Deep learning Based Image Super-resolution with Adaption and Extension of Convolutional Neural Network Models

by

Ha Viet Khanh

A thesis submitted in the fulfillment for the degree of

Doctor of Philosophy

Center for Signal and Image Processing

Department of Electronic and Electrical Engineering

University of Strathclyde, Glasgow

Supervised by

Professor Jinchang Ren

Professor Stephen Marshall

April 28, 2023

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Ha Viet Khanh

April 28, 2023

Acknowledgements

I would like to thank a number of people for their guidance throughout the duration of my research. In particular, I express my most sincere gratitude to my primary supervisor, Professor Jinchang Ren, for putting his faith in my skills and giving me this opportunity, constant support in guiding my research, writing and other issues, and cosupervisor, Professor Stephen Marshall, for his guidance and encouragement.

Thanks must also go to all authors in my publications, the journal reviewers involved in the peer-reviewed process of my publications. I am grateful as well to all my colleagues during this period for sharing their time and expertise.

This research would have been impossible without the funding from Vietnamese government and University of Strathclyde. I really appreciate their financial support and this research outcomes have returned those effort in the best possible way. I must thank as well the university staff for their guidance and treat.

Finally, a big thank you to my wife for her understanding and encouragement, with loving gratitude to my daughter for her patience.

Abstract

Image super-resolution is the process of creating a high-resolution image from a single or multiple low-resolution images. As one low-resolution image can yield several possible solutions for high-resolution images, image super-resolution is an ill-posed reversed problem. Deep learning-based approaches have recently emerged and blossomed, producing state-of-the-art results in image, language, and speech recognition areas. Thanks to the capability of feature extraction and mapping, it is very helpful to predict the details lost in the low-resolution image. In real-world problems, however, there are many existing factors that significantly affect the super-resolution results, including the model design, characteristics of a low-resolution image, and how features are exploited or combined from given data. This thesis focuses on improving the quality of image reconstruction using CNN-based models by tackling three problems or weaknesses in existing models and algorithms. First, the commonly used skip connection proposed in ResNet lacks discriminative learning ability for image super-resolution. It ignores the fact that natural images have a lot of structure, i.e., strong correlations between neighboring pixels, and some information is more important to predict HR images than others. The second problem that appears in image fusion CNN-based models is inadequately fusing features from multiple sources as well as a lack of regularisation for improving the generality of fusion-based models. Finally, a gradient regularisation approach has recently been proposed to improve the convergence of GAN but has shown instability during training. Hence, addressing

this issue of instability in this method will contribute to a super-resolution area that incorporates GAN.

Initially, contributions are introduced for a single image super-resolution using a novel highway connection-based architecture. The new highway connection, which composes of a non-linear gating mechanism, has efficiently learned different hierarchical features and recovered much more details in pixel-wise based image reconstruction. Besides, the introduced highway connection-based model can achieve faster and better convergence, which is less prominent in training problems than those using common skip connections in the well-known residual neural networks.

Second, a deep learning-based framework has been developed for enhancing the spatial resolution of the low-resolution hyperspectral image (Lr-HSI) by fusing it with the high-resolution multispectral image (Hr-MSI). To tackle the existing discrepancy in spectrum range and spatial dimensions, multi-scale fusion is proposed to efficiently address the disparity in spatial resolution between two source inputs. Furthermore, an auxiliary unsupervised task is proposed, which acts as an additional form of regularisation to further improve the generalisation performance of the supervised task.

Finally, the parameter-free framework that adaptively adjusts the strength of gradient regularisation is proposed to improve the stability and performance of Generative Adversarial Networks. The method proposes automatically differentiating the strength of the regulariser based on the difference in the discriminator's behaviour off the convergence point.

In summary, the outcome of this thesis makes contributions to the deep learning-based super-resolution community by proposing one architecture for single image super-resolution, one fusion-based framework for HSI super-resolution and one adaptive method for gradient regularisation in the Generative Adversarial Network. The novelty and robustness of the proposed methods have been fully

demonstrated by extensive experiments. The quantitative results are compared to the state-of-the-art, and thus give the potential to many users of signal and image analysis to improve the resolution of their final outputs.

Contents

List of Figures

List of Tables

List of Tables

Acronyms

AE	Auto-encoder.
CAE	Contractive Auto-encoder.
CNN	Convolutional Neural Network.
DAE	Denoising Auto-Encoder.
DCGAN	Deep Convolutional Generative Adversarial Network.
DenseNet	Dense Network.
DRAGAN	Deep Regret Analytic Generative Adversarial Networks.
ERGAS	Erreur Relative Globale Adimensionnelle de Synthèse (Relative Dimensionless Global Error).
FID	Frechet Inception Distance.
GAN	Generative Adversarial Network.
GP	Gradient Penalty.

HR	High-Resolution.
Hr-HSI	High-resolution Hyperspectral Image.
Hr-MSI	High-resolution Multispectral Image.
HS	Hyperspectral.
HSI	Hyperspectral Image.
IS	Inception Score.
LR	Low-Resolution.
Lr-HSI	Low-resolution Hyperspectral Image.
LSGAN	Least Squares Generative Adversarial Network.
MS	Multispectral.
MS/HS	Multispectral/Hyperspectral.
MSI	Multispectral Image.
PAN	Panchromatic.
PCA	Principal Component Analysis.
PSNR	Peak signal-to-noise ratio.
ReLU	Rectified Linear Unit.
ResNet	Residual Network.
RGB	Red, Green, and Blue.
RMSE	Root Mean Square Error.
SAE	Sparse Auto-encoder.
SAM	Spectral Angle Mapping.

SISR	Single Image Super Resolution.
SR	Super-Resolution.
SRGAN	Super-Resolution Generative Adversarial Network.
SSIM	Structural SIMilarity.
WGAN	Wasserstein GAN.
WGAN-GP	Wasserstein GAN + Gradient Penalty.

Chapter 1

Introduction

1.1 Motivation and aims

Image resolution describes the level of details contained in an image. The higher the resolution, the higher the quantity of information or precision in the image. The image resolution is first limited by the density of sensing elements in the imaging acquisition device. The higher the number of sensor elements per unit area, the higher the resolution of an imaging system can gain. Image resolution is also affected by the optical lens, where diffraction limit, aberration, and defocusing can lead to image degradation. To increase the image resolution, one straightforward way is to increase the density of the sensor and construct high-quality optical components. The imaging devices necessary to acquire very high-resolution images, therefore, are prohibitively expensive and not practical in most real applications, such as security surveillance cameras and cell phones. The super-resolution (SR) term is generally applied to the problem of overcoming the physical constraints of imaging systems by employing image processing algorithms that are reasonably inexpensive to implement. Basically, image SR is a process to obtain a high-resolution (HR) image from one or multiple low-resolution (LR) images. In other words, image SR aims to estimate the high-

resolution details that are missing in the original image. In recent years, image SR has attracted increasing attention for its wide range of applications, including medical imaging [1, 2], scene recognition [3], security surveillance imaging [4, 5], remote sensing [6], object detection [7, 8], and facial recognition [9, 10], among many others.

Image SR aims to solve an ill-posed inverse problem, as multiple high-resolution images can be reduced to the same LR image. Aggregation from multiple images that are captured the same scene is a plausible approach to producing a single high-resolution image. This approach refers to image fusion-based SR, which requires performing image registration for aligning the low-resolution images, determining the sensor's properties, and regularising the possible solution using a priori information from the image class. Unfortunately, all information is not usually available for image fusion, except for low-resolution images. Single image super-resolution, on the other hand, is based on a single image. This problem is more challenging than the multiple-image case, as less information about the scene is available.

The past few years have witnessed tremendous advances in SR where deep learning-based approaches have been applied. The remarkable capability of extracting and mapping features in Convolutional Neural Network (CNN) has been beneficial for SR task. By learning the relationship between LR-HR images through external training data, the missing details in the image can be precisely estimated. Although CNN-based SR approaches have demonstrated outstanding performance [11], the CNN is still described as a black-box model and the performance of CNN-based models is sensitive to the choice of parameters. There have been an increasing number of architectures and algorithms proposed for image SR, but an optimal solution has not been found yet. Tackling the ill-posed problem of SR using CNN requires improved accuracy in training and generalisation in testing. Achieving both targets is challenging for image SR in different

scenarios.

For single image super-resolution, most CNN-based architectures have widely employed residual connections proposed in ResNet to extend the network's capacity. Although this component helps to increase the network's depth, the very deep structure may not lead to improved performance [12]. Also, common problems in the feed-forward network, exploding gradients [13] and dying ReLU [14], are still present in the deep network. A fundamental question in single image super-resolution is the development of a CNN-based model that can better learn the relationship between the LR and HR images.

Different from conventional RGB image, which divide the light spectrum into broad visible Red, Green, and Blue bands, Hyperspectral Image (HSI) consists of contiguous bands over the specific electromagnetic spectrum, providing the representations of scenes, materials, and sources of illuminations. With the aid of rich spectral bands, HSI has been widely used in a range of applications, including precision agriculture [15], [16], [17], [18], target detection [19], image enhancement [20–22], land cover analysis [23], as well as measurement of chemical substances [24], and change detection [25], where required information relies upon an invisible spectrum. However, due to the limitations of the optical device and signal-to-noise ratio, there is always an inevitable trade-off between the spatial and spectral resolutions in capturing the HSI. This means that HSI images can not be acquired with both high spatial and high spectral resolutions at the same time. As HSI is high-dimensional, generating a high-resolution HSI from a single low-resolution HSI faces severe distortion. Fortunately, when additional high-resolution Multispectral Image (MSI) is provided, the fusion-based SR method can help to reduce the ill-posed problem and achieve promising high-resolution HSI. The question here is how to combine high spatial resolution MSI and high resolution HSI into an integrated product with both high spatial and high spectral resolution. Besides, due to the high dimensional and non-linear capacities of

HSI, research on regularisation methods can help improve the generality and performance of the HSI SR model.

The image can be reconstructed with high pixel-wise accuracy but may not be realistic to human eyes. With the development of GAN [26], images can be generated with high quality and visual perception. However, training the GANs model is usually unstable, and there has been increasing work to improve the stability of GAN training. One of the most effective methods for stabilising GAN training is the gradient penalty [27–29]. However, setting a predefined value of penalty weight is still challenging and can not adapt well to different training status. For example, GANs are notoriously difficult to train and usually face overfiting of the discriminator at any phase of the training process. Unfortunately, increasing the penalty weight to mitigate overfitting will over-penalise the model, resulting in a poor quality image. Therefore, the issue of how to enhance the effectiveness of the gradient regularisation with an adaptive method requires further study.

1.2 Research Objectives

The work presented in this thesis aims to improve the accuracy, generality, and stability of CNN-based models/algorithms for image super-resolution, which ranges from single image super-resolution, fusion-based image super-resolution and image generation. Specifically, the following objectives are defined:

1. To develop an optimal model-based architecture for a single image super-resolution approach that aims to improve accuracy while minimising the common problems of dying ReLU and exploding gradient in CNNs training.

2. To develop a framework that effectively fuses two data sources and a novel regularisation for fusion-based HSI super-resolution.

3. A dynamic scheme to adaptively select the strength of gradient penalty in GANs-based SR. This will improve the convergence of GAN model and ease the

notoriously unstable phenomenon inGAN.

1.3 Methodologies and Contributions

With the limitations already mentioned in mind, this thesis investigates several aspects of improving the quality of image reconstruction, which is in the context of both single image and multiple image super-resolution problems. The major original contributions of this thesis can be summarised as follows:

1.3.1 Methodology in Chapter 3

Chapter 3 deals with the single image super-resolution problem, where a CNNs-based architecture is introduced to learn a function which maps from a space of low-resolution patches to a space of target high-resolution patches. The network is trained to perform end-to-end upsampling from the training image. By examining the shortcomings of a ResNet connection that is widely used in deep networks, a new mechanism gate is designed for better performance in pixel-wise-based super-resolution.

Contributions:

1. Introducing a highway connection-based SISR architecture that differs from the majority of existing models while achieving competitive performance in widely used benchmarks and impressive visual performance.

2. The introduced highway connection-based model can achieve faster and better convergence, which is less prominent to the dying Rectified Linear Unit (ReLU) and exploding gradient problems than those using skip connections.

This work was published as a journal article, as detailed in A.2, Section 1.4.

1.3.2 Methodology in Chapter 4

In Chapter 4, a method is proposed to fuse a low-resolution hyperspectral image with a high-resolution multispectral image to produce a high-resolution hyperspectral counterpart that contains both high spatial and spectral resolutions. A CNN-based approach is employed by extracting and fusing the spatial and spectral features at multiple spatial scales and levels. Furthermore, multitask learning is constructed to regularise the estimator. The proposed multi-task framework benefits from the fact that the observed input image and to-be-estimated output image must share the same content representation.

Contributions:

1. A multi-scale spatial and spectral CNN-based architecture is proposed, which can effectively exploit and fuse the spatial and spectral features of both Hr-MSIs and Lr-HSIs.

2. An additional auxiliary unsupervised task is proposed, which acts as a form of regularisation to further improve the generalisation performance of the supervised task.

3. The above two frameworks are universal and can be widely applied to boost the performance of other CNN-based HSI-SR architectures. A simple addition of the auxiliary task can provide a solid improvement over the baseline.

This work was published as a journal article, as detailed in A.3, Section 1.4.

1.3.3 Methodology in Chapter 5

In Chapter 5, a dynamic schedule method is proposed for the zero-centered gradient penalty in GANs. The regularisation strength is modelled as a function of the training loss. According to the change of training loss, regularisation strength can be dynamically adjusted in the training procedure, thus balancing the underfitting and overfitting of GANs.

Contributions:

1. Improving the stability of the GANs by dynamically adjusting the gradient regularisation strength in the discriminator. This approach improves the convergence of GANs compared to the predefined schedule.

2. The proposed approach is the derivation of a parameter-free method, which does not increase the complexity of the existing model. The quantitative and visual results have validated that using a dynamic schedule can produce the synthesised images with higher quality and diversity.

This work is under preparation for submission, as detailed in C.1, Section 1.4.

1.4 Publications

To support the research in this thesis, the following research articles have been published/produced:

A. Journal publications

1. **Ha, V.K.**, Ren, J., Xu, X., Zhao, S., Xie, G. and Vargas, V.M., Hussain, A., "Deep learning based single image super-resolution: a survey," *International Journal of Automation and Computing*, vol. 16(4), pp. 413-426, 2019.

2. **Ha, V.K.**, Ren, J., Xu, X., Liao, W., Zhao, S., Ren, J. and Yan, G., "Optimized highway deep learning network for fast single image super-resolution reconstruction," *Journal of Real-Time Image Processing*, vol. 17(6), pp.1961-1970, 2020.

3. **Ha, V.K.**, Ren, J., Wang, Z., Sun, G., Zhao, H., and Marshall, S., "Multiscale spatial fusion and auxiliary task for Hyperspectral Image Super-Resolution," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARTS)*, vol. 15, pp.4583-4598, 2022.

B. Conference publications

1. **Ha, V.K.**, Ren, J., Xu, X., Zhao, S., Xie, G. and Vargas, V.M., "Deep learning

based single image super-resolution: A survey," in International Conference on Brain Inspired Cognitive Systems, pp. 106-119, 2018.

C. Journal papers under preparation

1. **Ha, V.K.**, Ren, J., and Marshall, S. "Dynamic schedule for gradient penalty in GANs", plan to submit on May 2023.

1.5 Thesis organisation

The remaining parts of this thesis are organised as follows:

Chapter 2 provides a survey of related work and the research background which is split into three main sections. The first section discusses the concept of super-resolution and reviews the current deep-learning-based approaches, corresponding to the topics covered in the three contribution chapters. The next section outlines the main theoretical notions and mathematical backgrounds that underlie the work of the thesis. Finally, this chapter ends with a summary of the practical applications that most CNN-based models have along with the introduction to some typical architectures. This chapter is supported by two published review articles, as detailed in A.1 and B.1, Section 1.4.

Chapter 3 presents the proposed single image super-resolution model that incorporates a new connection to regulate information through the network. This connection, which composes of a designed nonlinear gating mechanism, is demonstrated to be more suitable for pixel-wise regression than those widely using skip connection.

In Chapter 4, a fusion framework is presented to produce a high-resolution hyperspectral image from a low-resolution hyperspectral image and a high-resolution multispectral one. The difference in spatial and spectral resolutions of the two inputs is facilitated by spatial down-sampling one and fusing them at multiple levels. In addition, an unsupervised auxiliary task is proposed to further improve

the generalisability of proposed model.

In Chapter 5, a dynamic method is studied to control the gradient regularisation strength based on the change in the training loss. With this method, Experiments on both synthesis and real data have shown its efficacy in dealing with instability of recent proposed gradient penalty and its capacity of generating highly realistic images.

Finally, Chapter 6 concludes the work of the thesis and along with prospects for future work that can be expected to further improve the performance of the proposed methodologies.

Chapter 2

Related work and research background

Based on the motivations and objectives summarised in Chapter 1, the related work and relevant research background are introduced in this chapter. Section 2.1 briefly describes the different types of digital images. Section 2.2 first describes the concept of super-resolution, followed by reviews of the previous work on image super-resolution, including single image super-resolution, fusion-based image super-resolution, and image generation. A comprehensive description of the elements in convolutional neural networks is provided in Section 2.3. Although the great details of this topic are covered in various textbooks, the most relevant aspects used in the context of the thesis are briefly presented. Section 2.4 provides an overview of the network architectures, which can be widely applied to the CNN-based model of image super-resolution. Finally, a brief summary is given in Section 2.5.

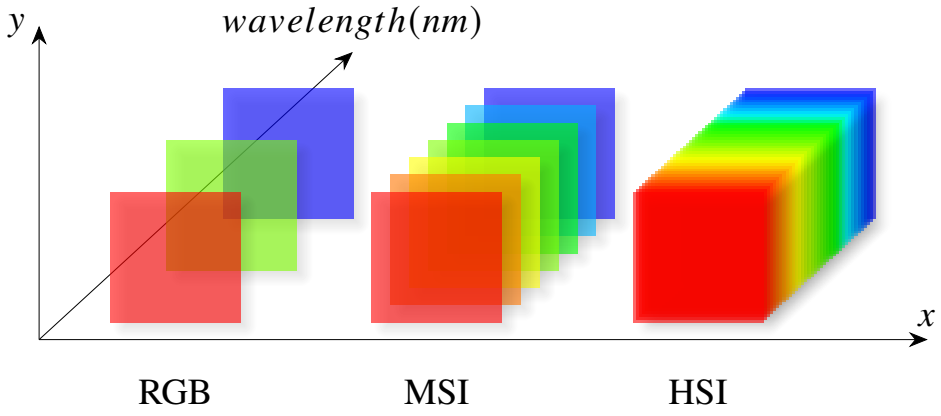


Figure 2.1: The difference between RGB, MSI, and HSI images.

2.1 Digital image

In digital imaging, understanding the content of captured images is the first step toward achieving fine-grained image reconstruction. Grayscale, colour, and hyperspectral images are among the most well-known types of images. A grayscale term refers to an image in which each pixel value presents the intensity of incident light. The darkest black to the brightest white are typically displayed in the grayscale image. In other words, the image only features black, white, and gray, in which gray has many levels. A colour image is a combination of three separate monochromatic images. Each pixel represents three data measurements, each captured from a different coloured filter. Different from conventional RGB images, which divide the light spectrum into broad visible Red, Green, and Blue bands, Hyperspectral Image (HSI) consists of contiguous bands over the specific electromagnetic spectrum, providing the representations of scenes, materials, and sources of illumination.

2.2 Deep learning-based Super-Resolution

The term "super-resolution" relates to image-enhancement techniques. As a result, the image has an increased number of pixels. This section provides an overview of the most prevalent uses of the term, as it can be applied to a variety of scenarios. Unless otherwise noted, the phrase super-resolution refers to the increase in the spatial resolution.

2.2.1 Single image super-resolution

Single image super-resolution (SISR) aims to reconstruct a high-resolution (HR) image from its corresponding low-resolution (LR) version. This image reconstruction problem is known as an ill-posed problem since there exist multiple solutions HR for any given LR image. Super-resolution methods can be divided into three main categories, i.e., interpolation-based, reconstruction-based, and learning-based methods. Interpolation-based methods approximate missing pixels based on the values of surrounding pixels. Image interpolations include: nearest neighbor, bilinear, bicubic, spline, sinc, lanczos, etc. Interpolation-based methods are the most classical and straightforward and tend to smooth the reconstructed image regardless of the image statistics. Reconstruction-based methods predefine certain knowledge prior or constraints, such as local structure similarity, non-local means, or edge prior to restrict the possible solution space. This prior knowledge is broad and varies depending on a particular dataset, which makes them challenging in practical applications. Example-based methods attempt to reconstruct the prior knowledge from a massive amount of internal or external LR-HR patch pairs. The relationship between LR and HR was applied to an unobserved LR image to recover the most likely HR version. Example-based methods can be classified into two types: internal learning and external learning-based methods.

Internal learning-based methods: The natural image has a self-similarity property that tends to recur many times within the same scale or across different scales inside the image. Glasner et al [30] first compared the original image and

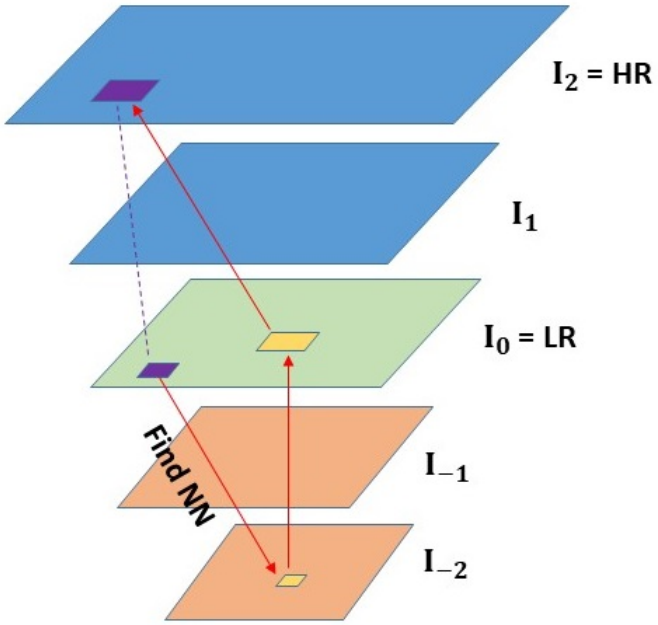


Figure 2.2: Pyramid model [30] for SISR. From the bottom, when a similar patch is found in a down-scale patch (dark green, dark red), its parent (light green, light red) is copied to an unknown HR image with an appropriate gap in scale and support of different kernels.

multiple cascades of images of decreasing resolution to determine the similarity. After that, a scale space pyramid procedure was used to match LR and HR pairs, as shown in Fig 2.2. To take advantage of abundant feature similarity, Huang et al. [31] expanded the search space to include both planar perspective and affine transforms of patches. However, the most important limitation lies in the fact that self-similarity-based methods lead to high complexity of computation due to huge numbers of searches, and the accuracy of algorithms varies according to natural properties of images.

External learning-based methods: The external learning-based methods attempt to search for similar information from other images or patches instead.

It was first introduced to estimate an underlying scene X with the given image data Y [32]. The algorithm aims to learn the posterior probability $P(X|Y) = \frac{1}{P(Y)}P(X,Y)$, by adding image patches X and its corresponding scenes Y as nodes in a Markov network. It was then applied for generating super-resolution images, where the input image is LR and the scene to be estimated is replaced by a HR image [33].

Locally linear embedding (LLE) is one of the manifold learning algorithms, based on the idea that the high dimensionality may be represented as a function of a few underlying parameters. LLE begins by finding a set of nearest neighbors for each point that can best describe that point as a linear combination of its neighbors. It is then determined to find the low-dimensional embedding of points, such that each point is still represented by the same linear combination of its neighbors. However, one of the disadvantages is that LLE handles non-uniform sample densities poorly because the feature represented by the weights vary according to regions in sample densities. The concept of LLE was also applied in SISR neighbor embedding [34], where the features are learned in the LR space before being applied to estimate HR images. There were several other studies based on local linear regression, such as: ridge regression [35], anchored neighborhood regression [36, 37], random forest [38], and manifold embedding [39].

Another group of algorithms that have received attention is sparsity-based methods. In the sparse representation theory, the data or images can be described as a linear combination of sparse elements chosen from an appropriately over-complete dictionary. Let $D \in R^{n \times K}$ be an over-complete dictionary ($K \gg n$), we can build a dictionary for most scenarios of inputs and then any new image (patch) $X \in R^n$ can be represented as $X = D \times \alpha$, where α is a set of sparse coefficients. Hence, there were dictionary learning problems and sparse coding problems to optimise D and α, respectively. The objective function for standard

sparse coding is:

$$\arg\min_{D} \sum_{i=1}^{N} \arg\min_{\alpha_i} \frac{1}{2} \parallel x_i - D\alpha_i \parallel^2 + \lambda\|\alpha_i\| \tag{1}$$

Unlike standard sparse coding, the SISR sparsity-based method works with two dictionaries to learn the compact representation of these patch pairs. Assuming that the observed low-resolution image Y is blurred and a down-sampled version of the high-resolution X:

$$Y = S.H.X \tag{2}$$

where H represents a blurring filter and S the down-sampling operation. Because the dictionary is over-complete or very large, the sparest α_0 can be unique for both dictionaries under mild conditions. Hence, the joint sparse coding can be represented as:

$$\begin{aligned} \arg\min_{D_x, D_y} \sum_{i=1}^{N} \arg\min_{\alpha_i} \frac{1}{2} \parallel x_i - D_x\alpha_i \parallel^2 \\ + \frac{1}{2} \parallel y_i - D_y\alpha_i \parallel^2 + \lambda\|\alpha_i\| \end{aligned} \tag{3}$$

The two dictionaries of high-resolution D_h and low-resolution D_l are co-trained to find the compact coefficients $\alpha_h = \alpha_l = \alpha$ [40], such that sparse representation of a high-resolution patch is the same as the sparse representation of the corresponding low-resolution patch. A dictionary D_l was first trained to best fit the LR patches, then the D_h dictionary was trained that worked best with α_l. When these steps were completed, α_l was then used to recover a high-resolution image based on the high-resolution dictionary D_h. One of the major drawbacks of this method is that the two dictionaries are not always linearly connected. Another problem is that HR images are unknown in the testing phase, hence the equivalence constraint on the HR sparse representation does not guar-

antee as it has been done in the training phase. Yang et al. [41] suggested a coupled dictionary learning process to pose constraints for two spaces of LR and HR. The main disadvantage of this method is that both dictionaries are assumed to be strictly aligned to achieve alignment between α_h and α_l or the simplifying assumption of $\alpha_h = \alpha_l$. To avoid this invariance assumption, Peleg et al. [42] connect α_h, α_l via a statistical parametric model. Wang et al. [43] proposed semi-couple dictionary learning, in which two dictionaries are not fully coupled. It was based on an assumption that there exists a mapping in sparse domain f(.): $\alpha_l \rightarrow \alpha_h$ or $\alpha_h = f(\alpha_l)$. Therefore, the objective function has one additional error term $\|\alpha_h - f(\alpha_l)\|^2$ and other regularisation terms. A Beta process for joint dictionary learning was proposed in [44], which enables the decomposition of these sparse coefficients to the element multiplication of dictionary atom indicators and coefficient values, providing the much-needed flexibility to fit each feature space. Although sparsity-based algorithms yield good results, they have remaining limitations in feature extraction and mapping, which are important for the success of external example-based SR methods. A large dataset or model can pose challenges for sparsity-based methods. For example, finding a dictionary of image patches from a huge image dataset would significantly increase the computational time and work burden. Moreover, the sparsity-based method is an iterative process, which usually solves separately optimization problems on usage. Therefore, it has a limitation of jointly predicting the SR images.

In recent years, deep learning-based algorithms, especially those based on Convolutional Neural Networks (CNNs), have recently proved extremely powerful for SISR tasks [11, 45–49]. The CNN-based SR methods are fully feed-forward [11], and do not need to solve complex optimisation on usage. In other words, all layers in a feed-forward model are jointly optimised for final prediction, instead of handling each component separately in conventional methods. As the result, the CNN-based SR methods can provide an efficient computation while also al-

lowing layers to learn hierarchical features from images. With the development of the deep network, i.e., ResNet [50], there have been increasing architectures employing residual learning approach to improve the performance of SR model. Indeed, ResNet can benefit most tasks but does not provide sufficient discriminative learning ability for SR. The skip connection in ResNet would treat all pixels of previous feature maps with an equal weight of 1. It does not offer detailed attention to some channels or spatial parts of feature maps, meaning capturing the structure of natural images is still limited.

2.2.2 Fusion-based hyperspectral image super-resolution

There is always an inevitable trade-off between the spatial and spectral resolutions in captured hyperspectral images. Because of the high dimension of hyperspectral imaging (HSI), recontructing Hr-HSI from only Lr-HSI usually introduces spectral or spatial distortions. Given the auxiliary information, such as a panchromatic, RGB, or multispectral image, the fusion-based HSI super-resolution has received increasing attention recently [21, 51–60]. This technique is originated from image pan-sharpening, in which a high-resolution Panchromatic (PAN) image and a low-resolution MS image are fused to construct a single high-resolution MS image [61, 62].

The MS/HS fusion is more challenging to solve than the pan-sharpening due to several major factors: (1) the PAN image is collected with a higher spatial resolution than an MS image because the PAN image contains only one wideband with a broad spectral range and therefore enables it to be captured with smaller detector while keeping a high signal-to-noise ratio; (2) the hyperspectral data has a high spectral dimension, which means a larger number of variables is required to use for MS/HS fusion.

Recently, various techniques have been proposed for MS/HS fusion based HSI SR. Typically, the HSI SR approaches can be roughly categorised into three

classes: i.e., dictionary-based sparse representation, maximum a-posteriori-based Bayesian, and deep learning. In sparse representation approaches, the source images are represented by a dictionary and the corresponding sparse coefficients, where the matrix factorisation and the tensor factorization are most commonly used.

A matrix factorisation can be used to decompose high dimensional data and fuse MS/HS data [63, 64]. Dong et al. [65] proposed a non-negative structured sparse representation (NSSR) method to jointly estimate the dictionary and sparse coefficients based on the prior knowledge of the spatial-spectral sparsity in the source images. Since the observed Lr-HSIs and Hr-MSIs have captured the same scene as the target Hr-HSIs, they are assumed to share the same underlying spectral materials or *endmembers*. Lanaras et al. [66] proposed the coupled spectral unmixing (CSU) method for the fusion problem, where the Lr-HSI and Hr-MSI are alternatively unmixed to estimate the spectral basis and abundances.

The tensor factorisation is an extension of the matrix factorisation to higher-order tensors, which are used to extract the underlying factors in high-order dataset [57,67,68]. Dian et al. [67] proposed non-local sparse tensor factorisation (NLSTF) assuming that each patch of estimated Hr-HSI as a core coefficient tensor and dictionaries of width the mode, the height mode and the spectral mode. The non-local spatial self-similarity of Hr-MSI is exploited through a clustering method to constrain the spatial correlation in the Hr-HSI. In other work, they proposed a low tensor-train rank representation (LTTR) [69] method by considering Hr-HSI as a four-dimension tensor and used non-local LTTR prior from Hr-HSI to regularise the fusion problem. The various tensor factorisation based approaches have been used for the super-resolution fusion problem, including non-local patch tensor sparse representation [70], subspace-based low tensor multi-rank regularisation [71], etc. The issue with the factorization-based method is that there is not a single unique decomposition and it is difficult to know how to choose the

basic elements or factorization rank. Some prior information of the Hr-HSI are introduced to regularise the super-resolution problem in previously mentioned work, including priors of spectral unmixing [66], nonlocal spatial similarities [65], sparse priors [67], and the nonlocal LTTR prior [69].

A Bayesian approach is a different framework, which typically builds the posterior distribution with maximum a posteriori probability (MAP) based on the prior knowledge of the observation model [51–53, 72–75]. The major drawback of all these methods is that their performance is dependent on the prior assumption, for example, the pre-defined spectral response function for generating Hr-MSI, and therefore less flexible to adapt to unobserved real-world datasets.

As compared with conventional methods, deep learning-based methods impose fewer assumptions on the prior knowledge of the to-be-estimated Hr-HSI and achieve good performance for MS/HS fusion task. The CNN is commonly employed network structure for image fusion, including pan-sharpening [61,76,77] and a MS/HS fusion [54,55,59,60]. Hu et. al [60] proposed a CNN-based HS/MS fusion architecture including channel attention and spatial attention to refine details from Lr-HSI and Hr-MSI, respectively. The work in [54, 55, 59] formulates a fusion problem with iterative algorithms, then incorporates CNN to repeatedly refine estimation at each step. The summation and concatenation are the only two operations for fusing feature maps or tensors, and both operations require that tensors have identical spatial dimensions. The common drawback of these CNN-based models is that they employ upsampling/downsampling the Lr-HSI/Hr-MSI to a desired space for convenience in order to fuse features by summation or concatenation. This will introduce more noise, which makes it difficult to refine information in a high-dimensional image like HSI as well as incur a high computational cost. For example, both Lr-HSI and Hr-MSI are one-step upsampled to an image size of Hr-HSI [55, 59, 60] or Hr-MSI is upsampled to an image size of Hr-HSI at each repeated phase in [54]. How to effectively fuse features extracted

from high spatial and high spectral resolution images is a central question in the fusion-based method. Without solving this question, more extracted features are redundant, and the model cannot achieve high performance. Second, there have been very few research papers on novel regularisation that can especially apply for HS/MS fusion. The widely used regularisation in CNN, such as L_1, L_2 may not be sufficient for specific problem.

2.2.3 Generative Adversarial Networks for Image super-resolution

The Generative Adversarial Network (GAN) [26] has been established as a key deep neural network model for unsupervised high-resolution image generation. The GAN typically is composed of a generator and a discriminator. The generator and the discriminator are parametrized as deep neural networks and optimise a mini-max (or zero-sum game) objective function. The task of the generator is to generate an image from a noise vector, whereas the discriminator's task is to distinguish the generated image (fake image) from the real image. The GAN will converge to its Nash equilibrium when the discriminator cannot distinguish the faked image from the real one. Ledig et al. [78] introduce a Super-Resolution Generative Adversarial Network (SRGAN) model in which a generative network up-samples the LR image to SR image and a discriminative network distinguishes between the ground truth HR image and SR image. The pixel-wise quality assessment metric has been widely criticised for showing poor perception quality for human vision. Incorporating an adversarial loss from GAN can solve the problem and produce highly perceptive, naturalistic images. Another work [79] that employ adversarial loss on the feature domain has improved the perceptual quality of the SR result significantly. First, the generated and real images are first fed to a pre-trained network to obtain intermediate feature maps. Then, the discriminator can use those feature maps as its input. One of the key weaknesses

of the previously mentioned GAN-based SR approaches is the quality of the reconstructed images, which is affected by the performance of GAN. Also, they are all supervised GAN-based approaches that rely on a pair of high-quality and low-resolution images, which is limited. They have still not tackled problems with super-resolutions, such as unknown degradation factors. Therefore, enhancing the stabilities of GANs as will be discussed below, using self-supervised method [80] that does not use high-quality reference images, and learning the degradation factor in an unsupervised manner [81,82] are among the straightforward approaches in the image super-resolution.

Compared to other deep networks, GAN models suffer from several training issues, such as non-convergence, mode collapse [83–87], diminishing gradients [27, 88–90]. To address the vanishing gradient issue due to the utilisation of cross-entropy loss in the original GAN, the Least Squares Generative Adversarial Network (LSGAN) [90] employs a least squares loss function for the discriminator. By using the Wasserstein-1 distance for the loss function, the Wasserstein GAN (WGAN) [27,88] resolves the non-convergence and mode collapse problems. The hinge loss-based GAN [89] and maximum mean discrepancy-based GAN [91–93] have also shown improvement over the original GAN.

Introducing new loss functions is not the only option to enhance the stability of GAN training. Deep Convolutional Generative Adversarial Network (DCGAN) [86] is one of the earliest and most important modifications to the GAN design. The configurations for model architecture and training in DCGAN lead to surprising stable training and high performance. The Progressive Growing GAN [94] progressively increases the model depth during the training process, can produce large, high-quality images of size 1024×1024, which is challenging for all previous GAN models.

The Lipchitz regularisations [27,29,95] have shown great success in stabilising general GAN training. The gradient penalty in [27] significantly improve the

stability of WGAN. Specially, using zero-center gradient penalty [29], an original GAN can generate large high-quality images without employing the Progress Growing GAN. Spectral normalisation technique [95] enforces Lipchitz continuity in operator space by regularising the spectral norm of the weights in the network. This technique has significantly improved the stability of GAN and has been widely used in various models.

Generally, GAN-based SR models can produce large, high-quality images, which was previously a difficult task for working with the large down-sampled factor. Although various regularisation approaches have been proposed for stabilising the training of GAN, achieving both convergence and stability for GAN is still challenging. For example, the gradient penalty methods in [27, 28] do not usually lead to convergence in most cases compared to zero-center gradient penalty [29] but guarantee more stable training. This leads to an approach that either improving the convergence of the former or the stability of the latter.

2.3 Convolutionnal Neural Networks

A CNN is a type of multi-layer perceptron that is especially well suited to image and audio processing. It is based on biological processes and takes advantage of the patterns of connectivity found between neurons in the human visual cortex. CNN is typically composed of multiple layers, such as convolution, pooling, and fully-connected layers, as well as activation functions, to learn the hierarchies of features automatically and adaptively via a back-propagation algorithm. The convolution and pooling layers perform feature extraction, while the fully-connected layers map extracted features to the output. In some cases, not all three types of layers are used. For example, CNN-based super-resolution models do not use a fully connected layer. The convolution layer, using particular strides, can replace the pooling layer for feature reduction. In this section, we

first introduce different related components of the CNN-based SR model that will be employed in Chapter 3, Chapter 4, and Chapter 5. Then the loss functions and quantitative metrics for image SR are presented. Finally, some architectures in single image super-resolution are reviewed.

2.3.1 Receptive field

The receptive field is a basic concept in deep CNNs, referring to the area of an input image that was used to calculate a specific feature [96]. When applying a convolution layer to an image, another image can be produced, whose each pixel has a limited view of the original image. The output size of the new image depends on the choice of kernel size, zero padding, and stride. The two basic hyper-parameters that determine the convolution operation are the kernel size and the number of kernels. The former is normally 3×3, but sometimes 5×5 or 7×7 and the latter is arbitrary, which defines the depth of output feature maps. The convolution operation does not allow the center of the kernel to overlap the input's outermost, and when the kernel size is greater than 1, this operation always leads to a reduction in the size of the output compared to the input. As a result, the CNN model can not go too deep. Padding, often zero padding, is a strategy to handle this issue, where rows and columns of zeros are placed on each side of the input, so as to fit the center of the kernel on the outermost elements and maintain the same size during the convolution operation. A stride is the distance between two successive kernel positions.

2.3.2 Convolution layer

Convolution layers are a specific sort of linear operation that is used for feature extraction in CNNs. It is performed by applying a small array of numbers called a kernel over the input, which is an array of numbers called a tensor. At each point

of the tensor, an element-wise product between each element of the kernel and the input tensor is computed and added together to generate the output value in the corresponding place of the output tensor, referred to as a feature map. This process is repeated with different kernels to generate a number of feature maps that represent different features of the input tensors; thus, different kernels can be thought of as different feature extractors.

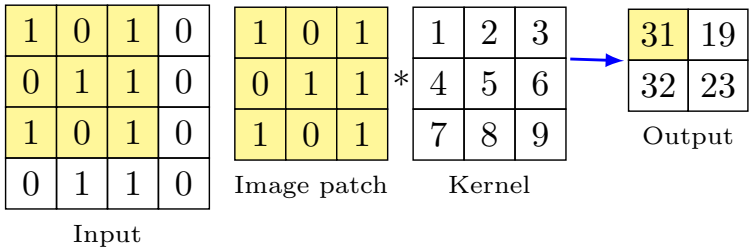


Figure 2.3: An example of convolution with kernel size = 3 and stride = 1.

The mathematical formulation of a 2-D convolution is given by:

$$y[i,j] = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h[m,n] \cdot x[i-m, j-n] \tag{2.1}$$

where x represents the input image matrix to be convolved with the kernel matrix h to result in a new matrix y. The indices i and j refer to the image matrices, whereas m and n refer to the kernel matrices. If the convolution kernel is 3×3 in size, the indices m and n range from -1 to 1.

