Experimentally Proven Bilateral Blur on SRCNN for Optimal Convergence

Divya Mishra[1(✉)], Tayyibah Khanam[2], and Ivanshu Kaushik[3]

[1] Jamia Millia Islamia, New Delhi, India
dvmshr946@gmail.com
[2] Aligarh Muslim University, Aligarh, India
tayyibahkhanam@zhcet.ac.in
[3] Birla Institute of Technology and Science, Pilani, India
ivanshukaushik918@gmail.com

Abstract. Image super-resolution is a contentious issue in developing computer vision applications such as satellite imaging, graphics industry, medical diagnostics, and real-time scene monitoring for security and safety. CNN-based image super-resolution algorithms are the simplest and most resource-efficient in deep learning. The original SRCNN framework, on the other hand, incorporated Gaussian blur, which takes a long time to converge. This is not only computationally expensive, but it also extends the training time. In light of this, we retrained original SRCNN utilizing widely available blurring techniques and observed that the bilateral filter beats the Gaussian filter at optimal convergence. Our goal is to preserve the original SRCNN features while reducing training time and computational resources and hence, speed up the architecture with optimum number of epochs. This is, to the best of our knowledge, the first study of its kind, and no other study has implemented this alternative blurring training on SRCNN and ranked the best of them for image super-resolution.

Keywords: Blurring techniques · Image quality metrics · Super-resolution · SRCNN

1 Introduction

Super-resolution is the approach of creating a high-resolution(HR) image from a low-resolution(LR) image. Multiple steps towards super resolution have been included in sparse-coding based methods [27,28], such as overlapping densely cropped patches followed by mean subtraction and normalisation as a preprocessing step for further encoding image patches by Low resolution dictionary to recover high resolution patches. Furthermore, these existing approaches lack optimization. The above-mentioned sequential methodologies are analogous to the behavior of a deep convolution neural network [12]. This motivation prompted the authors to propose the SRCNN [4], the first convolution neural network-based super-resolution model. SRCNN is a multi-layer, end-to-end

I. Woungang et al. (Eds.): ANTIC 2021, CCIS 1534, pp. 502–516, 2022.
https://doi.org/10.1007/978-3-030-96040-7_39

deep convolutional network that takes a low-resolution image and produces a high-resolution image. The network is lightweight, with only three CNN layers and a small number of filters. It can execute on a CPU as well. Aside from that, the network is quicker than prior sparse example-based approaches, which are hampered by a long chain of encoded dictionaries.

Bicubic interpolation is used to first upscale the low resolution (LR) input picture. Then, between high resolution and low resolution samples, a mapping function is learned that consists of three operations: generating feature maps through patch extraction and representation, non-linear mapping among high-dimensional vectors that represent a high resolution image, and finally aggregation of all high-dimensional vectors to reconstruct a super resolution image. They discovered that perceptual quality may be enhanced much more if:

- For training purposes, a large number of datasets are available.
- When a dense network is utilised with more layers than three, and the number of filters is increased, the Peak signal to noise ratio(PSNR) improves considerably, as shown in Table 1, where F1, F2 and F3 are filter size in consecutive network layers of SRCNN [4].
- To achieve more super-resolved results, the network could be able to control all three channels instead of only the Y channel.

Table 1. PSNR values on increasing filter size in SRCNN

Filter size in 3 layers	PSNR
F1 = 9, F2 = 1, F3 = 5	32.52
F1 = 11, F2 = 1, F3 = 7	32.57
F1 = 9, F2 = 3, F3 = 5	32.66
F1 = 9, F2 = 5, F3 = 5	32.75

In Image pre-processing, most of the models utilise a Gaussian filter. In 2019, authors [1] tried to improve results for same SRCNN model by introducing a change in existing Relu activation function as modified and bilateral ReLU. This helps them in overall improvement in image quality. Another approach, is used by presenting a resilient loss function merged with MSE used in SRCNN based on the Canny operator's preservation of edges [21] which shows better results on both PSNR and SSIM. The authors [14] modified SRCNN using color feature based image super-resolution algorithm for underwater image applications. The author proposes a bilateral up-sampling network [30] which consists of a bilateral up-sampling filter used for single image super-resolution with arbitrary scaling factors. Our plan is to put three additional blurring methods: the average blur, median blur and bilateral blur to test the original SRCNN [4] model and retrain them thrice on each blur filter. For all five pictures evaluated, we discovered that bilateral outperforms than Gaussian and rest blurs for a smaller number of epochs (nearly 25). This also implies that if the number of epochs is kept around 25, we may utilise bilateral for better outcomes.

