T.C.
IŞIK UNIVERSTIY
SCHOOL OF GRADUATE STUDIES

MASTER THESIS
DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
COMPUTER SCIENCE ENGINEERING PROGRAM

Salah eddine EL BALLOUTI

IMAGE SUPER RESOLUTION USING DEEP LEARNING TECHNIQUES

SUPERVISOR
Prof. Dr. M. Taner ESKİL

İSTANBUL, September 2024

T.C.
IŞIK UNIVERSITY
SCHOOL OF GRADUATE STUDIES

MASTER THESIS
DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
COMPUTER SCIENCE ENGINEERING PROGRAM

Salah eddine EL BALLOUTI

(22COMP5003)

IMAGE SUPER RESOLUTION USING DEEP LEARNING TECHNIQUES

SUPERVISOR

Prof. Dr. M. Taner ESKİL

İSTANBUL, September 2024

T.C.
IŞIK UNIVERSITY
SCHOOL OF GRADUATE STUDIES

MASTER THESIS
DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
COMPUTER SCIENCE ENGINEERING PROGRAM

Salah eddine EL BALLOUTI

(22COMP5003)

IMAGE SUPER RESOLUTION USING DEEP LEARNING TECHNIQUES

Date: 02 / 09 / 2024

Thesis Supervisor: Prof. Dr. M. Taner ESKİL / IŞIK UNIVERSITY

Jury Members:

Dr. Öğr. Üyesi Emine Ekin / IŞIK UNIVERSITY
Dr. Öğr. Üyesi Günet Eroğlu / BAHÇEŞEHIR UNIVERSITY

İSTANBUL, September 2024

ÖZET

GÖRÜNTÜLERİN DERİN ÖĞRENME TEKNİKLERİ İLE ÜSTÜN ÇÖZÜNÜRLÜKTE YENİDEN OLUŞTURULMASI

Derin Öğrenme Teknikleri kullanarak Görüntü Süper Çözünürlüğü (SR), görüntü kalitesini ve detayını geliştirmede önemli bir araştırma alanı haline gelmiştir. Bu tez, CARN, EDSR, ESPCN, RCAN, RDN, SRCNN, SRGAN ve VDSR olmak üzere sekiz gelişmiş derin öğrenme tabanlı SR yöntemini, DIV2K veri kümesini kullanarak incelemekte ve karşılaştırmaktadır. Değerlendirme, her yöntemin etkinliğini, verimliliğini ve yapısını tam anlamıyla anlamak için birden fazla yönü kapsamaktadır. Süper çözünürlüklü görüntülerin kalitesini değerlendirmek için tepe Sinyal-Gürültü Oranı (PSNR) ve Yapısal Benzerlik İndeksi (SSIM) gibi performans ölçümleri kullanılmaktadır.

Hesaplama verimliliği, çıkarım süresi ve bellek gereksinimleri temel alınarak değerlendirilmiştir. Eğitim süresi, DIV2K veri kümesinde eğitim için yakınsamanın hızı göz önünde bulundurularak analiz edilmiştir. Model karmaşıklığı incelenmiş, ağ derinliği gibi mimari detaylar ve kalıntı blokları ve dikkat mekanizmaları gibi özel unsurların entegrasyonu gibi konular araştırılmıştır. Ek olarak bu tezde, performans ve karmaşıklık arasındaki dengeyi net ve detaylı bir şekilde ortaya koyduk; daha karmaşık mimarilerin daha basit modellere kıyasla önemli ölçüde daha iyi sonuçlar verip vermediğini ve hesaplama maliyetinin geliştirmeleri haklı çıkarıp çıkarmadığını inceledik.

Son olarak, her tekniğin güçlü ve zayıf yönlerini vurgulamak için nitel bir karşılaştırma yaptık. Bu kapsamlı analiz aracılığıyla, bu tez, araştırmacıların ve uygulayıcıların çeşitli uygulamalar için en uygun yöntemi seçmelerine yardımcı olarak, derin öğrenme tabanlı görüntü süper çözünürlüğü alanında içgörüler sunmaktadır.

Anahtar Kelimeler: Görüntü Süper-Çözünürlük, Derin Öğrenme.

ABSTRACT

IMAGE SUPER RESOLUTION USING DEEP LEARNING TECHNIQUES

Image SR using Deep Learning Techniques has become a critical area of research, with significant progress in improving image quality and detail. This thesis examines and contrasts eight advanced deep learning-based SR methods: CARN, EDSR, ESPCN, RCAN, RDN, SRCNN, SRGAN, and VDSR, using the DIV2K dataset. The evaluation covers multiple aspects to offer a thorough understanding of each method's effectiveness, efficiency, and structure. Performance measurements such as PSNR and SSIM are utilized for evaluating the fidelity of super-resolved images. Computational efficiency is evaluated based on inference time and memory requirements.

Training time is analyzed, taking into account the speed of convergence for training on the DIV2K dataset. Model complexity is examined, exploring architectural details such as network depth, and the integration of specialized elements like residual blocks and attention mechanisms. Additionally, the thesis explains in a clear and detailed manner the trade-offs between performance and complexity, discussing whether more complex architectures deliver significantly better results compared to simpler models and whether the computational cost justifies the improvements.

Finally, a qualitative comparison is conducted to emphasize the strengths and weaknesses of each technique. Through this comprehensive analysis, this thesis offers insights into the field of deep learning-based image SR, assisting researchers and practitioners in choosing the most appropriate method for various applications.

Keywords: Image Super-Resolution, Deep Learning, Image Enhancement.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my thesis supervisor, Prof. Dr. M. Taner ESKİL, for his invaluable guidance, continuous support, and patience during my research. His immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life.

I must express my very profound gratitude to my parents and family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Lastly, I would like to thank IŞIK UNIVERSITY for providing the necessary facilities and resources to complete my research.

Salah eddine EL BALLOUTI

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS LIST

SR: Super Resolution

LR: Low Resolution

CARN: Cascading Residual Network

ESDR: Enhanced Deep Residual Networks

ESPCN: Efficient Sub Pixel Convolutional Neural Network

RCAN: Residual Channel Attention Networks

RDN: Residual Dense Network

SRCNN: Super Resolution Convolutional Neural Networks

SRGAN: Super Resolution Generative Adversarial Network

VDSR: Very Deep Super Resolution

DL: Deep Learning

PSNR: Peak Signal to Noise Ratio

SSIM: Structural Similarity Index Measure

RCAB: Residual Channel Attention Block

SISR: Single Image Super Resolution

ResNet: Residual Network

RIR: Residual In Residual

RG: Residual Group

LSC: Long Skip Connection

SSC: Short Skip Connection

MISR: Multi-Image Super Resolution

RDB: Residual Dense Block

CA: Channel Attention

BN: Batch Normalization

BI: Bicubic Interpolation

VGG: Visual Geometry Group

MSE: Mean Squared Error

LRL: Local Residual Learning

CM: Contiguous Memory

GFF: Global Feature Fusion

CHAPTER 1

1. INTRODUCTION

SR is a computational imaging technique employed in the realm of computer vision to enhance the resolution of low-quality images, resulting in HR outputs. The aim of this technique is to enhance the details, textures and visual quality of the images. It has many applications in various domains such as satellite imagery analysis, medical imaging and security surveillance.

This technology helps to reduce costs by enlarging LR images instead of capturing HR ones, and it advances technology, expanding applications in AI and image processing. Companies such as NVIDIA and Netflix employ this technology to enhance user experiences. NVIDIA's DLSS increases the resolution of game graphics in real-time to improve visuals without high performance costs, while Netflix scales up older shows to HD, optimizing bandwidth by streaming LR video and upscaling it on devices, thus maintaining quality during network congestion.

Despite having a long history, SR faces several challenges in computer vision research due to its complex nature. The process of extracting multiple HR images from a single LR image can be affected by a range of factors. such as the brightness and coloring of the images. This complexity makes SR a demanding research task.

However, breakthroughs in deep learning and techniques based on it have significantly advanced SR and helped achievement of cutting edge performance in benchmarks. Nevertheless, entering this field can be challenging due to the large volume of publications. Navigating through the pros and cons of various publications related to SR is a delicate and time-consuming task. The growing interest in this field, both in academia and industry, highlights the importance of SR and its real-world applications.

While most existing surveys focus on classic methods, this thesis mainly focuses on Deep Learning techniques that have shown superior performance compared to other methods. Some of the recently popularized methods in this field include improvements in image quality assessment, better datasets, normalization techniques, and innovative architectural approaches.

This thesis aims to review and analyze eight of the most superior methodologies in SISR. It also provides an overview of some of the new trends in this field. The next sections of this survey are organized in a way that provides insights into some of the definitions and metrics used by the majority of the published works on this topic.

1.1 BACKGROUND

SR refers to techniques that can produce HR images from LR ones. The SR domain includes SISR which focuses on a single image as the name suggests, and MISR which is focused more on videos and image sequences. In SISR, one LR input image can be transformed into a HR output image, while MISR can generate multiple HR output images from several LR images. Since SISR is more commonly used, most research in this field is focused on it. The majority of the techniques used in SISR can also be applied to the more complex MISR scenarios.

1.2 EVALUATION METHODS

Image quality evaluation typically involves using both subjective assessments based on human perception, such as determining whether an image appears realistic or not, and objective methods. SISR aims to produce images that align with human perception and exhibit high image quality. However, since subjective human perception assessments are unreliable, most of the research is based on objective methods. Nevertheless, objective methods may not always

capture the nuances of human perception, leading to notable discrepancies in the results obtained through subjective and objective approaches.

In this thesis, we will be focusing on two main evaluation methods: PSNR and SSIM. These two objective evaluation metrics will be used to evaluate the quality of the HR images produced by each of the eight selected deep learning techniques (See Chapter 4).

1.2.1 Peak Signal to Noise Ratio (PSNR)

The evaluation of the quality of images is important, especially in recent times where there are a lot of images being generated. Subjective evaluations are not always consistent, so it is useful to have an objective way to measure quality. One widely used objective evaluation method is PSNR, which is defined as the ratio of the maximum potential pixel value (L) to the Mean Squared Error of reference images (Keleş et al., 2021). With the approximation ŷ and the ground-truth y, the PSNR is defined as a value on the decibel scale as shown is this equation bellow:

$$\mathrm{PSNR}(\mathbf{y}, \hat{\mathbf{y}}) = 10 \cdot \log_{10} . \frac{L^2}{\sum_{p \in \Omega_{\mathbf{y}}} \left[\mathbf{y_p} - \widehat{\mathbf{y}_p}\right]^2} \qquad (1.1)$$

Although SR models are often evaluated using this metric (PSNR), it may not always produce satisfactory results in real-world scenarios. This is because the PSNR metric prioritizes pixel-level differences over overall image structures, which is not always aligned with human visual perception. Therefore, its correlation with the perceived image quality by humans can be weak. Even minor pixel adjustments, such as shifting, can lead to a significant reduction in PSNR, while the human eye may not be able to distinguish the difference. As a result, new evaluation metrics are emerging to better capture the structural features of an image from the perspective of humans.