Assuming the input has a square shape, the output size for convolution can be formulated as follows:

$$O = \frac{I + 2 \times P - K}{S} + 1 \tag{2.2}$$

where O, I are the output and input size; P, K, and S denote a padding, kernel size, and stride, respectively.

2.3.3 Transposed convolution

The requirement for transposed convolutions [97] typically arises from the desire to perform a transformation in the opposite direction of a normal convolution while retaining a connectivity pattern compatible with said convolution. This transformation can be used as the decoding layer of a convolutional autoencoder or to project feature maps into a higher-dimensional space. The process of the transposed convolution is illustrated in Fig. 2.4. Each element in the input is multiplied by the kernel to produce the corresponding values in the output. The final output, shown at the bottom of Fig. 2.4, is the sum of the products.

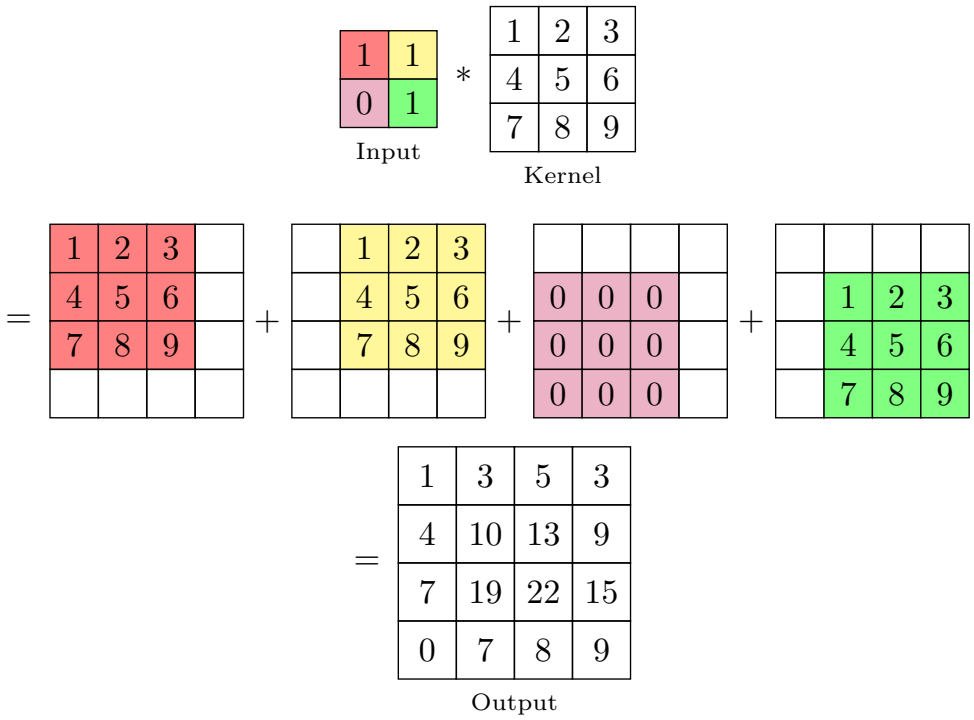


Figure 2.4: Transposed convolution with kernel size = 3, stride = 1 and padding = 0.

As opposed to convolution in Eq. (2.2), the formula for the transposed convolution can be calculated as follows:

$$O = (I - 1) \times S - 2 \times P + K \tag{2.3}$$

2.3.4 Pooling layer

The pooling (POOL) layer, which is generally used after a convolution layer, is a down-sampling operation. Using the pooling function, the output is replaced by a summary statistic of the neighbourhood inputs. Therefore, the pooling layer can reduce the dimensions of feature maps without additional parameters. Another advantage of the pooling operation is that the representation becomes invariant to small spatial translations of the input. The invariance property benefits the task, i.e., classification, for which detecting whether an object is present in the image is more important than its exact location. The most common pooling functions are max pooling and average pooling, where the maximum and average values of nearby inputs are taken, respectively. Fig. 2.5 illustrates an example of max pooling and average pooling. Assuming a one-pixel object has a pixel value of 8 at (0, 0), it is still included in the output using the max pooling function. If this object is not present at (0, 0), but at (1, 4) instead, the output will still be 8.

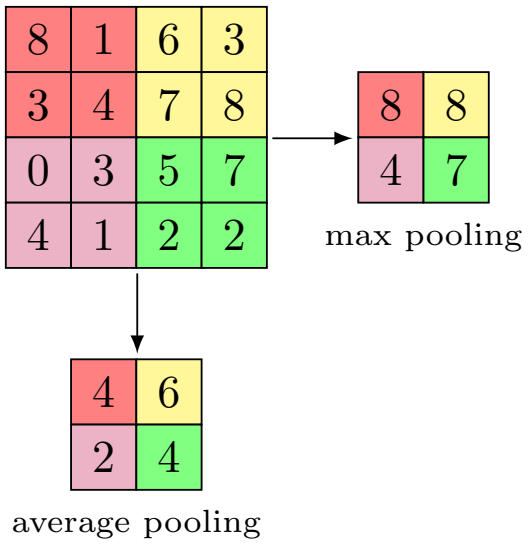


Figure 2.5: Max pooling and average pooling with 2×2 pool size, stride = 2.

Average pooling is distinct from max-pooling because it preserves less sig-

nificant information about a block. While max pooling discards them entirely by selecting the maximum value, average pooling incorporates them. This can be advantageous in a variety of circumstances where such knowledge is necessary. For example, max-pooling performs well in generalising the line on a black background, however, the line on a white backdrop vanishes completely. While average pooling does not suffer from such extreme consequences, max pooling is more successful when the images have a comparable dark background.

2.3.5 Fully-connected layer

The final convolution or pooling layer's output feature maps are typically flattened, that is, converted to a one-dimensional (1D) array of numbers (or vector), and connected to one or more fully connected layers, namely dense layers, in which the corresponding input is connected to each output via a learnable weight. Once the features extracted by the convolution layers and down-sampled by the pooling layers are formed, they are transferred to the network's final outputs, such as the probabilities for each class in classification tasks, using a subset of fully connected layers. Typically, the final fully linked layer has the same number of output nodes as the requested number of classes. In Fig. 2.6, the final output for two classes is followed by two fully-connected layers, which contain 11 and 6 nodes, respectively.

2.3.6 Activation functions

For image super-resolution, the commonly used activation functions are Sigmoid, Softmax, and Rectified Linear Unit (ReLU) [98], as detailed below.

Sigmoid A sigmoid activation function of a value z is defined by

$$f(z) = \frac{1}{1 + e^{-z}} \tag{2.4}$$

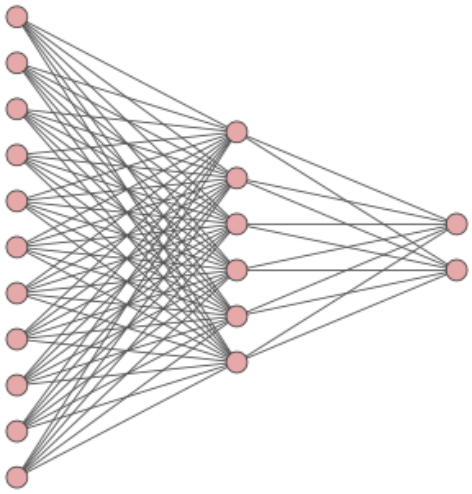


Figure 2.6: Fully-connected layer.

When $z \to -\infty$, $f(z) \to 0$; and when $z \to \infty$, $f(z) \to 1$. For this property, the sigmoid function is often used to produce the probability of a binary output. The sigmoid function was previously used as the activation function of the hidden layer to keep it within a range of [0, 1]. However, the hidden activation values are saturated to 0 or 1 when the input is strongly negative or positive, respectively. In other words, when $f(z)$ is close to 0 or 1, the derivative of sigmoid $f'(z) = f(z)(1 - f(z))$ is nearly 0 and can cause a neural network to be stuck in training. Furthermore, the sigmoid function also slows the learning process due to its non-zero mean as shown in Fig. 2.7.

Softmax A softmax activation function of a vector z of K real numbers is defined by

$$f(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.5}$$

where $\mathbf{z} = (z_1, \cdots, z_K)$ and $i = 1, \cdots, K$.

The Softmax is an activation function that is mostly used in a classification task. The Softmax function, when given an input vector, returns the probability distribution for all the classes of the model. The total of the distribution's values equals 1. Typically, the Softmax function receives the output of the last layer of

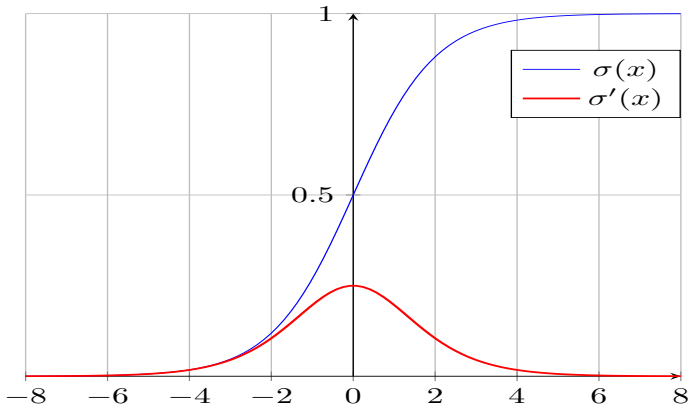


Figure 2.7: Sigmoid function and its derivative.

a neural network as its input.

ReLU A Rectified Linear Unit (ReLU) activation function of a value z is defined by:

$$f(z) = \begin{cases} z, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.6}$$

Since the ReLU was first introduced, it has been one of the most often used activation functions in deep neural networks [50, 98–100]. The ReLU activation function has several advantages. First, this activation helps to train or run the model faster. It is a simple function that does not involve any computational cost. Furthermore, the slope of the gradient does not saturate when the input gets large in comparison with the sigmoid and tanh functions. The second advantage of the ReLU is sparsity. Some of the outputs are set to zero if their inputs are negative. As a result, it is likely for any given unit not to activate at all, which leads to an a sparse network. The neurons in a sparse network are probably capturing meaningful aspects of the problem.

However, there are some problems with the ReLU activation, such as the exploding gradient and dying ReLU. It can be seen from Eq. (2.6), the output of the ReLU function is unbound for $z \geq 0$. The successive multiplication of large

positive output might cause the exploding problem, as the output of a ReLU will be an input to another ReLU. The dying ReLU occurs when the input to ReLU is negative and ReLU neurons would output 0 for the input. As the input does not receive the gradient to improve learning due to the 0 output, the output may not escape the negative part of the ReLU. Even if not all neurons are dead, the majority of them being inactive would lead to several problems, including computational cost, time consuming, and poor performance.

2.3.7 Batch Normalisation layer

In neural networks, the output of the first layer feeds into the second, the output of the second layer feeds into the third layer, and so on. During training, the parameters of a layer change, as does the distribution of inputs to the subsequent layers. The *Internal covariate shift* [101] is defined as a change in the input distribution to the network. When the input distribution changes, the hidden layers in turn try to learn and adapt to the new distribution. The training process is slowed down and badly affected as a result of this. Another issue arises when the statistical distribution of the input to the networks differs significantly from the input it has previously seen. The proposed solution to the *Internal covariate shift* problem is to move all mini-batches to the standard location. Each layer and activation function would deal with their input data within a closely resemble range. Using batch normalisation [101], each layer's inputs are normalised by using the mean and standard deviation (variance) of the values in the current batch.

$$\mu_{\mathcal{B}} = \frac{1}{m}\sum_{i=1}^{m} x_i \qquad //mini-batch\ mean$$

$$\sigma^2_{\mathcal{B}} = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad //mini-batch\ variance$$

$$\hat{x_i} = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad //nomalise$$

$$y_i = \gamma \hat{x_i} + \beta \qquad //scale\ and\ shift$$

where γ and β are parameters to be learned. $\mathcal{B}$ is the set of mini-batch ($\mathcal{B} = x_1$,x_2...x_m) and m is batch size. During testing, the mean $\mu_{\mathcal{B}}$ and deviation $\sigma_{\mathcal{B}}$ are estimated based on the entire training dataset.

2.3.8 Autoencoders - Unsupervised learning

An Auto-encoder (AE) [102] is an unsupervised learning technique that employs neural networks to learn representations. A neural network topology is specifically designed to induce the bottleneck, forcing a compressed representation of the original input as shown in Fig. 2.8. Principal Component Analysis (PCA) and AE can produce the same result if no non-linear function is used in the AE and the number of neurons in the hidden layer is smaller in dimension than the input. Otherwise, the AE can discover a new subspace. In the AE, the unlabelled

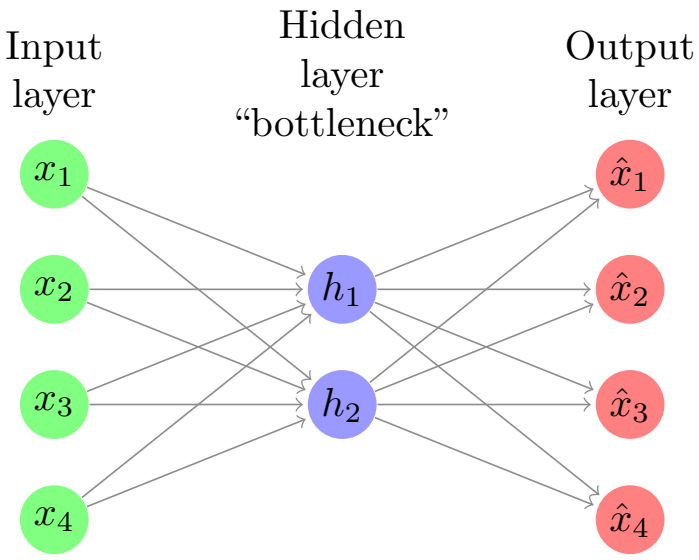


Figure 2.8: An autoencoder network.

dataset can be framed as a supervised learning problem using the output $\hat{\mathbf{X}}$, which is a reconstruction of the original input. This network can be trained by minimising the reconstruction error $\mathcal{L}(\mathbf{X}, \hat{\mathbf{X}})$ which measures the difference between the original input and the consequence reconstruction. The encoder function $\mathbf{h} = \mathbf{g}^{(1)}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$ and the decoder function that reproduces the output $\hat{\mathbf{x}} = \mathbf{g}^{(2)}(\mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)})$, where $\mathbf{W^{(1)}}$, $\mathbf{W^{(2)}}$, $\mathbf{b^{(1)}}$, and $\mathbf{b^{(2)}}$ are the learnable weights and biases, respectively. The loss function $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}})$ can be simply defined as:

$$\mathcal{L}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{\mathbf{N}} \parallel \mathbf{X} - \hat{\mathbf{X}} \parallel_2^2 \tag{2.7}$$

The bottleneck is the key attribute of the network design. Without the presence of an information bottleneck, the network can easily learn to memorise the input value. An auto-encoder with more hidden layers than the inputs runs the risk of learning an identity function. Thus, a further constraint is needed to separate useful information.

There are various types of AEs proposed to prevent the output layer directly copying the input data, e.g. Denoising Auto-Encoder (DAE) [103], Sparse Auto-encoder (SAE) [104], and Contractive Auto-encoder (CAE) [105]. In DAE [103], the input is randomly induced by noise while the last two explicitly impose penalties on the cost function. When the auto-encoder works well, the hidden layers contain most of the information from inputs and can be used as the input for classification instead of using the original input.

2.3.9 Generative Adversarial Network

The Generative Adversarial Network (GAN) was introduced in [42], targeting the minimax game between a discriminative network D and a generative network G. The generative network G takes the input z $\sim$ p($\mathbf{z}$) as a form of random noise, then outputs a new data G(z), whose distribution p_g is supposed to be close

to that of the data distribution p_{data}. The discriminative network D's task is to distinguish a generated sample G(z) $\sim p_g(\mathbf{G}(\mathbf{z}))$ and the ground truth data sample x $\sim p_{data}(\mathbf{x})$. In other words, the discriminative network determines whether the given images are natural-looking ones or artificially created ones. As the model is trained through alternative optimisation, both networks improve until reaching a point called Nash Equilibrium, where fake images are indistinguishable from the real images. This concept is consistent with the problem solved in image SR. The

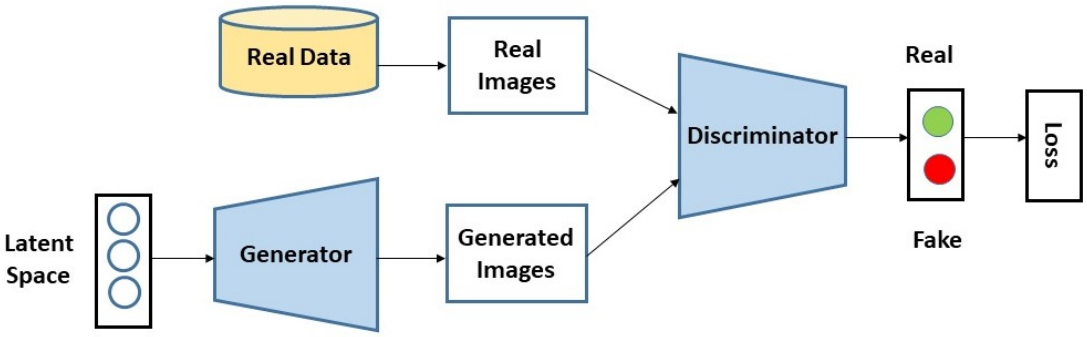


Figure 2.9: Generative Adversarial Networks.

Super-Resolution Generative Adversarial Network (SRGAN) model, in which a generative network upsamples LR images to super-resolution (SR) images, and the discriminative network is to distinguish between the ground truth HR images and the generated SR images. The pixel-by-pixel quality assessment metric has been criticised for performing poorly relative to human perception. As shown in Fig. 2.10, the distribution of reconstructed images using L_2 loss has one peak, which cannot match the multi-modal distribution of data. By combining an adversarial loss and L_2 loss, the GAN-based approach has resolved the problem by encouraging $p_g(\mathbf{G}(\mathbf{z})) = p_{data}$.

The GAN-based SISR model has been developed further, which has an improved SRGAN [78] by fusion of pixel-wise loss, perceptual loss, and texture transfer loss. The GAN-based model is to encourage reconstructed images to have a similar distribution as the ground truth images, which refer to adversarial

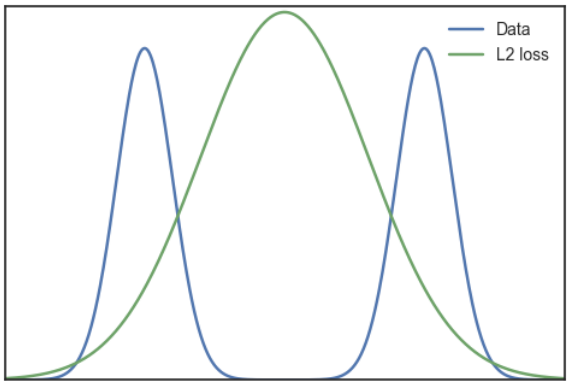


Figure 2.10: Distribution of image reconstruction using L_2 loss function cannot match distribution of real data.

loss as part of the perceptual loss in SRGAN. Adversarial learning is actually useful when faced with complicated manifold distributions in natural images.

2.3.10 Loss functions for super-resolution

A. Content loss

The classical content loss function for the regression problem are LAD (Least Absolutes Deviations) (or L_1) and LSE (Least Squared Errors) (or L_2) defined as follows:

$$L_1 = \sum_{i=1}^{N} |I^{HR} - I^{SR}| \tag{2.8}$$

$$L_2 = \sum_{i=1}^{N} (I^{HR} - I^{SR})^2 \tag{2.9}$$

where I^{HR} and I^{SR} are the ground-truth and reconstructed images, and the distance is calculated over all training examples. Using CNNs, the I^{SR} is the network's output of low-resolution input I^{LR}.

Note that the content loss will assume images are withdrawn from a uni-modal distribution with a single peak, which will not predict well for images from multi-modal distributions. Furthermore, a minor change in an image's pixels, i.e.,

shifting, can result in a significantly lower PSNR [106], even though both images appear identical to the human eye.

B. Adversarial loss

A key relationship between images and statistics is that images can be interpreted as samples from a high-dimensional probability distribution. The probability distribution goes over the image pixels and is what is used to define whether an image is natural or not. The adversarial loss measures the difference between two probability distributions, which is different from the Euclidean distance, i.e., L_1 and L_2 losses. Using GAN, the adversarial loss L_{adv} can be constructed based on the probability of the discriminator $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ given an input $G_{\theta_G}(I^{LR})$.

$$L_{adv} = \sum_{i=1}^{N} -logD_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{2.10}$$

where θ_D and θ_G are parameter of the discriminator D and the generator G, respectively. $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ is probability that reconstructed image $G_{\theta_G}(I^{LR})$ is natural image.

C. Feature loss

The feature space loss is calculated by comparing two images based on high-level representations from pre-trained Convolutional Neural Networks (trained on Image Classification tasks, i.e., the ImageNet dataset [107]). As shown in Fig. 2.11, the image is first trained by the Image Transform Net to produce an output, which is then fed to the pre-trained Loss Network. The feature loss can be defined by differences in ReLU activation between images.

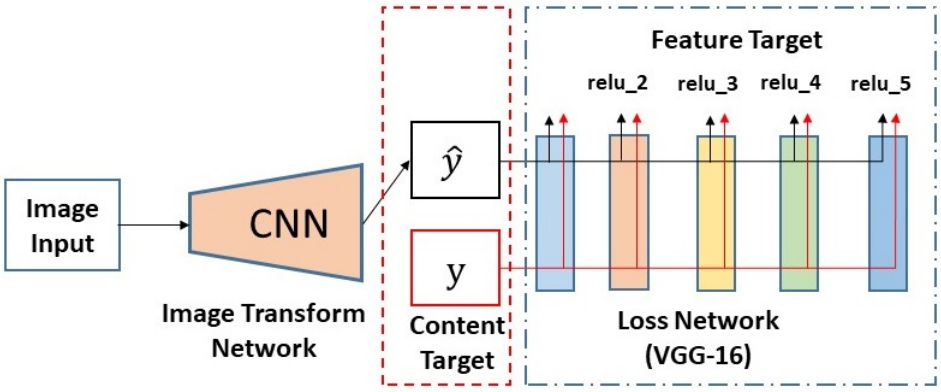


Figure 2.11: Model structure for calculating perceptual loss

2.3.11 Quantitative metrics for super-resolution

There are several quantitative metrics for image quality measurement. In this section, we present the most popular quantitative methods that will be employed in subsequent chapters. Note that, for the same metric, the calculation for RGB and HSI can be different. Let denotes $\mathbf{X} \in \mathbb{R}^{N_W \times N_H \times S}$ and $\hat{\mathbf{X}} \in \mathbb{R}^{N_W \times N_H \times S}$ are the ground-truth and the estimated images, respectively.

A. Quantitative metrics for single image super-resolution

Peak signal-to-noise ratio (PSNR) is the most widely used full-reference objective quality assessment metric for image restoration, which is defined as:

$$\mathbf{PSNR}(\mathbf{X}, \hat{\mathbf{X}}) = 10 \log_{10}(\frac{L^2}{\mathbf{MSE}}) \tag{2.11}$$

where $\mathbf{MSE} = \frac{1}{N_W \times N_H \times S} \parallel \mathbf{X} - \hat{\mathbf{X}} \parallel^2$ denotes the mean squared error between $\mathbf{X}$ and $\hat{\mathbf{X}}$, and L is the maximum pixel value of the image.

Structural SIMilarity (SSIM) index [108] is measured between two windows

x, y of the same size, i.e.,

$$\mathbf{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.12}$$

where μ and σ are the mean intensity and the standard deviation, respectively. The subscript x denotes reference and subscript y denotes the test image. C_1 and C_2 are two constants. The RGB image is assumed to be a grayscale image for calculating SSIM. The SSIM is used to compare the local patterns of pixel intensities between the two compared images and its values range between 0 and 1. The value of SSIM equal to 1 indicates that the reference and reconstructed images are identical.

B. Quantitative metrics for hyperspectral image super-resolution

Four quantitative picture quality indices (PQI) are utilised for performance evaluation, which include the Root Mean Square Error (RMSE), Structural SIMilarity (SSIM) index [108], Spectral Angle Mapping (SAM) [109] and the relative dimensionless global error in synthesis (ERGAS) [110].

The RMSE between the reconstructed and the original HSIs is defined as the average RMSE of all bands, e.g.,

$$\mathbf{RMSE}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{S}\sum_{i=1}^{S} RMSE(X^i, \hat{X}^i) \tag{2.13}$$

where X^i and $\hat{X}^i$ denote the ith band images of the ground-truth $\mathbf{X}$ and the estimated Hr-HSI $\hat{\mathbf{X}}$, respectively, and $RMSE(X^i, \hat{X}^i) = \sqrt{\frac{\sum_{j=1}^{N}\|X_j^i - \hat{X}_j^i\|_2}{N}}$ where $N = N_H \times N_W$. The RMSE is commonly used to compare the difference between two images by computing the variation in pixel values. The reconstructed image is close to the reference image when the RMSE value is near zero.

The structure similarity index measure is defined as the average value of all

bands, i.e.,

$$\mathbf{SSIM}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{S} \sum_{i=1}^{S} SSIM(X^i, \hat{X}^i) \tag{2.14}$$

where $SSIM(X^i, \hat{X}^i)$ is calculated by Eq. (2.12).

The Spectral Angle Mapping (SAM) is defined as an angle between the estimated pixel $\hat{x}_j$ and the ground truth pixel x_j over the whole image:

$$\mathbf{SAM}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{N} \sum_{j=1}^{N} arcos \frac{\hat{x}_j^T x_j}{\| \hat{x}_j \|_2 \| x_j \|_2} \tag{2.15}$$

The SAM is performed on a pixel-by-pixel base. A value of SAM equal to zero indicates no spectral distortion.

Finally, the ERGAS is defined as:

$$\mathbf{ERGAS}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{100}{d} \sqrt{\frac{1}{S} \sum_{i=1}^{S} \frac{MSE(\hat{X}^i, X^i)}{\mu_{\hat{X}^i}^2}} \tag{2.16}$$

where $\mu_{\hat{X}^i}$ is the mean of $\hat{X}^i$ and $MSE(\hat{X}^i, X^i)$ is the mean squared error between $\hat{X}^i$ and X^i, d is a spatial downsampling factor. The ERGAS is used to determine the image's quality in terms of the normalised average error of each band. A larger ERGAS indicates that the reconstructed image is distorted, whereas a smaller ERGAS means that the reconstructed image is more similar to the reference image.

C. Quantitative metrics for image generation

An in-depth overview of GAN evaluation measures is discussed in [111]. The two most widely used metrics to measure the quality of generated images are the Inception Score (IS) and the Frechet Inception Distance (FID).

Inception score [87] provides a method for quantitatively evaluating the quality of the generated samples. A large number of generated images are classi-

fied using the Inception v3 Network pre-trained on ImageNet. The probability of the image that belongs to each class is predicted and summarised for the inception score. Two properties of the generated image are reflected in the inception score: *image quality* and *image diversity*. Intuitively, the class label conditional on the generated image should have a low entropy, and the variety of generated images is expected to be high.

$$\begin{aligned} \textbf{IS} &= \exp\left(\mathbb{E}_{\mathbf{x}\sim p_g} D_{KL}(p(y|\mathbf{x}) \parallel p(y)\right) \\ &= \exp\left(H(y) - \mathbb{E}_{x\sim p_g}[\, H(y|\,\mathbf{x}\,)]\right) \end{aligned} \tag{2.17}$$

where $\mathbf{x}$ is an image sampled from p_g, $D_{KL}(p(y|\mathbf{x}) \parallel p(y))$ is the KL-divergence between the conditional class distribution and the marginal class distribution $p(y) = \int_{\mathbf{x}} p(y|\mathbf{x})\, p_g(\mathbf{x})$. $H(x)$ represents the entropy of variable x. The drawback of IS is that it does not consider the real image for measurement, which means one cannot interpret how well the generator approximates the real distribution. An in-depth review can be found in [112].

Frechet Inception Distance [113] is an alternative for determining the similarity of two images. Image samples from p_d and p_g are embedded into a feature space. Assuming that both the embedded data are followed a multivariate Gaussian distribution, the FID can be computed, i.e.

$$\textbf{FID} = \parallel \mu_x - \mu_y \parallel_2^2 + \mathrm{Tr}(\Sigma_x + \Sigma_y - 2(\Sigma_x\, \Sigma_y)^{\frac{1}{2}}) \tag{2.18}$$

where (μ_x, Σ_x) and (μ_y, Σ_y) are the mean and covariance of the embedded samples from p_d and p_g, respectively. The shortcoming of FID is that it employs a pre-train Inception model and limited statistics (i.e., mean and covariance), which may not be able to capture all the features. In addition, the FID measure requires a large sample size for high accuracy (i.e., a minimum of 10,000 samples), which can be computationally expensive for large images.

The high IS and low FID indicate that the generated samples are a realistic approximation of the distribution of natural images.

2.4 Network Architectures for single image super-resolution

The CNN-based methods use gradient descent training on a set of learnable parameters. Following that, the pre-trained network is utilised to predict the HR image from an input. The methods for CNN-based SR are always accompanied by up-sampling processes. Up-sampling can be classified into three types: interpolation, transposed convolution, and pixel shuffle. Up-sampling through interpolation is typically based on heuristics and does not incorporate learnable parameters.

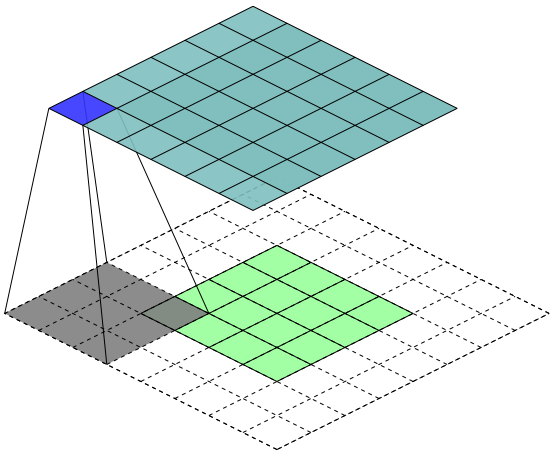


Figure 2.12: Transposed convolution of a 3 $\times$ 3 kernel over a 4 $\times$ 4 input with no padding and 1 $\times$ 1 strides results in an output of size 6 x 6.

Transposed convolution (see Section 2.3.3) [97] is the most widely used up-sampling method. It not only up-samples feature maps but also fills in details using the learnable parameters, as illustrated in Fig. 2.12. This method is more flexible than regular interpolation, as it can be trained jointly with convolutional

layers during the training process. One drawback of transposed convolution is that it may introduce checkerboard artifacts when the kernel size is not divisible by the stride [114]. To avoid such artifacts, pixel shuffle [115] is used as an efficient alternative for transposed convolution. Pixel shuffle implements efficient sub-pixel convolution, which first performs a standard convolution in a low-resolution space, and then follows by re-arranging a tensor of shape $(*, H, W, C \times r^2)$ to a tensor of shape $(*, H \times r, W \times r, C)$. Here, r is an upscale factor (see Fig. 2.13). The Pixel shuffle is a variant of transposed convolution. As shown in Fig. 2.4 (Section 2.3.3), the transposed convolution is equivalent to applying a smaller convolution on the input and then re-arranging the elements. The advantage of using pixel shuffle is twofold: first, its standard convolution works in a low-resolution space with a smaller spatial size, thereby reducing the computational cost; second, it avoids padding zero between pixels, which is one of the main causes of artifacts.

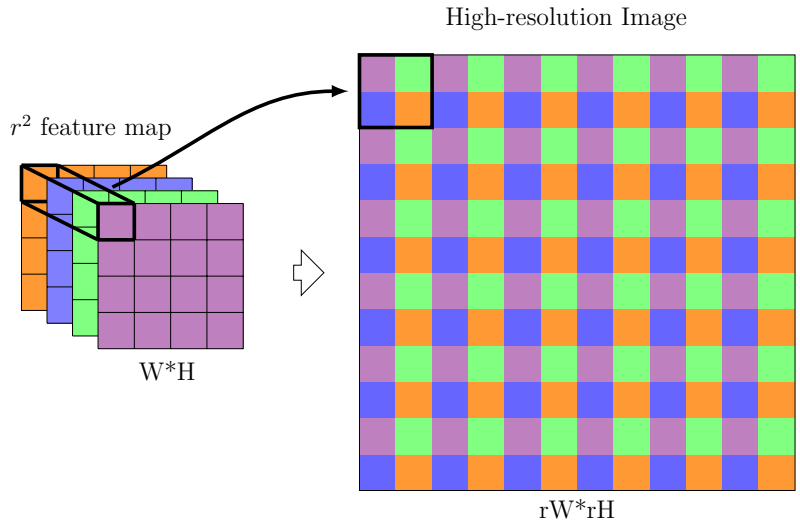


Figure 2.13: Re-arrange elements for up-sampling in pixel shuffle.

The CNN-based model for the SISR, namely the Super-Resolution Convolutional Neural Network (SRCNN), was first introduced in [11], which contains only three layers. Given a training set of LR and corresponding HR images x^i, y^i, i = 1...N, the objective is to find an optimal model f, which will then be applied to

accurately predict Y = f(X) on an unobserved example X. The SRCNN consists of the following 4 steps, as illustrated in Fig. 2.14.

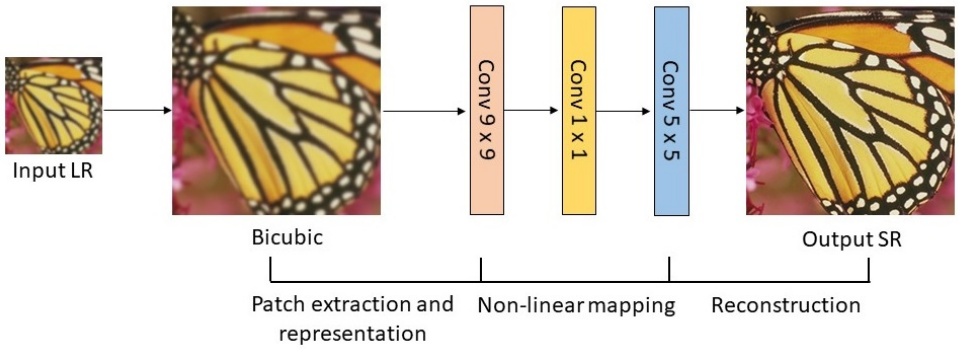


Figure 2.14: SRCNN model for SISR.

1. Preprocessing: Upscale the LR image to a desired HR image using the bicubic interpolation.

2. Patch extraction and representation: Extract a set of feature maps from the upscaled LR image.

3. Non-linear mapping: Maps the features between the LR and HR patches.

4. Reconstruction: Produce the HR image from the HR patches.

Going into deep networks, [45, 46] proposed Very Deep Convolutional Networks (VDSR) [45] and Deeply Recursive Convolutional Network (DRCN) [46] models, which both contained 20 convolutional layers. The VDSR is trained with a very high learning rate to speed up the training process, and gradient clipping is used to control the explosion problem. Instead of predicting the whole image, the VDSR used global residual connection to force the inside module to learn the difference between the input and the output. To further ease the training, DRRN employs both global and local residual connections. The use of residual learning in VDSR and DRRN has shown improvements in SISR. Since then, numerous CNN-based SISR architectures have been proposed. Some aspects that contribute to image reconstruction accuracy are discussed below.

Channel attention: Each of the learned filters operates with a local receptive field, and the interdependence between channels is entangled with spatial correlation. Therefore, the transformation output is unable to exploit information such as the interrelationship between channels outside the region. The Residual Channel Attention Network (RCAN) [116] has been the deepest model (about 400 layers) for the SISR task. It integrated a channel attention mechanism inside the residual block, as shown in Fig. 2.15. The input with the shape of H × W × C is squeezed into the channel descriptor by averaging through a spatial dimension of H × W to generate the output shape of 1 × 1 × C. This channel descriptor is put through a gate activation of the sigmoid f and an element-wise product with the input in order to control how much information from each channel is passed up to the next layer in the hierarchy.

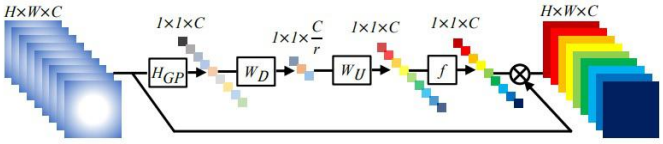


Figure 2.15: Channel attention block.

Feature concatenation: As the model goes deeper, the features in each layer are hierarchical, with different receptive fields. The information from each layer may not be fully used by recent residual learning methods. The Residual Dense Network (RDN) [117] proposed feature concatenation inspired by the DenseNet [118] to best use features from all layers, as shown in Fig. 2.16.

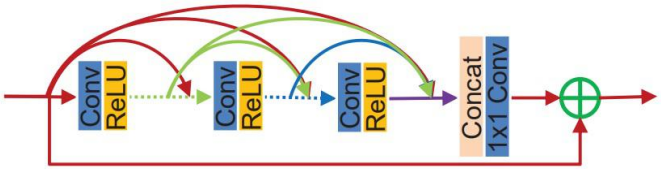


Figure 2.16: Residual dense block [117]. All features from previous layers are concatenated to build hierarchical features.

Wide activation in residual block: The efficiency and higher accuracy of SR model can be achieved [119] with fewer parameters by expanding the number of channels with a factor of $\sqrt{r}$ before RELU activation in residual blocks. As such, the residual identity mapping path slimmed as a factor of $\sqrt{r}$ to maintain constant output channels.

Feature cascading using 1 $\times$ 1 convolution: There are similar mechanisms in MemNet [120], RDN [117] and Cascading Residual Network (CARN) [121] models. In addition to the ResNet-based architecture, the 1 $\times$ 1 convolution layer is used as a fusion module to incorporate multiple features from previous layers, which has shown improved results.

Information Distillation Network (IDN): The IDN model [122] uses the distillation block, which combines an enhancement unit with a compression unit. In this block, the information is distilled inside the block before it passes to the next level.

RNN-CNN-based models: A ResNet with weight sharing can be interpreted as an unrolled single-state Recurrent Neural Network (RNN) [123]. A Dual-State Recurrent Network (DSRN) [124] allows both the LR path and the HR path to capture information in different spaces and connect at every step to contribute to the learning process. However, the average of all recovered SR images at each stage may have a deteriorated result. Furthermore, the down-sampling operation at every stage can lead to information loss at the final reconstruction layer.

In the view of memory in RNNs, CNNs can be interpreted as: *Short-term memory.* The conventional plain CNNs adopt a single-path feed-forward architecture, in which the latter feature is influenced by a previous state. *Limited long-term memory:* When the skip connection is introduced, one state is influenced by a previous state and a specific point in the prior state. To enable the latter state to see more prior states and decide whether the information should be

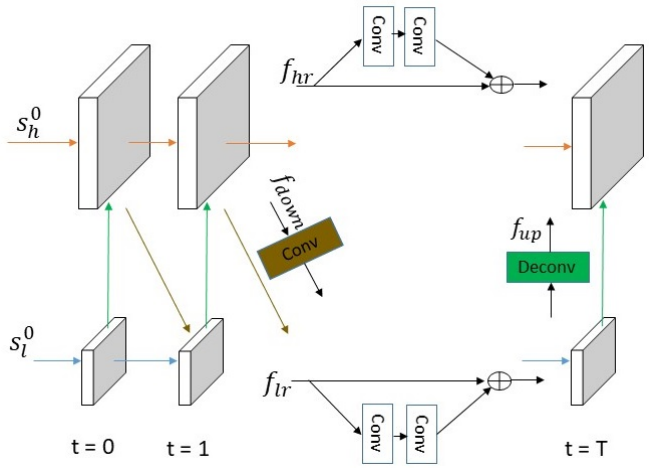


Figure 2.17: Dual State Model [124]. The top branch operates in the HR space, while the bottom branch works in the LR space. A connection from LR to HR using transposed convolution; a delayed feedback mechanism is to connect the previous predicted HR to LR at the next stage.

kept or discarded, Memory Network (MemNet) [120] uses recursive units followed by a memory unit to allow the combination of short and long-term memory, as illustrated in Fig. 2.18. In this model, a gate unit controls information from the prior recursive units, which extract features of different levels.

