Image Super-Resolution using Modified Residual Network

Ronak Jain, Ronak Mali and Ajinkya Kate
Department of Information Technology,
Vidyavardhini's College of Engineering and Technology, India

***Abstract*—The applications of image super-resolution techniques in recent years has increased noticeably. From a video playback on high-resolution LED TVs to Satellite Imaging all use image upscaling. The main purpose of image upscaling is to obtain high-resolution images from low-resolution images without any extra information except the input images, and these upscaled images should keep satisfactory visual qualities and natural textures. The most popular image upscaling algorithms are interpolation methods. But, in recent times Neural Network is used for upscaling the images more accurately. We propose a Neural Network based method to upscale the lower resolution image into a higher resolution image and improve the image quality without losing much detail in the process. We designed an architecture based on the modification of residual learning method which generates the high resolution image faster with some blurry effects which is removed in the post processing step. The experiments conducted using the proposed method shows that even with much fewer parameters and operations due to group convolution, this model achieves comparable results to other deep learning based methods.**

***Index Terms*—image super-resolution, deep learning, neural network, residual network, group convolution**

I. Introduction

Image super-resolution or upscaling is a method of constructing or generating a higher resolution image as an output from one or more lower resolution source images. There are many applications of this method. For example, it can be used when a low-resolution video or photo is being displayed on a higher resolution screen and increasing the apparent quality of the picture or video being shown. It can also have major functioning in the security surveillance process e.g. if a face of a suspect is caught on CCTV camera and is not of high enough quality for proper identification, intelligent image upscaling can approximate the details of the face for more accurate identification.

One of the two ways to achieve high-resolution image from lower resolution source image is called single image super-resolution where the output high-resolution image is generated from a single low-resolution image. The other method named as multiple-image super-resolution, where the high-resolution image is created from multiple source images. Both of these methods are used depending on the use case. Generally, multiple image super-resolution provides better results as there is more data to begin with. However, in certain circumstances, obtaining multiple minutely different images of the same subject is not practical. For instance, if one were to upscale an image from a CCTV video footage. Then, there may not be multiple images originating from the same angle at the same instant, hence making multi-image super-resolution techniques incapable.

There are many techniques for performing image upscaling like basic interpolation methods viz. Nearest Neighbour, Bilinear and Bicubic interpolation methods which are computationally cheap and are not complex. These methods are very common and widely used for image upscaling. In recent years deep learning based methods are getting more popular as such methods are able to output images with much better quality than interpolation techniques. But, these deep learning methods need more time and are computationally heavy to output upscaled image. So, these methods despite producing high quality images are not possible for real-time applications like upscaling video which requires the output to be produced more quickly. However, new methods are being proposed with lightweight and efficient network architecture which tries to produce the output high resolution images quicker and are not that computationally heavy.

In this paper, we use residual network along with group convolution to modify the residual block and implement the training model. This makes the network lightweight as it requires only few parameters which produces the output upscaled image faster. However, the result image is little blurred so we introduce a post processing step. The post processing step helps sharpen the image which improves the quality of the image. In further section, few existing image upscaling methodologies from a single source image using neural network architecture are discussed.

II. Related Work

Recent super-resolution algorithms are mostly neural network based or patch-based methods that learn a mapping between the low-resolution and high-resolution image spaces. This learning based methods are mainly Machine Learning and Neural Network methods. The Convolutional Neural Networks (CNN) which is the neural network based method specifically used for computer vision is used to produce the higher resolution image. Super-Resolution Convolutional Neural Network (SRCNN) was first such method proposed by Dong et al. [1]. This method outperformed the previous super-resolution algorithms. The SRCNN network learns the mapping which conceptually consists of three operations which are patch extraction and representation, non-linear mapping and reconstruction. Patch extraction is used to extract overlapping

patches from the low-resolution image and represents each patch as a high dimensional vector. These vectors consists of a set of feature maps. The number of feature maps equals to the dimensionality of the vectors. Non-linear mapping is used to map each high-dimensional vector onto another high dimensional vector. The vector which has been successfully been mapped is conceptually the representation of a high-resolution patch. These vectors consists of another set of feature maps. Finally, in reconstruction overlapping high-resolution patches which are predicted in the traditional methods, are often averaged to produce the final full image. This averaging of the patches can be considered as a filter which is predefined on a set of feature maps.