1.2.2 Structural Similarity Index Measure (SSIM)

SSIM is a metric of evaluation used to quantify the similarity between two images (Wang, Bovik, Sheikh, & Simoncelli, 2004). In the field of image SR, SSIM can be used to compare the HR ground truth image with the output of a SR algorithm to evaluate how well the algorithm has performed.

SSIM takes into account three aspects of similarity between images: luminance, contrast, and structure. It measures the similarity in luminance between corresponding pixels, the similarity in contrast between the images, and the similarity in structure, which is determined by comparing the local patterns of pixel intensities.

The SSIM index produces a value between -1 and 1, where 1 indicates perfect similarity between the images and -1 indicates complete dissimilarity. On a pixel level, SSIM is calculated on different sections of an image. Let x and y be two sections of the same size, the following equation defines SSIM:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \qquad (1.2)$$

Where μ_x and μ_y is the pixel average value (mean) of x and y respectively, σ_x^2 and σ_y^2 is the variance of x and y respectively, σ_{xy} is the covariance of x and y, and C_1 and C_2 are two variables to stabilize the division with weak denominator.

In image SR, researchers often use SSIM as one of the evaluation metrics to assess the quality of the generated HR images compared to the ground truth. A higher SSIM value suggests that the SR algorithm has preserved more structural information and details from the original LR image.

CHAPTER 2

2. LITERATURE SURVEY

2.1 CASCADE RESIDUAL NETWORK (CARN)

CARN is a technique that was introduced in 2018 (Ahn, Kang, & Sohn, 2018). The CARN technique leverages the power of deep neural networks, specifically cascading residual blocks, to achieve fast and accurate SR while maintaining a lightweight architecture.

The core of the CARN technique lies in its cascading residual blocks. These blocks are arranged in a cascading manner (Figure 2.1), where the output of one residual block is fed as input to the next residual block. This allows for deeper networks without suffering from vanishing gradients or degradation issues (Hochreiter, 1998).

Vanishing gradients happen when the gradients (derivatives) in deep neural networks become very small during training. This makes it hard for the network to learn because updates to the weights are too tiny to make a significant impact (Hu, Zhang, & Ge, 2021).

Degradation issues refer to the phenomenon where adding more layers to a deep neural network leads to higher training error. This happens when deeper networks are harder to optimize, partly due to vanishing gradients. Both issues can hinder the training of deep neural networks.

Like many modern deep learning architectures, CARN employs residual learning. Each residual block learns to predict the residual (difference) between the LR input image and its corresponding HR counterpart. By learning residuals, the network can focus on capturing the high frequency details that are essential for SR.

The initial layers of the network are responsible for extracting features from the LR input image. These features capture important information about edges, textures, and structures present in the image.

CARN typically employs upsampling layers to gradually increase the spatial resolution of the feature maps. These upsampling layers can use techniques like bicubic interpolation or learnable deconvolution layers that upsamples feature maps.

Towards the end of the network, the feature maps are processed to reconstruct the HR image. This reconstruction process involves combining the HR features learned by the network with the original LR input image to generate the final super resolved image.

Overall, CARN achieves a balance between speed, accuracy, and model size, making it suitable for real time applications and resource constrained environments while delivering high quality SR results (Ahn, Kang, & Sohn, 2018).

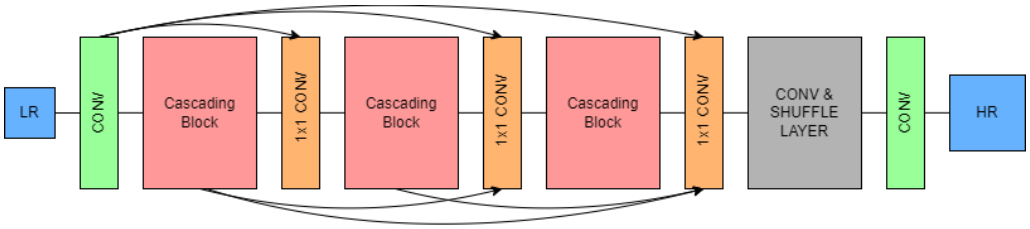


Figure 2.1: CARN Architecture (Ahn, Kang, & Sohn, 2018).

As observed in Figure 2.1, CARN leverages a cascading mechanism, integrating both local and global cascading modules into a ResNet (He, Zhang, Ren, & Sun, 2015a) based architecture. This design enables the incorporation of features from multiple layers, facilitating multi-level representation learning and efficient information propagation. By reconstructing LR images based on multi-level features, the models effectively restore image details and contexts simultaneously, improving both primitive and complex objects. The cascading

scheme enhances information propagation by providing multiple pathways for feature flow and allowing the model to learn optimal pathways.

However, the effectiveness of multiple shortcuts diminishes when using only one of the local or global cascading mechanisms, particularly the local connection.

2.2 ENHANCED DEEP SUPER RESOLUTION (EDSR)

EDSR is a technique particularly known for achieving very good performance (Lim, Son, Kim, Nah, & Lee, 2017). It is built upon the architecture of deep residual networks (ResNet) (He, Zhang, Ren, & Sun, 2015a). Residual networks are known for their ability to train very deep neural networks by using skip connections, which allow gradients to flow more easily during training. In EDSR, these skip connections are used to pass information from more shallow layers directly to deeper layers, facilitating the learning process as seen in Figure 2.2.

The EDSR method utilizes a set of convolutional layers to obtain hierarchical features from the LR input image. These layers are specifically designed to detect and capture both low-level and high-level features like edges, textures, and structures, which are crucial for achieving SR.

EDSR employs residual learning, similar to other deep learning structures. The network's residual blocks are used to predict the variance between the LR input image and its corresponding HR version. The network's focus on learning the residuals enables it to more accurately capture the high frequency details necessary for SR.

In order to train very deep networks effectively, EDSR makes use of skip connections not only within residual blocks, but also across multiple blocks. The purpose of these skip connections is to enable the propagation of information and gradients throughout the network, which helps in achieving effective training.

EDSR uses upsampling layers to enhance the feature maps' spatial resolution towards the end of the network. To achieve this, the upsampling layers employ various techniques such as learnable deconvolution layers. These techniques help in upscaling the feature maps, thereby increasing the spatial resolution.

The network combines the HR features it has learned with the original LR input image to generate the final output. This is achieved by adding the predicted residual to the LR input image, resulting in the final super resolved image.

Overall, EDSR achieves very good performance in image SR by effectively utilizing deep residual networks, residual learning, skip connections, and advanced feature extraction techniques. It produces high quality super resolved images while maintaining computational efficiency (Lim, Son, Kim, Nah, & Lee, 2017).

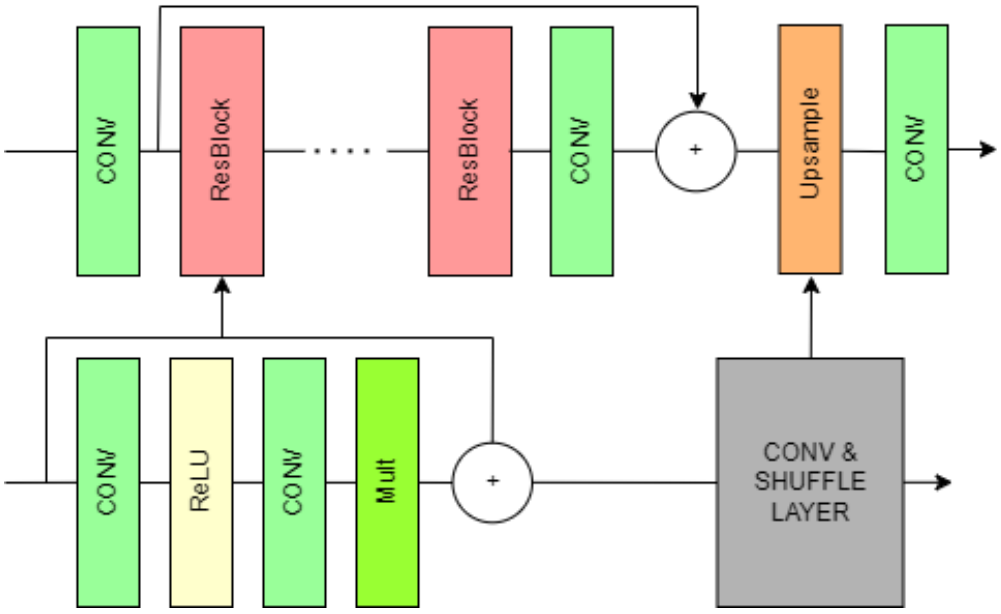


Figure 2.2: EDSR Architecture (Lim, Son, Kim, Nah, & Lee, 2017).

The objective of the model architectures is to improve the efficiency of SR performance. The architecture is based on the ResNet (He, Zhang, Ren, & Sun, 2015a) architecture, customized to optimize it for the SR task. By eliminating

batch normalization layers and integrating residual scaling, the model delivers superior performance while using less GPU memory. We will revisit this issue in Chapter 4.

The basic model for single-scale SR is built using residual blocks, but without the use of ReLU activation layers. To speed up the training process, a pre-training strategy is employed, where the parameters are initialized from pre-trained networks (Yosinski, Clune, Bengio, & Lipson, 2014). The scaling factors and model architectures are modified in order to achieve the best possible performance.

2.3 EFFICIENT SUB-PIXEL CONVOLUTIONAL NEURAL NETWORK (ESPCN)

ESPCN is a technique known for its efficiency in terms of computational resources (Shi et al., 2016). The key innovation of ESPCN lies in its use of sub-pixel convolutional layers. Unlike traditional convolutional layers that operate directly on pixel-level representations, sub-pixel convolutional layers operate at a sub-pixel level. These layers learn to upscale the spatial resolution of feature maps by a factor of the upscaling factor (e.g., 2x or 4x) using a convolutional kernel.

ESPCN utilizes several convolutional layers to extract hierarchical features from the LR input image. These layers are crucial in capturing important information, such as edges, textures, and structures that are essential for SR.

After extracting features, the network uses sub-pixel convolutional layers to improve the spatial resolution of the feature maps as depicted in Figure 2.3. Each sub-pixel convolutional layer rearranges the feature maps to achieve higher spatial resolution, which leads to upscaling of the input image.

