”Alexandru Ioan Cuza” University of Iași

Faculty of Computer Science

Master’s Thesis

Efficient Deep Single-Image Super-Resolution on Mobile Devices

Proposed by

Marian-Sergiu Nistor

Scientific coordinator

Prof. Dr. Radu Timofte

Session: *June - July, 2023*

”Alexandru Ioan Cuza” University of Iași
Faculty of Computer Science

Efficient Deep Single-Image Super-Resolution on Mobile Devices

Proposed by
Marian-Sergiu Nistor

Scientific coordinator
Prof. Dr. Radu Timofte

Session: *June - July, 2023*

Abstract

The task of single image super-resolution (SISR) has been approached several times in the recent history, due to its relevance in supporting other computer vision algorithms (e.g. face recognition, object detection), which might have certain requirements in terms of input data resolution, so as to provide accurate results. Modern implementations make use of representation learning for mapping the SR space. Thus, the paper represents a study on deep learning-based methods for efficient mobile single image super-resolution, using various up-scaling factors (x2, x3, x4), first optimizing for peak signal-to-noise ratio (PSNR), then for perceptual loss. In order to minimize the inference time, the considered architectures are designed to be as shallow as possible, balancing out the number of trainable parameters through width, which ensures a higher level of GPU parallelism.

Keywords: single image super-resolution, efficient super-resolution, mobile super-resolution, deep learning, computer vision

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Dr. Radu Timofte, for his invaluable guidance and encouragement, throughout my research. His expertise has been essential in shaping my scientific thinking and helping me navigate the hardships that came along with the development of this thesis. I would also like to thank my family and friends for supporting me through this year’s adversities.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

The task of single image super-resolution (SISR) implies increasing the resolution of a given low-resolution (LR) image, thus enhancing the level of detail by introducing hallucinated visual artifacts that fit the semantic context. This objective has been extensively explored since its establishment in the late 20th century [40], and its popularity has increased recently, with the emergence of representation learning-based methods [8], which managed to provide better results, both from a peak signal-to-noise ratio perspective and from a perceptual viewpoint.

There is an ongoing effort of improving the performance of SR networks, including in terms of inference time, in order to provide efficient execution, in the context of mobile devices. These constraints require such neural architectures to be shallow and compensate for their light nature through width, ensuring an adequately high trainable parameter count, so that the learned SR space mapping can be as representative as possible.

Super-resolution methods provide significant value by supporting other computer-vision algorithms, such as object detection [33] or facial recognition [44] models, which tend to produce better results on high resolution, feature-rich images. Another important use-case consists of adopting SR models in medical imaging [47] [3], allowing for detecting anomalies and making diagnoses with better accuracy. Lastly, a further relevant application for this task implies reconstructing media that has previously been down-scaled for compression purposes, in a lossy manner [19].

The following sections of this chapter present an outline of the report's context, and the advances yielded by it. Section 1.1 maps out the motivation for addressing the problem of efficient super-resolution on mobile devices. Then, Section 1.2 specifies the objectives and contributions to the thesis. Finally, Section 1.3 details the overall structure of the report.

1.1 Motivation

Recent improvements in terms of mobile compute capabilities, which continue to scale according to Moore's law [29], have enabled the usage of deep computer vision algorithms on smartphones. Given the wide availability of such devices, the constantly developing market requires migrating and deploying the existing models to mobile gadgets.

Concerning the single image super-resolution task, the current neural architectures that address this task have a deep nature, given the need for handling a complex problem space, which implies learning the appropriate feature representations and semantic relationships, in order to reconstruct suitable high-frequency details and super-resolve a low-resolution image. Inferring such deep networks can be costly when performed within the scope of constrained devices, which possess limited computing power, imposing the development of shallower and quicker networks, ideally without trading off efficiency.

1.2 Contributions

This report's objective is to develop and compare various neural network architectures for efficient mobile single image super-resolution, considering multiple up-scaling factors (x2, x3, x4), where the target HR image resolution is 2K (approximately 2048x1080). The models were benchmarked both on computer GPU's, aiming for real-time execution, and on mobile processing units, targeting an inference time of below 2 seconds. Post-training quantization is applied, so as to attempt further runtime reduction. The networks adhere to the constraints stated in the NTIRE 2022 image and super-resolution challenges [18] [23], and they were evaluated in relation to the 2022 event winners and the proposed baseline models.

The metrics considered in the NTIRE challenge benchmarks are peak signal-to-noise ratio [17] (PSNR, logarithmic measurement for determining image quality), structural similarity index measure [42] (SSIM, mathematical estimation for perceived visual fidelity), and the inference time. The studied models are initially trained to optimize for PSNR. Subsequent iterations minimize the learned perceptual image patch similarity score [46] (LPIPS), a measurement index that yields results that are closer to human perception, compared to the SSIM.

1.3 Thesis Structure

The overall structure of the thesis was chosen so as provide a clear and comprehensive understanding of the treated subject. Thus, Chapter 1 aims to describe the general problem setting, laying down the main contextual aspects, in terms of the motivation and the contributions brought by the established research. Then, a set of relevant prior works in the field of super-resolution are presented in Chapter 2, ordered from a thematic point of view, going from traditional computer vision approaches, towards advanced deep learning methods. Subsequently, Chapter 3 details the methods explored during the experimentation process, mentioning the baseline models, and the proposed methods, which represent the exploratory iterations. Chapter 4 first outlines the methodology used for developing and evaluating the models, then goes over the motivation behind the architectural decisions with regard to the network design process. Then, the chapter displays the results achieved by each model, in relation to the considered baselines. Finally, Chapter 5 iterates over the objectives achieved throughout the research process, in relation to the previously stated targets, then discusses potential future work directions.

Chapter 2. Prior Super-Resolution Research

Active research in the area of single image super-resolution began in the 20th century [40]. The state-of-the-art models that employ traditional computer vision algorithms have tackled this problem using learned atom dictionaries and neighbor embedding methods [37] [36], as presented in Section 2.1.

The development of more powerful compute hardware in the first two decades of the 21st century determined an abrupt rise in popularity of deep learning models and, more specific to the field of computer vision, convolutional neural networks, which were successfully utilized in the task of image super-resolution, as described in Section 2.2. Several variants for these network types were proposed over time. Section 2.3 depicts two lightweight models, ESPCN [34] and NCNet[27], which efficiently parallelize the computations, by operating in lower dimensional feature maps, and a high number of channels, reconstructing the high-resolution image using a sub-pixel convolutional layer. Section 2.4 describes the usage of feature distillation networks, which extract features hierarchically through multiple channel splits, using information multi-distillation blocks [16].

Section 2.5 outlines the adoption of attention mechanisms [4] [41] [43], in the context of convolutional neural networks, which, when combined with existing back-projection architectures [14], achieved state-of-the-art performance for single image super-resolution [26].

Section 2.6 provides an overview of U-Net architectures [31] used for super-resolving images [15] [32]. Such models have recently been utilized successfully for the denoising and super-resolving part of latent diffusion networks [30], which presented great performance for various generative tasks.

Finally, Section 2.7 discusses the prior incorporation of GANs [13] in the area of SISR [22], excelling in generating photo-realistic high-resolution images, through adversarial training.

2.1 Traditional Computer-Vision Approaches

State-of-the-art traditional computer-vision approaches reconstruct the high-resolution images based on sparse learned atom dictionaries and neighbor embedding methods [37] [36]. In the case of the A+ algorithm [36], the sparse dictionary atoms, which are comprised of LR and HR image patches, are learned through several iterations over the training dataset, using a dictionary learning algorithm, such as K-SVD [2] and OMP [45]. The neighbor embedding regression model is trained to cluster the dictionary atoms in neighborhoods, using a K-nearest-neighbours approach, where the considered distance metric is the correlation between the atoms. After that, for each LR atom, a projection matrix is computed, so as to associate it with the corresponding HR patch.

Thus, in the inference stage, the LR image is divided into overlapping patches, from which the HR patches are generated, using the neighbor embedding regression model, through the learned projection matrices. The HR image is then reconstructed by connecting the resulting patches.

The features considered by such algorithms are generally comprised of the luminance channel, since the human eye is most sensitive to intensity changes. The other color channels are up-scaled using regular interpolation methods (e.g. bicubic interpolation, nearest-neighbors interpolation).

2.2 Deep Convolutional Models

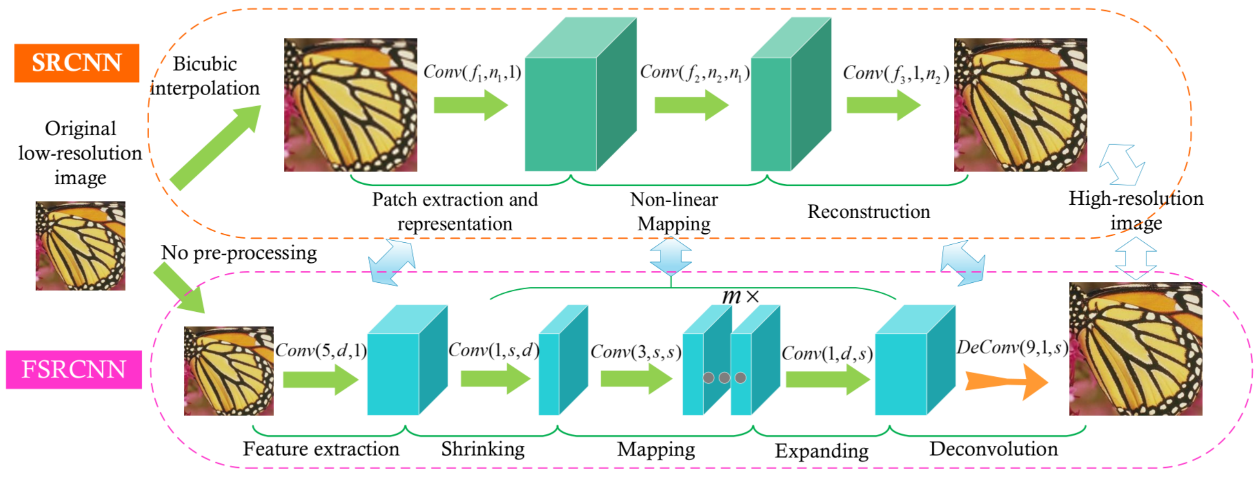


Figure 1: The architectures for the SRCNN [8] and FSRCNN [9] models. The main differences are comprised of the fact that SRCNN initially performs up-scaling through bicubic interpolation, while FSRCNN performs feature extraction on the original LR image, then shrinking the filter count, so as to speed-up the feature mapping process. Lastly, FSRCNN up-scales the resulting feature maps through a deconvolution operation. SRCNN is significantly slower due to the fact that the feature map dimensions are equal to the HR image's dimensions, while FSRCNN feature maps maintain the LR image dimensions.

Deep learning methods proved to achieve better performance for super-resolution tasks, both from a peak signal-to-noise ratio and from a structural similarity index measure perspective. Modern approaches make use of convolutional neural networks to learn the mapping function from the low-resolution to the high-resolution space.

SRCNN [8] represents the initial attempt of doing so, surpassing the state-of-the-art A+ algorithm [36], in terms of precision. Prior to the neural inference process, the LR image is up-scaled using bicubic interpolation. The first stage of the neural network performs patch retrieval through a convolutional layer, while also expanding the channel count, so as to extract image features. The second stage handles non-linear mappings between the LR and HR patches. The last part involves reducing the feature map count to the desired channel count of the HR image. The convolution operations are going to be costly, due to the interpolation operation that is being performed previous the neural network inference stage, which up-scales the LR image to HR resolution.

The FSRCNN [9] model represents a more efficient re-iteration for the original SRCNN model, which presented even better performance. Figure 1 provides a comparison between the two architectures, indicating that FSRCNN does not perform any interpolation in the pre-processing phase, thus maintaining the LR image dimensions throughout the convolutional feature maps. Furthermore, after the initial feature extraction phase, the channel count is additionally reduced, to minimize the runtime. After the non-linear mapping block, the features maps are expanded, so as to encapsulate as much information as possible, prior to the up-scaling phase, which makes use of a deconvolution layer, to increase the image to the destination HR dimensions.

2.3 Sub-Pixel Convolutional Neural Networks

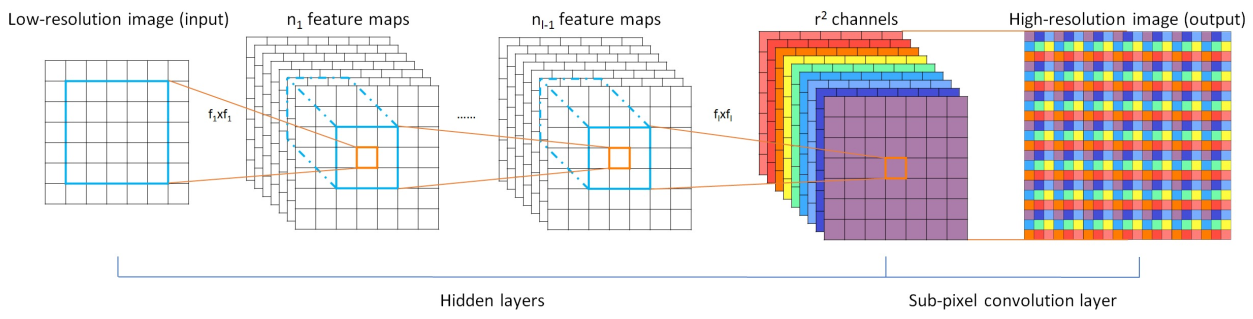


Figure 2: The ESPCN architecture [34], which consists of a fully convolutional feature extraction and non-linear mapping block, followed by the sub-pixel convolution layer, which blends feature map pixel values into the final HR dimension space.