However, this method has a large number of operations compared to its depth, as SRCNN network takes Bicubic upsampled images as input before applying convolution layer. There are two limitations in SRCNN. First the original low-resolution image needs to be upsampled as a pre-processing step to the desired size using Bicubic interpolation to form the input. Thus, the computation complexity of SRCNN grows with the spatial size of the high-resolution image.

This problem was solved by implementation of FSRCNN [2]. Since in this approach the images are upsampled at the end of the network. This helps in reducing the number of operations compared to SRCNN. But, if there are not enough layers, the overall performance can be degraded. In FSRCNN there is a trade-off between the time taken to produce a result and the quality of the image produced as the output. In FSRCNN the upscaled images are produced much faster than SRCNN while only having low impact on quality of the resultant images. While SRCNN being a deep learning method it has only three convolutional layers whereas other deep learning methods can have more than 100 layers. Method proposed by Lim et al. [3] Multi-Scale Deep Super-Resolution (MDSR) has more than 160 layers. From time to time the depth of the network and performance has improved but these methods are quite computationally heavy and requires a large amount of time to produce results. In order to make the network lightweight, the parameters and number of operations should be reduced. Residual learning helps in such situation where the network model needs to be lightweight. ResNet architecture proposed by Kaimin et al. [4] is being widely used in deep learning based image upscaling methods like the methods proposed by Christian et al. [5]. The ResNet architecture gives superior performance, provides ease of training and is quite efficient.

III. Implementation

Following the ResNet architecture, we build a model and modified the residual blocks with group convolutions instead of using depthwise convolution which makes the model more lightweight.

A. Upsampling method

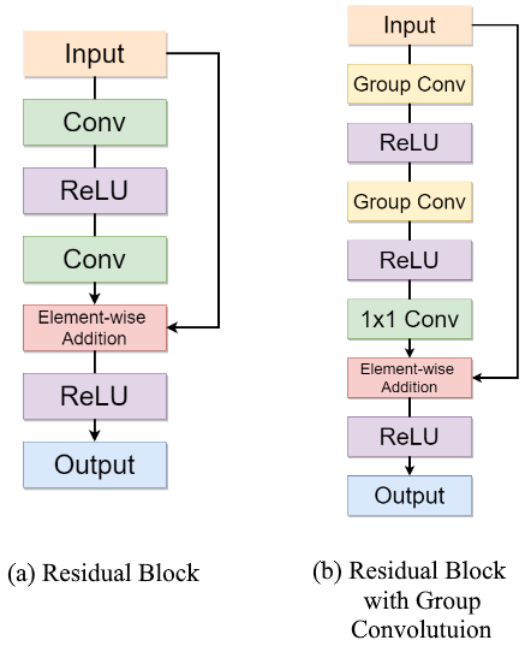
There has been variety of upsampling algorithms making use of neural networks which learns the end-to-end upsampling process directly. The upsampling operation is a major step in training the model. There are two types of upsampling process. First is based on the basic interpolation method. The traditional interpolation methods include nearest-neighbour, bilinear, bicubic interpolation, Sinc and Lanczos resampling [6] and others. These methods are interpretable and also easy to implement. Some of these interpolation methods are still used widely in deep learning based super-resolution. However, the interpolation methods introduces problems such as increasing the computational complexity of the model based on such interpolations, noise amplification and also producing blurry results. The other upsampling process is learning-based. It helps overcome the shortcomings of the above interpolation based methods. We use the learning based upsampling technique, sub-pixel layer for the upsampling operation. The sub-pixel layer has a larger receptive field and provides more contextual information to help generate better and accurate details. This method performs upsampling by generating numerous channels by convolution and then reshapes them.

