Edge Weights and Wavelet Transforms in Image Super-Resolution Networks

Chenhao Wang
Department of Computing Science
Simon Fraser University
cwa217@sfu.ca

Luciano Oliveira
Department of Computing Science
Simon Fraser University
loliveir@sfu.ca

Oleksandr Volkanov
Department of Computing Science
Simon Fraser University
avolkano@sfu.ca

Sara Pilehroudi
Department of Computing Science
Simon Fraser University
spilehro@sfu.ca

ABSTRACT

In this project, we explore the applications of edge weights, and wavelet transforms in the context of image super-resolution neural networks. We introduce modified edge-based and wavelet-based weighted loss functions and perform both qualitative and quantitative analysis on the impact of the weighted loss functions and wavelet transform coefficients for the image super-resolution task.

ACM Reference Format:
Chenhao Wang, Luciano Oliveira, Oleksandr Volkanov, and Sara Pilehroudi. 2020. Edge Weights and Wavelet Transforms in Image Super-Resolution Networks. In *Proceedings of CMPT 743.* ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

Image super-resolution (SR) is the process of recovering high-resolution (HR) images from low-resolution (LR) images, which is one of the fundamental problems in computer vision and image processing. Constructing an HR image from its LR representation is considered ill-posed, as there is no unique mapping between an LR and an HR image. In recent years, image super-resolution has proven to be useful in various fields. Medical [1] [2], satellite and microscopy image processing, astronomical studies [2], surveillance and security [3] are among the areas which utilize the output of SR models directly. Moreover, studies have shown that SR models are also useful in other vision tasks such as edge detection, semantic image segmentation, and digit and scene recognition [4].

Different deep learning methods have been applied to the image SR problem and have shown superior performance on the standard benchmark. However, the models tend to get more complex as they try to produce higher quality results such as RCAN [5], SAN [6] and EDSR [7]. It follows that there are different areas of focus in the image SR research area: while some studies attempt to improve the overall fidelity of the output results, others prefer to keep the model light-weight or improve the performance of complex models and making them less computationally expensive instead [8].

One of the defining features of HR images compared to the LR ones is the level of detail. Thus, it is imperative to design a model in a way that it learns to reconstruct the image by focusing on finer HR image details. As a way to facilitate this notion, we introduce a new weighted loss function, and test it in the context of two different image SR models: FSRCNN [9] and SRFBN [10]. These network architectures were chosen to test the impact on both the light-weight and state-of-the-art models.

The proposed loss function adds a new term to the models' baseline loss functions. The new term puts additional weight on image details by multiplying them with specially crafted constant weights. We explore obtaining the weights from the image details using edges, and wavelet transforms. We evaluate the new loss function performance in the context of both baseline networks. We also explore different wavelet-based modifications of the SRFBN model that change both the input and the output of the network.

2 RELATED WORK

Edge-based [11] [12], patch-based [11], and statistical methods [13] are among classical approaches that have addressed the super-resolution problem. However, deep learning, especially convolutional neural networks (CNNs), has outperformed the mentioned methods in regards the image SR task.

SRCNN [14] is one of the earliest works that learns an end-to-end mapping between low and high-resolution images directly. Dong et al. have used a CNN design that optimizes all layers jointly in comparison to its previous methods.

FSRCNN accelerates the SRCNN performance 40 times while keeping, and sometimes enhancing, the reconstructed image quality by redesigning the SRCNN in three aspects. First, having a deconvolution layer at the end of the network, which learns the SR image using LR input without any prepossessing. Further, it utilizes smaller convolution filter sizes, more mapping layers, and shrinking the input feature map before expanding it to pass to the final deconvolution layer.

Some also have chosen to go about the super-resolution problem by predicting the missing details of the low-resolution image in reconstructing its super-resolution representation. DWSR uses wavelet coefficients to collect coarse and detailed separation of the input image and predict missing coefficients using a deep neural network [8]. The approach's computational cost is lower and yet produces competitive and often better results than other state-of-the-art models at that time.

SRFBN is one of the current state-of-the-art networks proposed for the super-resolution problem. It addresses the lack of feedback mechanism, which exists in the human visual system, and introduces an RNN based model. The proposed model uses a feedback block that handles feedback connections to produce high-level representations. A curriculum learning strategy is also used to fit the model to handle the reconstruction of images that are corrupted by multiple types of degradation.

CMPT 743, Final Project Report, Apr 2020

3 METHODS

To improve the performance of image SR networks, we focused on finding a way to improve the results of certain parts of the images that are more visually apparent by the human visual system. For example, changing the contours of a cloud is more noticeable than changing the pixel values inside of it. As a potential solution, we propose a weighted loss function that applies a higher weight for those pixels that have a higher impact. This way, the network tries to minimize the amplified loss on those pixels, while giving less focus to the others. As an attempt to predict the pixels that should have higher importance, we explore both the edge-based and wavelet-based approaches.

3.1 Edge-based approach

The use of the image gradients is probably the most intuitive way to detect relevant edge pixels in an image. Our eyes are susceptible to changes in intensity, and edge detection algorithms can easily find sharp changes or discontinuities. These discontinuities could represent a change in depth, surface orientation, material property, lightning, or other variables. These are all examples of details that rapidly catch one's attention. If a super-resolution network fails to preserve such details, it outputs images that do not look appealing. Hence, attempting to upscale images while giving more focus to the performance along the edges should produce results that look more natural.

