Thèse n° 9549

Context-Aware Image Super-Resolution Using Deep Neural Networks

Présentée le 2 juillet 2021

Faculté des sciences et techniques de l'ingénieur
Laboratoire de traitement des signaux 5
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Mohammad Saeed RAD

Acceptée sur proposition du jury

Prof. P. Frossard, président du jury
Prof. J.-Ph. Thiran, directeur de thèse
Dr C. Musat, rapporteur
Dr C. Schroers, rapporteur
Prof. S. Süsstrunk, rapporteuse

École
polytechnique
fédérale
de Lausanne

2021

Let yourself be silently drawn by
the strange pull of what you really love.
It will not lead you astray.
— Rumi, 13th-century Persian poet

To my parents …

Acknowledgements

My long EPFL journey started almost 11 years ago as a bachelor student, and has been one of the most fabulous chapters in my life and filled with great experiences. During this long and exciting journey, I have met many amazing colleagues and friends, learned from them, grown up a lot, and built countless memories together. Here, I would like to take the opportunity to express my gratitude to some of these great people. This thesis would not have been possible without your sincere efforts of guidance, assistance, support, and encouragement.

First of all, my deepest gratitude goes to my Ph.D. advisor, Prof. Jean-Philippe Thiran, who made my Ph.D. journey possible by his belief in my research potential, his patience, and his scientific and moral support throughout my time at the LTS5 Laboratory. I am very grateful to him. He was not only a supervisor, but also a mentor and a friend, whose door was always open to me.

I am also very grateful to Prof. Hazim Ekenel and Dr. Behzad Bozorgtabar, who helped me a lot during my Ph.D. studies with their guidance and inspiring ideas. They supported me not only with constructive scientific discussions but especially with their positive characters and moods.

I would also like to extend my sincerest thanks to the jury members of my thesis committee, Prof. Pascal Frossard, Prof. Sabine Susstrunk, Dr. Claudiu Musat, and Dr. Christopher Schroers. I appreciate their comments on the manuscript and the fruitful discussion during the exam session.

I want to thank our wonderful secretary, Anne, for all her support and for taking care of administrative issues. Having such an amazing human being like her is one of my best memories of LTS5.

I have been lucky to be surrounded by wonderful labmates; in particular, I am very glad that I could work within the 'Vision group', a subgroup of the lab: Thank you Christophe, Damien, Marina, Christian, Remy, Gaspard, and Roser. From outside of the vision group (a less interesting part of the LTS5 of course), I would like to specifically thank Thomas, Jonathan, Francesco, Samuel, Sandra, Dimitris, Marco, David, and Ming. Thanks all of you for common lunches, coffees, nice chats, matches of 'baby-foot', and other great memories.

Along this journey, I've met many incredible people inside and outside of EPFL, who have become my best friends; Esmail, Saleh, Behnoush, Jeremy, Felipe, Farshid, Sina, Halima, Parmida, Saeed, Ludovic, and Sasan. I am so grateful to have met you guys and have you around me; you made this journey much easier and absolutely more fun. A very special thanks to my dear Azade for all her love and support; without her, reaching this point would have

been all more challenging.
I would also like to thank my oldest friends in Iran: Hesam, Ali and Sina, who proved to me that true friendship continues to grow, even over the longest distances. Thank you guys for great trips and all fun moments together.
I am really grateful to my older brothers Hossein and Mahdi for being cool and supportive ever since I can remember; especially Mahdi who without taking any credits, acted as an additional scientific advisor and helped me with all my publications, including this thesis. Last but not least, I am forever indebted to my parents for their sacrifices, unconditional love, support and encouragement in all my decisions. I am forever grateful to them. Having them standing by me makes me feel an extremely fortunate person.

Lausanne, March 24, 2021 Mohammad Saeed Rad

Abstract

Image super-resolution is a classic ill-posed computer vision and image processing problem, addressing the question of how to reconstruct a high-resolution image from its low-resolution counterpart. Current state-of-the-art methods have improved the performance of the single image super-resolution task significantly by benefiting from machine learning and artificial intelligence-powered algorithms, and more specifically, with the advent of Deep Learning-based approaches.

Although these advances allow a machine to learn and have a better exploitation of an image and its content, recent methods are still unable to constrain the plausible solution space based on the available contextual information within an image. This limitation mostly results in poor reconstructions, even for well-known types of objects and textures easily recognizable for humans.

In this thesis, we aim at proving that the categorical prior, which characterizes the semantic class of a region in an image (e.g., sky, building, plant), is crucial in super-resolution for reaching a higher reconstruction quality. In particular, we propose several approaches to improve the perceived image quality and generalization capability of deep learning-based methods by studying and exploiting the context and semantic meaning of images. To prove the effectiveness of this categorical information, we first propose a convolutional neural network-based framework that is able to extract and use semantic information to super-resolve a given image by using multitask learning, simultaneously for learning image super-resolution and semantic segmentation. The proposed decoder is forced to explore categorical information during training, as this setting employs only one shared deep network for both semantic segmentation and super-resolution tasks.

We further investigate the possibility of using semantic information by a novel objective function to introduce additional spatial control over the training process. We propose using conventional perceptual losses in a more objective way and penalizing images at different semantic levels using appropriate loss terms by benefiting from our new OBB (Object, Background, and Boundary) labels, generated from segmentation labels. We demonstrate that our proposed method produces more realistic textures and sharper edges compared to other state-of-the-art algorithms.

Then, we introduce a new test time adaptation-based technique to leverage high-resolution images with perceptually similar context to a given test image to improve the reconstruction quality. Contrary to perceptually driven approaches, we show that this approach generates images with both greater perceptual quality and minimal changes to the PSNR/SSIM with respect to the benchmark. We further validate this approach's effectiveness by using a novel numerical experiment analyzing the correlation between filters learned by our network and what we define as "ideal" filters.

Finally, we present a generic solution to enable adapting all our previous contributions in this thesis, as well as other recent super-resolution works trained on synthetic datasets, to "real-world" super-resolution problem. Real-world super-resolution refers to super-resolving images with real degradations caused by physical imaging systems, instead of low-resolution images from simulated datasets assuming a simple and uniform degradation model (i.e., bicubic downsampling). We study and develop an image-to-image translator to map the distribution of real low-resolution images to the well-understood distribution of bicubically downsampled images. This translator is used as a plug-in to integrate real inputs into any super-resolution framework trained on simulated datasets.

We carry out extensive qualitative and quantitative experiments for each mentioned contribution, including user studies, to compare our proposed approaches to state-of-the-art methods.

Keywords: *super-resolution, neural network, deep learning, generative adversarial networks, computer vision, image processing*

Résumé

La super-résolution est un problème classique de traitement d'image qui aborde la question de comment reconstruire une image haute résolution à partir de son homologue basse résolution. Les méthodes de l'état de l'art actuelles ont considérablement amélioré les performances de la reconstruction en super-résolution grâce à l'apprentissage automatique et aux algorithmes basés sur l'intelligence artificielle, et plus particulièrement avec l'avènement des approches basées sur l'apprentissage profond.

Bien que ces avancées permettent à une machine d'apprendre et de mieux exploiter une image et son contenu, les méthodes récentes sont encore incapables de contraindre l'espace de solutions plausibles en fonction des informations contextuelles disponibles dans une image. Cette limitation se traduit principalement par de mauvaises reconstructions, même pour des types d'objets et de textures très connus, facilement reconnaissables par l'homme.

Dans cette thèse, nous visons à prouver que l'information préalable, qui caractérise la classe sémantique d'une région dans une image (par exemple, ciel, bâtiment, plante), est cruciale pour atteindre une qualité de reconstruction supérieure. En particulier, nous proposons plusieurs approches pour améliorer la qualité de reconstruction perçue et la capacité de généralisation des méthodes basées sur l'apprentissage profond en exploitant le contexte et la sémantique des images. Pour étudier l'efficacité de ces informations caractéristiques, nous proposons tout d'abord une solution basée sur un réseau de neurones convolutif qui bénéficie d'un objectif d'apprentissage supplémentaire au cours de son processus d'apprentissage. Nous concevons un décodeur super-résolution capable d'extraire et d'utiliser des informations sémantiques pour reconstruire une version en super-résolution d'une image donnée en utilisant l'apprentissage multitâche, simultanément pour la super-résolution et la segmentation sémantique. Le décodeur proposé est obligé d'explorer les informations caractéristiques pendant l'entraînement, car ce paramètre n'utilise qu'un seul réseau profond partagé pour les tâches de segmentation sémantique et de super-résolution.

Nous étudions en outre la possibilité d'utiliser des informations sémantiques via une nouvelle fonction de coûts, pour introduire un contrôle spatial supplémentaire sur le processus d'apprentissage. Nous proposons d'utiliser les *perceptual losses* de manière plus objective et de pénaliser les images à différents niveaux sémantiques en utilisant des coûts appropriés en bénéficiant de nos nouveaux labels *OBB (Object, Background, and Boundary)*, générés à partir des labels de segmentation. Nous démontrons que notre méthode proposée produit des textures plus réalistes et des bords plus nets par rapport à d'autres algorithmes de l'état de l'art.

Ensuite, nous introduisons une nouvelle technique basée sur des adaptations durant le test pour tirer parti des images haute résolution avec un contexte perceptuellement similaire à une image de test donnée, afin d'améliorer la qualité de reconstruction. Contrairement aux approches axées sur la perception, nous montrons que cette approche génère des images avec à la fois une plus grande qualité perceptuelle et des changements minimes du PSNR/SSIM par rapport au résultat de référence. Nous validons encore l'efficacité de cette approche en utilisant une nouvelle expérience numérique analysant la corrélation entre les filtres appris par notre réseau et ce que nous définissons comme des filtres "idéaux".

Enfin, nous présentons une solution générale pour permettre d'adapter toutes nos contributions précédentes dans cette thèse, ainsi que d'autres travaux récents de super-résolution basés sur des données synthétiques, au problème de super-résolution "du monde réel". La super-résolution du monde réel fait référence à des images super-résolution avec des dégradations réelles causées par des systèmes d'imagerie physique, au lieu des images basse-résolution simulées à partir d'ensembles de données supposant un modèle de dégradation simple et uniforme (par exemple, un sous-échantillonnage bicubique). Nous étudions et développons une méthode de traduction d'image à image pour faire correspondre la distribution des images réelles à basse-résolution à la distribution connue des images bicubiquement sous-échantillonnées. Ce traducteur est utilisé comme un plug-in pour intégrer des entrées réelles dans toutes les méthodes de super-résolution entraînées sur des bases de données simulées.

Nous réalisons des expériences qualitatives et quantitatives approfondies pour chaque contribution mentionnée, y compris des tests utilisateurs, afin de comparer nos solutions proposées aux méthodes de l'état de l'art.

Mots clés : *super-résolution, réseau de neurones, apprentissage profond, réseaux antagonistes génératifs, vision par ordinateur, traitement d'image*

Contents

Acknowledgements **i**

Abstract (English/Français) **iii**

List of Figures **xiii**

List of Tables **xvii**

1 Introduction **1**
1.1 Motivation 1
1.2 Context-aware image super-resolution 4
1.3 Main contributions 5
1.4 Thesis outline 6

2 Brief Image Super-Resolution Review **9**
2.1 Problem formulation 9
2.2 CNN-based super-resolution 12
2.2.1 Deep architectures for SR 13
2.2.2 Objective functions 16
2.2.3 Generative adversarial networks in SR 20
2.3 Datasets 23
2.4 Evaluation metrics 26
2.4.1 Full-reference distortion measures 26
2.4.2 Learning-based metrics 26
2.4.3 User study 27
2.5 Towards real-world super-resolution 28
2.5.1 The challenge 28
2.5.2 Multiple-degradation handling approaches 29

3 Extracting Image Context by Multi-Task Learning **31**
3.1 Introduction 32
3.1.1 Does semantic information help? 33
3.1.2 Our contribution 34
3.2 Related work 35
3.2.1 Single image super-resolution 35

3.2.2 Super-resolution faithful to semantic classes 36
3.3 Multitask learning for image super-resolution 36
3.3.1 Architecture 37
3.3.2 Boundary mask 38
3.3.3 Loss function 39
3.3.4 Dataset 40
3.3.5 Training and parameters 40
3.4 Results and discussion 41
3.4.1 Effectiveness of boundary masks 41
3.4.2 Qualitative results 41
3.4.3 User experience 42
3.4.4 Ablation study 44
3.4.5 Results on standard benchmarks 45
3.4.6 Computational time 46
3.5 Conclusion and future work 47

4 Spatial Control Over Image Generation Process 49
4.1 Introduction 50
4.2 Related work 52
4.3 Methodology 53
4.3.1 Targeted perceptual loss 54
4.3.2 OBB: Object, background and boundary label 57
4.3.3 Architecture 57
4.4 Experimental results 58
4.4.1 Dataset and parameters 58
4.4.2 Qualitative results 59
4.4.3 Quantitative results 60
4.4.4 Inference time 64
4.5 Conclusion 64

5 Test-Time Adaptation Based on Perceptual Similarity 65
5.1 Introduction 66
5.2 Overview of the approach 69
5.3 Image activations in SR 71
5.4 Overfitting: the good, the bad, and the ugly 72
5.5 Experiments and results 73
5.5.1 Experimental settings 73
5.5.2 Filter selection analysis 75
5.5.3 Comparison to PSNR-based approaches 75
5.5.4 Comparison to perceptual-based approaches 76
5.5.5 Inference time 77
5.6 Conclusion 77

6 Integrating into Real-World SR **79**
6.1 Introduction 80
6.2 Related work 83
6.2.1 Real-World SR through real data 83
6.2.2 Real World SR through extended models 84
6.3 Methodology 85
6.3.1 Overall pipeline 85
6.3.2 Bicubic look-alike image generator 86
6.3.3 SR generator 89
6.3.4 Training parameters 89
6.4 Experimental results 90
6.4.1 Test images 90
6.4.2 Quantitative results 92
6.4.3 Qualitative comparison 93
6.4.4 User study 93
6.5 Conclusion 93

7 Conclusion **95**
7.1 Thesis summary 95
7.2 Limitations 96
7.3 Future work 97

A Appendix **99**
A.1 Related to Chapter 4 99
A.1.1 Results on standard benchmarks 99
A.1.2 Details of the user study 100
A.2 Related to Chapter 5 105
A.2.1 Training details 105
A.2.2 Effect of different convolutional layers 106
A.2.3 Best values for K and M 107
A.3 Related to Chapter 6 109
A.3.1 Generalization capabilities of the bicubic look-alike generator 109
A.3.2 Quantitative results 109
A.3.3 Ablation study 111
A.3.4 Computational cost 111

Bibliography **113**

Curriculum Vitae **125**

List of Abbreviations

Abbreviation	Description
SR	Super-resolution
SISR	Single image super-resolution
HR	High-resolution
LR	Low-resolution
DL	Deep learning
CNN	Convolutional neural network
NN	Neural network
ISP	Image signal processor
GAN	Generative adversarial network
WGAN	Wasserstein generative adversarial network
SD	Standard-definition
HD	High-definition
MAE	Mean absolute error
MSE	Mean squared error
PSNR	Peak signal to noise ratio
SSIM	Structural similarity index measure
LPIPS	Learned perceptual image patch similarity
NIQE	Naturalness image quality evaluator
PI	Perception index
MOS	Mean opinion score
DSLR	Digital single-lens reflex
CCD	Charge-coupled device
CMOS	Complementary metal-oxide-semiconductor
OBB	Object, boundary and background
RF	Reference

List of Figures

1.1 An illustration of importance of image resolution for TV application. 1
1.2 An illustration of image super-resolution decoder and the observation model. . 2
1.3 An example of super-resolving an image with standard definition (SD) resolution; left: image upsampled by common TV technologies, right: image reconstructed by our approach (SROBB, Chapter 2). 3
1.4 An example of context-aware image super-resolution. 4

2.1 Block diagram of SR observation model. 10
2.2 Examples of linear SR designs with: (a) Pre-upsampling, (b) Post-upsampling methods. 13
2.3 The design of a residual network. 14
2.4 Architecture of DRCN. 15
2.5 Architecture of LapSRN. 15
2.6 RCAN Channel attention 16
2.7 Comparing $L1$ and $L2$ losses. 18
2.8 Optimization with perceptual losses. 19
2.9 VGG architecture; visualzing low to high-level features. 19
2.10 The structure of GANs, illustrating the training procedure of an SR decoder alongside a discriminator. 21
2.11 The network architecture of the discriminator proposed by SRGAN 21
2.12 Some examples of test images from seminal datasets. 23
2.13 DPED dataset, (a) Setup used to capture images synchronously by different hardware, (b) Examples of images from the same scenes, taken by four different cameras. 25
2.14 The registration process of RealSR dataset; creating low-resolution and high-resolution image pairs by changing the focal length of the camera. 25
2.15 The pipeline of conventional supervised approaches for image super-resolution. 28
2.16 The real-word super-resolution challenge. 29

3.1 The proposed single image super-resolution using multitask learning. 33
3.2 Architecture of the decoder: training the SR decoder in a multitask manner by introducing a segmentation extension. 37

3.3 An example showing the accuracy and resolution of a pixel-wise semantic segmentation label of a low resolution image. 38
3.4 The boundary mask generation. 39
3.5 Two examples of segmentation prediction results. 41
3.6 Comparing the effect of segmentation mask 42
3.7 Qualitative results on COCO-stuff dataset, focusing on object/background textures. 43
3.8 An example of a bad reconstruction of boundaries compared to the SRGAN . . 44
3.9 The evaluation results of our user studies, comparing SRSEG (our method) with SRGAN. 45
3.10 Ablation study on different type of objects/backgrounds. 45
3.11 Sample results on the “baby” and “baboon” images from Set5 and Set14 datasets. 46

4.1 The illustration of the proposed targeted perceptual loss. 51
4.2 The effect of choosing different CNN layers to estimate the perceptual loss on different regions of an image. 55
4.3 Constructing an OBB label. We assign each area to one of the “Object”, “Background” or “Boundary” classes based on their initial pixel-wise labels. 57
4.4 Schematic diagram of the SR decoder. 58
4.5 Sample results on the “baby” and “baboon” images from Set5 and Set14 datasets. 59
4.6 Qualitative results on a subset of the COCO-Stuff dataset images. 61
4.7 The results of the user study. 63
4.8 The results of the ablation study. 63

5.1 Improving the SR perceptual quality by test-time adaptation. 67
5.2 The effect of fine-tuning on images which maximally activate specific filters in a pre-trained classification network. 68
5.3 The overview of the proposed method. 70
5.4 Top 9 activated images from a subset of 50 thousand images. 71
5.5 The effects of fine-tuning as a function of the number of epochs. 72
5.6 Illustration of how overfitting guides the SR network to a local minimum with superior perceptual quality. 73
5.7 The average correlations of the filters of the final layer of G' and G_{rand} to the filters of G_{per} as a function of the number of epochs of fine-tuning. 74
5.8 Qualitative comparison to PSNR-based approaches. 76
5.9 Comparing the proposed method and a perceptual-based approach. 78

6.1 An example SR produced by our system on a real-world LR image, for which no higher resolution/ground-truth is available. 81
6.2 Downsampling kernels estimated patchwise on a RealSR LR image and the same image bicubically downsampled from the HR image. 82
6.3 The proposed two-step pipeline for real world SR. 85
6.4 Schematic diagram of the bicubic-alike decoder. 86

6.5 The effectiveness of using bicubic perceptual loss. 88
6.6 Example images generated with and without the copying mechanism during training. 89
6.7 Qualitative results of ×4 SR on a subset of the DIV2k, RealSR HR, TV Streams, and DPED cell-phone images. 91
6.8 Results of the user study comprising forty one people. 92

A.1 Qualitative results on random images from BSD100 using bicubic interpolation, SRGAN, SROBB (ours). 101
A.2 Qualitative results on the images from Set5. 102
A.3 Qualitative results on the images from Set14. 103
A.4 Example screenshot of our online survey. 104
A.5 The network architecture of the generator. 105
A.6 Differences in the perceptual quality obtained at different VGG network layers. 106
A.7 Differences in the perceptual quality obtained with different combinations of K and M. 108
A.8 Comparison results of RCAN and ESRGAN methods on original images from the RealSR dataset and our transformed LR images. 110

List of Tables

4.1 Comparison of bicubic interpolation, LapSRN, SRGAN and SROBB for the “baby” and “baboon” images from Set5 and Set14 test sets. 62

5.1 Comparison LapSRN, RCAN, EDSR, and activated-SR (ours) on various test sets. 77

A.1 Comparison of bicubic interpolation, SRCNN, SelfExSR, LapSRN, SRGAN, and SROBB on the Set5 and Set14 test sets. 100
A.2 The average of decision making duration [seconds] for users to choose the reconstructed images of each method. 104
A.3 Comparison of bicubic interpolation, SRResNet, RCAN, EDSR, DPSR, RealSR, and RBSR on different presented test sets. 110
A.4 Comparing the effect of each proposed component of the bicubic look-alike generator on LR and HR images of RealSR test set. 111

1 Introduction

1.1 Motivation

The quality of digital images is characterized by several parameters such as ambient noises, artifacts, and motions; however, the primary and most important parameter affecting the visual quality is the image resolution. Resolution is simply the amount of detail that an image holds; it describes the image size as the number of pixels that it contains. The term "higher resolutions" refers to having more pixels per centimeter, resulting in higher-quality and more pleasant images. A lower resolution image has fewer pixels, and if those few pixels are shown on a large screen, they become stretched and visible. To better understand this parameter's importance, an illustration of higher and lower resolution images is shown in Figure 1.1. In this Figure, the 4K quality image provides details and contains sharp edges. Going towards the SD image, fine details gradually disappear with the reduction of the number of pixels, and square pixels become visible.

Due to the vital uses of digital images in many real-life computer vision applications such as medical imaging, video surveillance, security, robotics, radar imaging systems, and media, the

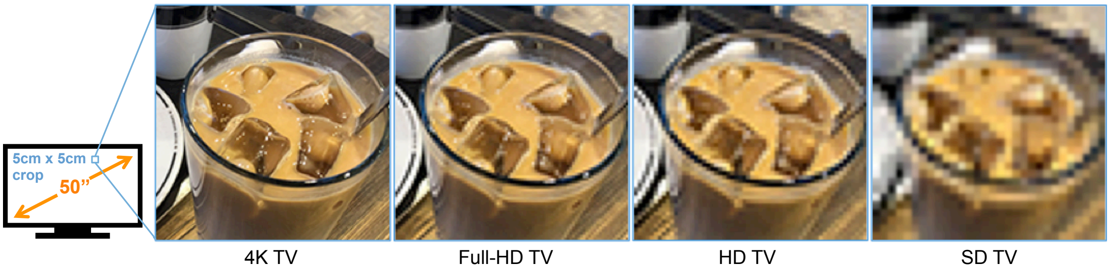


Figure 1.1 – An illustration of importance of image resolution for TV application. We show an example of content appeared in a 5 × 5 centimeters crop of a 127 centimeters (50 inches) TV (supporting up to 4K resolution), when the original content, e.g. streaming, have the following resolutions: 4K (3840 × 2160), Full-HD (1920 × 1080), HD (1280 × 720), and SD (640 × 480). Zoom in for the best view.

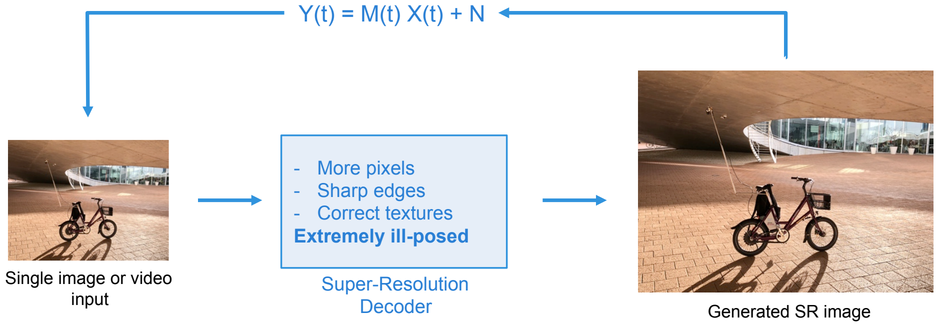


Figure 1.2 – An illustration of image super-resolution decoder and the observation model. Y represents the captured image (LR), M is the overall transformation matrix, X represents the real high-resolution scene (HR image), N is the noise naturally added during the acquisition process, and the variable t represents the timestamp of the captured image.

challenge of enhancing the resolution of images or, i.e., capturing more information is in great demand.

Currently, image quality is mostly limited to imaging hardware's capability and deficiencies, such as the quality and type of the sensor, sensor size, lenses, limitation in storage, etc. The sensor type mostly varies between Charge-Coupled Device (CCD) and Complementary Metal-Oxide-Semiconductor (CMOS); these technologies have their differences in how they capture each frame. However, increasing the image resolution is done similarly for both: 1- Increasing the sensor's spatial size to include more pixels, or 2- Decreasing the size of each pixel to have a higher number of pixels in a specific area. Increasing the sensor size is feasible and even introduces more advantages such as allowing manufacturers to offer wider ISO ranges while keeping noise low. However, it results in physically bigger devices and is not practical for many applications, e.g., smartphones. Decreasing pixel size is also challenging as it requires sophisticated manufacturing technologies resulting in higher development costs. Moreover, smaller pixels may not directly result in a higher resolution; there is a principal limit to the resolution of an optical system due to the physics of diffraction (Airy disk), which directly affects the maximum sampling frequency. Also, using smaller pixel decreases the amount of light that reaches a corresponding cell of the pixel on the sensor and increases the shot noise.

An alternative for improving imaging hardware is focusing on software improvement. This solution is based on designing algorithms, either capable of encoding information using fewer bits than the original representation (compression) or proposing methods to enhance the resolution accurately by generating/reconstructing missing information, namely **Super-Resolution (SR)** and the main focus of this thesis.

SR technology, benefiting from the new advances in computing units, such as graphic processing unit (GPU) and image signal processor (ISP), provides a promising computational imaging approach to increase the spatial resolution of images and generated higher resolution outputs with various scaling factors. In particular, as illustrated in Figure 1.2, SR aims at solving the

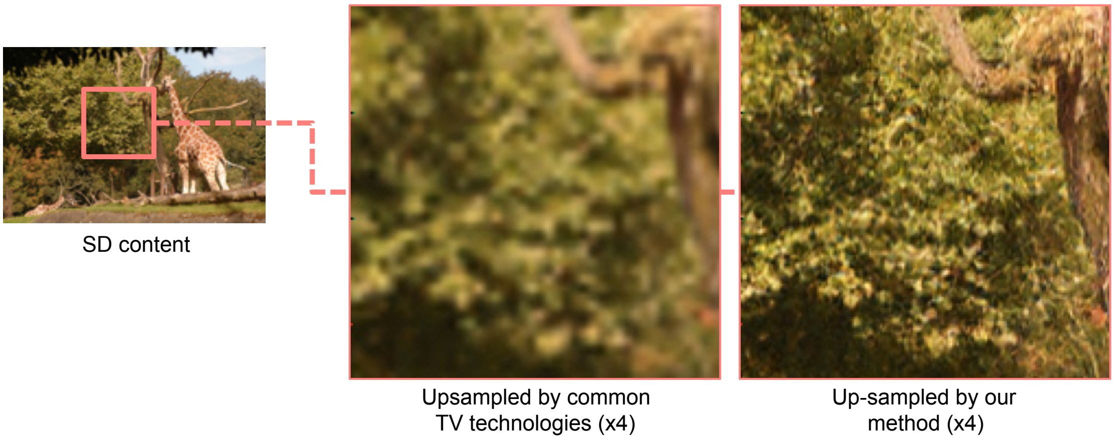

SD content
Upsampled by common TV technologies (x4)
Up-sampled by our method (x4)

Figure 1.3 – An example of super-resolving an image with standard definition (SD) resolution; left: image upsampled by common TV technologies, right: image reconstructed by our approach (SROBB, Chapter 2).

problem of recovering a high-resolution (HR) image from a low-resolution input image (LR) or video sequence. This is a classic ill-posed problem that has been one of the most active research areas since the work of Tsai and Huang [1] in 1984. Such software-based solution is common and widely used in computer vision applications such as ultra-high definition TVs, security and surveillance, low-resolution face recognition, remote sensing, medical imaging and any other application in which more image details are required on demand. In Figure 1.3, an example of SR application for TV technologies and media is shown.

The SR task, like many other fields of computer vision such as pose-estimation, image inpainting, object detection, semantic segmentation, and etc., have been revolutionized by the introduction of Deep Learning (DL). DL is a new part of the family of machine learning methods. DL benefits from the strong capacity of neural networks to learn the hierarchical representations of data/images. DL-based methods have proven significant superiority over other classic machine learning and image processing algorithms in recent years.

Despite these advances in image SR, there are still many challenging open topics for convolutional neural network (CNN)-based SR approaches, e.g., new architectures, new objective functions for single and multi-frame decoders. One of the less investigated yet critical challenges of SR is the reconstruction faithful to the context and categorical priors. Despite the strong ability of CNN-based methods to exploit this information, we observed minimal influence of images' global context on the recent SR reconstruction results. In the following section, we first discuss this limitation in more detail, alongside its different challenges. Finally, we summarize how we address this challenge in the scope of this thesis.

For this Ph.D. thesis, we aim to propose fast, accurate, and robust SR approaches with an additional focus on studying and benefiting from the local and global context of images to increase the perceptive reconstruction quality.

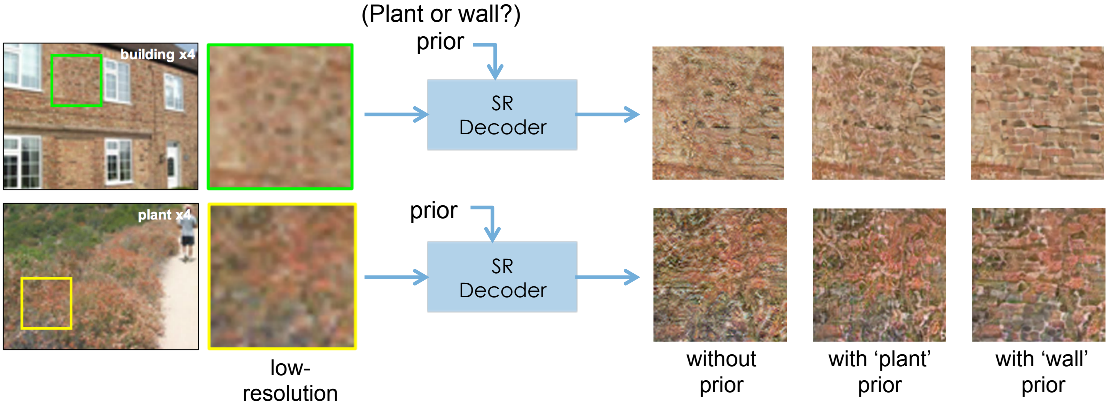


Figure 1.4 – Context-aware image super-resolution. In this example, the low-resolution crop of both 'plant' and 'wall' are similar to each other and their reconstructing without considering this prior, or considering the wrong prior, results in unrealistic textures. The images are generated by [2]. Zoom in for the best view.

1.2 Context-aware image super-resolution

There are many exciting challenges to face when developing SR methods for real-life applications. As humans, just by looking at a scene, our brains can quickly have an understanding of it; we see what each region of the image is representing, what kind of objects are available, even the texture of the materials are mostly guessed or recognized by us. Although recent machine learning and computer vision advances, particularly DL-based methods, now allow a machine to learn and have a better exploitation of image contents, unlike humans, SR methods still cannot fully exploit the contextual information of a scene. It mostly results in poor reconstructions, even for very well-known types of objects and textures, intuitively easy to reconstruct, e.g., a car, a tree, or fabrics.

A visual example of this limitation is shown in Figure 1.4. In this example, a CNN-based SR generator is trained in a supervised manner and under different settings; in the first attempt, the generator is trained in a conventional way, and without any pre-knowledge about categorical information; then, it was trained by having an additional segmentation map at the input level, specifying the category of regions expected in the output image. Looking at the test images in this example, we see that low-resolution crops of both 'plant' and 'wall' are visually very similar. One could not recognize them without looking into the context around them. By this experiment, we observe that their reconstruction results without considering this prior, or considering the wrong prior, results in unrealistic textures. This experiment confirms that, although the CNN-based generator had access to thousands of images of both trees and walls, it could not learn and benefit from more global contextual information within images in a conventional way.

Few studies such as [3, 4, 5] benefit from prior information for SR. Most recently, [2] uses an additional segmentation network to estimate probability maps as prior knowledge and uses them in the existing SR networks. Their approach recovers more realistic textures faithful to categorical priors; however, as an external segmentation network is required at the run-time, their method is computationally expensive and not practical for real-life applications. Another limitation of their method is relying on the segmentation network's performance and recognizing only a few categories of objects and backgrounds.

In this thesis, we aim to study and address some of the significant limitations of current SR approaches to benefit fully from the context within images and introduce a novel methodology and practical SR systems, which have a promising potential to be used in a large spectrum of real-life applications. We study and develop several such approaches that can be summarized as three main objectives:

- To develop a real-time SR generator capable of benefiting from categorical information of input images to improve its reconstruction quality and with minimal affect on the final computational cost and inference time of conventional approaches.
- To go one step further and benefit from external resources/images with perceptually similar contents and context to reach superior perceptual qualities.
- To propose an efficient way of extending all previous ideas to be compatible with real-world SR setting, where real images with real degradations need to be addressed instead of artificially downsampled images by a uniform degradation, i.e., bicubic downsampling kernel, commonly used in recent SR works.

In the next section, we summarize our main contributions to reach each mentioned objective.

1.3 Main contributions

The main contributions of this thesis focus on machine learning algorithms for single image super-resolution (SISR) on RGB digital images. Each proposed contribution addresses part of the challenges and objectives mentioned in Section 1.2. In this section, we summarize our main contributions and the essence of these methods:

- We propose an end-to-end and easily reproducible framework that uses the concept of multi-task learning during the training process to learn a CNN-based SR model in a content-aware manner. Unlike previous work, the proposed method does not require any prior information at the test time; therefore, its complexity remains practical for real-time applications.
- We present a novel objective function for learning image SR, which enables additional spatial control over reconstructing different regions of an image based on their own categorical information.

- We introduce a new method based on test time adaptation to the SR community. This contribution leverages from images with similar contextual information to the test image to reach a new perceptual quality level without significantly impacting the distortion metrics such as PSNR or SSIM.

- We develop a generic CNN-based image-to-image translation network to integrate all context-aware and other recent SR works into the real-world SR setting. We introduce a “bicubic look-alike generator” that aims to map the distribution of real LR images to the well-understood distribution of bicubically downsampled LR images. This generator is used as a plug-in to adapt any SR method to real inputs (images with real degradations caused by physical imaging systems.)

- To examine the effectiveness of the proposed frameworks and methods, extensive qualitative and quantitative evaluations on both simulated and real data were conducted. In this thesis, we designed and conducted user studies with more than 70 participants to overcome the known reliability issue of quantitative measures in the SR task and prove the superiority of our proposed methods in terms of reconstructing more appealing images for humans.

1.4 Thesis outline

The remainder of this thesis is organized as follows:

- Chapter 2 presents some necessary backgrounds and mathematical formulation of SR problem and conducts a brief review of the both conventional and state of the art SR methods, alongside their important contributions.

- Chapter 3 proposes a novel SR framework based on multitask learning for both image SR and semantic segmentation tasks. This chapter also discusses the importance of semantic information within the SR task and introduces a boundary mask to discard irrelevant information during the learning process.

- Chapter 4 introduces a new objective function for SR task, namely “Targeted perceptual loss”, to give learning-based SR methods a meaningful spatial control over the image generation process. Moreover, in this chapter, we study perceptual losses in general and learn about their nature, e.g., how a pre-trained CNN-based network sees an image.

- Chapter 5 further investigates how overfitting/fine-tuning on some selected images can be beneficial and change SR reconstructions for better or worse. In this chapter, we finally introduce a novel numerical experiment in the field of SR, where we quantitatively judge the learned weight and biases of a CNN-based network based on what we call “ideal” filters.

- Chapter 6 consists of our solution to integrate all our context-aware SR contributions (chapter 2 to 6) and other states of the art SR work into the real-world SR setting, where the downsampling kernel is real and not uniform (such as bicubic). In this chapter, we also introduce “bicubic perceptual loss” which enables building such a framework.
- Finally, in Chapter 7, a brief summary of the thesis, its limitations, and future directions conclude this thesis.

2 Brief Image Super-Resolution Review

Image super-resolution (SR) has a long history in computer vision and image processing. With the advent of Deep Learning (DL)-based approaches and the possibility of using powerful hardware and large datasets to train such methods, this field has become a hot topic in computer vision. As all the proposed methods in this thesis involve Neural Networks (NNs) to some extent, the majority of this chapter focuses on reviewing NN-based approaches (Section 2.2), including some conventional work, as well as important contributions of state of the art SR methods. Before describing the NN-based background, in Section 2.1, we present the problem formulation and some of its mathematical notations. In Section 2.3 and 2.4, we briefly describe some of the conventional image datasets and evaluation metrics used in recent SR work. Finally, we review the recent real-world SR problem related to real image degradation in Section 2.5.

2.1 Problem formulation

The ultimate goal of SR methods is to recover the HR image corrupted by the limitations of the optical imaging hardware; this is a typical example of an inverse problem where the original image (HR) is reconstructed from the available stored data (LR image). The forward model of this inverse problem can be summarized as a simple linear model; in the literature, its most common form, by far, is defined as:

$$Y(t) = M(t)X(t) + N, \tag{2.1}$$

where Y is the captured image (mostly considered as LR image in this thesis), M is the overall transformation matrix representing the imaging system, X represents the real high-resolution scene (HR image), and N represents the deterioration by the white Gaussian noise, created during the acquisition process. In this formula, the variable t represents the timestamp of the captured image. All images in this formula are in the vector form.

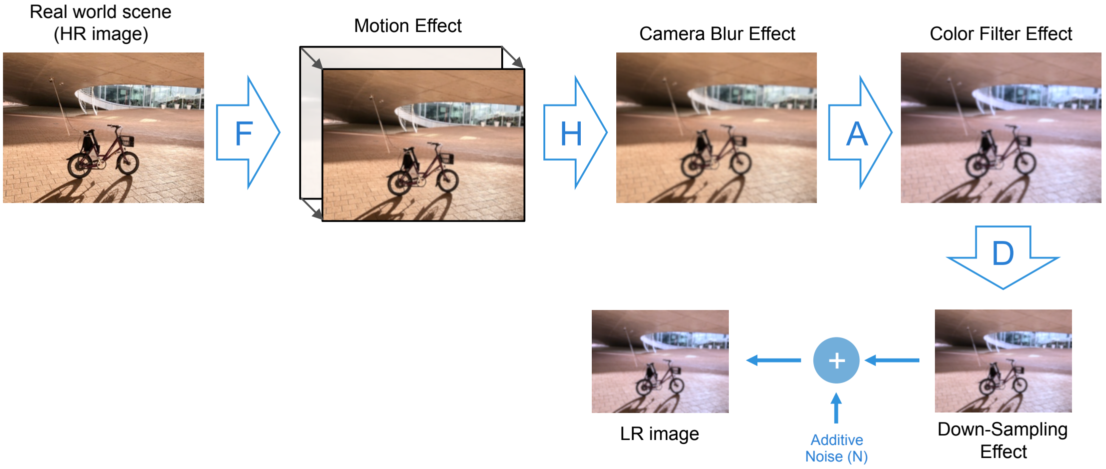


Figure 2.1 – Block diagram of SR observation model. The low-resolution (LR) images are the blurred, warped, decimated, and noisy version of a high-resolution (HR) image.