The following is a breakdown of the paper's structure: Sect. 2 briefly introduce widely used blurring techniques image processing techniques. Section 3 introduced by common image quality metrics like Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). Section 4 accumulates conventional interpolation based methods for image super-resolution. Section 5 describe about the first deep learning based image SR model and how different existing blurs can be used in that SRCNN. Section 6 summarize the dataset used and evaluation result on each blur for all 5 test images. Section 7 finally discusses the conclusion and the potential of future work.

2 Blurring Techniques Widely Used in Image Processing Techniques

Blurring is a word that is commonly used to describe something that is smeared or unclear. This term is quite frequently used by the Image Processing and Computer Vision fraternity, with the goal to refer smoothing or de-sharpening of images.

A crucial step in image processing and computer vision tasks is to identify distinct objects [25], measure their sizes, to find influence on face Recognition Performance [10], in image restoration task [9,19,24] obtain the shape matrices etc. In such a situation, the earliest phases of object identification and localisation need 'edge detection. Edges define a strong divide between two visually distinct sets of pixels, resulting in abrupt variations in pixel intensity all along the edge.

The objective of blurring an image is to smooth out these abrupt variations in pixel intensities, which can be simply translated to smoothing out the image. The Computer Vision community refers to blurring or smoothing as applying a low pass filter to an image, which basically eliminates the "noise" while preserving other features of the image intact.

2.1 Median Blur

Median filtering [2] is a class of non-linear edge preserving filtering that is frequently employed in digital signal processing. The median filter works by traversing pixel by pixel through each image and replacing each pixel with the median of adjacent pixels. The pattern of neighbours is referred to as the "window," and it slides across the entire image, pixel by pixel. While the window for one-dimensional data, such as electronic signals, must encompass all entries over a certain radius or ellipsoidal region, the window for higher-dimensional data, such as images, must include all entries within a given radius. The median is generated by sorting all of the pixel values in the window into numerical order, then replacing the pixel in concern with the middle (median) pixel value.

2.2 Average Blur

The average filter [23] operates similarly to the median blur in that it moves across the image pixel by pixel, replacing each value with the average value of neighbouring pixels, including itself. Because the filter is the simplest of all existing filters, it contributes equally to the formation of the final image, as illustrated in Eq. 1.

$$K = \frac{1}{9} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{1}$$

2.3 Gaussian Blur

A Gaussian filter can be considered as a non-uniform linear low-pass filter that preserves low spatial frequency and reduces image noise and speckles [5,18]. To avoid erroneous edge detection, such a denoising operation is necessary to eliminate the particularly high frequency components according to a given threshold. It's usually done by using a Gaussian kernel to convolve an image. As a result, Gaussian blur calculates a local average of intensities at each position. Gaussian filtering, as a consequence, is a weighted average of the intensity of neighbouring locations, with the weight decreasing as the distance from the centre (p) rises, expressed by Eq. (2). The Gaussian kernel in 2-D form is expressed in Eq. (3), where sigma is the standard deviation of the distribution and controls the variance along a mean value of a Gaussian distribution. The degree of blurring may thus be adjusted by varying the standard deviation and threshold values to detect the most general edges.

$$Gb[I]_p = \sum_{q \in S} G_\sigma(||p - q||) I_q \tag{2}$$

$$G_{2D}(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3}$$

Consider this visualisation of a two-dimensional Gaussian function in Fig. 1 to better understand how the equations above operate.

While the standard deviation or sigma determines the width or 'spread' of the curve (larger the deviation, more the spread and flatter the shape), the height is determined by the mean of the normal distribution, or the weight given to the underlying pixel in the kernel.