3.1.1 Edge detection algorithms. To improve the visually perceived output of the network, the choice of a suitable edge detection algorithm is essential. The optimal choice of the algorithm is crucial since the boundaries produced by the algorithm play a vital role in locating edge pixels that should get more focus form the network. Various edge detection algorithms can perform this task, with the two most common ones being Sobel and Canny edge detectors.

Sobel operator applies two convolution filters of size 3x3 over an image, one that estimates the gradients in the horizontal and the other in the vertical direction. These filters use the first-order derivative and then apply a smoothing effect to detect better horizontal and vertical gradients. Afterward, these gradients are combined using a simple Euclidean norm to obtain the absolute magnitude of the gradient. As a result, Sobel is a straightforward algorithm that usually returns thick edges.

On the other hand, Canny is a more refined algorithm for edge detection. It consists of several steps, and it could also include a Sobel operator in one of the steps. First, the whole image is smoothed by a Gaussian filter kernel to remove any noise. Then, an edge detection operator is used to compute the edge gradient and direction. After that, non-maximum suppression is applied as a measure to thin the edges. It does that by finding the most robust edges and then removing the neighbors with smaller gradient magnitude values. The neighboring pixels are found based on the edge directions. Finally, a double threshold step is applied to detect weak and robust edges.

For comparing the performance of both edge detection algorithms, we obtained their results from a set of images and then analyzed their outputs to the scope of our problem. Figure 1 showcases some of the results.

Figure 1: The pattern on the left is a cropped part of our original image. The image in the middle contains the edges obtained by the Sobel operator, whereas the last image contains the edges from the Canny algorithm.

By looking at figure 1, we can see the Sobel operator returns edges that are too thick, and do not correspond precisely to the contours of the pattern presented in the original image. In contrast, the Canny edge detector can return edges that lie more closely on top of the small round boundary shapes from the original image. Since increasing the performance, precisely on the edges, is vital to our SR network, we opted for using the Canny edge detector for our weighted loss function.

3.1.2 Weighted loss function. One of the proposed weighted loss function's advantages is being generic and applicable regardless of network structure as we only modify the loss function. The idea is to keep the original loss function, and before taking the mean loss of the pixels, apply a set of weights on them. This way, we can emphasize which pixels the model should try to minimize more. The set of weights can be defined as a 2-dimensional matrix M with the same size as the output. This matrix is computed based on the gradient estimation of the ground truth. The gradient is then thresholded between the edge and non-edge values. The corresponding non-edge pixel position is set to 1 in matrix M, while edge pixel positions are set to some constant w. The w is the weight that is applied on the edges, and its value should be higher than 1 to set higher importance for the image edges. The edge information is never passed directly into the network; therefore, no changes are needed in the model. It only requires a small change in the loss function to preserve the edges adequately. The proposed loss function can be formally written as:

$$loss = \frac{1}{N \times K} \sum_{i=1}^{N} \sum_{j=1}^{K} M_{ij} \times (\left\| \theta(LR_{ij}) - HR_{ij} \right\|_2) \qquad (1)$$

3.2 Wavelet-based approaches

Most of the deep learning-based image SR methods work on spatial domain data and aim to reconstruct pixel values as the output of the network. Image SR enhancement in the wavelet domain is a relatively new research topic, and recently many existing algorithms have been proposed [8] [5]. In this work, we explore the advantages of exploiting transform domain data in the SR task, especially for capturing more structural information in the images to avoid SR artifacts. More importantly, using residuals (differences) of wavelet coefficients as training data pairs further enhances the sparsity of training data, thus resulting in more efficient learning of filters and

activations. In other words, using wavelet coefficients encourages activation sparsity in the middle layers, as well as the output layer. Consequently, residuals for wavelet coefficients themselves become sparser and, therefore, easier for the network to learn.

3.2.1 Wavelet transform types. To increase the output SR image quality, we need to preserve the structural information of the input image. Here, we show two wavelet-based image resolution enhancement techniques. The first technique utilizes the discrete wavelet transform (DWT), while the second technique uses the stationary wavelet transform (SWT). In this project, we have explored both the discrete Haar (DWT) and the stationary Haar (SWT) wavelet transform.

The discrete wavelet transform (DWT) uses multi-resolution filter banks and special wavelet filters for image analysis and reconstruction. In 2D wavelet decomposition of an image, the 1D discrete wavelet transform is applied along the rows of the image. Then, the results are decomposed along the columns. This operation produces four decomposed sub-band images that are low-low (LL), low-high (LH), high-low (HL), and high-high (HH) frequencies. Frequency components of all these sub-bands cover the complete frequency spectrum of the input image. A single level 2D wavelet transform of an image can be captured by following the procedure in figure 2, along rows and columns, respectively. An example of a single level 2D DWT decomposition with Haar kernels is shown in figure 3. The right part of figure 3 is the notation of each sub-bands of wavelet coefficients. As mentioned earlier, we use Haar kernels in this work.