To construct the linear model M, we consider the three main aspects of the image formation process, motion, optical blur, and the sampling process, and formulate it as follows:

$$M = DAHF, \tag{2.2}$$

where F represents the intensity conserving, geometric warp operation capturing image motion, H is the blurring operation due to the imaging system's response to a point source or point object, namely optical Point Spread Function (PSF), D represent downsampling operation. Finally, A corresponds to the color filter effects (sampling operations specific to the color space).

We emphasize that, although the significant advances in convolutional neural networks and their way to solve inverse problems in recent years made the need for an accurate formulation of the forward model less important, a better understanding of the problem and the forward model is still crucial to be able to propose new directions and ideas to address the SR problem. To this end, the general pipeline of the observation model M is shown in Figure 2.1. In the following, we present each of these processes in more detail:

Motion (F) it is one of the most important causes of degradation in the image-capturing process as it is one of the main reasons for creating a blur effect in the final image. This effect -namely motion blur, happens when either one or more existing objects, or the camera itself, move significantly during integration time and result in a smeared image. The integration time (t), also known as the shutter speed, is the time during which the camera sensor can capture light. The electronic shutters significantly reduce this effect but the issue is still noticeable in case of severe movements.

This motion can be simplified and purely formulated as a 2D affine motion model, as all 3D motions of objects/the scene finally induce a 2D motion in the image. The resulted image by motion blur (I^F) can be formulated as [6]:

$$I^F(x,y) = \int_0^{\tau=t} I^{org}(u_\tau, v_\tau)d\tau, \text{ where } \begin{pmatrix} x \\ y \end{pmatrix} = F(b)_0^\tau \begin{bmatrix} u_\tau \\ v_\tau \end{bmatrix}, \tag{2.3}$$

where $F(b)_0^\tau$ is representing the estimated motion of the object/scene in t and b being the parameter chosen to describe the image transform corresponding to the object 3D motion.

Optical blur (H) this category of blurring effect is related to the physics behind the camera hardware, creating the image formation process, such as the physical lenses and their thickness. This blur can also happen by the camera being defocused. In the literature [6, 7, 8], H is mostly modeled by 2D Gaussian blur as a first approximation.

As presented in [9], the atmosphere also adds an additional blurring effect and motion blur into the observation model M. Taking this effect ($H_{atmosphere}$) into account, M can be reformulated in a more complex and accurate form as:

$$M^{'} = DAHFH_{atmosphere}, \tag{2.4}$$

The $H_{atmosphere}$ effect is usually neglected in the literature [7] as it has a much less significant impact on the captured image comparing to the H effect caused by the optical cameras and imaging hardware themselves.

Sampling process (DA) this process can be divided into downsampling operation D and color filter effect A, to distinguish between a down-sampling operation by the camera charge-coupled device (CCD) array (by a scale factor of s) and the sampling operations specific to the color space and creating the color image. CCD sensors are the primary technology used in the digital imaging community. The final image resolution is determined by the characteristics of the CCD array of the camera.

After building the forward model, explicitly or implicitly, a cost function needs to be defined in order to finally estimate the original image/scene X. This definition is the main key to assuring certain fidelity of how the final constructed image is close to the measured data. Historically, based on algebraic or a statistical point of view, different cost functions have been introduced; to our knowledge, the most common function is the least-square error, which minimizes the $L2$ norm of the difference between the solution and the measured data. Based on this definition, the reconstructed image can be defined as:

$$\widehat{X} = argmin_X \|Y - MX\|_2^2, \tag{2.5}$$

This formulation provides the maximum likelihood of estimating the original image X for the scenario where N is an additive white, zero-mean Gaussian [10].

To solve the SR problem, many classic works investigated different components of the observation model in detail and proposed various approaches based on trees structures, principal component analysis, projections, gradient profiles, etc. However, the imprecise and complex formulation of the observation model that maps the LR space into the HR space and the inefficiency of forming this high-dimensional mapping, make these methods incapable of reaching photo-realistic quality reconstructions.

Besides these methods, interpolation-based methods aim to produce the HR image by assuming that the observed low-resolution image is a direct downsampled version of the HR image. These methods, such as bicubic interpolation and Lanczos resampling, are extremely simple and fast. However, they suffer from severe blurring effects and the lack of fine texture details. More powerful methods utilizing statistical image priors were also proposed to restore fine structures; however, they are incapable of modeling complex and varying natural image contexts.

The following section reviews CNN-based SR approaches and explains why this branch of machine-learning approaches is becoming the dominant method for SR applications and is the primary focus of this thesis.

2.2 CNN-based super-resolution

DL is a relatively new subset of the vast family of machine learning methods based on artificial neural networks, aiming to learn the hierarchical representations of data. By benefiting from the strong capacity of neural networks to address substantial unstructured data, DL-based methods have proven significant superiority over other machine learning algorithms in various AI fields such as computer vision [2], natural language processing, and speech recognition [3].

SR field was not an exception and has been revolutionized by the significant advances in convolutional neural networks (CNNs), benefiting from their strong capacity of extracting effective high-level abstractions to map the LR space to HR space. Recent CNN-based SR methods have resulted in better reconstructions of high-resolution pictures, both quantitatively and qualitatively. Therefore, in this thesis, we focus more on CNN-based approaches.

In the remainder of this section, we present seminal CNN-based architectures, as well as

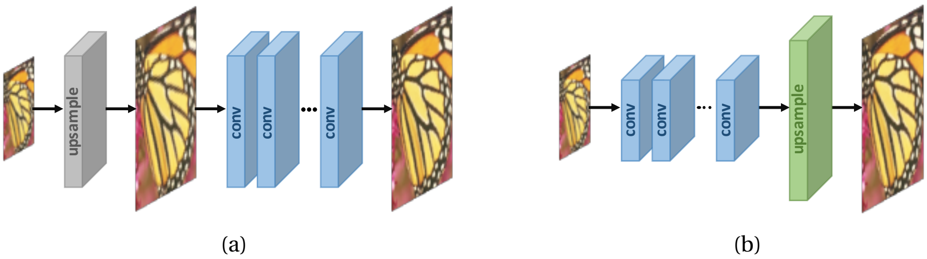


Figure 2.2 – Examples of linear SR designs with: (a) Pre-upsampling, (b) Post-upsampling methods. Figure taken from [17]

essential concepts related to conventional objective functions and SR evaluation metrics, that we need to know for better understanding the basis CNN-based SR works. **A closer related work to each of our contributions is presented in its corresponding chapter.**

2.2.1 Deep architectures for SR

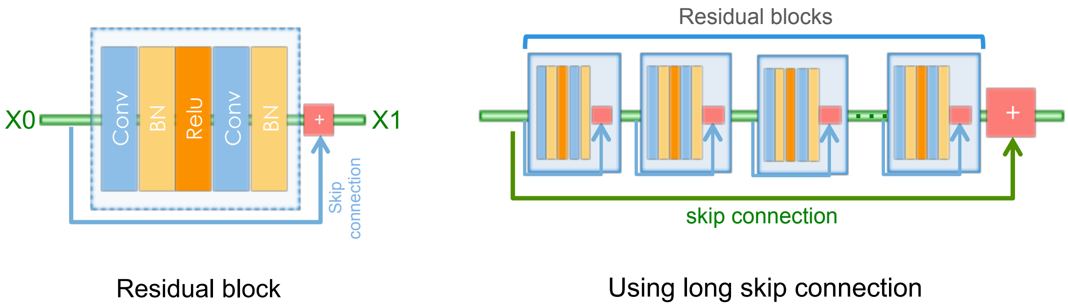
This field has witnessed a variety of end-to-end deep network architectures in recent years; in this section, we present some of the seminal architecture designs proposed for the SR task.

Linear designs

SRCNN [11] is the first architecture that used convolutional layers for the SR task and reached successful high reconstruction quality. This design benefits from a simple structure, consisting of only a single path for data flow without any skip connections, residual blocks, or multiple branches. Generally, in this design category, the input image is sequentially passed from several convolution layers stacked on top of each other to reach the final output layer. The input image can initially be upsampled very early before passing through the network [11, 12, 13, 14] (pre-upsampling) or either upsample the features around the output layers of the network (post-upsampling) in order to decrease the computational cost and dimensionality of the problem [15, 16]. Figure 2.2 shows an example of pre-upsampling and Post-upsampling methods in linear designs.

Residual networks

Using the concept of residual learning to reduce training difficulty and improve the learning ability was first introduced in ResNet [18]. Later, this design was also widely used for SR task and significantly boosted the reconstruction quality [19, 20, 21, 22]. This category uses skip connections [23] to alleviate vanishing gradients, which enables designing very deep networks [24]. In practice, this method's implementation is done by adding skip (shortcut) connections, mostly scaled by a learnable constant, and using either element-wise addition



(a) (b)

Figure 2.3 – The design of a residual network for SR task, introduced by SRGAN [19], (a) Using short skip connection to construct a residual block, (b) Using series of residual block and long skip connections to build the body of the SR feature extractor (upsampling block is not shown in this figure).

or concatenation. Figure 2.3 shows an example of residual design used in SRGAN [19] work. This work demonstrates the concept of residual blocks and skip-connections to facilitate the training of CNN-based SR decoders.

Recursive networks

Recursive networks use recursively connected units of either convolutional layers (sharing the same weights) or more complex units, such as residual blocks. The intuition behind the recursive designs is to break down the harder SR problem into a set of simpler ones. Some of the important examples of such designs are [25, 26, 27].

A seminal architecture using this method, shown in Figure 2.4, is DRCN [25]; this proposed method uses the same convolutional layers, with the same weights and biases, multiple times. As the layers are shared, the number of parameters remains the same for any number of recursions. In particular, DRCN is contained of three sub-networks: an embedding network, an inference network (based on recursive design), and a reconstruction network. The first and last sub-networks are designed to map the color image into feature maps and convert the final feature map back into RGB space, respectively. The inference network performs SR by analyzing the input feature map by recursively applying a single unit, consisting of a single layer convolution followed by a ReLU activation function. The spatial size of the feature map is increased after each recursion. This work also shows how deeper network architectures increase the performance of SR.

Progressive upsampling

Motivated by the difficulty of learning an SR model for large scaling factors (e.g., 8, 16) in one single step, various works [28, 29] propose to perform SR image progressively in multiple

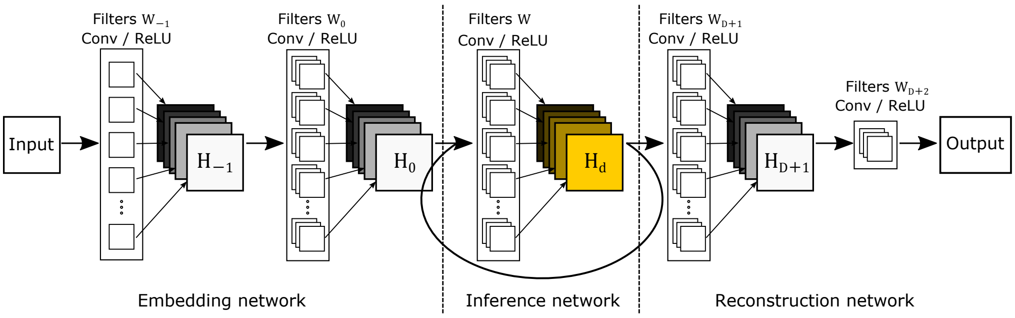


Figure 2.4 – Architecture of DRCN [25], an example of recursive networks. It consists of three parts: embedding network, inference network, and reconstruction network. In the inference network, the same filter W is applied to feature maps recursively. Figure taken from [25]

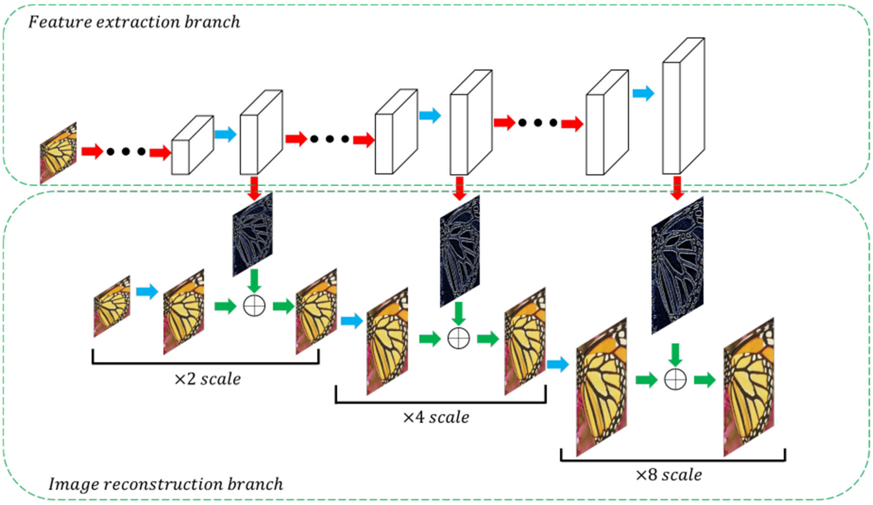


Figure 2.5 – Architecture of LapSRN [28], an example of progressive upsampling design. Red, blue and green arrows denote convolutional layers, upsampling layers, and element-wise addition operators, respectively. Figure taken from [30].

steps, for example, upsampling the image with a scale factor of two, followed by an additional upsampling with a scale factor of two (×16 as a result). Laplacian pyramid structure [28], shown in Figre 2.5, is an important example of progressively reconstructing the sub-band residuals of high-resolution images.

Densely connected networks

The idea of dense blocks was first proposed by [31] for the image classification task. Followed by its success, many works were introduced in different computer vision tasks, SR task included, based on the same idea of using densely connected CNN layers to improve the performance [32, 33, 34]. The main motivation in such SR decoders is to provide an effective way to combine the low and high-level features by propagating each layer's feature maps into

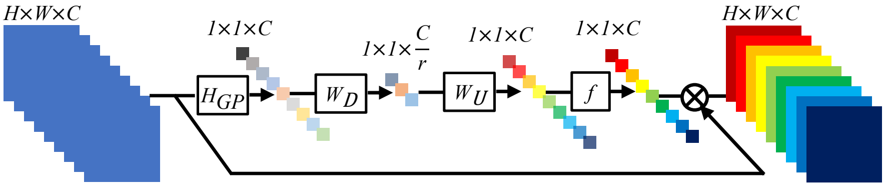


Figure 2.6 – RCAN Channel attention [35]. H_{GP} denotes the global pooling function, W_D and W_U denote the set of weights of a Conv layer used as channel downscaling and upscaling, respectively, with a scale of r, and the function $f(.)$ is the sigmoid function. Figure taken from [35].

all subsequent layers. [32] shows that this method results in richer feature representations and improves the reconstruction performance.

Attention-based architectures

Most of the seminal CNN-based networks proposed for image SR task consider the same importance for all spatial locations and channels and treat them in the same way. However, selectively weighting only a subset of features at a given layer or location could intuitively have benefits in several cases. Attention-based approaches are introduced to bring this flexibility into computer vision tasks and particularly have shown significant improvements in the SR task.

Residual Channel Attention Network (RCAN) [35] is an example of CNN-based SR approaches using attention mechanisms. They propose a channel attention mechanism to adaptively rescale channel-wise features by considering interdependencies among channels. In particular, in their design, each local residual block benefits from a channel attention mechanism such that the filter activations are collapsed from $h \times w \times c$ to a vector with $1 \times 1 \times c$ dimensions (after passing through a bottleneck) that acts as a selective attention over channel maps (see Figure 2.6).

In the next section, we review some of the essential cost functions used in the community for this task.

2.2.2 Objective functions

Despite various architectures proposed for the image SR task, the behavior of CNN-based methods is principally driven by their objective functions. The commonly used objective function for the SR task is presented in the following subsections.

Pixel-wise loss

The most used objective function for the SR task in the literature is the pixel-wise distance between the super-resolved and the ground-truth HR images. This loss is commonly used in two different forms:

- **L1 loss ($\mathscr{L}_{MAE}$)** calculates the Mean Absolute Error (MAE) between the ground-truth image and the reconstructed image. This loss can be formulated as:

  $$\mathscr{L}_{MAE}(I', I) = \frac{1}{hwc} \sum_{i,j,k} \left| I'(i,j,k) - I(i,j,k) \right|, \tag{2.6}$$

  where I denotes the original image, I' is the reconstructed image, and h, w, and c are representing the height, width, and number of channels of the image, respectively.

- **L2 loss ($\mathscr{L}_{MSE}$)** calculates the Mean Squared Error (MSE) between the ground-truth image (I) and the reconstructed image (I'). This loss can be formulated as:

  $$\mathscr{L}_{MSE}(I', I) = \frac{1}{hwc} \sum_{i,j,k} \left(I'_{i,j,k} - I_{i,j,k} \right)^2. \tag{2.7}$$

Both $L1$ and $L2$ losses are among the most common metrics used to measure the accuracy for continuous variables. As $L2$ loss calculates the square of the errors before averaging them, it results in relatively high weight for large errors; therefore, it is known to be more advantageous when specifically large errors are undesirable. However, in the SR task, $L1$ choice is getting more popular than $L2$, as it is observed to produce fewer artifacts [36]. Figire 2.7 compares two SR images reconstructed by using $L1$ and $L2$ losses. The most known drawback of using either of these pixel-wise loss forms as a cost function is reconstructing over-smoothed images due to the pixel-wise average of plausible solutions in the pixel space. To overcome this issue, perceptual losses -presented in the following section, have been introduced into the SR task.

Perceptual loss

Perceptual-driven approaches added a remarkable improvement to image SR in terms of visual quality. These loss functions are designed to optimize an SR model in a feature space instead of pixel space and tackle the problem of blurred textures caused by optimization of pixel-wise losses. The most known forms of perceptual losses are:

- **Content loss ($\mathscr{L}_{content}$)** or VGG loss ($\mathscr{L}_{VGG}$). Based on the idea of perceptual similarity [39], the content loss is proposed to minimize the error in a feature space using specific layers of a pre-trained feature extractor, for example, VGG-19 [40]. The idea behind this idea is to force the reconstructed image to be perceptually closer to the original (HR)

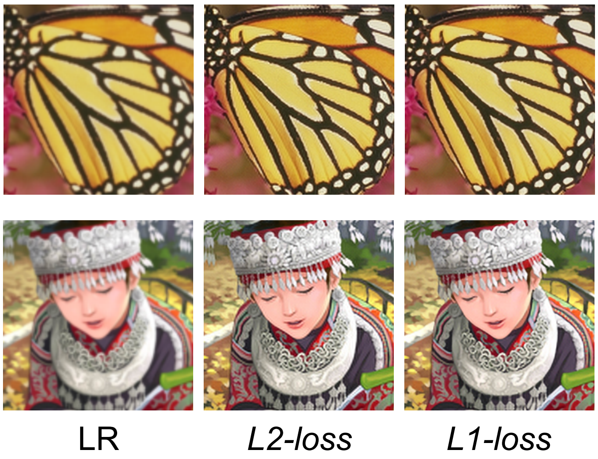


Figure 2.7 – Comparing $L1$ and $L2$ losses. Some artifacts, such as grating black edges around the butterfly's wing and the girl's face, are noticeable when $L2$ loss is used as the training cost function. Images are generated by [36]. Test images are from Set5 [37] and Set14 [38] datasets.

image instead of trying to match them in a pixel-wise manner (Figure. 2.8). The content loss can be formulated as:

$$L(I', I) = \frac{1}{lmn} \sqrt{\sum_{i,j,k} \left(\phi^{(d)}_{i,j,k}(I') - \phi^{(d)}_{i,j,k}(I) \right)^2}, \tag{2.8}$$

where $\phi^{(d)}$ is the feature extractor, commonly VGG-19, returning l-th layer feature map, and l, m and n are representing the dimensions of that layer. As previously mentioned, I and I' stand for the original image and the reconstructed image, respectively. We emphasize that different feature extractors, e.g., ResNet [18] could be used as ϕ, instead of VGG.

- **Texture matching loss** ($\mathscr{L}_{Texture}$) [41] develops a similar approach and further explores a patch-based texture loss. They use the idea of style transfer loss [42] for SR task and propose computing $\mathscr{L}_{Texture}$) patch-wise during training, to encourage the reconstruction of similar textures between I' and I. As the effectiveness of using texture loss while using content loss with the right setting and parameters seems to be insignificant in many recent works [2, 19, 43], we do not further investigate or use this loss term in this thesis.

In general, different perceptual approaches used different levels of features from the feature extractor to restore the original image. To understand the meaning of this choice, first, we need to know what each level of features represents; Figure 2.9 is an attempt to visualize VGG feature maps by maximizing filter activations at different levels. In this figure, we can see how shallow (low-level) features focus on local information such as edges, mid-level features

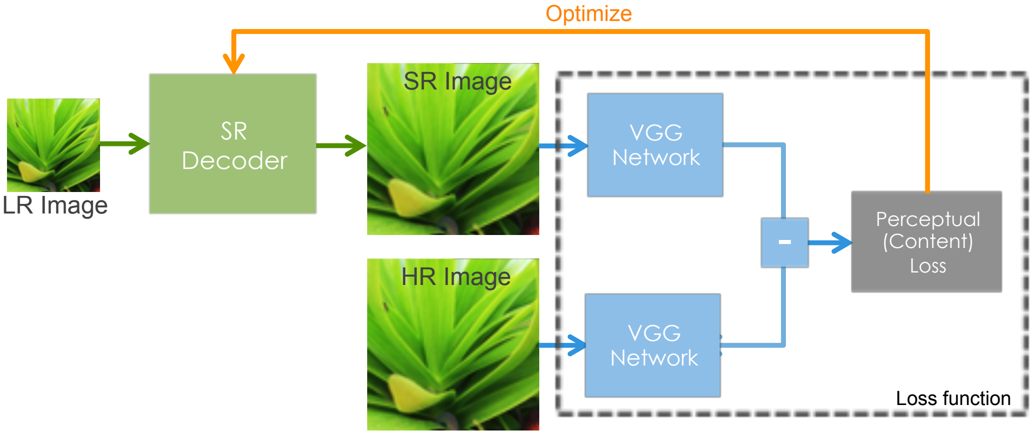


Figure 2.8 – Optimization with perceptual losses.

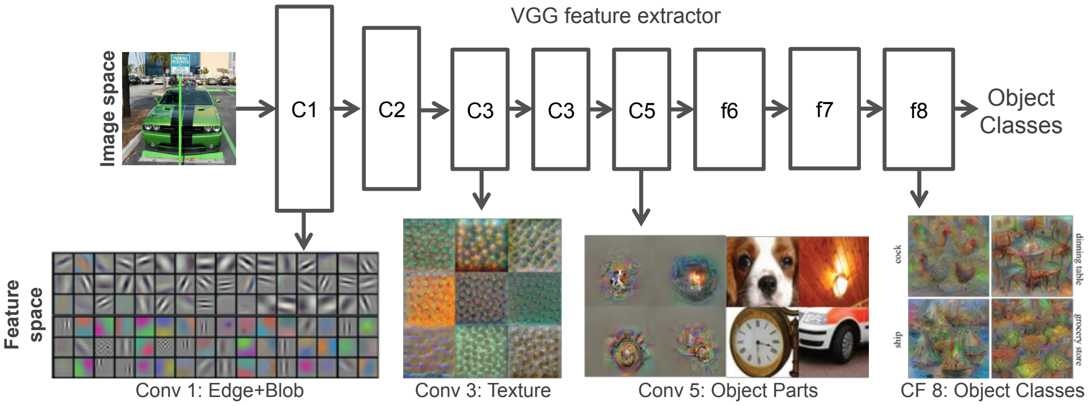


Figure 2.9 – VGG [40] architecture; visualzing low to high-level features.

represent textures, and finally, higher level features correspond to more semantic information of the image. We discuss the advantage of this knowledge and how we could benefit from it in more detail in Chapter 4.

In the remainder of this thesis, **referring to perceptual loss** ($\mathscr{L}_{perc.}$) **denotes specifically the content loss (VGG-19 loss)**, as this is the most common type of this category of cost functions.

Adversarial loss

Considering the importance of adversarial losses and their new advances in SR task, we present the idea behind the Generative Adversarial Networks (GANs), as well as the formulation of adversarial loss in Section 2.2.3.

Cycle Consistency Loss

Inspired by the CycleGAN [44], Cycle consistency loss was introduced in SR task [45] with the main goal of constraining their pixel-level consistency. This has be done by not only mapping the low-resolution image to the HR image, but also downsample the constructed image back to another LR image, identical to the input, through another CNN. The cycle consistency loss ($\mathscr{L}_{cycle}$) can be summarized as:

$$\mathscr{L}_{cycle}(I'^{LR}, I^{LR}) = \frac{1}{hwc}\sqrt{\sum_{i,j,k}\left(I'^{LR}_{i,j,k} - I^{LR}_{i,j,k}\right)^2}, \tag{2.9}$$

where I'^{LR} and I^{LR} denote the final reconstructed image mapped back again to low-resolution space and the input low resolution image, respectively.

2.2.3 Generative adversarial networks in SR

A generative adversarial network, in short GAN, proposed by [46], is a frameworks for estimating generative models via deep neural networks and an adversarial process, in which simultaneously two models are trained to contest with each other: (1) a generative model G, with the goal of fooling (2) a discriminator D, that is trained to be able to distinguish data from training samples (real) and generated data by model G (fake). The ultimate goal of this framework is to have a final G which is able to generate new, synthetic instances of data that can be passed for real samples, or at least superficially, be looking authentic to human observers.

This class of machine learning frameworks achieved considerable success in recent years and was used widely in many machine learning and computer vision applications, such as image and video generation [47, 48], image/text to image translation [49, 50], photo inpainting [51], voice generation [52], etc.

Recently, in terms of SR, two breakthroughs have been made and resulted in near-photorealistic reconstructions in terms of perceived image quality: 1- perceptual losses (see Section 2.2.2), 2- Introducing the discriminator component of GANs to SR applications, which encourages an SR decoder to favor solutions that resolve more realistic and natural images. Figure 2.10 illustrates the training procedure of a SR decoder, alongside with a discriminator. In the following sub-sections, we discuss both the SR generator and discriminator, as well as various objective functions to train them simultaneously, in more detail.

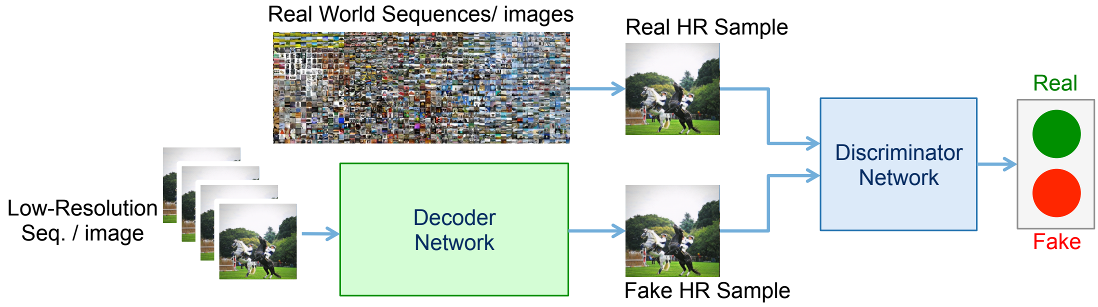


Figure 2.10 – The structure of GANs, illustrating the training procedure of an SR decoder alongside a discriminator.

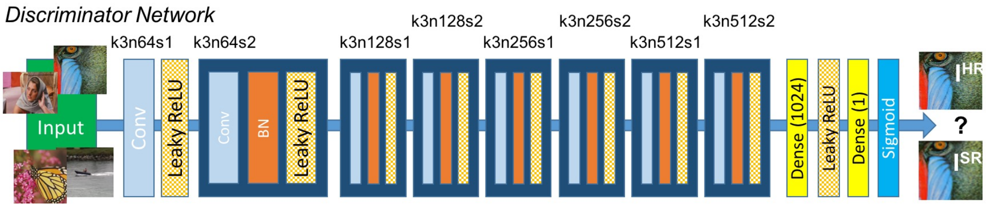


Figure 2.11 – The network architecture of the discriminator proposed by [19]. k denotes the kernel size, n denotes the number of feature maps, and s is the stride for each convolutional layer. Figure taken from [19].

Generator design

The generator tries to generate realistic images that fool the discriminator by benefiting from an additional loss term, generated by the discriminator network. Therefore, only the objective function of the generator is changed during the training process and its design does not require any specific form, to be compatible with GAN settings. As a result, all generator designs presented in Section 2.2.1 could be chosen based on their capabilities and be trained simultaneously with a discriminator.

Discriminator

The discriminator in a GAN is simply a classifier; therefore, all DNN-based classifiers designs could be used as discriminator network. A simple but yet effective network design for SR discriminator is inspired by VGG [40] with few modifications such as using leaky ReLU activations and strided convolutions instead of pooling layers to decrease the spatial dimensions of the image gradually. An example of discriminator network, proposed by [19] is shown in Figure 2.11.

Objective functions

Different formulations have been proposed to estimate both the generator loss ($\mathscr{L}_G$) and discriminator loss ($\mathscr{L}_D$); SRGAN [19] proposed to use adversarial loss based on cross-entropy:

$$\mathscr{L}_G = -\log\left(D(G(I^{LR}))\right), \tag{2.10}$$

where D is the discriminator, G is the SR generator, and I^{LR} is the input image. Consequently, the discriminator loss is formulated as:

$$\mathscr{L}_D = -\log\left(D(I^{HR})\right) - \log\left(1 - D(G(I^{LR}))\right), \tag{2.11}$$

where I^{HR} can be a random HR image from ground-truth. To increase the stability and reaching higher quality results, [45, 53] propose adversarial loss based on least square error, instead of cross-entropy. Their formulation is given as:

$$\mathscr{L}_G^{(ls)} = \left(D(G(I^{LR})) - 1\right)^2, \tag{2.12}$$

$$\mathscr{L}_D^{(ls)} = \left(D(I^{HR}) - 1\right)^2 + \left(D(G(I^{LR})\right)^2, \tag{2.13}$$

To further enhance the GANs stability of learning, [54] introduced the Wasserstein GAN (WGAN). In particular, WGAN proposes a loss function using Wasserstein distance (or earth mover's distance) with a smoother gradient, which results in learning regardless of whether the generator is producing good images. Due to the problems caused by weight clipping in WGAN design [55], the generator may still do not converge and produce low-quality images. To overcome this issue, [55] proposes WGAN with a gradient penalty -namely WGAN-GP, to enforce the Lipschitz constraint and claim to have a better performance than WGAN.

In terms of SR, [56] demonstrates the effectiveness of WGAN-GP to ensure more stable and converging training, compared to original GAN, with minimal hyperparameter tuning.

In contrast to the mentioned contributions concerning specific terms for adversarial losses in pixel space, [57] suggests to benefit from an additional discriminator in the feature domain. This method is motivated by the argument that pixel-level discriminators have the tendency to generate high-frequency noises, which are irrelevant to the input LR image. They show that the proposed discriminator enforces the generator to captures more meaningful attributes of original HR images. Another variation of GANs for SR is proposed by [43]; this work, namely ESRGAN, use the idea from relativistic GAN, and its discriminator predicts relative realness

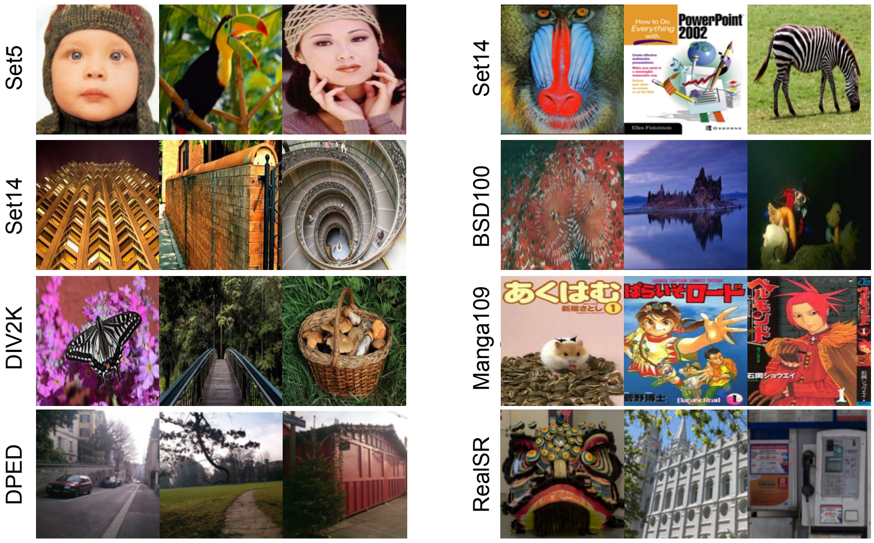


Figure 2.12 – Some examples of test images from Se5 [37], Set14 [38], BSD100 [58], Urban100 [59], DIV2K [60], Manga109 [61], DPED [62], and RealSR [63] datasets, used for training and evaluations.

instead of the absolute probability that input images are real or fake. The authors show that the proposed method results in recovering more detailed textures.

2.3 Datasets

This section presents some of the most famous and publicly available benchmark datasets introduced by the SR community. Some of the crucial ones, such as Set5 , Set14, DIV2K, and RealSR are repeatedly used in this thesis for both qualitative and quantitative evaluations, as well as user studies. Figure 2.12 contains some representative images from these datasets.

- **Set 5** [37] is probably the most known dataset in SR; most of its images can be seen in many SR articles used used for comparing and evaluating their proposed algorithms. This dataset contains only five famous images: a baby, a bird, a butterfly, a head, and a woman. These images are known by the same names in computer vision articles.
- **Set 14** [38] is considered as a complementary test set to Set5, which consists of 14 more images. Its relatively low number of images made this dataset very popular for SR qualitative and quantitative evaluations.
- **BSD100** [58] or the Berkeley Segmentation Dataset consists of 100 test images, including different categories, such as buildings, animals, landscapes, food, humans, plants, etc.
- **Urban100** [59] is a unique dataset because of its specific focus on the photographs of human-made structures, e.g., buildings, towers, windows, etc. Images from urban100

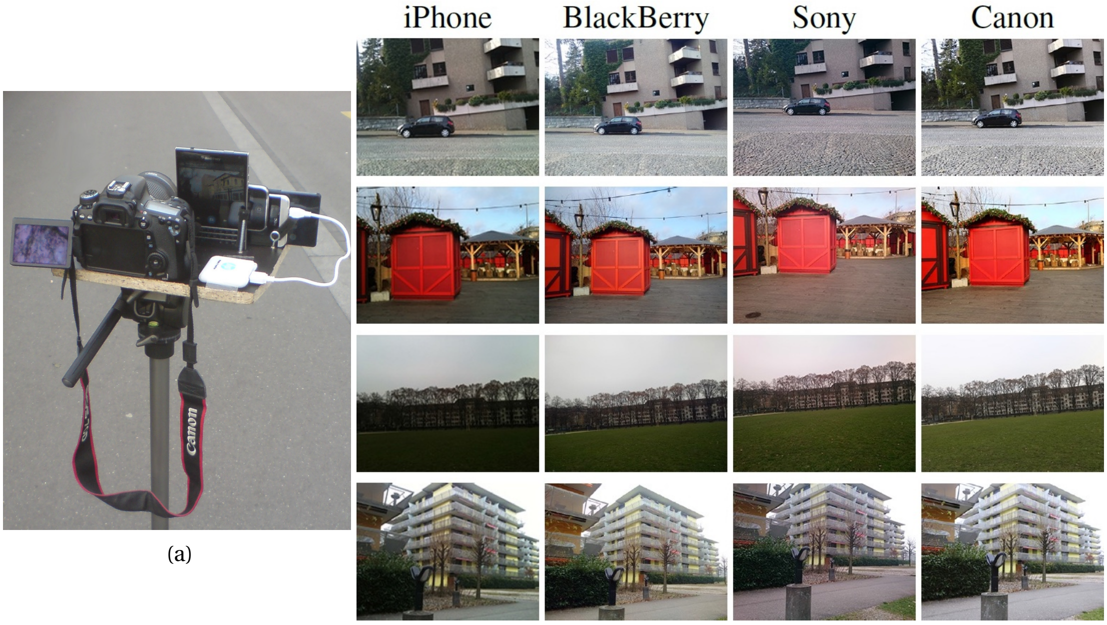
are very convenient for comparing edges and patterns in reconstructed images due to their specific nature.

- **DIV2K** [60] is a more recent SR dataset containing relatively higher quality images than previously mentioned datasets; all images have a 2K resolution. Their images are divided into three sub-groups of training (800 images), testing (100 images), and validation (100 images) sets. Their test set's ground-truth is not publicly available, but their training and validation sets are widely used by recent SR works.

- **Manga109** [61] is a dataset consisting of 109 art images, mostly created for comics or graphic novels, by professional Japanese. These mangas were commercially made available to the public only from the 1970s. The permission for the use of this dataset is only granted for academic purposes and non-profit organizations.

- **DPED** [62] is a new large-scale dataset presented in 2017 and consists of real photos captured from three types of smartphones and one high-end reflex camera. The goal of introducing such a dataset was to fill the gap between the quality of images taken by smartphone cameras and superior quality images taken by professional digital single-lens reflex (DSLR) cameras. To capture images simultaneously from different devices, but from the same scenes, four different devices were mounted on a single tripod and were activated remotely by a wireless controller (Figure 2.13.a). The smartphones used in this setup are iPhone 3GS, BlackBerry Passport, Sony Xperia Z, and the professional camera is Canon 70D DSLR. An example of images taken from the same scene but with different cameras is shown in Figure 2.13.b. These images were then re-aligned by using SIFT [64] descriptor matching, followed by a non-linear transform and further cropping, to address the problem of misalignment due to images taken from a slightly shifted position. In total, over 22K photos are available in this dataset.

- **RealSR** [63] is one of the few datasets with real pairs of low and high-resolution images (images of the same scenes with different resolutions). In all previously presented datasets, the low-resolution images are generated by applying a uniform and simple degradation such as bicubic downsampling to their original images. In the RealSR dataset, LR and HR images are generated by taking two camera pictures of the same scene and changing the camera's focal length between the two pictures. Hence, both are real images, but with the RealSR LR being degraded with the degradation from changing the camera's focal length (zooming out). In total, 559 images (459 images for training and 100 images for testing) exists in the latest version of RealSR. Figure 2.14 illustrates the registration process to create low and high-resolution pairs by only changing the focal length of the camera. We present the importance of this dataset, as well as the problem of real-world SR in Section 2.5 and Chapter 6, in more detail.



(a)

(b)

Figure 2.13 – DPED dataset [62], (a) Setup used to capture images synchronously by different hardware, (b) Examples of images from the same scenes, taken by four different cameras.

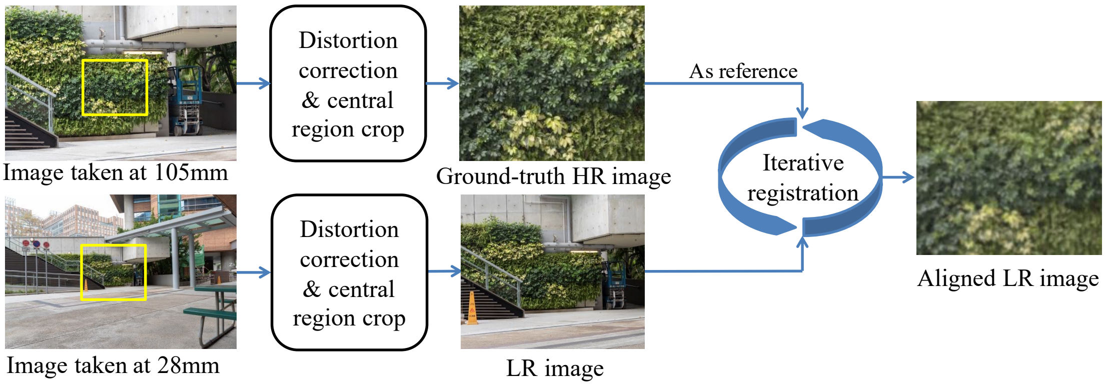


Figure 2.14 – The registration process of RealSR dataset; creating low-resolution and high-resolution image pairs by changing the focal length of the camera. Figure taken from [63].