Previous deep convolutional methods performed up-scaling either at the pre-processing step, or at the final stage of the network, using inverse convolutions. The new generation of models make use of sub-pixel convolutional layers, also called depth-to-space layers, to ensure higher parallelism and efficiency when reconstructing the HR image. The sub-pixel convolution does not operate over in the space dimension of the feature maps (width x height), but on the depth component, being highly parallelizable on the GPU cores, thus resulting in a lower inference time. The ESPCN [34] is the first architecture that makes use of the depth-to-space layer as its final block, as it can be seen in Figure 2.

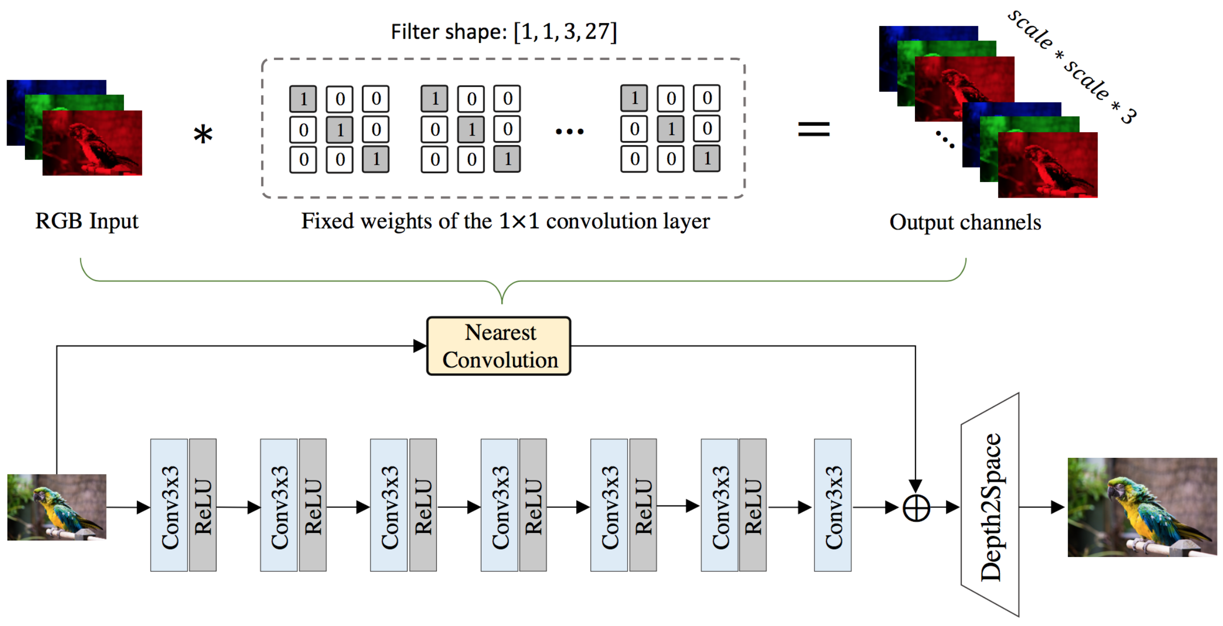
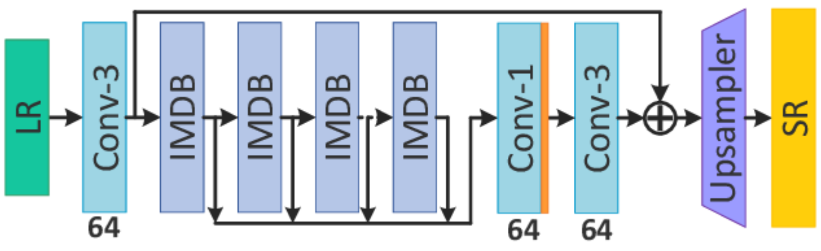


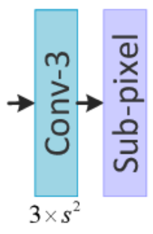
Figure 3: The NCNet model [27], which refines the original ESPCN architecture [34], by adding an additional interpolation branch that uses a nearest convolution, which is equivalent to a frozen convolutional layer with a kernel equal to the identity matrix, forcing the main branch to learn the residual between the HR image and the interpolated image.

As shown in Figure 3, the NCNet model [27] adds an additional up-scaling branch to the neural network, which can be executed in parallel with the main branch. The up-scaling branch is comprised of a nearest convolution operation, which is an untrainable frozen convolutional layer, with linear activation, whose kernel is equal to the identity matrix, thus interpolating towards the HR image dimension. The branches are intersecting through an addition operation, therefore forcing the main branch to learn the residual between the interpolation branch and the HR image, resulting in quicker convergence. The NCNet model does not suffer heavy performance loss after quantization, and, due to its nature, it is compatible with most mobile AI accelerators, making it suitable for running on mobile devices, without major loss in efficiency and accuracy.

2.4 Feature Distillation Networks

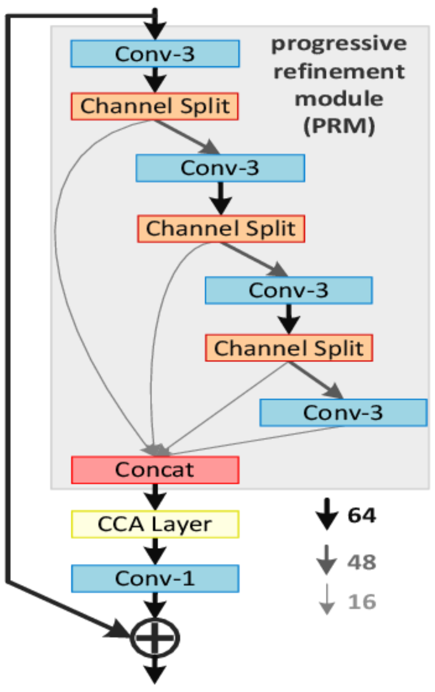


(a) IMDN

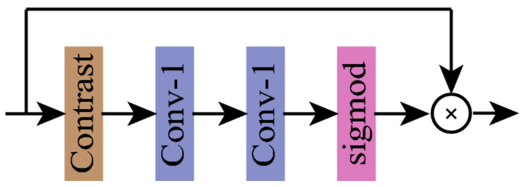


(b) Upsampler

Figure 4: The architecture for the IMDN model [16], which makes use of information multi-distillation blocks (IMDB) in order to extract features hierarchically, using channel splits.



(a) The information multi-distillation block (IMDB), which extracts features hierarchically through multiple channel splits. After concatenating the features, the contrast-aware channel attention layer (CCA) [48] computes the importance of the selected features.



(b) The contrast-aware channel attention layer (CCA) [48], used for computing the importance of the selected features.

Figure 5: The information multi-distillation block (IMDB) [16] architecture.

Another successful iteration over the sub-pixel convolutional neural architecture [34] for super-resolution was designed using feature distillation networks. Figure 4 illustrates the IMDN model [16], which represents a shallow variant of such architecture. IMDN makes use of information multi-distillation blocks (IMDB), which extract features hierarchically, through multiple channel splits, as shown in Figure 5. After concatenating the features, IMDB introduces a contrast-aware channel attention layer (CCA) [48], which computes the importance of the selected features.

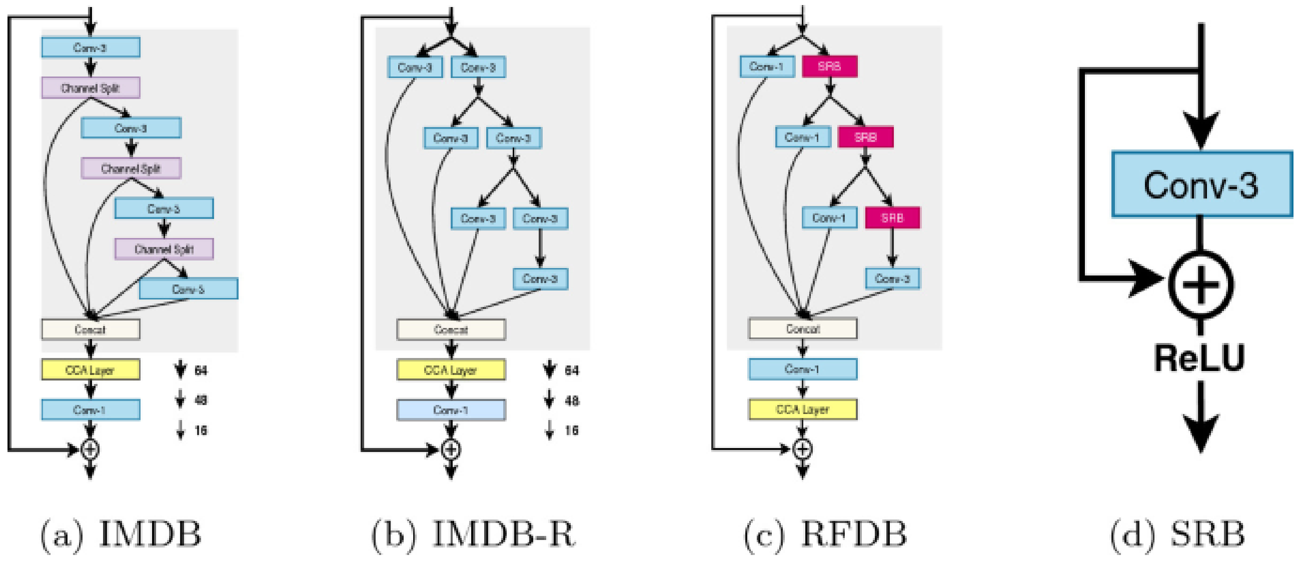


Figure 6: The enhancement brought to IMDN, through the RFDN architecture, using the shallow residual block (SRB) as a replacement for the convolutional layers in the IMDB block, so as to learn the discriminative feature representations.

The Residual Feature Distillation Network (RFDN) [25] is a variation of the IMDN architecture, that replaces the outer convolutional layers in the information multi-distillation block with a shallow residual block (SRB), in order to learn the discriminative feature representations, as shown in Figure 6.

2.5 Attention-Based Models

The adoption of attention mechanisms in the previous decade led to severe improvements in the performance of neural networks [4], enabling the models to focus on specific parts of the input, by computing an attention vector, which indicates what input sections are more relevant to the given context, thus being able to retain information about long-range dependencies. This method proved to be effective in natural language processing tasks, being the baseline for transformer networks, which use self-attention layers [41]. Furthermore, visual attention models were used successfully in convolutional networks [43].

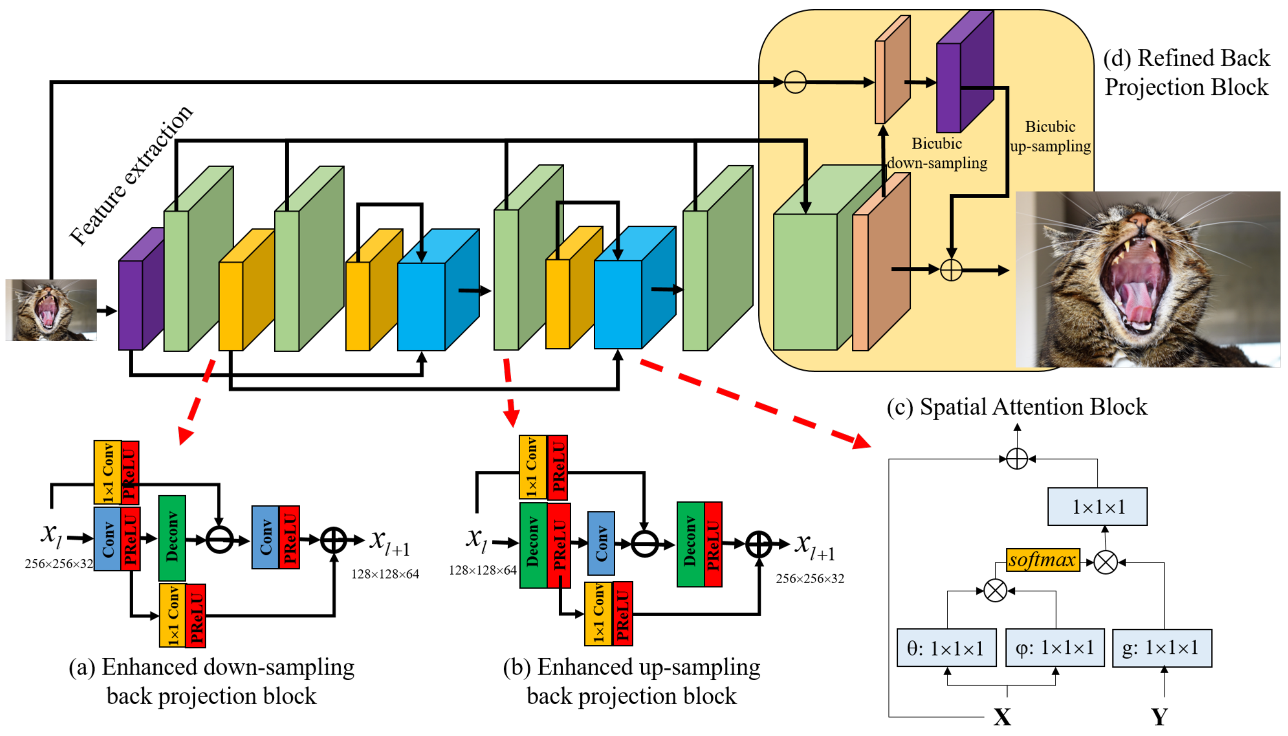

Feature extraction
(d) Refined Back Projection Block
Bicubic down-sampling
Bicubic up-sampling
(c) Spatial Attention Block
1×1×1
softmax
θ: 1×1×1
φ: 1×1×1
g: 1×1×1
X
Y
1×1 Conv
PReLU
Conv
Deconv
x_l
x_{l+1}
256×256×32
128×128×64
(a) Enhanced down-sampling back projection block
(b) Enhanced up-sampling back projection block

Figure 7: The ABPN model [26], enhancing the back projection network architecture [14], by introducing self-attention layers in the feature extraction phase and in the proposed refined back projection block. Additionally, spatial attention is used for computing cross-correlation for feature maps extracted at different depths.