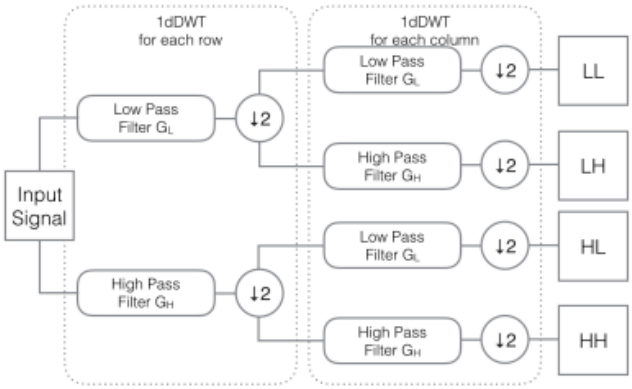

Figure 2: The procedure of a single level 2D DWT decomposition.

The stationary wavelet transform (SWT), also known as an undecimated wavelet transform, is an extension of the standard discrete wavelet transform. SWT applies high and low pass filters to the data at each level and produces two sequences at the next stage. The two new sequences have the same length as that of the original sequence. As a result, SWT is an inherently redundant transform, as the output of each level of SWT contains the same number of samples as the input. Consequently, its redundancy results in a higher reconstruction quality at the cost of requiring more memory resources than DWT.

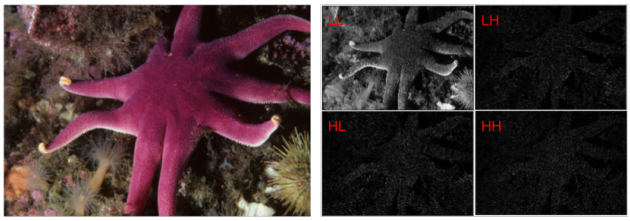

Figure 3: The image on the left is original RGB input, the right part combines the approximation (LL), vertical detail (HL), horizontal detail (LH), and diagonal detail (HH) sub-bands.

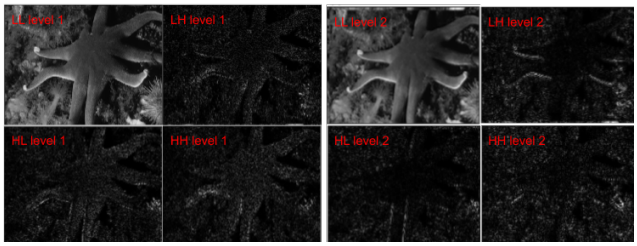

Figure 4: Using two level SWT, image is decomposed into four sub-bands in each level.

An example of 2-level SWT of an image is shown in figure 4. The reason for using SWT over DWT is that the size of the image is retained in SWT, whereas in DWT the down-sampling of sub-bands causes information loss.

3.2.2 Weighted loss function. Both the DWT and SWT Haar wavelets can be used as a part of the weighted loss function term. Similar to the edge-based weighted loss function approach, wavelet detail coefficients can be used to extract the image details, which are then used to correct the loss term. This approach has shown to achieve higher fidelity results in the FSRCNN model compared to the edge-based approach.

3.2.3 Model input and output. Both the DWT and SWT Haar wavelets can also be used as model input and output instead of the regular image color channels. Such an approach has shown to produce higher quality results in simpler CNN architectures, such as DWSR [8]. However, when applied to more complex architectures, such as SRFBN, we later show that they do not necessarily outperform the conventional image color channel designs.

4 EXPERIMENTS

4.1 FSRCNN

Fast super-resolution convolutional neural network (FSRCNN) is based on a simple network design to generate a super-resolution image efficiently [9], by re-designing existing SRCNN. It is shown to be faster with better-reconstructed image quality than the original SRCNN.

FSRCNN adopts the original low-resolution image as input without bicubic interpolation. A de-convolution layer is introduced at the end of the network to perform up-sampling. The non-linear mapping step in SRCNN is replaced by three steps in FSRCNN, namely the shrinking, mapping, and expanding step. These characteristic makes it easier for us to learn about the effect of each component, and also implement our approaches to improve the preference.

FSRCNN uses mean squared error (MSE) loss function to minimize the error between the reconstructed image and the ground truth. Minimizing this loss function may reduce the high-frequency content in the reconstructed image and thus may blur the edges in it. Also, the reconstructed image may not lie precisely in the manifold of the HR image. We have performed a large number of experiments to obtain a weighted loss function that improves the performance of FSRCNN.

4.1.1 Edge-based weighted loss function. In our edge-based method, we experimented using weights for the edges in the range of 1 to 6 with increments of 0.2. Using weights equal to 1 would be equivalent to the baseline model (without any weights at all). So we obtained a baseline model and twenty-five additional models, with different weights for comparison.

4.1.2 Wavelet-based weighted loss function. We apply Haar discrete wavelet transformation on the ground truth image. We obtain the horizontal (LH), vertical (HL), and diagonal (HH) details form the wavelet coefficients and then sum them up to build the detail mask of the input image. Figure 5 shown the steps of calculating weights from wavelet residuals. In this approach, we use the same formula as edges weighted loss function, apply the wavelet residuals weights instead of edges weights.

