14.8 (2017), pp. 1243–1247. DOI: `10.1109/LGRS.2017.2704122`.

[97] Sen Lei, Zhenwei Shi, and Zhengxia Zou. “Super-Resolution for Remote Sensing Images via Local–Global Combined Network”. In: *IEEE Geoscience and Remote Sensing Letters (GRSL)* 14 (2017), pp. 1243–1247.

[98] S. Lertrattanapanich and N. K. Bose. “High resolution image formation from low resolution frames using Delaunay triangulation”. In: *IEEE Transactions on Image Processing (TIP)* 11.12 (Dec. 2002), pp. 1427–1441.

[99] F. Li et al. “Super Resolution for Remote Sensing Images Based on a Universal Hidden Markov Tree Model”. In: *IEEE Transactions on Geoscience and Remote Sensing (TGRS)* 48.3 (Mar. 2010), pp. 1270–1278. DOI: `10.1109/TGRS.2009.2031636`.

[100] Jingyu Li et al. “HDRANet: Hybrid Dilated Residual Attention Network for SAR Image Despeckling”. In: *Remote Sensing* 11 (Dec. 2019), p. 2921. DOI: 10.3390/rs11242921.

[101] N. Li, N. Huang, and L. Xiao. “PAN-Sharpening via residual deep learning”. In: *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2017, pp. 5133–5136. DOI: 10.1109/IGARSS.2017.8128158.

[102] Yudong Liang et al. “Incorporating image priors with deep convolutional neural networks for image super-resolution”. In: *Neurocomputing* 194 (2016), pp. 340–347.

[103] L. Liebel and M. Körner. “SINGLE-IMAGE SUPER RESOLUTION FOR MULTISPECTRAL REMOTE SENSING DATA USING CONVOLUTIONAL NEURAL NETWORKS”. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLI-B3 (2016), pp. 883–890. DOI: 10.5194/isprs-archives-XLI-B3-883-2016. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLI-B3/883/2016/.

[104] Bee Lim et al. “Enhanced Deep Residual Networks for Single Image Super-Resolution”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 1132–1140.

[105] Ding Liu et al. “Non-local recurrent network for image restoration”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 1673–1682.

[106] X. Liu, Y. Wang, and Q. Liu. “Psgan: A Generative Adversarial Network for Remote Sensing Image Pan-Sharpening”. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018, pp. 873–877. DOI: 10.1109/ICIP.2018.8451049.

[107] J. Long, E. Shelhamer, and T. Darrell. “Fully convolutional networks for semantic segmentation”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[108] A. Lopes et al. “Structure detection and statistical adaptive speckle filtering in SAR images”. In: *International Journal of Remote Sensing* 14.9 (1993), pp. 1735–1758. DOI: 10.1080/01431169308953999. eprint: https://doi.org/10.1080/01431169308953999. URL: https://doi.org/10.1080/01431169308953999.

[109] J. Ma, J. C. Chan, and F. Canters. “Robust Locally Weighted Regression for Superresolution Enhancement of Multi-Angle Remote Sensing Imagery”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS)* 7.4 (Apr. 2014), pp. 1357–1371.

[110] J. Ma, J. Cheung-Wai Chan, and F. Canters. “An Operational Superresolution Approach for Multi-Temporal and Multi-Angle Remotely Sensed Imagery”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS)* 5.1 (Feb. 2012), pp. 110–124.

[111] W. Ma et al. “Achieving Super-Resolution Remote Sensing Images via the Wavelet Transform Combined With the Recursive Res-Net”. In: *IEEE Transactions on Geoscience and Remote Sensing (TGRS)* 57.6 (June 2019), pp. 3512–3527.

[112] X. Ma et al. “SAR Image Despeckling by Noisy Reference-Based Deep Learning Method”. In: *IEEE Transactions on Geoscience and Remote Sensing* 58.12 (2020), pp. 8807–8818. DOI: 10.1109/TGRS.2020.2990978.