B. Network Design

The network model is based on ResNet. The plain ResNet consists of multiple residual block as shown in Fig. 1(a) where each block has a convolution layer followed with an activation function ReLU [7] and then again convolution is applied. The output of the residual block is then formed by element-wise adding the input to this result and the activation function is applied again. We modify this residual block by introducing group convolution in this process. Fig. 2 shows the working of the group convolution which helps in decreasing the depth of the network and this making the model lightweight. Usually with group convolutions less amount of GPU memory is required in comparison to normal convolution based network which would be quite deep. This helps reduce plenty of parameters and operations as the input is split before applying convolution and the output is formed by concatenating the output of those individual convolutions. So, by introducing group convolution in the residual block and addition of the input to the 1×1 convolution, we apply the activation function ReLU and get the output of the corresponding residual block. We do this process nine times as shown in Fig. 3 and provide the output of the previous blocks as input for the remaining further blocks recursively. After the residual blocks have been implemented we apply the above discussed upsampling operation based on the scale factor. Scale factor can be x2, x3, x4 in this case. The final convolution is done on the upsampled image to generate the super-resolved image.

C. Residual Learning

Residual learning is used for learning residual instead of throughout mapping. Residual learning can be further differentiated as global residual learning and local residual learning. We use local residual learning in our proposed model. Local residual learning helps in training as it reduces the difficulty in training and also improves the rate of learning used in deep learning. Local residual learning is implemented by



(a) Residual Block (b) Residual Block with Group Convolutuion

Fig. 1: Residual Block Architecture

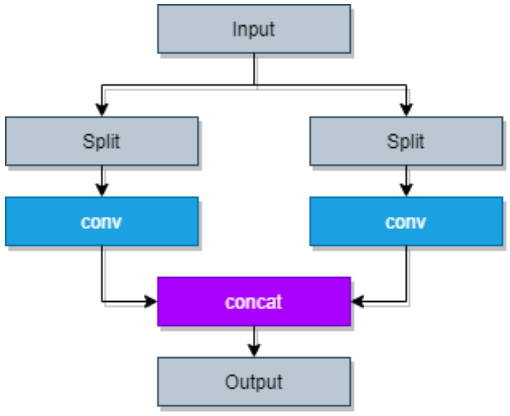


Fig. 2: Group Convolution Architecture.

shortcut connections which are implemented between multiple layers inside the network of different depths and element-wise addition. As proposed by Kaimin et al. [4], considering $\mathcal{H}(x)$ as an underlying mapping and x denoting the inputs to the first layers. The underlying mapping is to be fit by a few stacked layers. So the residual function after some approximation is $\mathcal{F}(x) + x$ where $\mathcal{F}$ is the super-resolution model.

Normally a residual learning block in Fig. 1(a) can be defined as:

$$y = \mathcal{F}(x, \{W_i\}) + x \tag{1}$$

Here, x and y denotes the input and output vectors of the layers considered. The residual mapping to be learned is represented by function $\mathcal{F}(x, \{W_i\})$. Therefore, $\mathcal{F} = W_2\sigma(W_1x)$ in Fig. 1(a) where there are two layers and σ denotes ReLU [7] which is an activation function. The biases are omitted for simplifying notations. This notation is used for denoting a residual block. We have modified this block in Fig. 1(b) where the depth wise convolutions are replaced by group convolutions for making the model more lightweight. The residual block with group convolution as shown in Fig. 1(b) consists of two 3×3 group convolution.

D. Loss Function

The loss function is used to optimize the super-resolution model. It is also used to measure the reconstruction error. There are various loss functions which can be adopted according to the requirements. Content Loss, Pixel Loss, Adversarial Loss are few such types of loss function. Here, we implement the pixel loss function which measures the difference between two images at the pixel level. Pixel loss function includes L1 loss which is mean absolute error and L2 loss which is mean sqaure error. The L2 loss function is related to peak signal-to-noise ratio which is widely used in image restoration task. Here, we use the L1 loss function instead of L2. Here we use the L1 loss function instead of L2 as it provides better convergence and performance. The L1 loss function is as given below.

$$\mathcal{L}_{\text{pixel\_l1}}(\hat{I}, I) = \frac{1}{hwc}\sum_{i,j,k}|\hat{I}_{i,j,k} - I_{i,j,k}| \tag{2}$$

$$\mathcal{L}_{\text{pixel\_l2}}(\hat{I}, I) = \frac{1}{hwc}\sum_{i,j,k}(\hat{I}_{i,j,k} - I_{i,j,k})^2 \tag{3}$$

Here, $\hat{I}$ represents the reconstructed high-resolution image, I represents the groundtruth image. The height, width and number of channels of the evaluated images are denoted by h, w and c respectively.