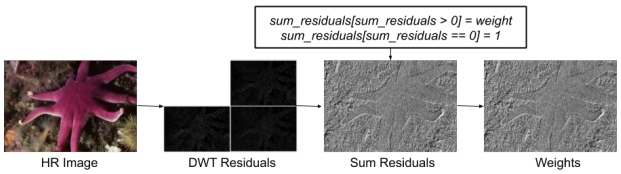

Figure 5: The process of calculating wavelet weights.

4.2 SRFBN

Super-resolution feedback network (SRFBN) is one of the current state-of-the-art image super-resolution deep learning networks. It features a feedback block mechanism that captures the output at the previous iteration step to correct the input of the following one.

The SRFBN can be unfolded into T iterations (1 to t), with each feedback block having G sequential projection groups with dense skip connections among them. The network features an adjustable number of base filters m, thus allowing one to fine-tune the model size by changing the respective hyper-parameters m, T, and G. While larger parameter values can lead to overall higher performance, they also result in a bulkier model that requires more time and memory to train. As a result, for this project, we have chosen the SRFBN-S (m=32, T=4, G=3) variation to be the baseline for our

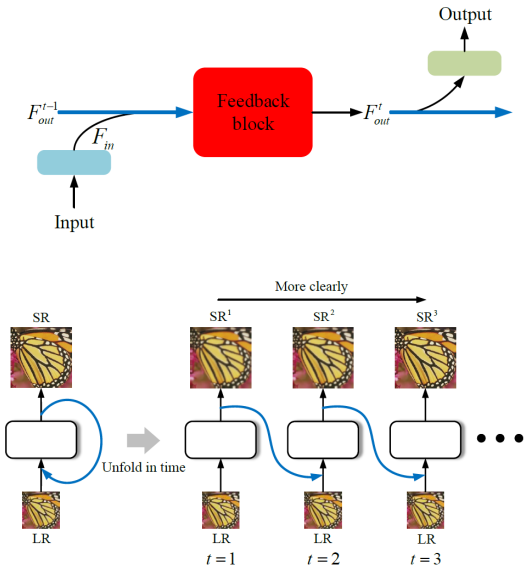

Figure 6: The feedback mechanism of the SRFBN model. The feedback block output at a step (t-1) is used as a feedback block input at a step (t). [10]

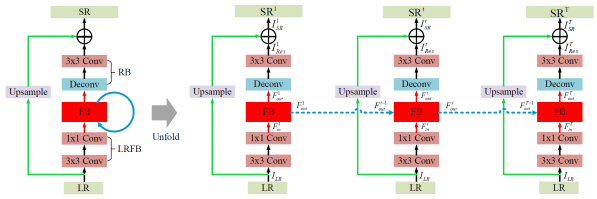

Figure 7: The architecture of the SRFBN model. Blue arrows represent feedback connections. Green arrows represent global residual skip connections. [10]

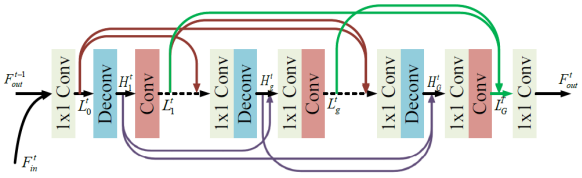

Figure 8: The feedback block (G=3) architecture of the SRFBN model. [10]

further experiments as a compromise between the model output quality and performance.

Further, we explore different SRFBN design to explore the impact of the edge-based and wavelet-based weighted loss functions, as well as the use of the Haar wavelet coefficients in the internal model

structure. While SRFBN-WL design features the original baseline model architecture with a changed loss function term, the other designs modify the model itself to accept different input and output, which, due to the recurrent architecture of the SRFBN, are then propagated throughout the whole network.

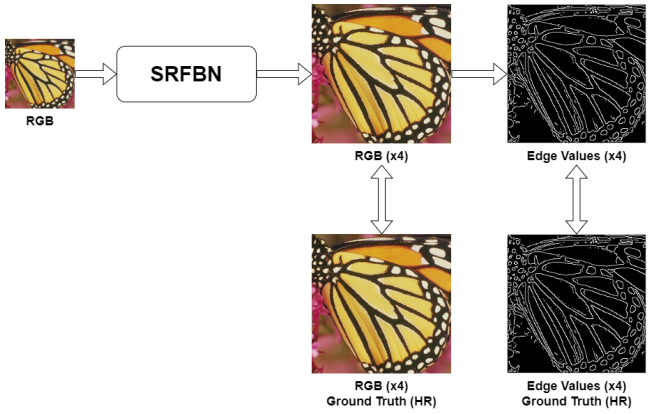

Figure 9: The SRFBN-WL design features a baseline SRFBN model along with a modified loss function term. The loss is taken on both the RGB image and the normalized image edge values.