[113] Xiao-Jiao Mao, Chunhua Shen, and Yubin Yang. “Image Denoising Using Very Deep Fully Convolutional Encoder-Decoder Networks with Symmetric Skip Connections”. In: *ArXiv* abs/1603.09056 (2016).

[114] Antonio Marquina and Stanley J. Osher. “Image Super-Resolution by TV-Regularization and Bregman Iteration”. In: *Journal of Scientific Computing* 37.3 (Dec. 2008), pp. 367–382.

[115] Marcus Märtens et al. “Super-Resolution of PROBA-V Images Using Convolutional Neural Networks”. In: *arXiv preprint arXiv:1907.01821* (July 2019).

[116] D. Martin et al. “A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics”. In: *Proc. 8th Int’l Conf. Computer Vision*. Vol. 2. July 2001, pp. 416–423.

[117] Giuseppe Masi et al. “Pansharpening by convolutional neural networks”. In: *Remote Sensing* 8.7 (2016), p. 594.

[118] Min Dai et al. “Bayesian wavelet shrinkage with edge detection for SAR image despeckling”. In: *IEEE Transactions on Geoscience and Remote Sensing* 42.8 (2004), pp. 1642–1648.

[119] A. B. Molini et al. “Deep Learning For Super-Resolution Of Unregistered Multi-Temporal Satellite Images”. In: *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. Sept. 2019, pp. 1–5. DOI: 10.1109/WHISPERS.2019.8920910.

[120] A. B. Molini et al. “DeepSUM: Deep Neural Network for Super-Resolution of Unregistered Multitemporal Images”. In: *IEEE Transactions on Geoscience and Remote Sensing* (2019), pp. 1–13. DOI: 10.1109/TGRS.2019.2959248.

[121] Andrea Bordone Molini et al. *Speckle2Void: Deep Self-Supervised SAR Despeckling with Blind-Spot Convolutional Neural Networks*. 2020. arXiv: 2007.02075 [eess.IV].

[122] Michael K. Ng et al. "A Total Variation Regularization Based Super-Resolution Reconstruction Algorithm for Digital Video". In: *EURASIP Journal on Advances in Signal Processing (JASP)* 2007.1 (Dec. 2007), p. 074585.

[123] Michael K Ng et al. "A total variation regularization based super-resolution reconstruction algorithm for digital video". In: *EURASIP Journal on Advances in Signal Processing (JASP)* 2007.1 (2007), p. 074585.

[124] N. Nguyen and P. Milanfar. "An efficient wavelet-based algorithm for image superresolution". In: *Proceedings, International Conference on Image Processing (ICIP).* Vol. 2. Sept. 2000, 351–354 vol.2. DOI: 10.1109/ICIP.2000.899387.

[125] N. Nguyen, P. Milanfar, and G. Golub. "Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement". In: *IEEE Transactions on Image Processing (TIP)* 10.9 (Sept. 2001), pp. 1299–1308.

[126] Nhat Nguyen, P. Milanfar, and G. Golub. "A computationally efficient superresolution image reconstruction algorithm". In: *IEEE Transactions on Image Processing (TIP)* 10.4 (Apr. 2001), pp. 573–583.

[127] S. Parrilli et al. "A nonlocal SAR image denoising algorithm based on LLMMSE wavelet shrinkage". In: *IEEE Transactions on Geoscience and Remote Sensing* 50.2 (Feb. 2012), pp. 606–616. DOI: 10.1109/TGRS.2011.2161586.

[128] Deepak Pathak et al. "Context Encoders: Feature Learning by Inpainting". In: 2016.

[129] A. J. Patti, M. Ibrahim Sezan, and A. Murat Tekalp. "High-resolution image reconstruction from a low-resolution image sequence in the presence of time-varying motion blur". In: *Proceedings of 1st International Conference on Image Processing (ICIP).* Vol. 1. Nov. 1994, 343–347 vol.1.