2.4 Evaluation metrics

2.4.1 Full-reference distortion measures

The most popular examples for mathematical distortion measures are the PSNR and the SSIM:

PSNR

Peak Signal to Noise Ratio (PSNR) is most commonly used to measure the quality of reconstruction in many computer vision tasks such as compression, image inpainting, etc. PSNR is mostly defined by using the mean squared error (MSE):

$$PSNR = 10\,.\log_{10}\left(\frac{max(I)^2}{\mathscr{L}_{MSE}(I', I)}\right), \tag{2.14}$$

where $max(.)$ is the function to find the maximum possible pixel value of the image, and $\mathscr{L}_{MSE}$ denotes the means squared error between reconstructed image I' and ground truth image I (2.7).

SSIM

The Structural Similarity Index (SSIM) is another commonly used full reference metric to measure the similarity between two images and estimate the perceived quality of SR. The Full-reference term refers to the requirement of the original images as the reference to judge the quality of the reconstructed image. The SSIM is calculated on various spatial windows of the image and is based on three independent comparison measurements between the ground-truth image and the reconstructed image: 1- structure, 2- luminance, and 3- contrast. The final SSIM is then calculated as a weighted combination of these measures.

2.4.2 Learning-based metrics

As it is emphasized in [2, 19, 41], distortion metrics such as SSIM and PSNR, previously discussed in this thesis, are not directly correlated to the human perception of image quality; they show that GAN-based super-resolved images could have higher errors in terms of these metrics while still generating more appealing and realistic images. Therefore, in this section we present further attempts to propose more reliable metrics for SR and image quality assessment, based on learning. These approaches range from full-reference measurements such as the LPIPS metric to non-reference methods such as NIQE and PI.

LPIPS

The Learned Perceptual Image Patch Similarity (LPIPS) metric [65] is recently introduced as a reference-based image quality assessment metric, which seeks to estimate the perceptual similarity between two images. This metric uses linearly calibrated off-the-shelf deep classification networks trained on the very large Berkeley-Adobe Perceptual Patch Similarity (BAPPS) dataset [65], including human perceptual judgments. However, as [66] also emphasizes, LPIPS has a similar trend as distortion-based metrics, e.g., SSIM, and would not necessarily imply photo-realistic images.

NIQE

Naturalness Image Quality Evaluator (NIQE) is a no-reference image quality score proposed by [67]. This metric is an entirely blind image quality assessment. It only benefits from statistical regularities learned from natural images and measurable deviations from these regularities to estimate an index of quality. In particular, at the test time, this approach fits a set of local features, extracted from predicted images, into a multivariate Gaussian (MVG) model. The quality score is then calculated by finding the distance between MVG's parameters and the reference MVG's parameters learned from a dataset of natural images.

PI

The Perception Index (PI) is proposed by the 2018 PIRM challenge [68] to introduce a more correlated metric with human perception of reconstruction quality. Their proposed metric is a linear combination of MA [69] and NIQE measurements. MA is a non-reference metric, regressing three types of low-level statistical features extracted from predicted images to estimate the perceptual quality. PI can be formulated as follows:

$$PI = \frac{1}{2}\left((10 - MA) + NIQE\right) \tag{2.15}$$

The proposed metric is claimed to be more reliable in terms of representing the perceptual quality of the images by estimating the Spearman's correlation coefficients between PI and the ratings from 35 human observers.

2.4.3 User study

Despite various metrics proposed for SR quality assessment, we emphasize that finding a reliable measurement is still an open challenge. Currently, the only way to reflect superior reconstruction quality is through the mean opinion score (MOS) or user studies.

Different user study designs were proposed in recent works. In general, participants are either

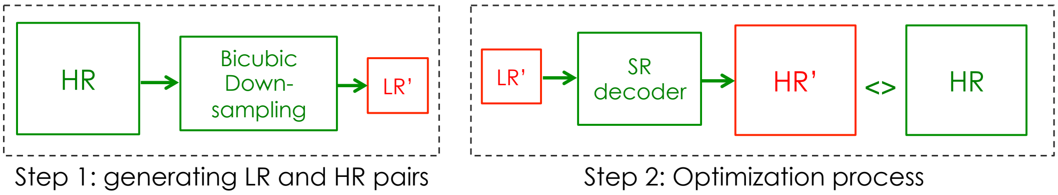

HR
Bicubic Down-sampling
LR'
Step 1: generating LR and HR pairs
LR'
SR decoder
HR'
<>
HR
Step 2: Optimization process

Figure 2.15 – The pipeline of conventional supervised approaches for image super-resolution; during the first step, a dataset of low and high-resolution pairs is created using bicubic downsampling kernel. In the second step, a CNN-based decoder is trained in a supervised fashion using the image dataset generated during the first step.

asked to choose their preferred choice among multiple images generated by different SR methods or assign an integral score based on their perception of quality, e.g., rating from one (for the worst quality) to five (for an excellent quality). This study can be done in controlled ways (mostly in person, with a piece of specific equipment and fixed conditions), or in a crowdsourcing way with less controlled conditions (but more participants).

In Chapter 4 and 6, include our own proposed procedure in detail to perform SR user studies.

2.5 Towards real-world super-resolution

In this section, we present the challenge of the real-world SR and some of the important solutions proposed for this solution.

2.5.1 The challenge

Most of the existing state-of-the-art image SR methods are based on the assumption of having a pre-defined and known downsampling operator such as the bicubic kernel. In other words, they build a simulated dataset by using this uniform degradation (bicubic downsampling) and train their method by these synthetic low and high-resolution image pairs (Figure 2.15).

To validate such approaches quantitatively, different synthetic test sets have been created by the same approach and uniform degradation. Although current methods proved to have outstanding results on such test sets, a significant drop in reconstruction quality has been observed in real applications, where the downsampling kernels were unknown. As it can be observed in Figure 2.16, this mismatch between the predefined kernel and real varying kernels (depending on camera hardware, motion, noises, etc.) causes undesired artifacts at the output level.

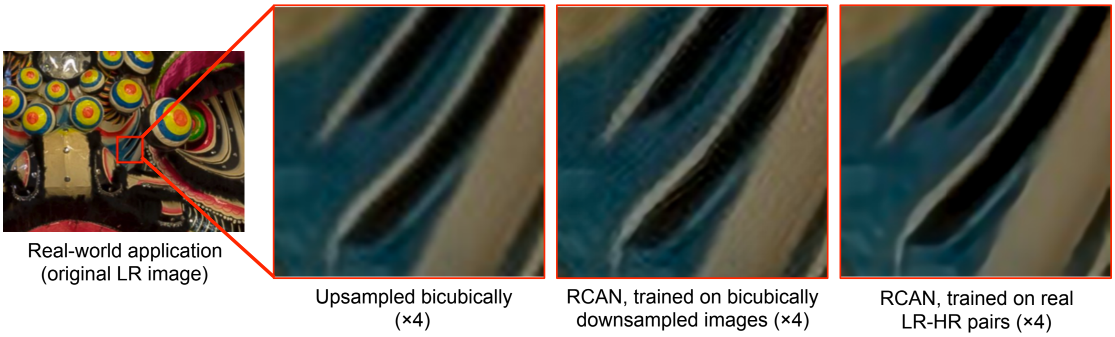


Figure 2.16 – The real-word super-resolution challenge; comparing images generated by bicubic kernel, original RCAN [35] method trained on artificially downsampled images, and an extended version of RCAN trained on more realistic images of RealSR dataset [63].

2.5.2 Multiple-degradation handling approaches

To overcome the real-world degradation variations, different approaches have been proposed in recent years; in the following, we outline some of the essential solutions:

Unsupervised / Weakly-supervised SR

To cope with the real-world SR problem and real LR-HR pairs, researchers focused more on unsupervised and weakly-supervised approaches, in which unpaired low and high-resolution pairs are sufficient to learn SR. An example of unsupervised approaches for SR is zero-shot SR (ZSSR) [70]. The authors propose to estimate the degradation kernel from a single image by using the proposed method by [71] at the test time and then use this kernel as well as some augmentation techniques to generate a small training set. Finally, a CNN-based network is trained specifically for this test image and predicts the final image.

[72] proposes a weakly supervised approach in two steps: 1- Learning an HR to LR mapping by GANs and in an unpaired manner, then, 2- Training another generator to learn the mapping from LR to HR, based on paired images generated by the first step. Authors validated this two-step method for the face SR task and shown significant increases in the quality of reconstructed images for real-world face images comparing to previous state-of-the-art works.

CinCGAN, proposed by [45], is another example of weakly supervised SR approaches. In this attempt, the authors propose a cycle-in-cycle SR network to learn a round-trip mapping consisting of mapping a real LR image into a noise-free LR image, then from noise-free LR to HR, and finally mapped back again into the real (noisy) LR space. Their training objectives include adversarial losses for each domain to match the target domain distribution and mapping validity.

Real-world SR through ‘real’ data

As mentioned previously, there have been several attempts to create realistic low and high-resolution images pairs, such as a two-step approach of [72], and CinCGAN [45]. Although these approaches could improve the reconstruction quality of real SR, the datasets created by them are still synthetic and not the same as real low-resolution images. In the remainder of this section, we present some work that uses natural images for both high and low-resolution images to overcome the real-world SR issue.

[45] proposes City100 dataset by studying the relationships between image resolution and field-of-view of camera hardware and proposing novel data acquisition strategies (using DSLR and smartphone cameras) to conduct a real-world dataset. This dataset set consists of only 100 image pairs. [73] uses optical zoom of cameras and solves the misalignment problem of captures images by a contextual bilateral loss, to create another real high and low resolutions pairs, namely SR-RAW dataset. Authors validate their approach in the specific application of ×4 and ×8 computational zooming. Finally, RealSR [73] trains its network on a dataset consisting of 559 real LR and HR images, built by taking two pictures of the same scene and changing the camera’s focal length hardware between the two pictures. More detail of this dataset is presented in Section 2.3.

To conclude this part, we should emphasize that works benefiting from ‘real’ data have experimentally demonstrated the superiority, in terms of the reconstruction quality, compared to using synthetic images for real-world SR problem.

3 Extracting Image Context by Multi-Task Learning

Knowing the context of an image helps humans easily recognize different types of objects and textures in a scene. As emphasized in Chapter 1, this knowledge has already been proven to be beneficial for image generation and super-resolution tasks, and shown to result in better reconstruction quality, if used appropriately [2, 49, 74]. However, it has very limited practicality as it requires such contextual information as an additional input at the test time. This information is usually extracted by an additional segmentation/classification network and injected into the final network as semantic maps to guide the reconstruction process. In this chapter, we study the benefits of contextual information in more detail and aim to expand this idea by addressing its main limitations; we focus on proving that a single SR decoder can learn this categorical and contextual knowledge by using the concepts of multitask learning. In particular, we demonstrate that a shared representation learned for two specific tasks -semantic segmentation and super-resolution in this study, can significantly improve each task's quality, particularly the SR task in this work.

In the following, this chapter is presented in the form of an article. The research and all experiments were performed by the first author (the author of this thesis). The manuscript was also written by him and was further revised by other authors. The article was published in **Neurocomputing Journal** (*Volume 398, 2020, Pages 304-313 DOI: 10.1016/j.neucom.2019.07.107*), with the original title of **Benefiting from Multitask Learning to Improve Single Image Super-Resolution**.

Benefiting from Multitask Learning to Improve Single Image Super-Resolution

Authors: Mohammad Saeed Rad[1], Behzad Bozorgtabar[1], Claudiu Musat[2], Urs-Viktor Marti[2], Max Basler[2], Hazım Kemal Ekenel[1, 3], Jean-Philippe Thiran[1].

[1] Signal Processing Laboratory 5, EPFL, Lausanne, Switzerland.
[2] AI Lab, Swisscom AG, Lausanne, Switzerland.
[3] Istanbul Technical University, Istanbul, Turkey.

Abstract

Despite significant progress toward super resolving more realistic images by deeper convolutional neural networks (CNNs), reconstructing fine and natural textures still remains a challenging problem. Recent works on single image super resolution (SISR) are mostly based on optimizing pixel and content wise similarity between recovered and high-resolution (HR) images and do not benefit from recognizability of semantic classes. In this chapter, we introduce a novel approach using categorical information to tackle the SISR problem; we present a decoder architecture able to extract and use semantic information to super-resolve a given image by using multitask learning, simultaneously for image super-resolution and semantic segmentation. To explore categorical information during training, the proposed decoder only employs one shared deep network for two task-specific output layers. At run-time only layers resulting HR image are used and no segmentation label is required. Extensive perceptual experiments and a user study on images randomly selected from COCO-Stuff dataset demonstrate the effectiveness of our proposed method and it outperforms the state-of-the-art methods.

Keyword Single Image Super-Resolution, Multitask Learning, Recovering Realistic Textures, Semantic Segmentation, Generative Adversarial Network

3.1 Introduction

Single image super-resolution (SISR) has many practical computer vision applications [75, 76, 77, 78], which aims at recovering high-resolution (HR) images from a set of prior examples of paired low-resolution (LR) images. Although many SISR methods have been proposed in the past decade, recovering high-frequency details and realistic textures in a plausible manner are still challenging. Having said that, this problem is ill-posed, meaning each LR image might correspond to many HR images and the space of plausible HR images scales up quadratically

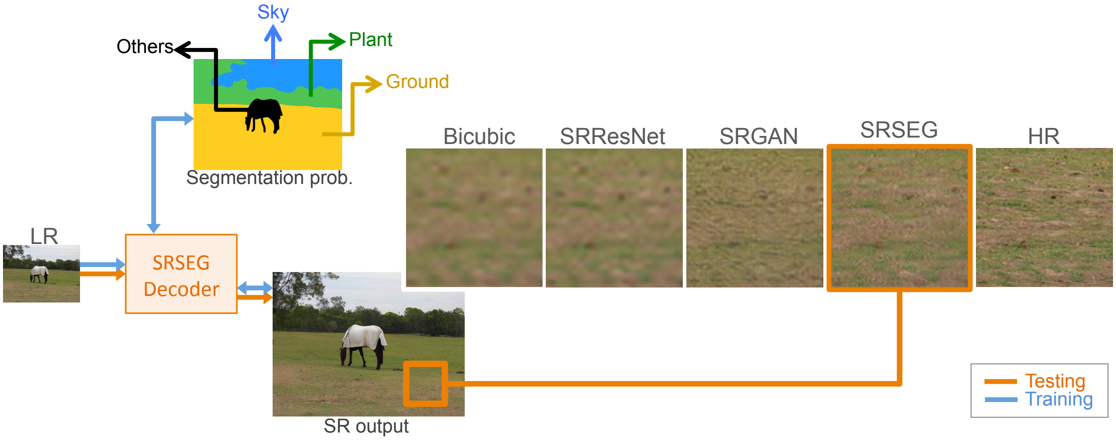


Figure 3.1 – The proposed single image super-resolution using multitask learning. This network architecture enables reconstructing SR images in a content-aware manner; during training (blue arrows), an additional objective function for semantic segmentation is used to force the SR to learn categorical information. At run-time we only reconstruct the SR image (orange arrows). In this work, we prove that learning semantic segmentation task in parallel with SR task can improve the reconstruction quality of SR decoder. Results from left to right: bicubic interpolation, SRResNet, SRGAN [19], and SRSEG (this work). Best viewed in color.

with the image magnification factor.

To tackle such an ill-posed problem numerous deep learning methods have been proposed to learn mappings between LR and HR image pairs [12, 16, 25, 79]. These approaches use various objective functions in a supervised manner to reach the current state-of-the-art. Conventional pixel-wise Mean Squared Error (MSE) is the commonly used loss to minimize pixel-wise similarity of the recovered HR image and the ground truth in an image space. However, [19, 41] show that lower MSE does not necessarily reflect a perceptually better SR result. Therefore, [80] proposed perceptual loss to optimize a SR model in a feature space instead of pixel space. Significant progress has been recently achieved in SISR by applying Generative Adversarial Networks (GANs) [2, 19, 81]; GANs are known for the ability to generate more appealing and realistic images and have been used in different image synthesis-based applications.

3.1.1 Does semantic information help?

Despite significant progress toward learning deep models to super resolve realistic images, the proposed approaches still cannot fully reconstruct realistic textures; intuitively, it is expected to have a better reconstruction quality for common and known types of textures, e.g., ground soil and sea waves, but experiments show that the reconstruction quality is almost the same for a known and an unknown type of texture, e.g., a fabric with a random pattern. Although loss functions used in image SR, e.g., perceptual and adversarial losses, generate appealing super-resolved images, they try to match the global level statistics of images without retaining the semantic details of the content. [2] shows that variety of different HR image patches could have very similar LR counterparts, and as a consequence, similar SR images are reconstructed for categorically different textures using current state-of-the-art methods. They also prove that more realistic textures could be recovered by using an additional network to obtain prior knowledge and afterward use it as a secondary input in SR decoder.

In this work, we prove that a single SR decoder is capable of learning this categorical knowledge by using multitask learning. As [82] emphasizes, multitask learning improves generalization by using the domain information contained in the training signals of related tasks. This improvement is the result of learning tasks in parallel while using a shared representation; in our case, what is learned for semantic segmentation task can help improving the quality of SR task and vice versa.

3.1.2 Our contribution

In this chapter, we propose a novel architecture to reconstruct SR images in a content-aware manner, without requiring an additional network to predict the categorical knowledge. We show that this can be done by benefiting from multitask learning simultaneously for SR and semantic segmentation tasks. An overview of our proposed method is shown in Figure 3.1. We add an additional segmentation output in a way that the same SR decoder learns to segment the input image and generate a recovered image. We also introduce a novel boundary mask to filter out unrelated segmentation losses related to imprecise segmentation labels. The semantic segmentation task forces the network to learn the categorical knowledge. These categorical priors learned by the network are characterizing the semantic classes of different regions in an image and are the key to recover more realistic textures. Our approach outperforms quality of recovering textures of state-of-the-art algorithms in both qualitative and user studies manner.

Our contributions can be summarized as follows:

- We propose a framework that uses segmentation labels during training to learn a CNN-based SR model in a content-aware manner.
- We introduce a novel boundary mask to have an additional spatial control over categorical information within training examples and their segmentation label, and filter out their irrelevant information for SR task.
- Unlike existing approaches for content-aware SR, the proposed method does not require any semantic information at the test time. Therefore, neither segmentation label nor additional computation is required at test time while benefiting from categorical information.

- Our method is trained end-to-end and is easily reproducible.
- Our experimental results, including an extensive user study, prove the effectiveness of using multitask learning for SISR and semantic segmentation and show that SISR of high perceptual quality can be achieved by using our proposed objective function.

In the remainder of this chapter, first, in Section 3.2, we review the related literature. Then, in Section 3.3, we give a detailed explanation about our design including the used dataset and our training parameters. In Section 3.4 we present experimental results and computational time, and discuss the effectiveness of our proposed approach. Finally, we conclude this chapter in Section 3.5 and also mention the future research directions.

3.2 Related work

3.2.1 Single image super-resolution

SISR has been widely studied for decades and many different approaches have been proposed; from simple methods such as bicubic interpolation and Lanczos resampling [83], to dictionary learning [84] and self-similarity [59, 85] approaches. With the advances of deep CNNs, the state-of-the-art SISR methods have been built based on end-to-end deep neural networks and achieved significantly superior performances, thus we only review relevant recent CNN-based approaches.

An end-to-end CNN-based approach was proposed by [11] to learn the mapping of LR to HR images. The concept of residual blocks and skip-connections [23, 25] were used by [19] to facilitate the training of CNN-based decoders. A laplacian pyramid network was presented in [28] to progressively reconstruct the sub-band residuals of high-resolution images. The choice of the objective function plays a crucial role in the performance of optimization-based methods. These works used various loss functions; the commonly used loss term is the pixel-wise distance between the super-resolved and the ground-truth HR images for training the networks [11, 12, 35, 41]. However, using those functions as the only optimization target leads to blurry super-resolved images due to the pixel-wise average of possible solutions in the pixel space.

A remarkable improvement in terms of the visual quality in SISR is the so-called perceptual loss [80]. This loss function benefits from the idea of perceptual similarity [39] and seeks to minimize the distance loss over feature maps extracted from a pre-trained network, e.g., VGG [40]. In a similar work, [86] proposes contextual loss to generate images with natural image statistics, which focuses on the feature distribution rather than merely comparing the appearance.

More recently, the concept generative adversarial network (GAN) [46] is used for image SR task, which achieves state-of-the-art results on various benchmarks in terms of reconstructing more appealing and realistic images [2, 19, 41]. The intuition behind its excellent performance is that GAN drives the image reconstruction towards the natural image manifold producing perceptually more convincing solutions. Having said that, it also uses a discriminator to distinguish between the generated and the original HR images, which is found to produce more photo-realistic results.

3.2.2 Super-resolution faithful to semantic classes

Semantic information has been used in different studies for variant tasks; [87] proposed a method to benefit from semantic segmentation for video deblurring. For image generation, [74] used semantic label to produce an image with photographic appearance. [49] used the same idea to perform image to image translation. The SISR method proposed by [2] is more relevant to our work. They use an additional segmentation network to estimate probability maps as prior knowledge and use them in existing SR networks. Their segmentation network is pre-trained on the COCO dataset [88] and then fine-tuned on the ADE dataset [89]. They show that it is possible to recover textures faithful to categorical priors estimated through the pre-trained segmentation network, which generates intermediate conditions from the prior and broadcasts the conditions to the SR network.

However, in this chapter, we do not have an additional segmentation network, instead our SR method is built on multitask end-to-end deep networks with the shared feature extraction parameters to learn semantic information. The intuition behind this proposed method is that the model can exploit features for both tasks, such a model, during training, is forced to explore categorical information while super-resolving the image. Therefore, the segmentation labels would be used only during the training phase and no additional segmentation labels would be required as the input at run-time.

3.3 Multitask learning for image super-resolution

Our ultimate goal is to train a SISR in a multitask manner, simultaneously for image SR and semantic segmentation. Our proposed SR decoder only employs one shared deep network and keeps two task-specific output layers during training to force the network learn semantic information. If the network converges for both tasks, we can be sure that the parameters of the shared feature extractor have explored categorical information while super-resolving the image. In this section we present our proposed architecture and the objective function used for training. We also introduce a novel boundary mask used to simplify the segmentation task.

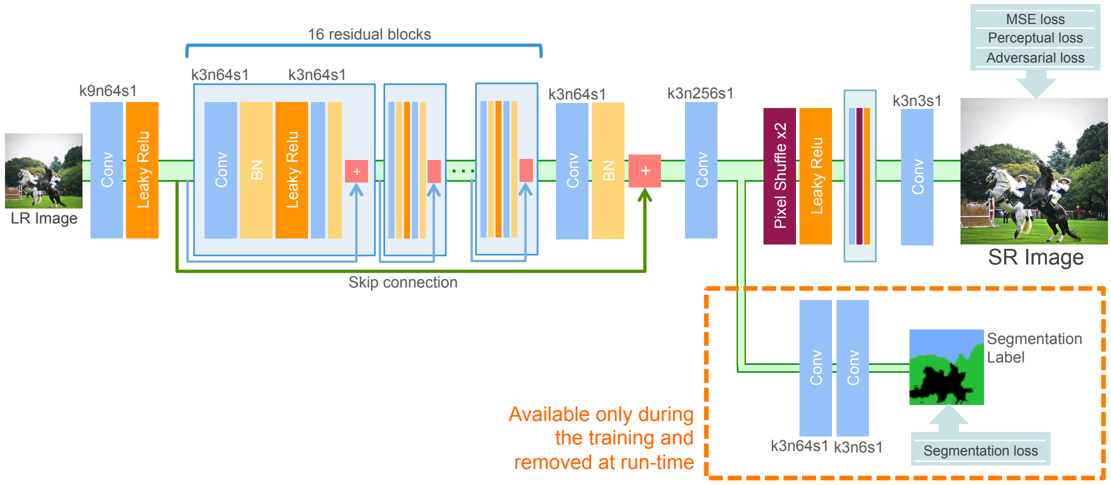


Figure 3.2 – Architecture of the decoder. We train the SR decoder (upper part) in a multitask manner by introducing a segmentation extension (lower part). Feature extractor is shared between both super-resolution and segmentation tasks. The segmentation extension is only available during the training process and no segmentation label is used at the run-time. In this schema, k, n and s correspond respectively to kernel size, number of feature maps, and strides.

3.3.1 Architecture

Figure 3.2 shows the multitask architecture used during training; the upper part (first row) shows SR generator, from the LR to HR image, while the lower part (second row) is the extension used to predict segmentation class probabilities. The role of segmentation extension layers of our design is to force the feature extractor parameters learn categorical information. These non-shared layers, generating segmentation probabilities, are not used during SR run-time. Each part is presented in more details as follows:

- **SR generator** The generator network is a feed-forward CNN; the input image I^{LR} is passed through a convolution block followed by LeakyReLU activation layer. The output is subsequently passed through 16 residual blocks with skip connections. Each block has two convolutional layers with 3×3 filters and 64 feature maps, each one followed by a batch normalization and LeakyReLU activation. The output of the final residual block, concatenated with the features of the first convolutional layer, is inputted through two upsampling stages. Each stage doubles the input image size. Finally, the result is passed through a convolution stage to get the super-resolved image I^{SR}. In this study, we only investigate a scale factor of 4, but depending on the desired scaling, the number of upsampling stages can be changed.

- **Segmentation extension** The segmentation extension uses the output of the SR generator feature extractor part, just before the first upsampling stage, and convert it to a segmentation probability by passing it through two convolutional layers. The com-

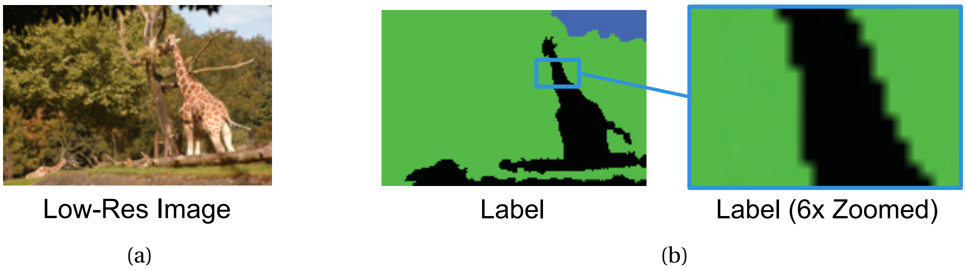


Figure 3.3 – An example showing the accuracy and resolution of a pixel-wise semantic segmentation label (b) of a low resolution image (a). As both segmentation and super-resolution networks share layers, the inaccurate segmentation labels result inaccurate edges in super-resolved images.

putational complexity of this stage needs to be as limited as possible, as we wish that shared-layers with SR generator learn categorical information and not only layers from segmentation extension.

The parameters of the generator, for both segmentation and SR tasks, are obtained by minimizing the $\mathscr{L}_{total}$ loss function presented in Section 3.3.3. This loss function consists also of a GAN [46]-based adversarial loss, which requires a discriminator network. This network discriminates real HR images from generated SR samples. We define our discriminator architecture similar to [19]; it consists of multiple convolutional layers with the kernels increasing by a factor of 2 from 64 to 512. We use Leaky ReLU and strided convolutions to reduce the image dimension while doubling the number of features. The resulting 512 feature maps are followed by two dense layers. Finally, the image is classified as real or fake by a final sigmoid activation function.

3.3.2 Boundary mask

Although segmentation labels of available datasets, e.g., [90], to be used for segmentation task, are created by an expensive labeling effort, they still lack of precision close to boundaries of different classes as can be seen in Figure 3.3. Our experiments show that as shared features are used for generating the SR image and segmentation probabilities, this lack of boundaries' precision in segmentation labels affects the edges in the SR image too. Therefore, we use a novel boundary mask ($M_{boundary}$) to filter out any segmentation losses from areas close to object boundaries from training images.

In order to generate such a boundary mask, first, we calculate the derivative of the segmentation label to get the boundaries of different classes in the low resolution image. Then, we compute the dilation of results with a disk of size d_1 to create a thicker strip around edges of each class. An example of converting the segmentation label to the boundary mask is shown in Figure 3.4. In Section 3.4 the effectiveness of using such boundary masks is shown.

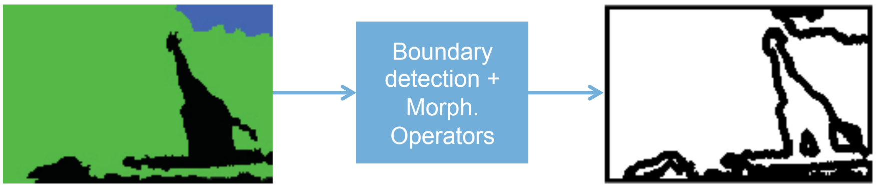


Figure 3.4 – The boundary mask generation. The black pixels of the results represent areas close to the edges while white pixels could be either background or foreground.

3.3.3 Loss function

We define the $\mathscr{L}_{total}$ as a combination of pixel-wise loss ($\mathscr{L}_{MSE}$), perceptual loss ($\mathscr{L}_{vgg}$), adversarial loss ($\mathscr{L}_{adv}$), and segmentation loss ($\mathscr{L}_{seg}$) filtered by our novel boundary mask ($M_{boundary}$) presented in Section 3.3.2. The overall loss function is given by:

$$\mathscr{L}_{total} = \alpha\mathscr{L}_{MSE} + \beta\mathscr{L}_{vgg} + \gamma\mathscr{L}_{adv} + \delta M_{boundary}.\mathscr{L}_{seg} \tag{3.1}$$

where α, β, γ, and δ are the corresponding weights of each loss term used to train our network. In the following, we present each term in detail:

- **Pixel-wise loss** The most common loss in SR is the pixel-wise Mean Squared Error (MSE) between the original image and the super-resolved image in the image space [19, 41, 79]. However, using it alone mostly results in finding pixel-wise averages of plausible solutions, which seems over-smoothed with poor perceptual qualities and lack of high-frequency details such as textures [39, 91, 92].

- **Perceptual loss** [41] and [19] used the idea of measuring the perceptual similarity by computing the distance of feature spaces of the images. First, both HR and SR images are mapped into a feature space by a pre-trained model, VGG-16 [40] in our case. Then, the perceptual loss is calculated by the $L2$ distance and using all 512 feature maps of ReLU 4-1 layer of the VGG-16.

- **Adversarial loss** Inspired by [19] we add the discriminator component of the mentioned GAN architecture to our design. This encourages our SR decoder to favor solutions that resolve more realistic and natural images, by trying to trick the discriminator network. It also results perceptually superior solutions to solutions obtained by minimizing pixel-wise MSE and perceptual loss.

- **Segmentation loss** While using segmentation for SR application is new for the community, semantic segmentation as a stand-alone task has been investigated for years. The most commonly used loss function for the task of image segmentation is a pixel-wise cross entropy loss (or log loss) [93, 94, 95]. In this work, we also use the cross entropy loss function to examine each pixel individually and compare the class predictions (depth-wise pixel vector) to the one-hot encoded label; it measures the performance of a pixel-wise classification model whose output is a probability value between zero and one for each pixel and category.

3.3.4 Dataset

Training the proposed network in a supervised manner requires a considerable number of training examples with ground-truths for both semantic segmentation and super resolution tasks. Therefore the choices of datasets are limited to the ones with available segmentation labels. We use a random sample of 60 thousand images from the COCO-Stuff database [90], which contains semantic labels for 91 stuff classes for segmentation task. We only choose images from five main background classes to be able to focus on texture quality and prove the concept: sky, ground, buildings, plants, and water. Each one of them contains multiple sub classes in COCO-Stuff dataset, e.g., water contains seas, lakes, rivers, etc. and plants contain trees, bushes, leaves, etc., but in this work we consider them as a single class. Any other object or background existing in an image is labeled as "others" (the sixth class). More than 12 thousand images from each category were used to train our network. We obtained the LR images for the SR task by downsampling the HR images of the same database using the MATLAB imresize function with the bicubic kernel and downsampling factor 4 (all experiments were performed with a scaling factor of ×4). For each image, we crop a random 82×82 HR sub image for training.

3.3.5 Training and parameters

In order to successfully converge to parameters compatible for both SR and the segmentation task, the training was done in different steps; first, the generator was trained for 25 epochs with only pixel-wise mean squared error as the loss function. Then the segmentation loss function was added and training continued for 25 more epochs. Finally, the loss function presented in Section 3.3.3 (including adversarial and perceptual losses) was used for 55 more epochs. The weights of each term in loss function presented in Eq. A.1 were chosen as follows: as proposed by [19], α, β, and γ were respectively fixed to 1.0, 2×10^{-6}, and 1×10^{-3}. δ were tuned and fixed to 0.8. The Adam optimizer [96] was used for all the steps. The learning rate was set to 1×10^{-3} and then was decayed by a factor of 10 every 20 epochs. We also alternately optimized the discriminator with the setting proposed by [19].

As explained previously, to not consider a segmentation prediction error close to boundaries of objects/backgrounds, the segmentation loss is filtered by a boundary mask as introduced in Section 3.3.2. Figure 3.5 shows the segmentation prediction results of two training images; the artifacts close to boundaries (imprecise edges and black strips around them) are the result of applying a boundary mask. This mask makes the network not consider the class probabilities around boundaries and have a random prediction on those areas.

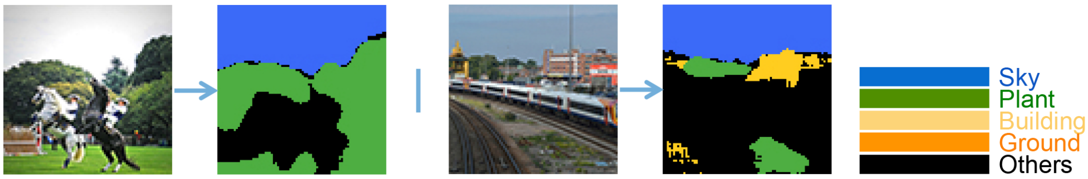


Figure 3.5 – Two examples of segmentation prediction results. The artifacts close to boundaries (imprecise edges and black strips around them) are the result of applying boundary mask in a way that the generator does not focus on class probabilities around boundaries and have a random prediction on those areas.

3.4 Results and discussion

In this section, we first investigate the effectiveness of using the presented boundary mask in the proposed approach. Then, we evaluate and discuss the benefits of introducing multitask learning for SR task by performing qualitative experiments, an extensive user study, and an ablation study. Finally, we discuss the computational time of the proposed approach.

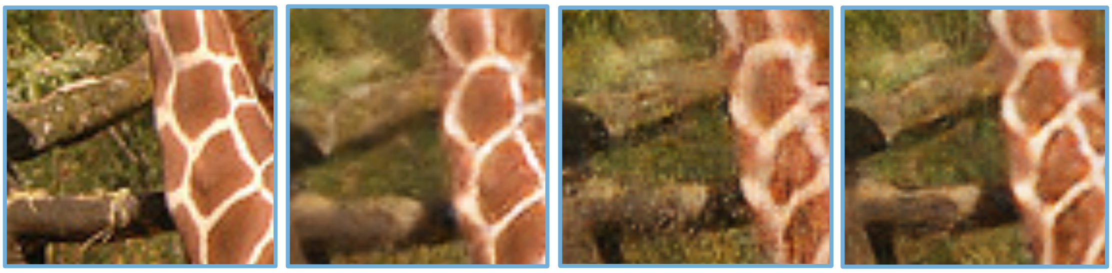
3.4.1 Effectiveness of boundary masks

As explained previously in Section 3.3.2 in this work we use a novel boundary mask ($M_{boundary}$) to filter out all segmentation losses from areas close to object boundaries during training. The goal of this masking is to avoid forcing SR network to learn imprecise boundaries existing in segmentation labels. Figure 3.6 shows the SR results comparing the effect of segmentation mask; comparing Figure 3.6.c to 3.6.d shows the improvement in reconstructing sharper edges using segmentation with mask rather than without mask. In this example, both Figures 3.6.c and 3.6.d have the closest textures to the ground-truth comparing to Figure 3.6.b, however, the object in the super-resolved image without using segmentation information has the sharpest edges; this can be explained by the fact that we only considered background categories (“sky”, “plant”, “buildings”, “ground”, and “water”) because of their specific appearance and to prove the concept. All type of objects, e.g., giraffe in this example, are included in “Other” category, therefore, no specific pattern is expected to be learnt for this category. As a future work, more object categories can be added to the training examples.

3.4.2 Qualitative results

Standard benchmarks such as Set5 [37], Set14 [38], and BSD100 [58] mostly do not contain the background categories studied in this research, therefore, first we evaluate our method on a test set consisting of random images of the COCO-stuff dataset [90].

Figure 3.7 contains visual examples comparing different models. In order to have a fair comparison, we re-trained the SRResNet [19], SFT-GAN [2], and SRGAN [19] methods on the same dataset and with the same parameters as ours. The generator and discriminator



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

Figure 3.6 – (a) Ground-truth, (b) SRGAN, (c) SRSEG, (d) Masked-SRSEG. While SRGAN still has the most accurate edges in this example, both masked and unmask SRSEG network constructs more realistic textures in the background and are closer to ground-truth. All images are cropped from Figure 3.3.a and zoomed by a factor of 6 (6×).

networks used in both SRGAN and our method are very similar (only layers resulting in segmentation probability output differ), which helps to investigate the effectiveness of our approach compared to the SRGAN, as the baseline. For RCAN, we used their pre-trained models in [35]. The MATLAB imresize function with a bicubic kernel is used to produce LR images.

The qualitative comparison shows that our method generates more realistic and natural textures by benefiting from categorical information. Our experiment shows that the trained model for both segmentation and SR tasks is generalized in a way that it reconstructs more realistic background compared to the approaches using the same configuration and without the segmentation objective.

As mentioned previously, to prove the concept, most of the test images contains specific background categories, however, it still reconstructs competitive results for objects without any labels during the training phase, e.g., the man with a tie in Figure 3.7. In some cases, we could also observe that our method can result in a less precise boundaries as shown in Figure 3.8.

3.4.3 User experience

As [2, 19, 41] mentioned, the commonly used quantitative measurements for SR methods, such as SSIM and PSNR, are not directly correlated to the perceptual quality; their experiments show that GAN-based methods have lower PSNR and SSIM values compared to PSNR-oriented approaches, however, they easily outperform them in terms of more appealing and closer images to the HR images. Therefore, we did not use these evaluation metrics in this work.

To better investigate the effectiveness of multitask learning simultaneously for semantic segmentation and SR, we perform a user study to compare the SRGAN [19] method and our approach which is a an extended version of SRGAN with an additional segmentation output.