E. Training Details

The RGB input patches having size of 64×64 from the low-resolution images are used for training. These patches are sampled and augmented with random horizontal flips and 90 degree rotation. The model is trained using ADAM optimizer [8]. The learning rate starts with 10^{-4}. The learning rate gets halved after every 4×10^5 steps. The weights and biases are initialized based on the number of channels of input feature maps. We use multiple narrow 1×1 convolution layers. The initialization of weights for these layers are sampled from a uniform distribution to ensure stability in the training process. We use the L1 loss function(i.e., mean absolute error) which provides better convergence and performance. This loss function constrains the output high-resolution image to be as close as possible to the groundtruth image on the pixel values.

F. Post Processing

The post processing step involves removing the blurry effects after the reconstruction of the output image in the final stage. We use a sharpening mechanism used by Vanmali et al. in [9] which achieves this and also improves the visual quality of the image. We first convert the output color image to HSV color space and filter the luminance component $I_T(x, y)$ by high-pass filter. The high-frequency components are extracted using this filter and added to the luminance component. The

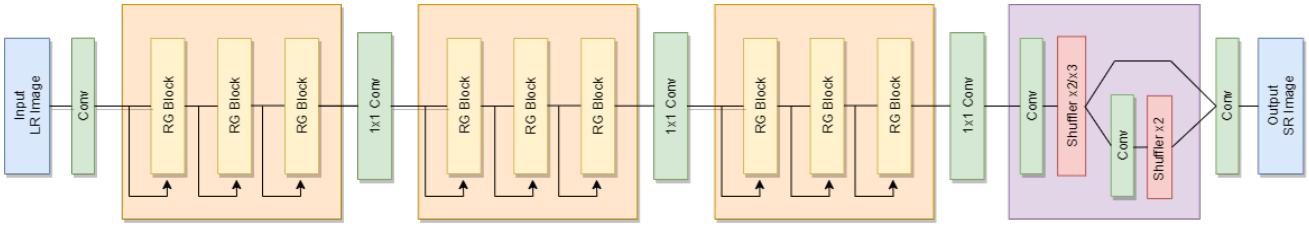


Fig. 3: Network Architecture of the proposed method.

output is converted back to the RGB color space to get the result $I_{Out}(x, y)$.

$$I_{Out}(x, y) = I_T(x, y) + \lambda[I_T(x, y) \otimes F(x, y)] \quad (4)$$

The $F(x, y)$ denotes the high pass filter whereas λ denotes the control parameter for sharpening which is greater than or equal to zero. Here, we set the value of $\lambda = 0.4$. Intead of linear high-pass filter we use the weighted median (WM) filters which provides sharpening along with immunity to noise. The WM filter mask used is shown below:

$$WM = \frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (5)$$

This produces a better visual quality image and removes the blur effect produced in the previous stage.

IV. Results and Discussion

A. Image quality assessment

The most widely used quality assessment metrics for super-resolution are Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). The PSNR tends to result in excessive smoothing. Also the results vary wildly between almost identical images. The SSIM [10] is used for measuring the structural similarity between two images since human visual system can extract this structural information from its viewing field. The SSIM is based on three comparisons which are luminance, contrast and structure. The perceptual assessment requirements are met since SSIM evaluates the reconstruction from human visual system perspective. Therefore, we use SSIM method for evaluating the quality of our super-resolved images.

B. Datasets

The most widely used datasets for training are the 291 image set by Yang et al. [11] and Berkeley Segmentation Dataset [12]. Since these two datasets do not have sufficient number of images required for training a deep neural network, we use the DIV2K dataset [13]. The DIV2K dataset is a high-quality image dataset, consisting of 1000 images out of which 800 are training images, 100 are validation images, and 100 are test images. The datasets consists images of environment, flora, fauna, handmade object, people, scenery, etc. The standard benchmark datasets such as Set5 [14], Set14 [15], BSD100 [16] and Urban100 [17] is used for testing and benchmarking.