The basic operation in linear image filtering is convolution by a positive kernel such as in Gaussian blur. Thus, The linear filter Gaussian blur [7] is an example of one that can be constructed quickly and efficiently.

2.4 Bilateral Blur

The bilateral filter [11], like the Gaussian convolution, is defined as a weighted average of pixels. While the Gaussian filter smoothes away noise or textures

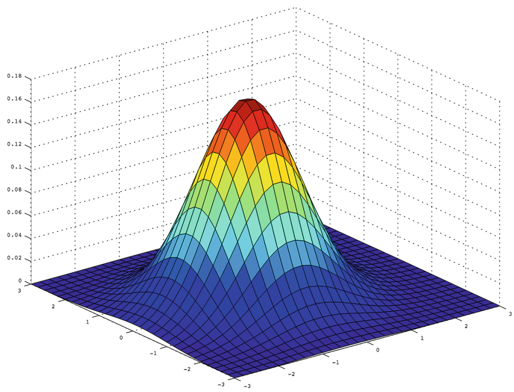

Fig. 1. Two dimensional Gaussian density function

without maintaining edges, the bilateral filter does so while taking intensity fluctuations into account. Bilateral filtering is based on the idea that two pixels are close to one another not only if their spatial locations are similar, but also if their photometric ranges are also similar. Equation (4) and Eq. (5) define the Bilateral filter and its normalization factor respectively.

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(||p-q||) G_{\sigma_r}(I_p - I_q) I_q \tag{4}$$

$$W_p = \sum_{q \in S} G_{\sigma_s}(||p-q||) G_{\sigma_r}(I_p - I_q) I_q \tag{5}$$

The filtering intensity is thus controlled by two weight parameters - σ_s and σ_r. As σ_r increases, the bilateral filter approaches purely gaussian characteristics and increasing σ_s smoothens larger features. The weights are multiplied in bilateral filtering, which implies that no smoothing happens as soon as one of the weights approaches zero. G_s is a spatial Gaussian that minimizes the influence of distant pixels, whereas G_r is a range Gaussian that minimizes the influence of pixels q with an intensity value differing from I_p as shown in Fig. 2.

3 Image Quality Metrics

Image enhancement, sometimes referred to as enhancing a digital image's visual quality, is a subjective practice. It is possible that the assertion that one approach creates a higher-quality image varies from person to person. As a result, quantitative and empirical measurements for comparing the impact of image enhancement algorithms on image quality must be established.

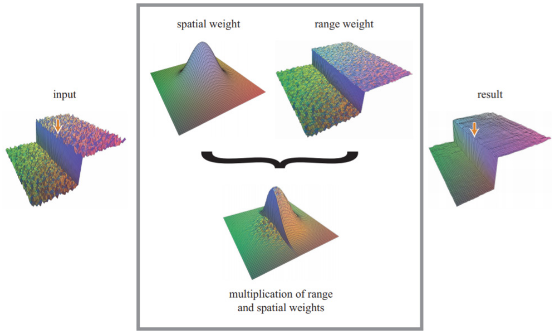

Fig. 2. Filtering process through bilateral blur

3.1 Mean Squared Error

In the recent past, the mean squared error (MSE) has been widely utilised for most practical uses. It allows us to compare our original image's "true" pixel values to our degraded image. The MSE is the square root of the "errors" in our actual and noisy images. The error is the difference in between values of the original image and the values of the degraded image. The MSE between an original image matrix f, and the degraded image matrix g is represented by Eq. (6):

$$MSE = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}||f(i,j) - g(i,j)||^2 \tag{6}$$

3.2 Peak Signal to Noise Ratio

PSNR (peak signal-to-noise ratio) is a term for the ratio of a signal's greatest potential value (power) to the strength of distorting noise that influences its representation quality. The PSNR is generally represented in logarithmic dB because many signals have a high dynamic range (the difference between the greatest and lowest possible values of a changing quantity). When dealing with images instead of signals, PSNR may be thought of as a more advanced version of MSE. The MSE is a measure of the cumulative squared error between the original and degraded image, whereas the PSNR is a measure of the peak error between the original and degraded image. Equation 7 represents the PSNR value of the pair of images (f, g) in terms of the MSE described in Eq. (6).

$$PSNR = 20log_{10}(\frac{MAX_f}{\sqrt{MSE}}) \tag{7}$$

where MAX_f is the maximum signal value that exists in our original known to be "good image".