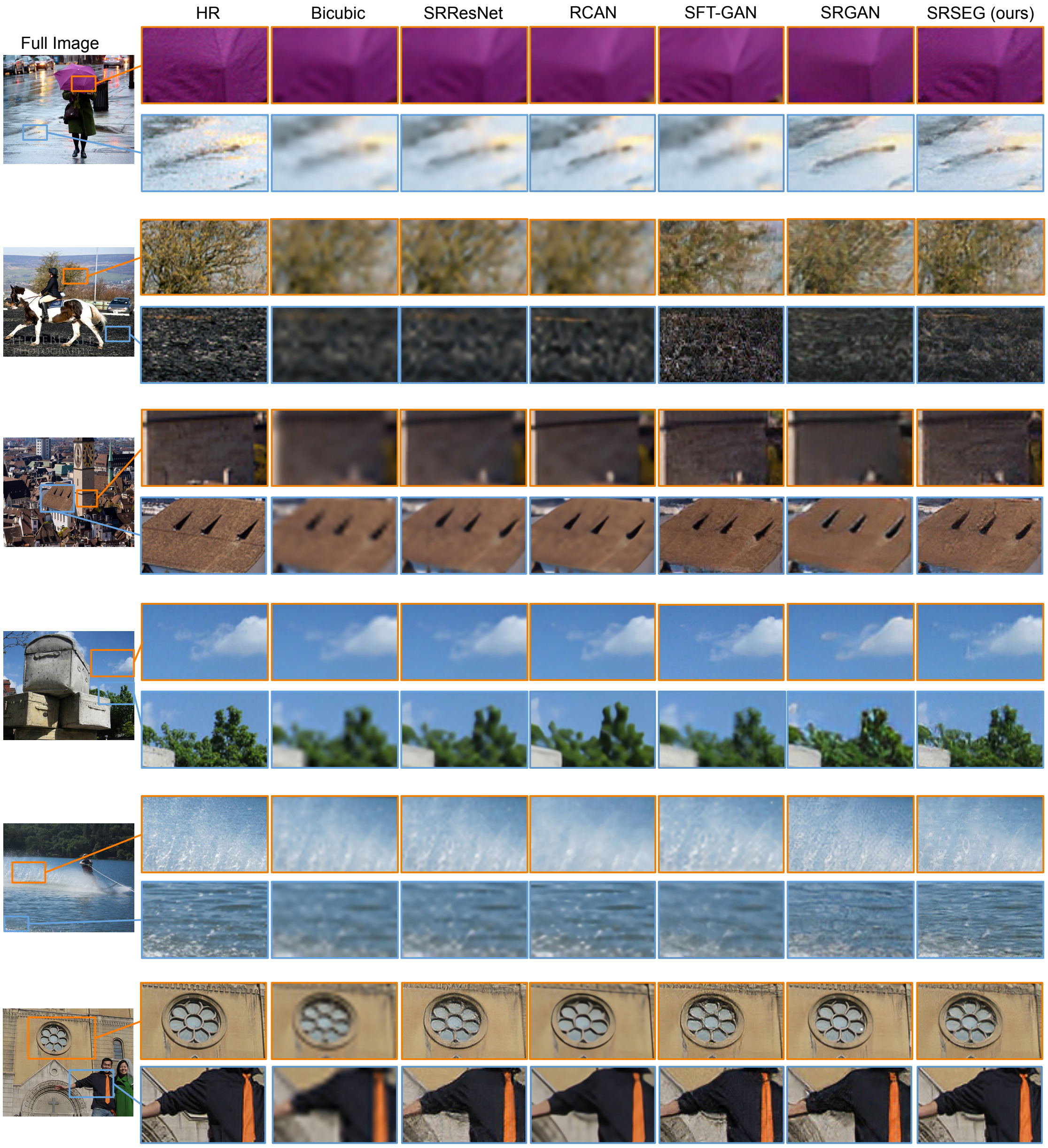


Figure 3.7 – Qualitative results on COCO-stuff dataset [90], focusing on object/background textures. The test images include images with the same categories as the one used during training (water, plant, building, sky, and ground). Cropped regions are zoomed in with a factor of 5 to 10. Images from left to right: High resolution image, bicubic interpolation, SRResNet [19], RCAN [35], SFT-GAN [2], SRGAN [19], and SRSEG (this work). Zoom in to have the best view.

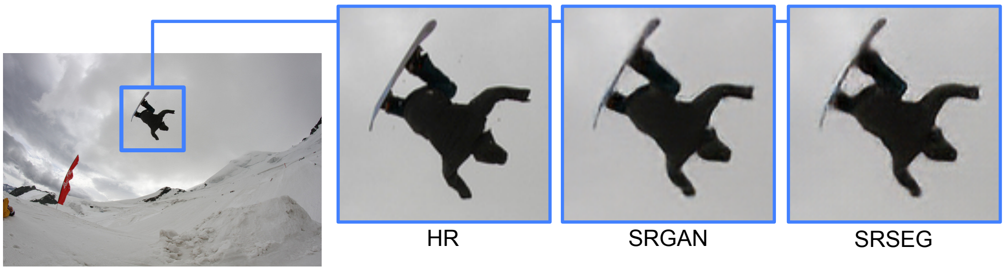


Figure 3.8 – An example of a bad reconstruction of boundaries compared to the SRGAN [19] method; this effect could be seen in some cases, specially in objects/backgrounds that have not been from training classes.

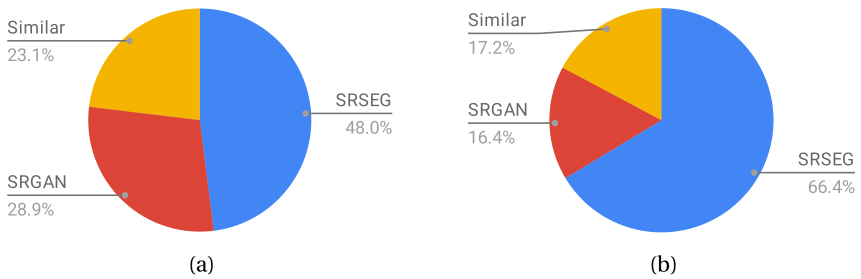
We design our experiment in two stages; first stage quantifies the ability of our approaches to reconstruct perceptually convincing images while we focus specifically on the quality of texture reconstruction regarding to ground-truth (real HR image).

During the first stage, users were requested to vote for more appealing images between SRGAN and our proposed method, SRSEG output pairs. In order to avoid random guesses in case of similar qualities, a third choice as "Similar" was also introduced for each image. 22 persons have participated in this experiment. 25 random images from COCO-Stuff [90] were presented in a randomized fashion to each person. The pie chart shown in Figure 3.9.a illustrates that the images reconstructed by our approach are more appealing to the users.

In the second stage, we focused only on enlarged texture patches, zoomed in with a factor of 8 to 10, mostly on parts of backgrounds that have been from training classes. The enlarged images represent only a reconstructed texture and no object was included in the image. The ground-truth was also shown to users. Each person was asked again to pick the texture closer to the ground-truth. 25 pairs of textures in addition to their ground-truth were shown to 22 persons in this stage. The results of this stage is shown in Figure 3.9.b. These results confirm that our approach reconstructs perceptually more convincing images for the users in terms of both overall and texture qualities of resolved images. However, comparing the results of the first and second stage of the user study shows that texture reconstruction quality of our proposed approach is by a large margin better than the quality of its object reconstruction. As a future work, adding more object categories to the training examples for both segmentation and SR tasks could also improve the reconstruction quality of the class “Others” with a similar margin.

3.4.4 Ablation study

Intuitively, by introducing additional segmentation task, our SR decoder extracts more specific features for both image reconstruction and semantic segmentation. To investigate the competence of these new features and the effectiveness of our approach for image SR, we perform an ablation study, by qualitatively comparing the reconstruction quality of our decoder, with and



(a) (b)

Figure 3.9 – The evaluation results of our user studies, comparing SRSEG (our method) with SRGAN [19]; (a) Focusing on visual quality of the resolved images, (b) Focusing only on enlarged textures. Both textures and overall qualities of resolved images resolved by our method are improved. Users prefer textures reconstructed by our proposed approach by a large margin.

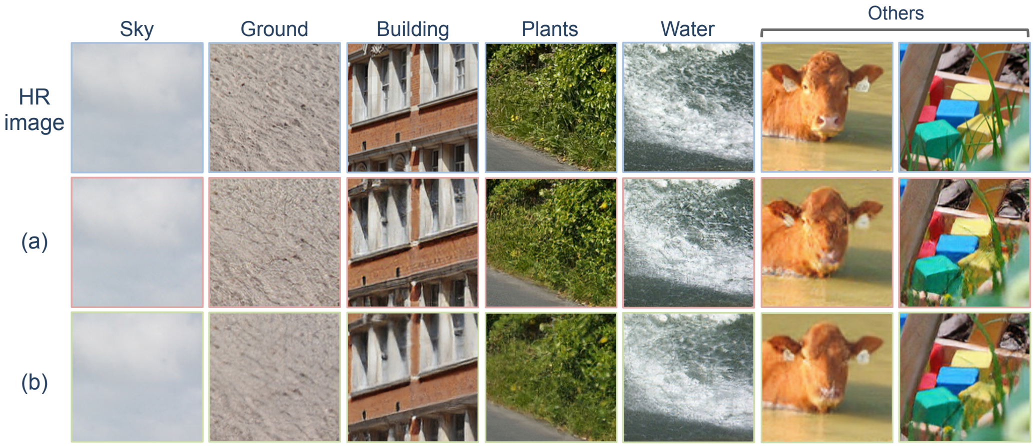


Figure 3.10 – Ablation study on different type of objects/backgrounds; comparing the reconstruction quality of our decoder: (a) with the segmentation extension during training, (b) without the segmentation extension. Zoom in for best view

without the segmentation extension. In Figure 3.10, we divide our results into different existing categories during training (sky, ground, buildings, plants, and water), as well as undefined categories in our dataset. We can see that the network trained with segmentation extension generates more photo-realistic textures for the available segmentation categories, while having competitive results for the other objects.

3.4.5 Results on standard benchmarks

During training, our approach focuses on optimizing the decoder by using an additional segmentation extension and loss term for recognizing specific categories, such as sky, ground, buildings, plants, and water. Even though many object and background categories are absent during the training phase, our experiment shows that the model generalizes in a way that it

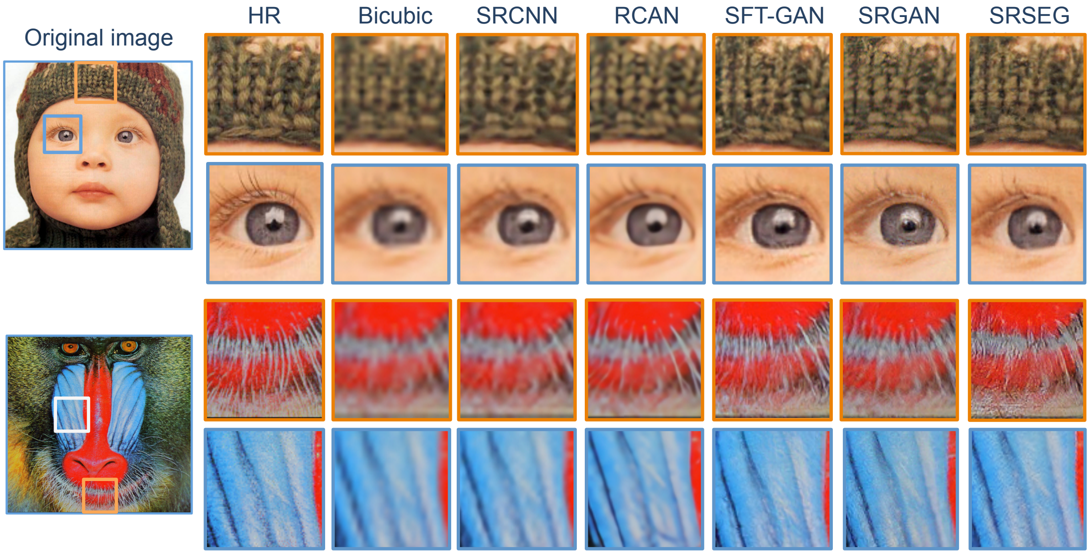


Figure 3.11 – Sample results on the "baby" (top) and "baboon" (bottom) images from Set5 [37] and Set14 [37] datasets, respectively. From left to right: HR image, bicubic, SRCNN [11], RCAN [35], SFT-GAN [2], SRGAN [19], and SRSEG (ours). Zoom in for the best view.

reconstructs either more realistic or competitive results for undefined objects/backgrounds as well. In this section, we evaluate the reconstruction quality of unknown objects, by using Set5 [37] and Set14 [37] standard benchmarks, where unlike our training set, in most of the images, outdoor background scenes are not present. Figure 3.11 compares the results of our SR model on the "baby" and the "baboon" images to recent state-of-the-art methods including bicubic, SRCNN [11], RCAN [35], SFT-GAN [2], and SRGAN [19]. In both images, despite the fact that their categories were not existed during training, we could generate more photo-realistic images compared to SRCNN and RCAN, while having competitive results with SFT-GAN and SRGAN. Their results were obtained by using their online supplementary materials.

3.4.6 Computational time

Our proposed method has similar running time to CNN-based SISR methods and faster than method such as [2], which uses a second network to predict segmentation probabilities. As the additional extension for segmentation, presented in this work, is removed at run-time and no segmentation label is required as an input, the running time is not affected by our proposed approach. However, using segmentation extension during the training phase increases our training time with a factor of 1.3 compared to SRGAN.

In particular, our Tensorflow implementation runs at 20.24 FPS on a GeForce GTX 1080 Ti graphic card to reconstruct HD images (1024×768) from their low-resolution counter-parts (256×192) with a scale factor of 4.

3.5 Conclusion and future work

In this work we presented a novel approach to use categorical information to tackle the SR problem. We introduced a SR decoder only benefiting from one shared deep network to learn simultaneously image SR and semantic segmentation by keeping two task-specific output layers during training. We also introduced a novel boundary mask to filter out unrelated segmentation losses caused by imprecise segmentation labels. We have conducted perceptual experiments including a user study on images from COCO-Stuff dataset and demonstrated that multitask learning can enable benefiting from semantic information in a single network and improves the recovering quality. As a future work, additional object/background categories can be introduced during the training in order to explore how it could affect the reconstruction quality.

4 Spatial Control Over Image Generation Process

As mentioned in Chapters 2 and 3, despite variant architectures proposed for the super-resolution task, the behavior of optimization-based methods are principally driven by the choice of the objective function. However, common SR objective functions do not take the global semantic information within images into account and estimate a reconstruction error for an entire image spatially in the same way. This chapter introduces a novel objective function that activates learning by benefiting from categorical information of input images; it mainly focuses on perceptual losses and uses the novel OBB (Object, boundary, and background) labels to add additional spatial control over the learning process. This chapter particularly introduces a new targeted perceptual loss that appropriately penalizes each region of the image during training, i.e., a suitable boundary loss for edges and texture loss for textures.

In the following, this chapter is presented in the form of an article. The research and all experiments were performed by the first author (the author of this thesis). The manuscript was also written by him and was further revised by other authors. The article was presented at the ICCV 2019 conference and published in the **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019**, *pp. 2710-2719, DOI: 10.1109/ICCV.2019.00280*, with the original title of **SROBB: Targeted Perceptual Loss for Single Image Super-Resolution**. The supporting information of this article can be found in Appendix A.1.

SROBB: Targeted Perceptual Loss for Single Image Super-Resolution

Authors: Mohammad Saeed Rad[1], Behzad Bozorgtabar[1], Urs-Viktor Marti[2], Max Basler[2], Hazım Kemal Ekenel[1, 3], Jean-Philippe Thiran[1].

[1] Signal Processing Laboratory 5, EPFL, Lausanne, Switzerland.
[2] AI Lab, Swisscom AG, Lausanne, Switzerland.
[3] Istanbul Technical University, Istanbul, Turkey.

abstract

By benefiting from perceptual losses, recent studies have improved significantly the performance of the super-resolution task, where a high-resolution image is resolved from its low-resolution counterpart. Although such objective functions generate near-photorealistic results, their capability is limited, since they estimate the reconstruction error for an entire image in the same way, without considering any semantic information. In this chapter, we propose a novel method to benefit from perceptual loss in a more objective way. We optimize a deep network-based decoder with a targeted objective function that penalizes images at different semantic levels using the corresponding terms. In particular, the proposed method leverages our proposed OBB (Object, Background and Boundary) labels, generated from segmentation labels, to estimate a suitable perceptual loss for boundaries, while considering texture similarity for backgrounds. We show that our proposed approach results in more realistic textures and sharper edges, and outperforms other state-of-the-art algorithms in terms of both qualitative results on standard benchmarks and results of extensive user studies.

Keyword Single Image Super-Resolution, Targeted Perceptual Loss, Image Semantic Meaning, Generative Adversarial Network

4.1 Introduction

Single image super-resolution (SISR) aims at solving the problem of recovering a high-resolution (HR) image from its low-resolution (LR) counterpart. SISR is a classic ill-posed problem that has been one of the most active research areas since the work of Tsai and Huang [1] in 1984. In recent years, this problem has been revolutionized by the significant advances in convolutional neural networks (CNNs) and has resulted in better reconstructions of high-resolution pictures than classical approaches [11, 25, 79]. More recently, another breakthrough has been made in SISR by employing perceptual loss functions for training feed-forward networks,

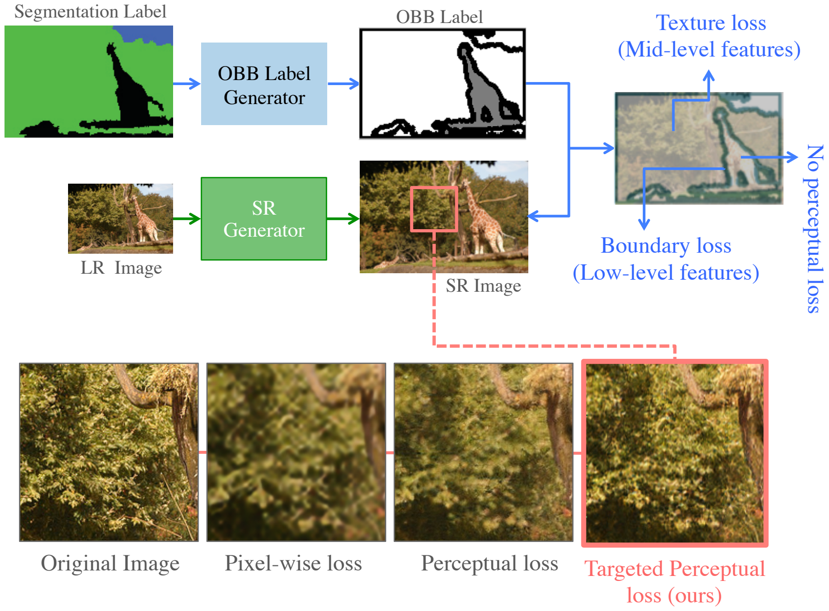


Figure 4.1 – We propose a method for exploiting the segmentation labels during training to resolve a high resolution image at different semantic levels considering their characteristics; we optimize our SISR model by minimizing perceptual errors that correspond to edges only at object boundaries and the texture on the background area, respectively. Results from left to right: original image, super-resolved images using only pixel-wise loss function, pixel-wise loss + perceptual loss function and pixel-wise loss + targeted perceptual loss function (ours), respectively.

instead of using per-pixel loss functions, e.g., mean squared error (MSE) [19, 41, 80]. It tackled the problem of blurred textures caused by optimization of MSE, and alongside with adversarial loss [46], it resulted in near-photorealistic reconstruction in terms of perceived image quality.

[41] and [19] benefit from the idea of using perceptual similarity as a loss function; they optimize their models by comparing the ground-truth and the predicted super-resolved image (SR) in a deep feature domain by mapping both HR and SR images into a feature space using a pre-trained classification network. Although this similarity measure in feature space, namely the perceptual loss, has shown a great success in SISR, applying it as it is on a whole image, without considering the semantic information, limits its capability.

To better understand this limitation, let us have a brief overview of the perceptual loss and see what a pre-trained classification network optimizes; considering a pre-trained CNN, in an early convolutional layer, each neuron has a receptive field with the size and shape of the inputs that affects its output. Small kernels, which are commonly used by state-of-the-art approaches, have also small receptive fields. As a result, they can only extract low-level spatial information. Intuitively, each neuron captures relations between nearby inputs considering their local spatial relations. These local relations are mostly presenting information about edges and blobs. As we proceed deeper in the network, the receptive field of each neuron with respect to earlier layers becomes larger. Therefore, deep layers start to learn features with global semantic meanings and abstract object information, and less fine-grained spatial details, while still using small kernels. This fact has also been shown by [97, 98], where they used some visualization techniques and investigated the internal working mechanism of the VGG network [40] by visualization of the information kept in each CNN layer.

Regarding the perceptual function, state-of-the-art approaches use different levels of features to restore the original image; this choice determines whether they focus on local information such as edges, mid-level features such as textures or high-level features corresponding to semantic information. In these works, perceptual loss has been calculated for an entire image in the same way, meaning that the same level of features has been used either on edges, foreground or on the image background. For example, minimizing the loss for details of the edges inside a random texture, such as the texture of a tree, would force the network to consider an unnecessary penalty and learn less informative features; the texture of a tree could still be realistic in the SR image without having close edges to the HR image. On the other hand, minimizing the loss by using mid-level features (more appropriate for the textures) around edges would not intuitively create sharper edges and would only introduce “noisy” losses.

To address the above issue, we propose a novel method to benefit from perceptual loss in a more objective way. Figure 4.1 shows an overview of our proposed approach. In particular, we use pixel-wise segmentation annotations to build our proposed OBB labels to be able to find targeted perceptual features that can be used to minimize appropriate losses to different image areas: e.g., edge loss for edges and textures’ loss for image textures during training. We show that our approach using targeted perceptual loss outperforms other state-of-the-art algorithms in terms of both qualitative results and user study experiments, and result in more realistic textures and sharper edges.

4.2 Related work

In this section, we review relevant CNN-based SISR approaches. This field has witnessed a variety of end-to-end deep network architectures: [25] formulated a recursive CNN and showed how deeper network architectures increase the performance of SISR. [19, 33, 41] used the concept of residual blocks [18] and skip-connections [23, 25] to facilitate the training of CNN-based decoders. [24] improved their models by expanding the model size. [43] removed batch normalization in conventional residual networks and used several skip connections to improve the results of seminal work of [19]. Laplacian pyramid structure [28] has been proposed to progressively reconstruct the sub-band residuals of high-resolution images. [27] proposed a densely connected network that uses a memory block consisting of a recursive unit and a gate unit, to explicitly mine persistent memory through an adaptive learning process. [35] proposed a channel attention mechanism to adaptively rescale channel-wise features by considering the inter-dependencies among channels. Besides supervised learning, other methods like unsupervised learning [45] and reinforcement learning [99] were also introduced to solve the SR problem.

Despite variant architectures proposed for the SISR task, the behavior of optimization-based methods is principally driven by the choice of the objective function. The objective functions used by these works mostly contain a loss term with the pixel-wise distance between the super-resolved and the ground-truth HR images. However, using this function alone leads to blurry and over-smoothed super-resolved images due to the pixel-wise average of all plausible solutions.

Perceptual-driven approaches added a remarkable improvement to image super-resolution in terms of the visual quality. Based on the idea of perceptual similarity [39], perceptual loss [80] is proposed to minimize the error in a feature space using specific layers of a pre-trained feature extractor, for example VGG [40]. A number of recent papers have used this optimization to generate images depending on high-level extracted features [100, 101, 102, 103, 104]. In a similar work, contextual loss [86] is proposed to generate images with natural image statistics, which focuses on the feature distribution rather than merely comparing the appearance. [19] proposed to use adversarial loss in addition to the perceptual loss to favor outputs residing on the manifold of natural images. The SR method in [41] develops a similar approach and further explores a patch-based texture loss. Although these works generate near-photorealistic results, they estimate the reconstruction error for an entire image in the same way, without benefiting from any semantic information that could improve the visual quality.

Many studies such as [3, 4, 5] also benefit from prior information for SISR. Most recently, [2] used an additional segmentation network to estimate probability maps as prior knowledge and used them in the existing super-resolution networks. Their segmentation network is pre-trained on the COCO dataset [88] and then is fine-tuned on the ADE dataset [89]. Their approach recovers more realistic textures faithful to categorical priors; however, it requires a segmentation map at test-time. In Chapter 3 of this thesis, we addressed this issue by proposing a method based on multitask learning simultaneously for SR and semantic segmentation tasks.

In this work, we investigate a novel way to exploit semantic information within an image, yielding photo-realistic super-resolved images with fine-structures.

4.3 Methodology

Following recent approaches [2, 19, 105] for image and video super-resolution, we benefit from deep networks with residual blocks to build-up our decoder. As explained previously, in this chapter, we focus on the definition of the objective function used to train our network; we introduce a loss function containing three terms: 1- Pixel-wise loss (MSE), 2- adversarial loss, and 3- our novel targeted perceptual loss function. The MSE and adversarial loss terms are defined as follows:

- **Pixel-wise loss** It is by far the most commonly used loss function in SR. It calculates the pixel-wise mean squared error (MSE) between the original image and the super-resolved image in the image domain [11, 12, 41]. The main drawback of using it as a stand-alone objective function is mostly resolving an over-smoothed reconstruction. The network trained with the MSE loss seeks to find pixel-wise averages of plausible solutions, which results in poor perceptual qualities and lack of high-frequency details in the edges and textures.
- **Adversarial loss** Inspired by [19], we formulate our SR model in an adversarial setting, which provides a feasible solution. In particular, we use an additional network (discriminator) that is alternatively trained to compete with our SR decoder. The generator (SR decoder) tries to generate fake images to fool the discriminator, while the discriminator aims at distinguishing the generated results from real HR images. This setting results in perceptually superior solutions to the ones obtained by minimizing pixel-wise MSE and classic perceptual losses. The discriminator used in this work is defined in more details in Section 4.3.3.

Our proposed targeted perceptual loss is described in the following subsection.

4.3.1 Targeted perceptual loss

The state-of-the-art approaches such as [41] and [19] estimate perceptual similarity by comparing the ground-truth and the predicted super-resolved image in a deep feature domain by mapping both HR and SR images into a feature space using a pre-trained classification network, e.g., VGG [40]. The output of a specific convolutional layer is used as the feature map. These approaches usually minimize the l_2 distance of the feature maps. In order to understand why minimizing this loss term in combination with adversarial and MSE losses is effective and results in more photorealistic images, we investigate the nature of the CNN layers used for the perceptual loss. Then, we propose a novel approach to take advantage of the perceptual similarity in a targeted manner and reconstruct more appealing edges and textures.

As explained previously, early layers of a CNN return low-level spatial information regarding local relations, such as information about edges and blobs. As we proceed towards deeper layers, we start to learn higher level features with more semantic meaning and abstract object information, and less fine-grained spatial details from an image. In this fashion, mid-level features are mostly representing textures and high-level features amount to the global semantic meaning. Figure 4.2 shows the difference between shallow and deep layers of a feature extractor, the VGG-16 in our case; two different layers, ReLU 1-2 and ReLU 4-1, are used to compute the perceptual loss and reconstruct an image. We compare each case on an

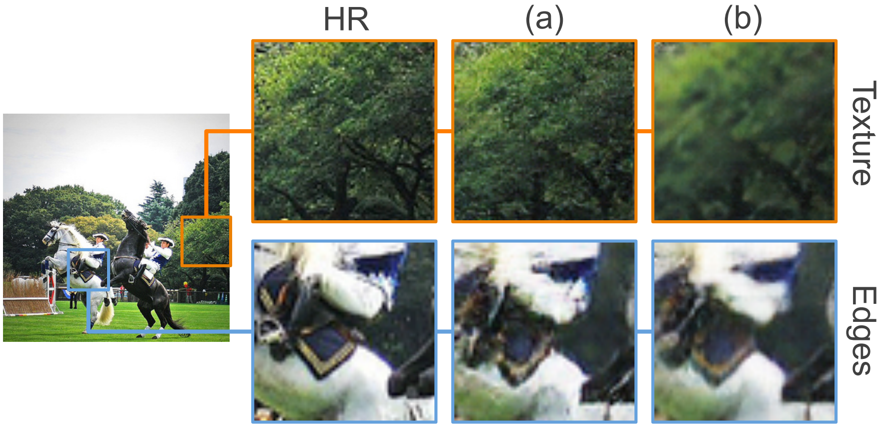


Figure 4.2 – The effect of choosing different CNN layers to estimate the perceptual loss on different regions of an image, e.g., edges and textures: (a) using a deeper convolutional layer (mid-level features), ReLU 4-1 of VGG-16 [40] and, (b) using an early convolutional layer (low-level features), ReLU 1-2 of the VGG-16 network.

edge and a texture region. In this figure, we can see using low-level features is more effective for reconstructing edges, while mid-level features resolve closer textures to the original image.

The targeted loss function tries to favor more realistic textures around areas, where the type of the textures seems to be important, e.g., a tree, while trying to resolve sharper edges around boundary area. To do so, we first define three types of regions in an image: 1- background, 2- boundaries, and 3- objects, then, we compute the targeted perceptual loss for each region using a different function.

- **Background** ($\mathcal{G}_b$) We consider four classes as background: “sky”, “plant”, “ground” and “water”. We chose these categories because of their specific appearance; the overall texture in areas with these labels are more important than local spatial relations and edges. We compute mid-level CNN features to estimate the perceptual similarity between SR and HR images. Here, we use the ReLU 4-3 layer of the VGG-16 for this purpose.
- **Boundary** ($\mathcal{G}_e$) All edges separating objects and the background are considered as boundaries. With some pre-processing (explained in more detail in Section 4.3.2), we broaden these edges to have a strip passing through all boundaries. We estimate the feature distance of an early CNN layer between SR and HR images, which focuses more on low-level spatial information, mainly edges and blobs. In particular, we minimize the perceptual loss at the ReLU 2-2 layer of the VGG-16.
- **Object** ($\mathcal{G}_o$) Because of the huge variety of objects in the real world in terms of shapes and textures, it is challenging to decide whether it is more appropriate to use features from early or deeper layers for the perceptual loss function; for example, in an image of a zebra, sharper edges are more important than the overall texture. Having said

> that, forcing the network to estimate the precise edges in a tree could mislead the optimization procedure. Therefore, we do not consider any type of perceptual loss on areas defined as objects by weighting them to zero and rely on the MSE and adversarial losses. However, intuitively, resolving more realistic textures and sharper edges by the "background" and "boundary" perceptual loss terms would result in more appealing objects, as well.

To compute the perceptual loss for a specific image region, we make binary segmentation masks of the semantic classes (having a pixel value of 1 for the class of interest and 0 elsewhere). Each mask categorically represents a different region of an image and is element-wise multiplied by the HR image and the estimated super-resolved image SR, respectively. In other words, for a given category, the image is converted to a black image with only one visible area on it, before being passed through the CNN feature extractor. Masking an image in this way creates also new artificial boundaries between black regions and the visible class. As a consequence, extracted features contain information about the artificial edges which do not exist in a real image. As the same mask is applied on both HR and the reconstructed image, the feature distance between these artificial edges will be close to zero and it does not affect the total perceptual loss. We can conclude that all non-zero distances in feature space between the masked HR and super-resolved image are corresponds to the contents of the visible area of that image: corresponds to edges by using a mask for boundaries ($M_{OBB}^{boundaries}$) and corresponds to textures by using a mask for the background ($M_{OBB}^{background}$).

The overall targeted perceptual loss function is given as:

$$\begin{aligned}\mathscr{L}_{perc.} = \alpha \cdot \mathscr{G}_e(I^{SR} \circ M_{OBB}^{boundary}, I^{HR} \circ M_{OBB}^{boundary}) \\ + \beta \cdot \mathscr{G}_b(I^{SR} \circ M_{OBB}^{background}, I^{HR} \circ M_{OBB}^{background}) \\ + \gamma \cdot \mathscr{G}_o \end{aligned} \tag{4.1}$$

where α, β and γ are the corresponding weights of the loss terms used for the boundary, background, and object, respectively. $\mathscr{G}_e(\cdot)$, $\mathscr{G}_b(\cdot)$ and $\mathscr{G}_o(\cdot)$ are the functions to calculate feature space distances between any two given images for the boundaries, background, and objects, respectively. In this equation, $\circ$ denotes element-wise multiplication. As discussed earlier, we do not consider any perceptual loss for objects areas, therefore, we set γ directly to zero. The value of other weights are discussed in detail in Section 4.4.1.

In the following subsection, we describe how to build a label indicating objects, the background, and boundaries for the training images. This labeling approach helps us to use specific masks for each class of interest (M_{OBB}^{object}, $M_{OBB}^{background}$ and $M_{OBB}^{boundary}$) and to guide our proposed perceptual losses to focus on area of interest within the image.

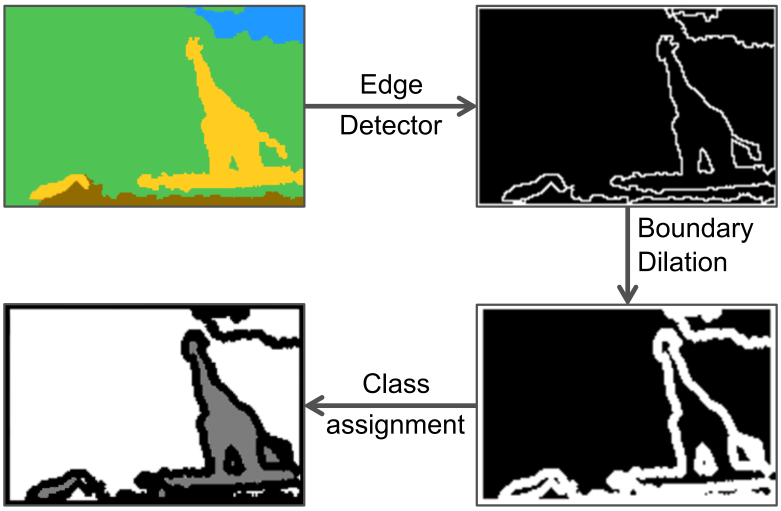


Figure 4.3 – Constructing an OBB label. We assign each area to one of the “Object”, “Background” or “Boundary” classes based on their initial pixel-wise labels.

4.3.2 OBB: Object, background and boundary label

In order to make full use of the perceptual loss-based image super-resolution, we enforce semantic details (where objects, the background, and boundaries appear on the image) via our proposed targeted loss function. In addition, existing annotations for the segmentation task, e.g., [90] only provide spatial information about objects and the background, and they do not use classes representing the edge areas, namely boundaries in this paper. Therefore, we propose our labeling approach (Figure 4.3) to provide a better spatial control of the semantic information for the images.

To create such labels (OBB label), first, we calculate the derivative of the segmentation label in the color-space to estimate the edges between object classes in the segmentation label as well as the edges between objects and background of the image. In order to have a thicker strip around all edges separating different classes, we compute the dilation with a disk of size d_1. We label the resulted area as “boundary” class, which covers boundaries between different classes inside an image. In particular, we consider “sky”, “plant”, “ground”, and “water” classes from the segmentation labels as the “Background”. All remaining object classes are considered as the “object” class.

4.3.3 Architecture

For a fair comparison with the SRGAN method [19] and performing an ablation study of the proposed targeted perceptual loss, we use the same SR decoder as the SRGAN. The generator network is a feed-forward CNN. The input image I^{LR} is passed through a convolution block followed by a ReLU activation layer. The output is subsequently passed through 16 residual

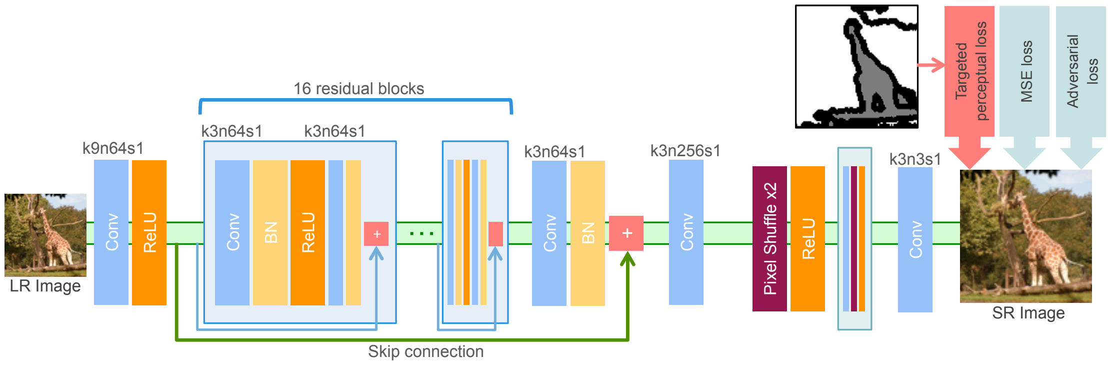


Figure 4.4 – Schematic diagram of the SR decoder. We train the SR decoder using the targeted perceptual loss alongside with MSE and adversarial losses. In this schema, k, n and s correspond to kernel size, number of feature maps and stride size, respectively.

blocks with skip connections. Each block has two convolutional layers with 3 × 3 filters and 64 channels feature maps, each one followed by a batch normalization and ReLU activation. The output of the final residual block is concatenated with the features of the first convolutional layer and is then passed through two upsampling blocks, where each one doubles the size of the feature map. Finally, the result is filtered by a last convolution layer to get the super-resolved image I^{SR}. In this chapter, we use a scale factor of four; depending on the desired scaling factor, the number of upsampling blocks could be modified. An overview of the architecture is shown in Figure 6.4.

The discriminator network consists of multiple convolutional layers with an increasing number of channels of the feature maps by a factor of 2, from 64 to 512. We use Leaky-ReLU and strided convolutions to reduce the image dimension while doubling the number of features. The resulting 512 feature maps are passed through two dense layers. Finally, the discriminator network classifies the image as real or fake by the final sigmoid activation function.

4.4 Experimental results

In this section, first, we describe the training parameters and dataset in details, then we evaluate our proposed method in terms of qualitative, quantitative, and running costs analysis.

4.4.1 Dataset and parameters

To create OBB labels, we use a random set of 50K images from the COCO-Stuff dataset [90], which contains semantic labels of 91 classes for the segmentation task. In this work, we considered landscapes with one or more of the “Sky”, “Plant”, “Ground”, and “Water” classes. We group these classes into one “Background” class. We use our proposed technique in Section 4.3.2 to convert pixel-wise segmentation annotations to OBB labels. In order to obtain

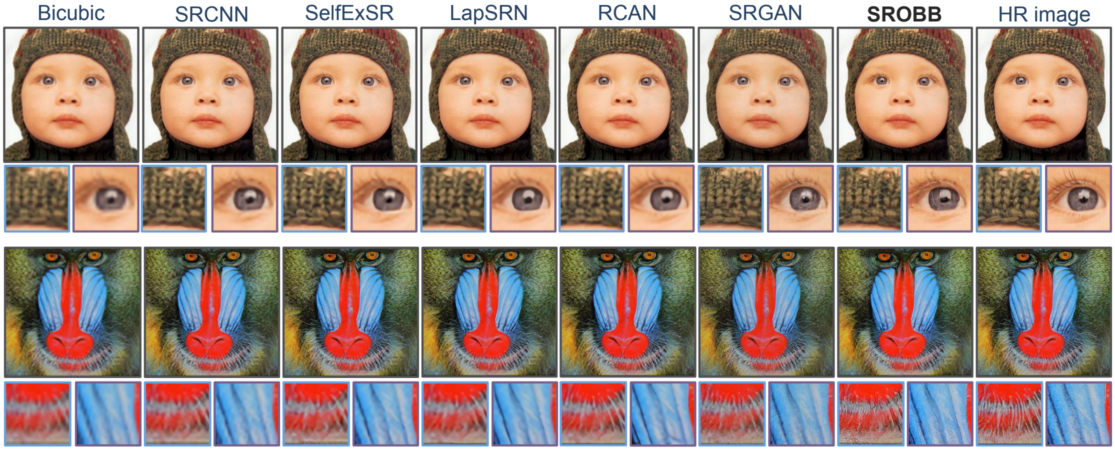


Figure 4.5 – Sample results on the "baby" (top) and "baboon" (bottom) images from Set5 [37] and Set14 4.5 datasets, respectively. From left to right: bicubic, SRCNN [11], SelfExSR [59], LapSRN [28], RCAN [35], SRGAN [19] and SROBB (ours), HR image, respectively.