ESPCN obtains the final output by stacking multiple sub-pixel convolutional layers that increases spatial resolution of feature maps progressively. The pixel domain is generated by converting output feature maps back, resulting in the final super-resolved image.

The efficiency of ESPCN stems from its use of sub-pixel convolutional layers, which significantly reduces the computational cost compared to traditional approaches that directly operate on HR feature maps. Despite its simplicity, ESPCN has demonstrated competitive performance in terms of both speed and quality, making it suitable for real-time applications and resource-constrained environments (Shi et al., 2016).

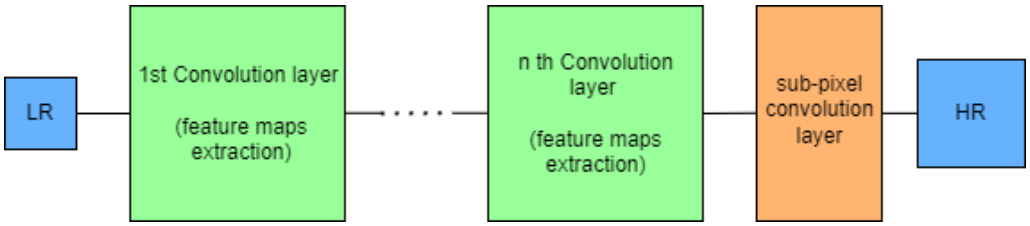


Figure 2.3: ESPCN Architecture (Shi et al., 2016).

The conventional method for image SR typically involves scaling up the LR image before processing. With the ESPCN architecture, convolutional neural networks (CNNs) are applied directly to the LR image, which is then upscaled using a sub-pixel convolution layer. This approach avoids the computational overhead of upscaling before convolution.

The sub-pixel convolution layer determines which parts of the filter to activate based on the sub-pixel location, effectively implementing upscaling in LR space. The efficient sub-pixel convolutional neural network (ESPCN) can achieve HR image reconstruction directly from LR feature maps, resulting in improved computational efficiency.

When training the network, the objective is to minimize the pixel-wise mean squared error between the reconstructed image and the corresponding ground truth HR image. The implementation of the sub-pixel convolution layer allows for faster training compared to deconvolution layers and traditional upscaling methods.

2.4 RESIDUAL CHANNEL ATTENTION NETWORKS (RCAN)

RCAN is a technique that has shown outstanding performance, particularly in reconstructing HR images with rich details (Zhang, Li, Li, Wang, Zhong, & Fu, 2018).

RCAN utilizes residual learning, which is similar to other deep learning architectures like ResNet (He, Zhang, Ren, & Sun, 2015a). In this approach, each residual block in the network undergoes training to anticipate the variance between a LR input image and its corresponding HR counterpart. By doing so, the network captures high frequency details that are crucial for SR.

The channel attention mechanism is one of the key components of RCAN. This mechanism enables the network to selectively focus on informative channels within feature maps. By assigning different attention weights to each channel based on its importance in reconstructing the HR image, the network can emphasize important channels and suppress less informative ones, leading to better performance.

RCAN is composed of multiple residual channel attention blocks (RCABs) that are stacked together (Figure 2.4). Each RCAB incorporates both residual learning and channel attention mechanisms. By doing so, these blocks help the network capture complex relationships between features across different channels and spatial locations.

Dense connectivity is another important technique used in RCAN. This approach creates a dense connection between residual blocks, allowing each block to receive inputs from all preceding blocks and pass its outputs to all subsequent blocks. This facilitates the flow of information throughout the network, enabling efficient feature reuse, and leading to better gradient flow, alleviating the vanishing gradient problem (Hochreiter, 1998; Hu, Zhang, & Ge, 2021).

To integrate features from different scales, RCAN incorporates multi-scale fusion techniques. By combining information from multiple scales, the network

can effectively capture details at various levels of abstraction, leading to more accurate and visually pleasing super-resolved images.

In addition to residual learning within individual blocks, RCAN also employs global residual learning across the entire network. This mechanism helps in preserving the global context of the input image throughout the SR process.

Overall, RCAN achieves state-of-the-art performance in image SR by effectively leveraging residual learning, channel attention mechanisms, dense connectivity, and multi-scale fusion techniques. It produces high quality super-resolved images with rich details and textures (Zhang, Li, Li, Wang, Zhong, & Fu, 2018).

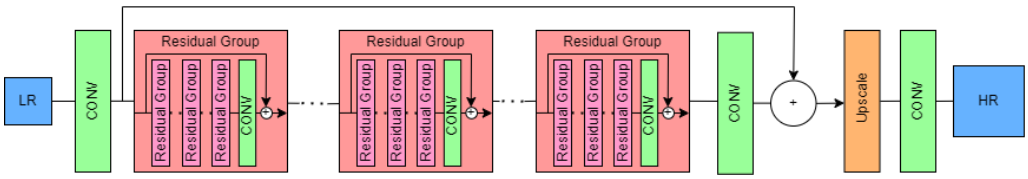


Figure 2.4: RCAN Architecture (Zhang, Li, Li, Wang, Zhong, & Fu, 2018).

Attention mechanisms (Vaswani et al., 2023), which guide processing resources to informative input components, have been applied in various deep neural networks, primarily in high-level tasks like image classification. Few works have explored attention for low-level tasks like image SR. High-frequency channel-wise features play a critical role in the reconstruction of HR images, making attention mechanisms promising for improvement. To address this, RCAN (Zhang, Li, Li, Wang, Zhong, & Fu, 2018) uses very deep residual channel attention networks, with the detailed architecture presented in Figure 2.4.

The RCAN network architecture consists of four main components: This system incorporates a process of extracting shallow features, utilizing deep residual in residual (RIR) feature extraction, incorporating an upscaling module, and culminating in a reconstruction component.

Shallow features are extracted using a single convolutional layer, followed by deep feature extraction using RIR, which contains multiple residual groups (RG). An upscale module is then applied to the deep features, and reconstruction is performed using another convolutional layer. The network is optimized using the L1 loss function and trained with stochastic gradient descent. The RIR structure includes RGs with long skip connections (LSC) and short skip connections (SSC) (He, Zhang, Ren, & Sun, 2015a; Huang, Liu, van der Maaten, & Weinberger, 2018) to facilitate training of very deep networks.

Channel attention (CA) is introduced to adaptively rescale channel-wise features, enhancing discriminative learning. The RCAB integrates CA into residual blocks, by enabling the network to concentrate its resources on informative components. The implementation details specify the number of RGs and RCABs, kernel sizes, filter numbers, and the use of ESPCNN for upscaling. The network architecture achieves notable performance improvements over older methods (Zhang, Li, Li, Wang, Zhong, & Fu, 2018).

2.5 RESIDUAL DENSE NETWORK (RDN)

RDN is a technique that has demonstrated impressive performance, particularly in reconstructing high quality images with fine details (Zhang, Tian, Kong, Zhong, & Fu, 2018).

RDN is based on residual learning, which is widely used in deep learning architectures like ResNet (He, Zhang, Ren, & Sun, 2015a). In residual learning, each residual block is trained to predict the residual or the difference between the LR input image and its corresponding HR image. The network focuses on learning residuals to better capture high-frequency details essential for SR.

Dense connectivity, which is one of the key components of RDN, ensures that every residual block is densely connected with all the preceding blocks and passes its outputs to all subsequent blocks. This connectivity pattern facilitates information flow throughout the network and enables feature reuse, leading to

better gradient flow and overcoming the vanishing gradient problem (Hochreiter, 1998; Hu, Zhang, & Ge, 2021).

RDN contains multiple residual dense blocks (RDBs) that are stacked together (Figure 2.5). Each RDB comprises multiple convolutional layers, which concatenate the output of each layer with the input to the block. The dense connection pattern in RDBs helps the network capture complex relationships between features across multiple layers, leading to more expressive representations.

Within each RDB, local feature fusion is performed by concatenating the outputs of all convolutional layers. This concatenation operation combines features extracted at different levels of abstraction, allowing the network to capture both low-level details and high-level semantic information.

In addition to local feature fusion, RDN also incorporates global feature fusion across multiple RDBs. This mechanism aggregates features from different scales and spatial locations, enabling the network to capture long-range dependencies and preserve the global context of the input image.

To increase the spatial resolution of the feature maps, RDN employs upsampling layers towards the end of the network. These upsampling layers use techniques like bicubic interpolation or learnable deconvolution layers to upsample the feature maps. The final output of the network is obtained by reconstructing the HR image from the upsampled feature maps.

Overall, RDN effectively leverages residual learning, dense connectivity, local and global feature fusion, and HR reconstruction techniques to achieve state-of-the-art performance in image SR. It produces high-quality super-resolved images with rich details and textures (Zhang, Tian, Kong, Zhong, & Fu, 2018).

The architecture of RDN is based on DenseNet (Huang, Liu, van der Maaten, & Weinberger, 2018), which introduced the DenseNet architecture and explained its effectiveness in various computer vision tasks. RDN differs from DenseNet in several aspects. Both utilize local dense connections, but RDN removes batch normalization (BN) and pooling layers to reduce computational

complexity and ensure efficient feature extraction for image SR. Additionally, RDN employs global feature fusion to fully utilize hierarchical features, which are neglected in DenseNet.

RDN method was thoroughly evaluated and compared in various aspects. The DIV2K dataset was used for both training and testing, and degradation models such as Bicubic Interpolation (BI) were used to simulate LR images for testing.

When compared with cutting-edge methods (includes methods discussed in this study) on the validation set, RDN outperforms some other methods, especially when compared to persistent CNN models like DenseNet (Huang, Liu, van der Maaten, & Weinberger, 2018). RDN effectively suppresses blurring artifacts (simulated) and handles noise while recovering more details, as will be elaborated in Chapter 4.

RDN was successfully applied to real-world images (from DIV2K dataset), outperforming some methods (like VDSR (Kim, Lee, & Lee, 2016)) in recovering sharper edges and finer details.

Overall, RDN showcases its potential for image SR tasks by demonstrating effectiveness, robustness, and superior performance across various degradation models and real-world scenarios (Zhang, Tian, Kong, Zhong, & Fu, 2018).

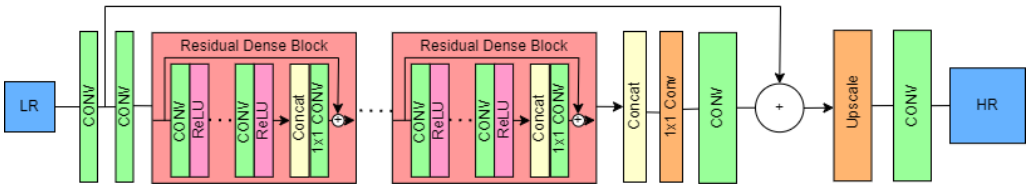


Figure 2.5: RDN Architecture (Zhang, Tian, Kong, Zhong, & Fu, 2018).