TABLE I: SSIM Results of x2 Upscaled images on Benchmark Test Datasets.

Method	Set5	Set14	BSD100	Urban100
Bicubic	0.917	0.856	0.844	0.839
FSRCNN	0.937	0.884	0.884	0.862
Proposed	**0.949**	**0.903**	**0.901**	**0.923**

TABLE II: PSNR Results of x2 Upscaled images on Benchmark Test Datasets.

Method	Set5	Set14	BSD100	Urban100
Bicubic	31.78	28.31	28.24	25.67
FSRCNN	33.98	29.82	29.72	27.22
Proposed	**35.61**	**31.22**	**30.75**	**30.28**

TABLE III: SSIM Results of x3 Upscaled images on Benchmark Test Datasets.

Method	Set5	Set14	BSD100	Urban100
Bicubic	0.853	0.761	0.739	0.736
FSRCNN	0.885	0.796	0.781	0.77
Proposed	**0.912**	**0.827**	**0.806**	**0.846**

TABLE IV: PSNR Results of x3 Upscaled images on Benchmark Test Datasets.

Method	Set5	Set14	BSD100	Urban100
Bicubic	28.62	25.73	25.88	23.12
FSRCNN	30.31	26.86	26.86	24.21
Proposed	**32.22**	**28.13**	**27.72**	**26.50**

TABLE V: SSIM Results of x4 Upscaled images on Benchmark Test Datasets.

Method	Set5	Set14	BSD100	Urban100
Bicubic	0.789	0.685	0.661	0.651
FSRCNN	0.828	0.72	0.697	0.696
Proposed	**0.876**	**0.764**	**0.733**	**0.779**

The results obtained shows the SSIM and PSNR values of the proposed method implemented on AMD Ryzen 3550H CPU and NVDIA GTX1650 GPU with 4GB of graphics memory, along with their corresponding Bicubic and FSRCNN counterparts. The time taken for producing the output was very small. The BSD100 dataset which contains 100 images

TABLE VI: PSNR Results of x4 Upscaled images on Benchmark Test Datasets.

Method	Set5	Set14	BSD100	Urban100
Bicubic	26.69	24.25	24.65	21.7
FSRCNN	28.08	25.16	25.35	22.62
Proposed	**30.11**	**26.10**	**26.25**	**24.55**

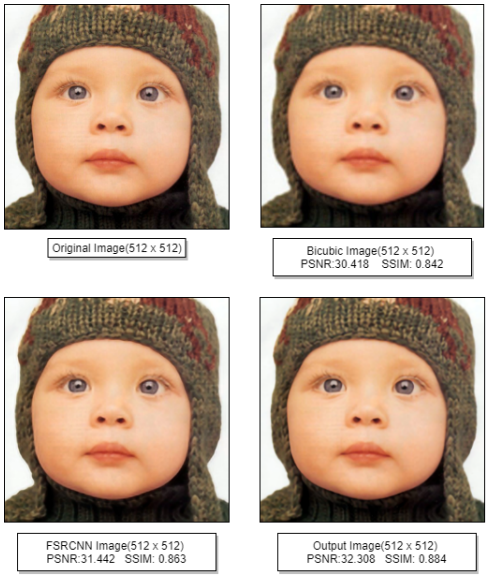


Fig. 4: Sample Output of x4 Upscaled Baby Image from Set5 dataset.(Image Courtesy: [14])

took approximately 3.1 to 3.2 seconds to produce all the output images at 4x scale that is one image every 0.03 seconds approximately. Similarly Urban100 dataset which also contains 100 images took nearly 3.5 to 3.6 seconds to produce all the output images at 4x scale. For x2 super-resolution we see better SSIM and PSNR values against normal Bicubic interpolation and FSRCNN values for Set5, Set14, BSD100 and Urban100 datasets. This is same for x3 and x4 super-resolution. SSIM is a better evaluation metric than PSNR when it comes to evaluating how similar two images are. So, in some cases the difference between the PSNR values can be excused where the difference in values is not that high. Here, we show the sample image outputs from above mentioned datasets with their groundtruth image for visual comparison.