LR images, we use the MATLAB imresize function with the bicubic kernel and the anti-aliasing filter. All experiments were performed with a downsampling factor of four.

The training process was done in two steps; first, the SR decoder was pre-trained for 25 epochs with only pixel-wise mean squared error as the loss function. Then the proposed targeted perceptual loss function, as well as the adversarial loss were added and the training continued for 55 more epochs. The weights of each term in the new targeted perceptual loss, α and β, were set to 2×10^{-6} and 1.5×10^{-6}, respectively. The weights of adversarial and MSE loss function, as in [19], were set to 1.0 and 1×10^{-3}, respectively. We set $d1$, the diameter of the disk used to generate OBB labels, to 2.0. The Adam optimizer [96] was used during both steps. The learning rate was set to 1×10^{-3} and then decayed by a factor of 10 every 20 epochs. We also alternately optimized the discriminator with similar parameters to those proposed by [19].

4.4.2 Qualitative results

Results on Set5 and Set14

Our approach focuses on optimizing the decoder with perceptual loss terms targeting boundaries and background by exploiting segmentation labels. Although, we do not apply the perceptual losses specifically on objects regions, our experiment shows that the trained model generalized in a way that it reconstructs more realistic objects compared to other approaches. We evaluate the quality of object reconstruction by performing qualitative experiments on two widely used benchmark datasets: Set5 [37] and Set14 [38], where unlike our training set, in most of the images, outdoor background scenes are not present. Figure 4.5 compares the results of our SR model on the "baby" and "baboon" images and the recent state-of-the-art methods including: bicubic, SRCNN [11], SelfExSR [59], LapSRN [28], RCAN [35] and SRGAN [19]. In the “baboon” image, we could generate more photo-realistic images with sharper edges compared to other methods while having competitive results for the “baby” image with SRGAN. Their results were obtained by using their online supplementary materials [1] [2] [3]. More qualitative results of Set5 and Set14 images are provided in the supplementary material.

Results on the COCO-Stuff dataset

We randomly chose a set of test images from the COCO-Stuff dataset [90]. In order to have a fair comparison, we re-trained the SFT-GAN[2], ESRGAN [43] and SRGAN [19] methods on the same dataset with the same parameters as ours. For the EnhanceNet and RCAN, we used their pre-trained models by [41] and [35], respectively. The MATLAB imresize function with a bicubic kernel is used to produce bicubic images. As illustrated in Figure 4.6, our method generates more realistic and natural textures by benefiting from our proposed targeted perceptual loss. Although ESRGAN produces very competitive results, it seems that their method is biased towards over-sharpened edges, which sometime leads to an unrealistic reconstruction and dissimilar to ground-truth.

4.4.3 Quantitative results

SSIM, PSNR and LPIPS

As it is shown in [2, 19, 41, 68], distortion metrics such as the Structural Similarity Index (SSIM) [106] or the Peak Signal to Noise Ratio (PSNR) used as quantitative measurements, are not directly correlated to the perceptual quality; they demonstrate that GAN-based super-resolved images could have higher errors in terms of the PSNR and SSIM metrics, but still generate more appealing images.

In addition, we used the perceptual similarity distance between the ground-truth and super-resolved images. The Learned Perceptual Image Patch Similarity (LPIPS) metric [65] is a recently introduced as a reference-based image quality assessment metric, which seeks to estimate the perceptual similarity between two images. This metric uses linearly calibrated off-the-shelf deep classification networks trained on the very large Berkeley-Adobe Perceptual Patch Similarity (BAPPS) dataset [65], including human perceptual judgments. However, as [66] also emphasizes, LPIPS has similar trend as distortion-based metrics, e.g., SSIM, and would not necessarily imply photorealistic images.
Table 5.1 shows the SSIM, PSNR, and LPIPS values estimated between super-resolved images of the “baby” and “baboon” and their HR counterparts, using bicubic interpolation, LapSRN [28], SRGAN [19], and our method, respectively. Considering this table and the visual comparison

---

[1] https://github.com/jbhuang0604/SelfExSR

[2] https://github.com/phoenix104104/LapSRN

[3] https://twitter.app.box.com/s/lcue6vlrd01ljkdtdkhmfvk7vtjhetog

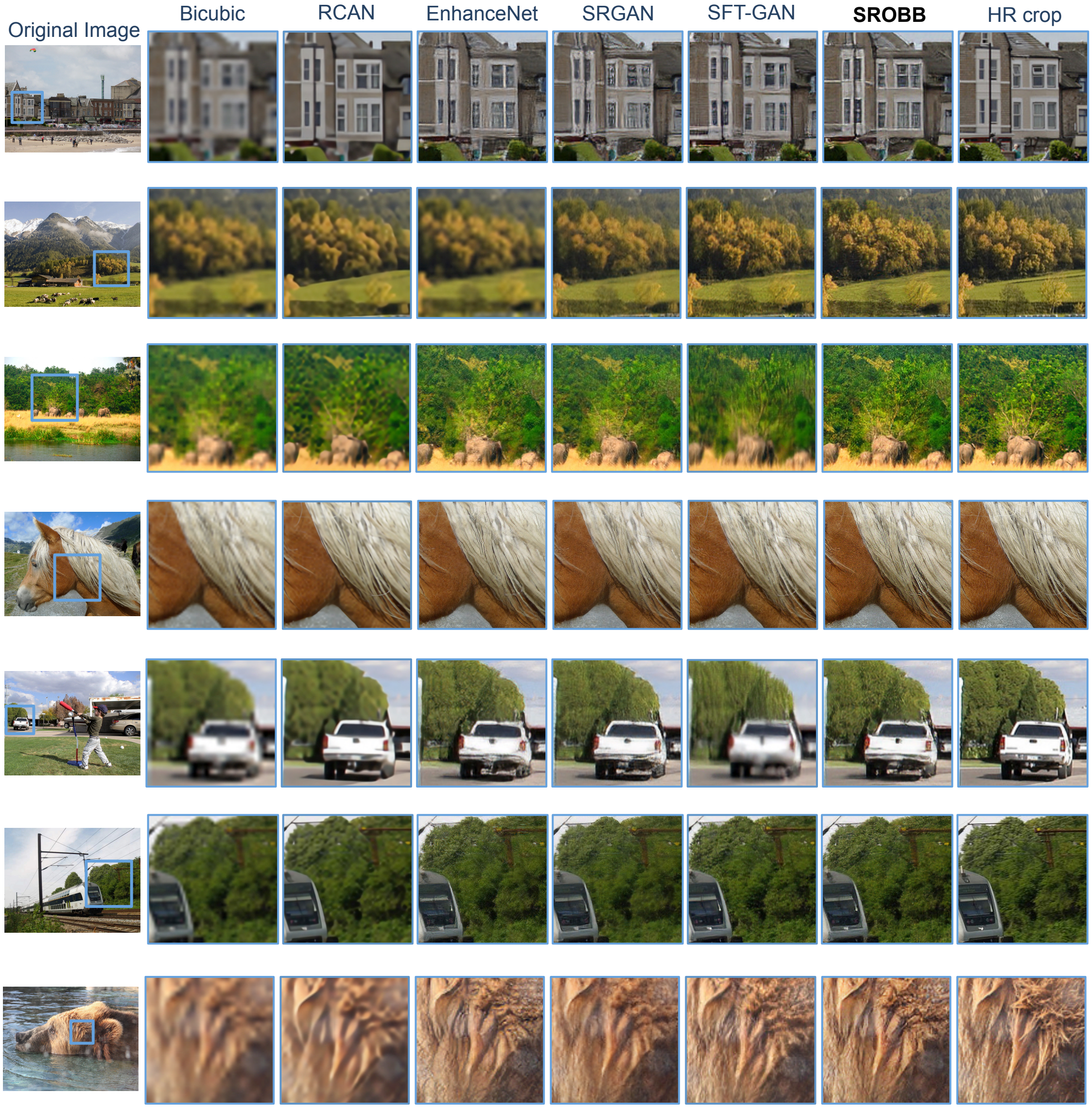


Figure 4.6 – Qualitative results on a subset of the COCO-Stuff dataset [90] images. Cropped regions are zoomed in with a factor of 2 to 5 to have a better comparison. Results from left to right: bicubic, RCAN [35], EnhanceNet [41], SRGAN [19], SFT-GAN [2], ESRGAN [43], SROBB (ours) and a high resolution image. Zoom in for the best view.

Image	Metric	Bicubic	LapSRN	SRGAN	SROBB
baby	SSIM	0.936	**0.951**	0.899	0.905
	PSNR	30.419	**32.019**	28.413	28.869
	LPIPS	0.305	0.237	0.112	**0.104**
baboon	SSIM	0.645	**0.677**	0.615	0.607
	PSNR	20.277	**20.622**	19.147	18.660
	LPIPS	0.632	0.537	**0.220**	0.245

Table 4.1 – Comparison of bicubic interpolation, LapSRN [28], SRGAN [19] and SROBB (ours) for the “baby” and “baboon” images from Set5 and Set14 test sets. Best measures (SSIM, PSNR [dB], LPIPS) are highlighted in bold. The visual comparison is shown in Figure 4.5.

of these images in Figure 4.5, we can infer that these metrics would not reflect superior reconstruction quality. Therefore, in the following section, we focus on the user study as the quantitative evaluation.

User study

We performed a user study to compare the reconstruction quality of different approaches to see which images are more appealing to users. Five methods were used in the study: 1- RCAN [35], 2- SRGAN [19], 3- SFT-GAN [2], 4- ESRGAN [43] and 5- SROBBB (ours). During the experiment, high-resolution images as well as their five reconstructed counterparts obtained by the mentioned approaches were shown to each user. Users were requested to vote for more appealing images with respect to the ground-truth image. In order to avoid random guesses in case of similar qualities, a choice as “Cannot decide” was also designed. Since SFT-GAN uses a segmentation network trained on outdoor categories, for a fair comparison with [2], we also used 35 images from COCO-Stuff [90], dedicated to outdoor scenes. All images were presented in a randomized fashion to each person. In order to maximize the number of participants, we created our online assessment tool for this purpose. In total, 46 persons participated in the survey. Figure 6.8 illustrates that the images reconstructed by our approach are more appealing to the users by a large margin. In terms of number of votes per method, reconstructions by the SROBB got 617 votes, while ESRGAN, SFT-GAN, SRGAN and RCAN methods got 436, 223, 201 and 33 votes, respectively. In addition, the “Cannot decide” choice provided in the survey was chosen 100 times. In terms of the best images by majority of votes, among 35 images, SROBB was a dominant choice in 15 images. These results confirm that our approach reconstructs visually more convincing images compared to mentioned methods for the users. Moreover, unlike SFT-GAN, the proposed approach do not require a segmentation map during the test time, while it takes advantage of semantic information and produces competitive results.

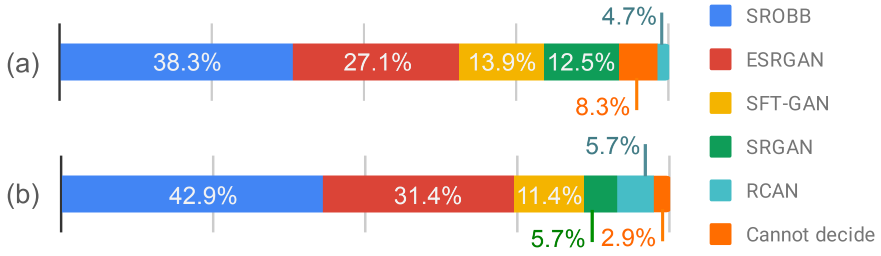


Figure 4.7 – The results of the user study, comparing SROBB (ours) with RCAN [35], SRGAN [19], ESRGAN [43] and SFT-GAN [2] methods. Our method produces visual results that are the preferred choice for the users by a large margin in terms of: (a) percentage of votes, (b) percentage of winning images by majority of votes.

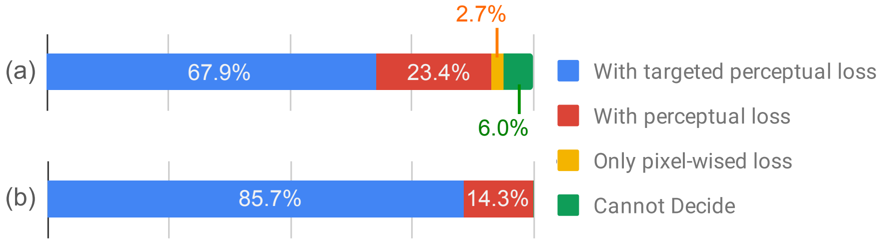


Figure 4.8 – The results of the ablation study showing the effect of the targeted perceptual loss; more convincing results have been obtained by a large margin, in terms of: (a) percentage of votes, (b) percentage of winning images by majority of the votes.

Ablation study

To better investigate the effectiveness of the proposed targeted perceptual loss, we performed a second user study with similar conditions and procedure to the one in the previous section. Specifically, we study the effect of our proposed targeted perceptual loss; we train our decoder with three different objective functions: 1- pixel-wise MSE only; 2- pixel-wise loss and standard perceptual loss similar to [19]; and 3- Pixel-wise loss and our proposed targeted perceptual loss (SROBB). The adversarial loss term is also used for both 2 and 3. In total, 51 persons participated in our ablation study survey. Figure 4.8 shows that users are more convinced when the targeted perceptual loss is used instead of the commonly used perceptual loss. It got 1212 votes, while objective functions 1 and 2 got 49 and 417 votes, respectively. In addition, the “Cannot decide” choice was chosen 107 times. In terms of the best images by majority of votes, among 35 images, third objective function was a dominant choice in 30, while 1 and 2 won only in 5 images. Images reconstructed only by the pixel-wise loss had minority number of votes, however, they got considerable number of votes for images in which the “sky” was the main class. This can be explained by the over-smoothed nature of the clouds, which suits distortion-based metrics.

4.4.4 Inference time

Unlike existing approaches for content-aware SR, our method does not require any semantic information at the input. Therefore, no additional computation is needed at the test time. We reach an inference time of 31.2 frame per second, with a standard XGA output resolution (1024 × 768 in pixels) on a single GeForce GTX 1080 Ti.

4.5 Conclusion

In this chapter, we introduced a novel targeted perceptual loss function for the CNN-based single image super-resolution. The proposed objective function penalizes different regions of an image with the relevant loss terms, meaning that using edges' loss for the edges and textures' loss for textures during the training process. In addition, we introduce our OBB labels, created from pixel-wise segmentation label, to provide a better spatial control of the semantic information for the images. This allows our targeted perceptual loss to focus on the semantic regions of an image. Experimental results verify that training with proposed targeted perceptual loss yields perceptually more pleasing results, and outperforms the state-of-the-art SR methods.

5 Test-Time Adaptation Based on Perceptual Similarity

In Chapters 3 and 4, we studied and showed how exploiting contextual information within images could improve the reconstruction quality of super-resolution methods. In this chapter, we go one step further and show that this information is not only beneficial for learning better image representations during the training process; we demonstrate that it can also be used to find complementary high-resolution references at the test time and benefit from them to generate perceptually more appealing images. In particular, this chapter introduces a new method based on test time adaptation to leverage perceptually similar images to test images to reach higher reconstruction quality.

In the following, this chapter is presented in the form of an article. The research and all experiments were performed by the first author (the author of this thesis). The manuscript was also written by him and was further revised by other authors. The article is submitted to **Conference on Computer Vision and Pattern Recognition (CVPR 2021)** for publication, with the original title of **Test-Time Adaptation for Super-Resolution: You Only Need to Overfit on a Few More Images**. The supporting information of this article can be found in Appendix A.2.

Test-Time Adaptation for Super-Resolution: You Only Need to Overfit on a Few More Images

Authors: Mohammad Saeed Rad[1], Thomas Yu[1], Behzad Bozorgtabar[1], Jean-Philippe Thiran[1].

[1] Signal Processing Laboratory 5, EPFL, Lausanne, Switzerland.

Abstract

Existing reference (RF)-based super-resolution (SR) models try to improve perceptual quality in SR under the assumption of the availability of high-resolution RF images paired with low-resolution (LR) inputs at testing. As the RF images should be similar in terms of content, colors, contrast, etc. to the test image, this hinders the applicability in a real scenario. Other approaches to increase the perceptual quality of images, including perceptual loss and adversarial losses, tend to dramatically decrease fidelity to the ground-truth through significant decreases in PSNR/SSIM. Addressing both issues, we propose a simple yet universal approach to improve the perceptual quality of the HR prediction from a pre-trained SR network on a given LR input by further fine-tuning the SR network on a subset of images from the training dataset with similar patterns of activation as the initial HR prediction, with respect to the filters of a feature extractor. In particular, we show the effects of fine-tuning on these images in terms of the perceptual quality and PSNR/SSIM values. Contrary to perceptually driven approaches, we demonstrate that the fine-tuned network produces a HR prediction with both greater perceptual quality and minimal changes to the PSNR/SSIM with respect to the initial HR prediction. Further, we present novel numerical experiments concerning the filters of SR networks, where we show through filter correlation, that the filters of the fine-tuned network from our method are closer to "ideal" filters, than those of the baseline network or a network fine-tuned on random images.

Keyword: Test-Time Adaptation, Super-Resolution, Overfitting, Fine-Tuning

5.1 Introduction

Super-resolution (SR) is the ill-posed problem of transforming low-resolution (LR) images (I_{LR}) to their high-resolution (HR) counterparts (I_{HR}) [17, 27, 79, 107, 108]. A common way to model the interaction between LR and HR images can be formulated as $I_{LR} = (I_{HR} * \mathbf{k}) \downarrow_s + N$, where $*$ denotes convolution, $\mathbf{k}$ is the blur kernel, $\downarrow_s$ denotes downsampling by a factor s, and N is noise. In this chapter, we focus on a common setting for SR, where the down-sampling kernel is known and is a bicubic downscaling kernel [107].

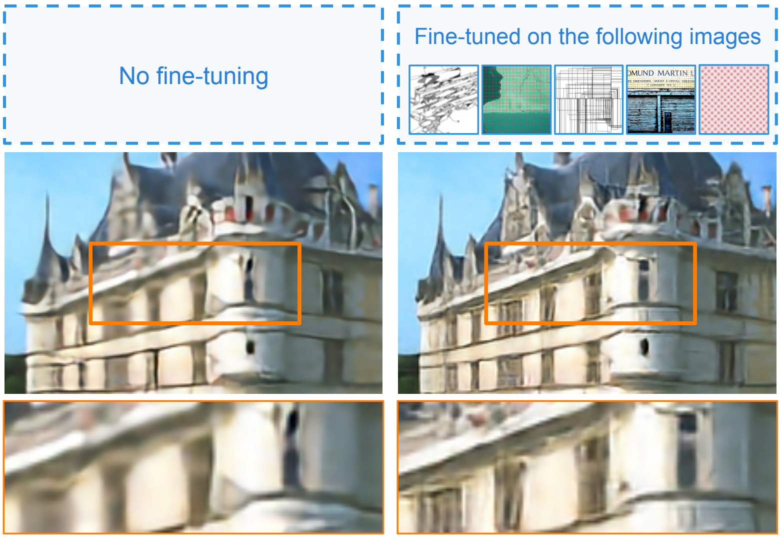


Figure 5.1 – We demonstrate how we can improve the perceptual quality of Super-Resolution images produced by a generic SR network and a given LR image by fine-tuning the network on specific images which activate the same filters of a pre-trained feature extractor as those activated by the initial SR prediction. Left: Initial SR predictions from the baseline network, right: Predictions from the network after fine-tuning for a few iterations on selected images by our method. Zoom in for the best view.

In this setting, deep learning algorithms [35, 43, 109] have made remarkable progress in image super-resolution that aim to obtain a I_{HR} output from one of its I_{LR} versions by leveraging the power of deep convolutional neural networks. Going even further, in the field of RF-based SR, an external high-resolution reference image is provided, where the reference image and I_{HR} share similar textures and qualities [110, 111, 112, 113]. In this way, the networks are trained to leverage additional information from the reference HR image. This has the drawback of assuming the existence of and finding HR images similar to a given LR image -in terms of content, colors, contrast as well as the increased size of the networks trained to incorporate the additional HR input.

In the SR literature, pixel-based metrics, which compare predicted HR images to the ground truth HR image such as the peak signal to noise ratio (PSNR) or structural similarity index (SSIM) are commonly used to judge the performance of SR methods [107]. However, it is known that optimizing neural networks for PSNR, SSIM, or other pixel-based metrics generally result in over-smoothed, perceptually unappealing HR images [19, 80, 114]. In fact, [114] shows that there is a mathematical tradeoff between performance on these pixel-based metrics and perceptual quality. However, we note that in theory, a perfect reconstruction would have the highest performance on both pixel-based metrics and perceptual quality. Strategies to increase perceptual quality include training networks with a perceptual loss [80], which computes the distance between predicted and ground truth images in the feature space using a pre-trained classification network. Generative adversarial networks (GANs) [46] are also used to improve perceptual quality [2, 19, 41, 43, 49]. However, these approaches significantly decrease PSNR,

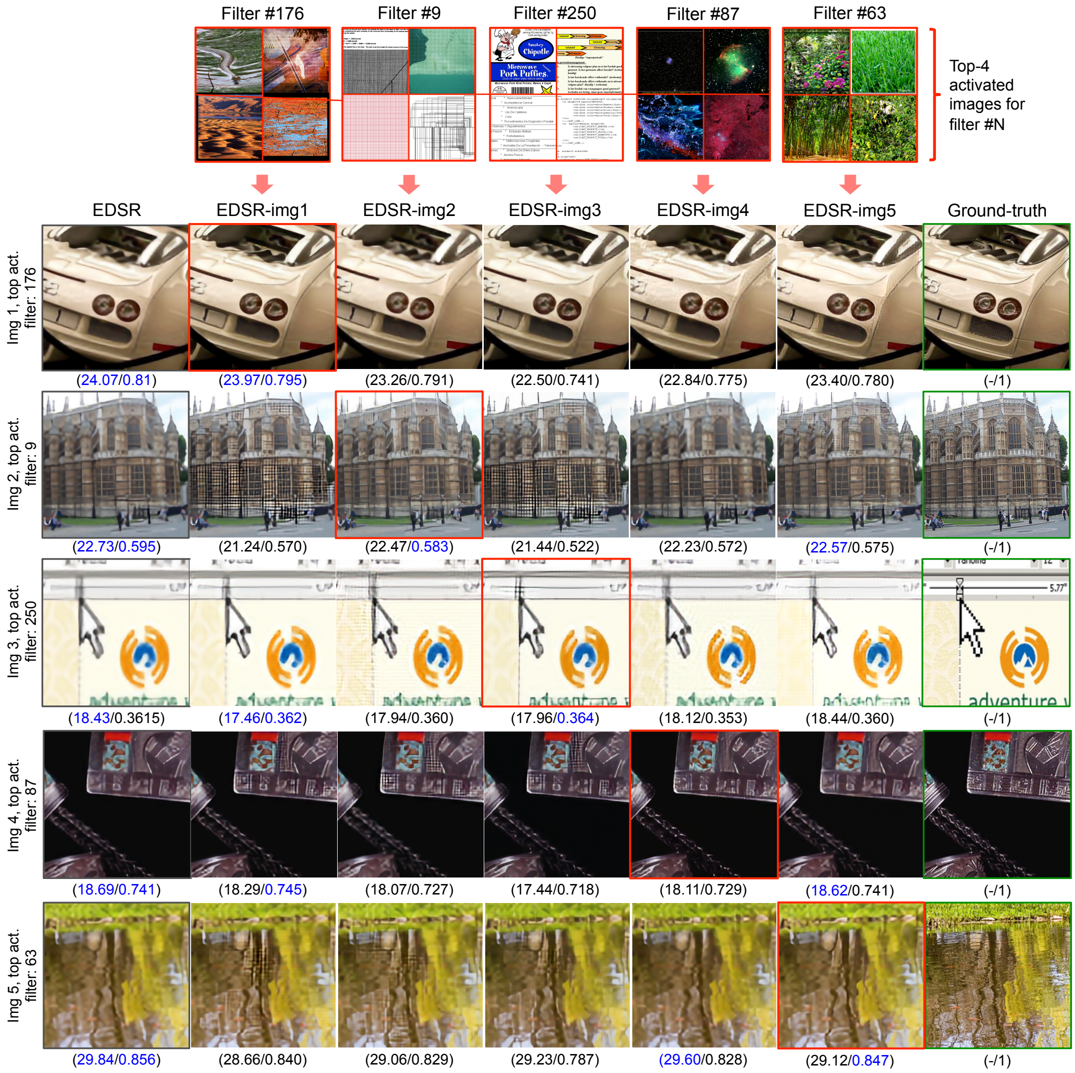


Figure 5.2 – We demonstrate the effect of fine-tuning on images, which maximally activate specific filters in a pre-trained classification network with respect to perceptual quality and PSNR/SSIM values. The first column shows the initial HR predictions from the baseline network while subsequent columns show predictions from the network after fine-tuning on the images bordered by red at the top. Note that in each row, the network fine-tuned on the image set which shares the filter of maximal activation with the initial HR prediction gives the best perceptual quality without affecting the PSNR or SSIM significantly. Fine-tuning on image sets which maximally activate different filters results in oversmoothing or image artifacts as compared to the ground truth. Two best values are in blue. Please zoom in on the screen.

SSIM and other pixel-based metrics with respect to trained networks using only the pixel-wise losses [19, 80, 114].

In this chapter, inspired by RF-based SR and previous analysis of learned filters of classification networks, we propose a novel method to increase the perceptual quality of the output of a generic PSNR-based SR network on a given LR image without significantly affecting the PSNR or SSIM. This is done through test-time adaptation of the generic SR network, to tailor it to a given LR image used for testing. Concretely, given an input LR image and the SR network pre-trained using only pixel-wise losses, e.g., L_1, we first obtain the initial HR prediction from the network. We then fine-tune the network on a few pairs of LR/HR images from the training dataset, where the images are chosen by the similarity of their activations of filters from a pre-trained classification network with respect to the corresponding activations of the initial HR prediction. We show that the perceptual quality of the HR image from the fine-tuned network increases without significantly decreasing the PSNR or SSIM values. Further, we demonstrate that this does not contradict past studies on the trade-off between PSNR and perceptual quality [19, 41], as this results from fine-tuning on images that activate the same filters as the initial LR input. The fine-tuned SR network performs worse on images dissimilar to the LR input; hence, overall performance is in conformity with the trade-off. As shown in Fig. 5.2, our method can improve perceptual quality with minimal impact on PSNR/SSIM with fine-tuning on images with similar activations as the LR input.

Our contributions are as follows:

- We propose a novel, test-time adaptation method to improve SR, which guides PSNR-based SR networks toward perceptually more compelling images by fine-tuning on selected images at the test-time, without significant impact on the PSNR or SSIM.
- To our knowledge, we are the first to investigate how overfitting/fine-tuning on selected images, which differ by what filters in a pre-trained classification network they maximally activate, can change SR reconstructions for better or worse.
- We also show, to our knowledge, novel numerical experiments in the field of SR, where we quantitatively relate the filters of the pre-trained SR network, the fine-tuned network, and an "ideal" SR network (ideal with respect to the given LR input) to show that our method moves the filters of the pre-trained SR network closer to the "ideal" filters.

5.2 Overview of the approach

The overview of the proposed method is shown in Fig. 5.3; the task is to predict an HR image from a given LR input by benefiting from a few more essential images with respect to the pre-trained model. The pipeline can be split into three main steps: First, we construct a reference dataset, namely the Activation dataset, containing essential images for further fine-tuning. Second, we use a novel technique to choose relevant images from the Activation

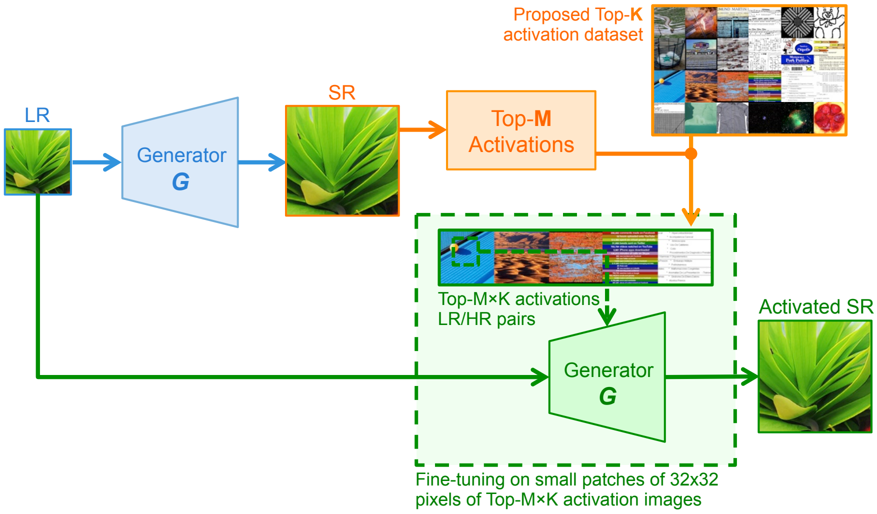


Figure 5.3 – The overview of the proposed method: First, the LR input is passed to the SR network to generate an initial SR prediction. We then find the top M filters of the third layer of the VGG [40] network which are activated by the initial SR prediction. Then, we fine-tune the SR network on a set of $M * K$ images chosen from the training data, which maximally activate the same M filters. Finally, we pass the LR input to the fine-tuned SR network for the final SR prediction. In this example, K = 1 and M = 5.

dataset. Finally, we fine-tune the pre-trained SR network on these images and produce the final reconstruction. In what follows, let $G, \mathscr{D}$ denote the baseline SR network and the dataset of paired LR and HR images used to train the SR network, respectively. We present each step in detail as follows:

Construction of Activation dataset We first construct a reference dataset from the HR images of $\mathscr{D}$ by extracting their corresponding activations from the third layer of the VGG classification network [40]. For each channel in the third layer ($conv3$), we order (descending) the images by the channel's corresponding activation and take the top K images. As there are 256 channels in the third layer, we form a reference dataset of $256 \times K$ HR images. We choose the third layer as the features from this layer have been shown to be more discriminative [115, 116]. As an example, in Fig. 5.4, we show for different filters in different layers of VGG19 [40], the top nine images by filter activation from a subset of 50 thousand images from ImageNet [117]. We further investigate the effectiveness of using other layers ($conv2, 4$, and 5, in Appendix A.2).

Test-Time Adaptation of the SR network We obtain an initial HR prediction from passing LR to G, which we call SR. We pass SR through the third layer of the VGG classification network [40] and note the top M filters with the highest activations. From this list of filters, we can use our reference dataset to define a set of $M \times K$ images where for each of the M filters, we take the top K images in our dataset in terms of activation of the filter. We then fine-tune G on this set of images for a set number of epochs determined by performance on the validation set.

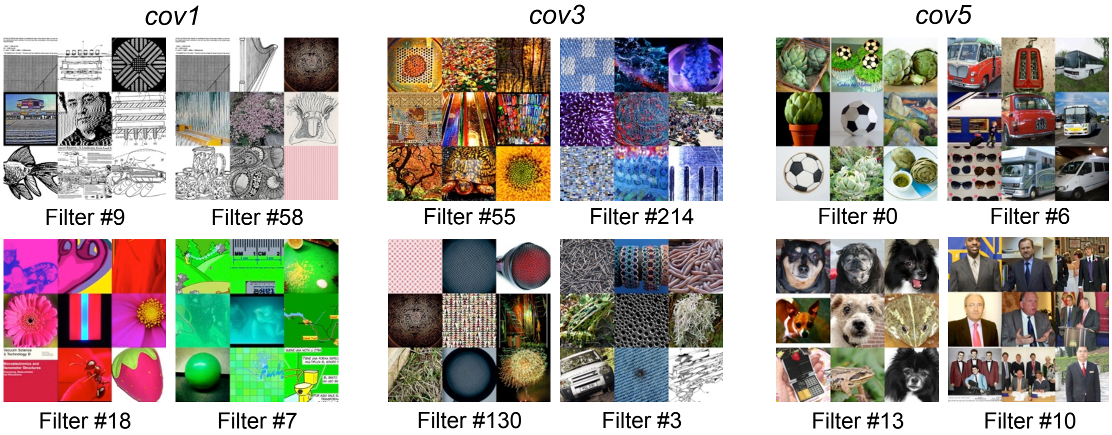


Figure 5.4 – Top 9 activated images from a subset of 50 thousand images from ImageNet [117] for different filters in the conv1, conv3 and conv5 layers of VGG19 [40], respectively.

Prediction After fine-tuning G, we again pass the LR image to G to obtain our final HR prediction, which we call activated SR. The activated SR image is perceptually more convincing than the initial SR, without significant decreases in its PSNR and SSIM values.

5.3 Image activations in SR

In the machine learning/computer vision literature, analysis of the activations of neural networks with respect to different inputs is often used for the purposes of understanding/interpretability [118] and extraction of relevant features for downstream processing, for instance, in unsupervised learning [115, 119]. In terms of SR, only perceptual loss uses this analysis by matching the activations, with respect to a layer of a pre-trained classification network, of the HR prediction and the ground-truth, showing the efficacy and importance of these features. We go further by explicitly analyzing the activations of the third layer of VGG19 [40] with respect to a large dataset of 50 thousand images. Then for each filter, we can assign a group of images with the highest activations. As perceptual loss shows that constructing images based on activations can improve perceptual quality, it stands to reason that fine-tuning a network on images that also triggers specific filters can enhance SR reconstructions on images that have similar activations with respect to those filters. Hence, in contrast to perceptual loss, we are able to exploit the analysis of activations by enhancing the perceptual quality of SR on a given LR input by using a set of images which are visually different from the LR input, but similar in terms of activation. To the best of our knowledge, we are the first to create and benefit from such a dataset for SR. Fig. 5.4 shows a few example images from the Activated dataset; the detailed procedure of generating this dataset is presented in section 5.2.

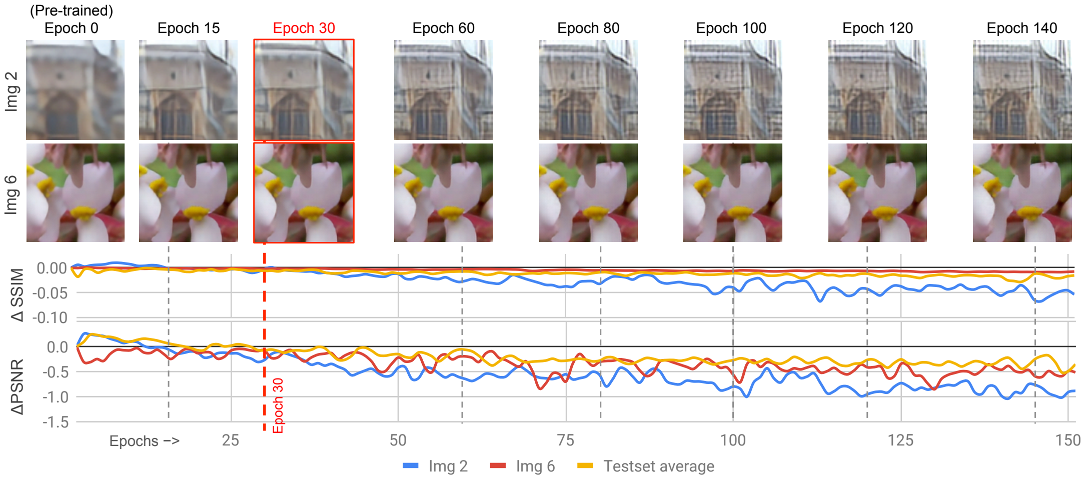


Figure 5.5 – The effects of fine-tuning as a function of the number of epochs. We show the average change of PSNR and SSIM values over the test set, as well as explicit examples of visual, PSNR, and SSIM evolution on two images. We see in image 2 that perceptual quality can dramatically increase with fine-tuning, while image 6 is not affected significantly. Please zoom in on the screen.

5.4 Overfitting: the good, the bad, and the ugly

Throughout this chapter, we have used the word "fine-tuning" for continuing the training of a pre-trained SR network on a small set of images. Implicitly, this assumes that such training has a beneficial effect for the purpose of the network, which is to perform SR on a given LR image (**"The good"**). However, as seen in Fig. 5.2, such fine-tuning could also be labeled as overfitting, since our method only improves reconstructions on images with similar patterns of filter activation as the given LR image; other inputs can result in image artifacts and over smoothing (**"The bad"**). That is, the fine-tuned network no longer generalizes to all image classes. This can be understood in terms of the tradeoff between perceptual quality, and PSNR established in [114]. We conjecture that we are able to gain perceptual quality with minimal changes to PSNR/SSIM precisely because this gain occurs only on images similar in filter activation to those used in the fine-tuning. As both PSNR and perceptual quality can decrease in other images, the overall performance does not contravene the tradeoff. Thus, for a given LR image, overfitting is actually good for improving SR reconstructions. However, we note that the outcome of fine-tuning is dependent on the number of epochs of additional training (**"The ugly"**). Further, while generalization of the network performance is clearly compromised, it is possible for the fine-tuning to have no effect, good or bad, on different classes of images. In Fig. 5.5, we show the effects of fine-tuning on visual quality, PSNR, and SSIM values as a function of the number of epochs as well as how it can dramatically increase the perceptual quality of some images while not affecting others.

It remains to address how overfitting using only a pixel-wise loss can improve perceptual

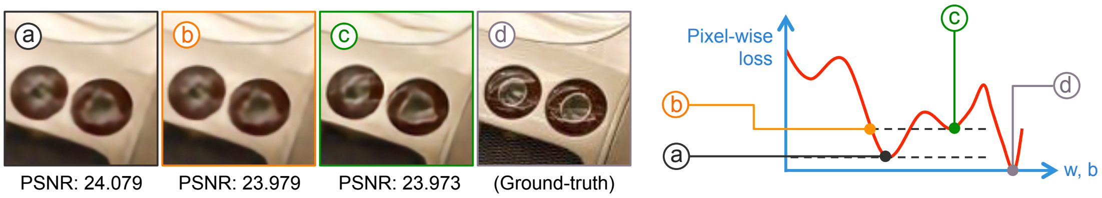


Figure 5.6 – Image (a) is obtained by a pre-trained baseline with pixel-wise optimization on a large dataset. Images (b,c) are obtained during the fine-tuning by our proposed method, reaching almost the same PSNR. Image (d) is the ground truth. We see from comparing images (b) and (c) that our method is guiding the SR network to a different local minimum with a better perceptual quality, as the same loss is achieved but with dramatically different quality.

quality. We emphasize that the fine-tuning is done with only L_1 loss; in contrast to perceptual loss or adversarial losses used to improve perceptual quality, only pixel-wise metrics are used in our approach. In Fig. 5.6, we show a diagram of our hypothesis that overfitting guides the SR network to a local minimum, where the pixel-wise error is only slightly different, while the perceptual quality is dramatically improved. As evidence, note that almost the same PSNR is achieved on image b (during the pretraining of the network, before fine-tuning by our approach) and image c (after fine-tuning), but image c is much sharper and realistic.

5.5 Experiments and results

5.5.1 Experimental settings

Generator architecture

While our method and experiments can generalize to arbitrary SR networks, we use an EDSR [24] as our baseline generator, which we denote as G. EDSR performs better than other conventional residual SR networks by eliminating some unnecessary modules e.g., batch normalization. This makes it a good candidate to investigate the effectiveness of our proposed approach as many other SR networks incorporate components designed for specific contributions/improvements that may not strictly be necessary. The architecture consists of 32 residual blocks and 256 filters per convolutional layer (more details in Appendix A.2). We train this network in a single step for 50 epochs, using the L_1 loss function. For the training data, we use a subset of 50 thousand images taken from Imagenet [117]. The Adam optimizer was used for the optimization. The learning rate was set to $1e{-}3$ and then decayed by a factor of ten every 20 epochs.