The integration of attention in back projection [14] networks presented state-of-the-art performance for the super-resolution task, even with shallow architectures. As shown in Figure 7, ABPN [26] makes use of the self-attention layer and proposes a modified version, the spatial attention block. Self-attention appears once in the feature extraction phase, before the back projection block, so as to identify relevant relationships in the LR image. Its second occurrence is in the refined back projection block, after stacking the LR image and the down-scaled form of the estimated SR image, which is generated by the back projection block. The spatial attention block computes cross-correlation for feature maps extracted at different depths, throughout the back projection block.

2.6 U-Net Architectures

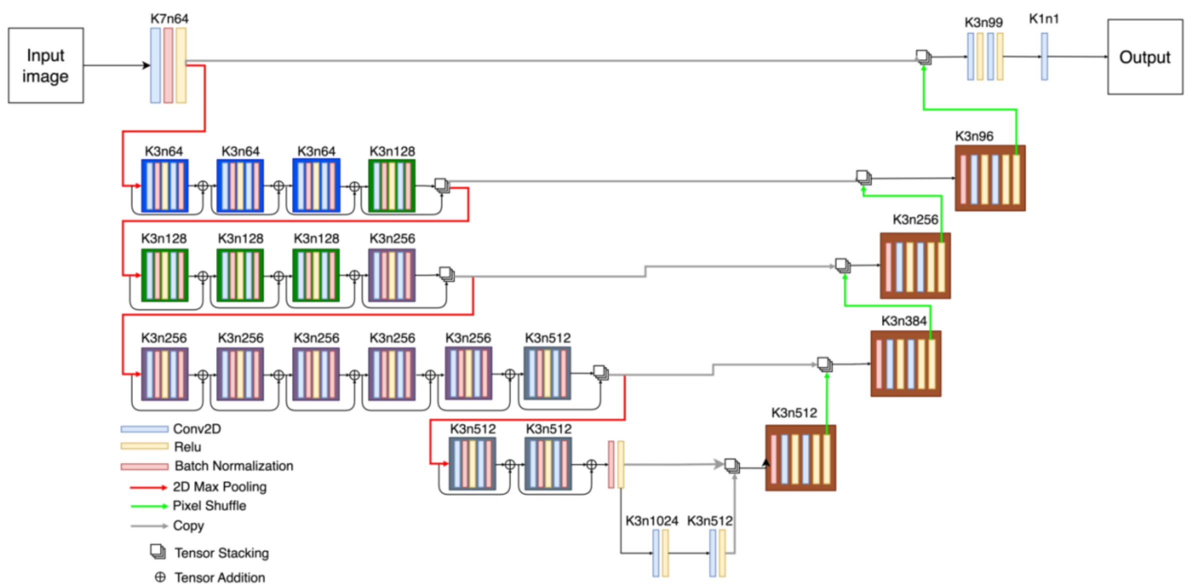


Figure 8: The RUNet architecture [15] [32], a variation of the U-Net model [31], which adds residual connections on the down-scaling stage, in order to enable the model to learn more complex features. Instead of up-scaling through deconvolution layers, the architecture makes use of pixel shuffle layers (depth-to-space layers).

Another neural architecture for performing image super-resolution is the U-Net [31], which recently became popular through the development of latent diffusion models [30]. One proposed variation for this architecture is the RUNet [15] [32]. This network operates on a bicubically up-scaled LR image. As shown in Figure 8, the feature map dimension is sequentially decreased through pooling layers, while the channel count increases over each downstream level. This particular stage takes the role of a feature extraction block. Unlike the original U-Net architecture [31], the convolutional sequences add residual connections, in order for the network to learn more complex features. In the second phase, the network performs feature map up-scaling through pixel shuffle layers (depth-to-space layers), while decreasing the channel count. Before each convolution operation in the second stage, the tensors are stacked with the features that were previously refined in the downward block, on the same level.

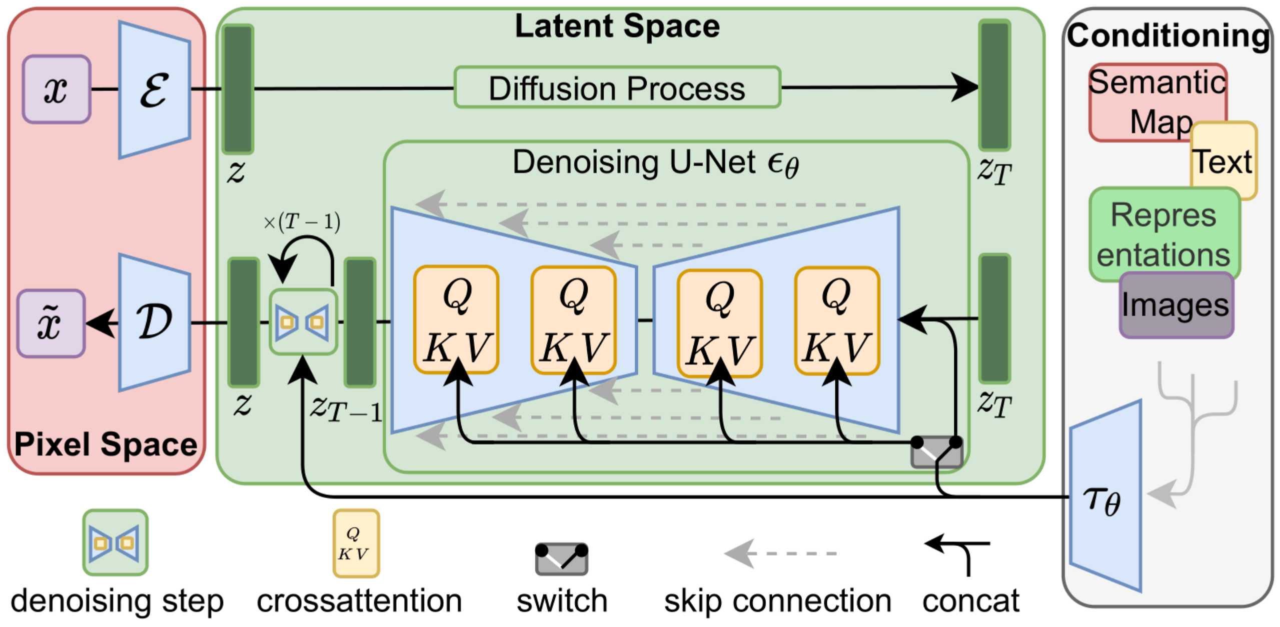


Figure 9: The LDM architecture [30], which maps the input features into a latent space through the diffusion process. Given the latent representation, the network generates an image and performs the denoising step, using an U-Net augmented with cross-attention.

Recent efforts in generative models development led to the design of latent diffusion networks [30], which presented great performance for various generative tasks, such as conditional image generation, inpainting, and super-resolution. Figure 9 illustrates the model, which leverages the transformer [41] and auto-encoder architecture [21]. The input features are mapped into a latent space through the diffusion process. Given the latent representation, the network generates an image and performs the denoising step, using an U-Net augmented with cross-attention.

2.7 Generative Adversarial Networks

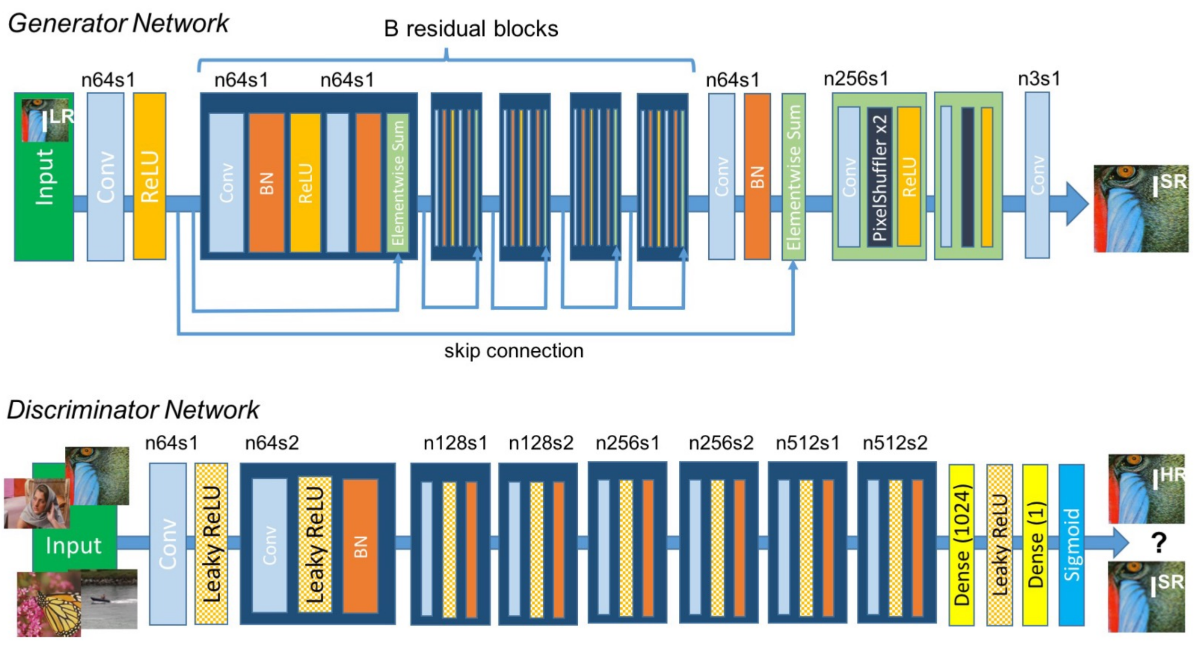


Figure 10: The architecture for the generator and discriminator network of SRGAN [22]. The generator uses a residual convolutional network architecture, while the discriminator is a convolutional model. Although SRGAN achieves below state-of-the-art performance in terms of peak signal-to-noise ratio, the results are photo-realistic, optimizing for visual perception.

Generative adversarial models [13] (GANs) were successfully applied to the task of generating photo-realistic SR images, optimizing for visual perception, beyond the results achievable by maximizing the structural similarity index measure [42] (SSIM). SRGAN [22] is an example of such model, leveraging the traditional GAN architecture. As displayed in Figure 10, the model is comprised of two separate convolutional neural networks, the generator and the discriminator, the second one being used only during the training process. The discriminator learns to distinguish between real samples and samples that were produced by the generator. In parallel, the generator learns to produce samples that are photo-realistic, in the case of super-resolution, so as to confuse the discriminator into believing that the generated images are actually real. Although SRGAN achieves below state-of-the-art performance in terms of peak signal-to-noise ratio, the results are photo-realistic, optimizing for visual perception.

Chapter 3. Explored Methods

This chapter first goes over the baseline models, specified in Section 3.1 that were considered for the experiments, then describes the architecture for the proposed networks, during Section 3.2. The design decisions that were regarded when building the models will be further motivated in Chapter 4.

The baselines that were used as a starting point for the exploration process were selected based on both performance and efficiency criteria, opting for shallow and swift architectures, so as to enable the addition of supplementary components, while also abiding by the inference time constraints, ensuring real-time execution on computer GPU's, and below 2 seconds inference time for mobile processors.

3.1 Baseline Models

The baseline models used as a starting point for the experiments represent variations of sub-pixel convolutional neural networks, given the shallow and efficient design of such architectures, achieving good performance and inference time in the context of constrained devices.

Section 3.1.1 describes the considered ESPCN architecture [34], which was used during the first stage of the experiments. Then, Section 3.1.2 goes over the chosen variant for the SC-SRN model [11], which expands the original sub-pixel convolutional architecture, introducing residual learning through concatenation-based skip connections, and a reparameterizable block, meant to increase the model's effectiveness through over-parameterization.