4.2.1 SRFBN-WL. The SRFBN-WL modification features our baseline SRFBN model, along with a modified loss function term. As the network itself was not changed, the number of input and output channels remained to be 3 RGB channels each. Similar to **??**, the new loss term is as follows:

$$loss_{WL} = \alpha * L_1(SR_{rgb}, HR_{rgb}) + \beta * L_1(SR_{edge}, HR_{edge}) \quad (2)$$

The α and β are the hyper-parameter constants chosen empirically to allow the adjustment of the edge loss term impact on the final loss value. To obtain SR_{edge} and HR_{edge} terms, we use a PyTorch implementation of the Canny algorithm, modified to produce normalized edge intensity values between 0 and 1. The threshold hyper-parameter th is used to adjust the minimum edge intensity value below which the values are set to 0.

4.2.2 SRFBN-DWT/SWT. The SRFBN-DWT and SRFBN-SWT modifications both use the Haar wavelet transform instead of the original RGB color channels design. Each RGB channel of the LR input image is first converted into its wavelet form. The wavelet coefficients are then concatenated to produce a 12 channel input and output design. The up-sample SR wavelet coefficients are then used to reconstruct the final SR RGB image.

4.2.3 SRFBN-DWT/SWT (YCbCr). Unlike the SRFBN-DWT and SRFBN-SWT models, which use RGB channels' wavelet representations, the SRFBN-DWT (YCbCr) and SRFBN-SWT (YCbCr) models utilize only the luminance channel (Y) of the input image. The luminance channel contains the target image details, while the chrominance channels (Cb, Cr) contain the target image color data. The SRFBN YCbCr modifications output the up-sampled SR wavelet

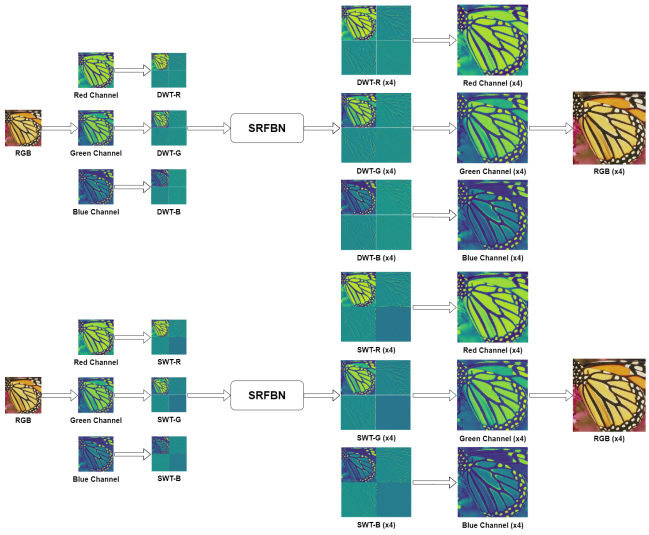

Figure 10: The SRFBN-DWT and SRFBN-SWT designs feature a modified SRFBN model that accepts the DWT and SWT Haar wavelet coefficients as an input. The model outputs the up-sampled SR wavelet coefficients that are used to reconstruct the SR RGB image.

representation of the Y channel, which, alongside the interpolated Cb and Cr image chrominance channels, is then used to reconstruct the SR RGB image.

4.2.4 SRFBN-RES-1. Unlike the original SRFBN design, which accepts 3 RGB channels as an input, the SRFBN-RES-1 modification accepts both the RGB channels and the 9 SWT Haar detail coefficients of each input RGB channel. These are concatenated to form a 12 channel model input, which is eventually transformed into a 3 channel RGB model output during the final reconstruction step. Such design presents the model with additional information about the input image residual details and is later shown to have a higher baseline validation PSNR score during the early training epochs.

4.2.5 SRFBN-RES-2. The SRFBN-RES-2 modification accepts only the 9 SWT Haar detail coefficients as an input. This approach changes the model's task from up-sampling the whole image, to only up-sampling the image's residual details. The RGB color channels are directly interpolated and added to the SR image details to reconstruct the final SR RGB image.

5 RESULTS

5.1 FSRCNN

5.1.1 Quantitative results. For the edge-based approach, we have generated twenty-five checkpoints with different weights. To compare them all with the baseline, we have created a graph with the PSNR result from all our models applied to the test set, as shown in figure 14.

The overall PSNR tends to go down when applying high weights. This is because the PSNR is mainly based on the mean squared