2.6 SUPER-RESOLUTION CONVOLUTIONAL NEURAL NETWORK (SRCNN)

SRCNN is one of the pioneering techniques in image SR using deep learning (Dong, Loy, He, & Tang, 2015). SRCNN initiates the process by dividing the input LR image into overlapping patches, which function as the input to the network. This allows the network to concentrate on small local regions of the image.

Each patch is then supplied to the SRCNN network, which includes several convolutional layers. These specific layers play a crucial role in the process of extracting hierarchical features from the input patches. The network's depth enables it to learn increasingly abstract representations of the input image.

The extracted features then undergo non-linear activation functions, such as rectified linear units (ReLU). The activation functions introduce non-linearity into the network, enabling it to learn complex mappings between LR and HR image patches. The network employs upsampling layers to increase the resolution of the feature maps after feature extraction and non-linear mapping. These upsampling layers use techniques like bicubic interpolation or learnable deconvolution layers to upsample the feature maps.

Finally, the HR image is reconstructed from the upsampled feature maps by combining the HR features learned by the network with the original LR input image to generate the final super-resolved image.

SRCNN is different from conventional SR techniques because it directly learns an end-to-end mapping between the LR input and the HR output images using deep convolutional neural networks. SRCNN learns to automatically generate HR images from LR inputs by training on pairs of LR and HR images, bypassing the need for handcrafted feature extraction or interpolation methods. This approach has delivered very good results in producing high-quality super-resolved images with rich details and textures (Dong, Loy, He, & Tang, 2015).

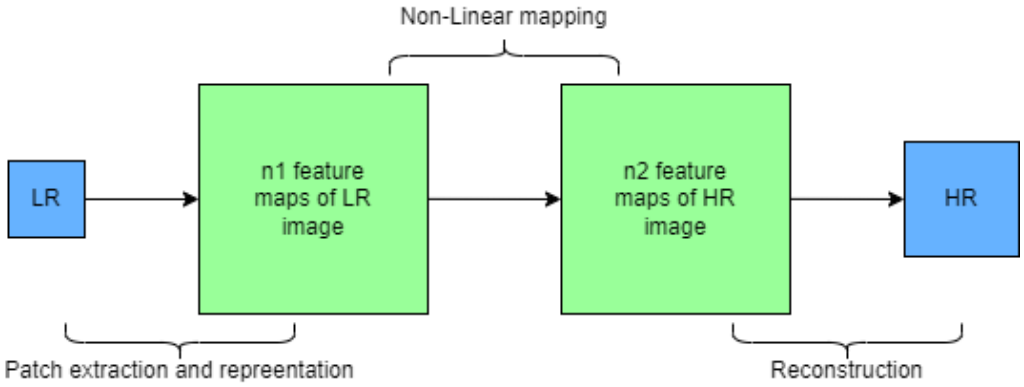


Figure 2.6: SRCNN Architecture (Dong, Loy, He, & Tang, 2015).

Figure 2.6 presents a simplified diagram for the architecture of SRCNN. The following are the three main operations of the SRCNN architecture:

The extraction and representation of patches: This involves extracting overlapping patches from a LR image and representing them as high-dimensional vectors using learned filters. It is formulated as a convolutional layer.

Non-linear mapping: This operation maps the high-dimensional vectors obtained from the first operation to another high-dimensional space using additional learned filters. It is also formulated as a convolutional layer.

Reconstruction: The final HR image is generated by aggregating the HR patch-wise representations. It is formulated as a convolutional layer.

Dong et al.'s work establishes that these operations form a convolutional neural network that requires optimization of filters and biases. The relationship between this CNN-based approach and sparse-coding-based SR methods shows that sparse-coding-based methods can be viewed as a special case of a convolutional neural network (Dong, Loy, He, & Tang, 2015).

During training, the loss (Mean Squared Error) between the reconstructed images and the HR images is minimized. The network is trained using stochastic gradient descent with backpropagation, and various hyperparameters are carefully chosen to optimize performance.

The implementation of the model uses PyTorch, with consideration for handling border effects during training and enabling application on images of arbitrary sizes during testing. Overall, the CNN-based approach demonstrates promising results for single-image SR tasks.

2.7 SUPER-RESOLUTION GENERATIVE ADVERSARIAL NETWORK (SRGAN)

SRGAN is a technique that employs a generative adversarial network (GAN) framework to produce high-quality super-resolved images with realistic textures and details (Ledig et al., 2017).

SRGAN is composed of a generator network (Figure 2.7) that generates HR images from LR inputs. The generator uses a deep convolutional neural network architecture, such as a modified version of the SRCNN or a more complex architecture like ResNet (He, Zhang, Ren, & Sun, 2015a).

SRGAN also includes a discriminator network (Figure 2.8) that acts as a critic to distinguish between real HR images and super-resolved images generated by the generator. The discriminator is trained to classify images as real or fake, providing feedback to the generator to improve the quality of its generated images.

To train both the generator and discriminator networks simultaneously, SRGAN uses adversarial training. During training, the generator aims to generate high-quality super-resolved images that are remarkably similar to real HR images, while the discriminator tries to find the differences between the real and the fake images.

In addition to the adversarial loss, SRGAN introduces a perceptual loss function based on features extracted from a pre-trained deep neural network (Yosinski, Clune, Bengio, & Lipson, 2014), such as VGG (Simonyan & Zisserman, 2015). This perceptual loss measures the perceptual difference between the generated and ground truth images in terms of high-level features, such as textures, structures, and semantics.

To encourage the generator to produce images that not only look visually pleasing but also maintain important details and textures from the LR input, SRGAN incorporates feature reconstruction loss. This loss is computed based on the difference between the features extracted from the generated and ground truth images.

During training, the generator and discriminator are trained iteratively in a min-max game. The generator aims to minimize the perceptual loss and feature reconstruction loss while maximizing the adversarial loss, whereas the discriminator aims to minimize its classification error.

Overall, SRGAN leverages the power of generative adversarial networks to produce high-quality and visually realistic super-resolved images. It achieves state-of-the-art performance in image SR and addresses the limitations of traditional SR techniques by generating images with improved perceptual quality and finer details (Ledig et al., 2017).

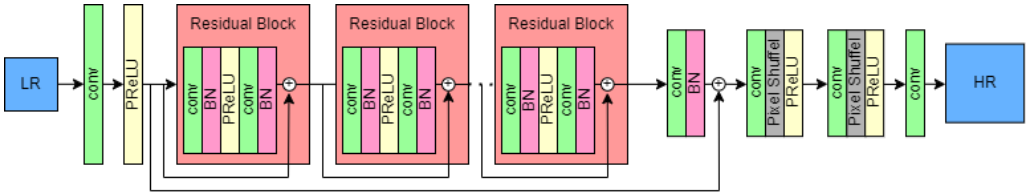


Figure 2.7: SRGAN Architecture - Generator Network (Ledig et al., 2017).

This method focuses on aiming to estimate an HR image from an LR input. It employs a generator network, trained to map LR images to HR counterparts using a specific loss function. The loss function is a combination of content loss and adversarial loss, aiming to capture perceptually relevant characteristics of the recovered SR image. The architecture includes a discriminator network, trained adversarially to distinguish between real HR images and generated SR samples.

The generator network consists of residual blocks, while the discriminator network follows architectural guidelines similar to those used in the VGG (Simonyan & Zisserman, 2015) network. The content loss is based on the VGG

(Simonyan & Zisserman, 2015) loss, measuring the Euclidean distance between feature representations of the reconstructed image and the reference image. The adversarial loss encourages solutions that resemble natural images and is defined based on the probabilities of the discriminator network. Figure 2.8 presents a simplified diagram for the architecture of the Discriminator half of SRGAN.

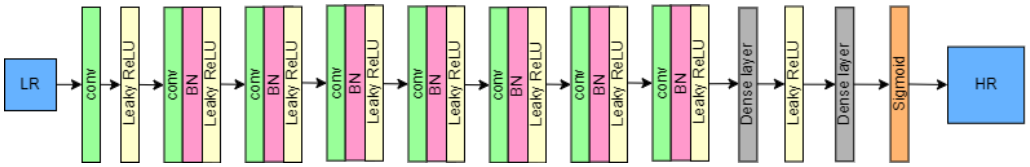


Figure 2.8: SRGAN Architecture - Discriminator Network (Ledig et al., 2017).

Deep learning-based methods, especially those with adversarial training like SRGAN, tend to be more robust to noise and other image degradations compared to traditional interpolation methods. However, the robustness may vary depending on the specific technique and the level of degradation present in the input images (Ledig et al., 2017).

Overall, the proposed framework aims to produce perceptually superior SR images by training the generator network to create solutions difficult to classify by the discriminator network (Ledig et al., 2017).

2.8 VERY DEEP SUPER-RESOLUTION (VDSR)

VDSR is a technique that focuses on training deep convolutional neural networks (CNNs) to directly learn the relationship between LR and HR images (Kim, Lee, & Lee, 2016).

VDSR uses a deep convolutional neural network architecture with multiple convolutional layers to learn complex mappings between LR and HR image patches. By capturing intricate details and textures, the network is able to produce high-quality super-resolved images.

Similar to other deep learning architectures, VDSR employs residual learning. The network is designed to predict residuals between the LR input

image and its corresponding HR version, thereby allowing it to better capture the high-frequency details needed for SR.

VDSR also uses identity mapping, where the input LR image is directly added to the predicted residuals. This helps to preserve the overall structure and global information of the input image during the SR process.

To train the network, VDSR utilizes advanced optimization techniques that rely on gradients, such as stochastic gradient descent (SGD) and the Adam optimizer, to enhance the training process and improve model performance. During training, the network learns to minimize a loss function that measures the difference between the predicted HR images and the ground truth HR images. The loss function used in VDSR typically includes a combination of pixel-wise loss (such as mean squared error) and regularization terms (such as L1 or L2 regularization) to encourage smoothness in the predicted HR images and prevent overfitting.

During the training process, VDSR is fed with paired LR and HR images. The network learns to predict HR images from their LR counterparts by iteratively updating the parameters of the convolutional layers to minimize the loss function.

Overall, VDSR achieves very good performance in image SR by leveraging deep convolutional neural networks to learn the complex mapping from LR to HR images. It produces high-quality super-resolved images with rich details and textures, surpassing the performance of traditional interpolation-based methods (Kim, Lee, & Lee, 2016).