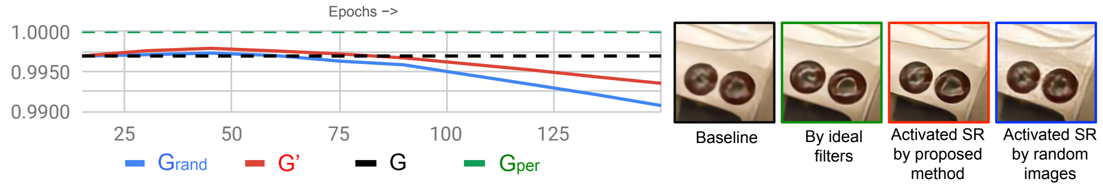


Figure 5.7 – The average correlations over the test-set images of the filters of the final layer of feature extractor of G' and G_{rand} to the filters of G_{per} as a function of the number of epochs of fine-tuning in red/blue respectively, with the correlation of the baseline as a dotted black line. We see that the correlation of G' to G_{per} is higher than G_{rand}; This is consistent with our hypothesis that the proposed method of fine-tuning transforms the filters of the baseline to be closer to the “ideal” filters for a particular image.

Fine-tuning/overfitting

Parameters: In order to force the fine-tuning to make changes to the filters of the network’ feature extractor rather than changing the last layers of the network, we freeze the convolutional layers related to up-sampling, more specifically, the filters coming after the pixel-shuffle layers. The images for fine-tuning are the random crops of 32×32 pixels from our constructed dataset. We choose a relatively low learning rate of $1e-4$ for gradual change.

K and M: We conduct sensitivity analysis to choose the best values for the number of images per filter K and the number of filter M used for our test image. We tune these parameters based on the perceptual quality of the generated images. The results of this work are produced by setting the values of K and M to two and five, respectively (10 images in total). a more detailed study can be found in Appendix A.2.

Stoppage condition: The criteria to stop the fine-tuning was basically defined based on qualitative comparison of reconstructed images at different epochs where we could see at epoch 30, the vast majority of the images from our validation set were perceptually more convincing as compared to other epochs. However, considering Fig. 5.6, we can see this choice can also be justified as this epoch also coincides with the beginning of a significant drop in SSIM and PSNR values over all images on the test set.

Test-set

For our test-set, we randomly chose 100 images from the ImageNet dataset (non-overlapping between activation and training datasets), as both our baseline network and the Activation dataset are trained on/using a subset of 50,000 ImageNet images. As it is shown [120, 121] that SR network quality drops when doing cross-dataset tests, therefore, we focus on showing a proof of concept of improving a generic SR network on a generic dataset and do not add an additional variable of different datasets to the mix.

5.5.2 Filter selection analysis

In the following, we provide, to our knowledge, novel experiments and investigations into SR networks, where we examine, at the level of the network' filters, how the SR network changes in response to our selective overfitting. For our experiments, we draw on [17], where authors found that two networks trained from scratch for the same task can have different filter orders and different filter patterns; however, fine-tuning a network to perform a different, but related task preserved the filter orders and patterns of the original network. They further show that the changes in filters by doing fine-tuning are gradual, by proposing to quantitatively assess the similarities between the filters of two different instances of the same network through correlation; concretely, given filter F_i, F_j,

$$\rho_{ij} = \frac{(F_i - \overline{F_i})(F_j - \overline{F_j})}{\sqrt{\|F_i - \overline{F_i}\|_2}\sqrt{\|F_j - \overline{F_j}\|_2}} \tag{5.1}$$

where ρ_{ij} is the correlation index. We use this correlation index to quantitatively study the changes in the filters of the SR network after fine-tuning. Given an LR image with HR ground truth, let G_{per} denote the EDSR baseline which is fine-tuned on solely this LR image to produce a perfect reconstruction. We can, in some sense, assume that G_{per} possesses the ideal or optimal set of filters for super-resolving this LR image, as we overfit it on this image; further, we verified that, consistent with [17], the overall structure/filter orders are preserved from the baseline network, indicating that G_{per} is not simply memorizing the image within its parameters.

Let G' denote the fine-tuned network produced from our method on this LR image. Let G_{rand} denote the EDSR network fine-tuned on a set of random images. In Fig 5.7, we show the average correlations of the filters of the final layer of G' and G_{rand} to the filters of G_{per} as a function of the number of epochs of fine-tuning. The average was computed by constructing G', G_{rand}, G_{per} for each image in the test set, then taking the average correlation over the images. We also show the correlation of the filters of the baseline G with G_{per}. We see that the correlation of G' to G_{per} is generally higher than those of G_{rand} and G, including at 30 epochs, which is the number that we use for our method. This provides evidence that our method of fine-tuning in some sense brings the baseline closer to the "ideal" set of filters for a given LR image.

5.5.3 Comparison to PSNR-based approaches

From the qualitative results in Fig. 5.2, we can observe that when we fine-tune the pre-trained EDSR network using the images chosen through our method, namely activated-SR approach, the perceptual quality increases with minimal impact on the PSNR/SSIM. This minimal impact on the PSNR/SSIM has been also shown in Fig. 5.5, where we can see that over a test set of 100 images, the mean changes in PSNR/SSIM are minimal.

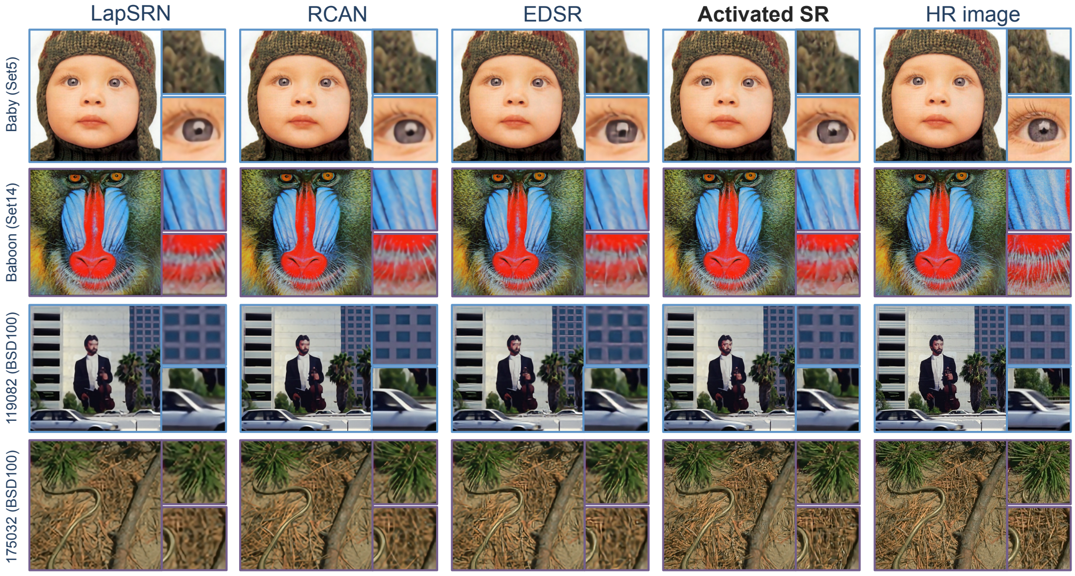


Figure 5.8 – Qualitative comparison to PSNR-based approaches. From left to right: Bicubic, LapSRN [28], RCAN [35], EDSR [24], Activated-SR (ours), and HR image, tested on images from Set 5 [37], Set14 [38] and BSD100 [58] testsets. We emphasize that our method is EDSR using our test-time adapation method. We show results from other networks for comparison. Zoom in for the best view.

In Fig. 5.8, we additionally compare our method to LapSRN [28], RCAN [35] and EDSR [24] methods and by using test images from Set5 [37], Set14 [38] and BSD100 [58] standard datasets. For a fair comparison, in this section, we only considered PSNR-based approaches as our methods still relies only on minimizing the pixel-wise distance of the SR and ground-truth images and does not benefit from any perceptual losses. This figures shows that activated-SR images produced by out method have superior perceptual quality, while Table 5.1 confirms that this increases had a minimal impact on the PSNR/SSIM over the whole test set.

5.5.4 Comparison to perceptual-based approaches

Finally, in Fig 5.9, we provide a comparison between SR network trained using our proposed method and using perceptual losses (pixel-wise loss + vgg loss + adversarial loss, with the same setting and discriminator as described in ESRGAN [43] work). We note that the perceptual loss adds more sharpness than that of our method, but can also provide highly distorted textures. In all cases, the images from our method are sharper/more detailed than those of the EDSR baseline, without distorting the texture. This can be explained by the fact that optimizing SR networks with only perceptual loss sometimes leads to the incitement of high frequency details in image e.g., sharp edges, entailing over-sharpened images. Therefore, they do not conform with the distortion based metrics.

On average, the decrease in PSNR and SSIM using perceptual loss is 628 and 355 percent

Dataset	Metric	LapSRN	RCAN	EDSR	**Ours**
Set5	SSIM	0.887	**0.918**	0.893	0.891
	PSNR	31.56	**32.61**	32.41	32.40
Set14	SSIM	0.772	0.773	0.774	**0.776**
	PSNR	28.20	**28.86**	28.81	28.70
BSD100	SSIM	0.742	0.815	0.802	**0.819**
	PSNR	27.41	**29.32**	29.24	29.15

Table 5.1 – Comparison LapSRN [28], RCAN [35], EDSR [24], and activated-SR (ours) on various test sets. We emphasize that our method is EDSR using our test-time adapation method. We show the results from other methods for comparison. Considering Fig. 5.8 the proposed method improves the perceptual quality of EDSR with minimal impact on the PSNR/SSIM.

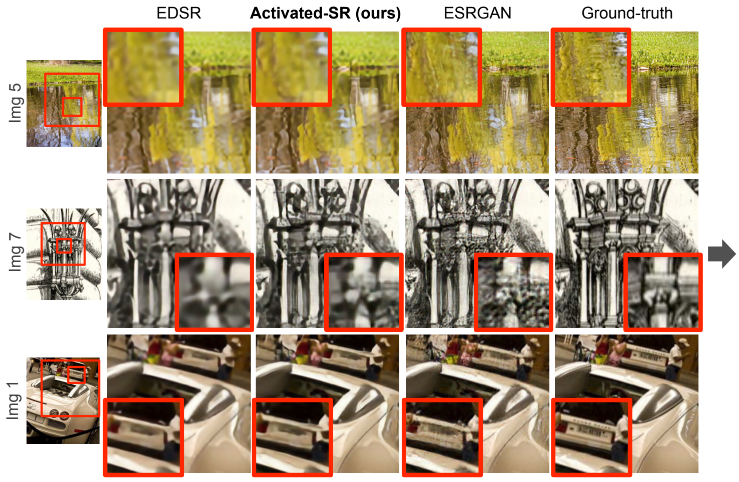
larger, respectively, than the corresponding decreases using our method. Hence, our method provides images with much greater fidelity to the ground truth, while increasing the perceptual quality without distorted textures.

5.5.5 Inference time

We note that as our method fine-tunes the baseline network for every test image, this is computationally more expensive than simply using the baseline network. However, we note that relatively small patches of 32 × 32 pixels, and a small number of images (10 in our case) used for fine-tuning still keeps the computation time practical for single image SR tasks; the additional fine-tuning takes ~13 seconds by using a GeForce GTX 1080Ti GPU, which results in a total time of ~14 seconds for a 2560 × 1920 pixel output.

5.6 Conclusion

In this chapter, we propose a novel approach to improve the perceptual quality of PSNR-based SR methods. In our approach, given a pre-trained SR network and LR input, we use test-time adaptation by fine-tuning the SR network on a subset of images from the training dataset with similar activation patterns as the initial HR prediction, with respect to the filters of a feature extractor. We show that the fine-tuned network produces an HR prediction with both greater perceptual quality and minimal changes to the PSNR/SSIM, in contrast to perceptually driven approaches. Further, in contrast to reference-based SR, we use only images from our proposed activation dataset for fine-tuning, eliminating the issue with the availability of HR reference images close to the input image. Finally, through numerical experiments novel to the field of SR, we show that our fine-tuning can be interpreted as within the test-time adaptation paradigm, where we update the model parameters to be closer to the parameters of an "ideal" SR network, which is overfitted on the given LR input.



	Ours	ESRGAN
Img 5	-0.72/ -0.009	-1.26/ -0.045
Img 7	-0.06/ -0.004	-0.21/ -0.078
Img 1	-0.10/ -0.015	-0.82/ -0.035
Avg. of testset	-0.21/ -0.009	-1.32/ -0.032

ΔPSNR/ΔSSIM

Figure 5.9 – Comparing the proposed method and a perceptual-based approach [43]. In general, the perceptual loss provides sharper edges but also more distorted textures, wheres the proposed method provides images which are sharper and contain more details than the baseline without distortion. In the table, we show that this is reflected in the decrease in the PSNR/SSIM; using perceptual loss decreases the PSNR/SSIM relative to the baseline far more than using our method.

6 Integrating into Real-World SR

In previous chapters, we studied and proposed various methods to benefit from contextual information within images to improve the reconstruction quality of learning-based super-resolution methods. This chapter addresses the challenges of “real-world” super-resolution, where the downsampling kernel is not bicubic and consists of a large variety of natural image degradations. In particular, we focus on a generic solution to make all state of the art SR works trained on synthetic datasets, including our context-aware SR contributions, compatible with the real-world super-resolution setting.

In the following, this chapter is presented in the form of an article. The research and experiments were mainly performed by the first author (the author of this thesis). The manuscript was also written by him as the leading author and was further revised by other authors. The article was presented at the WACV 2021 conference and published in the **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2021**, *pp. 1590-1599, DOI: 10.1109/WACV48630.2021.00163* with the original title of **Benefiting from Bicubically Down-Sampled Images for Learning Real-World Image Super-Resolution**. The supporting information of this article can be found in Appendix A.3.

Benefiting from Bicubically Down-Sampled Images for Learning Real-World Image Super-Resolution

Authors: Mohammad Saeed Rad[1*], Thomas Yu[1*], Claudiu Musat[2], Hazım Kemal Ekenel[1, 3], Behzad Bozorgtabar[1], Jean-Philippe Thiran[1].

[1] Signal Processing Laboratory 5, EPFL, Lausanne, Switzerland.
[2] AI Lab, Swisscom AG, Lausanne, Switzerland.
[3] Istanbul Technical University, Istanbul, Turkey.
[*] Equal contributions.

Abstract

Super-resolution (SR) has traditionally been based on pairs of high-resolution images (HR) and their low-resolution (LR) counterparts obtained artificially with bicubic downsampling. However, in real-world SR, there is a large variety of realistic image degradations and analytically modeling these realistic degradations can prove quite difficult. In this work, we propose to handle real-world SR by splitting this ill-posed problem into two comparatively more well-posed steps. First, we train a network to transform real LR images to the space of bicubically downsampled images in a supervised manner, by using both real LR/HR pairs and synthetic pairs. Second, we take a generic SR network trained on bicubically downsampled images to super-resolve the transformed LR image. The first step of the pipeline addresses the problem by registering the large variety of degraded images to a common, well understood space of images. The second step then leverages the already impressive performance of SR on bicubically downsampled images, sidestepping the issues of end-to-end training on datasets with many different image degradations. We demonstrate the effectiveness of our proposed method by comparing it to recent methods in real-world SR and show that our proposed approach outperforms the state-of-the-art works in terms of both qualitative and quantitative results, as well as results of an extensive user study conducted on several real image datasets.

Keyword: Real-world Super-Resolution, Generative Adversarial Networks, Deep Learning

6.1 Introduction

Super resolution is the generally, ill-posed problem of reconstructing high-resolution (HR) images from their low-resolution (LR) counterparts. Generally SR methods restrict themselves to super-resolving LR images downsampled by a simple and uniform degradations (i.e,

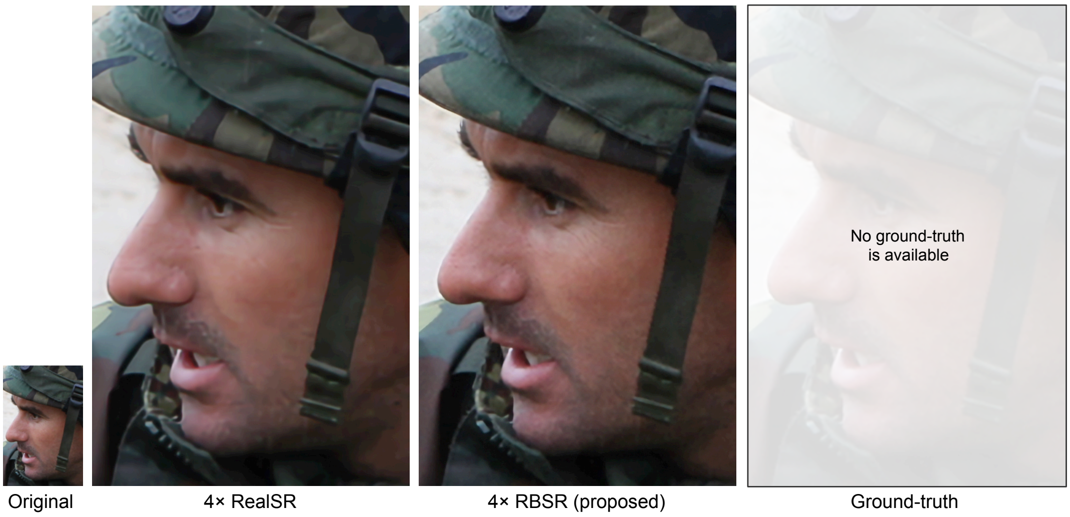


Figure 6.1 – An example SR produced by our system on a real-world LR image, for which no higher resolution/ground-truth is available. Our method is compared against the RealSR [63] method, a state-of-the-art of real SR method trained in a supervised way on real low-resolution and high-resolution pairs. The low-resolution image is taken from HR images in the DIV2K validation set [60].

bicubic downsampling) [17, 27, 79, 107, 108]. Although the performance of these methods on artificially downsampled images are quite impressive [43, 109], applying these methods on real-world SR images, with unknown degradations from cameras, cell-phones, etc. often leads to poor results [63, 122]. The real-world SR problem is then to super-resolve LR images downsampled by unknown, realistic image degradations [123].

Recent works try to resemble realistic degradations by acquisition instead of artificial downsampling, such as hardware binning, where LR corresponds to a coarser grid of photoreceptors [124], or camera focal length changes, which changes the apparent size of an object in frame [63]. These approaches could propose very limited number of physically real low and high-resolution pairs and their degradation models are limited to very few acquisition hardwares.

As shown in [125], correct modeling of the image degradation is crucial for accurate super-resolution. A general, analytical model for image degradation which is commonly assumed is $\mathbf{y} = (\mathbf{x} * \mathbf{k}) \downarrow_s + N$, where $\mathbf{y}$ is the LR image, $\mathbf{x}$ is the HR image, $*$ denotes convolution, $\mathbf{k}$ is the blur kernel, N is noise, and $\downarrow_s$ denotes downsampling by a factor s. However, as can be seen in Figure 6.2, these convolutional models are only approximations to the true, real degradations.

Recently, there has been a push to account for more realistic image degradations through physical generation of datasets with real LR to HR pairs [63], synthetically generating real LR to HR pairs through unsupervised learning or blind kernel estimation [122, 126], and simulating more complex image degradation models such as in equation 1, with and without restrictions

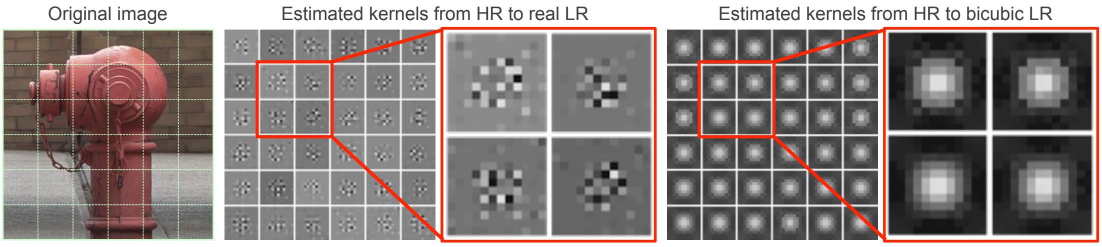

Original image
Estimated kernels from HR to real LR
Estimated kernels from HR to bicubic LR

Figure 6.2 – Downsampling kernels estimated patchwise on a RealSR [63] LR image and the same image bicubically downsampled from the HR image. Estimations were done using least squares optimization with regularization on the kernel using the LR and HR images, assuming the standard degradation model of kernel convolution followed by subsampling. We can see that the RealSR LR images are difficult to estimate with the standard image degradation model.

on **k** and $\downarrow_s$ [127, 128]. The pipelines of these approaches generally have the ultimate goal of training an end-to-end network to take as input a “real" image and output a SR image. Although these approaches result in better reconstruction quality, the real challenge of the real-world LR to HR problem is not only limited to a lack of real LR and HR pairs; the large variety of degraded images and the difficulty in accurately modeling the degradations makes realistic SR even more ill-posed than SR based on bicubically down-sampled images [129].

Main idea We propose to address real world SR with a two-step approach, which we call Real Bicubic Super-Resolution (RBSR). RBSR generally decomposes the difficult problem of real world SR into two, sequential subproblems: **1-** Transformation of the wide variety of real LR images to a single, tractable LR space. **2-** Use of generic, bicubic SR networks with the transformed LR image as input.

We choose to transform real LR images to the common space of bicubically downsampled images because of two main advantages. First, bicubic images are tractably generated with the standard convolutional model of image degradation, therefore the inverse transform is less ill-posed comparing to the cases of arbitrary/unknown degradations. Second, we can leverage the already impressive performance of SR networks trained on bicubically downsampled images, thanks to the availability of huge SR image datasets using bicubic kernels (see Figure 6.1).

In summary, our contributions are as follows:

1. We use a GAN to train a CNN-based image-to-image translation network, which we call a “bicubic look-alike generator”, to map the distribution of real LR images to the easily modeled and well understood distribution of bicubically downsampled LR images. We use a SR network with the transformed LR image by our proposed bicubic look-alike generator as input to solve the real-world super-resolution problem.

2. To this end, and for the consistency of the bicubic look-alike generator, we propose a novel copying mechanism, where the network is fed with identical, bicubically downsampled images as both input and ground-truth during training; this way, the network loses its tendency to merely sharpen the input images, as realistic low-resolution images usually seem to be much smoother.
3. We train our bicubic look-alike generator by using an extended version of perceptual loss, where its feature extractor is specifically trained for SR task and on bicubically downsampled images. The proposed "bicubic perceptual loss" is shown to have less artifacts.
4. We demonstrate the effectiveness of the proposed two-step approach by comparing it to an end-to-end setup, trained in the same setting. Furthermore, we show that our proposed approach outperforms the state-of-the-art works in terms of both qualitative and quantitative results, as well as results of an extensive user study conducted on several real image datasets.

In essence, training models on paired datasets of real LR and HR pairs requires expensive collection of big datasets; in addition, training a single model on multiple degradations for SR is ill-posed/vulnerable to instability [129]. Training on synthetic datasets coming from analytical degradation models have the benefit of much larger datasets and an easier task for the network, at the cost of being less realistic. However, this approach still has the ill-posedness problem of training on multiple degradations. In RBSR, we try to simultaneously keep the added information from realistic LR images and the impressive performance of SR networks on single, well-defined degradations.

6.2 Related work

The vast majority of prior work for Single image super-resolution (SISR) focuses on super-resolving low-resolution images which are artificially generated by bicubic or Gaussian down-sampling as the degradation model. We consider that recent research on addressing real-world conditions can be broadly categorized into two groups. The first group proposes to physically generate new, real LR and HR pairs and/or learn from real LR images in supervised and unsupervised ways (Section 6.2.1). The second group extends the standard bicubic downsampling model, usually by more complex blur kernels, and generates new, synthetic LR and HR pairs (Section 6.2.2).

6.2.1 Real-World SR through real data

Some recent works [63, 130] propose to capture real LR/HR image pairs to train SR models under realistic settings. However, the amount of such data is limited. The authors in [63, 130] proposed to generate real, low-resolution images by taking two pictures of the same scene, with camera parameters all kept the same, except for a changing camera focal length. Hence, the image degradation corresponds to "zooming" out of a scene. They generate a dataset of real LR and HR pairs according to this procedure and show that bicubically trained SR models perform poorly on super-resolving their dataset. Since this model's image degradation can be modeled as convolution with a spatially varying kernel, they propose to use a kernel prediction network to super-resolve images. In [122], the authors perform unsupervised learning to train a generative adversarial network (GAN) to map bicubically downsampled images to the space of real LR images with two unpaired datasets of bicubically downsampled images and real LR images. They then train a second, supervised network to super-resolve real LR images, using the transformed bicubically downsampled images as the training data. In a similar work, [72] trains a GAN on face datasets, for the specific face SR task, but their approach relies on unrealistic blur-kernels.

In [131], the authors model image degradation as convolution over the whole image with a single kernel, followed by downsampling. Given a LR image, they propose a method to estimate the kernel used to downsample the image solely from subpatches of the image by leveraging the self-similarity present in natural images. This is done by training a GAN, where the generator produces the kernel and the discriminator is trained to distinguish between crops of the original image and crops which are downsampled from original image using this estimated kernel. This method relies on the accuracy of the standard convolutional model of downsampling, which is shown to not hold for RealSR images in Figure 6.2. Further, the estimation of the kernel and subsequent SR are quite time consuming in comparison to supervised learning based methods; the calculation of the kernel alone for a 1000×1000 image can take more than three minutes on a GTX 1080 TI. In addition, their method constrains the size of the input images to be "large enough" since they need to downsample the input images during training. In [45], the authors propose an unsupervised cycle-in-cycle GAN, where they create one module for converting real LR images to denoised, deblurred LR images and one module for SR using these Clean LR images. They then tune these networks simultaneously in an end-to-end fashion, which causes this intermediate representation of the LR image to deviate from their initial objective.

6.2.2 Real World SR through extended models

In [128], the authors extend the bicubic degradation model by modeling image degradation as a convolution with an arbitrary blur kernel, followed by bicubic downsampling. They embed the super-resolution in an alternating iterative scheme where analytical deblurring is alternated with applying a SR network trained on bicubically downsampled images. Although this method generalizes to arbitrary kernels, one has to provide the kernel and the number of iterations as an input to the pipeline. In [127], the authors extend the bicubic degradation model by modeling image degradation as a convolution with a Gaussian blur kernel, followed by bicubic downsampling. They use an iterative scheme using only neural networks, where at each iteration the pipeline produces both the SR image and an estimate of the corresponding

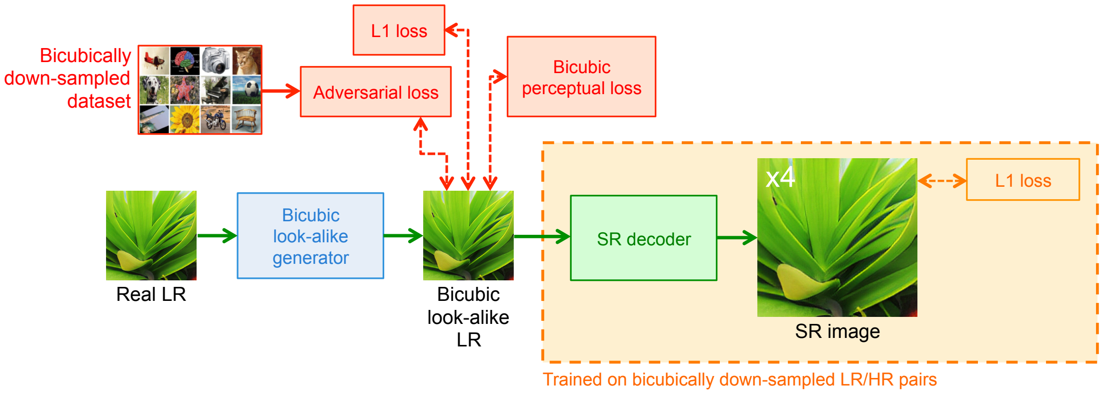

L1 loss
Bicubically down-sampled dataset
Adversarial loss
Bicubic perceptual loss
Real LR
Bicubic look-alike generator
Bicubic look-alike LR
SR decoder
x4
SR image
L1 loss
Trained on bicubically down-sampled LR/HR pairs

Figure 6.3 – We propose a two-step pipeline for real world SR. First, we transform real LR images to bicubically downsampled looking images through our bicubic look-alike generator. We then pass the transformed image as input to a generic SR decoder trained on bicubically downsampled images.

downsampling kernel. In [126], the authors also model image degradations as convolution with a blur kernel followed by bicubic downsampling. They estimate the blur kernel using a pre-existing blind deblurring method on a set of "real" images which are bicubically upsampled; they use the same dataset of low quality cell-phone pictures used in [122]. They then train a GAN to generate new, realistic blur kernels using the blindly estimated blur kernels. Finally, they generate a large synthetic dataset using these kernels and train an end-to-end network on this dataset to perform SR. These three methods all rely on an analytical model for image degradation as well as being reliant on restrictive kernels or blind kernel estimation.

6.3 Methodology

6.3.1 Overall pipeline

RBSR consists of two steps; first, we use a Convolutional Neural Network (CNN)-based network, namely the bicubic look-alike image generator, whose objective is to take as input the real LR image and transform it into an image of the same size and content, but which looks as if it had been downsampled bicubically rather than with a realistic degradation. We call this output the bicubic look-alike image. Second, we use any generic SR network trained on bicubically downsampled data to take as input the transformed LR image and output the SR image. Figure 6.3 shows an overview of our proposed pipeline. We restrict the upsampling factor to four. In the following subsections, we describe each component of our pipeline in more details.

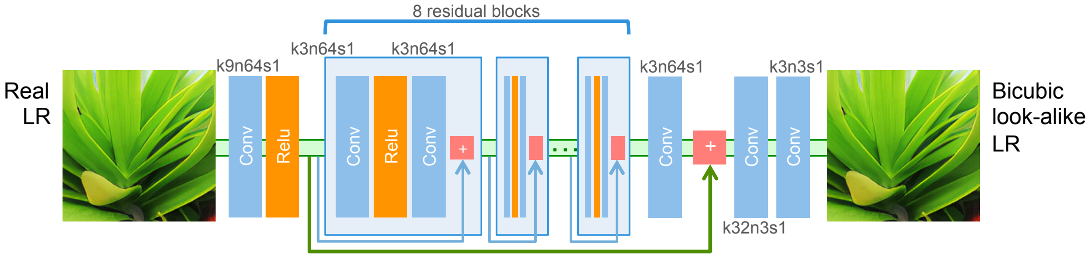


Figure 6.4 – Schematic diagram of the bicubic-alike decoder. We train the decoder using our new bicubic perceptual loss, alongside standard L_1 and adversarial losses. In this schema, k, n and s correspond to kernel size, number of feature maps and stride size, respectively.

6.3.2 Bicubic look-alike image generator

The bicubic look-alike image generator is a CNN, trained in a supervised manner. The main objective of this network is to transform real LR images to bicubic look-alike images. In this section, we present its architecture in detail. Then, we introduce a novel perceptual loss used to train it. Finally, we also introduce a novel copying mechanism used during training to make this transformation consistent.

Architecture

The architecture of the bicubic look-alike generator is shown in Figure 6.4. The generator is a feed-forward CNN, consisting of convolutional layers and several residual blocks, which has shown great capability in image-to-image translation tasks [132]. The real low-resolution image $I^{Real-LR}$ is passed through the first convolutional layer with a ReLU activation function with a 64 channel output. This output is subsequently passed through 8 residual blocks. Each block has two convolutional layers with 3×3 filters and 64 channel feature maps. Each one is followed by a ReLU activation. By using a long skip connection, the output of the final residual block is concatenated with the features of the first convolutional layer. Finally, the result is filtered by a last convolution layer to get the the 3-channel bicubic look-alike image ($I^{Bicubic-LR}$).

Loss functions

In the bicubic look-alike generator, we use a loss function ($\mathscr{L}_{total}$) composed of three terms: 1- Pixel-wise loss ($\mathscr{L}_{pix.wise}$), 2- adversarial loss, and 3- our novel bicubic perceptual loss function ($\mathscr{L}_{bic.perc.}$). The overall loss function is given by:

$$\mathscr{L}_{total} = \alpha \mathscr{L}_{pix.wise} + \beta \mathscr{L}_{bic.perc.} + \gamma \mathscr{L}_{adv} \quad , \tag{6.1}$$

where α, β and γ are the corresponding weights of each loss term used to train our network.

In the following, we present each term in detail:

• **Pixel-wise loss.** We use the L_1 norm of the difference between predicted and ground-truth images as this has been shown to improve results compared to the L_2 loss [36].

• **Adversarial loss.** This loss measures how well the image generator can fool a separate discriminator network, which originally was proposed to reconstruct more realistic looking images for different image generation tasks [19, 46]. However, in this work, as we are feeding the discriminator with bicubically downsampled images as the "real data", it results in images which are indistinguishable from bicubically downsampled images. The discriminator network used to calculate the adversarial loss is similar to the one presented in [19]; it consists of a series of convolutional layers with the number of channels of the feature maps of each successive layer increasing by a factor of two from that of the previous layer, up to 512 feature maps. The result is then passed through two dense layers, and finally, by a sigmoid activation function. The discriminator classifies the images as either "bicubically downsampled image" (real) or "generated image"(fake).

• **Bicubic perceptual loss.** Perceptual loss functions [19, 80] tackle the problem of blurred textures caused by optimization of using per-pixel loss functions and generally result in more photo-realistic reconstructions. In this work, we take inspiration from this idea of perceptual similarity by introducing a novel perceptual loss.

However, instead of using a pre-trained classification network, e.g. VGG [40] for the high-level feature representation, we use a pre-trained SR network trained on bicubically down-sampled LR/HR pairs. In particular, we use the output of the last residual block of our SR network, presented in Section 6.3.3, to map both HR and SR images into a feature space and calculate their distances. The bicubic perceptual loss term is formulated as:

$$\begin{aligned}\mathscr{L}_{bic.\_perc.} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \Big(\phi_k^{SR}\left(I^{Bicubic-LR}\right) \\ -\phi_k^{SR}\left(I^{T-LR}\right)\Big)^2,\end{aligned} \tag{6.2}$$

where $W_{i,j}$ and $H_{i,j}$ denote the dimensions of the respective feature maps. ϕ_k^{SR} indicates the output feature map of the k-th residual block from the SR decoder and I^{T-LR} denotes the transformed LR image. We conjecture that using a SR feature extractor, which is specifically trained for SR task and on bicubically down-sampled images, will better reflect features corresponding to the characteristics of bicubically downsampled images than using a feature extractor trained for image classification.

In Figure 6.5, we compare the effect of using the standard perceptual loss which uses a pre-trained classification network versus our bicubic perceptual loss. Note that the standard

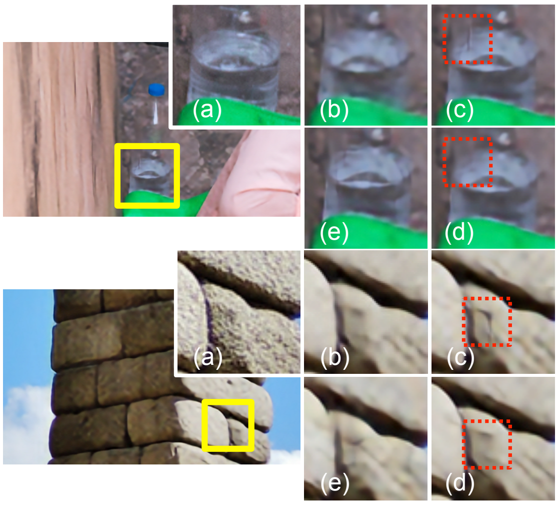


Figure 6.5 – The effectiveness of using bicubic perceptual loss: (a) HR image, (b) Only L1 loss, (c) perceptual loss, (d) bicubic perceptual loss, and (e) bicubic perceptual loss + adversarial loss. Red boxes show how using bicubic perceptual loss (c) decreases artifacts comparing to using conventional perceptual losses (d), while still producing sharper edges comparing to only using L_1 loss.

perceptual loss introduces artifacts in the transformed LR image which are avoided by the bicubic perceptual loss. Further, we see that using the bicubic perceptual loss produces sharper edges as compared to using just the L_1 loss.

Copying mechanism

Bicubically downsampled images are in general seem to be much sharper than realistic low-resolution images, therefore, training network by real LR images gives it this tendency to merely sharpen the input images instead of learning bicubic characteristics. To address this issue, we want the network to be consistent and apply minimal sharpening to already sharp images. To this end, we utilize a novel copying mechanism, where the network is periodically fed with identical, bicubically downsampled images as both input and output during training. This is done in order to prevent the network from just learning to sharpen images, as this can cause oversharpening or amplification of artifacts.

In Figure 6.6 we compare the outputs of the network trained with and without the copying mechanism. We can see clearly that training without the copying mechanism results in severe over-sharpening of the output image.

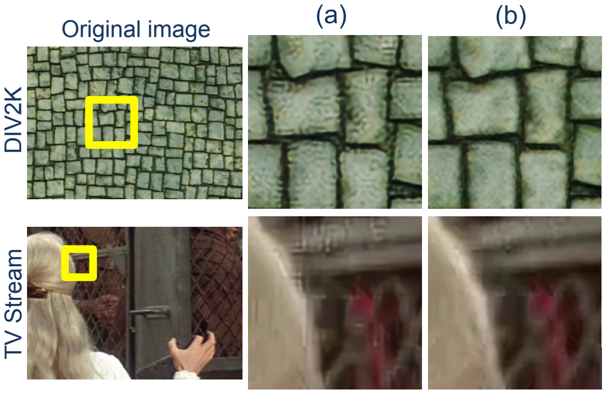


Figure 6.6 – Example images generated without (a) and with (b) the copying mechanism during training. We can clearly see that without the copying mechanism, resulting images suffer from oversharpening and artifact amplification.

6.3.3 SR generator

The second step of our pipeline is to feed the output of our bicubic-like image generator as the input to any SR network trained on bicubically downsampled images. For simplicity, we use a network based on EDSR [24]. The EDSR architecture is composed of a series of residual blocks bookended by convolutional layers. Crucially, batch normalization layers are removed from these blocks for computational efficiency and artifact reduction. For simplicity, as well as decreasing training/inference time, we only use 16 residual blocks, as compared to the 32 residual blocks used in EDSR. This generator is trained on DIV2K training images (track 1: bicubically downsampled images and HR pairs) and by using the L_1 loss function. We refer the reader to Appendix A.3 for more details about the network architecture.

6.3.4 Training parameters

Bicubic look-alike generator For the training data, as input, we use 400 RealSR [63] and 400 DIV2K Track 2 [60] LR images. The RealSR dataset contains real LR-HR pairs, captured by adjusting the focal length of a camera and taking pictures from the same scene. Track 2 images are downsampled using unknown kernels. As the desired output is the bicubic look-alike image, we use the bicubically downsampled RealSR and the bicubically downsampled DIV2K (track 1) images as the ground truth for the training inputs. In addition, as described in Section 6.3.2, we add 400 bicubically downsampled images from DIV2K, identical for both input and ground-truth, to make the generator consistent and avoid oversharpening or artifact amplification. We use the same 400 bicubically downsampled images from DIV2K as the real input of the discriminator. At each epoch, we randomly cropped the training images into 128 × 128 patches. The mini-batch size in all the experiments was set to 16. The training was done in two steps; first, the SR decoder was pre-trained for 1000 epochs with only the L_1 pixel-wise loss function. Then the proposed bicubic perceptual loss function, as well as the adversarial loss, were added and the training continued for 3000 more epochs. The weights of the L_1 loss, bicubic perceptual loss and adversarial loss function (α, β and γ) were set to 1.0, 3.0, and 1.0 respectively. The Adam optimizer [96] was used during both steps. The learning rate was set to 1×10^{-4} and then decayed by a factor of 10 every 800 epochs. We also alternately optimized the discriminator with similar parameters to those proposed by [19].