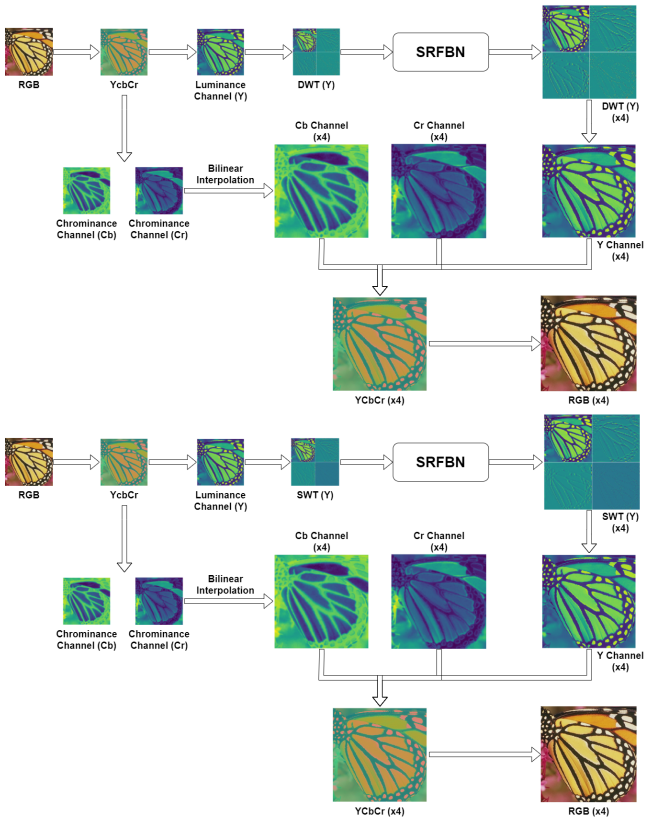

Figure 11: The SRFBN-DWT (YCbCr) and SRFBN-SWT (YCbCr) designs feature a modified SRFBN model that accepts the image luminance channel (Y) DWT and SWT Haar wavelet representation as an input. The model outputs the up-sampled SR wavelet representation of the Y channel which, along side the interpolated Cb and Cr image chrominance channels, is used to reconstruct the SR RGB image.

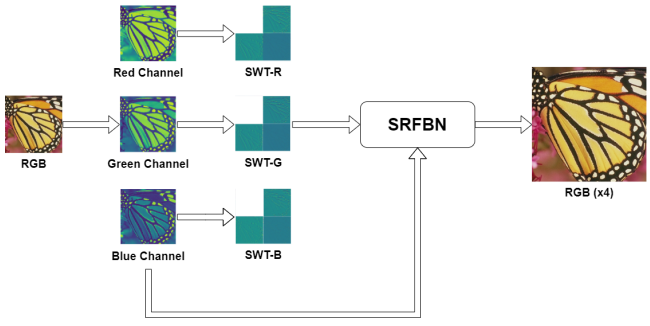

Figure 12: The SRFBN-RES-1 design features a modified SRFBN model that accepts both the RGB image and the image SWT Haar wavelet detail coefficients as an input. Similar to the original design, the model outputs the SR RGB image.

error (MSE), the same as the baseline's original loss function. Applying pixel-wise weights on the details (residuals or edges) results

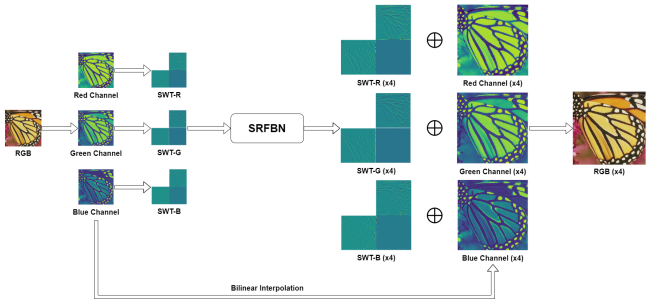

Figure 13: The SRFBN-RES-2 design features a modified SRFBN model that takes only the image SWT Haar wavelet detail coefficients as an input. The model outputs the up-sampled SR wavelet detail coefficients which, along side the interpolated RGB channels, are used to reconstruct the SR RGB image.

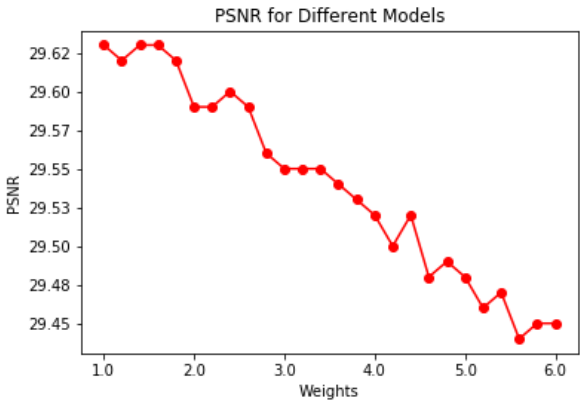

Figure 14: PSNR for the baseline model at weight equal to 1, and our models with higher weights with increments of 0.2.

in the generation of a model that is not minimized based on the standard MSE loss. Therefore, the computed PSNR is expected to be lower. Nevertheless, to check if our models with higher weights are generating images which have closer edges to the edges from ground truth, we also computed the PSNR over only the edges, as seen in figure 15.

Figure 15 demonstrates clearly that the models with higher weights are able to successfully generate images with edge shapes closer to the ground truth. By analyzing both graphs together, we have come to the conclusion that our model with weight equal to 1.6 presented the best results. This assumption was made because that model had an overall PSNR equal to the baseline (figure 14) while having an increased PSNR over the edges (figure 15).