Figure 2.9 presents a simplified diagram for the architecture of VDSR. The network consists of filters, each filter works on a region across multiple channels. To maintain feature map sizes, zero-padding is applied before convolutions. This ensures accurate predictions even near image boundaries.

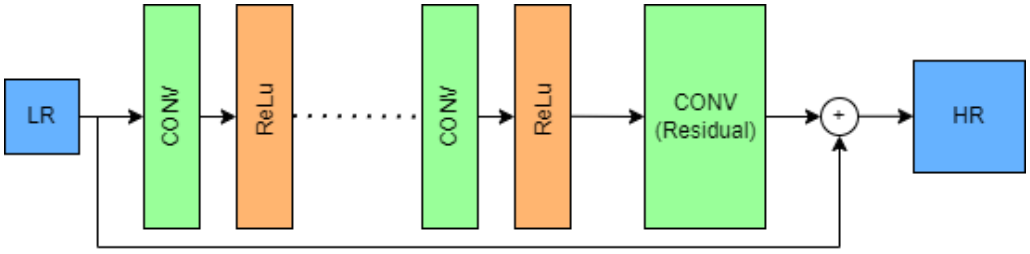


Figure 2.9: VDSR Architecture (Kim, Lee, & Lee, 2016).

The network starts with an input of LR image and predicts image details explicitly modeling residuals. This offers several advantages that will be discussed later. The objective of training is to minimize the loss (mean squared error) between the predicted HR image and the target HR image.

To mitigate vanishing/exploding gradients (Hochreiter, 1998), residual learning is employed. The network predicts the residual image, and the loss function is computed as the distance (Euclidean) between the reconstructed and ground truth image. To accelerate convergence in very deep networks, adjustable gradient clipping is utilized. This significantly reduces training time as compared to conventional methods.

Adjustable gradient clipping is a technique used during the training of neural networks to mitigate the issues caused by exploding gradients. It involves limiting the size of the gradients during training to prevent them from becoming too large and destabilizing the optimization process.

Additionally, a multi-scale model is trained to handle different scale factors efficiently. Parameters are shared across all predefined scales. Data preparation involves combining training datasets into one large dataset. Input patch sizes are the size of the target field, and images are sliced into non-overlapping sub-images.

Overall, the method is suitable for real world applications in image SR. It demonstrates improved performance and faster convergence than existing approaches (Kim, Lee, & Lee, 2016).

CHAPTER 3

3. APPROACH

In this chapter we detail our approach to conducting the comparative study on the eight SR methods' which involves implementing the algorithms necessary and building the model for each method discussed before. The implementation, training and validation was done locally on a personal laptop, given this fact the use of some pre-trained models was necessary (Yosinski, Clune, Bengio, & Lipson, 2014) to finish the training and evaluation in a timely manner, as the original methods were trained on much powerful hardware. This only sped up the process, which is a well known method for accelerating the training of large models (Chen et al., 2021).

3.1 DATASET

The DIV2K dataset is utilized as a training dataset and also for evaluating and comparing the performance of SISR algorithms. It is called "DIV2K" because it includes diverse 2K resolution images (Agustsson & Timofte, 2017).

The following are the main features of the DIV2K dataset:

- **High-Quality Images:** The dataset includes high-quality images of 2K resolution, which means images with 2048x1080 pixels or similar. These images are used as a reference for evaluating the performance of SR algorithms.
- **Diverse Content:** The dataset has images with various content that covers different types of scenes, textures, structures, and lighting conditions. This diversity ensures that SISR algorithms are evaluated on a broad range of image types and characteristics.

- **Training and Evaluation Sets:** The DIV2K dataset is divided into two main subsets: the training set and the evaluation set. The training set is used for training SISR algorithms, while the evaluation set is used for evaluating the performance of trained models. The evaluation set normally contains HR images that are not seen during training, ensuring unbiased evaluation.
- **Publicly Available:** The DIV2K dataset is publicly available, which makes it accessible to researchers and practitioners in the field of image SR. This accessibility facilitates reproducibility, comparison, and benchmarking of different SISR techniques.

Researchers and developers utilize the DIV2K dataset in the following ways for SISR tasks:

- **Training SISR Models:** HR images from the DIV2K training set are used to train and fine-tune SISR models. Models can learn to generalize better to different image characteristics and achieve high-quality SR results by training on a diverse set of high-quality images.
- **Evaluating SISR Performance:** After training SISR models, researchers evaluate their performance using images from the DIV2K evaluation set. The HR images in the evaluation set are used as ground truth references, against which the super-resolved images generated by the models are compared. Evaluation metrics like PSNR and SSIM are often used to quantify the similarity between generated SR images and target images.

DIV2K dataset takes an important role in advancing the newest research in SISR by providing a standardized benchmark for evaluating and comparing the performance of SISR algorithms (Timofte et al., 2017).

Prior to delving into the specifics of the experiment, it is important to explain the rationale behind utilizing only a single dataset for the thesis, aside from considerations of fairness.

When assessing eight image SR techniques on a single dataset using a laptop, there are limitations imposed by both time and hardware constraints. The processing of HR images through intricate algorithms is time-intensive, and standard laptops lack the necessary computational power and memory for

efficient processing, particularly for deep learning models. Additionally, extensive evaluation of results contributes to the time burden. Thermal throttling can further impede processing speed. Focusing on a limited number of techniques and one dataset allows for a manageable and comprehensive study within these constraints, ensuring dependable evaluations and establishing a groundwork for broader future research.

Next we plan to so a comparative analysis, for this reason all SR methodologies were trained and tested using the same dataset (DIV2K). This standardized approach ensures a fair and unbiased comparison, eliminating variations in training or testing data, and allowing for a consistent evaluation environment. By maintaining uniformity in the dataset, the comparative analysis provides a robust and fair assessment of each technique's capabilities, efficiency, and suitability for HR image SR tasks.

Figure 3.1: The original HR image and chosen regions for visual evaluation (EL Ballouti 2024).

To have a fair and clear comparison, training and validation were done on DIV2K dataset and for numerical evaluation, for aesthetic comparison we will use Figure 3.1 in the inference task to showcase the aesthetics of each method's resulting HR image, to replicate real world scenarios we chose an image that is not from the DIV2K dataset. This image was scaled down by a factor of 4 to obtain a LR image (Figure 3.2) and passed down to each trained model to infer the new HR image.

Four regions of each image was then cropped and compared to the original HR image (Figure 3.1) for visual evaluation and to show high frequency changes between the LR and HR-generated image.

Figure 3.2: The LR image and chosen regions for visual evaluation (EL Ballouti 2024).

Figure 3.3 zooms on the chosen 4 regions and highlights the differences between the HR and LR images. We will showcase now a similar comparison between each HR generated image and the target HR image.

(a) HR Region 1 (b) HR Region 2 (c) HR Region 3 (d) HR Region 4

(e) LR Region 1 (f) LR Region 2 (g) LR Region 3 (h) LR Region 4

Figure 3.3: Zooms on the chosen four regions and highlights the differences between the HR and LR images (EL Ballouti 2024).

We will highlight the advantages and disadvantages of each method under various conditions in the next chapter.

CHAPTER 4

4. EXPERIMENTS AND RESULTS

In this chapter, results of the experiments are discussed and we take a look at some examples of the visual qualitative comparison of each method's SR results.

4.1 CARN

The process of training involves the utilization of RGB input patches obtained from LR images. These patches are then augmented and trained using optimization techniques. The performance of CARN is comparable with the cutting-edge methods, we can see how effective it is in terms of PSNR, SSIM, and computation cost metrics, as it scores 32.23 (dB) for PSNR, and 0.8937 for SSIM, both these values are in the normal range compared to other methods (Table 4.1).

A trade-off study was conducted between performance, parameters, and operations to demonstrate the efficiency of the model compared to alternative approaches, the results of which are discussed later in this chapter. As a conclusion we found that this method provides results that are very close to the target image (Figure 4.1).

Training time for the model of this method took 2 days to train with moderate memory usage, which is expected given that the model had around 1.6 million parameters. As a result we can see that this model performed well compared to other methods in this thesis (Table 4.2).

Figure 4.1: CARN Generated Result HR image (EL Ballouti 2024).

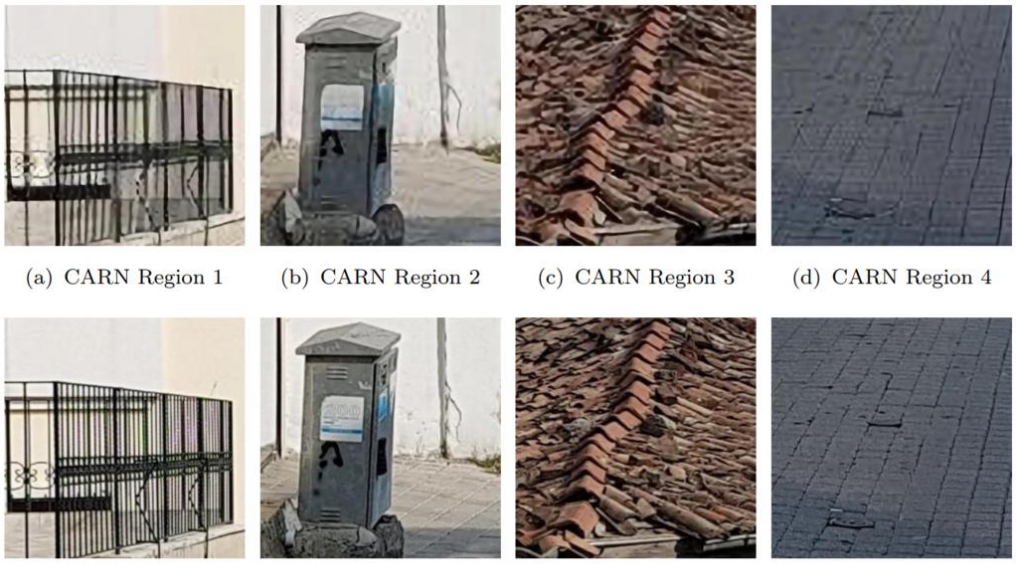


Figure 4.2: Four regions with high frequency details in CARN Generated Result HR image and the corresponding regions in HR image (EL Ballouti 2024).

Aesthetically we can see in Figure 4.2 that zooms on the chosen 4 regions and highlights the differences between the generated HR and target HR image that this method restored most of the high frequency details but did struggle to restore some of the details such as the vertical lines in the railings (sub-figure a) and the tiles on the ground (sub-Figure b and d).