Although it is self-evident that a greater PSNR (equivalent to a lower MSE) indicates a better quality of the degraded picture g and hence a superior reconstruction technique [6]. As a result, the metric's range of validity must be utilised with extreme caution. PSNR has difficulties in assessing the quality of two virtually similar pictures with a near-zero MSE. Because division by zero is undefined, PSNR can't be quantified as a meaningful quality evaluation metric in all circumstances because it relies only on numerical comparisons. PSNR has lately been found to perform badly in contrast to other quality evaluation methods [8,26] such as the Structural Similarity Index Metric (SSIM) which also considers biological factors of human visual system along with numeric comparisons.

3.3 Structural Similarity Index Metric

As mentioned in the previous section, SSIM [26] is a perception-based metric that takes into consideration not only the difference in structural information between two pictures, but also fundamental perceptual notions like luminance masking and contrast masking. The concept of structural information relates to the assumption that pixels, especially when they are spatially comparable, exhibit substantial interdependencies. These dependencies hold vital information about the structure of the visual scene's entities. For example, Contrast masking lowers the visibility of image distortions in areas with substantial activity or "texture" in the image, whereas luminance masking reduces the visibility of image distortions in bright areas (in this context). SSIM focuses on measuring the perceptual difference between the two available image matrices rather than determining which image is superior in terms of visual appearance.

Calculation of the SSIM index is done on various image windows. Between two windows x and y of common size N * N, the measure between them is represented by Eq. (8).

$$SSIM(x,y) = [l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma}] \tag{8}$$

where the SSIM formula is a weighted combination of three comparison measurements between the samples of x and y: luminance (l), contrast (c) and structure (s), where α, β and γ represent the individual weights of all three components respectively.

4 Image Super Resolution

The computer vision community has conducted substantial research on the technique of upscaling and enhancing an image, popularly known as super resolution. The goal is to convert a lower-resolution original image to a higher-resolution image that is perceptually pleasing and realistic.

In the coming sections, we will discuss some of the initial upscaling algorithm [15], all based on linear methods of interpolation. Interpolation is the problem of approximating the value of a function for a non-given point in some space given the value of that function in points around (neighbouring) that point.

4.1 Nearest Neighbors Interpolation

The simplest method of interpolation is the Nearest Neighbour interpolation [13, 20, 22], also commonly known as the 'box filter'. The closest neighbour method selects the value of the nearest point while neglecting the values of neighbouring points, yielding a piecewise constant interpolant. Rather of utilising weighting criteria to determine an average value or complicated procedures to provide an intermediate value, this method simply identifies the "nearest" neighbouring pixel and assumes its intensity value.

4.2 Bilinear Interpolation

Also known as the 'tent' function [15] is a re-sampling method that estimates a new pixel value by taking the distance weighted average of the four nearest pixel values. It is possible to interpolate functions of two variables (such as x and y) on a two-dimensional grid using a bilinear interpolation technique. By first conducting linear interpolation in one direction and then in the opposite direction, bilinear interpolation is achieved. This interpolation is exclusively linear along lines parallel to the X or Y axis, and quadratic along all other straight lines.

4.3 Bicubic Interpolation

Because the interpolated surface obtained from bicubic interpolation [3] is smoother than similar surfaces derived from bilinear or nearest-neighbor interpolation, bicubic interpolation is usually favoured over bilinear or nearest-neighbor interpolation in image resampling. This is because Bi-linear determines the output using four nearest neighbours, whereas Bi-cubic employs sixteen (4 * 4) neighbours, slowing down the process but providing a better upsampled rendition of the original image. To fill up the gaps, the techniques indicated above gather data from neighbouring pixels. But why don't these tried-and-true ways work? A well-known truth in data processing is that data can't be further processed to deliver any information that doesn't already exist, which simply translates to - data can't be further processed to provide any information that doesn't already exist. We can't upgrade images using data processing methodologies since they aren't predictive.