3.1.1 Efficient Sub-Pixel Convolutional Neural Network

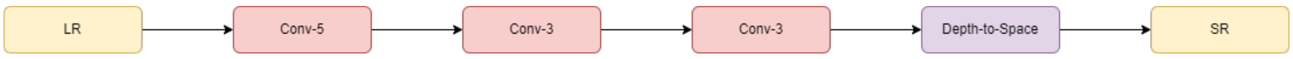


Figure 11: The architecture for the ESPCN [34], in the form that it was implemented during the report experiments. The model makes use of a decreasing kernel size and filter count in the feature extraction section. The reconstruction block is comprised of a convolution with a channel count equal to the squared upscaling factor multiplied by the target channel count, so that the feature maps can be transformed into the high-resolution image through the depth-to-space layer. The implementation utilized as a baseline uses hyperbolic tangent activation exclusively, and the weights for the convolutional layers are scaled during the initialization process using a random normal distribution of mean 0 and standard deviation 0.001.

The first considered baseline model is the Efficient Sub-Pixel Convolutional Neural Network (ESPCN) [34], which employs the sub-pixel convolutional layer to perform depthwise transposition, thus preserving a feature map dimension equal to the low-resolution image dimension, while adding complexity through increased depth. As presented in Section 2.3, ESPCN allows for increased parallelism for the convolutional operations, since GPU's can perform operations for different channels concurrently.

Figure 11 depicts the ESPCN architecture, which sequentially extracts features through non-linearly activated convolutional layers, using a decreasing kernel size and filter count in the feature extraction section. The second part of the network is comprised of a convolution with a channel count equal to the squared upscaling factor multiplied by the target channel count, so that the feature maps can be transformed into the high-resolution image through the depth-to-space layer, as pictured in Figure 2.

The Efficient Sub-Pixel Convolutional Neural Network architecture represents the starting point for all the performed experiments, as further expanded in Chapter 4.

3.1.2 Skip-Concatenated Super-Resolution Network

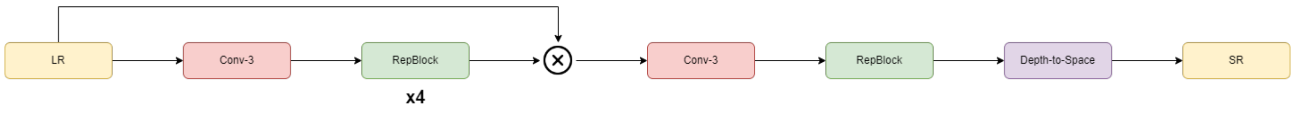


Figure 12: The Skip-Concatenated Super-Resolution Network (SCSRN) [11], which expands the ESPCN architecture [34] by adding a skip layer that concatenates the low-resolution image with the intermediary feature maps, as a more efficient way of propagating the residual, providing better inference time compared to an addition operation. As depicted in Figure 13, the network makes use of reparameterized blocks for improving performance through over-parameterization.

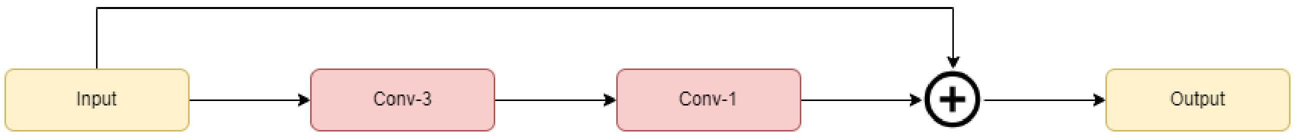


Figure 13: The reparameterized block architectures as utilized in the original Skip-Concatenated Super-Resolution Network (SCSRN) [11], using reparameterized convolutions, which act as two separate convolutional layers with kernel sizes 3 and 1, during the training stage. After training the network, the convolutional layers are merged into a single one, improving the block's execution time during inference. The implementation used throughout the report's experiments does not use reparameterization, opting to use the same network architecture during both training and inference, for simplicity.

Figure 12 portrays the baseline used for the second stage of the experiments, which is a modified version of the Skip-Concatenated Super-Resolution Network (SCSRN) [11], that extends the aforementioned sub-pixel convolutional architecture by adding a skip layer that concatenates the low-resolution image with the intermediary feature maps, as a more efficient way of propagating the residual, providing better inference time compared to an addition operation. In order to improve network performance through over-parameterization, while also maintaining efficiency, the authors propose the usage of a reparameterized block, which makes use of reparameterized convolutions, as shown in Figure 13. For simplicity, the model considered in the experiments utilizes simple convolution operations.

3.2 Proposed Models

The following sections are describing the architectures that were tested during the experimentation phase, and compared to the baselines presented in Section 3.1. As mentioned previously, the goals were to sequentially optimize for peak signal-to-noise ratio [17], then for the learned perceptual image patch similarity score [46], while also considering the structural similarity index measure [42], when designing the models. Additionally, one constraint that was considered when building the networks was maintaining real-time execution on computer GPU's, and below 2 seconds inference time on mobile processors.

The first iteration, UF-ESPCN, which is described in Section 3.2.1, enhances the ESPCN architecture [34] by using a fixed-size kernel and a constant number of feature maps for all convolutional layers, except for the last one, which is supposed to scale the number of channels in a manner that serves the pixel shuffle layer with the appropriate dimensions.

Then, Section 3.2.2 describes an alternative for the U-Net architecture [31], called FR-UNET, which removes the pooling and deconvolutional layers, to maintain a constant feature map dimension, and ensure that the low-resolution image can employ any size.

Section 3.2.3 and Section 3.2.4 describe two proposed networks that enforce residual learning for the baseline SCSRN model [11], through the addition of a nearest-neighbors upscaling block, followed by a depth-to-space, attempting to improve convergence speed and performance, without impacting the inference time, maintaining it through parallelization.

Lastly, Section 3.2.5 presents the final architecture decisions, which were tested on both the ESPCN and the SCSRN. The discussed model repositions the upscaling stage at the beginning of the main branch, so as to further improve performance by providing the principal stages of the network with an upscaled baseline. The network maintains the residual learning principle by continuing to enforce residual learning.

3.2.1 Uniform-Filter ESPCN

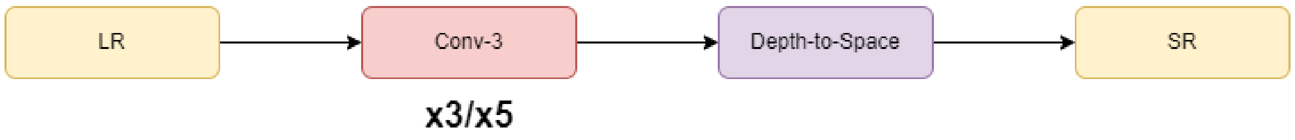


Figure 14: The UF-ESPCN model, which was tested in two variants, having 3 and, respectively, 5 convolutional layers. The architecture modifies the original ESPCN [34], by switching to ReLU activations, reversing the order for the clipping and depth-to-space layers, initializing convolutions using the Glorot normal distribution [12], and maintaining a constant number of 32 channels and a fixed kernel size of 3.

The first iteration implemented in the experimentation phase consists of enhancing the existing ESPCN architecture [34] described in Section 3.1.1, by switching to ReLU activations, to prevent the vanishing gradient issue, which occurs in sigmoidal activations, including the hyperbolic tangent, slowing down the network's convergence speed [28]. The original ESPCN implementation clips the output after super-resolving the image, to ensure that the output data is in the required range. However, reversing the order of the depth-to-space and clip layers caused an increase in inference speed. Another improvement consists of replacing the random normal weight initialization with the Glorot normal distribution [12], which ensures that the variation for the activations remains relatively constant across the network, during the training process. Finally, the convolutions were all set to a fixed number of 32 channels and a kernel size of 3, ensuring a consistent information flow, while also improving speed by reducing the number of floating-point operations. Given this property, the network was suggestively named Uniform-Filter ESPCN (UF-ESPCN). Two variants were tested, having 3 and, respectively, 5 convolutional layers, as shown in Figure 14.

3.2.2 Full-Resolution U-Net

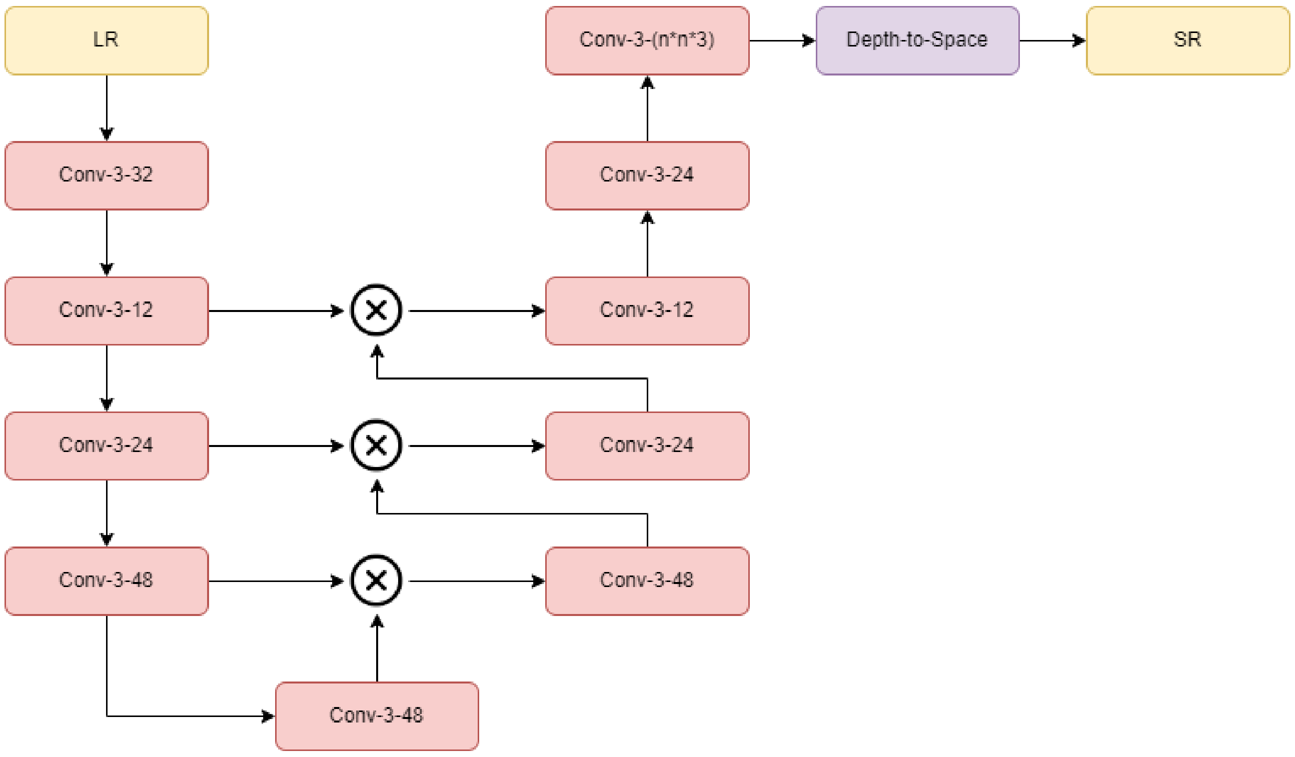


Figure 15: The architecture for the FR-UNET model, which adapts the original U-Net architecture [31], removing the feature map down-scaling and up-scaling operations, which, in the aforementioned network, were obtained using pooling and inverse convolutional layers. This is required in order to provide the model with an input size-agnostic characteristic.

The Full-Resolution U-Net (FR-UNET) represents an iteration over the original U-Net architecture [31] described in Section 2.6, which does not down-scale the feature maps using pooling layers, nor does it perform up-scaling through inverse convolution operations, as it can be observed in Figure 15. This was required to ensure that the network can super-resolve images of variable dimensions, since performing deconvolution after pooling does not ensure reconstruction to the original size, due to uneven division, which wouldn't allow concatenating feature maps from the down-scaling block with those from the up-scaling block. The feature extraction stage of the U-Net uses a 32-channel convolution, followed by a series of convolutional layers with the filter count increasing from 12 to 48, by a factor of two. The reconstruction phase is designed to be symmetric to the first stage, to enable the usage of concatenation. The last two convolutional layers are designed to bring the channels count up to the number that allows the pixel shuffle operation to reconstruct the final high-resolution image.

3.2.3 Residual SCSRN

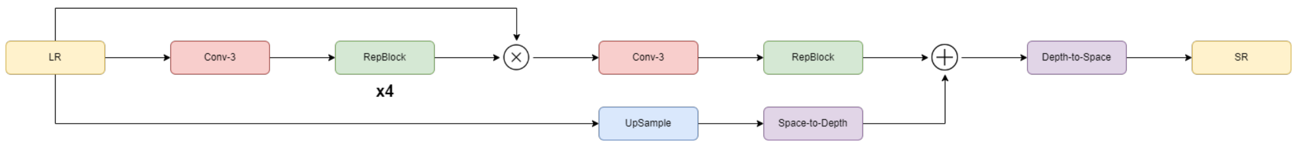


Figure 16: The R-SCSRN architecture, which extends the SCSRN model [11] by adding a nearest-neighbors up-scaling branch, so as to facilitate the network convergence speed, and its performance in terms of the optimized metric. The secondary branch performs a space-to-depth operation before being appended to the main stage, so as to ensure that the summed-up value is clipped through the clip operation that occurs before the pixel shuffle.