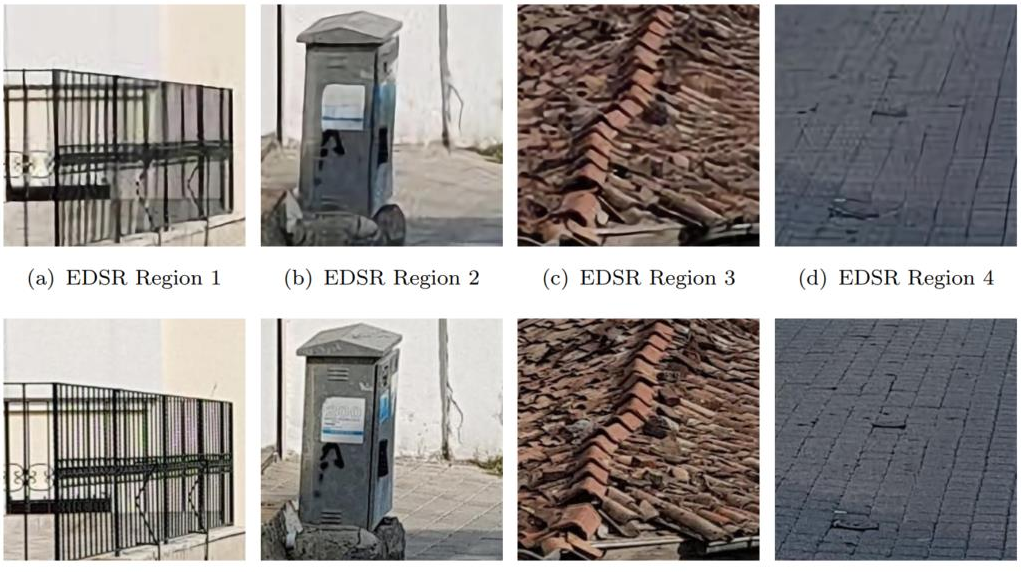
4.2 EDSR

The evaluation of image restoration tasks used the DIV2K dataset, which comprises 800 training, 100 validation, and 100 test high-quality images. The training process involved RGB input patches of size 48×48 from LR images with corresponding HR patches, which were augmented with random flips and rotations. L1 loss was employed instead of L2 due to better convergence, and we used the ADAM optimizer. The PyTorch framework was used to implement the models, and NVIDIA RTX 3070 GPU Mobile(ran locally on laptop) was used to train the model.

The evaluation on the DIV2K dataset shows improved performance compared to the original SR ResNet, particularly with modifications and self-ensemble (Lim, Son, Kim, Nah, & Lee, 2017). Benchmark results demonstrate significant improvements over some state-of-the-art methods, as this method scored 32.62 (dB) and 0.8984 on PSNR and SSIM respectively, which is in the high range compared to other methods in this thesis. It also resulted in very aesthetically pleasing results (Figure 4.3). Qualitative results confirm the successful reconstruction of detailed textures and edges in HR images, producing better-looking SR outputs compared to other works as can be seen in Figure 4.4.

Training time for the model of this method took several days to train with very high memory usage, which is expected given that the model had around 43 million parameters. As a result we can see that this model performed well compared to other methods (Table 4.2).

Figure 4.3: EDSR Generated Result HR image (EL Ballouti 2024).



(a) EDSR Region 1 (b) EDSR Region 2 (c) EDSR Region 3 (d) EDSR Region 4

(e) HR Region 1 (f) HR Region 2 (g) HR Region 3 (h) HR Region 4

Figure 4.4: Four regions with high frequency details in EDSR Generated Result HR image and the corresponding regions in HR image (EL Ballouti 2024).

4.3 ESPCN

The technique proposed in the original work was implemented and trained on DIV2K images using the network configurations discussed earlier in this thesis. The results are evaluated using PSNR and SSIM as the performance metrics and compares them against other models as shown later in Table 4.2, this method scored 30.9 dB and 0.8645 on PSNR and SSIM respectively, these are average results compared to other results (Table 4.1).

During the training, it was apparent that the benefits of the sub-pixel convolution layer and tanh activation function improved the performance to achieve the results above.

Figure 4.5: ESPCN Generated Result HR image (EL Ballouti 2024).

To evaluate the output of this method (Figure 4.5) we need to also take into consideration the efficiency of this method, as is took only a few hours to train

and had very low memory usage compared to the other methods in this thesis, this was thanks to the relative low number of parameters (about 21 thousand).

Lastly, run-time evaluations reveal significant speed improvements compared to some of the other methods being studied, making it an efficient choice for real-time SR tasks as the inference time was the lowest compared to the other methods (Table 4.2).

Figure 4.6: Four regions with high frequency details in ESPCN Generated Result HR image and the corresponding regions in HR image (EL Ballouti 2024).

Aesthetically we can see in Figure 4.6 that zooms on the chosen 4 regions and highlights the differences between the generated HR and target HR image that this method restored some of the high frequency details with a lot of smoothing happening on the edges in the restored image that can be seen in the ground tiles (sub-Figure b and d).

4.4 RCAN

The architecture of RCAN is quite intricate and comprises shallow feature extraction, Residual in Residual deep feature extraction, a module to upscale, and reconstruction part. The RIR structure makes it possible to train very deep networks, while the channel attention (CA) mechanism enhances discriminative learning by rescaling channel-wise features.

Figure 4.7: RCAN Generated Result HR image (EL Ballouti 2024).

Our results demonstrate the effectiveness of incorporating RIR (residual in residual) and CA and show significant gains over existing methods in both quantitative and qualitative evaluations. As can be seen in Figure 4.7. Additionally, this method scored the highest in both PSNR (32.62 dB) and SSIM (0.9002) respectively.

However, that came at the cost of being the slowest in inference time and the training time (Table 4.2), and with high memory usage during training

despite having only 15 million parameters, which can be attributed to the complexity of the architecture of the model. Visual comparisons further highlight RCAN's superiority in recovering details and reducing blurring artifacts, as can be seen in Figure 4.8.

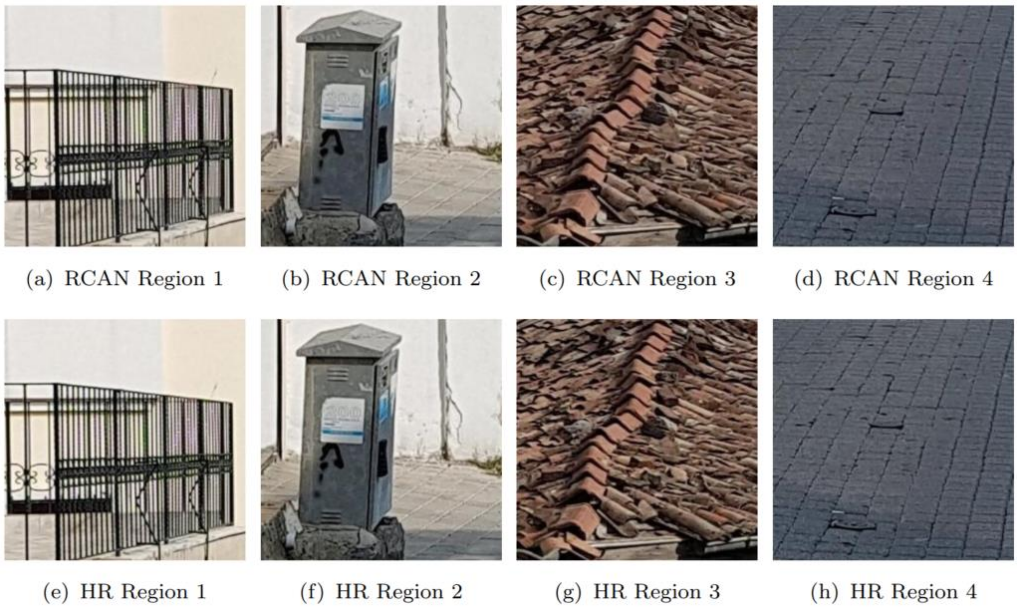


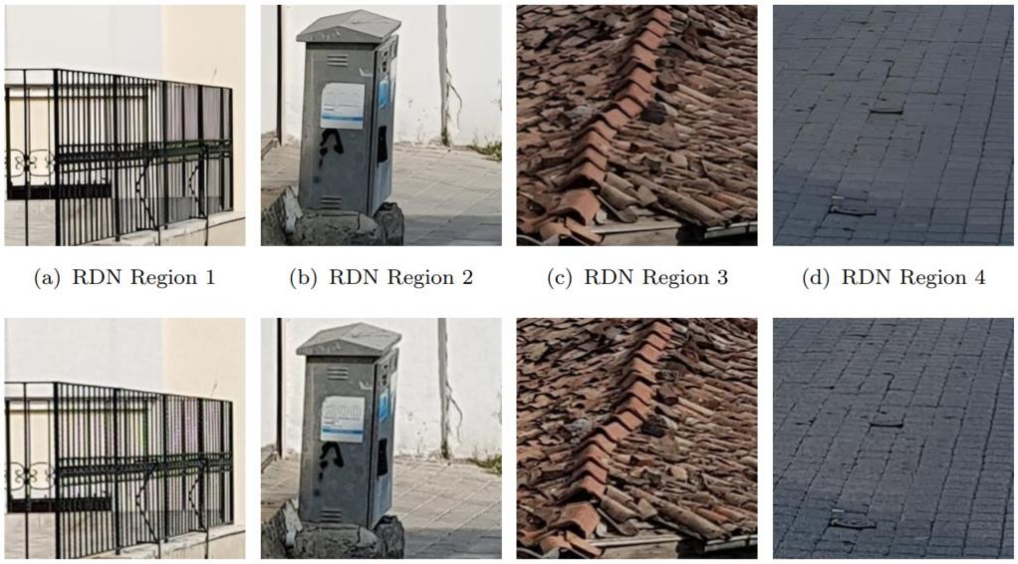
Figure 4.8: Four regions with high frequency details in RCAN Generated Result HR image and the corresponding regions in HR image (EL Ballouti 2024).

4.5 RDN

Our work shows how the method demonstrated superior performance compared to other methods for PSNR (32.47 dB) and SSIM (0.8990) performance metrics respectively. RDN effectively suppresses blurring artifacts and handles noise while recovering more details as can be seen in Figure 4.10.

Aesthetically we can see in Figure 4.10 that zooms on the chosen 4 regions and highlights the differences between the generated HR and target HR image that this method restored almost all of the high frequency details with some limited smoothing happening on the edges in the restored image that can be seen in the ground tiles (sub-Figure b and d).

Figure 4.9: RDN Generated Result HR image (EL Ballouti 2024).