This is where a neural network comes into play perfectly. Based on the weights it learns throughout the training process on a wide set of images, a neural network learns to hallucinate features. The first one in this direction was SRCNN [4] which simply learns a mapping between a low resolution and a high-resolution image. The aim is to train a neural network using a dataset made up of high-resolution images that have been downscaled and then input into the neural network. Because this is a kind of self-supervised learning, we may compare the neural network generated image to the ground truth image by downscaling high resolution samples to low resolution samples before feeding them into the network.

The SRCNN paper thus simply minimizes the squared difference (Mean Squared Error/MSE) of the pixel values while training to calculate the loss. Mean Squared Error, on the other hand, cannot be labeled a strong loss function since it only evaluates pixel-wise variations rather than structural information. As a result, the Structural Similarity Index (SSIM), a superior gauge of perceptual quality, works better in this situation. SSIM has been utilised as a loss function for image restorations and other reasons by a few academics, despite its origins as a quality evaluation tool.

5 SRCNN

Prior to SRCNN, image restoration was accomplished using a technique called Sparse Coding. Sparse coding, built on a complex pipeline and mathematical algorithms, extracted overlapping patches from an image, projected those patches to a higher resolution space, and then aggregated these high resolution vectors to reconstruct the image. Due to the complexity of sparse coding based image restoration, the authors of SRCNN attempted at recreating an identiacal Convolutional Neural Network pipeline for ease and simplicity.

Non-predictive techniques are unable to predict details in an image, resulting in upscaling losses; here is when the positives of using a neural network come into play. Based on the weights it learns throughout the training process on a wide set of images, a neural network learns to hallucinate features. The first one in this direction was SRCNN [4] which simply learns a mapping between a low resolution and a high-resolution image. The aim is to train a neural network using a dataset made up of high-resolution images that have been downscaled and then input into the neural network. Because this is a kind of self-supervised learning, we may compare the neural network generated picture to the ground truth image by downscaling high resolution samples to low resolution samples before feeding them into the network. To compute the loss, the SRCNN study simply minimises the squared difference (Mean Squared Error/MSE) of the pixel values while training.

5.1 Architecture Details

The SRCNN network is a short depth three-convolutional layer network. Each convolutional layer is responsible for a distinct function. The first convolutional layer serves as a low-level feature extractor, responsible for patch extraction and representation, while the second convolutional layer performs non-linear mapping and the third reconstructs the up-sampled image as shown in the Fig. 3.

The model receives as input a standard low resolution image created by adding Gaussian blur to the original high resolution image. When training, the output is set to the appropriate high-resolution image. In this case, the error is the difference between the ground truth high resolution image and the generated high resolution image from neural network.

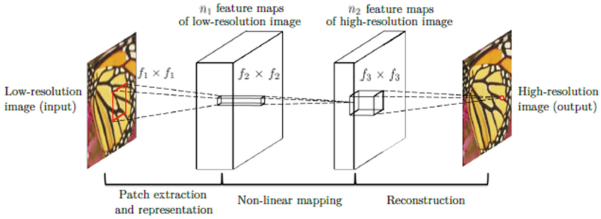

Fig. 3. Basic architecture of SRCNN [4]

The first convolutional layer separates or filters different locations from the low-resolution input image in an overlapping manner and turns them into a high-dimensional vector for further processing. The recovered high dimensional vector output of layer 1 is then nonlinearly mapped by the second convolutional layer to another high dimensional vector. Finally, the third convolutional layer reconstructs the original image by combining the aforementioned vectors into a high-quality image.

5.2 Modified-SRCNN

Advantages of Using Less Epochs: In machine learning, epochs represent the number of passes an algorithm makes across the whole training set. One epoch denotes a single run through the neural network algorithm of the whole training data forward and backward. Because gradient descent is an iterative process, updating the weights through a single epoch is unsatisfactory; hence, a sufficient number of epochs is required to make sure that our model's loss is properly mitigated. Although it is evident that more rounds of optimization would minimize the error on training data, there may come a point where the network becomes over-fit to the training data and begins to lose performance in terms of generalization to non-training (unseen) data. Furthermore, increasing the number of epochs not only increases training time but also becomes computationally expensive. As a result, a neural network is expected to be computationally efficient, converge as quickly as feasible in as few epochs as possible.