As depicted in Figure 16, the Residual SCSRN (R-SCSRN) network develops on top of the previous SCSRN architecture [11] detailed in Section 3.1.2, enforcing the model to learn the residual between the elementary nearest neighbors interpolation for the low-resolution image, and the target high-resolution image, aiming to speed up the network's convergence and, potentially, it's performance. The addition of an up-scaling branch does not introduce any new trainable parameters and can be executed concurrently with the main SCSRN branch, the two parts being independent of each other until they are summed-up, before performing the pixel shuffle operation. The last layer on the up-scaling branch is a space-to-depth layer, the inverse of the depth-to-space operation, which transposes pixels from a feature map-level to a depth level, based on the specified factor. This was used in order to clip the entire summed-up value, on the specified range, using the clip layer which occurs before the pixel shuffle.

3.2.4 Filtered-Residual SCSRN

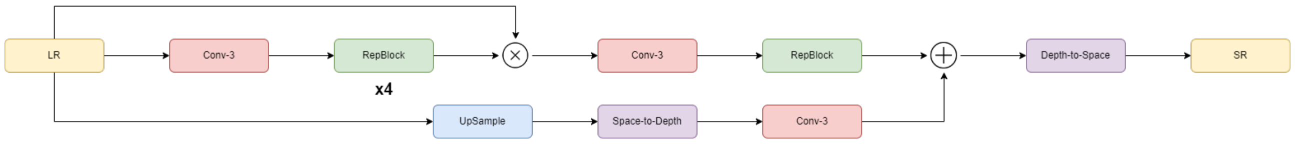


Figure 17: The FR-SCSRN architecture, which extends the R-SCSRN model by introducing an additional convolutional layer on the nearest-neighbors interpolation branch, so as to learn additional non-linear data mappings. Given the fact that the two branches can be executed concurrently, this improvement would lead to better performance, while maintaining efficiency in terms of the network's inference time.

The Filtered-Residual SCSRN (FR-SCSRN), which is pictured in Figure 17, iterates on the aforementioned R-SCSRN model discussed in Section 3.2.3, performing additional filtering on the secondary up-scaling branch, with the use of a convolutional layer which is supposed to refine the upscaled LR image features, so as to learn additional non-linear data mappings, increasing the network's goodness of fit. While this would increase the inference time for the naive interpolation stage, the overall efficiency for the proposed architecture would not suffer any significant decline, given the fact that the nearest-neighbors operation is not costly, and the main branch uses 7 times more convolutional layers. Given the high parallelism capabilities of modern GPU's, the two network stages would be able to be executed concurrently, resulting in no impact in terms of the total inference time.

3.2.5 Depthwise-Residual ESPCN & SCSRN

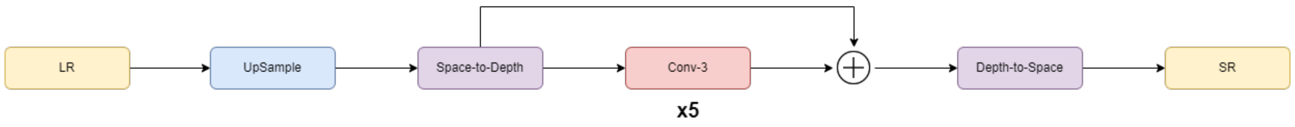


Figure 18: The DR-ESPCN architecture, which extends the original ESPCN model [34] by adding a residual learning component, similar to the one described in Section 3.2.3. Furthermore, the nearest-neighbors interpolated low-resolution image is used as a baseline for the main ESPCN stage, in an attempt to further improve convergence speed and performance, while trading off inference time.

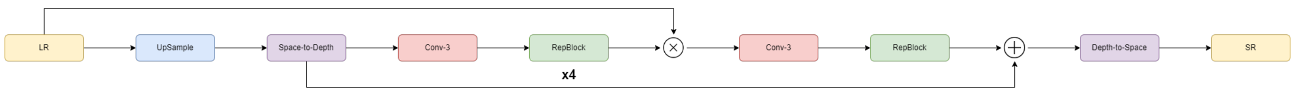


Figure 19: The DR-SCSRN model, that shifts the naive upscaling block defined in Section 3.2.3, and uses it as a foundation for the SCSRN branch, so as to ensure quicker convergence and a better goodness of fit for the network.

The Depthwise-Residual ESPCN (DR-ESPCN), represented in Figure 18 and the Depthwise-Residual SCSRN (DR-SCSRN), shown in Figure 19 represent the final experimental iterations explored in the report. These models enhance their original associated residual architecture with the depthwise-residual block, which is obtained by transferring the nearest-neighbors interpolation component to the main branch of the network. The purpose of this architectural shift is to provide the principal stage of the network with a baseline, which is then going to be refined, thus further enhancing performance and convergence speed, while trading off efficiency, to a certain extent. This architecture maintains the residual learning principle introduced in Section 3.2.3, re-adding the naively up-scaled low-resolution image to the main branch, before the clipping layer.

Chapter 4. Methodology and Experiments

This chapter will first go into the methodology used throughout the experiments, for the task of single image super-resolution, throughout Section 4.1, presenting the dataset used during both training and evaluation, the combination of hyperparameters, the test machines configuration, and the metrics that were used either as loss functions, or as indicators for tweaking the models. Then, Section 4.2 describes the iterative architectural decision process for each of the networks that were previously defined in Section 3.2, starting from the baselines mentioned in Section 3.1. Finally, Section 4.3 goes over the results yielded by the proposed models, for all upscaling factors, and motivates the choice for ultimately labeling DR-SCSRN as the best-performing model out of the ones that were benchmarked.

4.1 Methodology

This section presents the methodology used during the experimentation process, providing information regarding the dataset, the processing and augmentation pipeline, the hyperparameters used during training, and the experimental compute environment used for evaluating the models.

4.1.1 Dataset

Given the complexity of the tackled super-resolution problem, and the fact that deep learning requires large amounts of data in order for the model to achieve decent goodness of fit while generalizing the target distribution, the dataset of choice had to meet certain requirements, in terms of size and diversity. Consequently, the networks were trained on the DIV2K dataset [1], which consists of 800 training images, 100 validation images, and 100 test images, all of which have a 2K resolution. All images in DIV2K follow the RGB channel format. Only the training and validation sets are publicly available. Therefore, during the experiments, the first half of the validation dataset was used for validating the models, while the second half served as a test dataset. The networks for all the treated upscaling factors (x2, x3, x4) were trained and validated using the NTIRE 2017 guidelines [38], using the proposed bicubically downscaled training images. However, in addition to that, for x4 upscaling, the models were also trained using the NTIRE 2018 methodology [39], thus using the realistic mild training and validation sets, whose low resolution images present motion blur and Poisson noise that is image dependent, introducing the possibility of diverse pixel shifts.

4.1.2 Data Processing and Augmentation Pipeline

The dataset images were normalized, by scaling them in the 0-1 range, and the training data was stored in the form of memory maps, due to the limitations of the compute environment, which was limited to 14GB RAM. Memory maps are meant to store arrays on disk, only accessing the required chunks on demand, allowing the use of a bigger dataset. The training low-resolution images were split into disjoint 128x128 patches, and their high-resolution counterparts were split into patches of appropriate dimensions, based on the upscaling factor. During optimization, on each epoch, the training batches are randomly shuffled, and transformations are performed on each contained image pair, ensuring better generalization through data augmentation, by randomly flipping them horizontally and vertically, and arbitrarily rotating them clockwise or counterclockwise for a maximum of 0.2 radians.

4.1.3 Hyperparameters

The models were initially trained for 70 epochs, optimizing for L2 loss, and, consequently, for peak signal-to-noise ratio [17]. While PSNR is commonly used to measure image quality by assessing the distortion level through pixel-wise distance, it does not account for perceptual metrics, such as texture difference. Thus, for each upscaling factor, the best proposed model is fine-tuned for 30 epochs on learned perceptual image patch similarity score [46], so as to achieve more realistic results. LPIPS computes the distance between the activations for each tested image, as inferred through a visually specialized network, previously trained on ImageNet [7]. Therefore, it leverages the extracted features to compare two images, from a perceptual point of view. The network used for initializing LPIPS during the experiments was VGG-lin, which contains a frozen pre-trained VGG stub [35], augmented with a series of layers. The VGG-lin was formerly fine-tuned on Berkeley-Adobe Perceptual Patch Similarity Dataset (BAPPS). In addition to the aforementioned optimization metrics, during the design iteration process, both the inference time and the structural similarity score [42] were considered. The optimizer used during training was Adam [20], initialized with a 0.0003 learning rate. The images were grouped into batches of size 16.

4.1.4 Experimental Environment

The network runtime was computed both on NVIDIA T4 GPU's (16GB, 65.13 TFLOPS), and on mobile Qualcomm Adreno 642L GPU's (4GB, 1.65 TFLOPS), using the default quantization, on FP16 inference.

4.2 Ablation Studies

The following sections go over the outcomes achieved for each iteration that was performed during the experimenting process, comparing each model with the previously considered architecture, considering the previously stated metrics.

4.2.1 Uniforming the Convolutional Filters of ESPCN

The ESPCN network [34] described in Section 3.1.1 was the starting point for the first experimentation stage, given the light nature of the model, which facilitates the addition of supplementary components, since the inference time reported for x4 super-resolution was approximately 5 milliseconds for computer GPU's and 60 milliseconds for mobile processors. Thus, the first iteration was an extension of the ESPCN, called UF-ESPCN, which, as described in Section 3.2.1, switched to ReLU activations, to prevent the vanishing gradient issue, which occurs in sigmoidal activations. Then, the order for the clip and pixel shuffle layers was reversed, so as to optimize the clipping speed through parallelism. Finally, Glorot normal initialization was used for the convolutional weights, and their channels and kernel sizes were set to a fixed 32 and 3x3 size, therefore reducing the number of floating point operations. Strictly for x4 super-resolution, the FLOP count was reduced by 35%, due to a reduction in trainable parameters, from 37200 to 24016. The computer GPU runtime was diminished by 70%, and the mobile processor inference time dropped by 45%. Simultaneously, the PSNR suffered a significant increase of 3.18 decibels for normal x4 upscaling and 0.56 decibels for realistic x4 SR. The extended UF-ESPCN-5 architecture which added two additional convolutional layers did not bring any further benefit, resulting in a similar PSNR, while increasing the computer and mobile inference time by 16% and, respectively, 60%. The conclusion is that the UF-ESPCN network's performance does not scale with depth for the given task, suggesting that the next models should adopt a different architecture.

4.2.2 Maintaining a Constant Filter Size for U-Net

The next attempt represented the U-Net [31] variant depicted in Section 3.2.2, called FR-UNET, which does not use pooling or deconvolutional layers, so as to ensure that the network is invariant to the input image dimensions, and does not manipulate the feature map width or height, allowing for later concatenation. However, this implementation did not improve the current results. The PSNR decreased by 0.69 for x4 super-resolution and maintained a similar value for realistic x4 SR. When compared to the UF-ESPCN-5, the computer GPU runtime, and mobile processor inference time increased by 357% and 218%. For this reason, the idea of further expanding the U-Net architecture was dropped.

4.2.3 Adding a Nearest-Neighbors Upsampling Branch to SCSRN

The next iteration performed implied building on top of the SCSRN model [11] described in Section 3.1.2, by adding a nearest-neighbors interpolation branch, in order to enforce residual learning. As shown in Section 3.2.3, following the upsampling layer, the R-SCSRN performs a space-to-depth operation, transposing the spatial dimension into depth, by a factor of 4, so as to enable clipping for the entire summed-up high-resolution image, before applying the depth-to-space function. The baseline SCSRN model brought an improvement to the UF-ESPCN-5, in terms of PSNR, by 0.13 decibels for x4, 0.23 decibels for x3, and 0.32 decibels for x2. However, for realistic x4 super-resolution, SCSRN presented a decrease of 0.19 decibels. The computer GPU inference time increased by 390%, 500%, and 300%, for x4, x3, and x2 SR. The mobile processor runtime also increased, by 200%, 196%, and 265%. The runtime increase was determined by the higher network complexity and depth. Finally, relative to the SCSRN, the R-SCSRN achieved PSNR values higher than SCSRN, by 0.3 decibels, and 0.44 decibels, for factors x3 and x2, while only trading off 12% and 7% computer GPU inference time, and, respectively, 6% and 5% mobile execution time.

4.2.4 Appending a Convolution to the Upsampling Branch of R-SCSRN

Then, the FR-SCSRN model described in Section 3.2.4 extended the residual branch by adding another convolutional layer after the space-to-depth operation, in order to learn additional non-linear data mappings and refine the upscaled LR image representation. However, this proved to be detrimental to the network performance, whose PSNR dropped by 0.26 decibels for x3 SR, while also decreasing efficiency in terms of computer GPU and mobile runtime, by 5% and, respectively, 2%. Therefore, no further modifications were performed on the interpolation branch.