(a) RDN Region 1 (b) RDN Region 2 (c) RDN Region 3 (d) RDN Region 4

(e) HR Region 1 (f) HR Region 2 (g) HR Region 3 (h) HR Region 4

Figure 4.10: Four regions with high frequency details in RDN Generated Result HR image and the corresponding regions in HR image (EL Ballouti 2024).

Training time for the model of this method took several days to train with high memory usage, which is expected given that the model had close to 22 million parameters. As a result we can see that this model performed well compared to other methods in this thesis (Table 4.2).

4.6 SRCNN

For performance metrics, this method scored 30.49 (dB) and 0.8628 respectively on PSNR and SSIM, these score are some of the lowest in these methods (Table 4.2).

Figure 4.11: SRCNN Generated Result HR image (EL Ballouti 2024).

However this is expected knowing that this method had very low memory usage and took only a few hours to train, with one of the simplest network architectures and only 57 thousand parameters. It was also one of the fastest methods in terms of inference time with results like the one in Figure 4.11.

(a) SRCNN Region 1 (b) SRCNN Region 2 (c) SRCNN Region 3 (d) SRCNN Region 4

(e) HR Region 1 (f) HR Region 2 (g) HR Region 3 (h) HR Region 4

Figure 4.12: Four regions with high frequency details in SRCNN Generated Result HR image and the corresponding regions in HR image (EL Ballouti 2024).

This method outperforms the some of the newest techniques in performance and speed (Table 4.2), but the results shown in Figure 4.12 that zooms on the chosen 4 regions and highlights the differences between the generated HR and target HR image show that this method under-performed in restoring the high frequency details with some a lot of blurring happening on the edges in the restored image.

4.7 SRGAN

In this thesis we conducted experiments on the DIV2K dataset. The experiments involved a 4× scale factor between LR and HR images. The training process involved using a random sample of images from the DIV2K database. LR images were obtained through downsampling. The optimization process utilized the MSE loss, along with VGG losses (Simonyan & Zisserman, 2015), using the Adam optimizer.

The results indicate that SRGAN under-performed some of the other methods for photo-realistic image SR which is a subjective metric (Figure 4.13), for the objective evaluation metrics this method scored 29.4 dB and 0.8472 on PSNR and SSIM respectively, these are the lowest results compared to the other results (Table4.1). Aesthetically we can see in Figure 4.14 that zooms on the chosen 4 regions and highlights the differences between the generated HR and target HR image that this method did not restore any of the high frequency details with a lot of blurring happening that can be seen in Figure 4.14. Training time for the model of this method took several days to train with high memory usage, which is expected given that the model had close to 1.5 million parameters and a very complicated network architecture, the parameters are counting only the generator network, the discriminator is only used during training. As a result we can see that this model did not perform well compared to other methods in this thesis (Table 4.2) due to the complexity of the network.

Figure 4.13: SRGAN Generated Result HR image (EL Ballouti 2024).

Figure 4.14: Four regions with high frequency details in SRGAN Generated Result HR image and the corresponding regions in HR image (EL Ballouti 2024).

4.8 VDSR

This thesis evaluated the VDSR method on DIV2K datasets. During training, images from DIV2K dataset were used and data augmentation techniques like rotation or flip were employed to stay as close as possible to the original implementation of the authors of the method. The training parameters involved using a network with a depth of 20, batches of size 64. Weight initialization follows (He, Zhang, Ren, & Sun, 2015b) method, and training lasted for 80 epochs with the learning rate decreasing by a factor of 10 every 20 epochs. An example of the generated result of this method is presented in Figure 4.15. For performance metrics, this method scored 31.35 (dB) and 0.8838 respectively on PSNR and SSIM, these score are in the middle of these methods (Table 4.2). Surprisingly, this is unexpected in comparison with the other methods, since this method had very moderate memory usage and took only a few hours to train, with a straightforward network architectures and 665

thousand parameters. It was also one of the fastest methods in terms of inference time with results like the one in Figure 4.16.

Figure 4.15: VDSR Generated Result HR image (EL Ballouti 2024).

Aesthetically we can see in Figure 4.16 that zooms on the chosen 4 regions and highlights the differences between the generated HR and target HR image that this method had some problems restoring some of the high frequency details (sub Figures a and c) with a lot of smoothing happening that can be seen in the sub Figures b and d. The inference time for this method was average compared to other methods.

Figure 16: Four regions with high frequency details in VDSR Generated Result HR image and the corresponding regions in HR image (EL Ballouti 2024).

To summarize the results of this comparative study, we compiled all the relevant results in one table. Table 4.1 presents the results of the comparative analysis of the techniques based on the PSNR and SSIM criterions, the best results for each criterion in bold font.

Table 4.1: Results of the comparative analysis of the techniques based on the PSNR and SSIM criterions.

Technique	**CARN**	**EDSR**	**ESPCN**	**RCAN**	**RDN**	**SRCNN**	**SRGAN**	**VDSR**
PSNR (dB)	32.13	32.62	30.90	**32.62**	32.47	30.49	29.40	31.35
SSIM	0.8937	0.8984	0.8645	**0.9002**	0.8990	0.8628	0.8472	0.8838

Analyzing various SR methods listed across different criteria provides valuable insights into their performance, efficiency, and applicability.

4.9 PERFORMANCE METRICS (PSNR and SSIM)

PSNR and SSIM are commonly used metrics to evaluate the quality of super-resolved images. Among the methods studied, EDSR, RCAN, RDN, and CARN exhibited higher PSNR and SSIM scores compared to to the other methods in this thesis (Table 4.1). However, it's important to note that PSNR and SSIM do not fully capture perceptual quality, especially for super-resolved images. For instance, consider the ranking of the methods based on PSNR scores from highest to lowest: RCAN, EDSR, RDN, CARN, VDSR, ESPCN, SRCNN, SRGAN. Despite ESPCN achieving lower PSNR and SSIM scores, it produced visually more appealing images even with the smoothing of some details (Figure 4.12), which is sometimes considered a desirable effect that can enhance the subjective perceptual quality. Generally speaking, a combination of high PSNR and high SSIM are a good indicator of a visually pleasing image like the results of RCAN (Figure 4.7), EDSR (Figure 4.3) and RDN (Figure 4.9).

4.10 COMPUTATIONAL EFFICIENCY

In terms of computational efficiency, SRCNN and ESPCN are relatively lightweight compared to SRGAN, EDSR, RCAN, RDN, and CARN as can be deduced by the number of parameters and memory usage of the model seen in Table 4.2. They are designed to be computationally efficient, making them suitable for real-time applications (low inference time) and resource-constrained environments. On the other hand, methods like SRGAN, RCAN, and RDN, which incorporate more complex architectures and networks as we discussed in Chapter 2, and more operations like adversarial training and attention mechanisms, tend to be more computationally intensive as the long inference times suggest.

Table 4.2: Comprehensive Performance Evaluation of all the methods.

Method	N° of Parameters	Training Time	Inference Time	Memory Usage
CARN	1.6M	few days	30-50ms	moderate
EDSR	43M	several days	100-150ms	very high
ESPCN	21K	few hours	10-20ms	low
RCAN	15.6M	several days	150-200ms	high
RDN	22.3M	several days	100-150ms	high
SRCNN	57K	few hours	15-30ms	low
SRGAN	1.5M	several days	50-60ms	high
VDSR	665K	few hours	30-40ms	moderate

4.11 TRAINING TIME

Due to their complex architectures, methods like EDSR (Figure 2.2), RCAN (Figure 2.4), SRGAN (Figure 2.7, 2.8) and RDN (Figure 2.5) have long training times and deep networks. In contrast, ESPCN (Figure 2.3) and SRCNN (Figure 2.3) are quicker to train because of their simpler architectures. Table 4.2 presents the training times for each method. Factors such as convergence speed and stability also play a role in determining training time.

4.12 MODEL COMPLEXITY

Model complexity refers to factors such as the number of parameters, network depth, and the presence of specific architectural components. SRCNN and ESPCN are relatively simple with fewer parameters and shallow networks. In contrast, methods like SRGAN, RCAN, and RDN have more complex architectures with deeper networks, residual blocks, and attention mechanisms. As Table 4.2 summarizes, the number of parameters and the architectural depth of each method is as follows:

The SRCNN model has approximately 57,000 parameters and a depth of three layers. It features a straightforward architecture consisting of three

convolutional layers, which contributes to its low complexity. As one of the earliest deep learning-based super-resolution (SR) methods, SRCNN's simplicity allows for efficient use of computational resources. However, this simplicity also restricts its capacity to capture intricate image details compared to more advanced models.

The VDSR model has approximately 665,000 parameters and a depth of 20 layers. Its architecture is characterized by a deep network with convolutional layers and skip connections, which contributes to its moderate complexity. The increased depth enables VDSR to capture more detailed image features, and the use of residual learning facilitates the effective training of such deep networks.

The EDSR model has approximately 43 million parameters and a depth that typically includes 32 layers or more. Its architecture consists of a deep residual network with wide activation, contributing to its very high complexity. By eliminating unnecessary modules such as batch normalization from conventional residual networks, EDSR achieves enhanced performance. The model's high parameter count and depth enable excellent detail restoration, albeit with increased computational demands.

The RCAN model has approximately 15.6 million parameters and features a deep architecture with multiple residual groups and channel attention blocks. It incorporates channel attention mechanisms within its residual blocks, resulting in high complexity. The channel attention allows the network to adaptively rescale channel-wise features, enhancing its ability to focus on important features. This capability leads to high-quality image reconstruction, albeit with significant computational overhead.

The RDN model has approximately 22.3 million parameters and features a deep architecture with dense connections. It combines these dense connections within its residual blocks, resulting in high complexity. By leveraging dense connectivity, RDN improves feature reuse and gradient flow, enabling the learning of more complex representations for super-resolution. This enhances performance but also increases model complexity and memory usage.

The CARN model has approximately 1.6 million parameters and a moderate depth, featuring cascading residual blocks and group convolutions. Its architecture balances performance and efficiency by capturing detailed features within a compact network structure. The cascading residual blocks help reduce the computational burden while maintaining good super-resolution quality, resulting in moderate complexity.

The ESPCN model has approximately 21,000 parameters and a shallow depth, utilizing efficient sub-pixel convolution layers in its architecture. Its low complexity and focus on efficiency result in very fast inference times and low memory usage, making it suitable for real-time applications. However, while ESPCN excels in speed and efficiency, it may not achieve the highest image quality compared to more complex models.

The SRGAN model, typically featuring around 1.5 million parameters for the generator network, employs a deep architecture comprising both generator and discriminator networks. It utilizes adversarial training with perceptual loss, resulting in moderate to high complexity. By introducing adversarial training, SRGAN enhances perceptual quality, encouraging the network to generate more realistic images.