Figure 16 shows the results PSNR of all experiments in FSRCNN. It is evident that the DWT weighted loss improves the performance as expected, it is also shown that training with the weighted loss using discrete wavelet transform results in higher and more stable PSNR than the baseline model and model trained with edges weighted loss function. The table 1 records the best validated PSNR

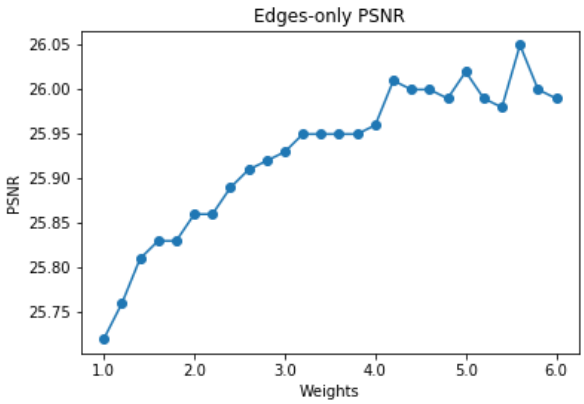

Figure 15: PSNR computed by taking only the edges into consideration.

in 100 epochs of each model. As we can see, the model with edges weighted loss gets the same PSNR as the baseline model, and the DWT weighted loss function improves the performance slightly. In the next section, we show some qualitative results.

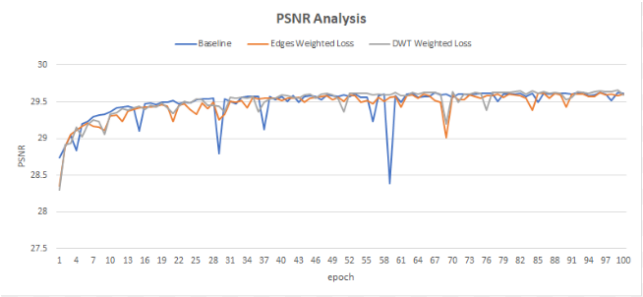

Figure 16: PSNR graphs using weighted loss functions.

Test Dataset	Baseline	Edges	DWT
DIV2K Validation Data	29.63	29.63	**29.66**

Table 1: Comparison of average validated PSNR with different loss function.

5.1.2 Qualitative results. By analyzing our output results, we could see that our models can up-sample the image resolutions with a shape more similar to the ground truth when the baseline fails to do so. In figure 17, for instance, we can see that the baseline failed to generate a round shape of the right side of the eye, while our models at the bottom could more successfully keep the round shape shown in the ground truth.

In figure 18, we could also notice that both our models at the bottom could produce results that are more similar to the round shapes presented in the ground truth. The baseline model, on the other hand, had more artifacts along the contours.

Figure 17: Cropped patch from "Bird" picture of Set5. On the right side, we have the ground truth at the top left, baseline result at top right, edge-based model at bottom left, and wavelet-based model at bottom right.

Figure 18: Cropped patch from "img_004" picture of urban100 dataset. From top to bottom, we have: ground truth, baseline model, edge-based model, and wavelet-based model.

Test Dataset	Baseline	Edges	DWT
bird in set5	32.66	32.79	**32.87**
img_004 in urban100	22.23	22.24	**22.27**

Table 2: Comparison of PSNR of single image with different loss function.

5.2 SRFBN

5.2.1 Quantitative results. In this part, we analyze the results of our experiments quantitatively done on the SRFBN model. We have used Peak Signal-To-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) metrics to evaluate the performance of our experiments. Using PSNR, which estimates absolute errors along with SSIM, is a perception-based model that leads to a more comprehensive understanding of the performance of each experiment. Figure 19 displays the PSNR and SSIM of all experiments. It can be seen that using the wavelet transformation of the RGB or YCbCr image as an input to the model did not improve the performance as expected. The reason behind this is, We have narrowed down all the experiments to be done on scale 4. Down-sampling an image 4

times results in losing a considerable amount of details. Therefore there is a lot less information to be extracted and start the training. However, using the stationary wavelet transform of the LR image as input showed to have a closer value of PSNR and SSIM to the baseline, as shown in figure 20.

Using the weighted loss function, on the other hand, proved to have a better performance as the original model. As displayed more clearly in figure 21, adding weighted loss term to the original loss using α = 0.7, β = 0.3 and threshold = 0.1 had the best performance among all the experiments and the baseline model itself. The other set of experiments was done by feeding residuals to the feedback network with and without RGB channels. Figure 22 shows the PSNR and SSIM of these two experiments. The figure also shows that providing this additional information did not enhance baseline performance.

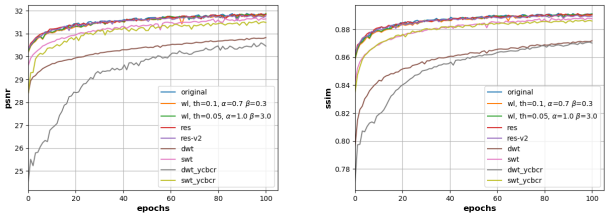

Figure 19: PSNR and SSIM graphs of all experiments.

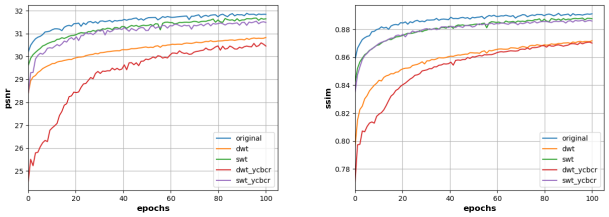

Figure 20: PSNR and SSIM graphs of SRFBN-DWT/SWT designs.