Non-local module: Different from the convolutional operation, which captures features by repeatedly processing local neighbourhoods of pixels, the non-local operation describes a pixel as a combination of weighted distance to all other pixels, regardless of their positional distance or channels. The convolutional operation can merely use the relevant local information, while the non-local operation can exploit the image self-similarity globally. However, the local and non-local

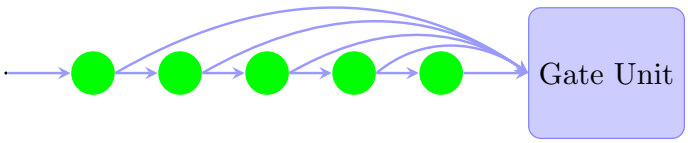


Figure 2.18: The memory block in MemNet [120] includes multiple Recursive Units(green circles) and a Gate Unit.

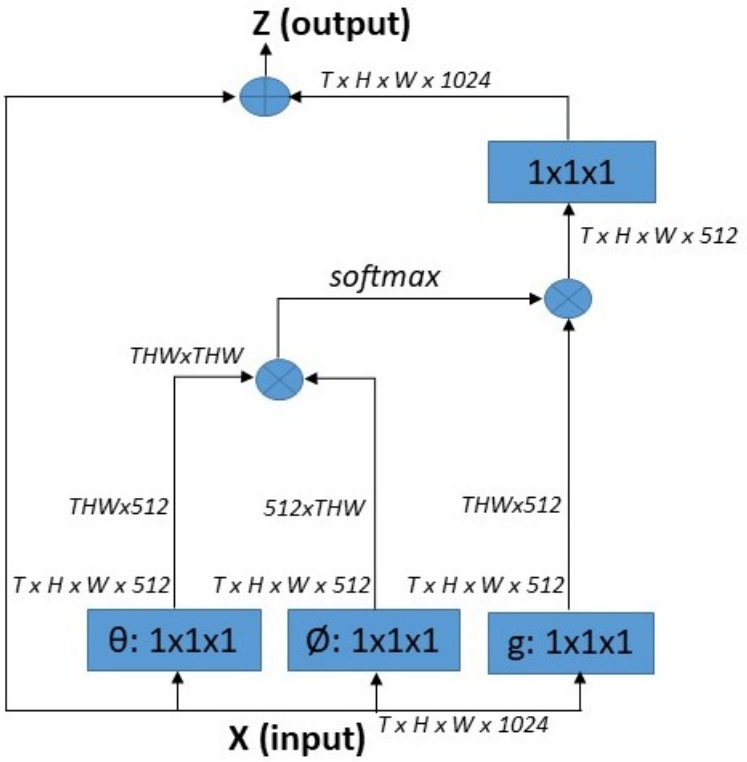


Figure 2.19: A non-local block.

based methods are treated separately, thereby not taking account of their advantages. The non-local block was first introduced in [125], where non-local operations were integrated into end-to-end training with local operation-based models such as CNNs. Each pixel at point i in an image can be described as:

$$y_i = \frac{1}{C(x)} \sum_{j \in \Omega} f(x_i, x_j) g(x_j) \tag{4}$$

where $f(x_i, x_j) = e^{\Theta(x_i)^T \varnothing(x_j)}$ is a weighted function, measuring how closely related the image at point i is to the image at point j. Thus, by choosing $\Theta(x_i) = W_\Theta x_i$, $\varnothing(x_j) = W_\varnothing x_j$ and $g(x_j) = W_g x_j$, the self-similarity can be jointly learned in embedding space by block shown in Fig. 2.19.

For a SISR task, Li et al. [126] incorporated non-local blocks into the RNN network and maintained two paths: a regular path that contains convolution

operations on an image and the other path that maintains non-local information at each step. Using non-local blocks has achieved a considerable performance gain. However, it consumes excessive computational resources when calculating the weighted distance between pixels.

Finally, Table 2.1 provides brief information of CNN-based SISR models, including SRCNN [11], VDSR [45], DRCN [46], DRRN [47], RED30 [127], RCAN [116], SRCliqueNet [128], RDN [117], CARN [121], IDN [122], LapSRN [48], EDSR [129], Zero Shot [80], and MemNet [120]. The brief performance comparison of those models is presented in Fig. 2.20. The four standard benchmark datasets are used, including Set5 [130], Set14 [131], BSD100 [132], Urban100 [31], which are popularly used for comparison of SR algorithms. The down-sampling scale factor is 4x, and the missing information that was not provided by the authors is marked by [-]. All quantitative results are duplicated from the original papers.

From Fig. 2.20, CARN stand out through their high accuracy using a small model. SRCliqueNet+ and RCAN+ have achieved higher accuracy in comparison with EDSR in term of PSNR/MMSI measurement whilst requiring a smaller model size. It is observed that all compared models perform best with Set5 [130] and worst with Urban100 [31] while CARN [121], RCAN [116], SRCliqueNet [128], RDN [117], and EDSR [129] among show better performance with Urban100 [31]. Set5 [130] is a quite small dataset of common image scenes. The Urban100 [31] is the most challenging because of its larger, more diverse dataset and is also the easiest to fool the super-resolution methods with its repeated features. The training DIV2K dataset contains images that cover some urban scenes; thus, the models that were trained on DIV2K [133] perform better with the testing dataset of Urban100 [31] than those that were not. The models that were trained on the BSD200 [132] perform better with the testing dataset of the BSD100 [132] as they are split from a larger dataset of 500 images.

Table 2.1: The comparison of different SISR models.

Models	Input	Type of network	No of params	Multadds	Reconstruct	Train data	Loss function
SRCNN [11]	LR + Bicubic	Supervised	8K	52.7G	Direct	Yang91 [134]	L2(MSE)
VDSR [45]	LR + Bicubic	Supervised	666K	612G	Direct	BSD200 [132], Yang91 [134]	L2
DRCN [46]	LR + Bicubic	Supervised	1,775K	17,974G	Direct	Yang91 [134]	L2
DRRN [47]	LR + Bicubic	Supervised	297K	6,796G	Direct	BSD200 [132], Yang91 [134]	L2
RED30 [127]	LR + Bicubic	Supervised	4,2M	-	Direct	BSD300 [132]	L2
MemNet [120]	LR + Bicubic	Supervised	677K	2,662G	Direct	BSD200 [132], Yang91 [134]	L2
LapSRN [48]	LR	Supervised	812K	29.9G	Progressive	BSD200 [132], Yang91 [134]	Charbonnie
Zero-Shot [80]	LR + Bicubic	Unsupervised	225K	-	Direct	-	L1(MAE)
EDSR [129]	LR	Supervised	43M	2890G	Direct	DIV2K [133], Flickr [129]	L1
IDN [122]	LR	Supervised	677K	-	Direct	BSD200 [132], Yang91 [134]	L1
CARN [121]	LR	Supervised	1,6M	222G	Direct	DIV2K [133], BSD200 [132], Yang91 [134]	L1
RDN [117]	LR	Supervised	22.6M	1300G	Direct	DIV2K [133]	L1
RCAN+ [116]	LR	Supervised	16M	-	Direct	DIV2K [133]	L1
SRCliqueNet+ [128]	LR	Supervised	-	-	Direct	DIV2K [133], Flickr [129]	L1 + L2

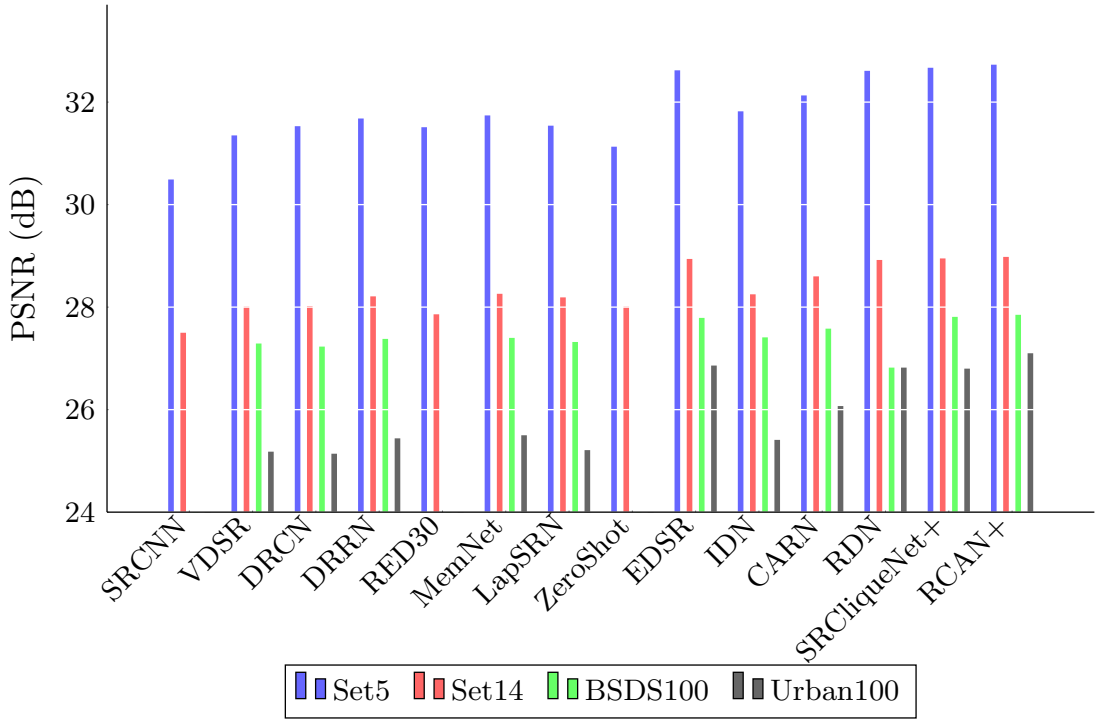


Figure 2.20: Comparing the PSNR accuracy of different algorithms on 4 Testing Datasets with factor of 4x.

2.5 Summary

This chapter describes the background and related work on image super-resolution, focusing specially on CNN-based SR methods. The concepts of super-resolution are introduced first, followed by a brief review of the literature on various areas of super-resolution, such as single image super-resolution, fusion-based image super-resolution, and generative adversarial network-based super-resolution. Compared to conventional methods, the CNN-based approaches can achieve superior performance thanks to the capacity of extracting high-level abstractions and mapping between LR and HR images. However, CNN-based models soon encounter diminishing returns, where increasing the network width or depth does not gain significant improvement. Therefore, to further improve the performance

of super-resolution requires developing specific architectures or methods that fit the desired task, i.e., single image super-resolution, fusion-based image super-resolution. Besides, even when the model seems to be optimally designed for the task, the unstable training is still present and degrades the performance, especially in GAN model. Addressing the instability issue would eventually lead to improved results.

Next, the technical background of Convolutional Neural Networks is presented to provide details of the components or subnetworks that are later employed in subsequent chapters. Finally, an overview of the network architectures in single image super-resolution, which can be widely applied to the CNN-based model of image super-resolution, is discussed. From the quantitative comparison of different architectures, it can be seen that architectural design has played an important role in the performance of a model.

Chapter 3

Single Image Super-Resolution

3.1 Introduction

Developing an architecture that learns to map from a low-resolution space to a high-resolution target image is inevitable for CNN-based SR. The depth of the network is of crucial importance not only for image recognition but also for image SR. From a shallow architecture [11] to deep ResNet-style SR networks [11, 45–47, 117, 118, 120, 121, 129], the learning-based networks have shown significant improvements over those using conventional methods. However, the performance reaches a point of diminishing returns when a network goes deeper. Although ResNet architecture [50] enables us to build an extremely deep architecture as of 1001 layers, the 28-layer ResNet with increased width [12] of filters has outperformed it significantly. Furthermore, [135] showed that the effective pathways are short in contrast to the total length of the network. The pathways that are just 10-34 layers deep provide the majority of the gradient in a residual network of 110 layers. This means that depth and width representations are both important for SR performance. Simply stacking residual blocks to build a deeper network can hardly yield better results. How to construct a relatively shallow SR network with more powerful representation than those using a ResNet-based

baseline remains to be explored.

On the other hand, learning-based methods optimally tune and deduce features for desired outcomes. Only underlying features that are relevant to the task are retained, while others are discarded. Accordingly, it is reasonable to expect that networks can have the ability to let relevant information go through and forget the ones that are not useful. A ResNet connection-based SR model, unfortunately, does not have that ability. The features are decided by whether they are being used yet, not how much. This inability to perform precise attention will hinder the representational power of the SR network.

To construct a compact model and practically resolve a precise attention ability that is missing from ResNet, a Highway Network for SISR (HNSR) is proposed. This model stacks the same topology with a new type of connection that is expected to improve the ability of attention. With the same target of mitigating gradient vanishing as skip connections, the highway connection design helps to stabilise the training and recover fine details of the lost high-frequencies.

The remaining parts of this chapter are organised as follows: Section 3.2 revisits the gradient regularisations and then the proposed approach is introduced. Section 3.3 presents the experimental results, including an ablation study and discussion. Some summary remarks are drawn in Section 3.4.

3.2 Proposed approach

3.2.1 Skip connection and Highway connection

Training deep learning networks can be challenging for several reasons, including the gradient vanishing and information morphs problems. Let x_n denote the network's input at layer n, constantly transforming at each layer $x_{n+1} = T(x_n)$ leads to information morphs, where it is difficult to exploit the best usable information in the past layers properly. Instead, both the Residual Networks (ResNet [50])

and the Highway Network [136] can be regarded as an application of LSTM, following the similar way of any state change: $x_{n+1} = x_n + \Delta x_{n+1}$. ResNet indeed does exactly that, which utilises extra identity connections to enhance information flow such that very deep neural networks can be effectively optimised. Such skip connections guarantee the direct propagation of signals among different layers, thereby avoiding gradient vanishing and also information morphs. Given the input x_n at layer n and a transformation $F_n(x_n, W_{fn})$, the output at layer $n+1$ is as follows:

$$x_{n+1} = x_n + F_n(x_n, W_{fn}). \tag{3.1}$$

where $F_n(x_n, W_{fn})$ is equivalent to Δx_{n+1}, the residual between x_n and x_{n+1}.

In practice, although the gradient vanishing has been solved, the subsequent change in distribution through the network can still lead to dying ReLU or exploding gradient problems. For example, if F function is as $Conv_1 - RELU - Conv_2$, and the incoming neurons to ReLU are entirely negative, the backpropagation gradients through ReLU will vanish, making $Conv_1$ difficult to learn. Over time, a large part of the network will be rendered unusable if such neurons are unable to recover from their negative state. In other words, the ReLU is always dying for those neurons. The converse of the range could lead to an exploding gradient. The ideal distribution of the input to ReLU should be symmetric with a zero mean. For that purpose, the batch normalisation [101] will normalise the distribution of layers before the ReLU activation, which can help to address both vanishing and exploding gradient problems. While batch normalisation enables faster and more stable training of deep networks, it can nevertheless be argued that batch normalisation loses scale information of images and reduces the range of activation. Therefore, removing batch normalisation would improve the performance in super-resolution [129].

The Highway Network [136] is another approach to solve the gradient vanish-

ing problem.

$$x_{n+1} = \sigma_n \odot x_n + (1 - \sigma_n) \odot \Delta x_{n+1} \tag{3.2}$$

where σ_n is a sigmoid function ($0 \leq \sigma_n \leq 1$) with trainable parameters, and $\odot$ is a Hadamard product or element-wise product. The best usable neurons in the past can be exploited by adaptively setting a particular σ_i to 1, avoiding gradient vanishing. By using such a highway connection in (3.2), the layer distribution hardly shifts to the extreme range in the network, since the output of a layer is always a convex combination of the input and the transformation. This property could allow the model to further increase the learning rate, speeding up the training whilst minimising gradient exploding or vanishing. A learning rate of 4e-4 is used with the baseline model but with different connections. The percentage of the positive responses induced by ReLU activation is measured. The lower the sk_p is, the more skewed the distribution is. It can also interpret $\frac{1}{sk_p}$ as a Coefficient of Variation (CV), which shows how much variance there is around the mean in the data. Assuming that x_i in $\boldsymbol{x}$ are independent but have the same mean and variance. The linear transform before the ReLU activation, named pre-activation $\boldsymbol{z} = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}$, will approach a normal distribution, according to the central limit theorem. The distribution is skew after the ReLU activation with $mode = 0, mean \geq 0$, and $standard\ variance \geq 0$. The Pearson's coefficient of skewness can be determined by:

$$sk_p = \frac{mean - mode}{standard\ variance} = \frac{mean}{standard\ variance} \tag{3.3}$$

It is well-known that network training converges faster if the average of each input variable over the training set has a zero mean [137]. From Fig. 3.1, it can be seen that most ReLU activations in the skip connection-based model have an extremely skewed distribution ($sp_k \approx 0.2$). This is caused by a significant number of 0 values outputted by ReLUs. In other words, the mean of the pre-activation $\boldsymbol{z}$

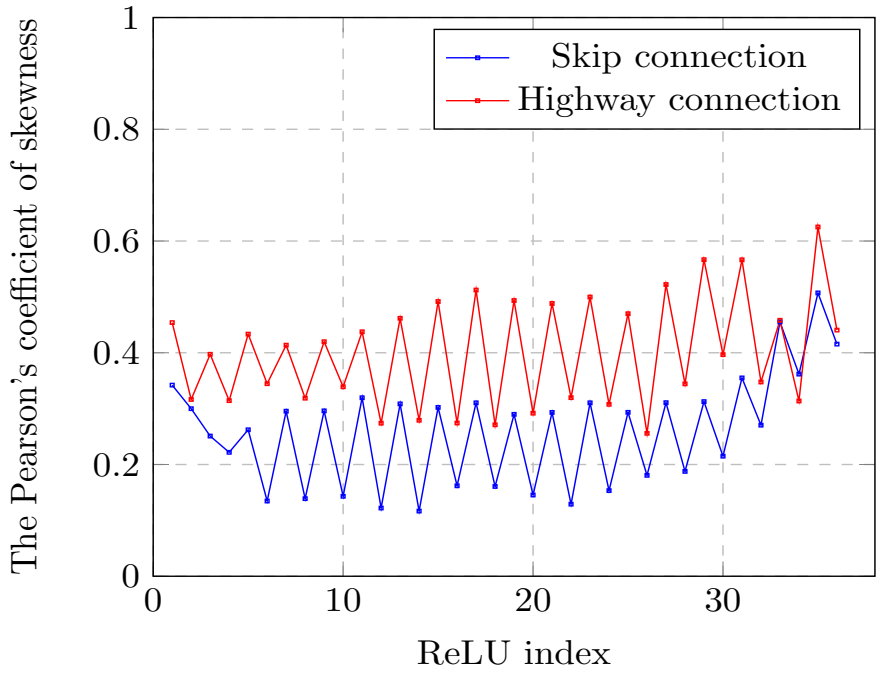


Figure 3.1: The Pearson's coefficient of skewness (sp_k) of ReLUs activation vs the network depth.

is more negative, therefore most values were zeroed out by the ReLU activation. A few last ReLU activations continuously aim to correct the bias shift introduced by previous ReLUs. At this learning rate, the model with skip connections can not learn effectively. The highway connection-based model, on the other hand, keeps all ReLU distributions in a stable range ($sp_k \approx 0.4$) with less correction. This advantage of highway connections can be explained by the convex combination giving an upper estimation of expectation of all Frobenius p-norms lower than that of skip connections.

$$\begin{aligned} \mathbb{E}[\|x_{n+1}\|_p] &\leq \mathbb{E}[\|c_n \odot x_n\|_p] + \mathbb{E}[\|(1-c_n) \odot h_n\|_p] \\ &< \mathbb{E}[\|x_n\|_p] + \mathbb{E}[\|h_n\|_p] \end{aligned} \tag{3.4}$$

where the first inequality holds by the Minkowski inequality and the second inequality holds since c_n is a sigmoid function ($0 \leq c_n, 1 - c_n \leq 1$).

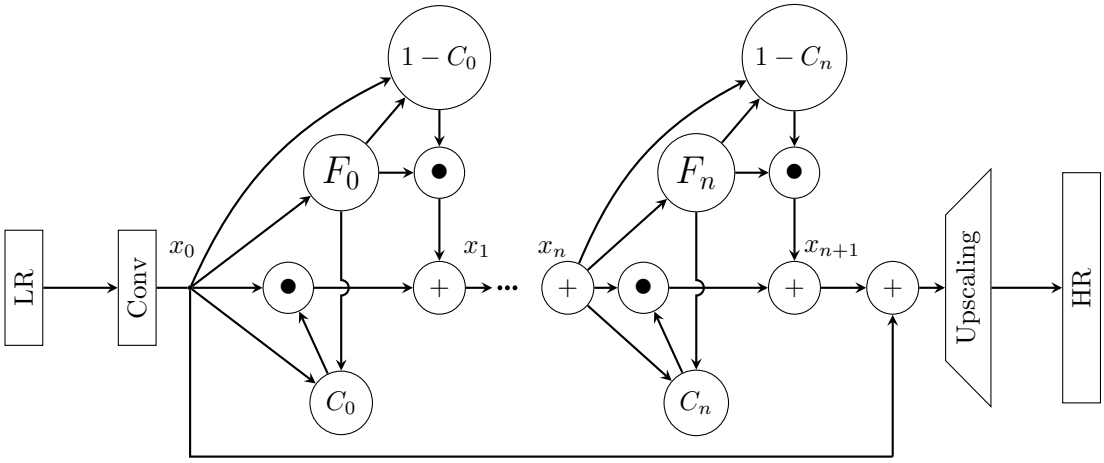


Figure 3.2: A proposed HNSR model.

3.2.2 Overall network structure

In this section, the proposed Highway Network for Super-Resolution (HNSR) is described, as shown in Fig. 3.2. A *carry* gate is inspired by the Gated Recurrent Unit (GRU [138]). The only global residual connection was used in the HNSR model. For each HNSR block, assume that x_n is the network's representation of the input x_0 at layer n. Let $h_n = F_n(x_n, W_{fn})$ be the intermediate transform function of the input x_n, $c_n = C_n([x_n, h_n], W_{cn})$, and $t_n = T_n([x_n, h_n], W_{tn})$ are the *carry* and the *transform* gates, typically utilise a sigmoid nonlinear function. The transform gate t_n is set to $1 - c_n$. The *carry* gate bias, b_{cn} as following is set to $+1$ at the start of training. Given the input x_n, the HNSR model is defined by:

$$h_n = F_n(W_{fn}x_n + b_{fn}). \tag{3.5}$$

$$c_n = C_n(W_{cn}[x_n, h_n] + b_{cn}). \tag{3.6}$$

$$x_{n+1} = c_n \odot x_n + (1 - c_n) \odot h_n. \tag{3.7}$$

where W denotes the trainable weights, and b is the trainable biases. The motivation behind this model is two-fold. First, we use highway connections to constrain

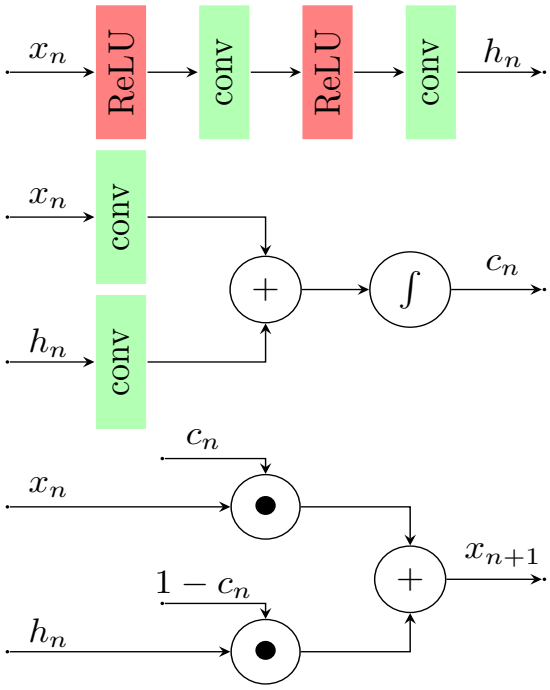


Figure 3.3: The structure of each HNSR block, transforming input x_n to output x_{n+1}.

the distribution of the output at the end of each block from going out of the optimal range for ReLUs activation, which helps to speed up the training. The significant change in the distribution will cause dying ReLUs, exploding gradient, or the ReLUs activation has to continuously correct the *bias shift*, which leads to slower training [139]. Second, an attention mechanism is formed that combines both the input x_n and the intermediate output h_n to enhance the discriminative learning ability. Since x_n and h_n have many features in common, combining them will help c_n to decide whether to disregard or retain features in x_n. This design is the difference between HNSR and the original Highway Network [136], where the latter blindly regulate information based on x_n only. The details of a HNSR block illustrating Eq. (3.5)-(3.7) are shown in Fig. 3.3. The transform function F_n in Eq. (3.5) is designed in the same way as a pre-activation Residual block [140], i.e. ReLU-Conv-ReLU-Conv. To form the attention or *carry* gate as in Fig. 3.3 (middle), the outputs of intermediary features h_n and input x_n are

convolved with filters of size 1×1 before being added and finally squashed by a sigmoid function. The 1×1 kernels will learn how much information from x_n and h_n being used. In Fig. 3.3 (bottom), the output, x_{n+1} is the result of a convex combination of x_n and h_n. It is worth noting that the c_n has the same dimension as the x_n and is pixel-wise multiplied with the x_n. A simple combination of the x_n and h_n will mitigate the gradient vanishing as it enables the backpropagation gradient to go through several routes. Meanwhile, the design of attention working on the pixel level helps to enhance the feature learning. In Fig. 3.2, the pixel shuffle operation, which was previously discussed in Section 2.4 is used for the up-sampling layer.

3.3 Experiments

3.3.1 Experiment settings

The Tensorflow framework is used to implement the proposed model. The HNSR model is evaluated by comparing the test accuracy with state-of-the-art SR architectures on the same dataset. The Nvidia GeForce GTX 1080 is utilised to conduct experiments.

3.3.2 Datasets

The 800 training images from the DIV2K dataset [133] were used for a training set. To fairly compare with other state-of-the-art methods, four benchmark datasets are used for testing: Set5 [130], Set14 [131], BSD100 [132], and Urban100 [31]. These four datasets are commonly used as benchmarks for evaluating algorithms in single super-resolution. They contain images that have abundant high-frequency components as well as low-frequency ones. In addition, those datasets also show the diversity of image scenes. For instance, BSD100 [132]

contains 100 images with key contents of animals, buildings, food, landscapes, people, and plants. Urban100 [31] is a collection of 100 images depicting urban scenes such as architecture, cities, structures, and urban. While BSD100 [132] intend to fail super-resolution approaches due to their low resolution images, the high resolution images in Urban100 [31] have repeated structure, making any prediction difficult. Although Set5 [130] and Set14 [131] are quite small datasets and the image patterns can be found in BSD100 [132], they were used for further validation of the super-resolution methods. Details of these datasets are given in Appendix A.

For training data, the three color channels (RGB) of the image are used [11] rather than transforming it into YCbCr and using only the luminance (Y) as in [141]. Compared to YCbCr, the RGB channels have a higher cross-correlation between them. When training using RGB, the PSNR result is up to 4 dB greater than when training with just the Y channel [11]. For each training mini-batch, 32 random LR patches and their corresponding HR patches are cropped as the input and the ground truth, respectively. To augment the training data, these training samples are randomly rotated by 90°, 180°, 270° and horizontally flipped. The input is normalised by subtracting the mean from each pixel and then dividing the result by the standard deviation. Noisy patches are also detected and removed from the training dataset. The commonly used mean squared error (MSE) is used as the loss function. Given a training set $\{I^i_{LR}, I^i_{HR}\}^N_{i=1}$, which contains N pairs of LR inputs and their HR counterparts, the goal of the training is to minimise the following loss function:

$$\mathcal{L} = \sum_{i=1}^{N} \| f_\theta(I^i_{LR}) - I^i_{HR} \|^2. \tag{3.8}$$

where f_θ presents a neural network and its parameter θ.

3.3.3 Hyperparameters

The batch size is set to 32, and the initial learning rate is 4e-4. A small batch size may lead to slower convergence of the learning algorithm than one using a larger batch size. In contrast, large batch sizes can help the learning algorithm to converge faster but may cause bad generalisation due to overfitting. The generalisation of small batch sizes is due to the fact that a small batch size can have a regularisation effect due to its high variance. Furthermore, the use of small batch size also requires a significant smaller Graphics Processing Unit (GPU) memory. To tune the learning rate, we choose the initial learning rate at 1e-3 and reduce it by a small amount if unstable training is observed. In training, the learning rate is decreased if the validation loss does not decrease after two additional epochs. The training process will be stopped when the loss ceases to reduce after three successive decreases in the learning rate. Only the checkpoint of the best validation accuracy is used to evaluate the test accuracy. The Adam optimisation [142] with default parameters is utilised for training.

3.3.4 Network depth

Due to the limitation of the GPU's capacity, the network uses 18 blocks for all experiments. We tune the number of blocks by monitoring the relative training error and validation error curves. Starting from a shallow network with a few blocks, when both training loss and validation loss are high, which refers to an underfitting case, we increase the depth of the network until we reach the hardware capacity. To make a fair comparison, we would like to compare the proposed model with similar small-size models.

Table 3.1: Average PSNR/SSIMs for scale 2x, 3x and 4x. Red color indicates the best, blue color indicates the second best performance, and missing information that was not provided by the authors is marked by [-/-].

Model	Set5 PSNR/SSIM	Set14 PSNR/SSIM
Scale = 2		
SRCNN [11]	36.66±0.48/0.954±0.001	32.42±0.54/0.906±0.002
FSRCNN [143]	37.00±0.35/0.955±0.0002	32.63±0.42/0.908±0.0002
VDSR [45]	37.53±0.25/0.958±0.0004	33.03±0.27/0.912±0.0002
DRCN [46]	37.63±0.26/0.958±0.0004	33.04±0.19/0.911±0.0003
LapSRN [48]	37.52±0.19/0.959±0.0005	33.08±0.20/0.913±0.0008
DRRN [47]	37.74±0.12/0.959±0.0006	33.23±0.13/0.913±0.001
MemNet [120]	37.78±0.13/0.959±0.007	33.28±0.14/0.914±0.002
SelNet [144]	37.89±0.04/0.959±0.004	33.61±0.11/0.916±0.005
IDN [122]	37.83±0.05/0.960±0.0004	33.30±0.08/0.914±0.004
CARN [121]	37.76±0.08/0.959±0.0007	33.52±0.10/0.916±0.004
HNSR	37.89±0.04/0.960±0.0003	33.33±0.07/0.915±0.003
Scale = 3		
SRCNN [11]	32.75±0.56/0.909±0.009	29.28±0.34/0.821±0.008
FSRCNN [143]	33.16±0.43/0.914±0.005	29.43±0.27/0.824±0.006
VDSR [45]	33.66±0.21/0.921±0.003	29.77±0.20/0.831±0.003
DRCN [46]	33.82±0.16/0.922±0.002	29.76±0.18/0.831±0.003
DRRN [47]	34.03±0.14/0.924±0.001	29.96±0.17/0.835±0.003
MemNet [120]	34.09±0.11/0.925±0.0009	30.00±0.14/0.835±0.004
SelNet [144]	34.27±0.07/0.925±0.002	30.30±0.19/0.840±0.004
IDN [122]	34.11±0.09/0.925±0.001	29.99±0.14/0.835±0.002
CARN [121]	34.29±0.05/0.925±0.001	30.29±0.12/0.840±0.002
HNSR	34.27±0.06/0.926±0.001	30.06±0.11/0.839±0.002
Scale = 4		
SRCNN [11]	30.48±0.78/0.862±0.016	27.49±0.53/0.750±0.007
FSRCNN [143]	30.71±0.4/0.865±0.012	27.59±0.34/0.753±0.009
VDSR [45]	31.35±0.24/0.883±0.006	28.01±0.26/0.767±0.005
DRCN [46]	31.53±0.21/0.885±0.004	28.02±0.24/0.767±0.006
LapSRN [48]	31.54±0.19/0.885±0.004	28.19±0.2/0.772±0.003
DRRN [47]	31.68±0.15/0.888±0.005	28.21±0.18/0.772±0.004
MemNet [120]	31.74±0.13/0.889±0.004	28.26±0.15/0.772±0.005
SelNet [144]	32.00±0.19/0.893±0.003	28.49±0.22/0.778±0.003
IDN [122]	31.82±0.21/0.89±0.003	28.25±0.16/0.773±0.0025
CARN [121]	32.13±0.18/0.893±0.004	28.60±0.20/0.780±0.003
HNSR	31.98±0.17/0.892±0.003	28.34±0.18/0.777±0.002

(a) Quantitative results from Set5 and Set14 testing datasets.

Model	BSD100 PSNR/SSIM	Urban100 PSNR/SSIM
Scale = 2		
SRCNN [11]	31.36±0.15/0.887±0.003	29.50±0.21/0.894±0.007
FSRCNN [143]	31.53±0.16/0.892±0.004	29.88±0.18/0.902±0.009
VDSR [45]	31.90±0.11/0.896±0.004	30.76±0.13/0.914±0.004
DRCN [46]	31.85±0.09/0.894±0.003	30.75±0.13/0.913±0.004
LapSRN [48]	31.80±0.12/0.895±0.006	30.41±0.16/0.910±0.004
DRRN [47]	32.05±0.05/0.897±0.003	31.23±0.13/0.918±0.006
MemNet [120]	32.08±0.05/0.897±0.004	31.31±0.09/0.919±0.005
SelNet [144]	32.08±0.06/0.898±0.04	-/-
IDN [122]	32.08±0.04/0.898±0.004	31.27±0.15/0.919±0.006
CARN [121]	32.09±0.02/0.897±0.004	31.92±0.06/0.925±0.003
HNSR	32.13±0.03/0.899±0.02	31.49±0.09/0.932±0.003
Scale = 3		
SRCNN [11]	28.41±0.19/0.786±0.01	26.24±0.45/0.799±0.02
FSRCNN [143]	28.53±0.26/0.791±0.04	26.43±0.31/0.808±0.007
VDSR [45]	28.82±0.16/0.797±0.008	27.14±0.26/0.828±0.006
DRCN [46]	28.80±0.10/0.796±0.006	27.15±0.24/0.827±0.005
DRRN [47]	28.95±0.08/0.8±0.005	27.53±0.20/0.837±0.005
MemNet [120]	28.96±0.07/0.8±0.005	27.56±0.18/0.837±0.004
SelNet [144]	28.97±0.06/0.802±0.004	-/-
IDN [122]	28.95±0.08/0.801±0.002	27.42±0.24/0.836±0.004
CARN [121]	29.06±0.006/0.803±0.003	28.06±0.10/0.849±0.003
HNSR	29.04±0.005/0.805±0.002	28.04±0.11/0.851±0.003
Scale = 4		
SRCNN [11]	26.90±0.36/0.71±0.009	24.52±0.42/0.722±0.01
FSRCNN [143]	26.98±0.38/0.715±0.006	24.62±0.40/0.728±0.013
VDSR [45]	27.29±0.11/0.725±0.008	25.18±0.16/0.752±0.008
DRCN [46]	27.23±0.14/0.723±0.007	25.14±0.18/0.751±0.007
LapSRN [48]	27.32±0.12/0.728±0.005	25.21±0.19/0.756±0.006
DRRN [47]	27.38±0.13/0.728±0.005	25.44±0.16/0.763±0.008
MemNet [120]	27.40±0.14/0.728±0.004	25.50±0.18/0.763±0.008
SelNet [144]	27.44±0.14/0.732±0.003	-/-
IDN [122]	27.41±0.12/0.729±0.002	25.41±0.26/0.763±0.007
CARN [121]	27.58±0.10/0.735±0.002	26.07±0.14/0.783±0.005
HNSR	27.53±0.13/0.736±0.001	25.97±0.14/0.784±0.004

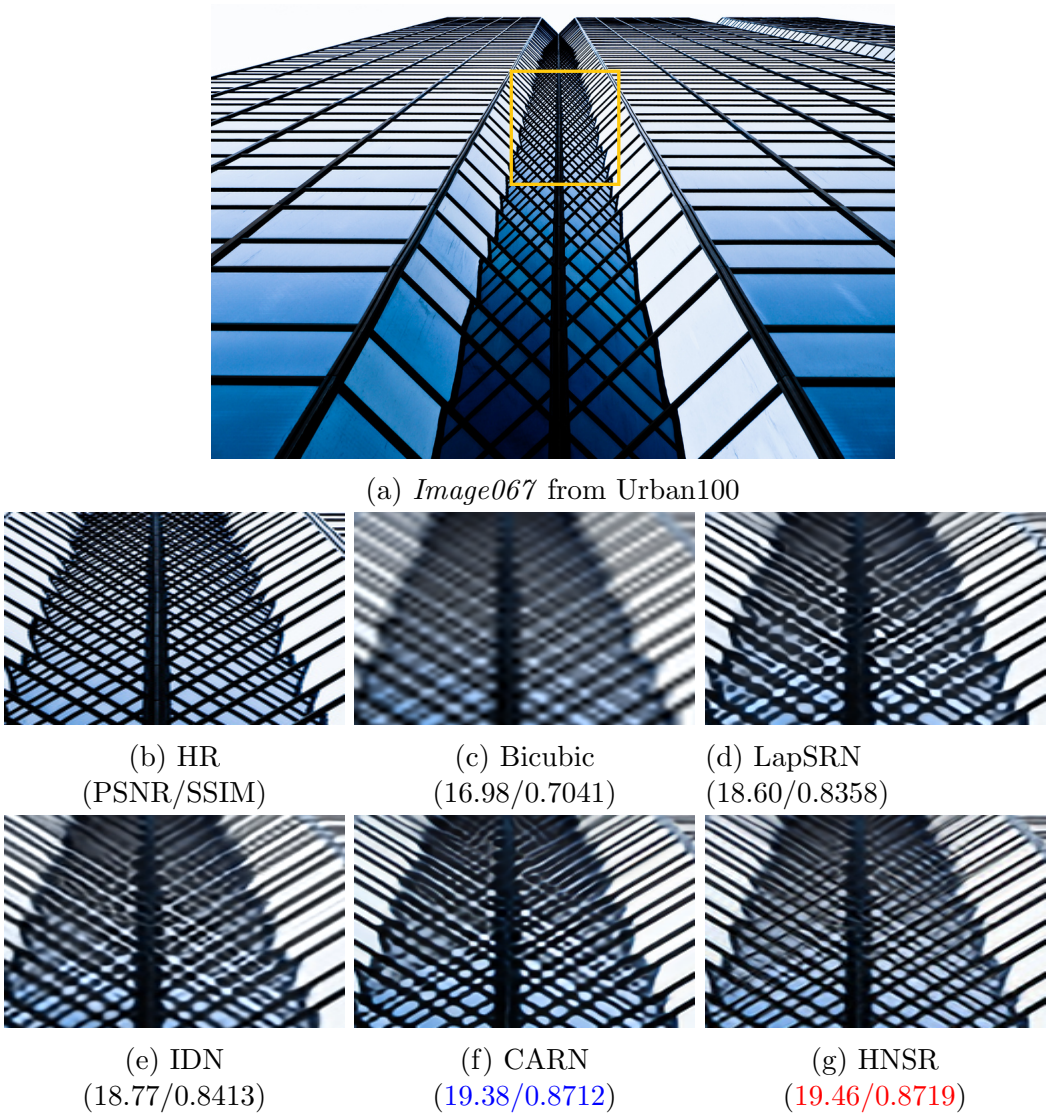
(b) Quantitative results from BSD100 and Urban100 testing datasets.

3.3.5 Results

A. Benchmark results

Table 3.1 compares the proposed method with bicubic interpolation and several state-of-the-art SR methods, including SRCNN [11], FSRCNN [143], VDSR [45], DRCN [46], LapSRN [48], DRRN [47], Memnet [120], SelNet [144], and CARN [121], which are considered as small-size models for a fair comparison. Following [11], the performance is evaluated with PSNR and SSIM [108] (presented in Subsection 2.3.11 (A)) on the Y channel (luminance) after transforming the images to the YCbCr space. As seen from Table 3.1, none of these methods can consistently outperform the others. Using large testing datasets, such as the Urban 100 and the BSD100, the proposed model and CARN both achieved the best performance in terms of PSNR and SSIM. Our PSNRs are slightly lower than those of CARN [121] by 0.58%, 0.11%, and 0.29% for scales of x2, x3, and x4, respectively, but our SSIMs consistently outperform those of CARN by 0.44%, 0.21%, and 0.05%, respectively. The CARN [121] is a ResNet-based model that implements a global skip connection and multiple-level local shortcut or skip connections. This strategy was previously employed in DRCN [46], DRRN [47], Memnet [120], and SelNet [144] to provide multiple paths for gradients flowing between layers. CARN's competitive result is likely due to the large training dataset combining DIV2K [133], Yang91 [134], and BSD200 [132] (see Table 2.1). Furthermore, the ResNet-based model, like CARN, shows ensemble-like behaviour. The final outcome will be a combination of multiple predictions, which provide a certain level of generality. However, the output image would be less sharp due to the averaged results. As expected, our model performs better than others in terms of SSIM. The structure of images is captured more easily with attention mechanisms.