As there is a trade-off between performance and quality, for better performance some quality will be sacrificed. Hence, we obtained slightly blurred images after upscaling. The post processing step sharpens the upscaled image. Although, the SSIM and PSNR value take a small hit by sharpening the image but, the sharpened image is usually preferred visually over the blurred upscaled image. The sharpened image is compared with ground truth image and the output image of proposed method in Fig. 6.

The results obtained shows that the proposed method produces approximately 3%, 5% and 7% average gains in SSIM values for x2, x3 and x4 super-resolution respectively over

Fig. 5: Sample Output of x4 Upscaled Lenna Image from Set14 dataset.(Image Courtesy: [15])

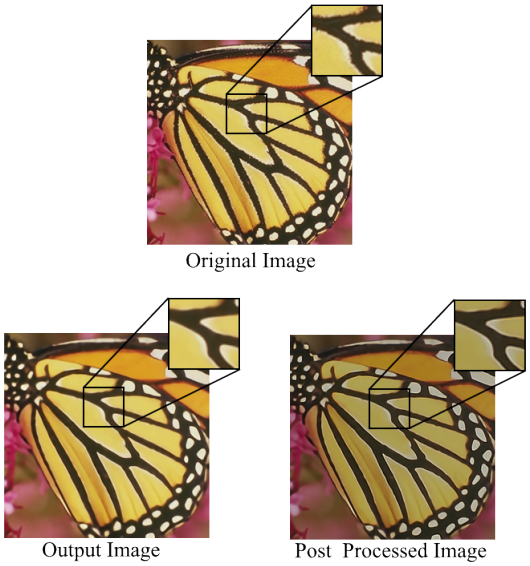


Fig. 6: Post Processed and Sample Output Image of x4 Upscaled Butterfly Image. (Image Courtesy: [14])

FSRCNN. Also, 6%, 5% and 5% average gain in PSNR values for x2, x3 and x4 super-resolution respectively over FSRCNN.

V. Conclusion

Many deep learning methods use a lot of computational power and takes a large amount of time for image upscaling. So, we proposed a method with modified residual block using group convolution. This makes the model lightweight and efficient and also requires less image parameters for producing upscaled image. We got 7% and 5% improvement

over FSRCNN in x4 super-resolution in SSIM and PSNR values respectively, which is frequently used scale for super-resolution. There is always a trade-off between quality and performance when it comes to super-resolution. By using less parameters the proposed method is quite fast in producing upscaled images and is comparable to other image upscaling methods like FSRCNN, but still there is scope of improvement in terms of image quality. The proposed method being lightweight may help in implementing real time application of image upscaling like upscaling video files which we tend to implement in the future and try to minimize the loss.

ACKNOWLEDGMENT

We thank Dr. Ashish Vanmali and other Professors from Vidyavardhini's College of Engineering and Technology for their continued support and guidance. We would like to thank them for sharing their wisdom during the course of writing this manuscript.

REFERENCES

[1] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb 2016.

[2] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," *CoRR*, vol. abs/1608.00367, 2016. [Online]. Available: http://arxiv.org/abs/1608.00367

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

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[5] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *CoRR*, vol. abs/1609.04802, 2016. [Online]. Available: http://arxiv.org/abs/1609.04802

[6] B. Madhukar and R. Narendra, "Lanczos resampling for the digital processing of remotely sensed images," in *Proceedings of International Conference on VLSI, Communication, Advanced Devices, Signals & Systems and Networking (VCASAN-2013)*. Springer, 2013, pp. 403–411.

[7] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 807–814.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[9] A. Vanmali and V. Gadre, "Visible and nir image fusion using weight-map-guided laplacian–gaussian pyramid for improving scene visibility," *Sadhana - Academy Proceedings in Engineering Sciences*, vol. 42, pp. 1–20, 06 2017.

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

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

[12] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, May 2011.

[13] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 1122–1131.

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

[15] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," vol. 6920, 06 2010, pp. 711–730.

[16] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[17] J. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 5197–5206.