Experimenting with Various Blurring: In addition to the Gaussian blur utilized by the authors of SRCNN to create a low-dimensional, blurred image from a high-dimensional ground truth image, we performed our tests with three additional types of blurring filters: average blur, median blur, and bilateral blur. The objective of our research was to see how well a super-resolution neural network performed on differently blurred input images. While upsampling a low-resolution image, we were able to evaluate the relevance of linear and nonlinear filters, edges, neighbouring pixels, and other factors.

Reason of Why Bilateral? The Gaussian filter is an isotropic filter that smoothens out the whole image, including edges and high-frequency features. The bilateral filter, on the other hand, gives more human-like results even after blurring. It avoids smoothing off edges, curves, and other structural features that don't need to be. The reason for this is that a bilateral filter is a more modified version of a Gaussian filter. If two nearby pixel values are numerically close to each other and represent a comparable picture, the Gaussian coefficient is multiplied by a quantity close to 1 and the outcome is similar to the Gaussian blur. On the other hand, if two nearby pixel values are significantly different from one another, reflecting a sudden dissimilar scene in an image, a number close to 0 is multiplied by the Gaussian coefficient, turning off the Gaussian filter and resulting in bilateral filtering. Furthermore, the author [17] has already analysed these four frequently used blurring techniques and evaluated them on ten test pictures, revealing that bilateral blurring is the best approach on potential image quality measures PSNR, SSIM, and FSIM [29]. We are looking into the SRCNN model architecture leveraging that result.

6 Evaluation

6.1 Dataset

We ran our experiments on the Berkeley Segmentation Data Set 500, BSDS500 [16]. Originally developed for segmentation tasks, BSDS500 contains 200 natural images in the training set, 100 images in validation set and 200 images in the test set. The training and validation sets were used for training the modified SRCNN in our experiments, whereas the test set was used as the validation set. For testing purposes, we randomly picked 5 high resolution images from the internet as shown in Fig. 4.

(a) Image-1 (b) Image-2 (c) Image-3 (d) Image-4 (e) Image-5

Fig. 4. Randomly selected 5 test images

6.2 Evaluation Results on Test Images

Our modified SRCNN architectures is trained for 15×10^6 back-propogations till 400 epochs using Stochastic Gradient Descent (SGD) optimizer and MSE loss. Additionally, to preserve the input image's original dimensions, we used padding that was not used previously in the original SRCNN paper.

From the observation through Table 2, Table 3, Table 4, Table 5 and Table 6, the PSNR values for bilateral blur among all blurs experimented are best till 25 epochs. The result is showing same trend for all 5 test-images. The re-implemented SRCNN is assessed using three distinct LR blurring techniques: average, median, and bilateral, and a potential image quality metric such as PSNR. The greater the value of the PSNR measure, the better the image quality.

Table 2. PSNR values for Image-1

Epochs	Gaussian (original)	Average blur	Median blur	Bilateral blur (Our)
5	13.6546	31.2339	30.8291	**31.5442**
10	32.0823	32.3148	31.5506	**34.7294**
15	32.5153	33.7281	32.8810	**38.2987**
20	33.4163	35.4244	34.7412	**38.6823**
25	34.3733	37.6884	36.5833	**38.7579**

Table 3. PSNR values for Image-2

Epochs	Gaussian (original)	Average blur	Median blur	Bilateral blur (Our)
5	12.3813	23.1625	23.8009	**26.2090**
10	25.0341	26.1183	26.2064	**27.6182**
15	26.6046	26.6944	26.8331	**28.7722**
20	26.9624	27.2860	27.4676	**28.8853**
25	27.4290	27.8657	27.8763	**28.8717**

Table 4. PSNR values for Image-3

Epochs	Gaussian (original)	Average blur	Median blur	Bilateral blur (Our)
5	12.8546	22.5428	22.4944	**24.0124**
10	23.1272	24.0977	24.0283	**24.5656**
15	24.4383	24.3235	24.3757	**25.0244**
20	24.6514	24.5969	24.7149	**25.1123**
25	24.8785	24.8410	24.9768	**25.1350**