B. Visual performance on test images



(a) *Image067* from Urban100

(b) HR (PSNR/SSIM)

(c) Bicubic (16.98/0.7041)

(d) LapSRN (18.60/0.8358)

(e) IDN (18.77/0.8413)

(f) CARN (19.38/0.8712)

(g) HNSR (19.46/0.8719)

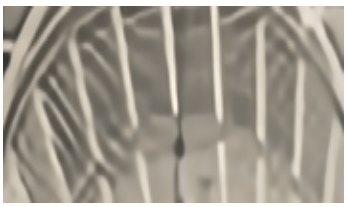
Figure 3.4: Visual qualitative comparison on the *Image067*, Urban100 dataset, magnified by a factor of 4.

As seen from Fig. 3.4, for *image_067*, most of the compared methods produce blurring artifacts along the diagonal lines, while the HNSR produces more sharp, faithful details. To make a fair comparison, the image *image_083* is tested, which achieved a slightly lower PSNR/SSIM, to demonstrate the detailed reconstruction ability. As seen from Fig. 3.5, most of the compared methods are incapable of

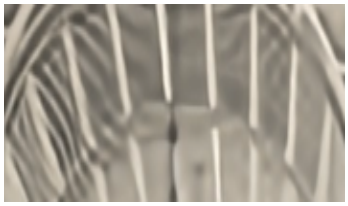
(a) *Image083* from Urban100

(b) HR
(PSNR/SSIM)

(c) Bicubic
(20.51/0.5578)



(d) LapSRN
(21.90/0.6767)

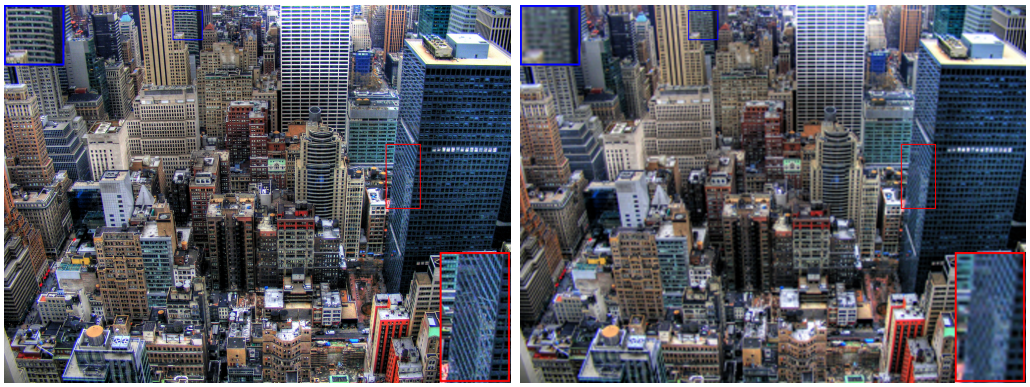


(e) IDN
(21.81/0.6740)

(f) CARN
(22.28/0.7013)

(g) HNSR
(22.22/0.7023)

Figure 3.5: Visual qualitative comparison on the *Image083*, Urban100 dataset, magnified by a factor of 4.



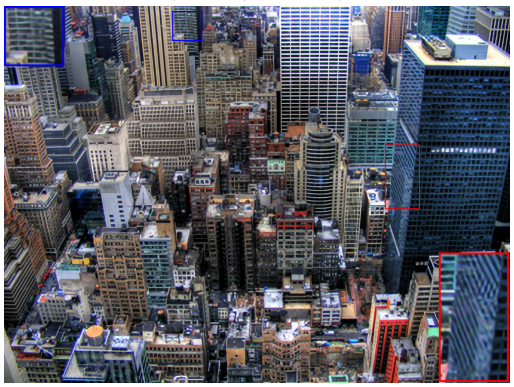
(a) HR
(PSNR/SSIM)

(b) Bicubic
(19.66/0.6127)

(c) LapSRN
(20.82/0.7120)

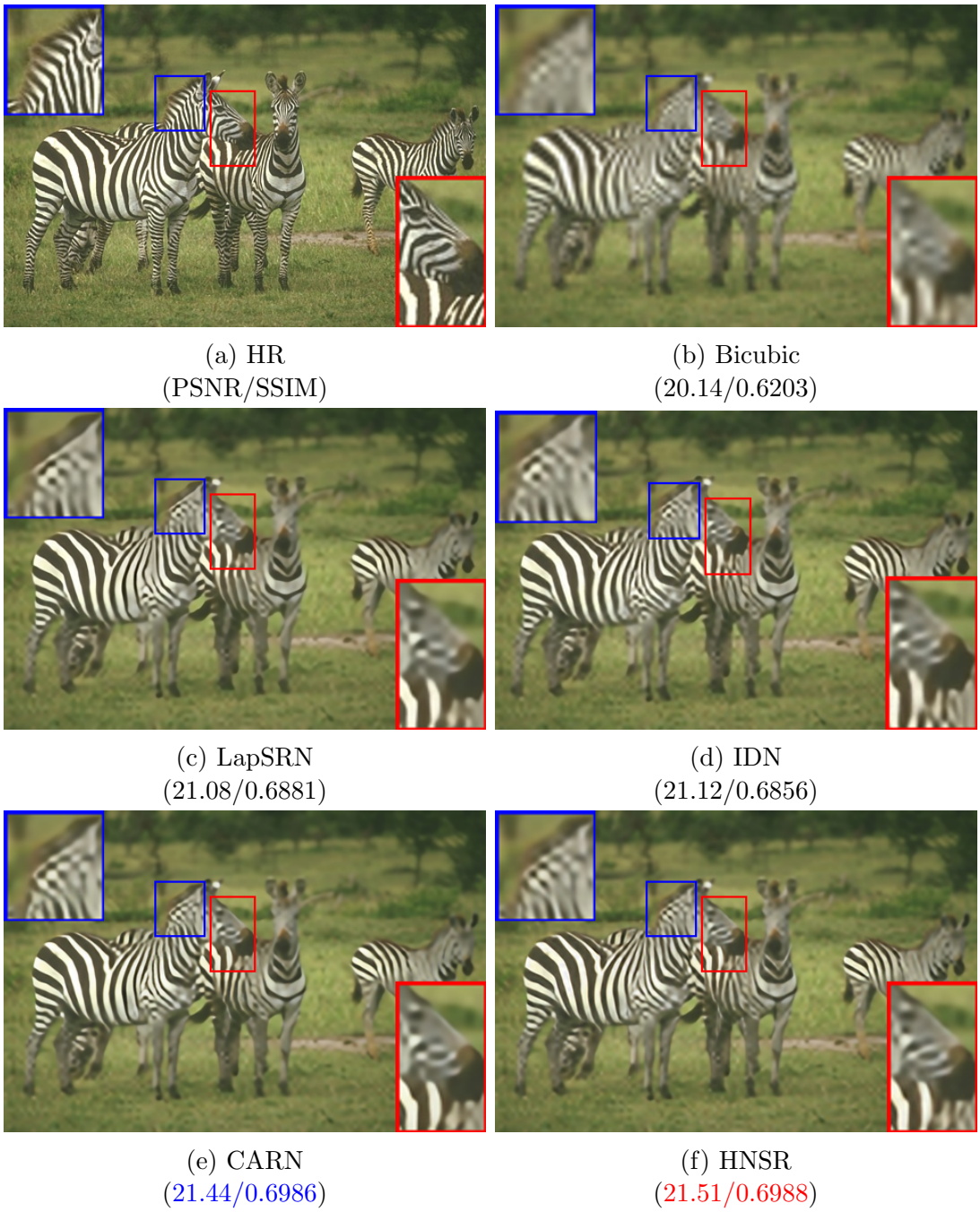
(d) IDN
(20.67/0.7046)

(e) CARN
(21.28/0.7394)



(f) HNSR
(21.38/0.7399)

Figure 3.6: Visual qualitative comparison on the *Image073*, Urban100 dataset, magnified by a factor of 3. All compared methods generate the wrong direction for the right diagonal lines except the HNSR method.



(a) HR (PSNR/SSIM)

(b) Bicubic (20.14/0.6203)

(c) LapSRN (21.08/0.6881)

(d) IDN (21.12/0.6856)

(e) CARN (21.44/0.6986)

(f) HNSR (21.51/0.6988)

Figure 3.7: Visual qualitative comparison on the *Image052*, BSD100 dataset, magnified by a factor of 4.

recovering the lattices and suffer from blurring artifacts. In contrast, the HNSR can alleviate the blurring artifacts better and recover more details. It can be

(a) HR (PSNR/SSIM)

(b) Bicubic (25.56/0.6483)

(c) LapSRN (26.77/0.7014)

(d) IDN (26.76/0.7015)

(e) CARN (26.86/0.7048)

(f) HNSR (26.87/0.7058)

Figure 3.8: Visual qualitative comparison on the *Image038*, BSD100 dataset, magnified by a factor of 4. Only HNSR can produce the forehead and legs of horses that are closest to the ground truth.

explained by the help of the gating mechanism, to forget irrelevant parts instead of remembering all features from the previous layer. Such obvious comparisons

demonstrate that the gating mechanism provides a more powerful representational ability to extract sophisticated features from the LR images. The more visual and quantitative comparisons are given in Fig. 3.6, Fig. 3.7, and Fig. 3.8. It is noticed that although CARN achieves high PSNR scores, the visual images are much more similar to those of the skip connection-used LapSRN and IDN methods, which show poor perceptual quality as reflected by their SSIM scores.

C. Ablation study

1) Comparison with skip connections

Since the learning rate of 4e-4 is not optimal for the skip connection-based model as discussed previously, a learning rate of 1e-4 is chosen for a fair comparison. As can be seen from Fig. 3.9, the highway-based model shows marginally faster convergence than the skip-based model right from the beginning, achieving a 27 dB accuracy at a step of 13.6K compared with 23.4K in the model with skip connections. At the step of 604K, the skip-based model is unable to improve further, while highway-based methods continue to learn more before stopping at the step of 676K. This observation is compatible with the results in Fig. 3.10, where we evaluate the loss of training and validation at the end of each epoch. As can be seen from Fig. 3.10, the highway-based models outperform those using skip-connections in both training and validation evaluation. Increasing the learning rate to 4e-4 enables the HNSR model to converge faster and to a better solution. In contrast, the performance of the skip connection-based model at that learning rate is worst, which verifies the previous observation in Fig. 3.1. With the same initialization, the HNSR takes the benefit of the convex combination property in highway connection to provide more stable training, thereby making the training faster. Meanwhile, the proposed attention mechanism enhances the discriminative learning ability and achieves better convergence.

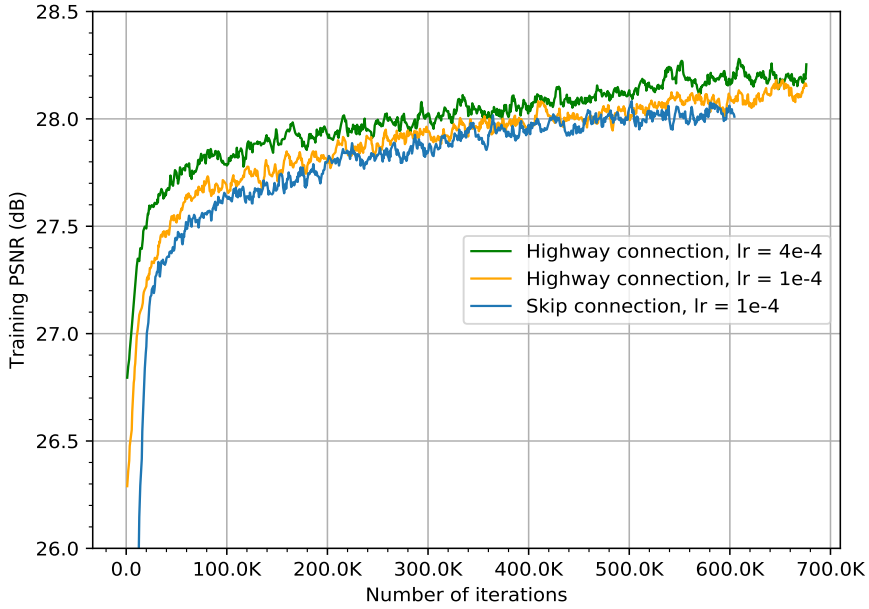


Figure 3.9: Training PSNR of the model with different types of connections. All parameters were initialised with the same seed values.

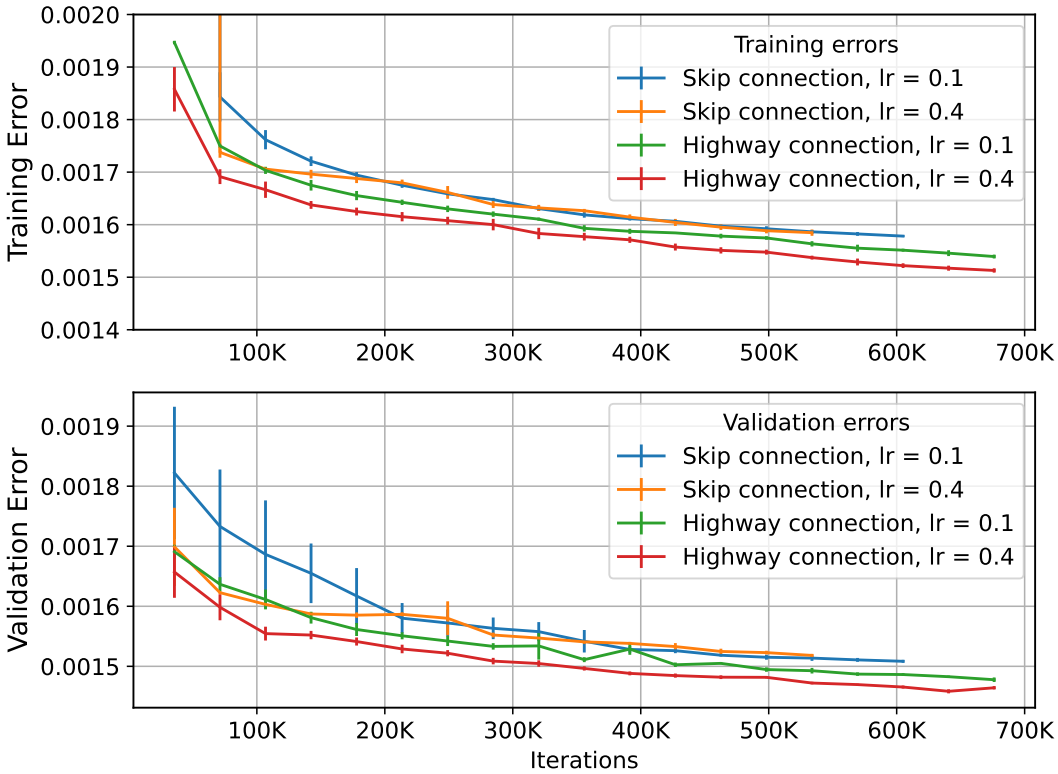
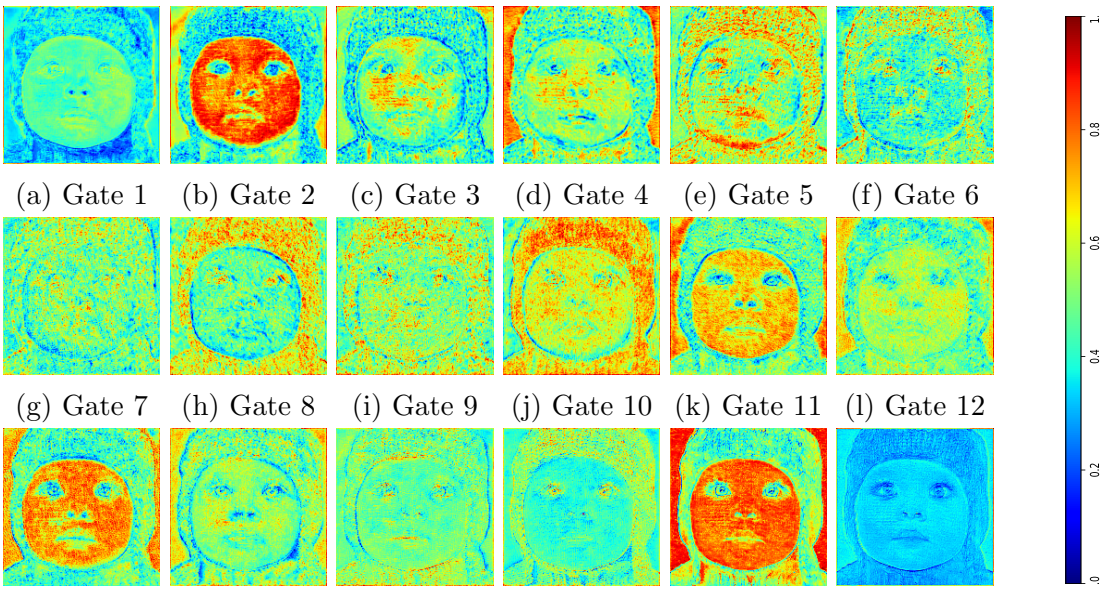


Figure 3.10: Training error and the corresponding validation error of models with different connections and learning rates.

2) Visual gate unit



(a) Gate 1 (b) Gate 2 (c) Gate 3 (d) Gate 4 (e) Gate 5 (f) Gate 6

(g) Gate 7 (h) Gate 8 (i) Gate 9 (j) Gate 10 (k) Gate 11 (l) Gate 12

(m) Gate 13 (n) Gate 14 (o) Gate 15 (p) Gate 16 (q) Gate 17 (r) Gate 18

Figure 3.11: The attention of 18 *carry* gates in 18 HNSR blocks on the image of a "baby" in Set5. The colormap from 0 to 1 shows the increasing level of attention on a particular area on the image.

To examine the attention of gates, the features through the *carry* gate, which is a sigmoid function are extracted. The feature maps of all the channel dimensions are averaged, then normalised to a range from 0 to 1, before associating with a heat map. Note that the gates regulate the information; they do not capture features to feed to the next layer. As seen in Fig. 3.11, the carry gates do not assign the same priority to a different position but give distinct attention to some specific regions. Because the CNNs have complicated interaction, it is difficult to explain why each gate gives particular attention. However, it can be seen that the focus on on one specific area is reducing or emphasising on the successor layer. Finally, the *carry* gate 18 achieves a balance for focused intensity.

3) Mitigating gradient vanishing and paying attention at the same time in each block

The following signs will help you identify the vanishing gradient problem: (1)

there are large changes in parameters of later layers, whereas the parameters of earlier layers change slightly or stay unchanged; (2) the model learns at a slow pace. It may obviously observe the vanishing gradient by replacing the ReLU activations with the sigmoid activations. The sigmoid function saturates at two end points where gradient or derivative values are significantly small. By comparing the gradient magnitude of the last layers to those of the earlier ones, the gradient issue can be detected. Unlike sigmoid activation, the ReLU activation does not saturate and therefore helps mitigate the problem. As the ReLU is now commonly used in CNN, it is better to identify the issue with the two

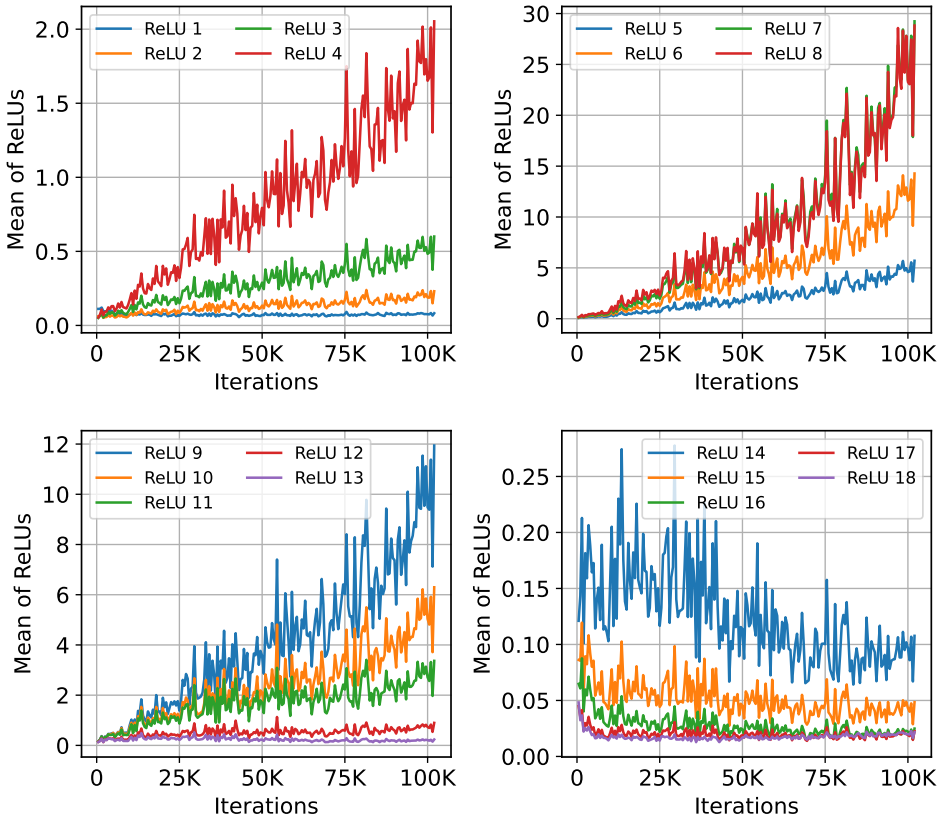


Figure 3.12: Mean of ReLU activations in a baseline network which does not contain any connection.

above criteria. The deep network still faces gradient issues, even with ReLU being used. To demonstrate the case of vanishing gradient in training CNN and how the highway connection can mitigate this problem, the experiments are conducted with a base network where all connections are not present and a highway network where a local highway connection is introduced in every block.

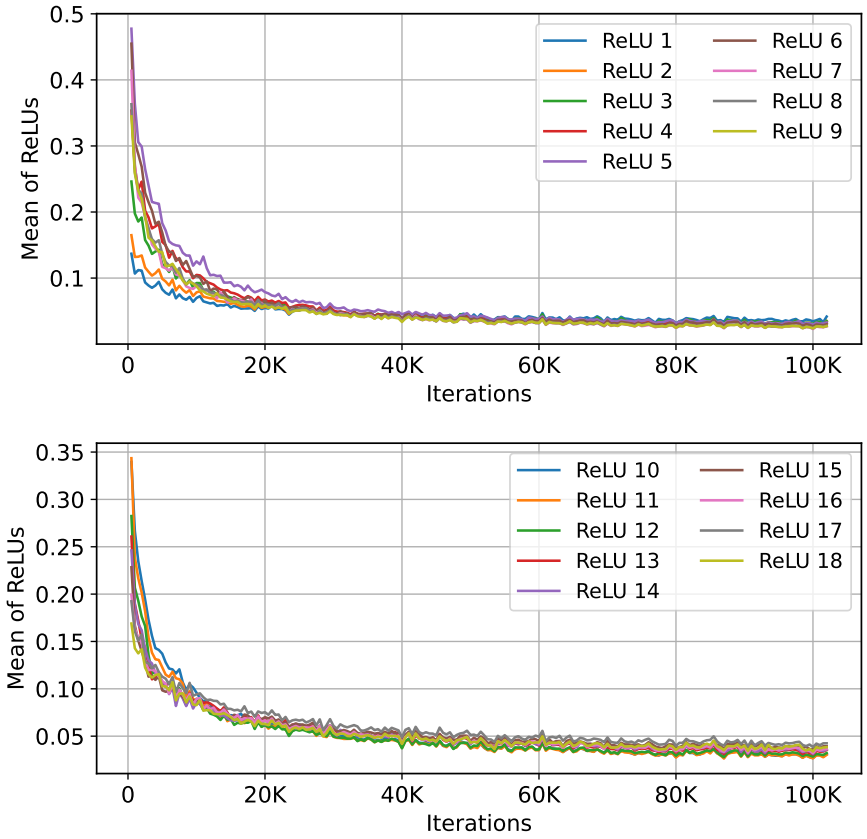


Figure 3.13: The average values of ReLU activations in a highway network.

Fig. 3.12 and Fig. 3.13 show the means of ReLU activation, which were measured at each of 18 blocks for the two compared networks. The mean of ReLU activations in the baseline network starts to increase for the first 8 blocks, reaching a peak of around 30 before decreasing back to a small non-zero value

of about 0.02. The first three blocks are less effective; they do not learn well or even provide noise to the rest. The next 5 blocks are correcting, resulting in a high value for the mean of ReLU in order to meet the range of output. Noting that the range of input and output images are normalised into a range of [-1, 1]. Therefore, high values of ReLU are unusual. The vanishing gradient issue does happen at the first three blocks in the baseline network. In contrast, the activation range for ReLU in the highway network is kept at a small non-zero value. All weights, from the first to the last layer, are learned. In theory, the highway connection can keep gradients flowing by setting sigmoid(.) = 1, which is equivalent to the skip connection in ResNet. The Fig. 3.14 shows the baseline network learning at a slow pace compared to that using highway connections.

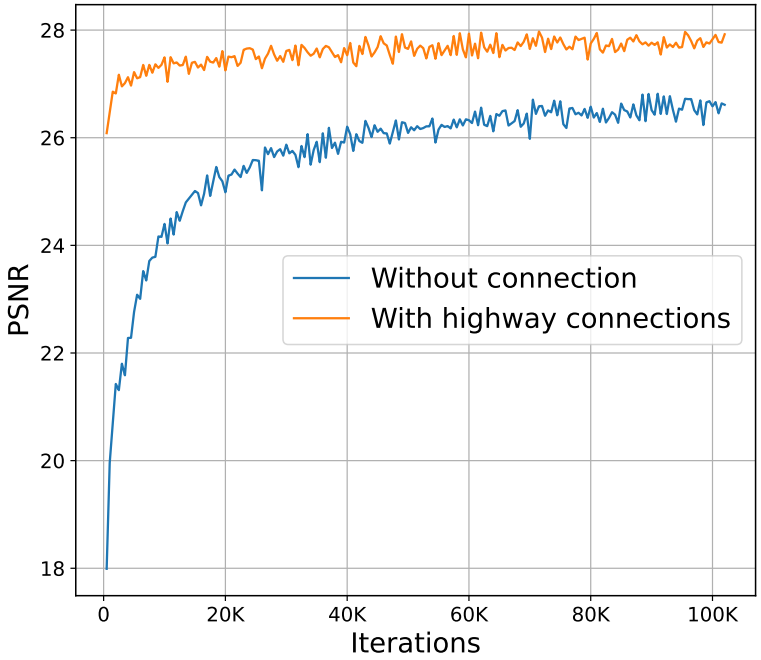


Figure 3.14: The PSNR of a network with/without highway connections.

The mean values of carry gates, which are calculated across both spatial and channel dimensions, are shown in Fig. 3.15. Their values range from 0 to 1,

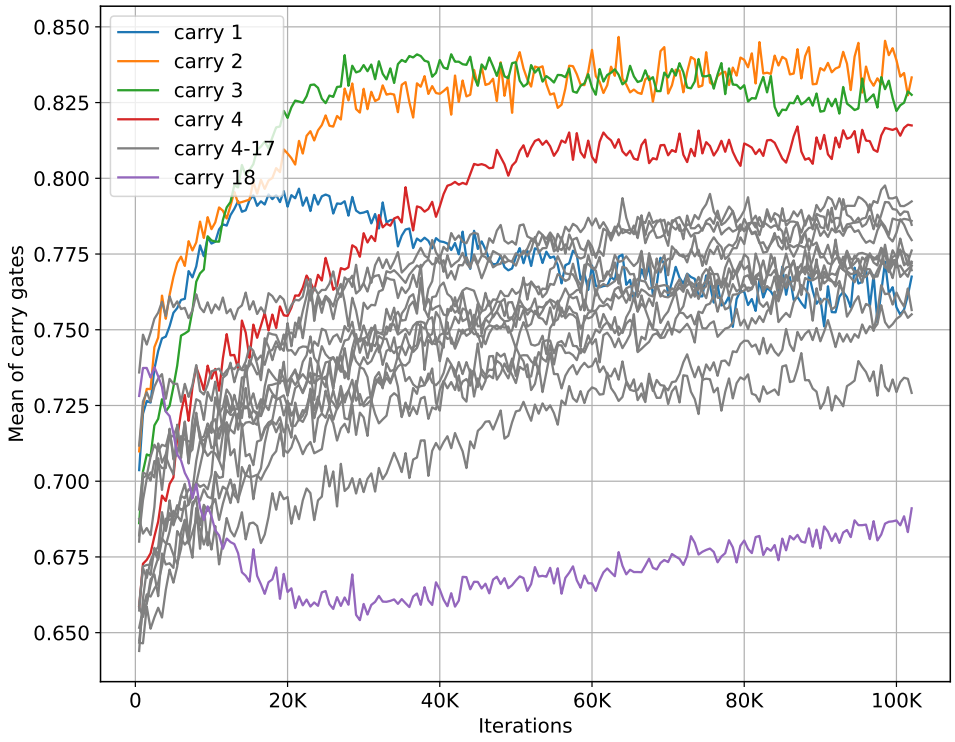


Figure 3.15: Mean of 18 carry gates.

indicating how much the features from the previous layers will be used. Fig. 3.15 also shows how the highway connection solves the gradient vanishing in the baseline network by setting high values for carry gates 2-4, ranging around 0.825. As a result, the mean values of ReLU 2-4 and subsequent ones do not increase as rapidly as the baseline one. The activation range of ReLU in the first block is not large, so the mean value of the first carry gate does not need to increase. The network's input is normalised with a zero mean, so its output is also expected to have a zero mean. As a result, the mean value of the 18th carry gate must be the lowest, close to 0.5 (sigmoid(0) = 0.5). The visual observation in Fig. 3.11 is compatible with the mean values of carry gates. They prove that the highway connections help to focus on particular parts of the image as well as

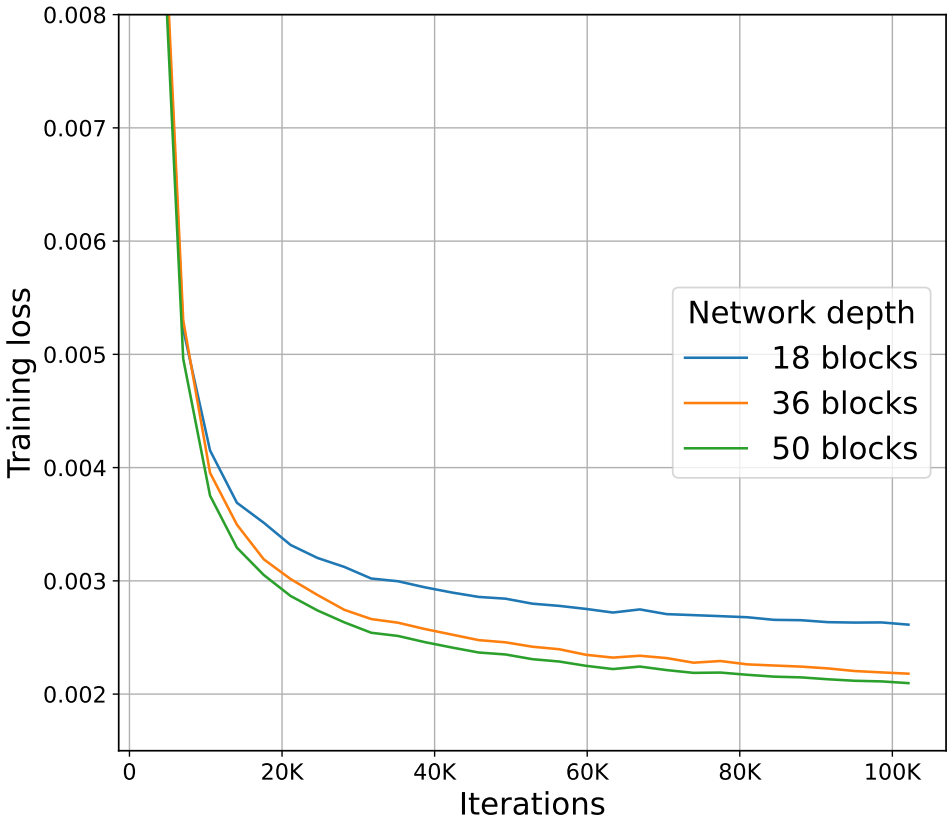


Figure 3.16: Convergence plots for various depths of network.

adjust gradients differently between blocks for the best learning.

To demonstrate whether performance will degrade with a deeper structure, several lightweight networks with the same proposed highway but varying depths are trained. Because the weight can learn to set the output of a sigmoid at 1, which is equivalent to skip connections, it is expected that the performance of a deeper network will not be worse than a shallower one. Fig. 3.16 shows the convergence plot for networks of 18, 36, and 50 blocks, respectively. It has shown no sign of optimisation difficulties for deeper networks. The network with a depth of 50 blocks achieves the lowest training error, followed by a network of 36 blocks. While there is a notable gap between the training errors of 18- and 36-block networks, the performances of networks with 36 and 50 blocks are

insignificantly different. Further experiments with network depths of more than 50 blocks have not shown a large improvement.

While the number of parameters is almost the same as for the skip connection, the downside of the proposed highway connection is that it is computationally expensive due to its exponent calculations. Models using highway connections are more likely to converge into local minima than those using skip connections. As previously discussed, the skip connection-based networks are less susceptible to that problem due to their ensemble behavior. However, training the ensemble network would prevent it from reducing training errors. Therefore, a highway connection inside a skip connection can balance the trade-off between accuracy and generality. The future of work with multiple-level highway connections is worth researching.

3.4 Summary

In this chapter, a novel CNN-based architecture is proposed for improving the accuracy of image reconstruction in single image super-resolution. The proposed model has a similar architecture to the ResNet [50] except for using highway connections instead of residual connections. It does not require the implementation of a complicated structure and enables stable training with fewer problems of exploding gradients or dying neurons. The quantitative results have shown the competitive performance of the proposed architecture compared to other skip-based networks. Visual observations reveal that all skip-based methods have a similar structural pattern of errors, despite their different architectures and PSNRs that vary widely. Hence, building a deeper network with skip connections will not guarantee improved perceptual quality. As opposed to SSIM, the MSE and thus PSNR perform badly in discriminating structural content in images since various types of degradation applied to the same image can yield the same

value of the MSE [145]. The SSIM approach is motivated by the observation that image signals are highly structured, meaning that samples of image signals have strong neighbor dependencies, and these dependencies carry important structures of objects in the image. As expected, the proposed model provides a higher value of SSIM than those using skip connections. With the attention mechanism, a different weight is forced to be learned for each pixel rather than treating every pixel equally as with skip connection. By this way, the neighboring dependencies can be captured, which will improve SSIM.

Chapter 4

Fusion-based Image Super-Resolution

4.1 Introduction

When the high spatial resolution data of the interest scene is available, one can employ data fusion algorithms to increase the spatial resolution of the hyperspectral data. Unfortunately, using multiple sources for MS/HS fusion is still limited. A few satellite platforms that incorporate two imagers, including Spot-5, Gaofen-2, and Gaofen-5, have just provided panchromatic and multispectral images, which are used for pan-shapening rather than MS/HS fusion. Some platforms, such as EO-1, PROBA-1, PRISMA, EnMap, etc., acquire only hyperspectral data. Theoretically, data fusion requires source data to have been collected using the same platform and under the same conditions of observation, such as the same atmosphere and illumination. Also, the images should capture the same scene with accurate image registration. Finally, it is important to provide relative sensor properties such as spectral response functions and point spread functions. Recently, the hyperspectral imager suite (HISUI) has become the first earth-observing sensor composed of hyperspectral and multispectral imagers, satisfying

these assumptions. The Hr-MSI and Lr-HSI are suitable for fusion algorithms due to their trade-offs and are now available in practice thanks to technological advances. Although the CNN-based methods have shown impressive performance in SISR, an effective approach to fuse two image sources for HSI SR is still questionable. With the aid of CNNs, the hierarchical features of Lr-HSI and Hr-MSI can easily be extracted, but the fusion-based approach requires to specify how to combine them. Furthermore, since HS data usually shows high dimensional and non-linear capacities, reconstructing HSI is more likely to suffer from distortions. Effectively fusing two types of extracted features and reducing ill-posed problems are needed to be considered for fusion-based HSI SR.

Improperly tackling the difference in spectrum range and spatial dimensions between the Lr-HSI and Hr-MSI may hinder the performance. For example, bicubic upsampling both Lr-HSI and Hr-MSI to the space of to-be-estimated HSI for the inputs of network [59, 60] has increased the computational complexity, which is the case for HSI SR. This approach will certainly impede discriminative learning ability. The interpolated images would contain uninformative pixels that could be treated as equally important as the original pixels by the network. Combining feature maps from different levels may cause performance degradation. The CNN is well-known for extracting features in raw data at various abstract levels. Each layer represents a different abstract feature representation of the input, where deeper levels provide more sophisticated and abstract features. When jointing feature maps from two inputs, the deep level feature maps from Hr-MSI may not provide useful spatial information for shallow feature maps from Lr-HSI, and vice versa. Therefore, the disparity in feature abstraction level should be paid attention when joining two sources of input images.

The reconstruction of high-dimensional HSI requires constraints to restrict the possible solutions and make the model generalisable. The regularisation methods in the generic CNN-based framework may not be sufficient for specific tasks or

image types. Hence, finding additional constraints to regulate HSI-SR solution is still a subject of active research. As the Hr-MSI and the Hr-HSI are both HR images and capture the same scene, the representation of Hr-MSI can be utilised to further improve the generality of the MS/HS fusion network.

The organisation of this chapter is described as follows: Section 4.2 introduces the proposed CNN-based method for HSI SR, where the theoretical basis that forms a proposed model is described, followed by a detailed network structure. Section 4.3 presents the experiment results, including the ablation learning, comparison analysis, and further discussions. Finally, some summary remarks are given in Section 4.4.

4.2 The proposed method

4.2.1 Progressive downsampling and upsampling

The joint learning operation in a CNN-based model is to combine features using the summation or concatenation of the tensors, which normally requires tensors to have the same spatial dimension. Since the observed Hr-MSI and Lr-HSI have different spatial resolutions, two stages are employed for the fusion framework, as detailed below. In the first stage, the Hr-MSI is progressively downsampled into multi-scales and then fused with the Lr-HSI of the same spatial size. For the second phase, there are three commonly used upsampling techniques for image super-resolution, i.e., pre-upsampling, post-upsampling, and progressive-upsampling. When the upsampling factor is large, the first two techniques increase either the parameters of the network or the difficulty of training. The progressive upsampling method, however, allows the training to gradually shift its attention from the large-scale structure of image to finer-scale details, instead of having to learn all scales simultaneously. Therefore, the architecture appears similar to the U-Net [147], which can not only significantly reduce the learning

difficulty but also improve the performance.

4.2.2 Multi-Task learning

Multi-task learning [148] has been shown to improve the generalisation performance. Apart from the Hr-HSI task, an auxiliary unsupervised task is introduced, which reconstructed Hr-MSI from given Hr-MSI. Intuitively, the observed Hr-MSI and estimated Hr-HSI must share similar spatial information, as shown in Fig. 4.1; otherwise, the MS/HS fusion task becomes trivial.

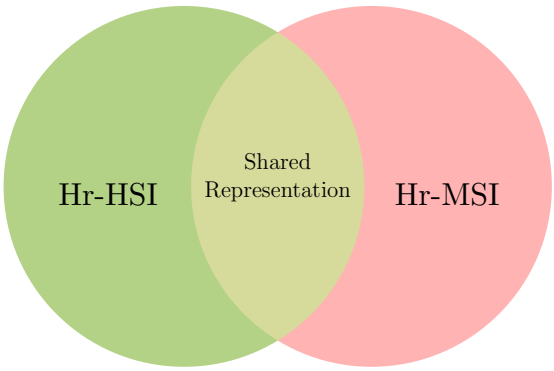


Figure 4.1: Both observed Hr-MSI and estimated Hr-HSI share the spatial representation.