[130] T. Peleg and M. Elad. "A Statistical Prediction Model Based on Sparse Representations for Single Image Super-Resolution". In: *IEEE Transactions on Image Processing (TIP)* 23.6 (June 2014), pp. 2569–2582.

[131] M. Protter and M. Elad. "Super Resolution With Probabilistic Motion Estimation". In: *IEEE Transactions on Image Processing* 18.8 (Aug. 2009), pp. 1899–1904. DOI: 10.1109/TIP.2009.2022440.

[132] M. Protter et al. "Generalizing the Nonlocal-Means to Super-Resolution Reconstruction". In: *IEEE Transactions on Image Processing (TIP)* 18.1 (Jan. 2009), pp. 36–51. DOI: 10.1109/TIP.2008.2008067.

[133] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015.* Ed. by Nassir Navab et al. Springer International Publishing, 2015, pp. 234–241.

[134] M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch. "EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 4501–4510. DOI: 10.1109/ICCV.2017.481.

[135] J. Salvador and E. Pérez-Pellitero. "Naive Bayes Super-Resolution Forest". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 325–333.

[136] G. Scarpa, S. Vitale, and D. Cozzolino. "Target-Adaptive CNN-Based Pansharpening". In: *IEEE Transactions on Geoscience and Remote Sensing* 56.9 (2018), pp. 5443–5457. DOI: 10.1109/TGRS.2018.2817393.

[137] Kevin Schawinski et al. "Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit". In: *Monthly Notices of the Royal Astronomical Society: Letters* 467.1 (Jan. 2017), pp. L110–L114. DOI: 10.1093/mnrasl/slx008. URL: https://doi.org/10.1093/mnrasl/slx008.

[138] C. J. Schuler et al. "A Machine Learning Approach for Non-blind Image Deconvolution". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1067–1074. DOI: 10.1109/CVPR.2013.142.

[139] S. Schulter, C. Leistner, and H. Bischof. "Fast and accurate image upscaling with super-resolution forests". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 3791–3799.

[140] Z. Shao and J. Cai. "Remote Sensing Image Fusion With Deep Convolutional Neural Network". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11.5 (2018), pp. 1656–1669. DOI: 10.1109/JSTARS.2018.2805923.

[141] H. Shen et al. "A MAP Approach for Joint Motion Estimation, Segmentation, and Super Resolution". In: *IEEE Transactions on Image Processing (TIP)* 16.2 (Feb. 2007), pp. 479–490.

[142] Huanfeng Shen et al. "Super-Resolution Reconstruction Algorithm To MODIS Remote Sensing Images". In: *The Computer Journal* 52 (Jan. 2009), pp. 90–100.

[143] Wenzhe Shi et al. "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 1874–1883.

[144] David Shuman et al. "The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains". In: *IEEE Signal Processing Magazine (SPM)* 3.30 (2013), pp. 83–98.

[145] Martin Simonovsky and Nikos Komodakis. “Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* July 2017, pp. 29–38.

[146] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations.* 2015.

[147] A.B. Smith, C.D. Jones, and E.F. Roberts. “Article Title”. In: *Journal* 62 (Jan. 1920), pp. 291–294.

[148] S. Solbo and T. Eltoft. “Homomorphic wavelet-based statistical despeckling of SAR images”. In: *IEEE Transactions on Geoscience and Remote Sensing* 42.4 (2004), pp. 711–721.

[149] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: `http://jmlr.org/papers/v15/srivastava14a.html`.

[150] Henry Stark and Peyma Oskoui. “High-resolution image recovery from image-plane arrays, using convex projections”. In: *Journal of the Optical Society of America* 6.11 (Nov. 1989), pp. 1715–1726.

[151] Jian Sun, Zongben Xu, and Heung-Yeung Shum. “Gradient Profile Prior and Its Applications in Image Super-Resolution and Enhancement”. In: *IEEE Transactions on Image Processing (TIP)* 20 (Nov. 2010), pp. 1529–42.