4.2.5 Using Rudimentary Interpolation as a Baseline for ESPCN and SCSRN

Eventually, the last iterations implied the addition of the depthwise-residual component to both the ESPCN [34] and the SCSRN [11] models, resulting in the development of the DR-ESPCN and DR-SCSRN, which are described in Section 3.2.5. The proposed depthwise-residual networks are constructed by transferring the aforementioned nearest-neighbors interpolation branch, to the main branch, maintaining the residual learning principle, while also providing the principal stage with a naively upscaled baseline, in order to enhance the convergence speed and the goodness of fit. Firstly, the DR-ESPCN performed worse than the previously defined R-SCSRN, for x3 super-resolution, in terms of PSNR, by 0.14 decibels. However, this solution was significantly quicker, improving the computer GPU and mobile processor inference times by 74% and, respectively, 52%. Finally, the DR-SCSRN provided the best performance out of all the models that were tested during the experimentation phase. For default and realistic x4 super-resolution, the network surpassed the SCSRN baseline's PSNR by 0.3 and 0.19 decibels, while presenting an increase in runtime of 20% and 18% for computer and mobile GPU's. For SR with factors x3 and x2, it exceeded R-SCSRN by 0.05 and 0.004 decibels with respect to the PSNR, while the computer GPU inference time increased by 0.1% and 12%, and the mobile GPU runtime raised by 4.7% and 1.4%. For this reason, the DR-SCSRN was further fine-tuned for learned perceptual image patch similarity (LPIPS) [46].

4.3 Results

The results were computed for the baseline and proposed methods, using the previously constructed test dataset, which consists of the last 50 images of the DIV2K validation set [1], as specified in Section 4.1. The PSNR, SSIM, and LPIPS are reported both for the original and the quantized models. The computer runtime considers the original unquantized model, inferring using the NVIDIA T4 GPU, while the mobile execution time is computed using the quantized network, with FP16 inference, on the Qualcomm Adreno 642L GPU. All timing-related metrics are measured in milliseconds, and consider the average inference time for 10 executions, super-resolving to 2K resolution.

4.3.1 Experimental Results for x4 Super-Resolution

Model	PSNR	SSIM	LPIPS	Computer GPU Runtime	Mobile GPU Runtime	Params	GFLOPS
Bicubic	26.661	0.744	0.382	N/A	N/A	N/A	N/A
ESPCN-3	23.415 / 23.439	0.674 / 0.674	0.437 / 0.436	4.99	160	37 200	9.63
UF-ESPCN-3	26.596 / 26.528	0.739 / 0.734	0.384 / 0.368	1.52	87	24 016	6.22
UF-ESPCN-5	26.514 / 26.386	0.736 / 0.729	0.385 / 0.376	1.82	140	42 512	11.01
FR-UNET	25.824 / 25.695	0.729 / 0.724	0.392 / 0.400	8.33	445	112 208	29.06
SCSRN	26.640 / 26.533	0.737 / 0.721	0.373 / 0.369	8.92	432	80 416	20.82
DR-SCSRN	**26.940** / **26.925**	**0.751** / **0.749**	0.364 / 0.359	10.7	510	93 376	24.19
DR-SCSRN (LPIPS fine-tuned)	26.295 / 26.246	0.713 / 0.713	**0.303** / **0.305**	10.7	510	93 376	24.19

Table 4.1: The experimental results for x4 super-resolution, based on bicubically downscaled LR images.

Table 4.1 shows that, for x4 super-resolution, except for DR-SCSRN, all other considered models performed worse than the bicubic upscaling baseline, from a PSNR perspective, given the experimental methodology described in Section 4.1. However, SCSRN and UF-ESPCN-3 presented results that are similar to the bicubic interpolation method. Thus, when considering both PSNR and SSIM, DR-SCSRN achieved the best performance on the test dataset, with a 26.940 db PSNR and 0.751 SSIM for the default model, which decreased to 26.925 db and 0.749, after quantization. After perceptual similarity fine-tuning, the network scored a 0.303 LPIPS, and 0.305 given its quantized version. The inference time was 10.7 milliseconds for computer GPU's, and 510 milliseconds for mobile processors.

4.3.2 Experimental Results for Realistic x4 Super-Resolution

Model	PSNR	SSIM	LPIPS	Computer GPU Runtime	Mobile GPU Runtime	Params	GFLOPS
Bicubic	17.607	0.449	0.605	N/A	N/A	N/A	N/A
ESPCN-3	18.240 / 18.240	0.500 / 0.500	0.665 / 0.665	4.99	160	37 200	9.63
UF-ESPCN-3	18.803 / 18.797	0.505 / 0.503	0.641 / 0.644	1.52	87	24 016	6.22
UF-ESPCN-5	**18.812** / **18.811**	**0.514** / **0.513**	0.648 / 0.644	1.82	140	42 512	11.01
FR-UNET	**18.818** / **18.817**	**0.510** / **0.509**	0.658 / 0.656	8.33	445	112 208	29.06
SCSRN	18.619 / 18.592	0.460 / 0.449	0.641 / 0.643	8.92	432	80 416	20.82
DR-SCSRN	**18.809** / **18.807**	0.502 / 0.500	0.637 / 0.635	10.7	510	93 376	24.19
DR-SCSRN (LPIPS fine-tuned)	18.349 / 18.341	0.437 / 0.436	**0.556** / **0.557**	10.7	510	93 376	24.19

Table 4.2: The experimental results for realistic x4 super-resolution.

As pictured in Table 4.2, for realistic x4 SR, all models performed better than the bicubic interpolation method, given the harder nature of the problem, due to the fact that deep neural networks can learn complex relationships for the provided data. UF-ESPCN-5, FR-UNET, and DR-SCSRN yielded the highest PSNR values. FR-UNET had slightly better performance when compared to the other two models, with 18.818 db PSNR, and 18.817 db after quantization. DR-SCSRN presented 18.809 db and 18.807 db. In terms of SSIM, UF-ESPCN-5 achieved 0.514 and 0.513, while FR-UNET scored 0.510 and 0.509. Finally, DR-SCSRN achieved 0.502 and 0.500. The perceptual similarity fine-tuned DR-SCSRN reached 0.556 and 0.557 LPIPS.

4.3.3 Experimental Results for x3 Super-Resolution

Model	PSNR	SSIM	LPIPS	Computer GPU Runtime	Mobile GPU Runtime	Params	GFLOPS
Bicubic	28.194	0.806	0.312	N/A	N/A	N/A	N/A
UF-ESPCN-5	28.231 / 28.218	0.809 / 0.804	0.294 / 0.292	2.05	230	36 443	16.77
SCSRN	28.464 / 28.370	0.813 / 0.805	0.298 / 0.284	12.3	682	57 988	26.68
R-SCSRN	28.764 / 28.747	**0.820** / **0.819**	0.292 / 0.284	13.8	726	57 988	26.69
FR-SCSRN	28.500 / 28.446	0.814 / 0.808	0.286 / 0.281	14.5	738	64 576	29.72
DR-ESPCN	28.625 / 28.610	0.817 / 0.816	0.300 / 0.299	3.53	346	43 355	19.96
DR-SCSRN	**28.812** / **28.766**	**0.822** / **0.819**	0.296 / 0.283	13.9	773	64 900	29.88
DR-SCSRN (LPIPS fine-tuned)	28.221 / 28.208	0.802 / 0.801	**0.245** / **0.246**	13.9	773	64 900	29.88

Table 4.3: The experimental results for x3 super-resolution, based on bicubically downscaled LR images.

Table 4.3 shows that, for x3 super-resolution, DR-SCSRN yielded the best PSNR, with 28.812 db before, and 28.766 db after quantization. As in the realistic SR case, all models managed to achieve better PSNR values, in relation to the bicubic upscaling method. In terms of SSIM, both R-SCSRN and DR-SCSRN presented similar results, and the latter scored 0.822 and 0.819. The LPIPS score for the DR-SCSRN was 0.245 and 0.246, after fine-tuning the model. The network's inference time was 13.9 milliseconds for computer GPU's, and 773 milliseconds for mobile GPU's.

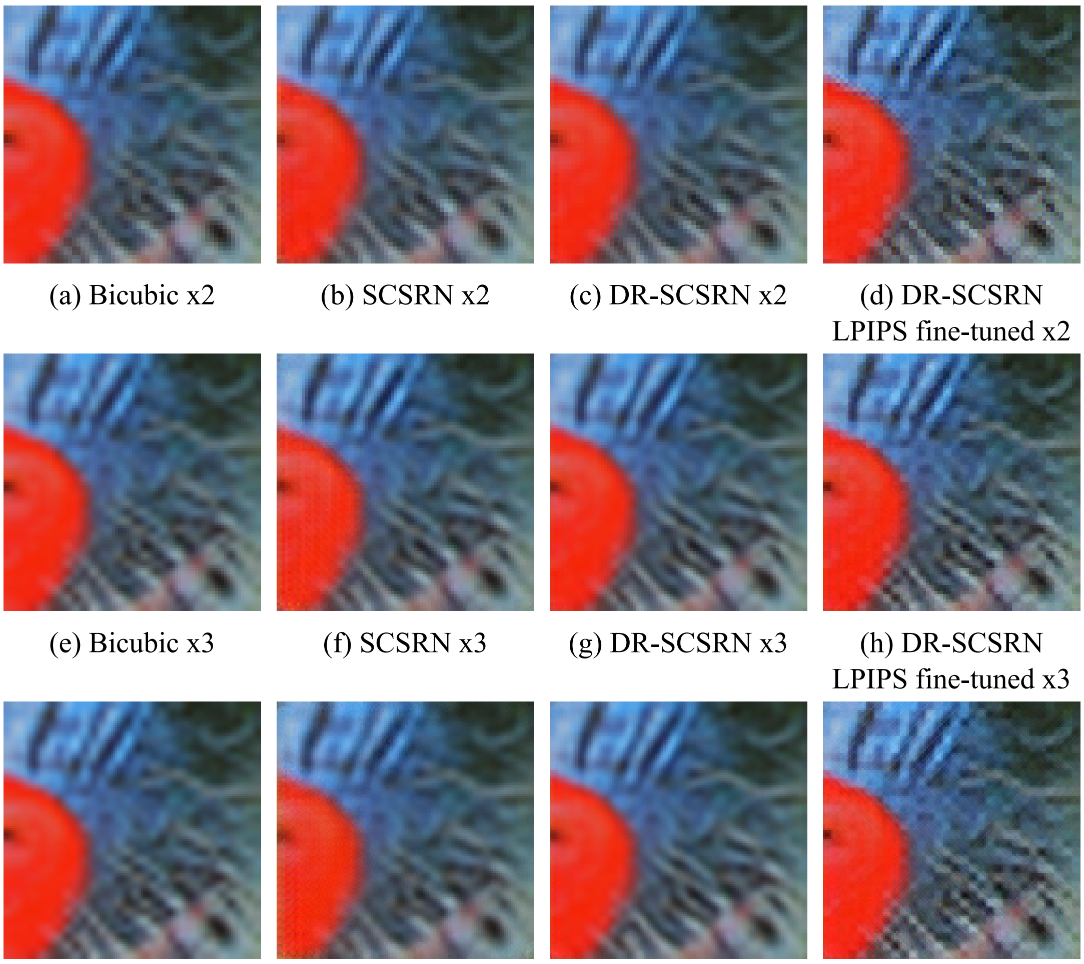
4.3.4 Experimental Results for x2 Super-Resolution

Model	PSNR	SSIM	LPIPS	Computer GPU Runtime	Mobile GPU Runtime	Params	GFLOPS
Bicubic	30.914	0.888	0.213	N/A	N/A	N/A	N/A
UF-ESPCN-5	30.586 / 30.283	0.883 / 0.869	0.219 / 0.215	6.08	471	32 108	33.23
SCSRN	30.904 / 30.804	0.885 / 0.879	0.212 / 0.203	24.4	1720	47 368	49.03
R-SCSRN	**31.343** / **31.297**	**0.893** / **0.891**	0.199 / 0.189	26.1	1805	47 368	49.03
DR-SCSRN	**31.347** / **31.279**	**0.895** / **0.892**	0.200 / 0.189	27.3	1830	49 960	51.72
DR-SCSRN (LPIPS fine-tuned)	30.148 / 30.134	0.854 / 0.853	**0.156** / **0.159**	27.3	1830	49 960	51.72

Table 4.4: The experimental results for x2 super-resolution, based on bicubically downscaled LR images.

Lastly, for the task of x2 super-resolution, only R-SCSRN and DR-SCSRN managed to surpass the bicubic method's PSNR and SSIM values, both networks achieving similar results. The PSNR for DR-SCSRN was 31.347 db and 31.279 db, and the SSIM was 0.895 and 0.892. Fine perceptual similarity fine-tuned DR-SCSRN scored 0.156 and 0.159 LPIPS. The runtime for the model was 27.3 milliseconds for computer GPU's, and 1830 milliseconds for mobile GPU's, which is below the desired upper bound hard limit of 2000 milliseconds.

4.3.5 Experimental Results Observations



(a) Bicubic x2 (b) SCSRN x2 (c) DR-SCSRN x2 (d) DR-SCSRN LPIPS fine-tuned x2

(e) Bicubic x3 (f) SCSRN x3 (g) DR-SCSRN x3 (h) DR-SCSRN LPIPS fine-tuned x3

(i) Bicubic x4 (j) SCSRN x4 (k) DR-SCSRN x4 (l) DR-SCSRN LPIPS fine-tuned x4

Figure 20: Visual comparison between an image patch upscaled using bicubic interpolation, SCSRN, and the proposed DR-SCSRN method, before and after fine-tuning the model on LPIPS.