This shared representation is essential for estimating both the Hr-MSI and Hr-HSI, where this feature is representative for the Hr-MSI data and also crucial for estimating the Hr-HSI. Directly estimating of the Hr-HSI from any given Lr-HSI and Hr-MSI is likely an under-constrained problem. This means solutions can be found to well fit the data but often fail to extract the underlying patterns in the data, resulting in poor generalisation. Introducing an auxiliary task for reconstructing Hr-MSI will train the model to find the solution over a small area of the intersection of two tasks rather than on a broader area of a single task. Therefore, this can help the network achieve faster and better convergence. Moreover, the auxiliary task acts as a regulariser by introducing a reductive bias,

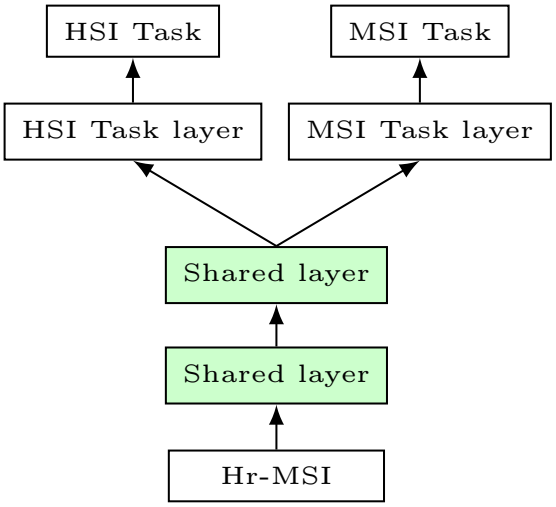


Figure 4.2: Hard parameter sharing for multi-task learning in deep neural networks.

where the number of possible solutions can be reduced.

Hard parameter sharing is the most commonly used approach in multi-task learning with a neural network, as shown in Fig. 4.2. It is generally applied by sharing the hidden layers between all tasks while keeping several task-specific output layers. When training jointly, both Hr-MSI and Hr-HSI tasks must have agreement on features to reduce the total error reconstruction, which enables the shared layer to capture the common features of both. This is equivalent to the sparse representation-based method, for example, the Non-Local Sparse Tensor Factorisation (NLSTF) [67] method, that uses the non-local self-similarity of Hr-MSI to impose spatial constraints on estimated Hr-HSI.

4.2.3 Denoising with the autoencoders

Given Hr-MSI as the high-resolution image, an autoencoder can also be used as an auxiliary task for learning a compressed representation of Hr-MSI, which is then used to impose regularisation on the HSI SR. The convolutional autoencoder is an unsupervised learning method that first learns the representations by performing

convolution and downsampling on the input. These representations are then decoded by up-sampling and convolutions to reconstruct the original image of the input. The denoising autoencoders [103], is an extension to the classical autoencoder, which reconstructs the input from a corrupted version of it.

4.2.4 Network architecture

For notational convenience, all Lr-HSI, Hr-MSI, and Hr-HSI are denoted as two-dimensional matrices. Let the matrix representing the Lr-HSI be $\mathbf{Z} \in R^{C \times hw}$ with C bands and spatial dimension hw, and let denote $\mathbf{Y} \in R^{c \times WH}$ the obtained Hr-MSI with c spectral bands and spatial dimension WH. The goal is to estimate the Hr-HSI, present as $\mathbf{X} \in R^{C \times WH}$, with both high spatial and spectral resolutions. In general, Hr-MSI has much higher spatial resolution than Lr-HSI ($HW \gg hw$), and Lr-HSI has a much higher spectral resolution than the Hr-MSI ($C \gg c$).

The Lr-HSI can be regarded as a spatially down-sampled version of the Hr-HSI:

$$\mathbf{Z} = \mathbf{XBS} \tag{4.1}$$

where $\mathbf{B} \in R^{WH \times WH}$ represents a convolution between the point spread function (PSF) of the sensor and the Hr-HSI band, and $\mathbf{S} \in R^{WH \times wh}$ is a downsampling matrix. Similarly, the Hr-MSI, e.g. a RGB/PAN image, can be taken as a spectrally downsampled version of the Hr-HSI:

$$\mathbf{Y} = \mathbf{RX} \tag{4.2}$$

where $\mathbf{R} \in R^{c \times C}$ is the corresponding camera spectral response function.

The problem of HSI SR can be solved by learning the mapping between $\mathbf{X}$ and the coupled $\mathbf{Y}$, $\mathbf{Z}$ below in a fully convolutional fashion using the gradient

descent. The proposed multi-task objective is represented as:

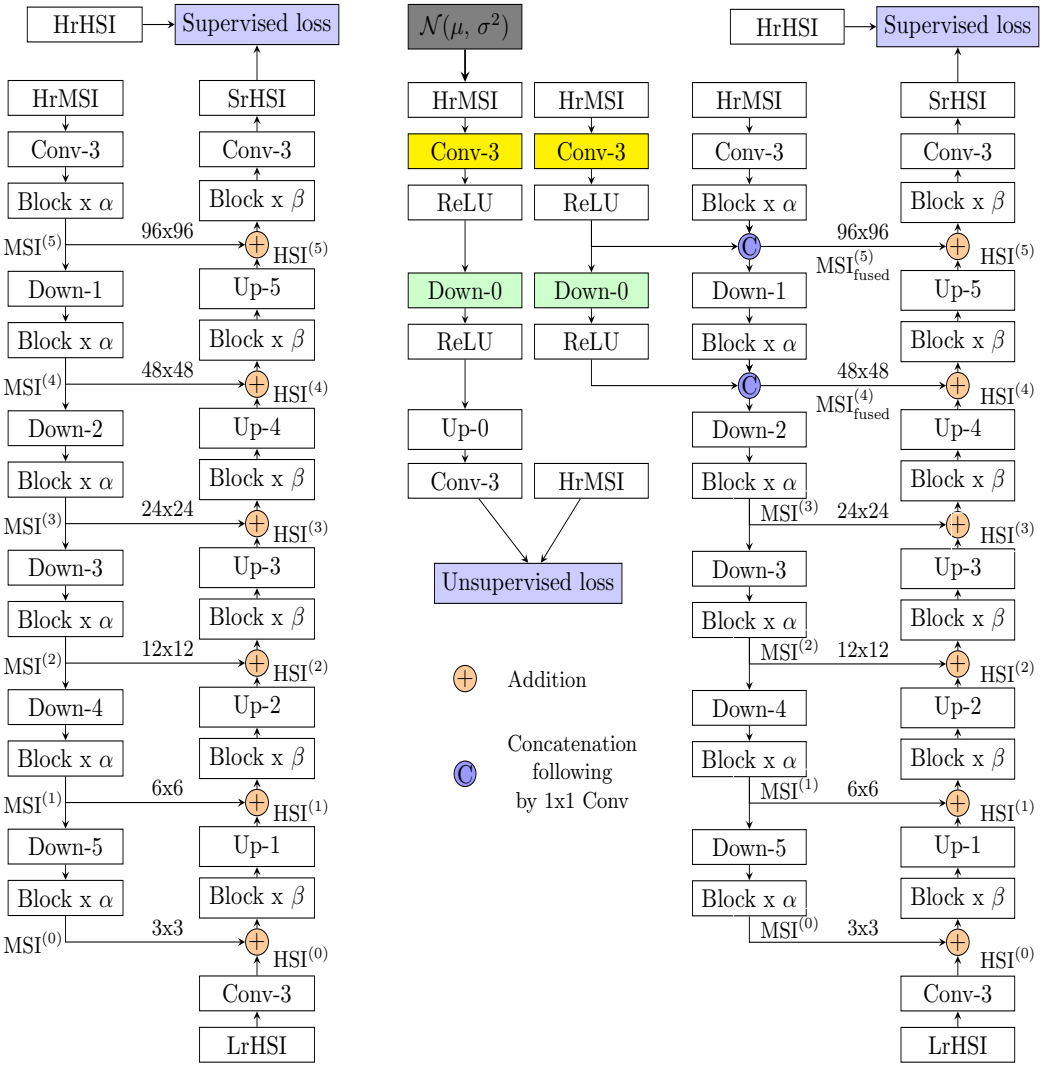
$$\underset{\theta,\psi}{\operatorname{argmin}} \parallel \mathbf{f}(\mathbf{X}|\boldsymbol{\theta},\mathbf{Y},\mathbf{Z}) - \mathbf{X} \parallel_2^2 + \gamma \parallel \mathbf{g}(\mathbf{Y}|\boldsymbol{\psi},\widetilde{\mathbf{Y}}) - \mathbf{Y} \parallel_2^2 + \eta R(\mathbf{X}) \tag{4.3}$$

where $\mathbf{f}(\mathbf{X}|\boldsymbol{\theta},\mathbf{Y},\mathbf{Z})$ and $\mathbf{g}(\mathbf{Y}|\boldsymbol{\psi},\widetilde{\mathbf{Y}})$ are the outputs of the proposed network; $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ are trainable parameters of two sub-networks. During the multi-task learning, part of $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ is shared, as illustrated in Fig. 4.2. The first and second terms are the pixel-wise L_2 distance between the network outputs and the corresponding ground-truth $\mathbf{X}$ and $\mathbf{Y}$, respectively. The final term refers to the $L2$ regularisation, which aims to penalise more with large learnable parameters and less with smaller ones. The regularisation coefficients γ and η are two predefined values that need to be chosen to balance generality and accuracy in the main task. When γ and η are large, the denoising task and the $L2$ regularisation outweigh the primary task and prevent it from reducing training errors.

There are two major objectives for designing the proposed fusion network. One is to reduce the spatial discrepancy between the two observed data. The other is to improve the generalised representation by sharing the main supervised task with an unsupervised auxiliary task. These representations are not only useful to support the decision for the supervised task but also work as a regulariser for more effective HSI SR [149].

A MS/HS fusion network is detailed in Fig. 4.3, in which Fig. 4.3 (a) illustrates the baseline architecture and Fig. 4.3 (b) a baseline architecture extended with the proposed auxiliary task. The construction of proposed model involves a top-down pathway, a bottom-up pathway, an auxiliary task, and some lateral connections, as introduced below.

Top-down pathway (MSI branch). In this pathway, the given training Hr-MSI is progressively down-sampled with a scaling factor of 2 into five hierarchical spatial levels, starting from an image sized of $96 \times 96 \times 3$ to $3 \times 3 \times 31$. Often,



(a) Baseline architecture without auxiliary task.

(b) MSAT architecture.

Figure 4.3: The architecture of proposed MSAT. The same yellow or green colour boxes indicates that those variables are shared between supervised and unsupervised tasks.

there are many layers that produce output maps of the same size, which are defined in the same network *stage* or *level*. Let $Y^{(s-1)}$ and $Y^{(s)}$ denote the input and output feature maps of the s-th *level* in the MSI branch, and the relation

between $Y^{(s-1)}$ and $Y^{(s)}$ is formulated by:

$$Y^{(s-1)} = Resblock(Downsample(Y^{(s)})) \tag{4.4}$$

where $Resblock(\cdot)$ and $Downsample(\cdot)$ denote respectively the ResNet block and a downsample operation using a convolution layer with *strike* = 2. The highest *level* (s = 5) is the feature maps extracted from the observed Hr-MSI without downsampling.

An auxiliary task (Denoising branch): In the proposed model (Fig. 4.3b), a *light* Denoising Auto-Encoder (DAE) is introduced as an auxiliary task, which is trained to reconstruct the original observation $\mathbf{Y}$ from its corrupted version $\widetilde{\mathbf{Y}}$ by minimising the error between the input $\mathbf{Y}$ and its reconstruction $\mathbf{g}(\mathbf{Y}|\boldsymbol{\psi}, \widetilde{\mathbf{Y}})$ from the corrupted $\widetilde{\mathbf{Y}}$. With the presence of noise, the DAE is forced to learn the representation of the data, which later is able to reconstruct the original input. The corrupted $\widetilde{\mathbf{Y}} = \mathbf{Y} + \mathcal{N}(\mu, \sigma^2)$ is used to train the DAE with the clean version Y fed into the Encoder to extract the underlying representation for both tasks. Formally, the representation of Hr-MSI at multiple levels $\bar{Y}_s, \bar{Y}_{s-1}, \ldots, \bar{Y}_0$ are extracted as follows:

$$\hat{Y}_s, \hat{Y}_{s-1}, \ldots, \hat{Y}_0 = \mathbf{Encoder}(\widetilde{Y}) \tag{4.5}$$

$$\dot{Y}_s, \dot{Y}_{s-1}, \ldots, \dot{Y}_0 = \mathbf{Decoder}(\hat{Y}_s, \hat{Y}_{s-1}, \ldots, \hat{Y}_0) \tag{4.6}$$

$$\bar{Y}_s, \bar{Y}_{s-1}, \ldots, \bar{Y}_0 = \mathbf{Encoder}(Y) \tag{4.7}$$

The features $\bar{Y}_s, \bar{Y}_{s-1}, \ldots, \bar{Y}_0$ that come after the ReLU activation are used for a supervised task, and proposed model is trained in an end-to-end manner.

Lateral connections between the main task and the auxiliary task: The DAE relies on a certain number of training (noisy) examples to learn the representations/patterns before transferring them to the main task. Our pri-

mary task should thus be to determine when to use such information and when to discard irrelevant ones. The simple mechanism is to use a 1×1 convolution layer. However, one problem with this is that the main task may neglect shared representations by setting the kernel parameters to zero. The total loss is then minimised by decreasing each supervised and unsupervised loss separately. The representations learnt by the DAE thus are of no use to the main task. To avoid this unwanted effect, a compression mechanism is introduced by taking advantage of a 1×1 convolution layer. Concretely, the feature maps extracted from Hr-MSI in both the denoising task and the main task are concatenated and sent to a 1×1 convolution layer. This layer performs dimensionality reduction and forces the main task to utilise the information from the denoising task.

$$Y_{fused}^{(l)} = Conv_{1x1}(Concatenate(Y^{(l)}, \bar{Y}^{(l)})) \tag{4.8}$$

Bottom-up pathway (HSI branch): The bottom-up pathway hallucinates higher resolution features by up-sampling the spatial feature maps from lower levels of the Lr-HSI.

$$X^{(s-1)} = Upsample(Resblock(X^{(s-2)})) \tag{4.9}$$

where $Resblock(\cdot)$ denotes ResNet block and $Upsample(\cdot)$ is a upsampling operation using a transposed convolution layer. The up-sampled map is then merged with the corresponding top-down map by element-wise addition.

$$\hat{X}^{(s-1)} = X^{(s-1)} + Y_{fused}^{(s-1)} \tag{4.10}$$

The top-down pathway is rich in spatial information, while the bottom-up pathway contains a high level of spectral information.

To build a deep network without changing the network topology, the param-

eters α and β control the depth of the network. Only one residual block ($\alpha = 1$, $\beta = 1$) is used at a certain spatial levels unless stated otherwise. Our residual block is derived from the MobileNetV1 [150], in which the conventional 3×3 convolution is replaced by a 3×3 depth-wise separable convolution. The down-sampling and up-sampling blocks refer to one-step convolution with stride = 2 and a transpose convolution, respectively.

4.3 Experimental results

4.3.1 Experimental Datasets

For performance evaluation, we conduct experiments on five public benchmark datasets: CAVE [151], Harvard [152], ICVL [153], Chikusei [154], and a space-borne image of *Roman Colosseum* acquired by World View-2. The first three datasets are widely used in hyperspectral image super-resolution. They contain images with sufficient high spatial and spectral resolutions, as well as showing the diversity of objects, conditions in which images were captured, and number of bands. The airborne and spaceborne hyperspectral datasets are limited; they do not satisfy the above requirements or even lack ground truth, or their dataset size is insufficient. Therefore, Chikusei and *Roman Colosseum* are airborne and spaceborne, respectively, and were simply chosen to follow previous work [54]. Additional details of five datasets are given in Appendix A.

The CAVE dataset [151] comprises 32 indoor HSIs captured under controlled illumination. The images have 31 spectral bands with a spatial dimension of 512 $\times$ 512 pixels, and a spectral sampling gap of 10nm from 400nm to 700nm. The Harvard dataset [152] has 50 indoor and outdoor images, recorded under daylight illumination, where 27 images were under artificial or mixed illumination. With a spatial size of 1392 $\times$ 1040 pixels, each HSI has 31 spectral bands, with a 10-nm spectral sampling gap within [420, 720] nm. The ICVL dataset [153] contains

201 HSIs of real-world indoor and outdoor scenes, has 31 spectral bands each ranging from 400 nm to 700 nm at a 10 nm increment. Only the top left 1024 × 1024 pixels is used for convenience of the spatial down-sampling. The Chikusei scene [154] is an airborn HS image taken over Chikusei, Ibaraki, Japan. The image has a spatial dimension of 2517 × 2335 pixels, comprising 128 bands in the spectral range from 363 to 1018 nm. A 500 × 2210 pixel-size image from the top area of the original data is selected for training. Besides, 16 non-overlapped 448 × 448 images are extracted as the testing set.

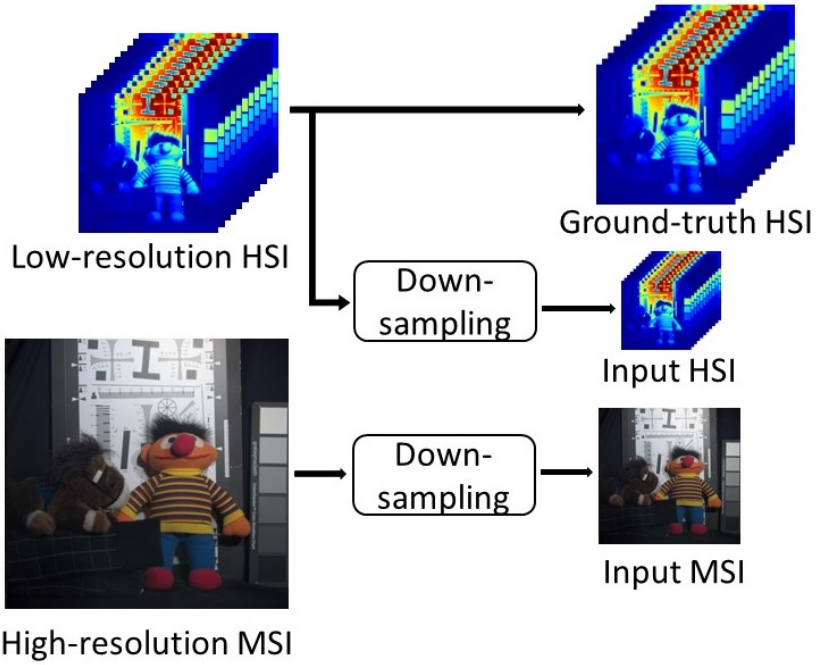


Figure 4.4: Procedure to create training data when the Hr-HSI is unavailable.

The sample images of the Roman Colosseum contain an Hr-MSI (RGB image) of size 1676 × 2632 × 3 and Lr-HSI image of size 419 × 658 × 8. The 208 × 658 and 836 × 2632 pixels image from Lr-HSI and Hr-MSI are selected for training and the remaining for testing data. Since the ground truth is not available in this case, we follow Wald's protocol [155] to create the simulated experiments, as illustrated in Fig. 4.4. All original images are blurred by a 9 × 9 Gaussian

smoothing kernel and downsampled by a factor of 4. These downsampled images are treated as training data, and the original Lr-HSI is regarded as the ground truth. For each of four other databases, the original Hr-HSI are used as the ground-truth images. The Hr-HSI is then downsampled by averaging the 32 $\times$ 32 disjoint spatial blocks to generate the Lr-HSI. The Hr-MSI (RGB image) of the same scene is stimulated by down-sampling X with a spectral model using a spectral down-sampling matrix derived from the response of a Nikon D700 camera. The CAVE, Harvard, and ICVL datasets are split into a training set of 20 images, 30 images, and 75 images and a test set of 12 images, 20 images, and 25 images, respectively.

To prepare the training samples, the 96 $\times$ 96 overlapped patches from the training images are extracted as reference Hr-HSI images. The Hr-HSI, Hr-MSI and Lr-HSI images are sized of 96 $\times$ 96 $\times$ S, 96 $\times$ 96 $\times$ 3 and 3 $\times$ 3 $\times$ S, respectively, where S refers to the number of spectral bands in each experimental datasets. The weighting factor γ is fixed within [1e-3, 1e-2] to balance the supervised loss and the unsupervised loss. When γ is too small, i.e., 1e-4, the problem (4.3) is reduced to solving one single-task learning problem. On the other hand, when γ is too large, i.e., 1e-1, the auxiliary task can prevent the primary task from reconstructing the details.

4.3.2 Training Setup

All experiments are run on TensorFlow with CUDA 9.0 and cuDNN backends on an NVIDIA GeForce GT 1030 GPU. We train the model with 40,000 iterations using a batch size of 16. The ADAM optimization [142] algorithm was used with an initial learning rate of 0.00035, which reduces by 30% after every 10,000 iterations. Only the flipping was used as a data augmentation to reduce the training time. In the denosing branch, Gaussian noise added to the original inputs is zero-mean with a variance within [0.05, 0.2].

4.3.3 Experimental results

The parameters α and β refers to the numbers of Residual blocks that would be used in the top-down pathway and bottom-up pathway shown in Fig. 4.3. We have found that increasing the number of blocks in a top-down pathway does not improve accuracy while introducing more computational cost. The bottom-up pathway is the main part, which is responsible for reconstructing the HSI, and its depth will affect the accuracy. The number of blocks at each stage in the bottom-up pathway is set according to the size of the dataset. Therefore, the MSAT is set up with $\alpha = 0$ and $\beta = 1$ for small training dataset of CAVE, Harvard, Chikusei, and Roman Colosseum and $\alpha = 0$ and $\beta = 2$ for large dataset of ICVL. Since deep learning-based method needs training, the performance on the testing set is compared instead of the full dataset. The comparion methods include: non-local sparse tensor factorization (NLSTF) [1] [67], non-negative structured sparse representation (NSSR) [2] [65], and low tensor-train rank representation (LTTR) [3] [69] methods, which represent the state-of-the-art sparse representation based approaches; the hyperspectral super-resolution network (HSRnet) [4] [60] and the model-guided deep convolutional network (MoG-DCN) [5] [59] represent the state-of-the-art deep learning-based SR methods. For quantitative evaluation, RMSE, ERGAS, SAM and SSIM (see Subsection 2.3.11 (B)) are utilised. Table 4.1 shows the average results of the compared methods on the CAVE testing set, where the best results are highlighted in bold for clarity. As seen, the proposed method achieves the better performance than all others in terms of ERGAS, SAM and SSIM, although the RMSE is not the least. With just a few samples used for training suggests that proposed model has the potential to further improve the RMSE scores when more training images are available.

[1] https://github.com/renweidian/NLSTF
[2] https://see.xidian.edu.cn/faculty/wsdong/HSI_SR _Project.htm
[3] https://github.com/renweidian/LTTR
[4] https://github.com/liangjiandeng/HSRnet
[5] https://github.com/chengerr/Model-Guided-Deep-Hyperspectral-Image-Super-resolution

Table 4.1: Average quantitative results of the compared methods using 12 testing images on the CAVE dataset.

Method	**RMSE↓**	**ERGAS ↓**	**SAM↓**	**SSIM↑**
NLSTF [67]	3.14±1.24	0.46±0.30	6.57±2.41	0.976±0.012
NSSR [65]	2.77±1.29	0.42±0.31	5.70±2.02	0.980±0.011
LTTR [69]	**2.64**±1.59	0.38±0.26	6.24±2.25	0.982±0.010
HSRnet [60]	3.36±1.70	0.39±0.33	4.78±1.14	0.980±0.010
MoG-DCN [59]	3.33±1.68	0.37±0.28	4.57±0.99	0.984±0.007
MSAT	3.25±1.61	**0.36±0.25**	**4.25±0.93**	**0.985±0.005**

The quantitative averages on the Harvard database are compared in Table 4.2. Although none of these methods can consistently outperform others, the LTTR [69] seems to perform better on the Harvard dataset. The proposed approach achieves competitive results in terms of RMSE and SSIM, where the ERGAS and SAM are slightly worse than others.

Table 4.2: Average quantitative results of the compared methods over 20 testing images on the Harvard dataset.

Method	**RMSE↓**	**ERGAS ↓**	**SAM↓**	**SSIM↑**
NLSTF [67]	2.66±1.30	**0.31±0.21**	3.36±1.72	0.974±0.014
NSSR [65]	2.52±1.24	0.34±0.22	3.23±1.60	0.975±0.014
LTTR [69]	2.19±1.15	0.34±0.21	**3.09±1.29**	**0.979**±0.011
HSRnet [60]	2.62±1.33	0.36±0.26	3.46±1.17	0.973±0.016
MoG-DCN [59]	2.21±1.13	0.35±0.24	3.39±1.50	0.979±0.011
MSAT	**2.18±1.05**	0.35±0.23	3.39±1.49	**0.979±0.010**

Fig. 4.5 shows a reconstructed image from the Harvard test dataset. As the NLSTF [67] method is actually a variation of the NSSR [65] algorithm, visual inspection validates that the former closely resembled patterns in the latter. The reconstructed images from three deep learning-based methods also follow the closely mirrored patterns. Among them, the LTTR [69] and the proposed MSAT recover more spatial details of the HSI.

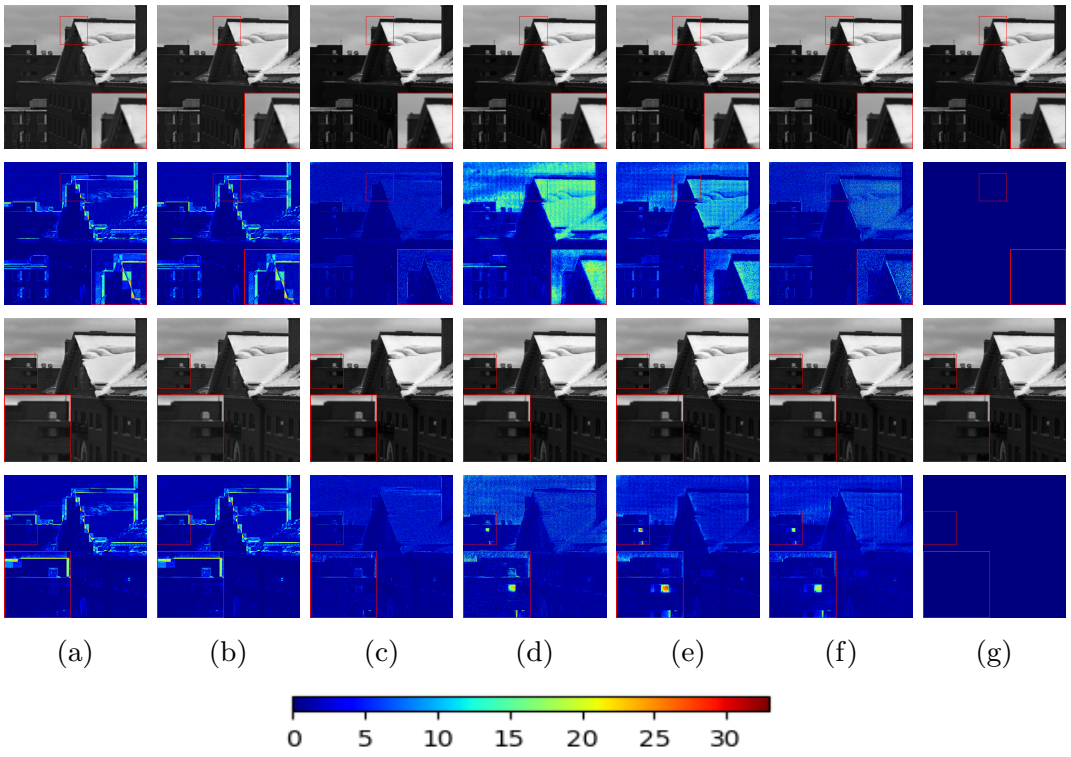


Figure 4.5: First and second row: the reconstructed images and the corresponding error images of the compared methods for Harvard at 460nm band. Third row and Fourth row: reconstructed images and corresponding error images of the compared methods for Harvard at 620nm band. (a) the NLSTF method [67] (RMSE = 3.33, ERGAS = 0.19, SAM = 2.34, SSIM = 0.96). (b) the NSSR method [65] (RMSE = 3.36, ERGAS = 0.20, SAM = 2.51, SSIM = 0.96). (c) the LTTR method [69] (RMSE = 1.87, ERGAS = 0.161, SAM = 2.27, SSIM = 0.972). (d) the HSRnet method [60] (RMSE = 3.12, ERGAS = 0.193, SAM = 2.59, SSIM = 0.963). (e) the MoG-DCN method [59] (RMSE = 2.62, ERGAS = 0.189, SAM = 2.41, SSIM = 0.972). (f) Proposed MSAT (RMSE = 2.37, ERGAS = 0.173, SAM = 2.38, SSIM = 0.972). (g) Ground-truth.

Obviously, deep learning-based methods require sufficient features by a grant from a larger amount of training data or properties of the datasets. As a result, the small training dataset, as well as the high training/test split ratio from CAVE (20 images/12 images $\approx$ 62.5/37.5%) or Harvard (30 images/20 images $\approx$ 60/40%), will cause high variance in the training of model or overfitting. Another issue is an unrepresentative training dataset, which means that the data

Table 4.3: Average results of the compared methods (25 testing images, 75 training images).

Method	**RMSE↓**	**ERGAS ↓**	**SAM↓**	**SSIM↑**
NLSTF [67]	1.73±0.63	0.12±0.05	1.06±0.37	0.991±0.003
NSSR [65]	1.74±0.60	0.128±0.047	1.05±0.35	0.991±0.003
LTTR [69]	1.13±0.39	0.08±0.04	0.10±0.32	0.994±0.001
HSRnet [60]	1.65±0.56	0.11±0.04	1.09±0.36	0.996±0.001
MoG-DCN [59]	1.24±0.38	0.08±0.04	1.03±0.34	**0.998**±0.002
MSAT	**1.03±0.32**	**0.07±0.04**	**0.99±0.31**	**0.998±0.000**

available during training is insufficient to capture the model, relative to the validation dataset. Without increasing the model complexity, 100 images from the ICVL dataset are randomly chosen, where 75 images are used for training and the remaining 25 for testing. The performance of the proposed method now consistently outperforms the compared methods significantly with a more considerable margin, as shown in Table 4.3. As seen from Table 4.3, the proposed MSAT method significantly outperforms the compared models of NLSTF [67], NSSR [65], LTTR [69], HSRnet [60], and MoG-DCN [59] in terms of all the four quantitative metrics. Furthermore, the proposed model produced consistently lower variance around the average score than all others.

In Fig. 4.6 and Fig. 4.7, the reconstructed images and the error images are shown, where the test results are for an outdoor image *BGU_0403-1419-1* and an indoor image *objects _0924-1629* from the ICVL dataset. The NLSTF [67] and NSSR [65] again perform worse as shown in the changed brightness while the LTTR [69] and the proposed MSAT approaches perform better regarding the well preserved spatial and spectral structures. The HSRnet [60] and the MoG-DCN [59] are still unable to surpasses the LTTR [69] in ICVL dataset.

Table 4.4 compares the quantitative average of all compared methods using 16 testing images on the Chikusei dataset. As the training and test samples are cropped from the same image, they have common features and do not suffer from

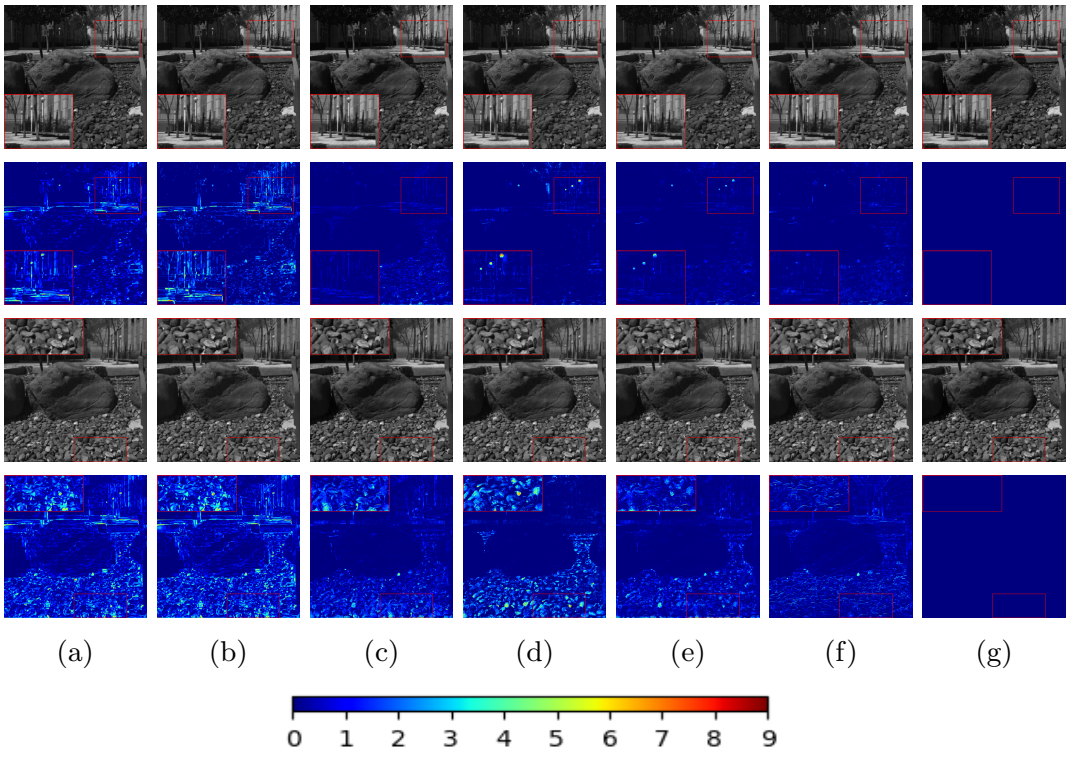


Figure 4.6: The reconstructed images and corresponding error images of the compared methods for ICVL at 460nm band (first two rows) and at 620 nm (the last two rows). (a) the NLSTF method [67] (RMSE = 1.96, ERGAS = 0.13, SAM = 1.17, SSIM = 0.99). (b) the NSSR method [65] (RMSE = 1.93, ERGAS = 0.13, SAM = 1.07, SSIM = 0.99). (c) the LTTR method [69] (RMSE = 1.15, ERGAS = 0.085, SAM = 1.06, SSIM = 0.994). (d) the HSRnet method [60] (RMSE = 1.36, ERGAS = 0.091, SAM = 1.07, SSIM = 0.994). (e) the MoG-DCN method [59] (RMSE = 1.13, ERGAS = 0.067, SAM = 0.098, SSIM = 0.995). (f) Proposed MSAT (RMSE = 0.96, ERGAS = 0.05, SAM = 0.90, SSIM = 0.996). (g) Ground-truth.

overfitting and unrepresentative training dataset. Fig. 4.8 shows the composition of test samples with bands of 70, 100, and 36 as a false-color image with the error image given in all three channels. As seen, the three sparse representation-based approaches perform worse compared to deep learning-based methods. The proposed method significantly outperforms three sparse representation-based methods with a large margin while still performing better than the HSRnet [60] and the MoG-DCN [59]. The composition image obtained from the proposed method

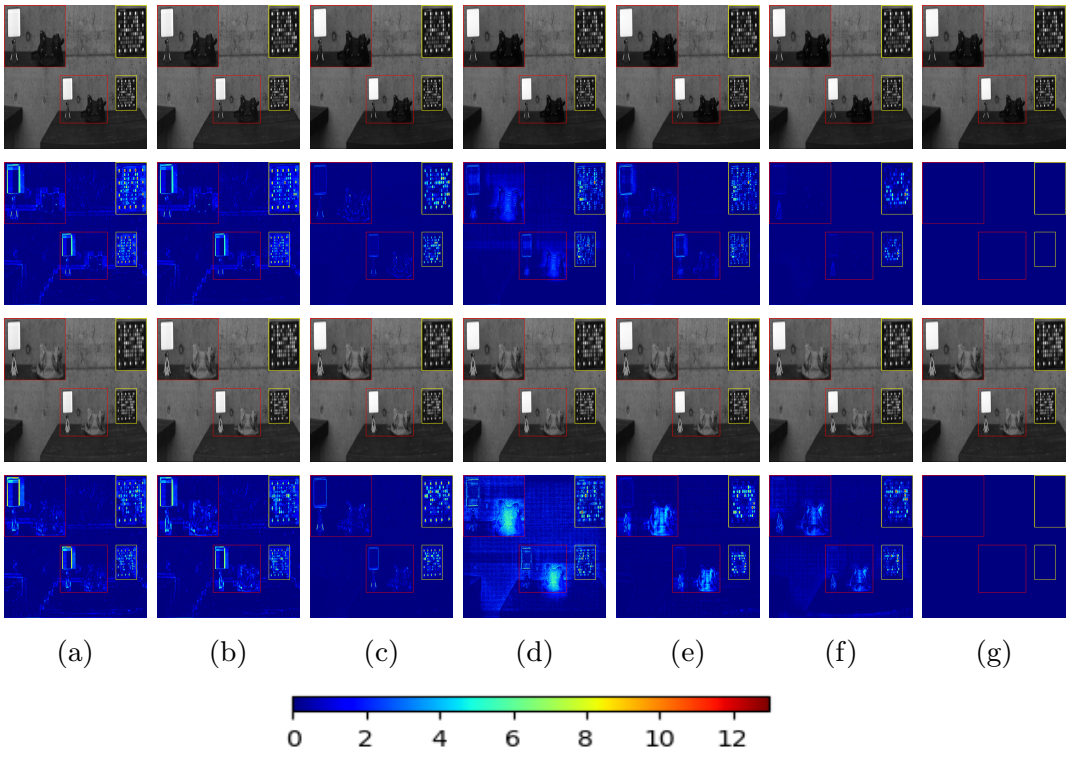


Figure 4.7: The reconstructed images and corresponding error images of the compared methods for ICVL at 540nm band (first two rows) and at 620 nm (the last two rows). (a) the NLSTF method [67] (RMSE = 1.75, ERGAS = 0.07, SAM = 0.64, SSIM = 0.98). (b) the NSSR method [65] (RMSE = 1.69, ERGAS = 0.07, SAM = 0.60, SSIM = 0.99). (c) the LTTR method [69] (RMSE = 1.26, ERGAS = 0.548, SAM = 0.69, SSIM = 0.992). (d) the HSRnet method [60] (RMSE = 1.48, ERGAS = 0.067, SAM = 0.62, SSIM = 0.990). (e) the MoG-DCN method [59] (RMSE = 1.22, ERGAS = 0.534, SAM = 0.68, SSIM = 0.993). (f) Proposed MSAT (RMSE = 1.19, ERGAS = 0.04, SAM = 0.64, SSIM = 0.993). (g) Ground-truth.

is closest to the ground truth, while other methods show obvious unsatisfactory reconstruction.

The fusion result on the real spaceborne HS dataset is shown in Fig. 4.9. As the ground-truth Hr-HSIs are unavailable, the procedure of training is followed and the performance is measured by comparing the result image with an up-sampled image of Lr-HSI. As seen, the result image obtained from the proposed method is much closer to Lr-HSI and Hr-MSI. Furthermore, Fig. 4.10 compares

Table 4.4: Average results of the compared methods over 16 testing samples in the Chikusei dataset.