[152] J. R. Sveinsson and J. Atli Benediktsson. “Almost translation invariant wavelet transformations for speckle reduction of SAR images”. In: *IEEE Transactions on Geoscience and Remote Sensing* 41.10 (2003), pp. 2404–2408.

[153] H. Takeda, S. Farsiu, and P. Milanfar. “Kernel Regression for Image Processing and Reconstruction”. In: *IEEE Transactions on Image Processing (TIP)* 16.2 (Feb. 2007), pp. 349–366.

[154] Yapeng Tian et al. “TDAN: Temporally Deformable Alignment Network for Video Super-Resolution”. In: *CoRR* abs/1812.02898 (2018).

[155] R. Timofte, V. De, and L. V. Gool. “Anchored Neighborhood Regression for Fast Example-Based Super-Resolution”. In: *IEEE International Conference on Computer Vision (ICCV).* Dec. 2013, pp. 1920–1927.

[156] B. C. Tom and A. K. Katsaggelos. “Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images”. In: *Proceedings, International Conference on Image Processing.* Vol. 2. Oct. 1995, 539–542 vol.2.

[157] Kato Toshiyuki, Hino Hideitsu, and Murata Noboru. “Multi-frame image super resolution based on sparse coding”. In: *Neural Networks* 66 (2015), pp. 64–78.

[158] R.Y. Tsai and T.S. Huang. "Multiframe image restoration and registration". In: *Advances in Computer Vision and Image Processing.* Vol. I. JAI Press. 1984, pp. 317–339.

[159] Caglayan Tuna, Gozde Unal, and Elif Sertel. "Single-frame super resolution of remote-sensing images by convolutional neural networks". In: *International Journal of Remote Sensing* 39.8 (2018), pp. 2463–2479. DOI: `10.1080/01431161.2018.1425561`. eprint: `https://doi.org/10.1080/01431161.2018.1425561`. URL: `https://doi.org/10.1080/01431161.2018.1425561`.

[160] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Instance Normalization: The Missing Ingredient for Fast Stylization". In: *arXiv preprint arXiv:1607.08022* (2016).

[161] D. Valsesia and P. T. Boufounos. "Universal encoding of multispectral images". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Mar. 2016, pp. 4453–4457.

[162] D. Valsesia and E. Magli. "A Novel Rate Control Algorithm for Onboard Predictive Coding of Multispectral and Hyperspectral Images". In: *IEEE Transactions on Geoscience and Remote Sensing (TGRS)* 52.10 (Oct. 2014), pp. 6341–6355.

[163] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. "Deep Graph-Convolutional Image Denoising". In: *arXiv preprint arXiv:1907.08448* (2019).

[164] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. "Image Denoising with Graph-Convolutional Neural Networks". In: *IEEE International Conference on Image Processing (ICIP).* 2019.

[165] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. "Sampling of Graph Signals via Randomized Local Aggregations". In: *IEEE Transactions on Signal and Information Processing over Networks (TSIPN)* 5.2 (June 2019), pp. 348–359.

[166] S. Vitale et al. "Guided Patchwise Nonlocal SAR Despeckling". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.9 (2019), pp. 6484–6498.

[167] L. Wang et al. "Edge-Directed Single-Image Super-Resolution Via Adaptive Gradient Magnitude Self-Interpolation". In: *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)* 23.8 (Aug. 2013), pp. 1289–1299.

[168] P. Wang, H. Zhang, and V. M. Patel. "Generative adversarial network-based restoration of speckled SAR images". In: *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP).* 2017, pp. 1–5.

[169] P. Wang, H. Zhang, and V. M. Patel. “SAR Image Despeckling Using a Convolutional Neural Network”. In: *IEEE Signal Processing Letters* 24.12 (Dec. 2017), pp. 1763–1767. DOI: 10.1109/LSP.2017.2758203.