Table 5. PSNR values for Image-4

Epochs	Gaussian (original)	Average blur	Median blur	Bilateral blur (Our)
5	21.7155	23.9870	23.8865	**25.4093**
10	24.7007	25.5559	25.6138	**26.3961**
15	25.8845	26.0202	26.2040	**27.1997**
20	26.2099	26.4581	26.7967	**27.3426**
25	26.5939	26.9021	27.2382	**27.3803**

Table 6. PSNR values for Image-5

Epochs	Gaussian (original)	Average blur	Median blur	Bilateral blur (Our)
5	12.4835	29.0525	29.1541	**30.9659**
10	29.8682	31.2727	31.1806	**32.5813**
15	32.0604	32.0230	32.1268	**34.8543**
20	32.1702	33.0057	33.2220	**35.0118**
25	32.7468	34.6621	34.6660	**35.0745**

7 Conclusion and Future Work

Image super-resolution is a popular and in-demand application in the field of image processing, but resource management is a significant difficulty in deep learning-based research. To resolve the convergence time, we conducted an experiment by training original SRCNN model thrice by using three different widely used blurring techniques. Those three blurring filters are average or box blur, median and bilateral blur. The goal of this experiment is to obtain the best outcomes with the minimal number of running epochs possible in order to conserve computational resources and time without compromising on image quality. On five randomly selected images, we utilised PSNR as a conventional image quality statistic to evaluate the findings. The least effective blur is as expected the average blur and the best is bilateral blur. This research will be useful not just for convolution neural network-based designs, but also for generative models such as Generative Adversarial Networks (GAN)-based super-resolution models in the future. New filters with features equivalent to bilateral filters can be designed as a result of this experiment, allowing them to converge even quicker than bilateral filters. In this way, we can actively contribute to the growing body of knowledge that aids practitioners, academics, and developers in choosing optimal blurring algorithms for image super-resolution with the minimum number of epochs.

References

1. Ahn, H., Chung, B., Yim, C.: Super-resolution convolutional neural networks using modified and bilateral ReLU. In: 2019 International Conference on Electronics, Information, and Communication (ICEIC), pp. 1–4 (2019). https://doi.org/10.23919/ELINFOCOM.2019.8706394
2. Chang, C., Hsiao, J., Hsieh, C.: An adaptive median filter for image denoising. In: 2008 Second International Symposium on Intelligent Information Technology Application, vol. 2, pp. 346–350 (2008). https://doi.org/10.1109/IITA.2008.259
3. Dengwen, Z.: An edge-directed bicubic interpolation algorithm. In: 2010 3rd International Congress on Image and Signal Processing, vol. 3, pp. 1186–1189, October 2010
4. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks (2015)
5. Gedraite, E.S., Hadad, M.: Investigation on the effect of a Gaussian blur in image filtering and segmentation. In: Proceedings ELMAR-2011, pp. 393–396 (2011)
6. Gupta, P., Srivastava, P., Bhardwaj, S., Bhateja, V.: A modified PSNR metric based on HVS for quality assessment of color images. In: 2011 International Conference on Communication and Industrial Application, pp. 1–4, December 2011. https://doi.org/10.1109/ICCIndA.2011.6146669
7. He, H., Siu, W.C.: Single image super-resolution using gaussian process regression. In: CVPR 2011, pp. 449–456, June 2011. https://doi.org/10.1109/CVPR.2011.5995713
8. Horé, A., Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: 2010 20th International Conference on Pattern Recognition, pp. 2366–2369 (2010). https://doi.org/10.1109/ICPR.2010.579
9. Huang, H.Y., Tsai, W.C.: Blurred image restoration using fast blur-kernel estimation. In: 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 435–438, August 2014
10. Knežević, K., Mandić, E., Petrović, R., Stojanović, B.: Blur and motion blur influence on face recognition performance. In: 2018 14th Symposium on Neural Networks and Applications (NEUREL), pp. 1–5, November 2018. https://doi.org/10.1109/NEUREL.2018.8587028
11. Kornprobst, P., Tumblin, J., Durand, F.: Bilateral filtering: theory and applications. Found. Trends Comput. Graph. Vis. **4**, 1–74 (01 2009). https://doi.org/10.1561/0600000020
12. LeCun, Y., et al.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989). https://doi.org/10.1162/neco.1989.1.4.541
13. Li, F., Shang, C., Li, Y., Yang, J., Shen, Q.: Interpolation with just two nearest neighboring weighted fuzzy rules. IEEE Trans. Fuzzy Syst. **28**(9), 2255–2262 (2020). https://doi.org/10.1109/TFUZZ.2019.2928496
14. Li, Y., et al.: Underwater image high definition display using the multilayer perceptron and color feature-based SRCNN. IEEE Access **7**, 83721–83728 (2019). https://doi.org/10.1109/ACCESS.2019.2925209
15. Lukin, A., Krylov, A., Nasonov, A.: Image interpolation by super-resolution (2010)
16. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings 8th International Conference Computer Vision, vol. 2, pp. 416–423, July 2001