Method	**RMSE↓**	**ERGAS ↓**	**SAM↓**	**SSIM↑**
NLSTF [67]	2.55±0.67	0.478±0.056	2.78±0.66	0.971±0.007
NSSR [65]	3.94±1.11	0.772±0.112	3.90±0.92	0.943±0.015
LTTR [69]	4.53±1.32	0.683±0.121	3.11±0.53	0.952±0.013
HSRnet [60]	2.32±0.44	0.827±0.163	2.94±0.49	0.970±0.005
MoG-DCN [59]	1.36±0.26	0.483±0.060	2.64±0.32	0.989±0.001
MSAT	**0.93±0.12**	**0.475±0.053**	**2.09±0.30**	**0.992±0.001**

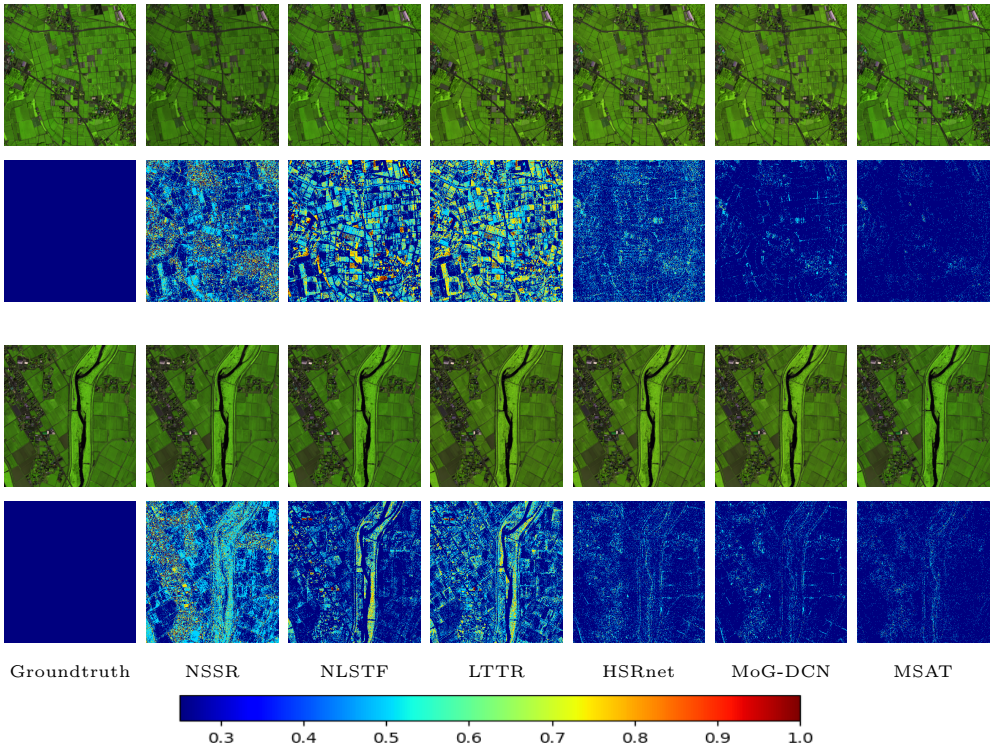


Figure 4.8: The HSI-SR results on the Chikusei dataset of all competing methods. First and Fourth row: the false-color image with bands (70, 100, 36). Second and Fifth row: the corresponding error images compared to the ground-truth.

the performance of three deep learning-based HS/MS fusion methods over the validation set. The HSRnet [60] performs the worst among the three methods, while the MoG-DCN [59] cannot outperform the proposed smaller-size baseline

Figure 4.9: The Hr-MSI (RGB) and Lr-HSI images are of the left bottom area of *Roman Colosseum* acquired by World View-2. The composite image of the HS image with bands 5-3-2 as R-G-B is displayed.

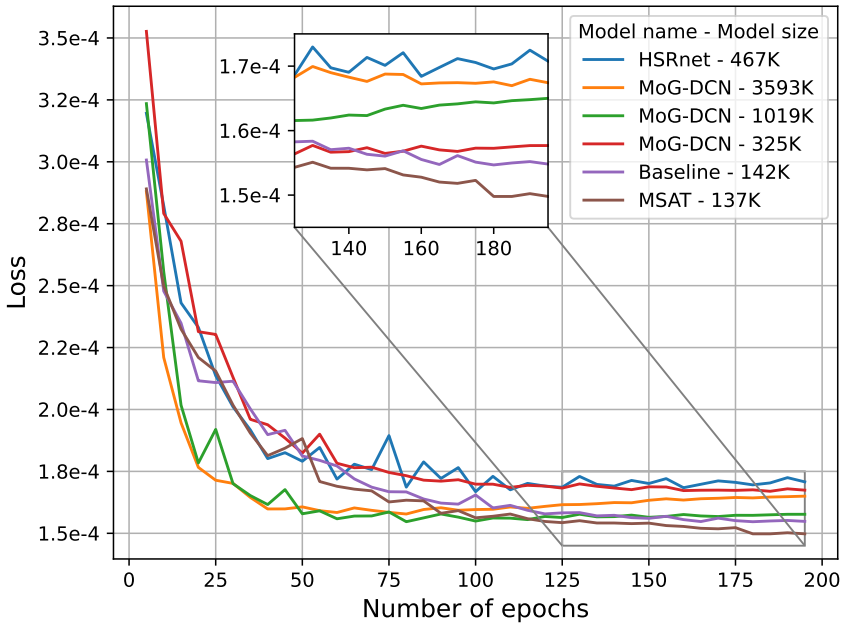


Figure 4.10: Comparison of the proposed MSAT to two deep learning-based methods (HSRnet [60] and MoG-DCN [59]) over the validation set in the *Roman Colosseum* dataset.

model. An introduced auxiliary task provides a consistent gain in generality and achieves the best performance.

4.3.4 Ablation study

A. The effectiveness of multi-scale image decomposition and auxiliary task

An ablation study is performed to verify the effect of Hr-MSI decomposition and the proposed auxiliary task used in training on the CAVE dataset, where the L2 regularisation is turned off for a fair comparison in these evaluations. *w/o 3×3* is denoted as the case without Hr-MSI decomposition to the spatial size of 3×3 whilst keeping other settings the same. It is observed that more scales the Hr-MSI is decomposed, the better performance it generates. As shown in Fig. 4.11 and

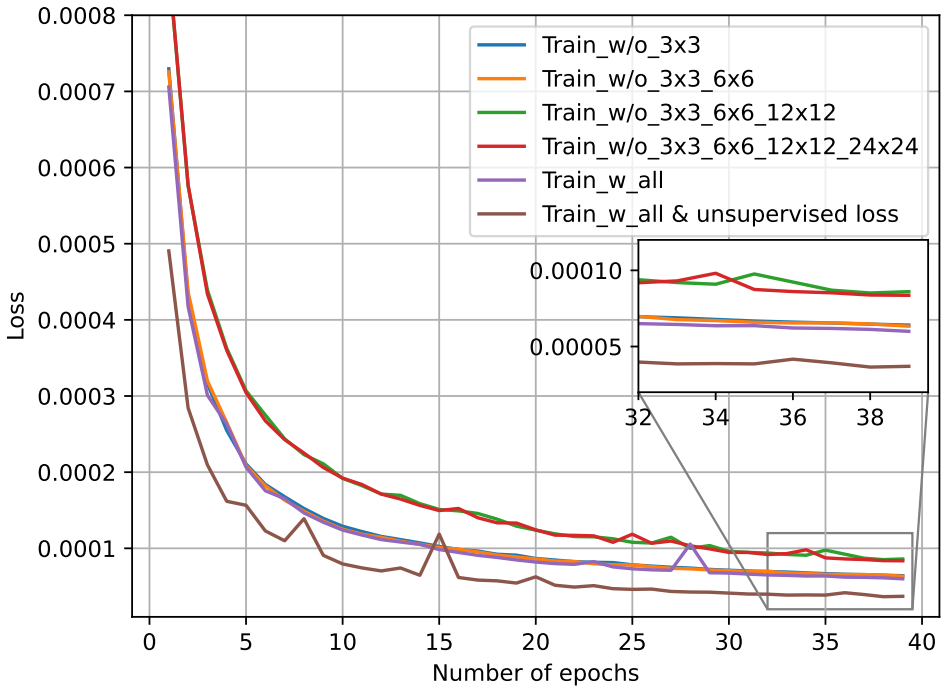


Figure 4.11: The training loss of model with different level of decomposition and with/without unsupervised loss.

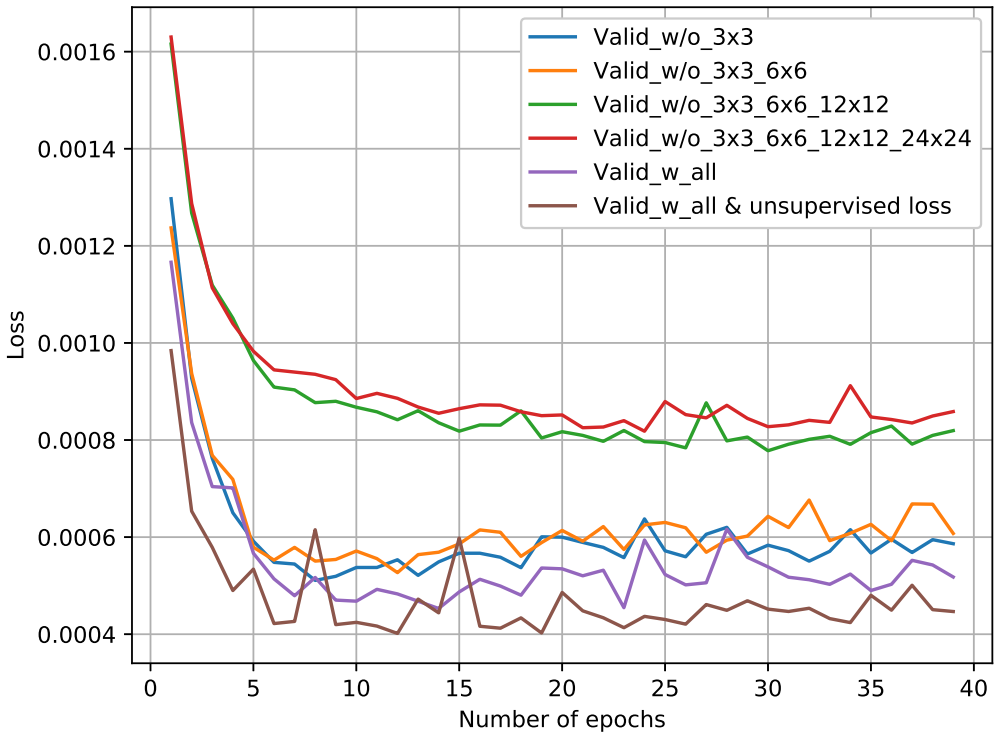


Figure 4.12: The validation loss of model with different level of decomposition and with/without unsupervised loss.

Fig. 4.12, the lowest reconstruction loss in both the training and validation sets is achieved when the Hr-MSI is decomposed into the maximum scales of five, of which the final scale has a spatial size equal to that of the Lr-HSI. Reducing one level of decomposition may result in performance degradation. The main reason is that each smaller scale of the image contains features to approximate the original image, and the early applying of the joint-training can further refine information in a coarse-to-fine manner. Although the Lr-HSI can not be decomposed further from the image of size 3×3, the results shown in Fig. 4.11 and Fig. 4.12 suggest that joint learning from the smallest levels would reduce the reconstruction error. Finally, the combination of both five-level decompositions and an unsupervised loss induced by the auxiliary task significantly outperforms all others after about only 10 epochs during the training or about 5 epochs during the validation. The

turbulences at 8 and 15 epochs indicate the outliers of the unsupervised features from the auxiliary task. Although they do not degrade the final performance, reducing the noise level in the auxiliary task or global learning rate can avoid these spikes.

Table 4.5 shows the testing results with and without the auxiliary task on the ICVL dataset. As seen, the introduced auxiliary task does improve the overall performance in both shallow and deeper networks. The accuracy, however, does not improve further while increasing the number of residual blocks. One possible reason here is that the lightweight model can sufficiently fit with the 75 training images, thus increasing the depth of the model can not produce further improvement.

Table 4.5: Average performance of the Baseline network (without the proposed auxiliary task) and MSAT (with the auxiliary task) over testing images of the ICVL dataset.

Method	**RMSE↓**	**ERGAS ↓**	**SAM↓**	**SSIM↑**
Baseline ($\beta = 1$)	1.37±0.45	0.086±0.043	1.043±0.327	0.994±0.0012
MSAT ($\beta = 1$)	**1.15±0.34**	**0.072±0.035**	**0.998±0.314**	**0.995±0.0010**
Baseline ($\beta = 2$)	1.26±0.33	0.079±0.038	1.041±0.347	**0.998±0.0005**
MSAT ($\beta = 2$)	**1.03±0.32**	**0.065±0.035**	**0.990±0.306**	**0.998±0.0005**

To further demonstrate the effectiveness of multi-scale reconstruction, comparisons with other CNN-based methods, such as SRCNN [11] and VDSR [45] are included, where pre-upsampling is used. The SRCNN [11] model has only 3 simple convolutional layers, while the VDSR [45] contains 20 convolutional layers. In addition, experiments with a more powerful architecture based on the ResNet, namely HSI-ResNet, are re-conducted with the same configurations as the ResNet, including the number of blocks, optimization method of network training, epoch number, training and testing samples, etc. The HSI-ResNet does not fuse Lr-HSI and Hr-MSI at multi-stages as it has done in the proposed MSAT

model. As shown in Fig. 4.13, the Lr-HSI is spatially upsampled before concatenated with the Hr-MSI. The CNN network consists of five residual blocks, which has a similar depth as the proposed model. Table 4.6 illustrates that progressive fusion at multiple stages has an obvious advantage over single-stage fusion.

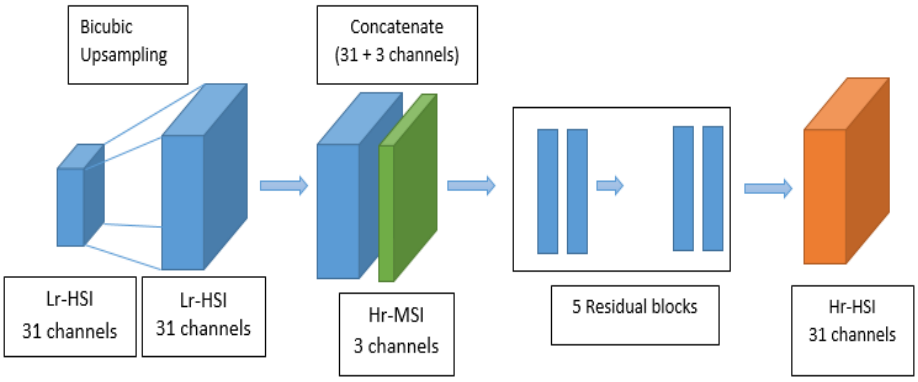


Figure 4.13: HSI ResNet model.

Table 4.6: Quantitative results on CAVE dataset. Baseline model indicate that the proposed model do not include auxiliary task.

Method	**RMSE↓**	**ERGAS ↓**	**SAM↓**	**SSIM↑**
SRCNN [11]	4.32±2.22	0.54±0.41	6.17±1.47	0.961±0.018
VDSR [45]	4.14±2.15	0.49±0.37	5.98±1.36	0.970±0.014
HSI-ResNet	3.96±1.87	0.44±0.31	5.33±1.20	0.977±0.007
Baseline	**3.45±1.69**	**0.38±0.25**	**4.81±1.07**	**0.981±0.006**

B. Tuning the noise level in denoising autoencoders

Several denoising autoencoders are trained with different noise levels to understand the qualitative effect of the noise across different datasets. The variation of RMSE, ERGAS, SAM, and SSIM values when varying the noise levels from 0.0 to 0.3 for CAVE, Harvard, ICVL, Chikusei, and Roman Colosseum datasets are shown in Fig. 4.14. As can be seen in Fig. 4.14, with the increased level of noise, the performance metric also begins to improve, which may become plateau

and then degrade for all datasets. The appropriate noise levels were discovered to be dependent on the quality of collected images as well as the number of training samples, which may affect the training performance and the model accuracy. Adding a large amount of noise to noisy images could degrade the performance. The images in the CAVE dataset, for example, are clean and contain fewer noises than those in the Harvard and the ICVL datasets. Therefore, applying a large noise level ($\sigma = 0.2$) leads to improving performance for the CAVE dataset, while increasing errors for the Harvard and the ICVL. As the training set for the Chikusei and Roman Colosseum datasets is limited, only the top part of an image is used, the smaller noise level of 0.05 is the most appropriate.

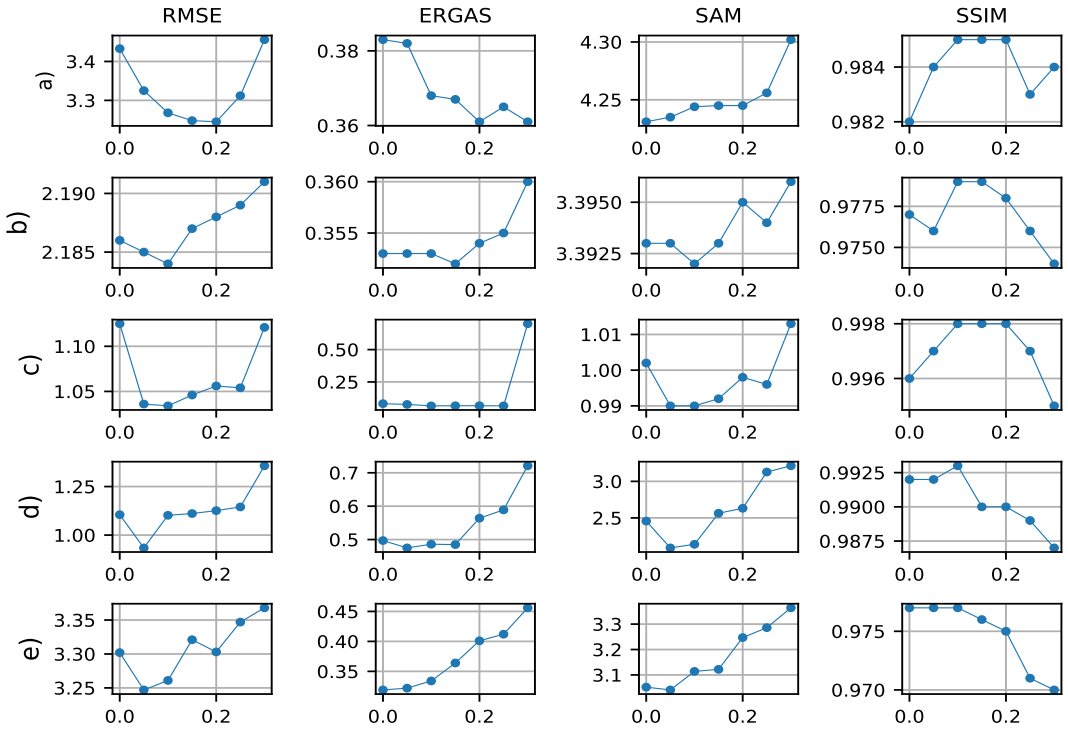


Figure 4.14: The variation of RMSE, ERGAS, SAM, and SSIM with the noise levels σ in the denoising autoencoder for five datasets. (a) the CAVE. (b) the Harvard. (c) the ICVL. (d) the Chikusei. (e) the Roman Colosseum. We select $\sigma = 0.2$ for the CAVE dataset, $\sigma = 0.1$ for both Harvard and the ICVL datasets, $\sigma = 0.05$ for both Chikusei and the Roman Colosseum, respectively.

C. Noise Robustness

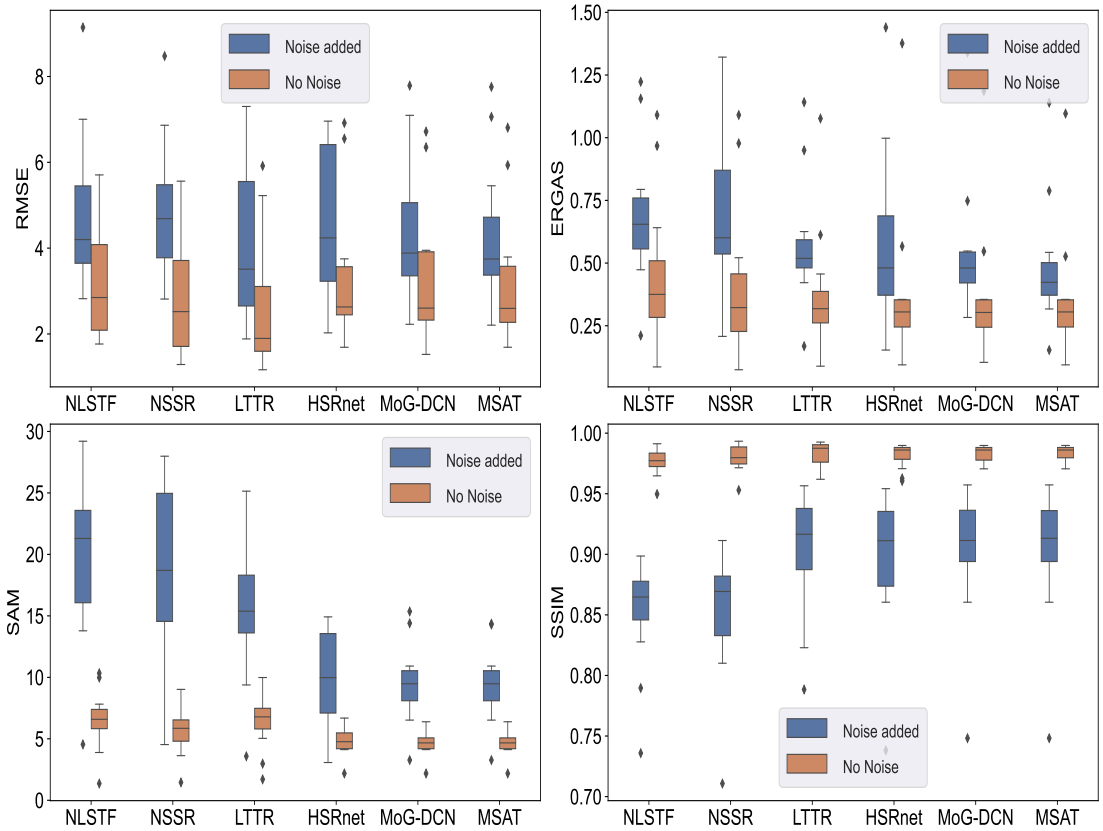


Figure 4.15: Quantitative result of noisy cases on CAVE testing set.

In practice, noise from various aspects can corrupt the Lr-HSIs and Hr-MSIs during image acquisition, transmission, and compression. To test the robustness against noise of all compared methods, the Gaussian noise is added to the Lr-HSI and Hr-MSI inputs and then fuse them to produce a Hr-HSI. The SNRs of the noisy Lr-HSI and Hr-MSI are set to 20dB and 25dB, respectively. The quality metric values in the noisy cases are shown in Table 4.7 and visually compared with those noise-free ones (as referred to Table 4.1) in Fig. 4.15. As seen, the performance of NLSTF [67], NSSR [65], and LTTR [69] methods drops faster than three deep learning-based methods in all four metrics and degenerates sharply in the SAM measure. The RMSE of the LTTR [69] increases from 2.640 ± 1.590

to 4.064 ± 1.913 by 53.9% ± 20.3% while the proposed approach is more robust, increasing only from 3.245 ± 1.610 to 4.282 ± 1.712 or by 31.9% ± 6.1%. The architecture of the MoG-DCN contains autoencoders that are robust to noise. The RMSE of the MoG-DCN [59] increases from 3.330 ± 1.676 to 4.390 ± 788 by 31.9%±6.6%.

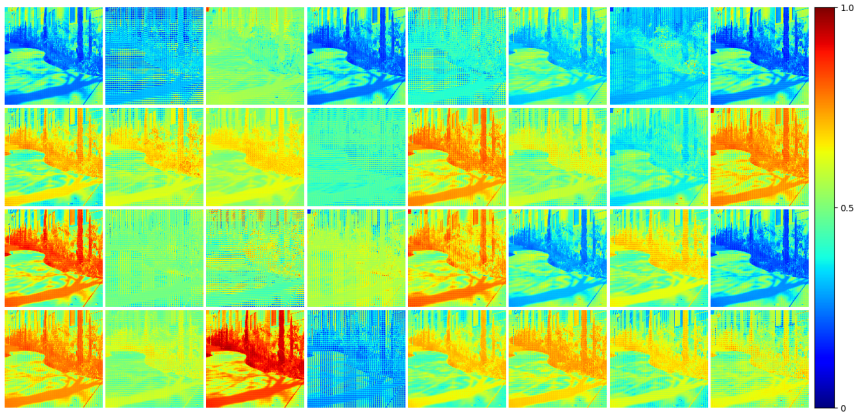
Table 4.7: Quantitative results of a noisy case on the CAVE dataset.

Method	**RMSE↓**	**ERGAS ↓**	**SAM↓**	**SSIM↑**
NLSTF [67]	4.81 ±1.87	0.70 ±0.28	20.07±7.07	0.851±0.047
NSSR [65]	4.90±**1.64**	0.71±0.33	19.06±7.09	0.850±0.053
LTTR [69]	**4.06**±1.91	0.58±0.26	15.62±5.81	0.902±0.051
HSRnet [60]	4.58±1.85	0.58±0.35	10.08±3.93	0.894±0.058
MoG-DCN [59]	4.39±1.79	0.55±0.27	9.53±3.25	**0.902**±0.056
MSAT	4.28±1.71	**0.49±0.25**	**9.44±3.09**	**0.902±0.050**

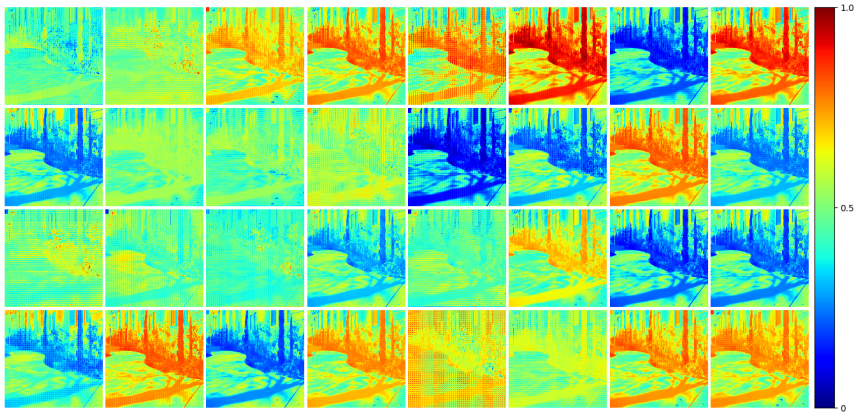
D. Feature map

Differing from RGB images, HSIs have the characteristics of high spectral resolution across many narrow bands. Therefore, it is not straightforward to interpret the meaningful feature maps at the lower layers, which typically display features in a spatial manner. To visualise the features learnt from the proposed CNN-based network, one testing image is selected from the CAVE dataset, followed by a forward path to show the learnt feature maps from the fifth (top) block in Fig. 4.16. It is worth noting that the transposed convolutions are used when up-sampling the input feature map at each stage. This is a well-known operation that may introduce severe checkerboard artifacts and tends to be most prominent with a higher up-sampling scale factor [114]. The checkerboard pattern can be observed in the feature maps of Fig. 4.16 (b) and Fig. 4.16 (c), where they have shown that the feature maps extracted from the model without the unsupervised loss will suffer more from horizontal and vertical stripes in the final prediction. By contrast, the feature maps from the model with the proposed additional unsupervised loss can successfully suppress such artifacts.

(a) An example of RGB image *bgu-0403-1523* from the ICVL dataset.



(b) Feature maps from 32 channels learned by fifth block without unsupervised loss. Each channel has the size of 96 x 96 pixels. Feature maps at (row, column) (1,2), (1, 7), and (4, 4) still suffer checkerboard artifacts.



(c) Feature maps from 32 channels learned by fifth block with unsupervised loss. Each channel has the size of 96 x 96 pixels. Only feature map at (row, column) (4, 5) has a checkerboard artifact.

Figure 4.16: Visualization of feature maps learned by the fifth block of the reconstruction network: (a) 3 channels of the observed RGB image; (b) Without using the proposed unsupervised auxiliary loss. (c) Using the unsupervised auxiliary loss.

4.4 Summary

This chapter has presented an effective CNN-based method for fusing the observed Lr-HSI and Hr-MSI to reconstruct high-resolution HSI. By decomposing the Hr-MSI into multiple spatial scales, the discrepancy in spatial resolution between the observed Lr-HSI and Hr-MSI is facilitated, and the spectral features from Lr-HSI can fuse with spatial-reduced features from Hr-MSI to reconstruct high-resolution HSI in a coarse-to-fine manner. In addition, the primary task is integrated with a proposed auxiliary task to form a multi-task learning framework, which can help to reduce overfitting and improve the generalisation capability of the main task. By using a denoising autoencoder for the auxiliary task, our model is naturally more robust to noise presented in the image than all other methods tested. The testing results on five public datasets have demonstrated that the proposed method can provide improvements over the state-of-the-art methods in terms of both objective assessment and subjective visual quality. In future research, automatic and adaptive determination of balance between the primary task and the proposed auxiliary task will be explored. In addition, a natural progression of this work is to investigate other auxiliary tasks for improving the performance of the primary task.

Chapter 5

Generative Adversarial Networks-based Super-Resolution

5.1 Introduction

Since Goodfellow presented the Generative Adversarial Network (GAN) [26], it has made significant progress and has been deployed to a variety of applications, including image in-painting [156], image super-resolution [78], style transfer [157–160], and image editing [161, 162]. Researchers are still working on ways to improve GANs, particularly for strategies to address GANs' problems of mode collapse and instability. Training a GAN-based model is difficult owing to three significant issues: non-convergence, vanishing gradient, and mode collapse.

These three major problems in GAN training arise from the fact that the discriminator always tends to learn better than the generator. The optimal discriminator does not provide informative feedback for the generator to make progress. Stabilising the GANs in training can be accomplished in a variety of ways, mostly by the choice of architectures [50, 86], loss functions [88, 90], nor-

malisation [95] and regularisations [27–29, 163]. There have been several naïve ways to balance competition between the discriminator and the generator. The simplest and most effective way is to guarantee that the discriminator and the generator have a symmetric architecture. Another simple approach is to update the discriminator k time for every update of the generator. However, neither approach is adequate for the complex training scenario of GANs. The spectral normalisation [95] and gradient penalty [27, 29] methods for imposing the Lipschitz regularisation on the discriminator have shown great success in stabilising the training of GANs. Among the gradient penalty methods, the zero-gradient and coupled gradient penalties are the most widely discussed. In this chapter, we will particularly study the unstable behaviour of the zero-centre gradient penalty proposed in [29]. Although this gradient penalty method has improved performance, there is still a limitation with a pre-defined gradient penalty strength. Concretely, setting a fixed penalty weight cannot avoid the overfitting of the discriminator through training and also leads to an over-penalised model even when the overfitting problem is not detected.

The contribution made here is the development of a parameter-free adaptive schedule that increases the resilience of the generator by adaptively adjusting the strength of regularisation on the discriminator. The regulation strength is adjusted based on the relative change of the training loss between generator and the discriminator, hence enabling the generator to catch up with the discriminator. When compared to the original regularisation [29] using a predefined schedule, introducing a dynamic schedule can improve both convergence and generality. The proposed schedule will improve the performance of various GAN-based applications where there are significant differences in features between generator's input and groundtruth, such as image generation [26], image-to-image translation [157, 159, 164], text-to-image translation [165, 166], photograph editing [167, 168], and photo inpainting [169, 170]. For those applications, the

discriminator is more likely to perform better than the generator. In the image super-resolution area, if the input of the generator is a low-resolution image instead of noise, then the imbalance between the generator and the discriminator is not significant. However, the benefits of the proposed method can still be seen in several aspects. First, when the up-scaling factor is large, for example, 256 times in [94], the unstable training still suffers. Second, the degradation and downsampling kernels are unknown in a real-world scenario, GAN-based model can be used to learn how to degrade and downsample a high-resolution image [171], estimate unknown degradation/downsampling kernels [81, 82]. In other words, improving the pure GAN will support a super-resolution approach to real-world problems where degradation factors and ground truth are unavailable.

The remaining parts of this chapter are organised as follows: Section 5.2 revisits the gradient regularisations and presents the proposed approach. Section 5.3 shows the experimental results, including an ablation study and discussion. Some concluding remarks are drawn in Section 5.4.

5.2 Adaptive method for gradient penalty

5.2.1 Gradient penalty

The objective of classical GANs [26] is given by the following minimax objective function:

$$\min_G \max_D \mathcal{L}(D,G) = \min_G \max_D \mathbb{E}_{x\sim p_d}[\log(D(x))] + \mathbb{E}_{z\sim p_g}[\log(1-D(G(z))] \quad (5.1)$$

which can be achieved by an alternatively training G and D, presented in Eq. (5.2) and Eq. (5.3), respectively.

$$\max_D \mathcal{L}(D) = \max_D \mathbb{E}_{x\sim p_d}[\log(D(x))] + \mathbb{E}_{x\sim p_g}[\log(1-D(x))] \quad (5.2)$$

$$\min_G \mathcal{L}(G) = \min_G \mathbb{E}_{x\sim p_g}[\log(1 - D(G(x))] \tag{5.3}$$

The quality of the image output by G depends on the gradients that receive from D, which is

$$\nabla_x \mathcal{L}(D, G) = \underbrace{\nabla_{D(x)} \mathcal{L}(D, G)}\underbrace{\nabla_x D(x)} \tag{5.4}$$

The first term on the right-hand side of Eq. (5.4) is defined by the GAN objective, or the choice of distributional divergence. The second term is the gradients of output $D(x)$ w.r.t input x. As previously mentioned in Sec 5.1, the stability of GAN training is dependent on the choice of the loss function represented by the first term, normalisation and regularisation, according to the second term. As the discriminator is generally observed to be too powerful for the generator, the gradient may not be reliable. One regularisation strategy is to employ gradient penalty to limit the discriminator's modelling capabilities. The common gradient penalties for discriminators take the following form:

$$\max_D \mathbb{E}_{x\sim p_d}[\log(D(x))] + \mathbb{E}_{x\sim p_g}[\log(1 - D(x))] + \lambda \mathbb{E}[R(\parallel \nabla_v D(v) \parallel_2] \tag{5.5}$$

where $\lambda \in \mathbb{R}$ is the weight of the gradient penalty term, R is the real function, and v is the sample point where the gradient is computed w.r.t. Table 5.1 shows the details of the distribution and function R used in the common gradient penalty. Such gradient penalties can be broadly classified into two categories: 1-GP, where the gradient penalty regulariser aims to enforce 1-Lipschitz continuity; and 0-GP, where the regulariser forces the gradient norm to be zero. Details of the two categories are discussed as follows:

1-GP. As the gradient penalty is computed with specific samples, the 1-GP is imposed at different points in WGAN-GP [27] and DRAGAN [28]. The WGAN-GP forces the norm of gradients w.r.t. points at an interpolation between the real images and the faked images to be 1. The DRAGAN penalises the norm

Table 5.1: The property of different gradient penalties for general GANs.

GP	$\mathcal{L}_{\mathcal{GP}}$	v	Lipschitz continuity
1-GP [27]	$\lambda\mathbb{E}[(\parallel \nabla D_v \parallel_2 -1)^2]$	$(1-\alpha)x + \alpha\hat{x}$	$\mathrm{K} \to 1$
1-GP [28]	$\lambda\mathbb{E}[(\parallel \nabla D_v \parallel_2 -1)^2]$	$x + \epsilon$	$\mathrm{K} \to 1$
0-GP [29]	$\lambda\mathbb{E}[(\parallel \nabla D_v \parallel_2)^2]$	x or $\hat{x}$	$\mathrm{K} \to 0$
0-GP [172]	$\lambda\mathbb{E}[(\parallel \nabla D_v \parallel_2)^2]$	$(1-\alpha)x + \alpha\hat{x}$	$\mathrm{K} \to 0$

of gradients w.r.t. points around the data manifold. Although WGAN-GP has been successful, imposing regularisation on the space outside the support of the generator and data distribution may hinder the convergence. Obviously, the interpolated point of two real images is often not a real image. Also, 1-GP allows the gradient norm to be smaller than 1 in some regions and larger than 1 in others.

0-GP. In [29], it is reasonable to argue that the zero-centered gradient penalty makes the GAN training converge. However, improperly applying 0-GP can result in worse convergence. Let X and Ω be the domain and the range of a neural network D with a scalar output, respectively (i.e., $\mathrm{f}: \mathrm{X} \to \Omega$). Over-penalising or continuously penalising would restrict the output of f to a very small interval. At this point, the gradient penalty term is also zero, as long as the generator does not produce samples whose output by D still remains within that interval. This proposed 0-GP is highly aimed at convergence while providing less support to the generator, which aims to learn features and fool the discriminator. Unfortunately, the 0-GP regulariser does often lead to over-penalising and is unstable for long training. Unless the generator improves through training, the strength of GP is hard to reduce. The size of label data is finite, while the generator takes noise inputs of infinite size. Therefore, the discriminator will always go ahead of the generator, and the gradient penalty will be imposed continuously (see Fig. 5.1). The quality of generated images starts to improve from the beginning of training, reaches the plateau with the best quality, and then becomes worse.

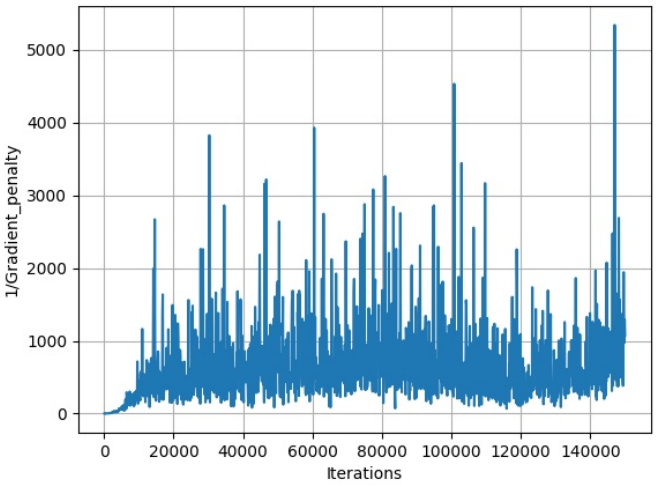


Figure 5.1: The $\frac{1}{gradient\ penalty}$ value have shown that the strength of gradient penalty keep increasing during the training.

5.2.2 Parameter-free dynamic schedule for gradient penalty

In Eq. (5.4), the gradient penalty is always proportional to the error distance of the discriminator given real and fake examples when λ is fixed. Unfortunately, the error distance is usually high at the start of training and decreases later, and the loss curve has complicated local minima and maxima. As a result, fixing the trade-off λ will not be well adapted and will run into a gradient over-penalising problem and even overfitting of the discriminator. This section presents an Adaptive 0-GP schedule, which iteratively selects strength λ for adaptively mitigating these problems. The objective of the discriminator is presented as:

$$\min_{D} \min_{\lambda} \mathbb{E}_{x\sim p_d}[\log(D(x))] + \mathbb{E}_{x\sim p_g}[\log(1-D(x))] + \lambda \mathbb{E}[R(\parallel \nabla_{x\sim p_d} D(x) \parallel_2] \quad (5.6)$$

However, this is a min-min problem and is computationally infeasible to solve. Therefore, a common approach is to find an approximate solution. The proposed method here is to find the smallest λ at the current point. This can be done by reducing the strength of GP when the generator learns well and increasing the

strength of regularisation when the discriminator goes far ahead of the generator.

The outputs of the discriminator for the real and the generated samples are evaluated to determine if the generator has caught up with the discriminator. The trust region algorithm [173] is employed to decide the strength of the gradient penalty based on the predetermined threshold. Let $\mathbf{f}(\mathbf{x}) = D(x, \theta)$ be the output of discriminator before taking into account the cross-entropy. It is worth noting that increasing λ would lead to tighten Lipschitz constant and $\mathbf{f}(\mathbf{x})$ getting closer to $\mathbf{f}(\mathbf{x} + \xi)$ for arbitrary small ξ. The fact that the penalty is applied to real samples does not imply that the K-Lipschitz is satisfied at any generated sample. The original trust region methods define a region in which they trust the model to be an appropriate approximation of the objective function and then choose the step to be the approximate minimiser of the model within this trust zone. Let $f_{real}(x)$ be the objective function $x \in p_d$ and $f_{fake}(x)$ denote a model with $x \in p_g$, with latter's minimiser attempting to be an adequate representation of the former. The purpose of the dynamic scheduler is to iteratively select the smallest value of λ when $f_{fake}(x)$ is accurately approximated by $f_{real}(x)$. To define the trust region, let $dist(f^i_{real}, f^i_{fake}) = f^i_{real} - f^i_{fake}$ denote the distance between two functions at step i. The strategy for choosing λ is based on the trust region radius at step i. This choice can be based on the agreement between the model function f^i_{fake} and the object function f^i_{real} :

$$\rho^i = \frac{f^i_{real} - f^i_{fake}}{f^{i-1}_{real} - f^{i-1}_{fake}} \tag{5.7}$$

The numerator and the denominator in Eq. (5.7) are called *predicted reduction* and *actual reduction*, respectively. Instead of using an intermediate output of the discriminator f_{real} and f_{fake} at step i, we define f^i_{real} and f^i_{fake} as an exponential moving average of each output until step i, which is usually less noisy. Note that the discriminator attempts to maximise the distance between f^i_{real} and f^i_{fake}

when the use of 0-GP is to minimise that distance. The value of λ at step i-1 will determine the value of ρ^i in Eq. (5.7).