SR generator The SR decoder is also trained in a single step for 4000 epochs and using the L_1 loss function. For the training data, we only use track 1 images of DIV2K, which consists of 800 pairs of bicubically downsampled LR and HR images. Similar to the training of the bicubic look-alike generator, the Adam optimizer was used for the optimization process. The learning rate was set to 1×10^{-3} and then decayed by a factor of 10 every 1000 epochs.

End-to-end baseline To investigate the effectiveness of RBSR, which super-resolves a given input in two steps, we also fine-tune the EDSR architecture with the same datasets used to train the bicubic look-alike generator. This dataset consists of 400 RealSR and 400 DIV2K Track 2 LR and HR pairs. We further noticed that the inclusion of 400 bicubically downsampled LR and HR pairs in this dataset adds more robustness to the performance. In order to keep the same number of parameters as in the RBSR pipeline, we increase the number of residual blocks of this end-to-end generator to 24. The training parameters used for this baseline is similar to the ones used in [24].

6.4 Experimental results

In this section, we compare RBSR to several SOTA algorithms (CVPR 2019, ICCV 2019) in real-world SR both qualitatively and quantitatively. We show standard distortion metrics for the datasets with ground truth, and we show a comprehensive user study conducted over six image datasets with varying image quality and degradations. In all cases, we use an upsampling factor of four.

We emphasize that the distortion metrics are not directly correlated to the perceptual quality as judged by human raters [2, 19, 41, 68]; the super-resolved images could have higher errors in terms of the PSNR and SSIM metrics, but still generate more appealing images. Moreover, the RealSR images represent only a limited group of realistic images from Nikon and Canon cameras. Therefore, we validate the effectiveness of our approach by qualitative comparisons and by an extensive user study in the following sections.

6.4.1 Test images

Lack of ground-truth in real-world SR

One of the main challenges of real-world SR is the lack of real low and high resolutions pairs, for both training and testing. As mentioned previously, most of the known benchmarks in super-resolution had no choice but using a known kernel to create a counterpart with lower

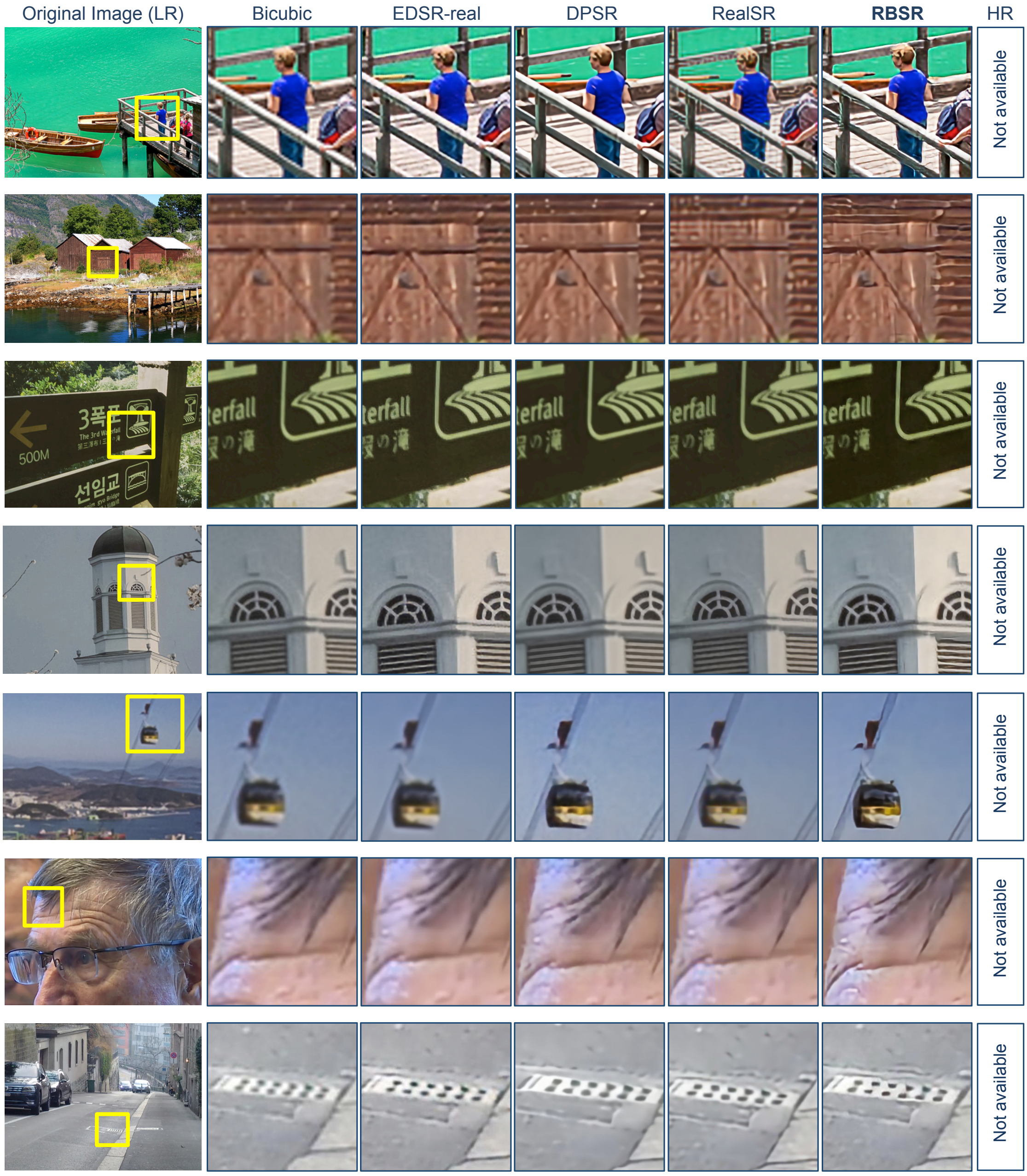


Figure 6.7 – Qualitative results of ×4 SR on a subset of the DIV2k [60] (Rows 1-2), RealSR HR [63] (Rows 3-4), TV Streams (Row 5), and DPED cell-phone images [62] (Row 6). Results from left to right: bicubic, EDSR [24] fine-tuned with real LR and HR pairs, DPSR [128], RealSR [63], and RBSR (ours). Please note that no ground-truth is available for these images. Zoom in for the best view.

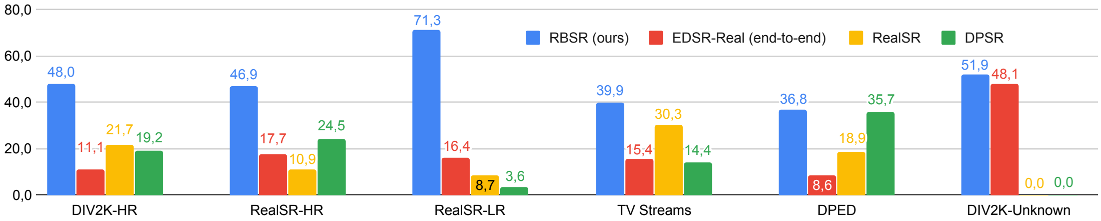


Figure 6.8 – Results of the user study comprising forty one people, comparing EDSR [24], fine-tuned with real LR and HR pairs, DPSR [128], RealSR [63], and RBSR (ours), on six different datasets: DIV2K HR [60], RealSR [63] HR, RealSR LR, TV Stream images, DPED [62] Mobile Phone images, and DIV2K Unknown Kernel LR.

resolution. To the best of our knowledge RealSR [63] is the only dataset with real images of the same scenes with different resolutions: their LR and HR images are generated by taking two camera pictures of the same scene, but changing the focal length of the camera between the two pictures. Hence, both are real images, but with the RealSR LR being degraded with the degradation from changing the focal length of the camera (zooming out). DIV2K Unknown kernel LR images [60] is another attempt to create pairs of real low and high-resolutions images. They generate synthetically real low and high resolution images by using unknown/random degradation operators.

Images without ground-truth

In addition to RealSR LR and DIV2K Unknown kernel datasets, we also evaluate our method on four datasets of real images, without having any ground-truth as it is the main focus of real-world SR task: 1- RealSR [63] HR test images, 2- DIV2K HR [60] validation images (real), 3- DPED [62] Mobile Phone images, 4- TV Stream images (unknown, depending on the original content of the TV). The DPED Mobile Phone dataset is a dataset of real images where cell-phones were used to take pictures of same scenes. The TV stream images are decoded images from an actual TV channel stream at HD (1920 × 1080) resolution; our acquisition algorithm captured one image every ten minutes over a period of two days, to ensure that our these test images cover different types of content. We note that no information is available about their type of degradations, as the original resolutions of the contents before streaming are unknown. Further, we note that we only have the ground-truth high-resolution images for the DIV2K Unknown Kernels images and the RealSR LR images.

6.4.2 Quantitative results

In this work, calculating distortion metrics such as PSNR and SSIM is not possible for test images that truly reflect the real-world problem (original images from smartphones, TV streams, etc.), as in real cases the downsampling operator is not known and therefore no ground-truth is available. RealSR [63] is the only dataset with physically produced high and low-resolution image pairs. Readers can refer to **Appendix A.3** to find PSNR, SSIM and perception index (PI) metric evaluated by using this dataset.

6.4.3 Qualitative comparison

For the qualitative comparison, we compare the following real world SR algorithms: 1- RBSR (Ours), 2- EDSR-real: the EDSR [24] network trained end-to-end on the same data/settings as RBSR, 3- The pretrained RealSR network [63], and 4- The pre-trained DPSR network with default settings for real-world SR [128]. We compare with the end-to-end EDSR network in order to show the efficacy of splitting the problem into two steps. We compare to RealSR and DPSR as they are two of the most recent state-of-the-art algorithms. We use their pre-trained models along with the default settings for real images they provide[1,2]. In Figure 6.7, we show qualitative results on a random subset of the image datasets described in the previous sections.

6.4.4 User study

We also conducted a user study comprising forty one people in order to gauge the perceptual image quality of SR images using the image datasets described in the previous section. We chose five images randomly from each dataset, with thirty total images. For each image, the users were shown four SR versions of the image, each corresponding to the real-world SR algorithms being compared. Users were asked to select which SR image felt more realistic and appealing. The images were shown to users in a randomized manner. As the datasets reflect a wide range of image quality, etc., we show the evaluations of the algorithms for each dataset separately. Our metric of evaluation for the algorithms is the percent of votes won. We show the results of the user study in Figure 6.8. We find that RBSR won the largest percent of votes over all six image datasets individually. RBSR decisively won the largest percentage of votes, by a margin of 10 to 55% from the second ranked algorithm, on the DIV2K HR, the RealSR-HR, the RealSR-LR, and the TV stream image datasets. The second place algorithm on these datasets alternated from RealSR, DPSR, and EDSR-Real, and RealSR respectively. We note that on the RealSR-LR dataset, for which the RealSR algorithm is tailored and trained, RBSR and EDSR-Real are the first and second place. This shows the efficacy of both the two step approach of RBSR and introducing bicubically downsampled images into the training dataset. On the DPED dataset, RBSR won by a small margin over DPSR.

6.5 Conclusion

In this work, we have shown that the challenges of super resolution on realistic images can be partly alleviated by decomposing the SR pipeline into two sub-problems. First, is the conver-

---

[1] https://github.com/csjcai/RealSR

[2] https://github.com/cszn/DPSR

sion of real LR images to bicubic look-alike images using our novel copying mechanism and bicubic perceptual loss. Second, is the super-resolution of bicubically downsampled images. Each sub-problem addresses a different aspect of the real-world SR problem. Converting real low-resolution images to bicubic look-alike images allows us to handle and model the variety of realistic image degradations. The super-resolution of bicubically downsampled images allows for the application of state-of-the-art super-resolution models, which have achieved impressive results on images with well defined degradations. We show that our approach (RBSR) outperforms the SOTA in real-world SR both qualitatively and quantitatively using a comprehensive user study over a variety of real image datasets.

7 Conclusion

7.1 Thesis summary

In this thesis, we studied and developed several CNN-based methods for the SR task with the main focus of benefiting from the context of images to improve SR reconstruction quality. We proposed innovative solutions that address the majority of the current context-aware SR works limitations. Majority of our contributions are suited for real-time applications, and can run on moderate computational resources.

We first presented some previous works in Chapter 2 and reviewed relevant SR architectures, seminal deep-learning techniques, SR image datasets, and available evaluation metrics for SR that we used in our work. In Chapter 3, we introduced a novel approach to use categorical information while doing SR, without any additional cost at the test time. We developed a generator that only benefited from one shared deep network to learn simultaneously image SR and semantic segmentation by keeping two task-specific output layers during training. This chapter also introduced a novel boundary mask to discard unrelated segmentation losses caused by imprecise segmentation labels. This chapter's contributions have been validated by perceptual experiments, including a user study on images from COCO-Stuff [90] dataset.

To ensure a meaningful spatial control over the training of CNN-based approaches, in Chapter 4, we introduced a novel targeted perceptual loss function for SR task. This loss function was designed to penalize different regions based on their categorical meaning, e.i., using edges' loss for the edges and textures' loss for textures. To making this spatial control possible, we introduced the new OBB (Object, boundary, and background) labels created from pixel-wise segmentation labels and injected additional semantic information into the training process. Our extensive evaluations, including a user study, showed that training with proposed targeted perceptual loss yields perceptually more pleasing results than four other state-of-the-art works.

In this thesis, we further investigated how overfitting/fine-tuning on some selected images can be beneficial for the SR task in Chapter 5. For the first time, we proposed a test-time adaptation technique to improve SR methods' perceptual quality. Given a pre-trained SR network and a low-resolution input, we proposed fine-tuning/overfitting the SR network on a subset of images from the training dataset with similar activation patterns as the initial HR prediction, with respect to the filters of a pre-trained feature extractor. We demonstrated that the fine-tuned network produces perceptually more appealing predictions with minimal changes to the PSNR and SSIM metrics (in contrast to perceptually driven approaches). We further validated this hypothesis by a novel numerical experiment, where we quantitatively judged the learned parameters of the fine-tuned network by comparing them to what we introduced as "ideal" filters. Unlike reference-based SR, we used only images from our proposed activation dataset for fine-tuning, eliminating the issue with the availability of high-resolution reference images close to the input image.

In all our previously mentioned contributions, we used synthetic datasets based on the hypothesis that downsampling kernel is uniform and known, i.e., bicubic downsampling kernel. To address the problem of unknown blur and downsampling kernels in real scenarios, namely real-world SR problem, in Chapter 6, proposed a generic solution to adapt all SR works trained on synthetic datasets to the real-world SR setting. We decomposed the SR task on realistic images into two sub-problems: First, converting the real LR images to bicubic look-alike images using our CNN-based image-to-image translator, trained in a GAN setting. Second, super-resolving images by any SR network trained on bicubically downsampled images. By converting real low-resolution images to bicubic look-alike images, we could handle and model various realistic image degradations. Moreover, this approach enabled re-using state-of-the-art SR models, which have achieved impressive results on images with well-defined degradations. We showed that this two-stage approach outperforms recent real-world SR methods, both qualitatively and quantitatively, using a comprehensive user study over various real image datasets.

In the end, we should emphasize that while context-aware SR methods usually require prior information such as an additional segmentation map at the input, our proposed methods mostly require minimal information only for the training stage and not at the test time. This fact proved our contributions to be practical for real-case scenarios in the sense that they can significantly improve the reconstruction quality without requiring more computational cost, compared to conventional CNN-based approaches. Furthermore, our techniques are mostly not only limited to the SR problem, but they are more general and can be applied to any image generation tasks, such as image inpainting, face generation, etc.

7.2 Limitations

This research demonstrated that a learning-based method could benefit from categorical and contextual information within images to improve its reconstruction quality. However, extracting this information, even for humans, is not easy to obtain in some images. We can encounter images, e.g., some abstract arts, that their context or even the type of available shapes and objects within them are not easily recognizable. In these cases, in case of a wrong recognition by the SR network, it can lead to a biased reconstruction toward a wrong category. This limitations could mostly affect the approach presented in Chapter 3, where specific classes of objects and backgrounds were used in a multitask learning-based setting. In general, the validations in this work were mostly done on random images of different datasets and categories. Further investigations would be needed to study and evaluate the advantages and disadvantages of context-aware methods on these specific images, where a depiction of visual reality is not recognizable.

In this work, we also addressed the problem of the unknown blur kernel of real images, namely the real-world SR problem; the essence of this problem is only shown by doing SR on images that have not HR counterparts. Therefore, this approach was mostly validated through a user study on images for which no ground-truth where available. This user study confirmed that our approach could construct more appealing images for users compared to other state-of-the-art approaches; however, we have no information about how this reconstruction is close to the real scene and a hypothetical ground-truth. In other words, without having a ground-truth, voters may find a ‘tree’ reconstructed by our method more realistic. However, it is not proven that it is more closer to the actual tree in the real scene, comparing to reconstructions of other methods. This limitation can become more significant in medical applications, where the actual truth is indeed more important than what is called more ’realistic’ for observers.

7.3 Future work

In this thesis, we developed several approaches for SISR benefiting from contextual information. In this section, we briefly describe some interesting and promising research directions, which are worth investigating further based on the findings of this work.

New evaluation metrics: Despite recent advances in SR and achieving perceptually appealing results, having a reliable and efficient evaluation metric still remains the biggest challenge of this domain. As mentioned earlier, conventional image quality metrics such as SSIM and PSNR -used commonly in many SR works as quantitative measurements, or even more recent learning-based metrics such as LPIPS and NIQE, are not correlated to the actual perceptual quality perceived by humans. Currently, the only way to reflect superior reconstruction quality in a trustworthy way is through the mean opinion score or user studies. In Chapter 4, we proved that penalizing our optimization process by considering categorical information leads to higher reconstruction quality. From this result, we can also conclude that this new objective function is reflecting the reconstruction quality better than other metrics previously used as objective functions. This idea of creating a targeted metric, evaluating each area of image based on its categorical information, would be definitely one of the future directions of this work. To this end, a thorough study would be needed to find the correlation between the output of this function and scores from extensive user studies.

Towards multi-frame super-resolution: In this thesis, we showed that we are able to recover more realistic images by benefiting from semantic information for the SISR task. The ideas presented in Chapters 3 and 4 were only applied for single image SR, as we could benefit from available datasets containing a considerable number of segmentation labels, e.g., [90] and trained them in a supervised manner. As a future work, we aim to design new frameworks to benefit from the same ideas for video sequences, where no available segmentation database exists. We expect to improve the SR decoder's performance by using semantic information within video frames in addition to multi-frame information.

New objectives for multitask learning-based approaches: As we also emphasized previously, multitask learning improves generalization by using the domain information contained in the training signals of related tasks. This improvement is the result of learning tasks in parallel while using a shared representation. In Chapter 3, we proved the effectiveness of multitask learning for single image SR by learning an SR model simultaneously for single image SR and semantic segmentation. As a future work, we aim at investigating the potential of using multitask learning to improve video SR, specifically with more related and suitable tasks for multi-frame SR, e.g., estimating motions in image sequences; simultaneously learning the best optical flow representation relating two consecutive frames at time t and $t-1$ and video SR.

Higher scale factors: In all our contributions, the majority of experiments focused on solving the SR problem of a scale factor four (×4 SR). The main reason behind this choice was the feasibility of comparing it to other state-of-the-art SR methods, where this scale factor is the dominant choice. We believe that the categorical priors and different ideas presented in this work would be even more effective and significant when performing SR for higher scale factors, such as ×8, ×16 or ×16, where objects and textures become even more difficult to recognize. We emphasize that in such extreme SR, recovering fine details and sharp edges become much more challenging. Furthermore, the idea of proposing a framework capable of handling arbitrary scale factors could be interesting for real-life applications where the upscaling factors are unknown.

A Appendix

A.1 Related to Chapter 4

In this secton, you find the supporting information of the article **"SROBB: Targeted Perceptual Loss for Single Image Super-Resolution"**, presented in Chapter 4. In particular, first, we provide additional qualitative and quantitative results on super-resolution benchmarks such as Set5 [37], Set14 [38] and BSD100 [58]. Then, we present more details of our extensive user study and the time span taken by the users for decision making.

A.1.1 Results on standard benchmarks

Quantitative results

In this subsection, we conduct an evaluation study based on the quantitative metrics. Table A.3 summarizes the average of SSIM, PSNR and LPIPS values of the Set5 and Set14 images, respectively. Because of the fact that the human eye is most sensitive to luma information, we compute the PSNR and SSIM values only for the intensity (luma) channel in YCbCr space.

As also emphasized in the paper, these metrics would not reflect the reconstruction quality; the reconstructed images using both our method and the SRGAN are not ranked first in terms of mentioned metrics, however, they generate more realistic and appealing super-resolved images comparing to the other methods. Therefore, here, we only present the qualitative results on the BSD100 test set.

Qualitative results

In this part, we evaluate and compare the visual results of our method with SRGAN and bicubic interpolation methods on random images from the BSD100 test set, as well as the images from the Set5 and Set14 datasets, respectively. Figure A.1 corresponds to the reconstructed images from the BSD100 dataset. Results on Set5 are shown in Figure A.2 while Figure A.3 shows some

Testset	Metric	Bicubic	SRCNN	SelfExSR	LapSRN	SRGAN	SROBB	HR image
Set5	SSIM	0.811	0.863	0.862	0.884	0.848	0.817	1.0
	PSNR	28.43	30.51	30.34	31.54	29.41	28.93	∞
	LPIPS	0.340	0.214	0.171	0.121	0.083	0.087	0.0
Set14	SSIM	0.704	0.756	0.757	0.772	0.739	0.678	1.0
	PSNR	26.01	27.52	27.41	28.19	26.04	25.43	∞
	LPIPS	0.440	0.332	0.301	0.312	0.148	0.162	0.0

Table A.1 – Comparison of bicubic interpolation, SRCNN [11], SelfExSR [133], LapSRN [28], SRGAN [19] and SROBB (ours) on the Set5 and Set14 test sets. Red color indicates the best measures (SSIM, PSNR [dB], LPIPS) and blue color indicates the second bests. The visual comparison of the images from these test sets are shown in Figures A.2 and A.3.

reconstructed images from the Set14. The upscaling factor of all images is set to four (Best viewed in zoom in).

A.1.2 Details of the user study

Figure A.8 shows a screenshot of the survey that we used to evaluate our proposed method. The subjects were shown five reconstructed images and were asked to choose the image that looks more appealing to them. We also added the real high-resolution image in the same page as the reference. We cropped each image vertically to be able to fit all versions of the same image side by side within a single page. The height of the images are remained the same as the original size.

Time span analysis for the user decision making In total, 51 persons participated in our ablation study. Among them, eight persons have been subject to a new experimental setting, under an additional controlled situation: we recorded the time span that each user spent to respond each question. As each user has different speed to complete the survey, we normalized all times to the average time by all users, 10:52 minutes (in average, 18.62 seconds per question). Table A.2 shows the time that users spend to choose each of the following options: 1- the reconstructed image only by pixel-wise loss, 2- pixel-wise loss and standard perceptual loss, 3- pixel-wise loss and targeted perceptual loss (this work), and finally, 4- the "Cannot decide" option (The adversarial loss term is used for both 2 and 3). For images, where our method was the preferred choice, the average time span taken by the users to make decision was relatively shorter than other methods. We can conclude that, in cases where SROBB was not the winning choice, the difference between the super-resolved images using different loss terms was less significant, therefore, users had more difficulties to choose the best option. Meanwhile, users seem to be more sure when they are voting in favor of reconstructed images by the SROBB method. As a future work, to be able to validate this conclusion and to be sure that time

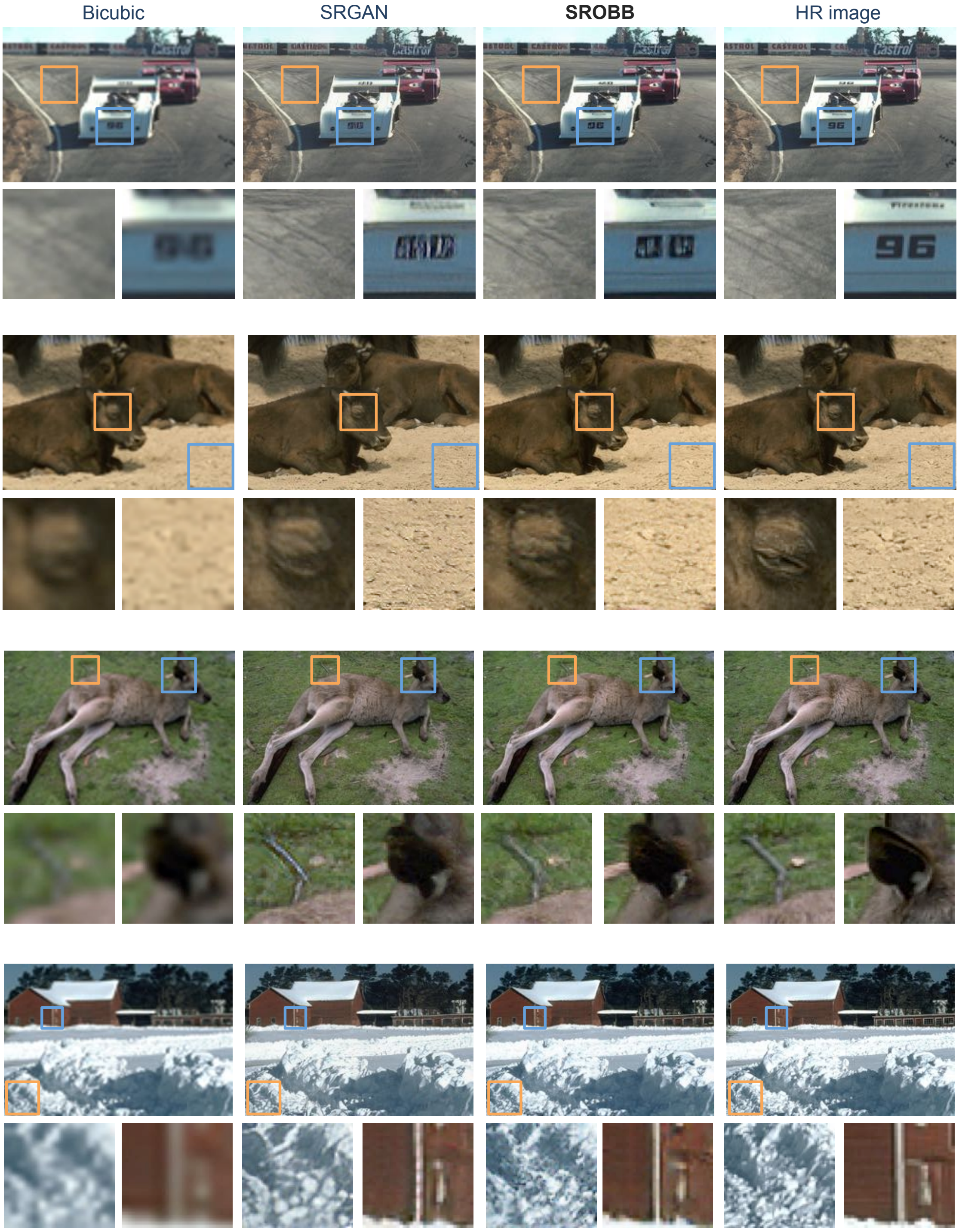


Figure A.1 – Qualitative results on random images from BSD100 [58] using bicubic interpolation, SRGAN[19], SROBB (ours), respectively. Zoom in for the best view. [4× upscaling]

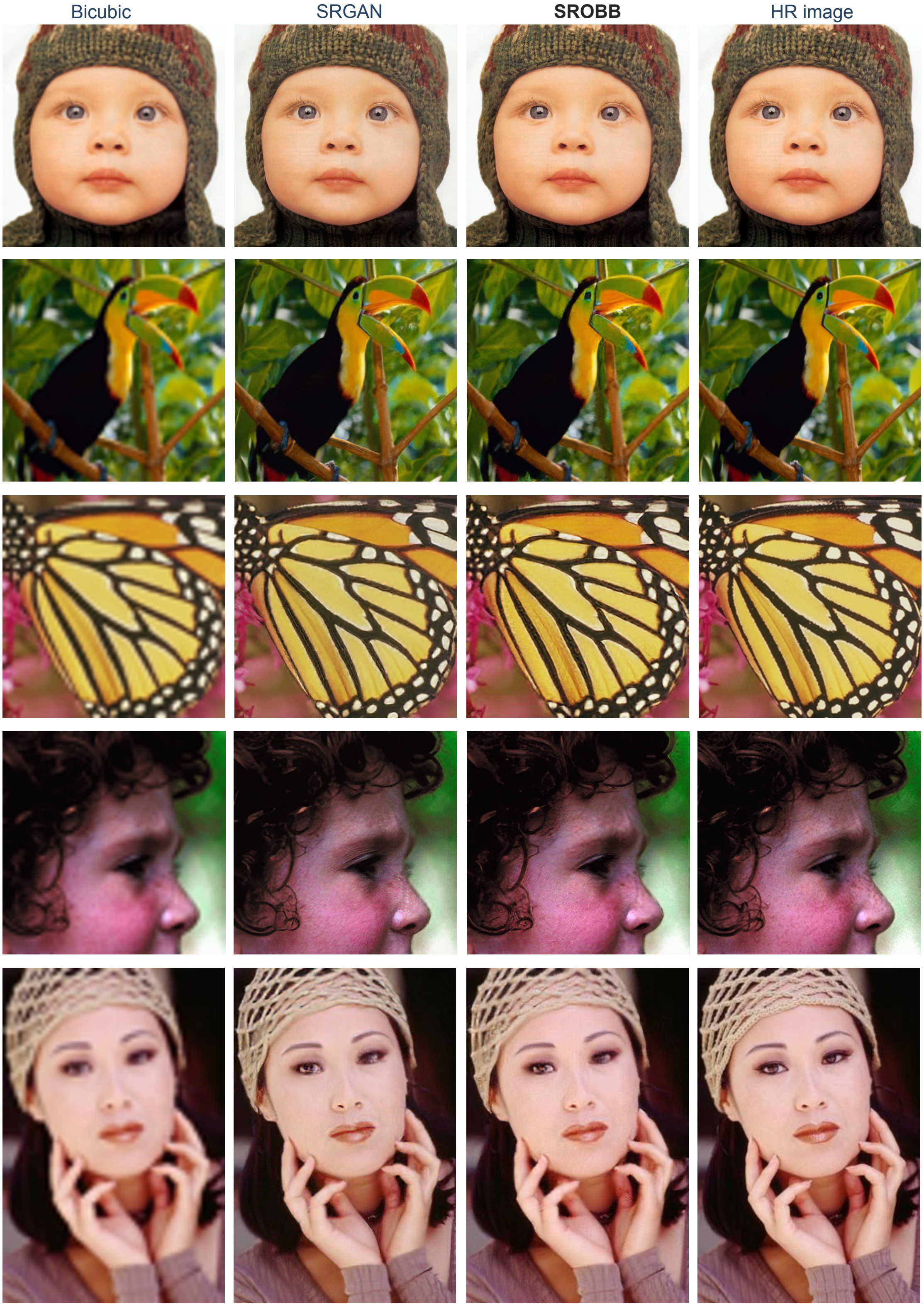


Figure A.2 – Qualitative results on the images from Set5 [37] using bicubic interpolation, SRGAN[19], SROBB (ours), respectively. Zoom in for the best view. [4× upscaling]

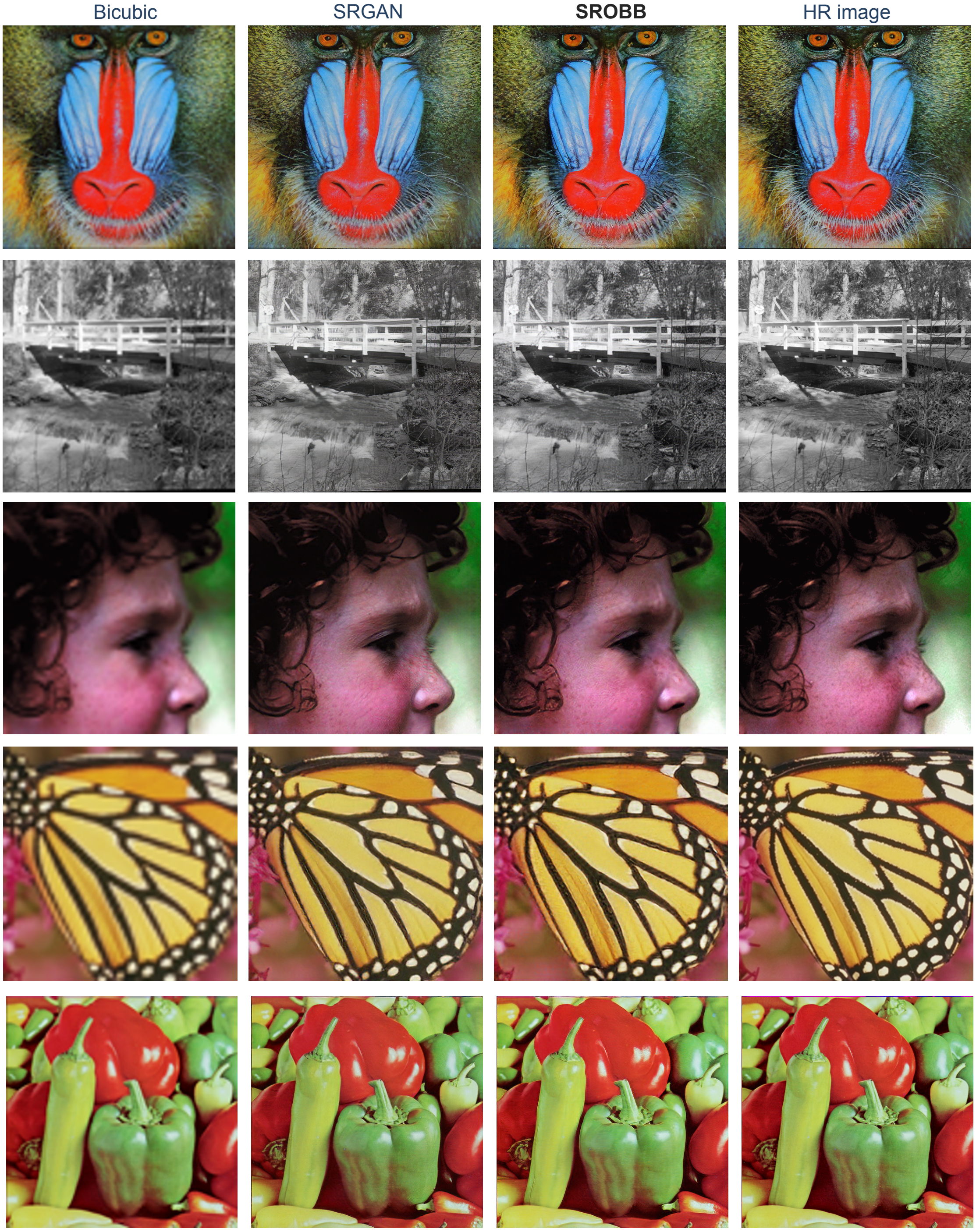


Figure A.3 – Qualitative results on the images from Set14 [38], using bicubic interpolation, SRGAN[19], SROBB (ours), respectively. Zoom in for the best view. [4× upscaling]

Options	Cannot decide	Only pixel -wise loss	With perc- eptual loss	With targeted perceptual loss
Average time	22.60	23.85	24.09	17.81

Table A.2 – The average of decision making duration [seconds] for users to choose the reconstructed images of each method.

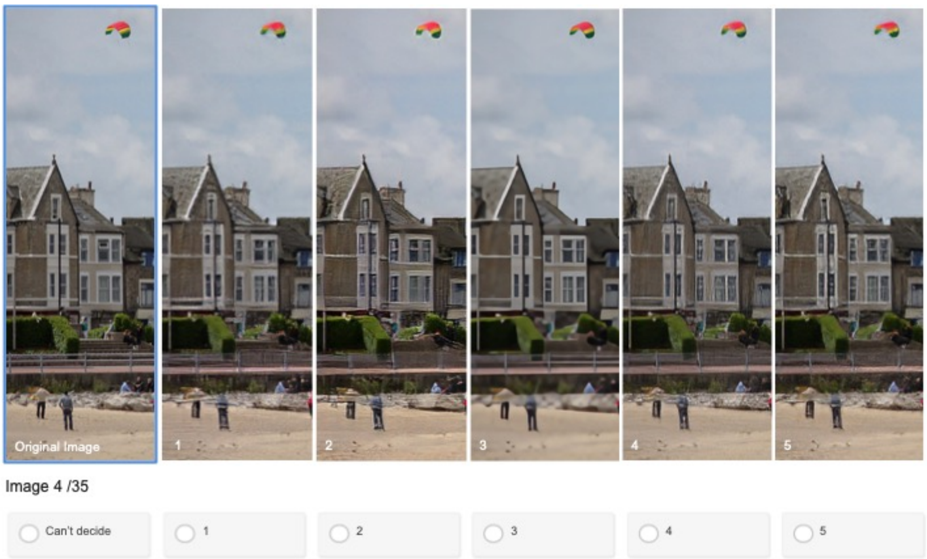


Figure A.4 – Example screenshot of our online survey, to perform a user study and compare our method to state-of-the-art PSNR and GAN-based approaches. In total, 46 persons participated in this survey and 1610 votes were obtained. Users selected the images produced by SROBB (ours) 38.3% while ESRGAN, SFT-GAN, SRGAN, RCAN, and “Cannot decide” had 27.1%, 13.9%, 12.5%, 4.7%, and 8.3% of the votes, respectively. In total, in 42.9% of images we were the winning choice by the majority of votes for SROBB.

span for decision making is not biased by the type of the image, this experiment needs to be extended with significantly more number of images.

A.2 Related to Chapter 5

In this section, you find the supporting information of the article **“Test-Time Adaptation for Super-Resolution: You Only Need to Overfit on a Few More Images”**, presented in Chapter 6. In particular, first, we present more detail about the training of our model, including the generator’s architecture and fine-tuning parameters. Then, the effect of different convolutional layers in our approach is investigated. Finally, we discuss our tuning method to find appropriate values for K and M (these variables are introduced in Chapter 6).

A.2.1 Training details

Generator’s architecture The architecture of the generator, based on [24], is shown in Figure A.5. The generator is a feed-forward CNN, consisting of convolutional layers and several residual blocks; the low-resolution image I^{LR} is passed through the first convolutional layer with a ReLU activation function and a 64 channel output. This output is subsequently passed through 32 residual blocks. Each block has two convolutional layers with 3×3 filters and 256 channel feature maps. Each one is followed by a ReLU activation. By using a long skip connection, the output of the final residual block is concatenated with the features of the first convolutional layer and is then passed through two upsampling blocks, where each one doubles the size of the feature map. Finally, the result is filtered by the last convolutional layer to get the super-resolved image I^{SR}. This setup aims at upsampling with a scale factor of four; the number of upsampling blocks could be modified based on different scaling factors.