4.13 TRADE-OFFS BETWEEN PERFORMANCE AND COMPLEXITY

There is often a trade-off between SR performance and model complexity. While more sophisticated architectures like RCAN, and RDN may achieve better results in terms of perceptual quality, they come at the cost of increased computational complexity. The choice of method depends on the specific application requirements and available computational resources (See Table 4.2), for example:

The SRCNN and ESPCN models deliver modest improvements in image quality compared to traditional interpolation methods. They are highly efficient, featuring low parameter counts and shallow networks, which makes them suitable for real-time applications and devices with limited computational

resources. Their inference time is very fast due to the fewer layers and parameters, and their training time is relatively short because of the simplicity of the models.

The VDSR model provides a significant boost in image quality over simpler models like SRCNN and ESPCN. However, it has a deep network consisting of 20 layers, which requires more computational resources. While its inference time is moderately fast, it is not as quick as SRCNN or ESPCN. The training time for VDSR is longer due to the depth of the network, necessitating more epochs for convergence.

EDSR achieves state-of-the-art results in terms of PSNR and SSIM, showcasing its superior performance in super-resolution tasks. However, its complexity is extremely high with 43 million parameters, a deep network architecture, and wide residual blocks. Consequently, the inference time is slower compared to simpler models, primarily due to the large number of parameters and depth of the network. Training EDSR is also very time-consuming, demanding extensive computational power and time investment due to its complex architecture and depth.

RCAN excels in capturing fine details and enhancing perceptual quality through the integration of channel attention mechanisms. However, its complexity is high, characterized by a substantial parameter count of 15.6 million and a sophisticated architecture that incorporates attention mechanisms. As a result, its inference time ranges from moderate to slow, reflecting the computational demands of its complex design and attention mechanisms during prediction. Training RCAN is also lengthy, requiring more iterations to optimize performance due to its intricate architecture and high parameter count.

RDN offers excellent image quality by leveraging dense connections within its residual blocks. However, its complexity is high, driven by a parameter count of 22.3 million and the intricate network depth resulting from dense connections. Consequently, its inference time is slower compared to less complex models, reflecting the detailed computations required by its architecture during prediction. Training RDN is also significant, demanding extensive

computational resources and time to effectively optimize its performance due to its complex structure and large parameter count.

CARN achieves a balance between performance and efficiency, delivering good image quality with a relatively compact model. It operates with moderate complexity, featuring 1.6 million parameters and utilizing cascading residual blocks that optimize the trade-off between network depth and parameter count. As a result, CARN offers faster inference times compared to highly complex models like EDSR and RCAN, making it suitable for applications with moderate computational resources. Training CARN is reasonable in duration, as its efficient architecture is designed to maintain performance without requiring extensive computational resources.

SRGAN does not achieve the highest PSNR or SSIM scores compared to other methods. It operates with moderate to high complexity, primarily due to its adversarial training approach, which adds to its computational demands. Despite this complexity, SRGAN maintains a moderate inference time by balancing detail restoration with computational efficiency during prediction. However, its training time can be extensive because of the adversarial training process, which involves training both the generator and discriminator networks iteratively to improve the generated image quality.

Models with similar architectures and parameter counts (e.g., SRCNN and ESPCN) exhibit comparable training and inference times. Their simplicity and efficient design contribute to quick processing. On the other hand, more complex models like EDSR, RCAN, and RDN have significantly longer training and inference times due to their deeper architectures and higher parameter counts. The presence of sophisticated components like residual blocks and attention mechanisms also increases the computational burden.

Training time is influenced by factors like convergence speed and stability. Models with advanced architectures take longer to converge but often achieve higher performance once fully trained as seen in Table 4.2. Inference time is primarily affected by the depth and complexity of the network, with more

complex models requiring more computational resources and time to process each image.

The training time for each technique is closely linked to the model's complexity. Simpler models like SRCNN and ESPCN train quickly due to their fewer parameters and shallow networks. In contrast, more complex models like EDSR and RCAN require significantly more time to train because of their depth, larger number of parameters, and additional architectural features like attention mechanisms. These complex models often necessitate more extensive computational resources and longer training periods to achieve optimal performance. However, the increased training time and complexity often result in superior image quality and better restoration of finer details as we demonstrated earlier in this chapter.

4.14 QUALITATIVE COMPARISON WITH BASELINES

Deep learning-based methods generally outperform traditional interpolation methods like bicubic interpolation in terms of visual quality and restoration of finer details (Bahattin, 2022). They can produce more realistic and visually pleasing results, especially at higher magnification factors. However, traditional methods may still have advantages in terms of computational efficiency and simplicity (Wang, Chen, & Hoi, 2020).

This comprehensive performance evaluation illustrates that while PSNR and SSIM are important metrics, they do not fully capture perceptual quality. The choice of SR method should consider the specific application requirements, balancing between computational efficiency, performance requirements, model complexity, the desired image quality, and the specific application scenarios. Each method has its strengths and limitations, and it's important to consider these factors when selecting the most suitable technique for a particular task.

CONCLUSION

In this research, an extensive comparative analysis of modern image SR methods has been carried out to clarify their respective performance, computational efficiency, and suitability for HR image reconstruction. The techniques examined include CARN, EDSR, ESPCN, RCAN, RDN, SRCNN, SRGAN, and VDSR, all trained and assessed using the standardized DIV2K dataset to ensure consistency and rigor in the methodology.

The main findings of the study showcased the superiority of certain SR techniques over others. A thorough evaluation of PSNR and SSIM metrics highlighted the outstanding performance of RCAN and EDSR in terms of fidelity to ground truth images. Additionally, RDN stood out for producing visually appealing and lifelike images, emphasizing the importance of perceptual quality in image reconstruction tasks. Further examination confirmed the effectiveness of RCAN and RDN in preserving intricate details and reducing blurring artifacts, affirming their practical usefulness. RDN, stood out for its ability to generate aesthetically pleasing results, and found favor in applications where human-centric perception of image quality is crucial.

In terms of computational efficiency, SRCNN and ESPCN emerged as models of efficiency, making them suitable for real-time applications with limited resources. On the other hand, EDSR and RCAN, while delivering commendable performance, imposed significant computational overheads, highlighting the inherent trade-off between performance and efficiency.

In terms of model complexity, we observed that SRCNN and ESPCN demonstrated optimal efficiency without compromising performance. Conversely, EDSR, RCAN, and RDN showcased intricate architectures with deep networks, offering superior performance at the expense of increased computational demands.

The choice of an appropriate SR technique depends on the specific requirements of the application. For applications requiring unparalleled image

fidelity, RCAN and EDSR are exemplary choices, despite their computational demands. Conversely, SRCNN and ESPCN emerge as practical options for real-time applications or those constrained by computational resources. RDN, with its ability to produce visually appealing outputs, is well-suited for domains where perceptual quality is paramount.

Looking ahead, potential areas of research could include integrating these methods with emerging hardware accelerators to alleviate computational burdens. Furthermore, exploring hybrid models that combine the strengths of different methods holds promise for innovative advancements. Additionally, a thorough investigation into the robustness of SR techniques across various degradation types and real-world noise scenarios could further enhance their practical utility.

In conclusion, this study has provided a comprehensive understanding of leading SR methods, offering some insights to assist in carefully selecting and improving SR models for a wide range of practical applications.

REFERENCES

Agustsson, E., & Timofte, R. (2017). *NTIRE 2017 challenge on single image super-resolution: Dataset and study. In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 1122-1131). https://doi.org/10.1109/CVPRW.2017.150

Ahn, N., Kang, B., & Sohn, K. A. (2018). *Fast, accurate, and lightweight super-resolution with cascading residual network.* (Conference session*)*. https://arxiv.org/abs/1803.08664

Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., & Gao, W. (2021). *Pre-trained image processing transformer. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 12294-12305). https://doi.org/10.1109/CVPR46437.2021.01212

Dong, C., Loy, C. C., He, K., & Tang, X. (2015). *Image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(2), 295-307*. https://arxiv.org/abs/1501.00092

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition.* (Conference session*)*. https://arxiv.org/abs/1512.03385

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification.* (Conference session). https://arxiv.org/abs/1502.01852

Hochreiter, S. (1998). *The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(2), 107-116*. https://doi.org/10.1142/S0218488598000094

Hu, Z., Zhang, J., & Ge, Y. (2021). *Handling vanishing gradient problem using artificial derivative. IEEE Access, 9, 22371-22377*. https://doi.org/10.1109/ACCESS.2021.3056232

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2018). *Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1(2),* 4700-4708. https://arxiv.org/abs/1608.06993

Keleş, O., Yılmaz, M., Tekalp, A., Korkmaz, C., & Doğan, Z. (2021). *On the computation of PSNR for a set of images or video. In 2021 Picture Coding Symposium (PCS)* (pp. 1-5). https://doi.org/10.1109/PCS50896.2021.9477434

Kim, J., Lee, J. K., & Lee, K. M. (2016). *Accurate image super-resolution using very deep convolutional networks.* (Conference session). https://arxiv.org/abs/1511.04587

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., & Shi, W. (2017). *Photo-realistic single image super-resolution using a generative adversarial network. (Conference session)*. https://arxiv.org/abs/1609.04802

Lim, B., Son, S., Kim, H., Nah, S., & Lee, K. M. (2017). *Enhanced deep residual networks for single image super-resolution.* (Conference session). https://arxiv.org/abs/1707.02921

Maral, B. C. (2022). *Single image super-resolution methods: A survey.* arXiv. https://arxiv.org/abs/2202.11763

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., & Wang, Z. (2016). *Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network.* (Conference session). https://arxiv.org/abs/1609.05158

Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition.* (Conference session). https://arxiv.org/abs/1409.1556

Timofte, R., Agustsson, E., Van Gool, L., Yang, M. H., Zhang, L., Lim, B., et al. (2017). *NTIRE 2017 challenge on single image super-resolution: Methods and results. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.* https://doi.org/10.1109/CVPRW.2017.150

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2023). *Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS),* 30, 5998-6008. https://arxiv.org/abs/1706.03762

Wang, Z., Bovik, A., Sheikh, H., & Simoncelli, E. (2004). *Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4), 600-612*. https://doi.org/10.1109/TIP.2003.819861

Wang, Z., Chen, J., & Hoi, S. C. H. (2020). *Deep learning for image super-resolution: A survey.* arXiv. https://arxiv.org/abs/1902.06068

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., & Fu, Y. (2018). *Image super-resolution using very deep residual channel attention networks.* (Conference session). https://arxiv.org/abs/1807.02758

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). *Residual dense network for image super-resolution. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2472-2481).* https://doi.org/10.1109/CVPR.2018.00261

CURRICULUM VITAE