Generally, if the f^i_{fake} is a poor approximation of the f^i_{real}, the λ is increased and vice versa. Concretely, if ρ^i is greater than σ_1 ($\sigma_1 > 1$), this indicates that f^i_{fake} is moving far away from f^i_{real}, then λ will be increased by the factor of $\eta^+ > 1$ ($\lambda^{i+1} = \eta^+\lambda^i$) for the next iterations. If, on the other hand, ρ^i is less than σ_2 ($\sigma_2 < 1$), this means that f^i_{fake} and f^i_{real} are well matched, the coefficient λ will be reduced by the factor $\eta^- < 1$ ($\lambda^{i+1} = \eta^-\lambda^i$) for the subsequent iterations. Otherwise, when $\sigma_2 < \rho^i < \sigma_1$, then the λ^i is kept the same for λ^{i+1}. Using this approach, the dynamic schedule can iteratively find the appropriate λ to mitigate overfitting and over-penalising of the discriminator. It is important to note that two consecutive reductions of λ still guarantee that the generator can be improved, but it is not the case for two consecutive rises of λ. The latter will cause the over-penalising problem. To avoid this case, λ will be bound by a constant value λ_{max}.

Note that $\rho^i > \sigma_1$ or $\rho^i < \sigma_2$ each can be divided in two cases. If $\rho^i > \sigma_1$, the two cases are: (1) f^i_{fake} and f^i_{real} are both going far away from the equilibrium; and (2) the speed of f^i_{real} moving away from equilibrium is faster than the speed of f^i_{fake} getting close to the equilibrium. The λ will only be increased for the latter by including the second condition $f^i_{fake} < f^{i-1}_{fake}$. Finally, rather than using arbitrary values for η^+ and η^-, we chose $\eta^+ = \eta^- = \rho^i$. When $\rho^i > \sigma_1$ then $\eta^+ = \rho^i > 1$. In contrast, when $\rho^i < \sigma_2$ then $\eta^- = \rho^i < 1$ (note that we have $\sigma_2 < 1 < \sigma_1$). Using this setting, λ can be precisely adjusted based on the magnitude of ρ^i.

The algorithm 1 describes the schedule that adaptively adjusts the strength of 0-GP regularisation. The algorithm 2 describes the overall training update with GAN that includes Algorithm 1 as a procedure.

Algorithm 1 Dynamic Schedule of Adjusting Regularisation Strength of 0-GP

Initial gradient penalty weight λ_0, fix number of iterations T (typically, $T = 1000$) to adjust λ, threshold $0 < \sigma_2 < 1 < \sigma_1$.
while discriminator updating **do**
    for every T iterations **do**
        Calculate ratio ρ^i from 5.7
        if $\rho^i > \sigma_1$ and $f^i_{fake} < f^{i-1}_{fake}$ **then**
            $\lambda^{i+1} = \min\{\rho^i \lambda^i, \lambda_{max}\}$
        else if $\rho^i \leq \sigma_2$ and $f^i_{fake} > f^{i-1}_{fake}$ **then**
            $\lambda^{i+1} = \rho^i \lambda^i$
        else
            $\lambda^{i+1} = \lambda^i$
        end if
    end for
end while

Algorithm 2 Minibatch stochastic gradient descent training

Initialization: Number of iteration M, number of iteration T for updating λ, minibatch size **m**, and variables in Algorithm 1.
for i = 1,...,M **do**
    while discriminator updating **do**
        Sample minibatch of **m** real examples from p_r
        Sample minibatch of **m** latent variables from prior p_g
        Update the discriminator by ascending its stochastic gradient
        for every T iterations **do**
            Update a balanced parameter with Algorithm 1
        end for
    end while
    Sample minibatch of **m** fake examples
    Update the generator by ascending its stochastic gradient
end for

5.3 Experimental results

5.3.1 Implementation details

To verify the effectiveness of the proposed dynamic schedule for GP method, the DCGAN [86] architecture is chosen as the backbone. This is mainly because the architecture of the DCGAN can propose stable training without using any regu-

larisation. The hyper-parameters are set according to the recommended default values. Concretely, both the generator G and the discriminator D models are implemented using convolutional neural networks (CNNs) with batch normalisation [101]. The Adam optimiser [142] is used to train both G and D with a learning rate of 2e-4 and momentum β_1 of 0.5 as default recommended in DCGAN [86]. The G and D are alternatively updated once in each iteration, and the GAN model is trained for 200,000 generator steps. All experiments are implemented on TensorFlow with CUDA 9.0 and cuDNN back-ends using a GPU, NVIDIA GeForce GT 1030. The batch size is set to 32, which is to balance the trade-off between performance and available GPU memory resources.

5.3.2 Experiments on synthetic datasets

The performance of GAN using different gradient penalties is tested on the three most widely used toy datasets [27]: i.e., 8 Gaussians, 25 Gaussians, and Swiss Roll. The datasets of 8 Gaussians and 25 Gaussians are generated using a mixture of 8 Gaussians and 25 Gaussians, respectively, with the modes that are uniformly distributed in a circle or in a grid. The contours of the generator's samples and discriminator's samples are displayed in varying and transparent colours, respectively. Fig. 5.2 shows that all 1-GP and 0-GP can guarantee the convergence and discover all the modes. With both 0-GP methods, the generated samples from one mode often do not lie in regions like those generated for other modes. The contours displayed in the third and fourth rows have shown that the Adaptive 0-GP model outperforms the conventional 0-GP model. The generated points from the Adaptive 0-GP can not only approximate the label data but also well separate each mode. Both 1-GPs cannot converge on Swiss Role toy data. They assign the same level set for all real data, which results in a high loss for the discriminator. The corresponding losses are presented in Table 5.2, which compares the performance of the discriminator with different gradient penalty methods. In

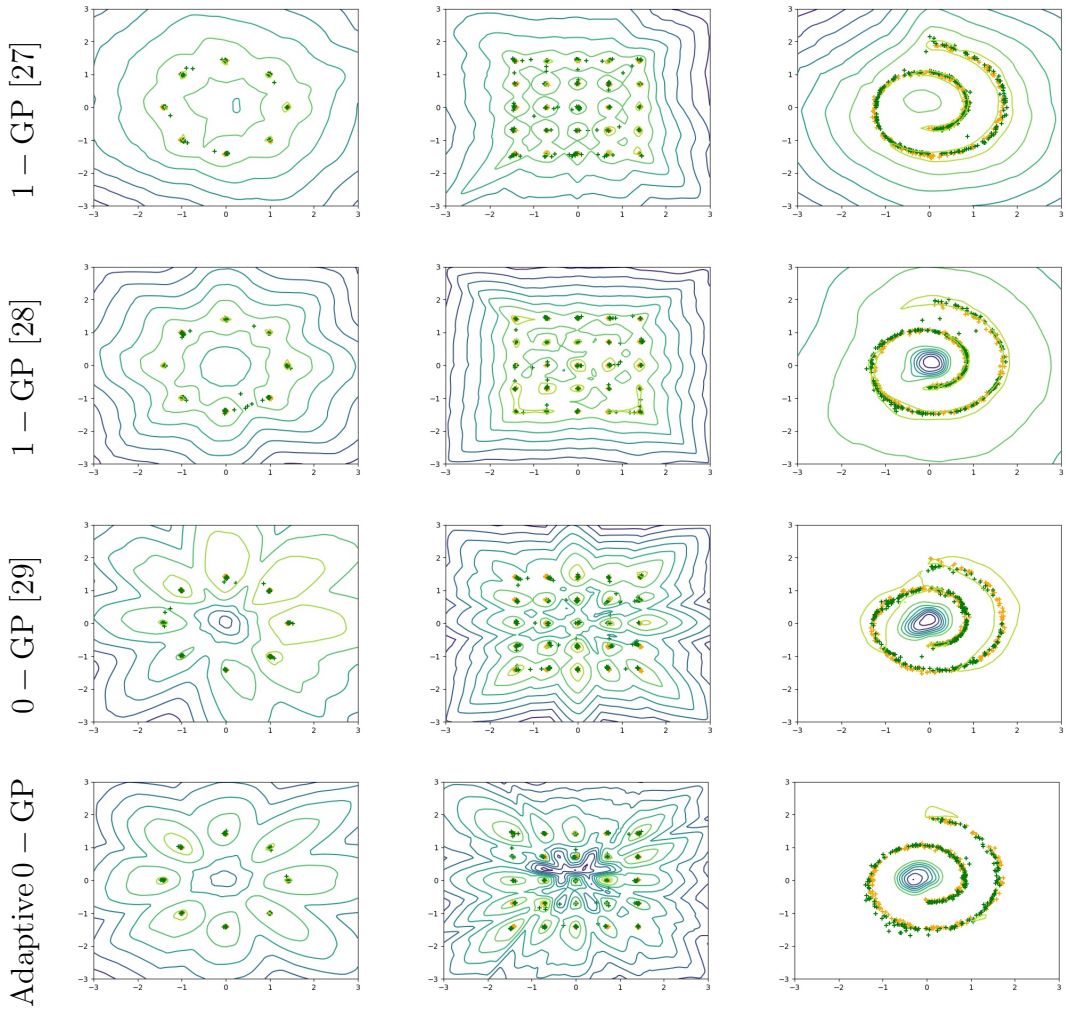


Figure 5.2: Discriminator contour patterns generated while training a model on 2D toy datasets. The orange points are samples from the true data distribution, the green points are samples from the generator distribution.

all three toy datasets, the 0-GP model cannot converge, given a fixed trade-off parameter λ. However, when combining it with an additional dynamic schedule, it can achieve convergence successfully.

5.3.3 Experiments on real datasets

Two publicly available datasets are used in the experiments, including CIFAR-10 [174] with 60,000 images from 10 object classes and CelebA [175] with more

Table 5.2: The losses of discriminators using different gradient penalty methods were evaluated on 8 Gaussians, 25 Gaussians, and Swiss Roll datasets.

	1-GP [27]	1-GP [28]	+ 0-GP	+ Adaptive 0-GP
8 Gaussian	0.6921	0.6930	0.690020	**0.693138**
25 Gaussian	0.6925	0.6930	0.692249	**0.693147**
Swiss-roll	0.6958	0.6964	0.693234	**0.693151**
Ideal equilibrium	0.693147	0.693147	0.693147	0.693147

than 200,000 face images from 10,177 celebrities. All the images of size 178 $\times$ 218 from the CelebA dataset are center cropped and resized to a small resolution of 64 $\times$ 64 to reduce the computation cost and model parameters. Additional details of two datasets can be found in Appendix A. For quantitative evaluation of generative models, the Frechet Inception Distance (FID) and the Inception Score (IS) (see Subsection 2.3.11 (C)) are utilised.

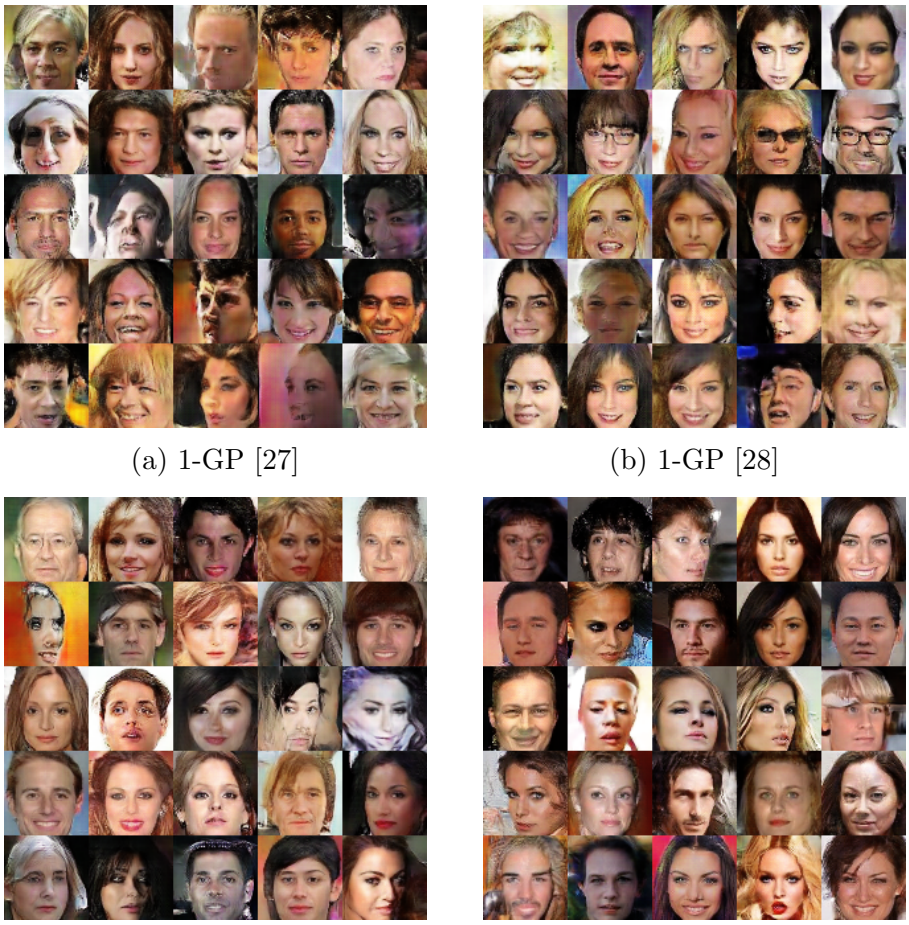
The visual results with different gradient penalty approaches on the CIFAR-10 and the CelebA datasets are shown in Fig. 5.3 and Fig. 5.4, respectively. The objects from images generated by Adaptive 0-GP are in more detail compared to those from all other gradient penalty methods. Both 1-GP methods perform similarly, roundly highlighting the objects against the background, but fail to generate detailed structures and fine object boundaries. We find that both 1-GP cannot converge with further training and the generated images have not shown better results. It seems both 1-GP methods perform worse in the real dataset than in the toy datasets. The main reason for the better result in the toy case is that both 1-GP methods impose on the data point, which is also Gaussian noise lying inside the training data distribution. However, in real image data, linearly interpolated data points between two images are not usually an image. Therefore, performing the gradient penalty on that interpolated image does not have the desired effect. The 0-GP and the Adaptive 0-GP have both produced clear visual results on the two datasets, while Adaptive 0-GP generates images

(a) 1-GP [27] (b) 1-GP [28]

(c) 0-GP [29] (d) Adaptive 0-GP [29]

Figure 5.3: Randomly generated 100 images by the generator at step 100K using different gradient penalty methods on a CIFAR-10

that are smooth and less distorted. This superior performance is mainly due to the utilisation of the proposed dynamic schedule in adaptively adjusting the penalty weight based on the training status, which has enhanced the stability of GAN training than using a fixed penalty weight. Fig. 5.5 compares the images generated from the same noise with the baseline 0-GP and the Adaptive 0-GP methods. The 0-GP baseline generates images with limited changes in facial



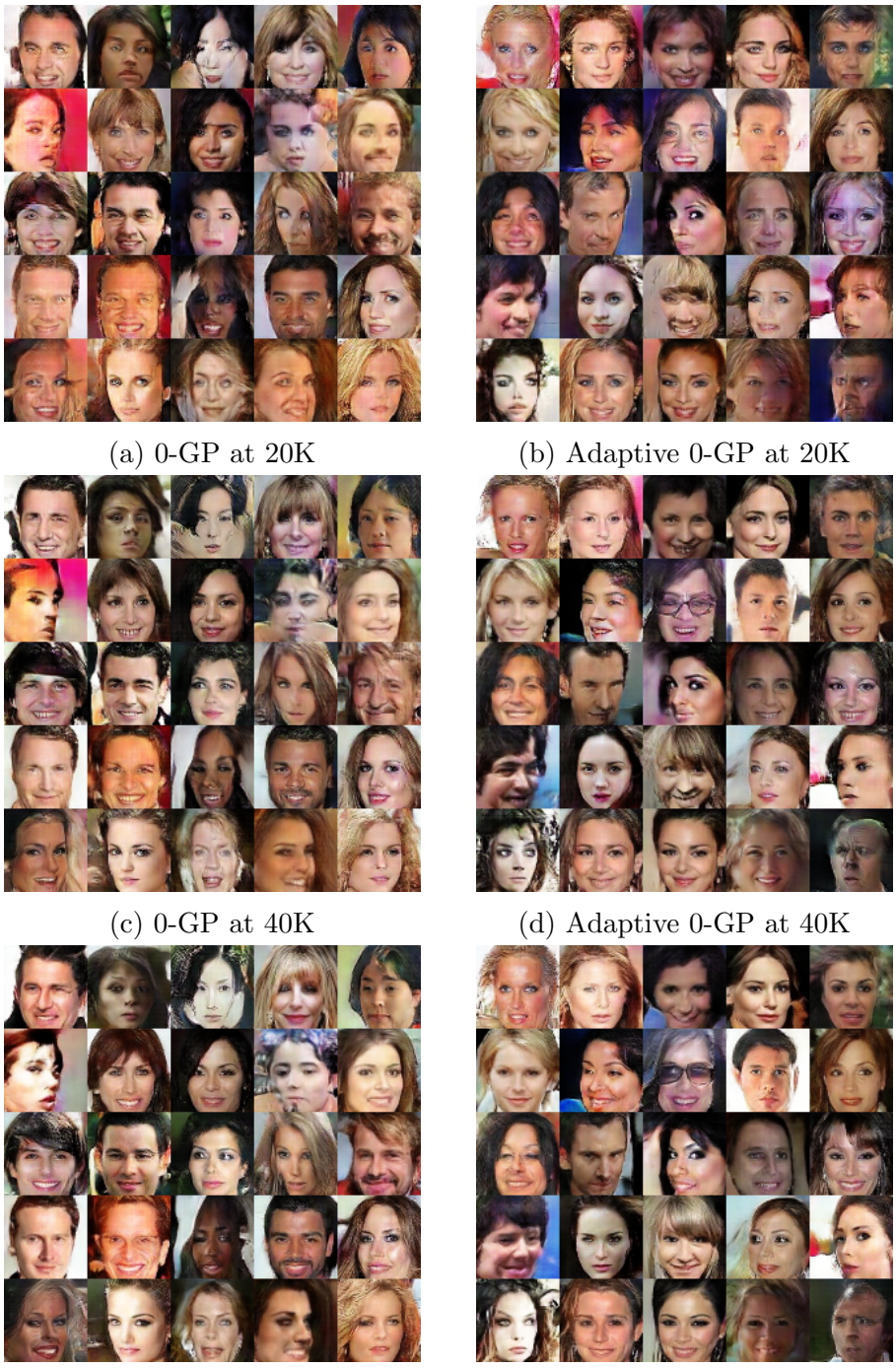
(a) 1-GP [27] (b) 1-GP [28]

(c) 0-GP [29] (d) Adaptive 0-GP [29]

Figure 5.4: Generated images at iteration 50K using different gradient penalty methods on a CelebA dataset.

attributes, and some images have been almost unchanged from iteration 20K to 80K. This indicates that the 0-GP method over-penalising model results in degrading the generality. The Adaptive 0-GP method has produced smoother and more diverse faces compared to its counterpart.

Table 5.3 shows the qualitative results of the compared gradient penalty methods on the CIFAR-10 and the CelebA. As seen, the Adaptive 0-GP surpasses three other gradient penalty methods in both datasets and metrics. The Adaptive 0-GP consistently outperforms the 0-GP in all cases.



(a) 0-GP at 20K (b) Adaptive 0-GP at 20K

(c) 0-GP at 40K (d) Adaptive 0-GP at 40K

(e) 0-GP at 80K (f) Adaptive 0-GP at 80K

Figure 5.5: Compare images generated from the same random noise vector using the 0-GP and the Adaptive 0-GP methods.

Table 5.3: FID (↓) and IS (↑) scores of different gradient regulation methods on CIAFR-10 and CelebA datasets. The DCGAN architecture [86] is used as a baseline model.

	CIFAR-10		CelebA	
Method	FID (↓)	IS (↑)	FID (↓)	IS (↑)
DCGAN	45.32	6.49	35.61	2.53
+ 1-GP [27]	31.94	6.98	29.93	2.68
+ 1-GP [28]	32.88	7.02	28.75	2.64
+ 0-GP [29]	29.23	6.97	26.38	2.84
+ Adaptive 0-GP [29]	**28.68**	**7.21**	**21.23**	**2.91**

Fig. 5.6 shows the result of image interpolation produced by 0-GP and the Adaptive 0-GP methods at different iterations. A series of noise vectors obtained by linearly interpolating between two noise vectors is propagated through the pre-trained models, and the resulting images in the image space are expected to show smooth transposition. In Fig. 5.6, the leftmost and rightmost images of each row are generated from two noises, and 8 middle images result from a series of linear interpolated noises. As seen, images from the 0-GP baseline are more distorted than those from the Adaptive 0-GP ones. The Adaptive 0-GP shows smooth transposition between the leftmost and rightmost images. The use of dynamic schedule helps to achieve greater image quality while maintaining the smoothness of image transposition.

The process of adapting gradient penalty weight in training is illustrated in Fig. 5.7. The weight is initially in a downtrend for about 150K iterations, then increases afterwards. As the size of the noise input can be considered indefinite, the weight is usually increased at the end, but it is more stable and less spiky compared to the first phase. In Fig. 5.8, the gradient norms are shown for the 0-GP and the Adaptive 0-GP methods that were trained on the CIFAR-10 dataset. The mean and variance of the gradient norm of the 0-GP method keep increasing with increased iterations. As a result, the generated images would worsen, and

(a) 0-GP at 20K

(b) Adaptive 0-GP at 20K

(c) 0-GP at 40K

(d) Adaptive 0-GP at 40K

(e) 0-GP at 80K

(f) Adaptive 0-GP at 80K

Figure 5.6: Interpolation of training examples on the CelebA 64 × 64 dataset. Both the 0-GP and the Adaptive 0-GP generate images with the same random noise vector.

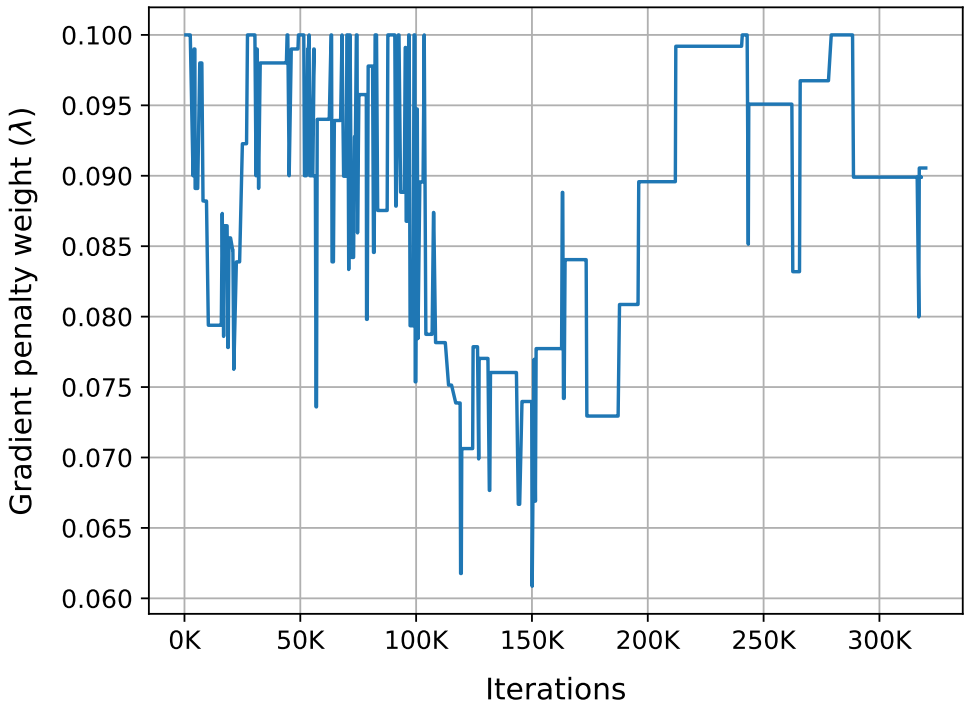
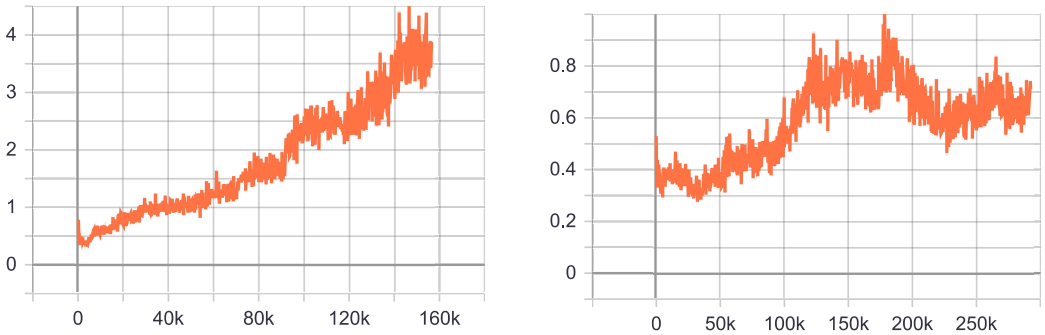


Figure 5.7: The gradient penalty weight is adapted through the training process.



(a) 0-GP method.

(b) Adaptive 0-GP method.

Figure 5.8: Gradient norms measured from 0-GP and Adaptive 0-GP methods with the CIFAR-10 dataset.

each object would not be clearly shown due to the mixing of the features of more than one object. This means the generator does not learn well to reduce the norm with a fixed λ. In contrast, the gradient norm from the Adaptive 0-GP is lower and does not follow an increasing pattern. As a result, the Adaptive 0-GP

provides more stable training and higher quality of generated images.

5.3.4 Comparison with predefined schedule methods

In order to compare the effectiveness of the proposed dynamic schedule and the predefined schedule in adjusting the strength of regularisation, two predefined schedules are defined, including (1) the λ is set to a fixed value; and (2) the λ is decaying annealed from an initial high value to a small non-zero value through the training, following the work in [176]. The λ value is initially set to 0.1, for which all schedules do not suffer from either overfitting or underfitting problems. From Fig. 5.9, it can be seen that the annealing schedule does not outperform the fixed schedule. The annealing schedule surpasses the fixed schedule from the beginning of training until iteration of 40K and then starts to perform worse than the fixed schedule. Due to the indefinite number of noise samples compared to a fixed

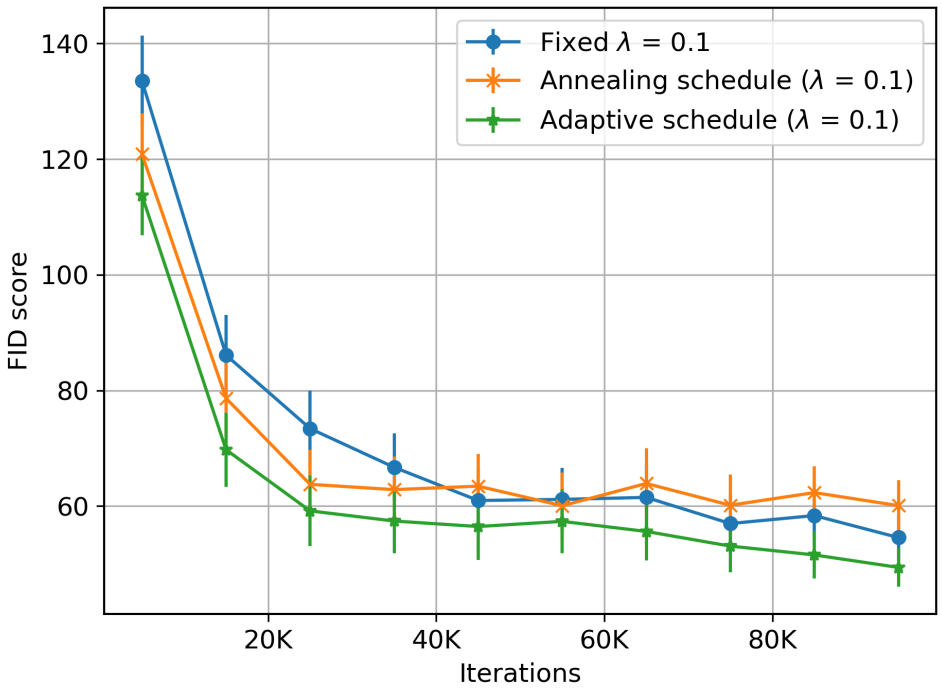


Figure 5.9: Comparison on FID score of various regularisation schedule.

number of real data samples, the error would gradually increase. Reducing the strength of regulation would, therefore, lead to overfitting of the discriminator. In other words, the annealing schedule is possibly effective when the size of the noise sample is limited. The variance of FID with the fixed schedule is the highest among the three compared schedules. Penalising with a high value of λ leads the model to converge to local minima, resulting in higher errors corresponding to incoming training samples. The dynamic schedule achieves the lowest FID score, consistently outperforming both the fixed schedule and the annealing schedule. Adaptively adjusting the strength of regulation can provide stable training and avoid both over-penalising and overfitting problems.

5.3.5 Ablation study on parameters of the Adaptive 0-GP

We study two key elements of the Adaptive 0-GP, i.e., the threshold σ_1, σ_2, and the magnitude η^+, η^-. The version used with our main experiment is denoted as the "main" version. The $\sigma_1 > 1$ and $\sigma_2 < 1$ are set as $1 + \tau$ and $1 - \tau$, where τ is arbitrarily small number ($\tau < 1$). The $\eta^+ > 1$ and $\eta^- < 1$ are empirically set to 1.1 and 0.9, respectively.

Table 5.4: Ablation study on thresholds and magnitudes of the gradient penalty of the Adaptive 0-GP train on the CelebA dataset.

	$\eta^+ = \eta^- = \rho$				$\eta^+ = 1.1,\ \eta^- = 0.9$			
Metric	$\tau = 0.1$	$\tau = 0.15$	$\tau = 0.2$	$\tau = 0.25$	$\tau = 0.1$	$\tau = 0.15$	$\tau = 0.2$	$\tau = 0.25$
FID (↓)	**21.23**	24.51	26.98	26.89	23.46	25.71	26.79	26.74
IS (↑)	**2.91**	2.76	2.80	2.85	2.77	2.76	2.77	2.83

As reported in Table 5.4, when τ is set to 0.2 and 0.25, the model with the Adaptive 0-GP and the one with original 0-GP perform similarly (see Table 5.3). When τ is 0.1 and 0.15, the Adaptive 0-GP achieves the best results with adaptive $\eta^+ = \eta^- = \rho$. Setting η^+ and η^- to fixed values requires more iterations

to adjust strength. This is mainly because the adjustment step is high when λ is high, which is less precise when λ is naturally increasing.

5.4 Summary

This chapter has addressed the instability of GAN training, focusing particularly on the gradient penalty [29]. The drawback of 0-GP [29] is that it often over-penalises the model in order to achieve convergence. As a result, the model converges to local minima, resulting in increasing errors after a long time of training, thus degrading the generality. Furthermore, setting a penalty weight to a fixed value cannot efficiently prevent the overfitting of the discriminator through training. A dynamic schedule for adjusting the strength of the gradient penalty is a new approach to tackling these problems. The variant of trust-region methods is first developed by defining a region around the current point in which the agreement between the generator and the discriminator is measured. If the generator is catching up to the discriminator, the dynamic schedule would reduce the strength of the gradient penalty to avoid the over-penalising problem. When the discriminator starts to overfit, it adaptively increases the magnitude of the gradient penalty to mitigate this problem. By adaptively selecting the proper strength of regularisation, the informative feedback from the discriminator to the generator is maintained. The quantitative and visual results have shown that combining 0-GP and a dynamic schedule can outperform the model that uses 0-GP with a pre-defined schedule. Finally, this proposed method reduces the sensitivity of the choice to a penalty weight and does not require additional computation.

Chapter 6

Discussion, Conclusion and Future Work

6.1 Advantages and disadvantages of deep learning for super-resolution problems

Deep learning-based methods for super-resolution have several advantages and disadvantages compared to conventional methods. Two major advantages of deep learning-based models are reconstruction accuracy and efficiency. The ability to learn from massive amounts of data and identify complex patterns is the key to their accuracy. The efficiency of deep learning-based models is considered in two aspects. At first, there is no need for feature engineering, which is really burdensome work for engineers dealing with large datasets. Instead, the end-to-end mapping scheme will adjust the kernels to enable choosing the relevant features for a given task. Secondly, the architecture of deep learning is reusable, which means one architecture can apply to many scenarios and types of problems. For example, EfficientNet [177] and U-Net [147] have shown great architectures for classification and segmentation, respectively, but they share the same pattern with super-resolution in terms of transforming features from one space to another.

Therefore, these architectures have been beneficial for super-resolution tasks as well.

Besides its advantages, a deep learning-based model has faced several challenges for SR tasks. First, the prominent challenge of DL is that it requires an extensively large amount of data to achieve good performance. Some imaging datasets are not always available to collect; for example, hyperspectral images are discussed in Chapter 4. Second, one of the most common criticisms of deep learning is its black box behavior. In recent years, although some research has been proposed to understand how the model works, interpretability is still the biggest challenge for deep learning-based methods. The output is usually the result of interactions between thousands or millions of parameters. Therefore, it is difficult to explain the learned representations, individual predictions, and model behavior, and hence improve the model's performance. Third, choosing the complexity of the model is another challenge, and it is difficult to find the sweet spot between over-fitting and under-fitting. To achieve the best performance, the complexity of the model must be appropriate to the complexity of the data. The common solution for overfitting is using regularisations to restrict the capacity of the model, and the solution for underfitting is increasing the model's complexity. Fourth, as deep learning-based models are trained based on gradient descent, the vanishing and exploding gradient problems exist for all tasks. Fortunately, with the choice of architecture and regularisation, these two problems have been addressed. Fifth, the model, like CNN for SR, is incapable of multitasking. Models can only perform targeted tasks and process data on which they are trained. They perform worse if there is a discrepancy between the distributions of training and testing data. Finally, all deep learning-based methods require memory and computational resources, which are frequently limited on mobile devices.

6.2 Conclusion

The main objective of this thesis was motivated by the rapid development and the rise in demand for effective deep learning-based SR approaches. The contributions cover a wide range of super-resolution areas, including single image super-resolution, the fusion-based hyperspectral image super-resolution, and GAN-based image super-resolution.

Producing a high-resolution image from one or multiple LR images is always a challenging problem and has been intensively investigated for decades. Recently, with the rapid growth of deep learning-based techniques, the implementation of deep learning-based SR models has been a great success and has achieved state-of-the-art performance on various benchmarks. However, the problem remains unsolved, particularly due to the ill-posed problem of image SR and the difficulty in training the deep learning models. There are always multiple HR images corresponding to a given low-resolution image, the LR and HR examples of experiments are generated by a pre-defined degradation. In the real world, the quality of LR images is affected by a variety of factors, such as the image system and imaging conditions, resulting in much more complicated and unknown degradation of LR images. The performance of algorithms trained on artificial pairs would certainly suffer when being applied to actual LR images, due mainly to the significant disparity between the real and artificial LR images. The difficulties in applying deep learning-based methods involve both general and particular aspects. In general, the deep CNNs are considered black boxes, as it is difficult to comprehensively understand and explain the behaviour of these networks, for example, how neurons interact with each other to make decisions, especially when the connections are complicated.

Various contributions have been proposed in this thesis to address these problems as well as their limitations, which are summarised below.

1. In Chapter 3, a new highway connection for CNN-based SR architecture is proposed to replace the commonly used local skip connection. With the design of the attention mechanism, each block in the proposed SR architecture can pay attention to more detail of the information it should retain and forget rather than the binary decision in the ResNet block. As a result, this highway connection-based model will capture image structure better than skip connection-based ones. Furthermore, the convex combination (Eq. 3.7) in proposed connection improves the model's robustness to dying ReLU and gradient vanishing/exploding problems than those using skip connections. When batch normalisation and layer normalisation are successfully applied to tackle these two problems in CNNs, they also degrade the distinguishing features for recovering pixel accuracy. The proposed architecture helps to avoid the use of batch normalisation while still achieving stability in training and high accuracy in image reconstruction. The faster convergence and better generality validate the effectiveness of the proposed connection over the ResNet connection. The major limitation of this architecture is its high computational cost due to the use of the sigmoid function in the proposed connection, meaning it may not be suitable for very deep architectures.

2. In Chapter 4, a novel fusion-based method is proposed for hyperspectral image super-resolution. The multi-scale spatial fusion strategy is used to tackle the spatial disparity between the low-resolution hyperspectral image (Lr-HSI) and the high-resolution multispectral image (Hr-MSI). Under this strategy, the hierarchical feature maps extracted from two data sources can be fused at various levels, allowing for the reconstruction of coarse-to-fine detail of a high-resolution hyperspectral image (Hr-HSI). Besides, a multi-task learning framework is introduced for a novel regularisation. The input Hr-MSI and the estimated Hr-HSI must have common representation because they both capture the same scene. Because Hr-MSI has a high resolution, using features extracted from it is far more reliable than using those extracted from Lr-HSI. As a result, the denoising

autoencoder performed on Hr-MSI input is introduced as an auxiliary task. Intuitively, when two Hr-MSI inputs produce very similar features in the bottleneck of the denoising autoencoder, the two output Hr-HSIs must not have much feature disparity. The various experimental results validate that the introduced auxiliary denoising autoencoder is found to improve the generality and reconstruction accuracy of the proposed model. One drawback of the proposed architecture is the additional parameters introduced by the denoising subnetwork. Moreover, although multi-task learning has been shown in several studies to improve main task performance, the exact trade-off weight between the main task and auxiliary task must be predefined.

3. In Chapter 5, a novel dynamic schedule for choosing the strength of the gradient penalty is proposed for GAN-based image super-resolution. This method aims to improve the performance of the 0-GP approach, whose results are sensitive to the choice of penalty coefficient and are still unstable through the GAN training. The trust region method is proposed to adjust the penalty weight. We first define a region around the current best solution to assess whether a given generated image can approximate a real image through the loss of the discriminator. The strength of the gradient penalty then requires an appropriate adjustment within the region. Using this schedule, the capacities of the discriminator and the generator are balanced, which helps to mitigate both the overfitting of the discriminator and the over-penalising of the discriminator. This proposed schedule does not introduce additional computation, and experimental results demonstrated that using a dynamic schedule can improve both the training stability and the high quality of images compared to those using a pre-defined schedule. The major drawback of the proposed schedule is that it only applies to zero-centered gradient penalties and does not work for 1-Lipchitz regularisation, including commonly used one-centered gradient penalties. Moreover, the threshold set in the schedule varies according to the dataset used.

6.3 Future Work

From the presented results and conclusions, some areas are identified that can be explored further in future work. These can be summarised as follows:

1. To deal with the highly computational cost of the proposed highway connection, dimension reduction may be employed on the input of the sigmoid function, then the dimension of its output would be expanded back to the original dimension. Although replacing the local skip connections with the proposed highway connections has shown improvements, the global skip connection must keep guaranteeing the stability of training. The skip connection is still essential for a deep network; therefore, investigating the optimal proportion of highway connections to skip connections within a deep network is still an open question for future research.

2. The performance and robustness of the proposed multi-task learning framework will be enhanced by choosing the best trade-off between losses. One approach, for instance, modifies task weighting based on the gradient norm ratio for each task [178]. The application of this research to the proposed model would be an interesting topic to investigate in the future.

3. In Chapter 4, only a high-resolution multispectral image is employed for the auxiliary task. Though the observed hyperspectral image is of low resolution, the question is whether it can be considered for another auxiliary task or under what conditions?