Fine-tuning In Fig. A.5, the trainable convolutional layers are highlighted with the yellow box; other parameters are frozen. This has be done specifically to force the fine-tuning to make changes to the filters of the network’ feature extractor rather than manipulating the upsampling layers of the network, thereby yielding a plausible solution. The fine-tuning is

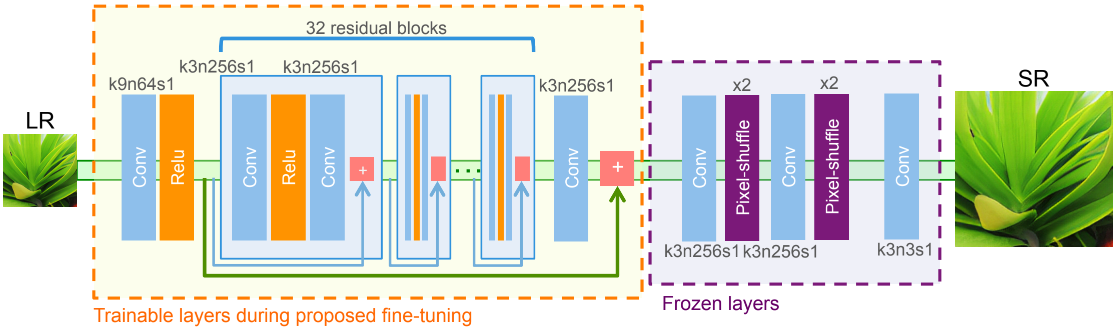


Figure A.5 – The network architecture of the generator. We highlight (yellow bounding box) the feature extractor layers which have been trained during fine-tuning stage, while keeping the other upsampling layers (purple bounding box) frozen.

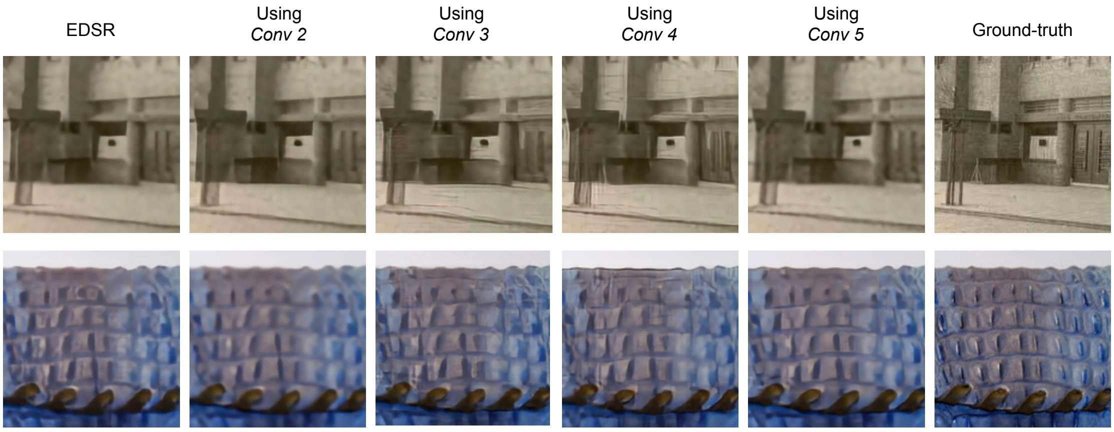


Figure A.6 – Differences in the perceptual quality obtained at different VGG network layers including conv2, conv3, conv4, and conv5, respectively.

performed with the mini-batches of 4 images, corresponding to random crops of 32 × 32 pixels from our constructed dataset. We choose a relatively low learning rate of $1e-4$ for a gradual change in the network parameters.

Baseline with perceptual loss The generator used in this setting is the same as our PSNR-based approach.

The training is divided into two steps; first, the SR decoder was pre-trained with only the pixel-wise cost function for 20 epochs. Then, for the second step, we continue the training for 35 more epochs with a new loss function containing three loss terms: 1- Pixel-wise loss ($\mathscr{L}_1$), 2- an adversarial loss ($\mathscr{L}_{adv}$), and 3- the perceptual loss function [80] ($\mathscr{L}_{vgg}$) using a layer of the pretrained VGG-19 network [40]. The total loss can be formulated as follows:

$$\mathscr{L}_{total} = \alpha\mathscr{L}_1 + \beta\mathscr{L}_{vgg} + \gamma\mathscr{L}_{adv} \tag{A.1}$$

where α, β, γ and δ are the corresponding weights of the loss terms used to train our network, and as proposed by [43], were set to $1e-2$, $5e-3$ and 1, respectively. The Adam optimizer [96] was used during both steps. The learning rate was set to $1e-3$ and then has been decayed by a factor of 10 every 20 epochs. We also alternately optimized the discriminator with similar architecture and settings to those proposed by [43].

A.2.2 Effect of different convolutional layers

In this section, we investigate the effectiveness of using different convolutional layers of the VGG network in our approach. Specifically, we show results using the conv2, conv4, and conv5 layers in Fig. A.6. We base our selection on the visual/perceptual quality of the outputs. For example, we found that conv2 and conv5 produced suboptimal results compared to other layers. Ultimately, based on the visual/perceptual quality, we chose to use the conv3 layer.

A.2.3 Best values for K and M

In this section, we go more into detail about how we chose the number of images per filter K to construct our dataset used in the fine-tuning and the number of filters M to consider with respect to the test image. We presented results using $K = 2, M = 5$. We tuned these parameters based on the perceptual quality of the images generated by varying K and M over a range of values. We focused on the best perceptual quality, as some decreases in PSNR/SSIM values are expected. In Fig. A.7, we show the results for the combinations generated by $K = 1, 2, 5, 9$ and $M = 1, 2, 5, 10$. We can observe that results obtained by very few images for fine-tuning (e.g. $K = 1$ and $M = 1$) contain artefacts, while increasing both K and M results in more realistic and appealing results ($2 \leq K, M \leq 5$). Finally, we note that increasing both K and M significantly ($K, M > 5$) produces blurry images, toward same solution as the EDSR baseline.

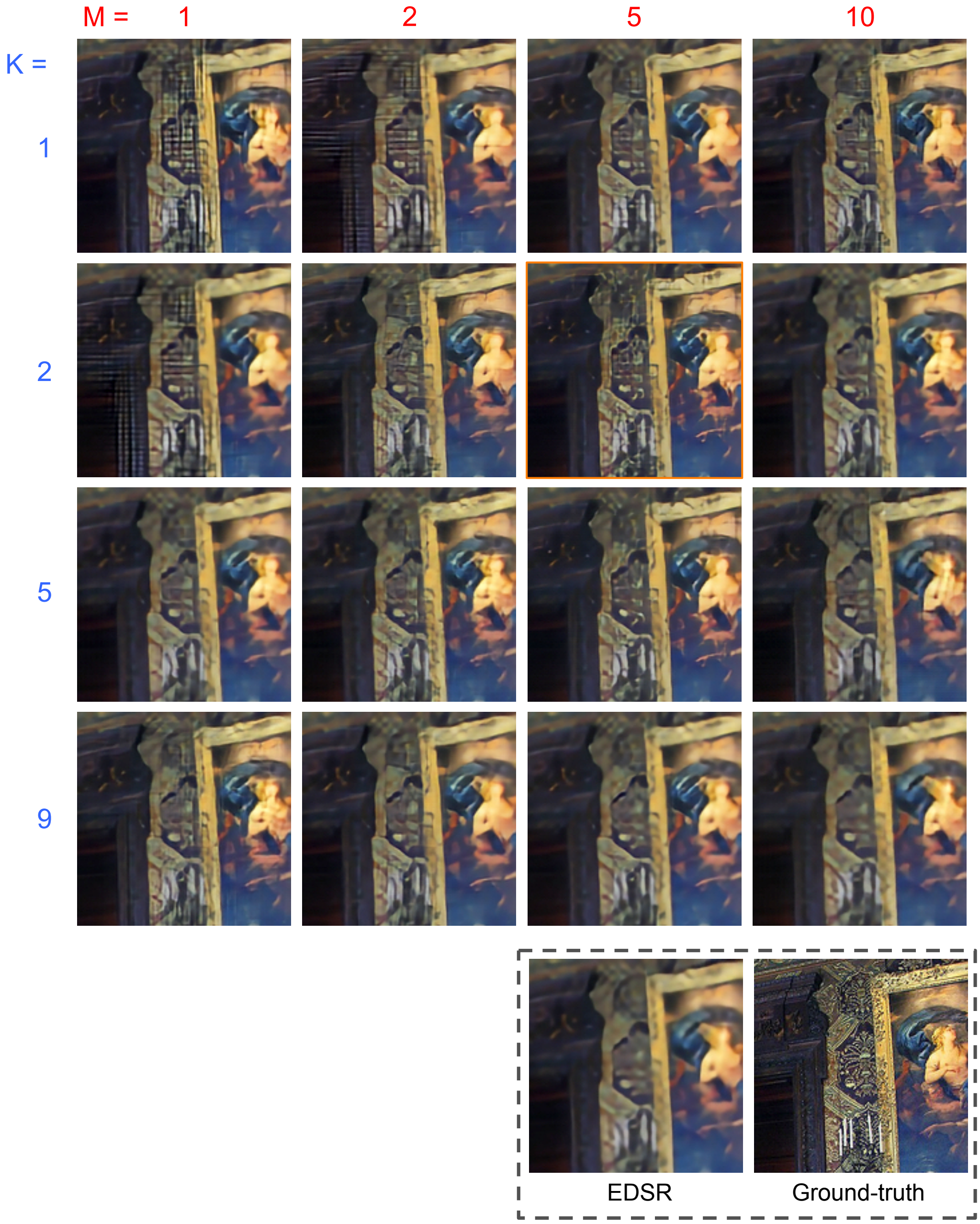


Figure A.7 – Differences in the perceptual quality obtained with different combinations of K and M.

A.3 Related to Chapter 6

In this secton, you find the supporting information of the article **“Benefiting from Bicubically Down-Sampled Images for Learning Real-World Image Super-Resolution”**, presented in Chapter 6. In particular, in this section, we first use ESRGAN [43], and RCAN [35] methods for the SR decoder as the second step of our proposed method (RBSR) to demonstrate the generalization capability of the bicubic look-alike generator. Then, we provide a quantitative analysis of the proposed approach by using the RealSR [63] test set images. In addition, we present more details concerning the computational cost of the proposed method.

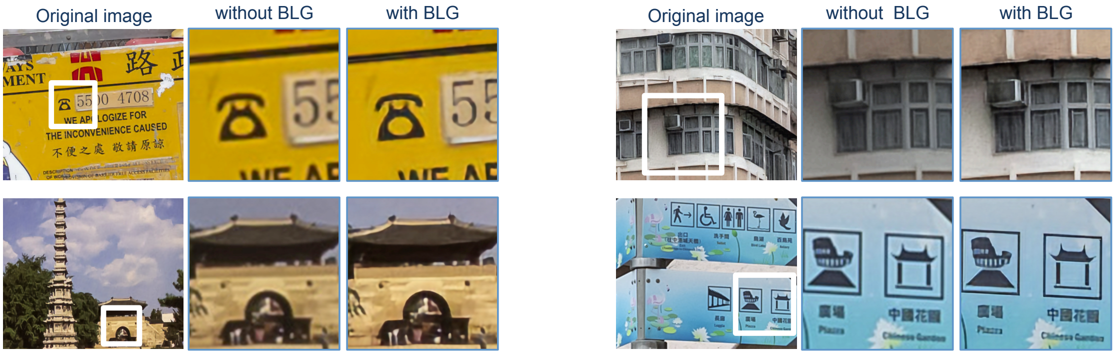
A.3.1 Generalization capabilities of the bicubic look-alike generator

Our proposed approach (RBSR) is a two step procedure. The first step transforms the real LR image using the bicubic look-alike generator. The second step uses any generic SR decoder trained on bicubically downsampled images, taking the transformed LR image as input. In the paper, for the qualitative comparison and the user study, we used a pre-trained EDSR network for this second step. Here, we show the robustness and generalizability of our two step approach by replacing the EDSR network with pretrained ESRGAN and RCAN models. To do so, we compare the results of these models on real LR images and our transformed LR images obtained from the bicubic look-alike generator. Experimental results demonstrate that these SR methods generate more plausible results with greater perceptual quality when fed with transformed LR images instead of real LR images (see Figure A.8).

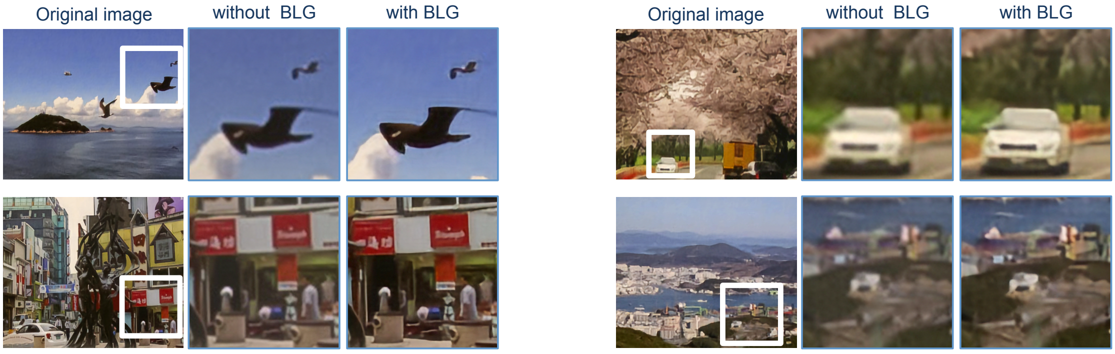
A.3.2 Quantitative results

In this work we tackle the real-world SR problem, where the downsampling operator is not known and therefore no ground-truth is available. Hence, calculating distortion metrics such as PSNR and SSIM is not possible for test images that truly reflect this problem (original images from smartphones, TV streams, etc.). Although, as mentioned previously, RealSR [63] is the only dataset with physically produced high and low-resolution image pairs and is the closest existing dataset to real low and high resolution pairs.

Table A.3 shows the SSIM and PSNR values estimated between super-resolved images of RealSR LR test images and their HR counterparts, using bicubic upsampling, EDSR-real [24], the RealSR network [63], DPSR [128] and our proposed method. The training details of each method is presented in Section 4.3 of the main manuscript. We also add the perception index (PI) metric to our evaluation; this index combines two no-reference image quality measures of Ma et al. [69] and NIQE [67] and was shown to have a higher correlation with human opinion than other commonly used metrics [68]. As PI is a no-reference metric, it can be also used for test images that have no ground-truth.



(a) RCAN [35]



(b) ESRGAN [43]

Figure A.8 – Comparison results of RCAN (a) and ESRGAN (2) methods on original images from the RealSR dataset and our transformed LR images, generated by our bicubic look-alike generator (BLG). Experimental results demonstrate that these SR methods generate more plausible results with greater perceptual quality when fed with transformed LR images instead of real LR images.

Dataset	Method	bicubic	SRResNet	RCAN	EDSR-real	DPSR	RealSR	**RBSR**
RealSR	SSIM	0.77	0.79	0.80	0.81	0.79	0.81	**0.82**
	PSNR	26.63	26.98	27.11	26.51	27.02	**28.05**	26.54
	PI	9.28	9.06	9.19	7.94	9.12	8.97	**7.76**
DIV2K	SSIM/PSNR	-	-	-	no ground-truth	-	-	-
HR	PI	10.02	9.62	9.81	9.01	9.36	9.19	**8.48**
DPED	SSIM/PSNR	-	-	-	no ground-truth	-	-	-
(cellphones)	PI	10.24	9.91	10.02	9.62	9.73	9.55	**7.92**
TV	SSIM/PSNR	-	-	-	no ground-truth	-	-	-
Streams	PI	11.52	10.71	10.64	**10.04**	11.19	10.32	10.15

Table A.3 – Comparison of bicubic interpolation, SRResNet [19], RCAN [35], EDSR [24], DPSR [128], RealSR [63] and RBSR (ours) on different presented test sets. Best measures (SSIM ↑, PSNR [dB] ↑, PI ↓) are highlighted in bold.

Name	Description	SSIM	PSNR
$RBSR_{MSE}$	only $\mathscr{L}_{MSE}$ loss	0.788	27.69
$RBSR_{E}$	only $\mathscr{L}_1$ loss	0.792	**27.95**
$RBSR_{EP}$	$\mathscr{L}_1 + \mathscr{L}_{perceptual}$	0.811	26.98
$RBSR_{EPA}$	$\mathscr{L}_1 + \mathscr{L}_{perceptual} + \mathscr{L}_{adversarial}$	0.798	26.60
$RBSR_{EBA}$	$\mathscr{L}_1 + \mathscr{L}_{bicubic perceptual} + \mathscr{L}_{adversarial}$	**0.835**	26.73
$RBSR$	$\mathscr{L}_1 + \mathscr{L}_{bicubic perceptual} + \mathscr{L}_{adversarial}+$ Copying mechanism	0.820	26.54

Table A.4 – Comparing the effect of each proposed component of the bicubic look-alike generator on LR and HR images of [63] test set. Best measures (SSIM ↑, PSNR [dB] ↑) are highlighted in bold. As mentioned earlier, **these metrics are not directly correlated to the perceptual quality, therefore, we chose our best baseline based on qualitative comparison shown in Figure 5 and Figure 6 of the manuscript, comparing $RBSR_{EPA}$ to $RBSR_{EBA}$ and $RBSR_{EBA}$ to $RBSR$, respectively.**

A.3.3 Ablation study

In this section, we perform another study to investigate the effectiveness of each proposed component of the bicubic look-alike generator. We compare the performance of our network trained with the combinations of different settings such as different loss functions, and trainings with and without copying mechanism. These setting are listed in Table A.4. We calculate PSNR and SSIM for each setting on RealSR [63] test set, the only available dataset with ground-truth for real-world SR task. For each setting, SSIM and PSNR values are calculated after upsampling the picture by a fixed ×4 SR decoder and comparing it to the RealSR ground-truth.

As it is already emphasized in the Section 4 of the manuscript, the distortion metrics are not directly correlated to the perceptual quality as judged by human raters, therefore, we chose our best baseline based on qualitative comparisons such as Figure 5 and Figure 6 of the mains manuscript. Our best baseline is then compared to state-of-the-art works on real-world SR by an extensive user study, following the standard procedure of the ICCV AIM 2019 challenge [123] on Real-world Super-Resolution.

A.3.4 Computational cost

In our paper, we compared our two step approach (RBSR), our end-to-end comparison (EDSR-real), RealSR [63], and DPSR [128]. In terms of computational cost, both RealSR and DPSR have different disadvantages. RealSR's network calculations take place in the high-resolution space, incurring a heavy memory overhead cost. For example, running the model on CPU requires 19 GB of RAM for an image of size 1200 × 1200, which is the maximum possible. DPSR is an iterative algorithm, requiring multiple forward passes and multiple deblurring steps in order to converge to an acceptable solution; DPSR uses an iterative approach by default for real LR images. Hence, these two algorithms have either high memory overhead or high computation time overhead. In contrast, RBSR requires two forward passes per input image. The first network is relatively lightweight, as it operates exclusively in the LR space. The second network can be any generic SR decoder for bicubically downsampled images. The complete pipeline (using EDSR as the SR decoder) reconstructs 1024 × 768 pixel images at 26.9 FPS, using a GeForce GTX 1080 Ti. Our end-to-end setting (EDSR-real) reconstructs the same size images at 33.7 FPS using the same GPU.

Bibliography

[1] R. Tsai and T. Huang, "Multiframe image restoration and registration," *Advances in Computer Vision and Image Processing*, vol. 1, p. 317–339, 1984.

[2] X. Wang, K. Yu, C. Dong, and C. Change Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 606–615, 2018.

[3] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Transactions on Image Processing*, vol. 20, pp. 1838–1857, July 2011.

[4] J. Sun, J. Zhu, and M. F. Tappen, "Context-constrained hallucination for image super-resolution," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 231–238, June 2010.

[5] R. Timofte, V. D. Smet, and L. V. Gool, "Semantic super-resolution: When and where it is useful?," *Computer Vision and Image Understanding*, Sept. 2015.

[6] B. Bascle, A. Blake, and A. Zisserman, "Motion deblurring and super-resolution from an image sequence," in *Computer Vision — ECCV '96* (B. Buxton and R. Cipolla, eds.), (Berlin, Heidelberg), pp. 571–582, Springer Berlin Heidelberg, 1996.

[7] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, vol. 14, no. 2, pp. 47–57, 2004.

[8] D. Thapa, K. Raahemifar, W. Bobier, and V. Lakshminarayanan, "Comparison of super-resolution algorithms applied to retinal images," *Journal of biomedical optics*, vol. 19, p. 56002, 05 2014.

[9] S. Lertrattanapanich and N. K. Bose, "High resolution image formation from low resolution frames using delaunay triangulation," vol. 11, p. 1427–1441, Dec. 2002.

[10] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1646–1658, 1997.

[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, pp. 295–307, 2014.

[12] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1646–1654, 2015.

[13] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[14] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," pp. 2808–2817, 07 2017.

[15] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," vol. 9906, pp. 391–407, 10 2016.

[16] 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," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1874–1883, 2016.

[17] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[19] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105–114, 2016.

[20] N. Ahn, B. Kang, and K. Sohn, "Fast, accurate, and, lightweight super-resolution with cascading residual network," *CoRR*, vol. abs/1803.08664, 2018.

[21] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sept. 2018.

[22] Z. Hui, X. Wang, and X. Gao, "Fast and accurate single image super-resolution via information distillation network," *CoRR*, vol. abs/1803.09454, 2018.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *ECCV*, 2016.

[24] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," *CoRR*, vol. abs/1707.02921, 2017.

[25] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1637–1645, 2016.

[26] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2790–2798, 2017.

[27] 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 (ICCV)*, pp. 4539–4547, 2017.

[28] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5835–5843, 2017.

[29] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 370–378, 2015.

[30] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3106–3121, 2019.

[31] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

[32] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4809–4817, 2017.

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

[34] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1664–1673, 2018.

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

[36] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss functions for image restoration with neural networks,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[37] M. Bevilacqua, A. Roumy, C. Guillemot, and M. line Alberi Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” in *Proceedings of the British Machine Vision Conference*, pp. 135.1–135.10, BMVA Press, 2012.

[38] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in *Curves and Surfaces* (J.-D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M.-L. Mazure, and L. Schumaker, eds.), (Berlin, Heidelberg), pp. 711–730, Springer Berlin Heidelberg, 2012.

[39] J. Bruna, P. Sprechmann, and Y. LeCun, “Super-resolution with deep convolutional sufficient statistics,” *CoRR*, vol. abs/1511.05666, 2015.

[40] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.

[41] M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch, “Enhancenet: Single image super-resolution through automated texture synthesis,” *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4501–4510, 2017.

[42] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423, 2016.

[43] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, “Esrgan: Enhanced super-resolution generative adversarial networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[44] J. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, 2017.

[45] 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 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 814–81409, 2018.

[46] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2672–2680, Curran Associates, Inc., 2014.

[47] M. Mehralian and B. Karasfi, “Rdcgan: Unsupervised representation learning with regularized deep convolutional generative adversarial networks,” in *2018 9th Conference*

on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium, pp. 31–38, 2018.

[48] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," 12 2018.

[49] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, 2017.

[50] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Mirrorgan: Learning text-to-image generation by redescription," pp. 1505–1514, 06 2019.

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

[52] Y. Gao, R. Singh, and B. Raj, "Voice impersonation using generative adversarial networks," pp. 2506–2510, 04 2018.

[53] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2813–2821, 2017.

[54] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 214–223, PMLR, 06–11 Aug 2017.

[55] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, pp. 5767–5777, Curran Associates, Inc., 2017.

[56] Z. Chen and Y. Tong, "Face super-resolution through wasserstein gans," 05 2017.

[57] 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 (ECCV)*, Sept. 2018.

[58] 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 Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, pp. 416–423 vol.2, July 2001.

[59] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *IEEE Conference on Computer Vision and Pattern Recognition)*, 2015.

[60] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, B. Lim, *et al.*, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[61] A. Fujimoto, T. Ogawa, K. Yamamoto, Y. Matsui, T. Yamasaki, and K. Aizawa, "Manga109 dataset and creation of metadata," MANPU '16, (New York, NY, USA), Association for Computing Machinery, 2016.

[62] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey, and L. Van Gool, "Dslr-quality photos on mobile devices with deep convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3277–3285, 2017.

[63] J. Cai, H. Zeng, H. Yong, Z. Cao, and L. Zhang, "Toward real-world single image super-resolution: A new benchmark and a new model," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[64] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," vol. 60, p. 91–110, Nov. 2004.

[65] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

[66] M. W. Gondal, B. Schölkopf, and M. Hirsch, "The unreasonable effectiveness of texture transfer for single image super-resolution," *arXiv preprint arXiv:1808.00043*, 2018.

[67] A. Mittal, R. Soundararajan, and A. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, pp. 209–212, 2013.

[68] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, "2018 PIRM challenge on perceptual image super-resolution," *CoRR*, vol. abs/1809.07517, 2018.

[69] C. Ma, C. Yang, X. Yang, and M. Yang, "Learning a no-reference quality metric for single-image super-resolution," *CoRR*, vol. abs/1612.05890, 2016.

[70] A. Shocher, N. Cohen, and M. Irani, ""zero-shot" super-resolution using deep internal learning," 2017.

[71] T. Michaeli and M. Irani, "Nonparametric blind super-resolution," in *2013 IEEE International Conference on Computer Vision*, pp. 945–952, 2013.

[72] 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)*, pp. 185–200, 2018.

[73] X. Zhang, Q. Chen, R. Ng, and V. Koltun, "Zoom to learn, learn to zoom," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3757–3765, 2019.

[74] Q. Chen and V. Koltun, “Photographic image synthesis with cascaded refinement networks,” *CoRR*, vol. abs/1707.09405, 2017.

[75] W. Zou and P. C. Yuen, “Very low resolution face recognition problem,” *IEEE Transactions on Image Processing*, vol. 21, pp. 327–340, Jan. 2012.

[76] W. Shi, J. Caballero, C. Ledig, X. Zhuang, W. Bai, K. Bhatia, A. M. S. M. de Marvao, T. Dawes, D. O’Regan, and D. Rueckert, “Cardiac image super-resolution with global correspondence using multi-atlas patchmatch,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, (Berlin, Heidelberg), pp. 9–16, Springer Berlin Heidelberg, 2013.

[77] J. Park, B. Nam, and H. Yoo, “A high-throughput 16× super resolution processor for real-time object recognition soc,” in *2013 Proceedings of the ESSCIRC (ESSCIRC)*, pp. 259–262, Sept. 2013.

[78] Q. Yang, R. Yang, J. Davis, and D. Nistér, “Spatial-depth super resolution for range images,” *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[79] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 184–199, Springer International Publishing, 2014.

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

[81] B. Wu, H. Duan, Z. Liu, and G. Sun, “SRPGAN: perceptual generative adversarial network for single image super resolution,” *CoRR*, vol. abs/1712.05927, 2017.

[82] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, pp. 41–75, July 1997.

[83] C. E. Duchon, “Lanczos filtering in one and two dimensions,” *Journal of Applied Meteorology*, vol. 18, no. 8, pp. 1016–1022, 1979.

[84] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, “Coupled dictionary training for image super-resolution,” *Trans. Img. Proc.*, vol. 21, pp. 3467–3478, Aug. 2012.

[85] C.-Y. Yang, J.-B. Huang, and M.-H. Yang, “Exploiting self-similarities for single frame super-resolution,” in *Computer Vision – ACCV 2010* (R. Kimmel, R. Klette, and A. Sugimoto, eds.), (Berlin, Heidelberg), pp. 497–510, Springer Berlin Heidelberg, 2011.

[86] R. Mechrez, I. Talmi, F. Shama, and L. Zelnik-Manor, “Learning to maintain natural image statistics,” *arXiv preprint arXiv:1803.04626*, 2018.

[87] W. Ren, J. Pan, X. Cao, and M. Yang, “Video deblurring via semantic segmentation and pixel-wise non-linear kernel,” *CoRR*, vol. abs/1708.03423, 2017.

[88] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[89] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, p. 4, IEEE, 2017.

[90] H. Caesar, J. R. R. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1209–1218, 2018.

[91] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 658–666, Curran Associates, Inc., 2016.

[92] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *CoRR*, vol. abs/1511.05440, 2015.

[93] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, 2017.

[94] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014.

[95] M. Kampffmeyer, A. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 680–688, June 2016.

[96] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[97] W. Yu, K. Yang, Y. Bai, H. Yao, and Y. Rui, "Visualizing and comparing convolutional neural networks," 2014.

[98] A. Mahendran and A. Vedaldi, "Visualizing deep convolutional neural networks using natural pre-images," *International Journal of Computer Vision*, vol. 120, p. 233–255, May 2016.

[99] K. Yu, C. Dong, L. Lin, and C. C. Loy, "Crafting a toolchain for image restoration by deep reinforcement learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2443–2452, 2018.

[100] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks," *CoRR*, vol. abs/1505.07376, 2015.

[101] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *CoRR*, vol. abs/1508.06576, 2015.

[102] J. Yosinski, J. Clune, A. M. Nguyen, T. J. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *CoRR*, vol. abs/1506.06579, 2015.

[103] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, vol. abs/1312.6034, 2013.

[104] S. Vasu, T. M. Nimisha, and A. N. Rajagopalan, "Analyzing perception-distortion tradeoff using enhanced perceptual super-resolution network," in *ECCV Workshops*, 2018.

[105] E. Pérez-Pellitero, M. S. M. Sajjadi, M. Hirsch, and B. Schölkopf, "Photorealistic video super resolution," in *Workshop and Challenge on Perceptual Image Restoration and Manipulation (PIRM) at the 15th European Conference on Computer Vision (ECCV)*, 2018.

[106] 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, pp. 600–612, Apr. 2004.

[107] J. Van Ouwerkerk, "Image super-resolution survey," *Image and vision Computing*, vol. 24, no. 10, pp. 1039–1052, 2006.

[108] S. Anwar, S. Khan, and N. Barnes, "A deep journey into super-resolution: A survey," *arXiv preprint arXiv:1904.07523*, 2019.

[109] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11065–11074, 2019.

[110] Z. Zhang, Z. Wang, Z. Lin, and H. Qi, "Image super-resolution by neural texture transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7982–7991, 2019.

[111] H. Zheng, M. Ji, H. Wang, Y. Liu, and L. Fang, "Crossnet: An end-to-end reference-based super resolution network using cross-scale warping," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 88–104, 2018.

[112] J. Jiang, Y. Yu, Z. Wang, S. Tang, R. Hu, and J. Ma, "Ensemble super-resolution with a reference dataset," *IEEE transactions on cybernetics*, 2019.

[113] W. Yang, S. Xia, J. Liu, and Z. Guo, “Reference-guided deep super-resolution via manifold localized external compensation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1270–1283, 2018.

[114] Y. Blau and T. Michaeli, “The perception-distortion tradeoff,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6228–6237, 2018.

[115] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149, 2018.

[116] R. Zhang, P. Isola, and A. A. Efros, “Split-brain autoencoders: Unsupervised learning by cross-channel prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1058–1067, 2017.

[117] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[118] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, “Network dissection: Quantifying interpretability of deep visual representations,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

[119] M. Simon and E. Rodner, “Neural activation constellations: Unsupervised part model discovery with convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1143–1151, 2015.

[120] Z. Han, E. Dai, X. Jia, S. Chen, C. Xu, J. Liu, and Q. Tian, “Unsupervised image super-resolution with an indirect supervised path,” *arXiv preprint arXiv:1910.02593*, 2019.

[121] Y. Wei, S. Gu, Y. Li, and L. Jin, “Unsupervised real-world image super resolution via domain-distance aware training,” *arXiv preprint arXiv:2004.01178*, 2020.

[122] A. Lugmayr, M. Danelljan, and R. Timofte, “Unsupervised learning for real-world super-resolution,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[123] A. Lugmayr, M. Danelljan, R. Timofte, M. Fritsche, S. Gu, K. Purohit, P. Kandula, M. Suin, A. N. Rajagopalan, N. H. Joon, Y. S. Won, G. Kim, D. Kwon, C.-C. Hsu, C.-H. Lin, Y. Huang, X. Sun, W. Lu, J. Li, X. Gao, and S. Bell-Kligler, “Aim 2019 challenge on real-world image super-resolution: Methods and results,” 2019.

[124] T. Köhler, M. Bätz, F. Naderi, A. Kaup, A. K. Maier, and C. Riess, “Benchmarking super-resolution algorithms on real data,” *CoRR*, vol. abs/1709.04881, 2017.

[125] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin, “Accurate blur models vs. image priors in single image super-resolution,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2832–2839, 2013.

[126] R. Zhou and S. Susstrunk, “Kernel modeling super-resolution on real low-resolution images,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2433–2443, 2019.

[127] J. Gu, H. Lu, W. Zuo, and C. Dong, “Blind super-resolution with iterative kernel correction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1604–1613, 2019.

[128] K. Zhang, W. Zuo, and L. Zhang, “Deep plug-and-play super-resolution for arbitrary blur kernels,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1671–1681, 2019.

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

[130] C. Chen, Z. Xiong, X. Tian, Z.-J. Zha, and F. Wu, “Camera lens super-resolution,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1652–1660, 2019.

[131] S. Bell-Kligler, A. Shocher, and M. Irani, “Blind super-resolution kernel estimation using an internal-gan,” 2019.

[132] M.-Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in *Advances in neural information processing systems*, pp. 700–708, 2017.

[133] J.-B. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transformed self-exemplars,” in *IEEE Conference on Computer Vision and Pattern Recognition)*, 2015.

Personal information
04 October 1990

Contacts
radmosa@gmail.com
in saeed-rad
S radmosa

Saeed Rad

Education

Ph.D. Student, LTS5, EPFL, Lausanne, Switzerland 2018 - present

In Electrical Engineering, with the main focus on deep learning methods for computer vision tasks, under the supervision of Prof. Jean-Philippe Thiran.
- Thesis: *Context-Aware Image Super-Resolution Using Deep Neural Networks*.

M.Sc., EPFL, Lausanne, Switzerland 2013 - 2015

In Electrical Engineering and Information Technology (IT). Classes included: *Advanced Signal Processing, Image Processing, Image Analysis and Pattern Recognition, Information theory and coding and Computer Graphics.*

B.Sc., EPFL, Lausanne, Switzerland 2010 - 2013

In Electrical Engineering. Classes included: *Software Engineering, Digital System Design, Signal Processing, Control System, and Circuits and Systems.*

Work Experience

Research Intern, Siemens Healthineers, Lausanne June 2020 - Oct 2020

Studying and developing deep learning-based CV methods for 3D segmentation of human body parts by benefiting from contextual information within 3D data.

Scientific Collaborator, LTS5, EPFL, Lausanne Aug 2015 - Mar 2018

- 'Clean City Index' project: Study and development the first deep learning-based system to detect wastes and estimate a cartography of city pollution, using images taken by cameras mounted on street sweeper cars and drones.
- 'Dynamic image analysis for Cervical Cancer Detection': Study and development of an algorithm to process and extract temporal information from captured videos by smartphones, taken by medical assistants, to detect cervical cancer. In collaboration with Geneva University Hospital.

Teaching Assistant, LTS5, EPFL, Lausanne

- **Supervision of B.Sc. and M.Sc projects** Over 10 master and bachelor students for their semester and master projects. Aug 2015 - present
- **Image analysis and pattern recognition** course for master students. Coordinator and responsible for providing structure and guidance for student projects and exams. Feb 2017 - present
- **Fundamentals of electrical circuits and systems** course for master students. Responsible for preparing lab sessions, and answering and helping students for their exercises. Sept 2019 - present
- **Signal processing** course for bachelor students. Responsible for answering and helping students for their exercises and labs, as well as preparing and grading the exams. Aug 2015 - Feb 2016
- **C++ Programming** course for bachelor students. Responsible for answering and helping students for their exercises and projects, also grading their oral and written exams and projects. Feb 2015 - June 2015

Intern, ABB Groups, Baden, Switzerland Aug 2014 - Feb 2015

Developing machine learning-based algorithms for automated power grid inspection, using images taken from drones, flying around electricity pylons.

Application developer, ESL, EPFL June 2013 - Sept 2013

Developing monitoring applications for Android mobile phones, to capture ECG signals from an embedded device and to analyze the received data.

Languages
English: Fluent
French: Fluent
Persian: Native

Hobbies
Sports: Football, Chess, Cycling and Swimming.

Others: Reading, Video Games and Watching Movies.

List of Publications

S. Rad, T. Yu, B. Bozorgtabar, J-Ph. Thiran, **'Test-Time Adaptation for Super-Resolution: You Only Need to Overfit on a Few More Images'** *2021*, Under review.

S. Rad, T. Yu, C. Musat, H.K. Ekenel, B. Bozorgtabar, J-Ph. Thiran, **'Benefitting from Bicubically Down-Sampled Images for Learning Real-World Image Super-Resolution'**, *Winter Conference on Applications of Computer Vision (WACV), 2021*.

S. Rad, B. Bozorgtabar, C. Musat, U. V. Marti, M. Basler, H.K Ekenel, J-Ph. Thiran, **'Benefiting from multitask learning to improve single image super-resolution'**, **Rad** et al., *Neurocomputing Journal, 2020*.

S. Rad, U. V. Marti, M. Basler, B. Bozorgtabar, H.K Ekenel, J-Ph. Thiran, **'SROBB: Targeted Perceptual Loss for Single Image Super-Resolution'**, *International Conference in Computer Vision (ICCV), 2019*.

B. Bozorgtabar, **S. Rad**, D. Mahapatra, J-Ph. Thiran, **'SynDeMo: Synergistic Deep Feature Alignment for Joint Learning of Depth and Ego-Motion'**, *International Conference in Computer Vision (ICCV), 2019*.

B. Bozorgtabar, **S. Rad**, H.K. Ekenel, J-Ph. Thiran, **'Using Photorealistic Face Synthesis and Domain Adaptation to Improve Facial Expression Analysis'**, *International Conference on Automatic Face and Gesture Recognition (FG), 2019*.

B. Bozorgtabar, **S. Rad**, H.K. Ekenel, J-Ph. Thiran, **'Learn to synthesize and synthesize to learn'**, *Computer Vision and Image Understanding Journal (CVIU), 2019*.

S. Rad, A. von Kaenel, A. Droux, F. Tieche, N. Ouerhani, H.K Ekenel, J-Ph. Thiran, **'A computer vision system to localize and classify wastes on the streets'**, *International Conference on Computer Vision Systems (ICVS), 2017*.

Other Projects

Artistic Image to Poetry Translation, IDIAP Laboratory.
- In this project, we added an artistic twist into the world of "text to image to text" translation by using generative adversarial networks.

Apizoom project, Supervising students at LTS5 laboratory.
- We used deep learning to quantify the Verroa parasite in honey bee hives.

Cervical Cancer Detection, Master thesis at LTS5 Laboratory.
- We studied and developed an algorithm to detect cervical cancer from RGB images.

Predicting Local Twitter Users Based on Their Tweets, IDIAP Laboratory.
- We used natural language processing techniques to extract features from tweets and identify local users.

Pigeon Messenger
- We developed an Android application to transfer messages by virtual pigeons.

Designing Games for Embedded Systems
- We developed games in VHDL for Xilinx-5 FPGA and nintendo DS console.

Patent

'Method and System for Automated 3D Segmentation of Human Body Parts', **S. Rad**, B. Marechal, T. Hilbert, 2020, filed at the European Patent Office.

Skills

Python, C++, Tensorflow, PyTorch, Keras, Tensorlayer, TensorRT, OpenCV, Matlab, Java, Git, Adobe after effect, Adobe photoshop, 3DsMax, Corel draw, MySQL, SQL Server

Honors

- Leading "Clean City Index" project to become the headline in many medias, including:
  - EPFL front page (this link)
  - SwissInfo (this link)
  - RTS (this link)
  - Le Matin (this link)
  - 24 heures (this link)
  - 20Minutes journal
- Finalist for the best paper award at ICVS 2017.
- Reviewer for CVPR, ECCV, ACCV, WACV and FGCS conferences and journal.