Given the aforementioned performance measures, DR-SCSRN achieved the best results, in most cases, in terms of PSNR and SSIM, the only exception being realistic x4 super-resolution, in which case it achieved performance that is comparable to the leading model.

As expected, the networks achieved better results for lower upscaling factors, given the easier nature of the problem, having to generate fewer details for the SR image. The highest runtimes were achieved for x2 super-resolution, since the considered LR images had a larger size, while the SR images had the same sizes for all tasks. The results for realistic x4 SR were considerably worse than the ones achieved for x4 SR using bicubically downscaled LR images, since the realistic set of images contained additional artifacts, presenting motion blur and Poisson noise that is image dependent, introducing the possibility of diverse pixel shifts. As observed, the PSNR dropped after fine-tuning DR-SCSRN on LPIPS. This phenomenon is known as the perception-distortion tradeoff [5], which states that optimizing metrics that mathematically measure distortion does not yield better performance from a visual perception point of view. Hence, a lower PSNR or SSIM does not necessarily imply better perceptual similarity, and vice-versa. Although both LPIPS and SSIM represent similarity scores, SSIM is sensitive to changes in image structure and texture [42], while LPIPS is more sensitive to visual perceptual distortions, which might not be captured by SSIM.

4.3.6 Participation in the NTIRE 2023 Real-Time Super-Resolution Challenge

While participating in the NTIRE 2023 Real-Time Super-Resolution challenge for x2 and x3 super-resolution [6], DR-SCSRN finished 18th and, respectively, 16th. The dataset used during scoring was comprised of images acquired from generative models, digital art, video games, and camera photos. For x2 SR, DR-SCSRN achieved 34.07 db PSNR, and 36.86 db when considering only the luma channel. Then, for x3 SR, the proposed network attained 31.64 db when considering all channels, and 34.25 db for the luminance channel. Finally, in terms of SSIM, it achieved 0.8830 and 0.8292 for x2, and, respectively, x3 super-resolution.

Chapter 5. Conclusion

This chapter summarizes the topics and experiments that are addressed throughout the paper, which, given Section 5.1, cover the initially defined scope of the project, tackling the task of efficient single image super-resolution on mobile devices. Then, Section 5.2 mentions a set of potential directions that could be followed through future efforts of expanding the methods defines in this thesis, which faced certain limitations in terms of computing resources and dataset size.

5.1 Discussion

To summarize, this report covers the research of single image super-resolution models, using deep learning, for real-time inference in the context of computer GPU's, and for below 2 seconds runtime with regard to constrained, mobile devices. The models are benchmarked for upscaling factors x2, x3, and x4, and are initially trained to optimize for peak signal-to-noise ratio, then fine-tuned on learned perceptual image patch similarity, so as to achieve more realistic super-resolved images. The dataset employed during training and evaluation was DIV2K [1], representing a standard dataset for SISR. Given the imposed execution time restrictions, the models are designed to be as shallow as possible, opting for width over depth, since GPU's have strong parallelization capabilities, due to their large number of cores, allowing for concurrent processing.

The experiments started from a set of baseline models, namely ESPCN [34] and SCSRN [11], which previously achieved significant results on the super-resolution task, and were used as starting points for the experimental iterations. Alternatively, a U-Net-inspired implementation [31] was benchmarked against the other models.

The best performing proposed model, DR-SCSRN, which employs a depthwise-residual architecture built on top of the aforementioned SCSRN network, was used during the participation to the NTIRE 2023 Real-Time Super-Resolution challenge [6], eventually achieving competitive results, finishing 18th for x2 super-resolution, and 16th for x3 super-resolution.

5.2 Future Work

Given the hardware resource limitations that constrained the experimentation process, in terms of computation, a relevant aspect that could improve the current performance for the proposed models would be to train them for a larger number of epochs, using a larger number of GPU's. Potentially, that could lead to a much better goodness of fit from both a PSNR, and a LPIPS perspective.

Additionally, future iterations could make use of multiple training datasets, so as to achieve better generalization through a higher variety of data. While DIV2K [1] is a standard super-resolution dataset, the current trajectory in deep learning implies augmenting the existing data with synthetic artificially generated datasets, since an existing bottleneck of deep learning is insufficient data [10]. Furthermore, given its nature, the task of super-resolution allows for training networks in a fully self-supervised manner [24], which enables facile acquisition of training data, the only considerations that would need to be taken into account being the variety, in terms of the data distribution.

Finally, an alternative that could lead to better results than the one adopted in the experimentation process for this report, would be to employ generative adversarial networks, so as to optimize for visual perceptual similarity, instead of using the learned perceptual image patch similarity metric.

Bibliography

[1] Eirikur Agustsson and Radu Timofte. “Ntire 2017 challenge on single image super-resolution: Dataset and study”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 126–135.

[2] Michal Aharon, Michael Elad, and Alfred Bruckstein. “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation”. In: *IEEE Transactions on signal processing* 54.11 (2006), pp. 4311–4322.

[3] Yasin Almalioglu et al. “EndoL2H: deep super-resolution for capsule endoscopy”. In: *IEEE Transactions on Medical Imaging* 39.12 (2020), pp. 4297–4309.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[5] Yochai Blau and Tomer Michaeli. “The perception-distortion tradeoff”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6228–6237.

[6] Marcos V Conde, Eduard Zamfir, Radu Timofte, Marian-Sergiu Nistor, et al. “Efficient Deep Models for Real-Time 4K Image Super-Resolution. NTIRE 2023 Benchmark and Report”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2023.

[7] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[8] Chao Dong et al. “Image super-resolution using deep convolutional networks”. In: *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), pp. 295–307.

[9] Chao Dong, Chen Change Loy, and Xiaoou Tang. “Accelerating the super-resolution convolutional neural network”. In: *Computer Vision–ECCV 2016: 14th European Confer-*

ence, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. Springer. 2016, pp. 391–407.

[10] Adrien Gaidon, Antonio Lopez, and Florent Perronnin. "The reasonable effectiveness of synthetic visual data". In: *International Journal of Computer Vision* 126.9 (2018), pp. 899–901.

[11] Ganzorig Gankhuyag et al. "Skip-Concatenated Image Super-Resolution Network for Mobile Devices". In: *IEEE Access* (2022).

[12] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[13] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[14] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. "Deep back-projection networks for super-resolution". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1664–1673.

[15] Xiaodan Hu et al. "RUNet: A robust UNet architecture for image super-resolution". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 0–0.

[16] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. "Lightweight image super-resolution with information multi-distillation network". In: *Proceedings of the 27th acm international conference on multimedia*. 2019, pp. 2024–2032.

[17] Quan Huynh-Thu and Mohammed Ghanbari. "Scope of validity of PSNR in image/video quality assessment". In: *Electronics letters* 44.13 (2008), pp. 800–801.

[18] Andrey Ignatov et al. "Efficient and accurate quantized image super-resolution on mobile NPUs, mobile AI & AIM 2022 challenge: report". In: *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*. Springer. 2023, pp. 92–129.

[19] Mehrdad Khani, Vibhaalakshmi Sivaraman, and Mohammad Alizadeh. “Efficient video compression via content-adaptive super-resolution”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 4521–4530.

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

[21] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[22] Christian Ledig et al. “Photo-realistic single image super-resolution using a generative adversarial network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690.

[23] Yawei Li et al. “NTIRE 2022 challenge on efficient super-resolution: Methods and results”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1062–1102.

[24] Heng Liu et al. “Perception consistency ultrasound image super-resolution via self-supervised CycleGAN”. In: *Neural Computing and Applications* (2021), pp. 1–11.

[25] Jie Liu, Jie Tang, and Gangshan Wu. “Residual feature distillation network for lightweight image super-resolution”. In: *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer. 2020, pp. 41–55.

[26] Zhi-Song Liu et al. “Image super-resolution via attention based back projection networks”. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE. 2019, pp. 3517–3525.

[27] Ziwei Luo et al. “Fast nearest convolution for real-time efficient image super-resolution”. In: *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*. Springer. 2023, pp. 561–572.

[28] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. “Rectifier nonlinearities improve neural network acoustic models”. In: *Proc. icml*. Vol. 30. 1. Atlanta, Georgia, USA. 2013, p. 3.

[29] Gordon E Moore. “Cramming more components onto integrated circuits”. In: *Proceedings of the IEEE* 86.1 (1998), pp. 82–85.

[30] Robin Rombach et al. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10684–10695.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[32] Rishabh Sharma et al. “A Deep Learning Approach to Upscaling “Low-Quality” MR Images: An In Silico Comparison Study Based on the UNet Framework”. In: *Applied Sciences* 12.22 (2022), p. 11758.

[33] Jacob Shermeyer and Adam Van Etten. “The effects of super-resolution on object detection performance in satellite imagery”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 0–0.

[34] Wenzhe Shi et al. “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1874–1883.

[35] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[36] Radu Timofte, Vincent De Smet, and Luc Van Gool. “A+: Adjusted anchored neighborhood regression for fast super-resolution”. In: *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV 12*. Springer. 2015, pp. 111–126.

[37] Radu Timofte, Vincent De Smet, and Luc Van Gool. “Anchored neighborhood regression for fast example-based super-resolution”. In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 1920–1927.

[38] Radu Timofte 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.

[39] Radu Timofte et al. “NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2018.

[40] Roger Y Tsai and Thomas S Huang. “Multiframe image restoration and registration”. In: *Multiframe image restoration and registration* 1 (1984), pp. 317–339.

[41] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[43] Kelvin Xu et al. “Show, attend and tell: Neural image caption generation with visual attention”. In: *International conference on machine learning*. PMLR. 2015, pp. 2048–2057.

[44] Ling-Yi Xu and Zoran Gajic. “Improved network for face recognition based on feature super resolution method”. In: *International Journal of Automation and Computing* 18.6 (2021), pp. 915–925.

[45] Roman Zeyde, Michael Elad, and Matan Protter. “On single image scale-up using sparse-representations”. In: *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*. Springer. 2012, pp. 711–730.

[46] Richard Zhang et al. “The unreasonable effectiveness of deep features as a perceptual metric”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 586–595.

[47] Shengxiang Zhang, Gaobo Liang, Shuwan Pan, and Lixin Zheng. “A fast medical image super resolution method based on deep learning network”. In: *IEEE Access* 7 (2018), pp. 12319–12327.

[48] Wei Zhang, Zhongqiang Fan, Yan Song, and Yagang Wang. “Lightweight image super-resolution with group-convolutional feature enhanced distillation network”. In: *International Journal of Machine Learning and Cybernetics* (2023), pp. 1–16.

Appendices

A.1. Model Analysis

The following section represents a discussion regarding certain analytical aspects of the developed DR-SCSRN model, which was previously defined in Section 3.2.5.

A.1.1. DR-SCSRN Weights Distribution

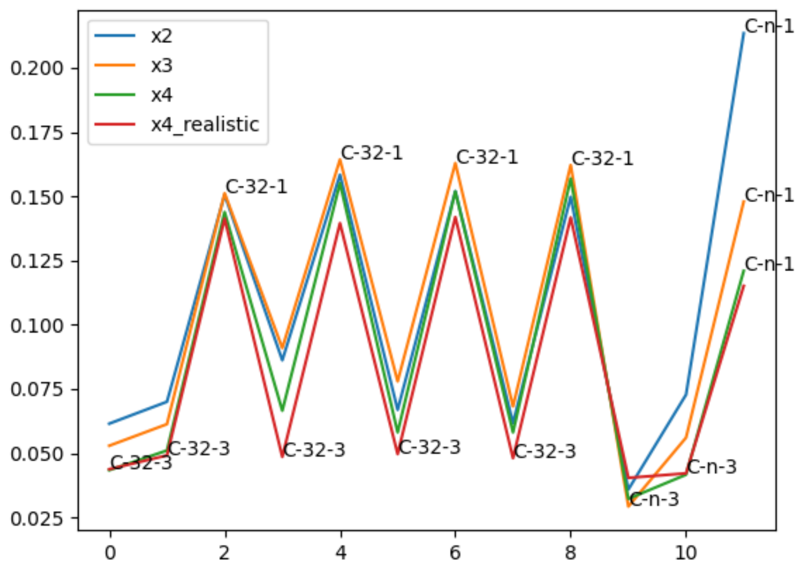
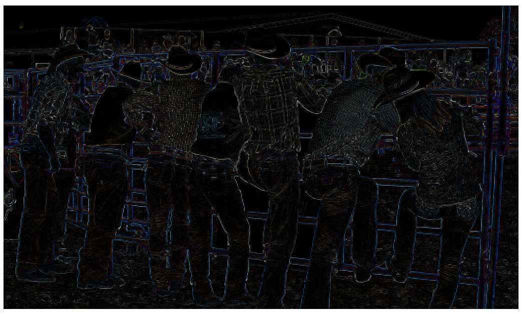


Figure 21: The distribution for the average convolutional kernel weights of the DR-SCSRN model, ordered by depth, which remains mostly similar for all upscaling factors, the only exception occurring in the last layer, which precedes the depth-to-space operation. Convolutional layers with lower kernel sizes have higher weights, indicating that the network focus on learning texture-level details, given the fact that the higher-level features are treated by the nearest neighbors interpolation operation which is summed up to the main model branch.