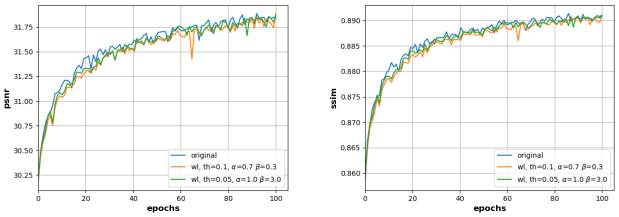

Figure 21: PSNR and SSIM graphs of SRFBN-WL designs.

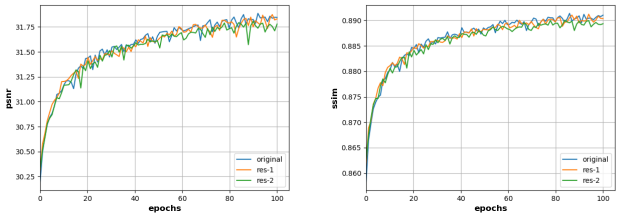

Figure 22: PSNR and SSIM graphs of SRFBN-RES designs.

5.2.2 Qualitative results. In this part, we show the qualitative results of our experiments. Then further display the results of the best performing experiment, which is the model trained with the weighted loss with threshold = 0.1, α = 0.7, and β = 0.3.

The experiments have been tested on the SR test benchmark (B100, Manga109, Set5, Set14, Urban100). Figure 23 and 24 show the selected results of all of the experiments on the "baby" and "woman" image of Set5, respectively.

Figure 23: Results from the "baby" picture in Set5.

Figure 24: Results from the "woman" picture in Set5.

Our weighted loss function performed best mostly on faces and places with sharp intensity change. Naturally, faces have larger flat areas; therefore, the changes of the intensity on the edges such as lips, nose, and eyebrows are more prominent and defined as shown in figure 25 and 26.

Figure 25: Results from the "comic" picture in Set14.

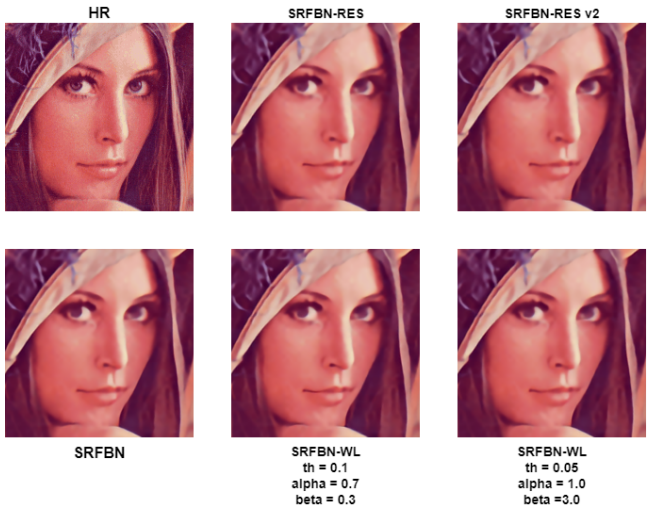

Figure 26: Results from the "lenna" picture in Set14.

6 CONCLUSION

In this paper, we explore the applications of the edge weights, and the wavelet transforms in the context of image super-resolution. We perform both quantitative and qualitative experiments on the impact of the weighted loss function, and the impact of wavelet transforms embedded in the model design. Further, we showcase the results generated by our modified FSRCNN and SRFBN model designs and conclude that the weighted loss function can have a positive impact on the learned model layer weights and, consequently, the generated SR image results.

REFERENCES

[1] Yawen Huang, Ling Shao, and Alejandro F. Frangi. Simultaneous super-resolution and cross-modality synthesis of 3d medical images using weakly-supervised joint convolutional sparse coding. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] Amanjot Singh and Jagroop Singh. Super resolution applications in modern digital image processing. *International Journal of Computer Applications*, 150(2):6–8, 2016.

[3] Liangpei Zhang, Hongyan Zhang, Huanfeng Shen, and Pingxiang Li. A super-resolution reconstruction algorithm for surveillance images. *Signal Processing*, 90(3):848–859, 2010.

[4] Yuhua Chen Dengxin Dai, Yujian Wang and Luc Van Gool. Is image super-resolution helpful for other vision tasks?

[5] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. *Computer Vision – ECCV 2018 Lecture Notes in Computer Science*, page 294–310, 2018.

[6] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[8] Tiantong Guo, Hojjat Seyed Mousavi, Tiep Huu Vu, and Vishal Monga. Deep wavelet prediction for image super-resolution. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[9] Xiaoou Tang Chao Dong, Chen Change Loy. Accelerating the super-resolution convolutional neural network.

[10] Zhen Li. Feedback network for image super-resolution.

[11] Gilad Freedman and Raanan Fattal. Image and video upscaling from local self-examples. *ACM Transactions on Graphics*, 30(2):1–11, 2011.

[12] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[13] Kwang In Kim and Younghee Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1127–1133, 2010.

[14] Dong, Chao, Loy, Chen Change, Tang, and Xiaoou. Image super-resolution using deep convolutional networks, Jul 2015.