17. Mishra, D., Mishra, A.: Comparison of blurring techniques for generative adversarial network-based superresolution models: an empirical study. Int. J. Res. Eng. Appl. Manag. **07** (2021)
18. Misra, S., Wu, Y.: Chapter 10 - machine learning assisted segmentation of scanning electron microscopy images of organic-rich shales with feature extraction and feature ranking. In: Misra, S., Li, H., He, J. (eds.) Machine Learning for Subsurface Characterization, pp. 289–314. Gulf Professional Publishing (2020). https://doi.org/10.1016/B978-0-12-817736-5.00010-7. https://www.sciencedirect.com/science/article/pii/B9780128177365000107
19. Motohashi, S., Nagata, T., Goto, T., Aoki, R., Chen, H.: A study on blind image restoration of blurred images using R-MAP. In: 2018 International Workshop on Advanced Image Technology (IWAIT), pp. 1–4, January 2018. https://doi.org/10.1109/IWAIT.2018.8369650
20. Ni, K.S., Nguyen, T.Q.: Adaptable k-nearest neighbor for image interpolation. In: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1297–1300, March 2008. https://doi.org/10.1109/ICASSP.2008.4517855
21. Pandey, R.K., Saha, N., Karmakar, S., Ramakrishnan, A.G.: MSCE: an edge preserving robust loss function for improving super-resolution algorithms (2018)
22. Rukundo, O., Maharaj, B.T.: Optimization of image interpolation based on nearest neighbour algorithm. In: 2014 International Conference on Computer Vision Theory and Applications (VISAPP), vol. 1, pp. 641–647, January 2014
23. Singh, T.: Comparative analysis of image deblurring techniques. Int. J. Comput. Appl. **153**, 39–44 (2016). https://doi.org/10.5120/ijca2016912068
24. Veeramani, T., Rajagopalan, A.N., Seetharaman, G.: Restoration of foggy and motion-blurred road scenes. In: 2013 IEEE International Conference on Image Processing, pp. 928–932, September 2013. https://doi.org/10.1109/ICIP.2013.6738192
25. Wang, R., Li, W., Qin, R., Wu, J.: Blur image classification based on deep learning. In: 2017 IEEE International Conference on Imaging Systems and Techniques (IST), pp. 1–6, October 2017. https://doi.org/10.1109/IST.2017.8261503
26. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861
27. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, June 2008. https://doi.org/10.1109/CVPR.2008.4587647
28. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. IEEE Trans. Image Process. **19**(11), 2861–2873 (2010). https://doi.org/10.1109/TIP.2010.2050625
29. Zhang, L., Zhang, L., Mou, X., Zhang, D.: FSIM: a feature similarity index for image quality assessment. IEEE Trans. Image Process. **20**(8), 2378–2386 (2011)
30. Zhang, M., Ling, Q.: Bilateral upsampling network for single image super-resolution with arbitrary scaling factors. IEEE Trans. Image Process. **30**, 4395–4408 (2021). https://doi.org/10.1109/TIP.2021.3071708