[170] T. Wang et al. “Aerial Image Super Resolution via Wavelet Multiscale Convolutional Neural Networks”. In: *IEEE Geoscience and Remote Sensing Letters* 15.5 (2018), pp. 769–773. DOI: 10.1109/LGRS.2018.2810893.

[171] Xintao Wang et al. “ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks”. In: *Computer Vision – ECCV 2018 Workshops*. Ed. by Laura Leal-Taixé and Stefan Roth. Springer International Publishing, 2019, pp. 63–79.

[172] Y. Wei and Q. Yuan. “Deep residual learning for remote sensed imagery pansharpening”. In: *2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP)*. 2017, pp. 1–4. DOI: 10.1109/RSIP.2017.7958794.

[173] Y. Wei et al. “Boosting the Accuracy of Multispectral Image Pansharpening by Learning a Deep Residual Network”. In: *IEEE Geoscience and Remote Sensing Letters* 14.10 (2017), pp. 1795–1799. DOI: 10.1109/LGRS.2017.2736020.

[174] N. A. Woods, N. P. Galatsanos, and A. K. Katsaggelos. “Stochastic methods for joint registration, restoration, and interpolation of multiple undersampled images”. In: *IEEE Transactions on Image Processing (TIP)* 15.1 (Jan. 2006), pp. 201–213.

[175] Junyuan Xie, Linli Xu, and Enhong Chen. “Image Denoising and Inpainting with Deep Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012, pp. 341–349. URL: https://proceedings.neurips.cc/paper/2012/file/6cdd60ea0045eb7a6ec44c54d29ed402-Paper.pdf.

[176] Weiying Xie, Yunsong Li, and Xiuping Jia. “Deep convolutional networks with residual learning for accurate spectral-spatial denoising”. In: *Neurocomputing* 312 (2018), pp. 372–381. DOI: https://doi.org/10.1016/j.neucom.2018.05.115. URL: http://www.sciencedirect.com/science/article/pii/S0925231218307124.

[177] H. Xu, G. Zhai, and X. Yang. “Single Image Super-resolution With Detail Enhancement Based on Local Fractal Analysis of Gradient”. In: *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)* 23.10 (Oct. 2013), pp. 1740–1754.

[178] Zhaoyi Yan et al. “Shift-Net: Image Inpainting via Deep Feature Rearrangement”. In: (Jan. 2018).

[179] J. Yang et al. “Coupled Dictionary Training for Image Super-Resolution”. In: *IEEE Transactions on Image Processing (TIP)* 21.8 (Aug. 2012), pp. 3467–3478.

[180] J. Yang et al. “Image Super-Resolution Via Sparse Representation”. In: *IEEE Transactions on Image Processing (TIP)* 19.11 (Nov. 2010), pp. 2861–2873.

[181] J. Yang et al. “Image Super-Resolution Via Sparse Representation”. In: *IEEE Transactions on Image Processing (TIP)* 19.11 (Nov. 2010), pp. 2861–2873.

[182] J. Yang et al. “PanNet: A Deep Network Architecture for Pan-Sharpening”. In: *2017 IEEE International Conference on Computer Vision (ICCV).* 2017, pp. 1753–1761.

[183] Q. Yang et al. “Low-Dose CT Image Denoising Using a Generative Adversarial Network With Wasserstein Distance and Perceptual Loss”. In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1348–1357. DOI: `10.1109/TMI.2018.2827462`.

[184] Wei Yao et al. “Pixel-wise regression using U-Net and its application on pansharpening”. In: *Neurocomputing* 312 (2018), pp. 364–371. DOI: `https://doi.org/10.1016/j.neucom.2018.05.103`.

[185] Q. Yuan et al. “Hyperspectral Image Denoising Employing a Spatial–Spectral Deep Residual Convolutional Neural Network”. In: *IEEE Transactions on Geoscience and Remote Sensing* 57.2 (2019), pp. 1205–1218. DOI: `10.1109/TGRS.2018.2865197`.