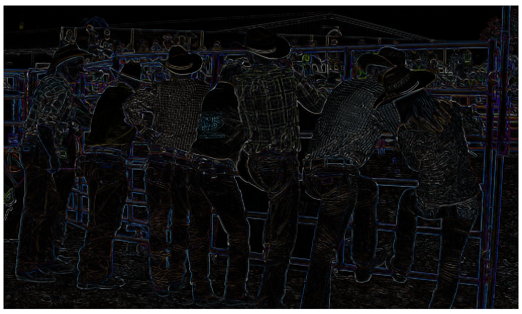
Figure 21 depicts the distribution for the averaged DR-SCSRN convolutional kernel weights, ordered by depth. The last three convolutions have different numbers of channels, each being equal to the squared upscaling factors, multiplied by a factor of 3. As expected, the distribution remains mostly similar, for all upscaling factors. However, the analysis shows that the average kernel weight values for lower upscaling factors tend to be slightly higher when compared to higher factors. Moreover, for the last layer, which occurs before the depth-to-space operation, the difference in values becomes significantly higher. Additionally, convolutions with larger kernel sizes tend to have substantially lower weights, when compared to convolutional layers with smaller kernels, proving that the network emphasizes learning lower-level features, such as edges and textures, rather than higher-level characteristics. In fact, the higher-level features are already present in the nearest neighbors interpolated image which is summed up to the main model branch output, thus enforcing the network to only learn to append texture-level details.

A.1.2. DR-SCSRN Main Branch Output

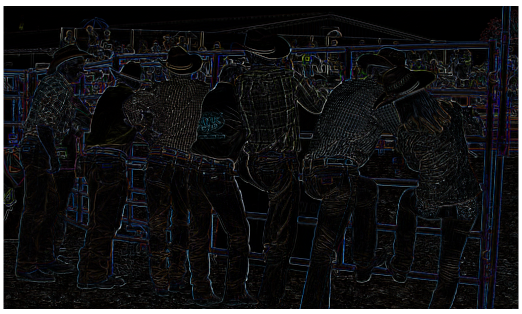
(a) The 4th image in the DIV2K validation set [1]



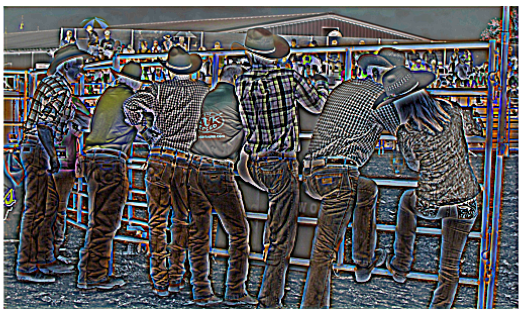
(b) SR x2



(c) SR x3



(d) SR x4

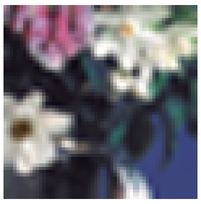


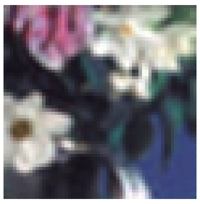
(e) SR x4 realistic

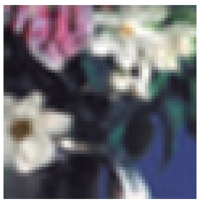
Figure 22: The delta between each DR-SCSRN model's output, and the LR image upscaled using nearest-neighbors interpolation, as inferred on the 4th image in the DIV2K validation set [1], indicated the higher-level of detail yielded by the network that was trained using the realistic x4 SR dataset. The differences were exaggerated by a factor of 5.

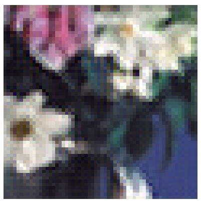
Figure 22 shows the outcome of subtracting the LR image upscaled using nearest-neighbors interpolation from the DR-SCSRN model's inferred output, for each upscaling factor. The results indicate the fact that the models trained using the bicubically downscaled LR images only enhance the LR image by adding edge and texture-level details, while the model trained on the realistic x4 SR dataset tends to add additional signal to the image, attempting to compensate for the Poisson noise and pixel shifts that occurred when generating the LR images, as per the methodology specified in Section 4.1.

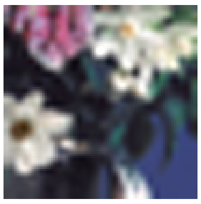
A.1.3. Model Visual Comparison

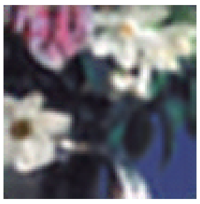

Bicubic x2

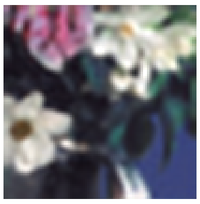

SCSRN x2

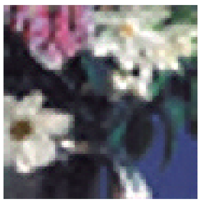

DR-SCSRN x2


DR-SCSRN LPIPS fine-tuned x2

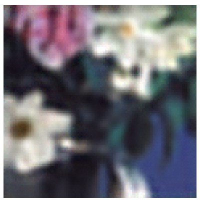

Bicubic x3

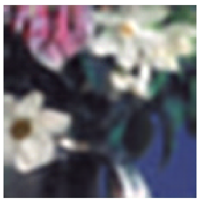

SCSRN x3

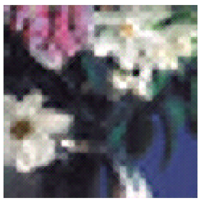

DR-SCSRN x3

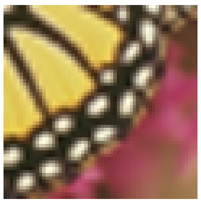

DR-SCSRN LPIPS fine-tuned x3

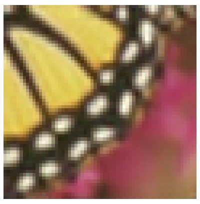
Bicubic x4

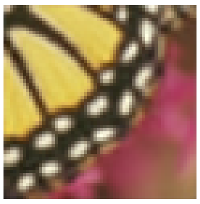

SCSRN x4

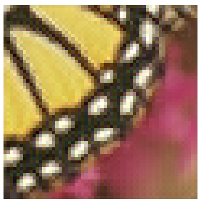

DR-SCSRN x4

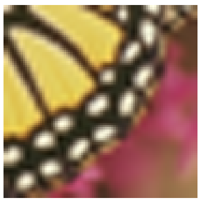

DR-SCSRN LPIPS fine-tuned x4

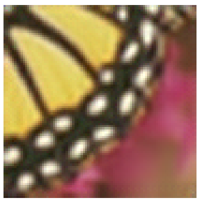

Bicubic x2

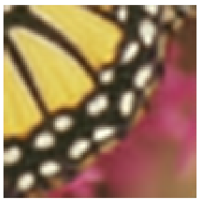

SCSRN x2

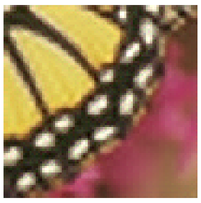

DR-SCSRN x2

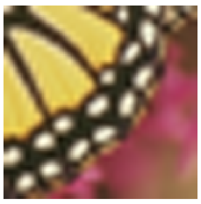

DR-SCSRN LPIPS fine-tuned x2

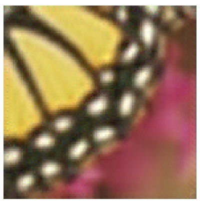

Bicubic x3

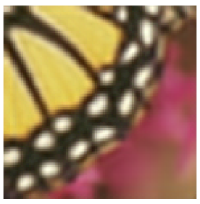

SCSRN x3

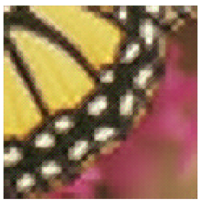

DR-SCSRN x3


DR-SCSRN LPIPS fine-tuned x3


Bicubic x4


SCSRN x4


DR-SCSRN x4


DR-SCSRN LPIPS fine-tuned x4

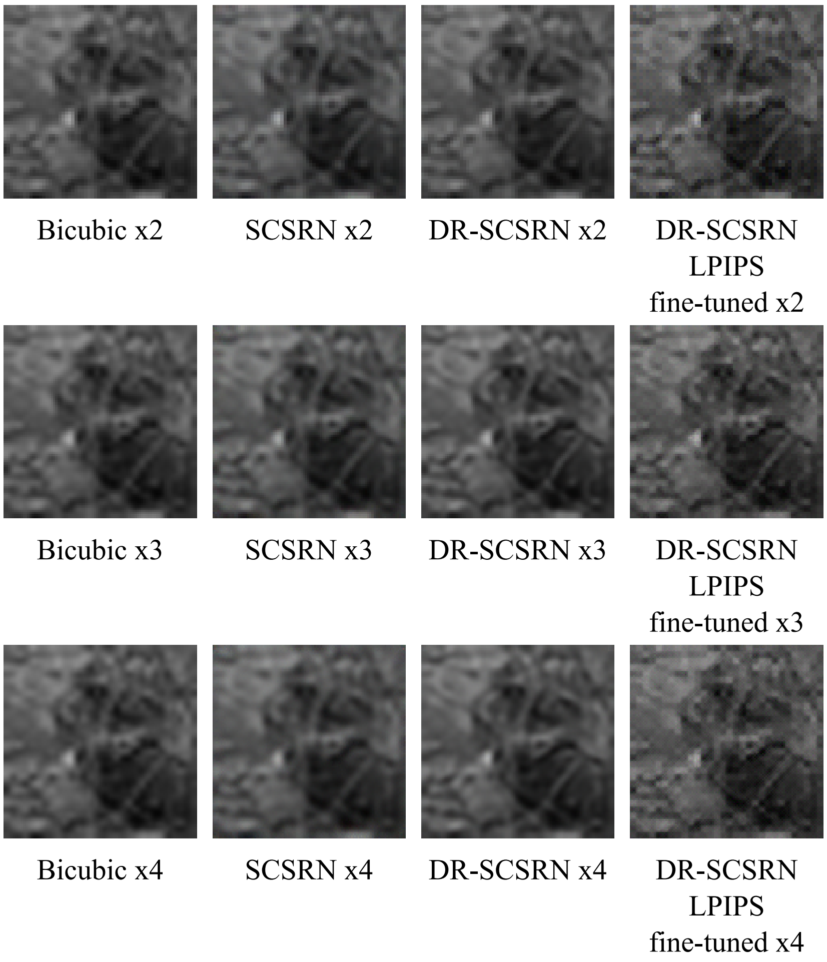


Figure 23: Visual comparison for multiple images, between patches upscaled using bicubic interpolation, SCSRN, and the proposed DR-SCSRN method, before and after fine-tuning the model on LPIPS.

A.2. Mobile Application

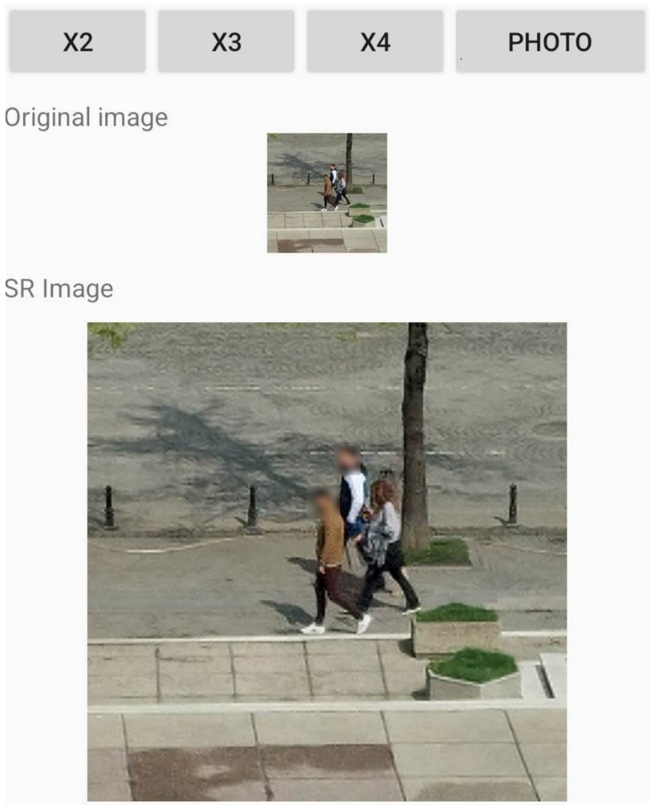


Figure 24: A screenshot of the demonstrative mobile application, which enables the user to take a picture, and super-resolve it using the specified upscaling factor. The faces of the individuals that appear in the image were blurred, ensuring anonymity, so as to avoid potential privacy issues.

Figure 24 represents a screenshot of the demonstrative mobile application, which exhibits the proposed DR-SCSRN model, by allowing the user to take a picture, and super-resolve it using the specified upscaling factor (x2, x3, or x4). The models used for inference were quantized and exported in a TFLite form, ensuring efficiency of execution. The user can choose whether the networks will run on the mobile device's GPU, or its CPU. The application was developed by modifying the existing Android example for TFLite-based SR model execution, provided by Tensorflow.