4. In Chapter 5, although the weight coefficient of regularisation can be adapted, the range for which it is adjusted still must be predefined. Setting a small range may result in the coefficient alternately switching between two values, causing the discriminator to have insufficient time to learn meaningful patterns. Conversely, when the range is set too wide, the dynamic schedule becomes a pre-defined schedule. Therefore, developing theoretical support for choosing the

appropriate threshold is an important direction for future research and will tackle the weakness of the proposed schedule.

5. The gradient penalty is not the only type of regularisation that can penalise the model. There are also other forms of regularisation, such as data augmentation, consistency regularisation [179], etc. The combination of two or more types of regularisation may affect the effectiveness of the selected strength of gradient penalty on the dynamic schedule process. In future work, studying the adaptive schedule in the combination of various regularisation approaches is a topic worth researching.

References

[1] H. Greenspan, "Super-resolution in medical imaging," *The computer journal*, vol. 52, no. 1, pp. 43–63, 2009.

[2] Y. Huang, L. Shao, and A. F. Frangi, "Simultaneous super-resolution and cross-modality synthesis of 3d medical images using weakly-supervised joint convolutional sparse coding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6070–6079.

[3] D. Dai, Y. Wang, Y. Chen, and L. Van Gool, "Is image super-resolution helpful for other vision tasks?" in *IEEE Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–9.

[4] L. Zhang, H. Zhang, H. Shen, and P. Li, "A super-resolution reconstruction algorithm for surveillance images," *Signal Processing*, vol. 90, no. 3, pp. 848–859, 2010.

[5] P. Rasti, T. Uiboupin, S. Escalera, and G. Anbarjafari, "Convolutional neural network super resolution for face recognition in surveillance monitoring," in *International Conference on Articulated Motion and Deformable Objects*, 2016, pp. 175–184.

[6] T. Lillesand, R. W. Kiefer, and J. Chipman, *Remote sensing and Image Interpretation*. John Wiley & Sons, 2015.

[7] J. Shermeyer and A. Van Etten, "The effects of super-resolution on object detection performance in satellite imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[8] M. Haris, G. Shakhnarovich, and N. Ukita, "Task-driven super resolution: Object detection in low-resolution images," in *International Conference on Neural Information Processing*, 2021, pp. 387–395.

[9] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Learning face hallucination in the wild," in *Twenty-ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 3871—-3877.

[10] P. Li, L. Prieto, D. Mery, and P. J. Flynn, "On low-resolution face recognition in the wild: Comparisons and new techniques," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2000–2012, 2019.

[11] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[12] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[13] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[14] L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis, "Dying relu and initialization: Theory and numerical examples," *arXiv preprint arXiv:1903.06733*, 2019.

[15] Z. Zhong, J. Li, D. A. Clausi, and A. Wong, "Generative adversarial networks and conditional random fields for hyperspectral image classification," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3318–3329, 2019.

[16] H. Li, G. Xiao, T. Xia, Y. Y. Tang, and L. Li, "Hyperspectral image classification using functional data analysis," *IEEE Transactions on Cybernetics*, vol. 44, no. 9, pp. 1544–1555, 2013.

[17] Y. Zhou and Y. Wei, "Learning hierarchical spectral–spatial features for hyperspectral image classification," *IEEE Transactions on Cybernetics*, vol. 46, no. 7, pp. 1667–1678, 2015.

[18] J. Zabalza, J. Ren, Z. Wang, S. Marshall, and J. Wang, "Singular spectrum analysis for effective feature extraction in hyperspectral imaging," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 11, pp. 1886–1890, 2014.

[19] C. Zhao, X. Li, J. Ren, and S. Marshall, "Improved sparse representation using adaptive spatial support for effective target detection in hyperspectral imagery," *International Journal of Remote Sensing*, vol. 34, no. 24, pp. 8669–8684, 2013.

[20] J. Tschannerl, J. Ren, H. Zhao, F.-J. Kao, S. Marshall, and P. Yuen, "Hyperspectral image reconstruction using multi-colour and time-multiplexed led illumination," *Optics and Lasers in Engineering*, vol. 121, pp. 352–357, 2019.

[21] R. Dian, S. Li, L. Fang, T. Lu, and J. M. Bioucas-Dias, "Nonlocal sparse tensor factorization for semiblind hyperspectral and multispectral image fusion," *IEEE Transactions on Cybernetics*, vol. 50, no. 10, pp. 4469–4480, 2019.

[22] Y. Chen, W. He, N. Yokoya, and T. Z. Huang, "Hyperspectral image restoration using weighted group sparsity-regularized low-rank tensor decomposition," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3556–3570, 2019.

[23] J. Zabalza, J. Ren, J. Zheng, H. Zhao, C. Qing, Z. Yang, P. Du, and S. Marshall, "Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging," *Neurocomputing*, vol. 185, pp. 1–10, 2016.

[24] J. Tschannerl, J. Ren, F. Jack, J. Krause, H. Zhao, W. Huang, and S. Marshall, "Potential of uv and swir hyperspectral imaging for determination of levels of phenolic flavour compounds in peated barley malt," *Food Chemistry*, vol. 270, pp. 105–112, 2019.

[25] X. Lu, Y. Yuan, and X. Zheng, "Joint dictionary learning for multispectral change detection," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 884–897, 2016.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[27] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[28] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, "On convergence and stability of gans," *arXiv preprint arXiv:1705.07215*, 2017.

[29] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for gans do actually converge?" in *International Conference on Machine Learning*, 2018, pp. 3481–3490.

[30] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 349–356.

[31] J. B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.

[32] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, 2000.

[33] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[34] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2004, pp. I–I.

[35] C. Y. Yang and M. H. Yang, "Fast direct super-resolution by simple functions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 561–568.

[36] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1920–1927.

[37] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*, 2014, pp. 111–126.

[38] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3791–3799.

[39] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn, "Psyco: Manifold span reduction for super resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1837–1845.

[40] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[41] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3467–3478, 2012.

[42] T. Peleg and M. Elad, "A statistical prediction model based on sparse representations for single image super-resolution," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2569–2582, 2014.

[43] S. Wang, L. Zhang, Y. Liang, and Q. Pan, "Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2216–2223.

[44] L. He, H. Qi, and R. Zaretzki, "Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 345–352.

[45] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[46] J. Kim, J. K. Lee, and K. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645.

[47] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3147–3155.

[48] W. S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 624–632.

[49] V. K. Ha, J. C. Ren, X. Y. Xu, S. Zhao, G. Xie, V. Masero, and A. Hussain, "Deep learning based single image super-resolution: A survey," *International Journal of Automation and Computing*, vol. 16, no. 4, pp. 413–426, 2019.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[51] Q. Wei, N. Dobigeon, and J.-Y. Tourneret, "Fast fusion of multi-band images based on solving a sylvester equation," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4109–4121, 2015.

[52] Q. Wei, J. Bioucas-Dias, N. Dobigeon, and J.-Y. Tourneret, "Hyperspectral and multispectral image fusion based on a sparse representation," *IEEE*

Transactions on Geoscience and Remote Sensing, vol. 53, no. 7, pp. 3658–3668, 2015.

[53] Q. Wei, N. Dobigeon, and J.-Y. Tourneret, "Bayesian fusion of multi-band images," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 6, pp. 1117–1127, 2015.

[54] Q. Xie, M. Zhou, Q. Zhao, D. Meng, W. Zuo, and Z. Xu, "Multispectral and hyperspectral image fusion by ms/hs fusion net," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1585–1594.

[55] W. Wang, W. Zeng, Y. Huang, X. Ding, and J. Paisley, "Deep blind hyperspectral image fusion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4150–4159.

[56] R. Dian, S. Li, and X. Kang, "Regularizing hyperspectral and multispectral image fusion by cnn denoiser," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 3, pp. 1124–1135, 2020.

[57] K. Zhang, M. Wang, S. Yang, and L. Jiao, "Spatial–spectral graph regularized low-rank tensor decomposition for multispectral and hyperspectral image fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 4, pp. 1030–1040, 2018.

[58] R. Dian, S. Li, A. Guo, and L. Fang, "Deep hyperspectral image sharpening," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5345–5355, 2018.

[59] W. Dong, C. Zhou, F. Wu, J. Wu, G. Shi, and X. Li, "Model-guided deep hyperspectral image super-resolution," *IEEE Transactions on Image Processing*, vol. 30, pp. 5754–5768, 2021.

[60] J. F. Hu, T. Z. Huang, L. J. Deng, T. X. Jiang, G. Vivone, and J. Chanussot, "Hyperspectral image super-resolution via deep spatiospectral attention convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[61] J. Yang, X. Fu, Y. Hu, Y. Huang, X. Ding, and J. Paisley, "Pannet: A deep network architecture for pan-sharpening," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5449–5457.

[62] X. Fu, Z. Lin, Y. Huang, and X. Ding, "A variational pan-sharpening with local gradient constraints," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 265–10 274.

[63] R. Kawakami, Y. Matsushita, J. Wright, M. Ben-Ezra, Y.-W. Tai, and K. Ikeuchi, "High-resolution hyperspectral imaging via matrix factorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2329–2336.

[64] N. Akhtar, F. Shafait, and A. Mian, "Sparse spatio-spectral representation for hyperspectral image super-resolution," in *European Conference on Computer Vision*, 2014, pp. 63–78.

[65] W. Dong, F. Fu, G. Shi, X. Cao, J. Wu, G. Li, and X. Li, "Hyperspectral image super-resolution via non-negative structured sparse representation," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2337–2352, 2016.

[66] C. Lanaras, E. Baltsavias, and K. Schindler, "Hyperspectral super-resolution by coupled spectral unmixing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3586–3594.

[67] R. Dian, L. Fang, and S. Li, "Hyperspectral image super-resolution via non-local sparse tensor factorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5344–5353.

[68] S. Li, R. Dian, L. Fang, and J. M. Bioucas Dias, "Fusing hyperspectral and multispectral images via coupled sparse tensor factorization," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4118–4130, 2018.

[69] R. Dian, S. Li, and L. Fang, "Learning a low tensor-train rank representation for hyperspectral image super-resolution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2672–2683, 2019.

[70] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Nonlocal patch tensor sparse representation for hyperspectral image super-resolution," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 3034–3047, 2019.

[71] R. Dian and S. Li, "Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 5135–5146, 2019.

[72] M. Simoes, J. Bioucas-Dias, L. B. Almeida, and J. Chanussot, "A convex formulation for hyperspectral image superresolution via subspace-based regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3373–3388, 2014.

[73] N. Akhtar, F. Shafait, and A. Mian, "Bayesian sparse representation for hyperspectral image super resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3631–3640.

[74] Y. Zhang, S. De Backer, and P. Scheunders, "Noise-resistant wavelet-based bayesian fusion of multispectral and hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 11, pp. 3834–3843, 2009.

[75] Y. Chang, L. Yan, X. L. Zhao, H. Fang, Z. Zhang, and S. Zhong, "Weighted low-rank tensor recovery for hyperspectral image restoration," *IEEE Transactions on Cybernetics*, vol. 50, no. 11, pp. 4558–4572, 2020.

[76] W. Huang, L. Xiao, Z. Wei, H. Liu, and S. Tang, "A new pan-sharpening method with deep neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 5, pp. 1037–1041, 2015.

[77] G. Masi, D. Cozzolino, L. Verdoliva, and G. Scarpa, "Pansharpening by convolutional neural networks," *Remote Sensing*, vol. 8, no. 7, p. 594, 2016.

[78] C. 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*, 2017, pp. 4681–4690.

[79] S. J. Park, H. Son, S. Cho, K. S. Hong, and S. Lee, "Srfeat: Single image super-resolution with feature discrimination," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 439–455.

[80] A. Shocher, N. Cohen, and M. Irani, ""zero-shot" super-resolution using deep internal learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3118–3126.

[81] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 701–710.

[82] Y. Zhang, S. Liu, C. Dong, X. Zhang, and Y. Yuan, "Multiple cycle-in-cycle generative adversarial networks for unsupervised image super-resolution," *IEEE transactions on Image Processing*, vol. 29, pp. 1101–1112, 2019.

[83] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," *arXiv preprint arXiv:1612.02136*, 2016.

[84] L. Mescheder, S. Nowozin, and A. Geiger, "The numerics of gans," *arXiv preprint arXiv:1705.10461*, 2017.

[85] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," *arXiv preprint arXiv:1611.02163*, 2016.

[86] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[87] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in Neural Information Processing Systems*, vol. 29, pp. 2234–2242, 2016.

[88] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.

[89] J. H. Lim and J. C. Ye, "Geometric gan," *arXiv preprint arXiv:1705.02894*, 2017.

[90] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[91] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," *arXiv preprint arXiv:1801.01401*, 2018.

[92] C. L. Li, W. C. Chang, Y. Cheng, Y. Yang, and B. Póczos, "Mmd gan: Towards deeper understanding of moment matching network," *arXiv preprint arXiv:1705.08584*, 2017.

[93] T. Unterthiner, B. Nessler, C. Seward, G. Klambauer, M. Heusel, H. Ramsauer, and S. Hochreiter, "Coulomb gans: Provably optimal nash equilibria via potential fields," *arXiv preprint arXiv:1708.08819*, 2017.

[94] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[95] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[96] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[97] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[98] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning*, 2010.

[99] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[100] I. Safran and O. Shamir, "Spurious local minima are common in two-layer relu neural networks," in *International Conference on Machine Learning*, 2018, pp. 4433–4441.

[101] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[102] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[103] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine learning*, 2008, pp. 1096–1103.

[104] A. Ng *et al.*, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[105] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *International Conference on Machine Learning*, 2011.

[106] V. K. Ha, J. Ren, X. Xu, S. Zhao, G. Xie, and V. M. Vargas, "Deep learning based single image super-resolution: A survey," in *International Conference on Brain Inspired Cognitive Systems*, 2018, pp. 106–119.

[107] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[108] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[109] R. H. Yuhas, J. W. Boardman, and A. F. Goetz, "Determination of semi-arid landscape endmembers and seasonal trends using convex geometry spectral unmixing techniques," in *JPL, Summaries of the 4th Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop*, 1993.

[110] L. Wald, *Data fusion: definitions and architectures: fusion of images of different spatial resolutions*. Presses des MINES, 2002.

[111] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019.

[112] S. Barratt and R. Sharma, "A note on the inception score," *arXiv preprint arXiv:1801.01973*, 2018.

[113] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[114] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, 2016.

[115] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

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

[117] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481.

[118] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[119] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, and T. Huang, "Wide activation for efficient and accurate image super-resolution," *arXiv preprint arXiv:1808.08718*, 2018.

[120] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4539–4547.

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

[122] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 723–731.

[123] Q. Liao and T. Poggio, "Bridging the gaps between residual learning, recurrent neural networks and visual cortex," *arXiv preprint arXiv:1604.03640*, 2016.

[124] W. Han, S. Chang, D. Liu, M. Yu, M. Witbrock, and T. S. Huang, "Image super-resolution via dual-state recurrent networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1654–1663.

[125] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[126] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[127] X. J. Mao, C. Shen, and Y. B. Yang, "Image restoration using convolutional auto-encoders with symmetric skip connections," *arXiv preprint arXiv:1606.08921*, 2016.

[128] Z. Zhong, T. Shen, Y. Yang, Z. Lin, and C. Zhang, "Joint sub-bands learning with clique structures for wavelet domain super-resolution," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[129] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, 2017, pp. 136–144.

[130] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.

[131] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International Conference on Curves and Surfaces*, 2010, pp. 711–730.

[132] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings IEEE International Conference on Computer Vision*, vol. 2, 2001, pp. 416–423.

[133] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results,"

in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, 2017, pp. 114–125.

[134] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

[135] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[136] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[137] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*, 2012, pp. 9–48.

[138] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[139] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[140] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*, 2016, pp. 630–645.

[141] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV 12*. Springer, 2015, pp. 111–126.

[142] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[143] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European Conference on Computer Vision*, 2016, pp. 391–407.

[144] J.-S. Choi and M. Kim, "A deep convolutional neural network with selection units for super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, 2017, pp. 154–160.

[145] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE signal processing magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[146] A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 2366–2369.

[147] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and Computer-assisted Intervention*, 2015, pp. 234–241.

[148] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[149] L. Le, A. Patterson, and M. White, "Supervised autoencoders: Improving generalization performance with unsupervised regularizers," *Advances in Neural Information Processing Systems*, vol. 31, pp. 107–117, 2018.

[150] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural

networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[151] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, "Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2241–2253, 2010.

[152] A. Chakrabarti and T. Zickler, "Statistics of real-world hyperspectral images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 193–200.

[153] B. Arad and O. Ben-Shahar, "Sparse recovery of hyperspectral signal from natural rgb images," in *European Conference on Computer Vision*, 2016, pp. 19–34.

[154] N. Yokoya and A. Iwasaki, "Airborne hyperspectral data over chikusei," *Space Appl. Lab., Univ. Tokyo, Tokyo, Japan, Tech. Rep. SAL-2016-05-27*, 2016.

[155] L. Wald, T. Ranchin, and M. Mangolini, "Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images," *Photogrammetric engineering and remote sensing*, vol. 63, no. 6, pp. 691–699, 1997.

[156] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5505–5514.

[157] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of*

the IEEE International Conference on Computer Vision, 2017, pp. 2223–2232.

[158] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[159] T. C. Wang, M. Y. Liu, J. Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798–8807.

[160] Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8789–8797.

[161] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain gan inversion for real image editing," in *European Conference on Computer Vision*, 2020, pp. 592–608.

[162] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9243–9252.

[163] H. Petzka, A. Fischer, and D. Lukovnicov, "On the regularization of wasserstein gans," *arXiv preprint arXiv:1709.08894*, 2017.

[164] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[165] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International conference on machine learning*. PMLR, 2016, pp. 1060–1069.

[166] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5907–5915.

[167] G. Perarnau, J. Van De Weijer, B. Raducanu, and J. M. Álvarez, "Invertible conditional gans for image editing," *arXiv preprint arXiv:1611.06355*, 2016.

[168] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE transactions on circuits and systems for video technology*, vol. 30, no. 11, pp. 3943–3956, 2019.

[169] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[170] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5485–5493.

[171] A. Bulat, J. Yang, and G. Tzimiropoulos, "To learn image super-resolution, use a gan to learn how to do image degradation first," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 185–200.

[172] H. Thanh Tung, T. Tran, and S. Venkatesh, "Improving generalization and stability of generative adversarial networks," *arXiv preprint arXiv:1902.03984*, 2019.

[173] T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization," *SIAM Journal on Numerical Analysis*, vol. 20, no. 3, pp. 626–637, 1983.

[174] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, Ontario, Tech. Rep., 2009.

[175] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision*, December 2015.

[176] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, "Stabilizing training of generative adversarial networks through regularization," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[177] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[178] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *International conference on machine learning*. PMLR, 2018, pp. 794–803.

[179] H. Zhang, Z. Zhang, A. Odena, and H. Lee, "Consistency regularization for generative adversarial networks," *arXiv preprint arXiv:1910.12027*, 2019.

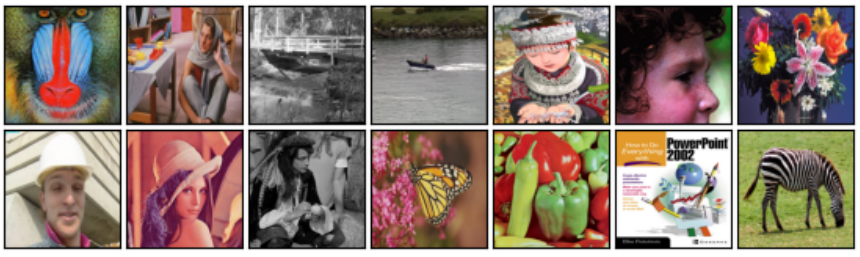
Appendix A

Image Data Sets

A.1 Set5 and Set14

The Set5 dataset [130] consists of 5 images, which are one medium size image (*baby*) and four small ones (*bird*, *butterfly*, *head*, and *woman*). The Set14 dataset [131] contains 14 images and more diverse than the Set5 dataset.

(a) Set5 dataset.



(b) Set14 dataset.

Figure A.1: Set5 and Set14 datasets.

A.2 BSD100

BSD100 is a set of 100 testing images from Berkeley Segmentation Dataset [132]. The BSD100 dataset contains 100 images of natural and cultural scenery.

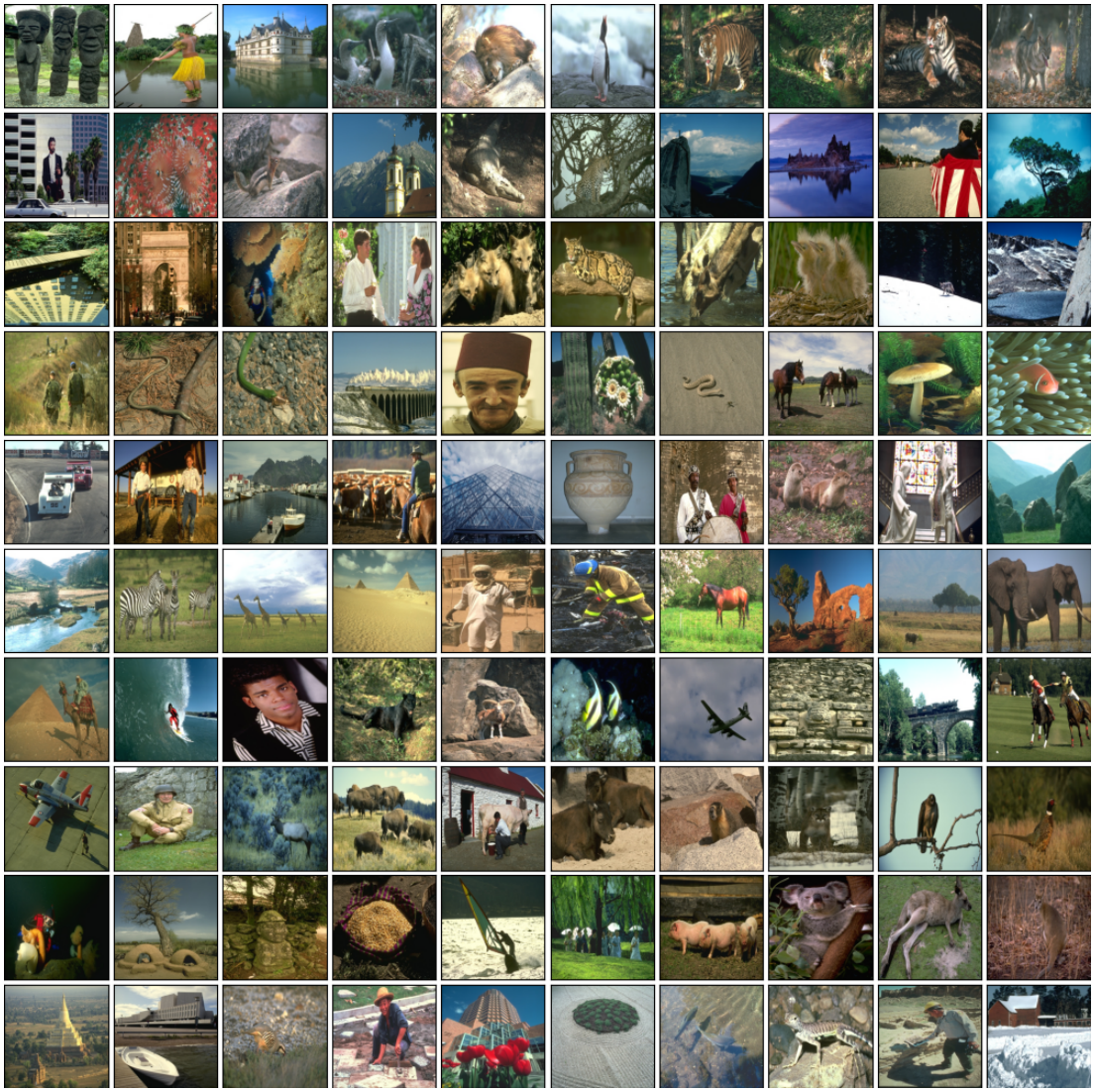


Figure A.2: BSD100 dataset.

A.3 Urban100

Urban100 [31] contains 100 images of building in urban areas with repetitive patterns and high self-similarity.

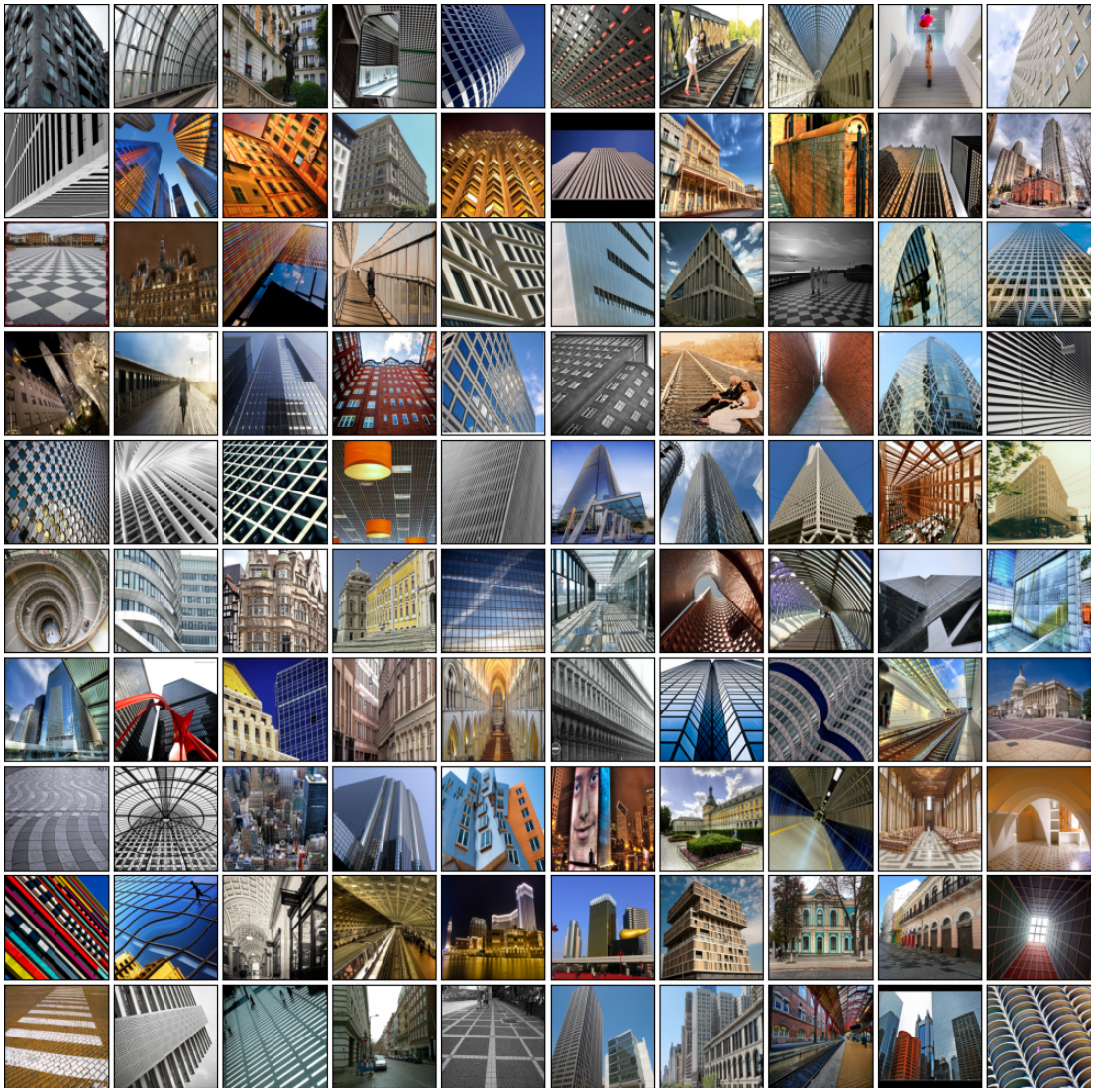


Figure A.3: Urban100 dataset.

A.4 DIV2K

DIV2K [133] contains 1000 DIVerse 2K resolution images, in which each image has a high resolution of 2K pixels on at least horizontal or vertical axes. DIV2K covers a large diversity of contents, ranging from people, handmade objects and environments, to flora and fauna, and natural sceneries. The dataset is split into training, validation, and test sets with 800 images, 100 images, and 100 images, respectively.

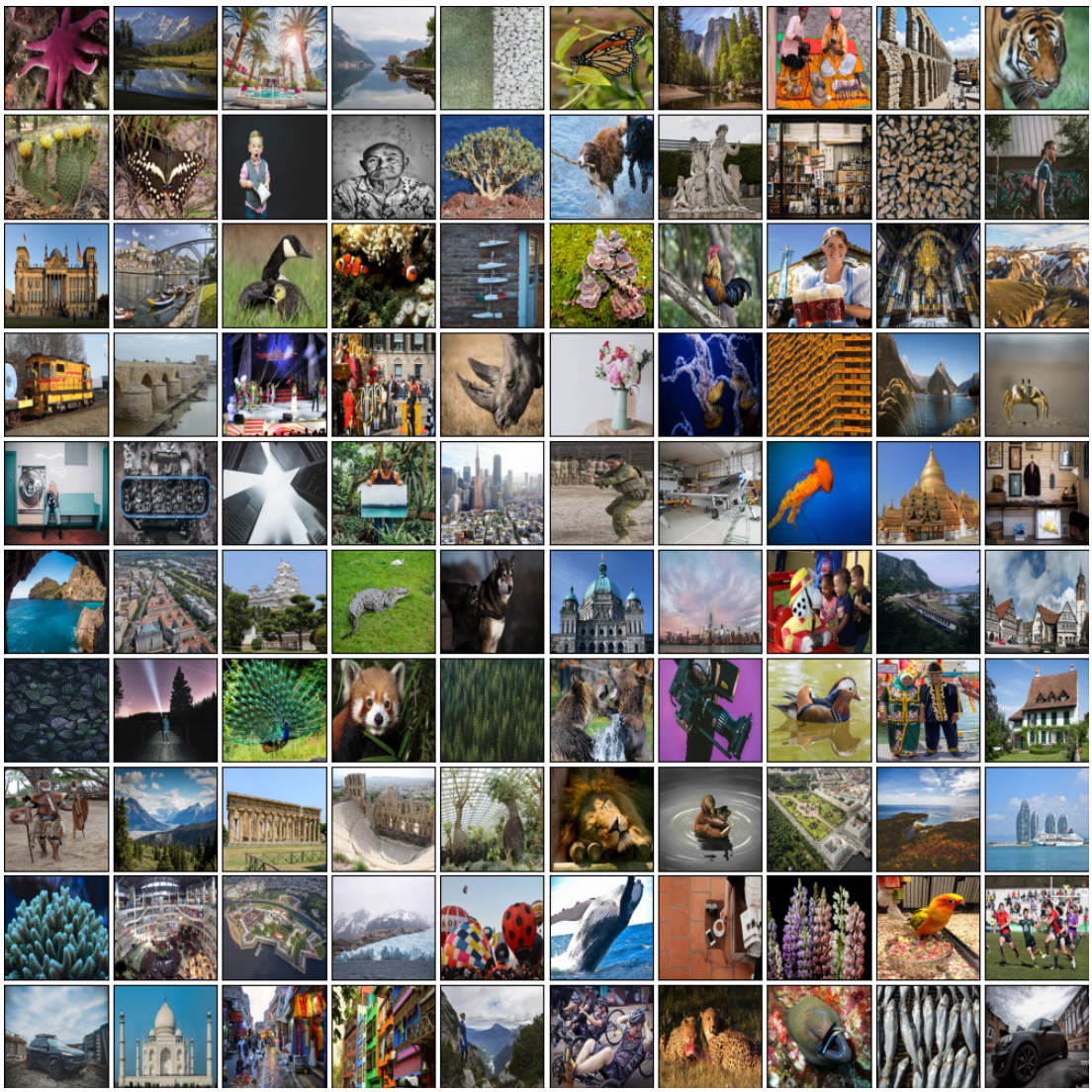


Figure A.4: DIV2K 100 training images.

A.5 CAVE

The CAVE dataset [151] comprises 32 indoor HSIs captured under controlled illumination. The images have 31 spectral bands with a spatial dimension of 512×512 pixels, and a spectral sampling gap of 10nm from 400nm to 700nm.

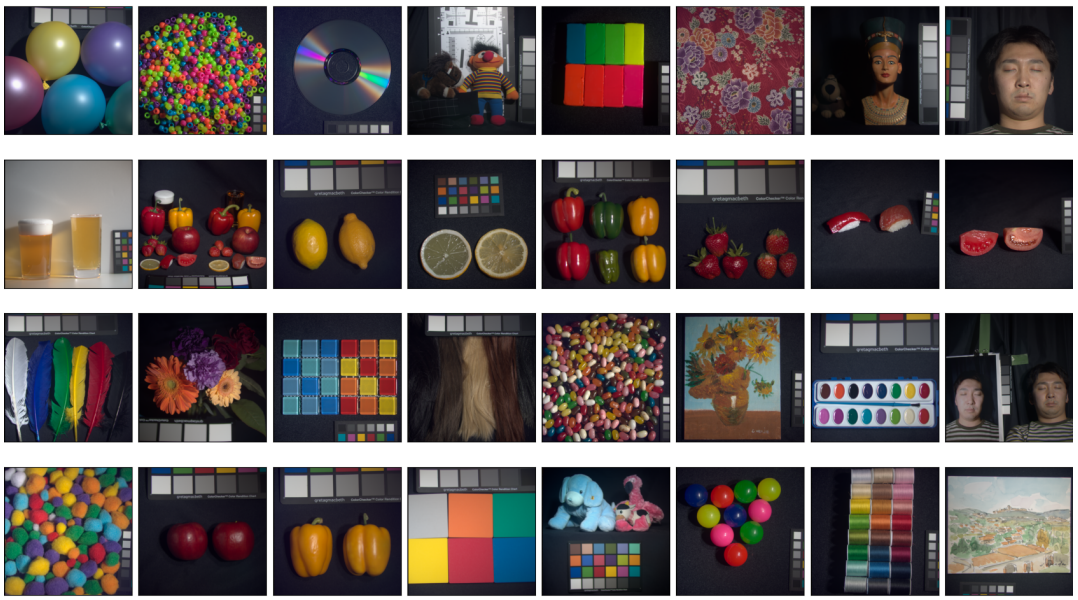


Figure A.5: Multispectral images of the CAVE dataset.

Table A.1: Image Capture Information.

Camera	Cooled CCD camera (Apogee Alta U260)
Resolution	512 x 512 pixel
Filter	VariSpec liquid crystal tunable filter
Illuminant	CIE Standard Illuminant D65
Range of wevelength	400nm - 700nm
Steps	10nm
Number of band	31 band
Focal length	f/1.4
Focus	Fixed (focused using 550nm image)
Image format	PNG (16bit)

A.6 Harvard

The Harvard dataset [152] has 50 indoor and outdoor images, recorded under daylight illumination, where 27 images were under artificial or mixed illumination. With a spatial size of 1392×1040 pixels, each HSI has 31 spectral bands, with a 10-nm spectral sampling gap within [420, 720] nm.

Figure A.6: Multispectral images of the Harvard dataset.

A.7 ICVL

The ICVL dataset [153] images were acquired using a Specim PS Kappa DX4 hyperspectral camera and a rotary stage for spatial scanning. The dataset contains 201 HSIs of real-world indoor and outdoor scenes, has 31 spectral bands each ranging from 400nm to 700nm at a 10nm increment. The images of dataset captured from a variety of urban (residential/commercial), suburban, rural, indoor and plant-life scenes.

Figure A.7: 42 multispectral images from the ICVL dataset.

A.8 Chikusei

The Chikusei scene [154] is an airborn hyperspectral image taken over Chikusei, Ibaraki, Japan. The image has a spatial dimension of 2517 × 2335 pixels, comprising 128 bands in the spectral range from 363 to 1018 nm.

A.9 Roman Colosseum

The Roman Colosseum dataset contains a spaceborne image taken by World View-2 over Roman Colosseum in Rome, Italy. The high-resolution multispectral

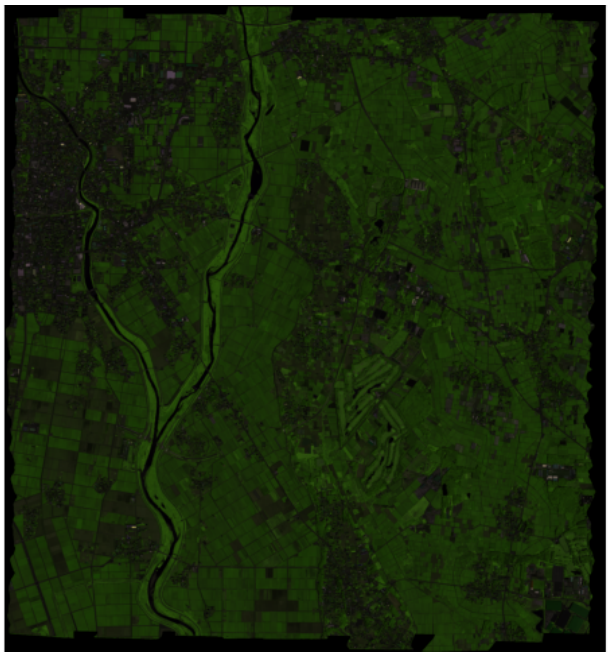


Figure A.8: The false-color image with bands (70, 100, 36) as a RGB from Chikusei dataset.

image is of size 1676 × 2632 × 3 and low-resolution hyperspectral is of size 419 × 658 × 8, while the high-resolution hyperspectral image is not available. 8 spectral bands include: Red, Green, Blue, Red Edge, Coastal, Yellow and two near-infrared (NIR) bands.

A.10 CIFAR-10

The CIFAR-10 dataset [174] consists of 60,000 32×32 colour images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images. The dataset is divided into five training batches and one test batch, each with 10,000 images. The test batch contains exactly 1,000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5,000 images

Figure A.9: Roman Colosseum image acquired by World View-2.

from each class.

A.11 CelebA

CelebFaces Attributes Dataset (CelebA) [175] is a large-scale face attributes dataset with more than 200000 celebrity images of the size 178 $\times$ 218, which mainly contains frontal portraits and is particularly biased towards groups of ethnicity white. The CelebA has a wide variety, a huge quantity, and rich annotations. The images of dataset cover a wide range of poses and cluttered backgrounds. With 10,177 identities, 202,599 face photos, 5 landmark locations, and 40 binary attribute annotations per image. The dataset can be used as the training and test sets for the computer vision tasks, including: face attribute recognition, face recognition, face detection, landmark (or facial component) localisation, and face editing and synthesis. The fact that it presents very controlled illumination settings and good photo resolution, makes it considerably easy.

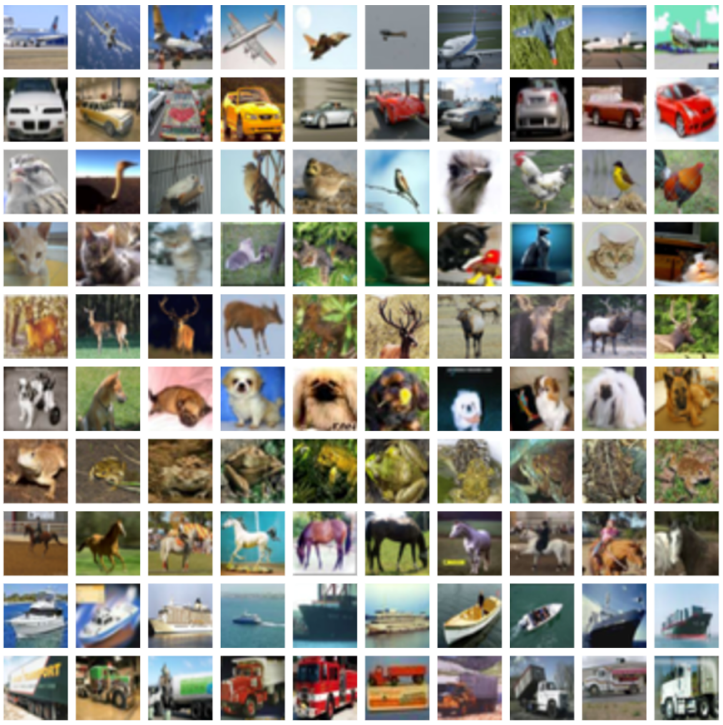


Figure A.10: Example of the original CIFAR-10 images in 10 classes. From top to bottom: Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, Truck.

Figure A.11: Image samples on the CelebA dataset with 128 × 128 resolution.