[186] Y. Yuan, J. Sun, and J. Guan. “Blind SAR Image Despeckling Using Self-Supervised Dense Dilated Convolutional Neural Network”. In: *ArXiv* abs/1908.01608 (2019).

[187] Matthew D. Zeiler. “ADADELTA: An Adaptive Learning Rate Method”. In: *arXiv e-prints*, arXiv:1212.5701 (Dec. 2012), arXiv:1212.5701.

[188] Y. Zeng et al. “Fusion of satellite images in urban area: Assessing the quality of resulting images”. In: *2010 18th International Conference on Geoinformatics.* 2010, pp. 1–4. DOI: `10.1109/GEOINFORMATICS.2010.5568105`.

[189] Roman Zeyde, Michael Elad, and Matan Protter. “On Single Image Scale-Up Using Sparse-Representations”. In: *Curves and Surfaces.* Ed. by Jean-Daniel Boissonnat et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 711–730.

[190] Roman Zeyde, Michael Elad, and Matan Protter. “On Single Image Scale-Up Using Sparse-Representations”. In: *Curves and Surfaces.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 711–730.

[191] Hongyan Zhang, Liangpei Zhang, and Huanfeng Shen. “A Blind Super-Resolution Reconstruction Method Considering Image Registration Errors”. In: *International Journal of Fuzzy Systems (IJFS)* 17.2 (June 2015), pp. 353–364.

[192] Hongyan Zhang et al. “Super-Resolution Reconstruction for Multi-Angle Remote Sensing Images Considering Resolution Differences”. In: *Remote Sensing* 6 (Dec. 2013).

[193] Jing Zhang, Wenguang Li, and Yunsong Li. “SAR Image Despeckling Using Multiconnection Network Incorporating Wavelet Features”. eng. In: (2019), pp. 1–5.

[194] K. Zhang et al. “Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising”. In: *IEEE Transactions on Image Processing* 26.7 (July 2017), pp. 3142–3155.

[195] K. Zhang et al. “Single Image Super-Resolution With Non-Local Means and Steering Kernel Regression”. In: *IEEE Transactions on Image Processing (TIP)* 21.11 (Nov. 2012), pp. 4544–4556.

[196] Kaibing Zhang et al. “Single Image Super-Resolution With Multiscale Similarity Learning”. In: *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)* 24 (2013), pp. 1648–1659.

[197] Qiang Zhang et al. “Hybrid Noise Removal in Hyperspectral Imagery With a Spatial–Spectral Gradient Network”. eng. In: *IEEE transactions on geoscience and remote sensing* 57.10 (2019), pp. 7317–7329.

[198] Qiang Zhang et al. “Learning a Dilated Residual Network for SAR Image Despeckling”. In: *Remote Sensing* 10 (Feb. 2018), pp. 1–18. DOI: 10.3390/rs10020196.

[199] S. Zhang and E. Salari. “Image denoising using a neural network based nonlinear filter in wavelet domain”. In: *Proceedings. (ICASSP ’05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* Vol. 2. 2005, ii/989–ii/992 Vol. 2. DOI: 10.1109/ICASSP.2005.1415573.

[200] W. Zhang et al. “RankSRGAN: Generative Adversarial Networks With Ranker for Image Super-Resolution”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV).* 2019, pp. 3096–3105. DOI: 10.1109/ICCV.2019.00319.

[201] Xin Zhang et al. “Application of Tikhonov Regularization to Super-Resolution Reconstruction of Brain MRI Images”. In: *Medical Imaging and Informatics.* Ed. by Xiaohong Gao et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 51–56.

[202] Y. Zhang, Y. Xiang, and L. Bai. “Generative Adversarial Network for Deblurring of Remote Sensing Image”. In: *2018 26th International Conference on Geoinformatics*. 2018, pp. 1–4. DOI: 10.1109/GEOINFORMATICS.2018.8557110.

[203] Yulun Zhang et al. “Residual Dense Network for Image Super-Resolution”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 2472–2481.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.