

# Resizing and cleaning of histopathological images using generative adversarial networks

Gaffari Çelik<sup>a,\*</sup>, Muhammed Fatih Talu<sup>b</sup>

<sup>a</sup> Agri Ibrahim Cecen University, Department of Computer Technology, Agri, Turkey

<sup>b</sup> Inonu University, Department of Computer Science, Malatya, Turkey



## ARTICLE INFO

### Article history:

Received 28 May 2019

Received in revised form 31 August 2019

Available online 17 September 2019

### Keywords:

SRGAN

Noise cleaning

Image resizing

Bicubic

Camelyon17

## ABSTRACT

Bilinear and Bicubic interpolation techniques are frequently used to increase image resolution. These techniques with data modeling approach are replaced by intelligent systems that can learn automatically from data. SRGAN is a modern Generative Adversarial Network developed as an alternative to classical interpolation techniques. Its ability to produce images in super resolution has attracted the attention of many researchers.

In this study, noise elimination performance of super resolution generative adversarial network (SRGAN) with image magnification was investigated. The results of the noise cleaning were compared with the classical approaches (mean, median, adaptive filters). SSIM, PSNR and FFT\_MSE metrics were evaluated in experimental studies using images in the data set Camelyon17. When the results were evaluated, it was observed that SRGAN was superior to the classical approaches not only in increasing the resolution but also in the noise cleaning area.

© 2019 Published by Elsevier B.V.

## 1. Introduction

Processing on low resolution (LR) or noisy images in the image and video processing area is a major problem [1]. This negatively affects the performance of algorithms such as object detection, segmentation, modeling and tracking. In medical applications, low resolution images have a negative effect on the diagnosis and diagnosis of the disease.

High-resolution images can be obtained using high-cost hardware devices. In addition, low-resolution images from low-cost imaging devices can be enlarged with the help of specific algorithms. Thus, the cost of hardware can be saved significantly.

There are many methods to increase image resolution [2]. Pixel multiplexing, nearest neighbor, Lanczos [3], Cubic [4], bilinear and bicubic interpolation [5,6] are some of these. In recent years, current approaches have been proposed in the field of increasing the resolution with the development of deep learning approaches [7]. In the resolution area, convolutional neural network (CNN) models [8,9] and generative adversarial networks (GANs) [10–12] have been introduced. Of course, the SRGAN architecture is one of the most modern techniques proposed to improve resolution [8].

## 2. Classical approaches to increasing resolution

### 2.1. Bilinear interpolation

With Bilinear interpolation technique, image sizes can be enlarged as much as desired. The following linear equation is used in the process of magnifying a small image [6].

$$u(x, y) = ax + by + cxy + d \quad (1)$$

\* Corresponding author.

E-mail addresses: [gcelik@agri.edu.tr](mailto:gcelik@agri.edu.tr) (G. Çelik), [fatih.talu@inonu.edu.tr](mailto:fatih.talu@inonu.edu.tr) (M.F. Talu).

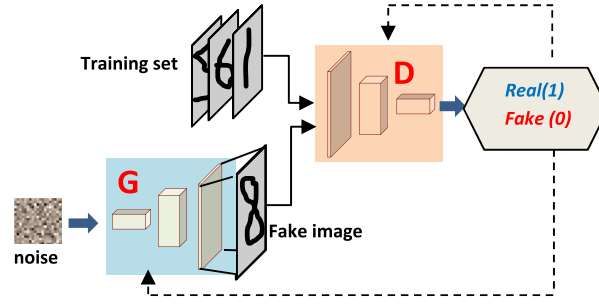


Fig. 1. GAN architecture [21,22].

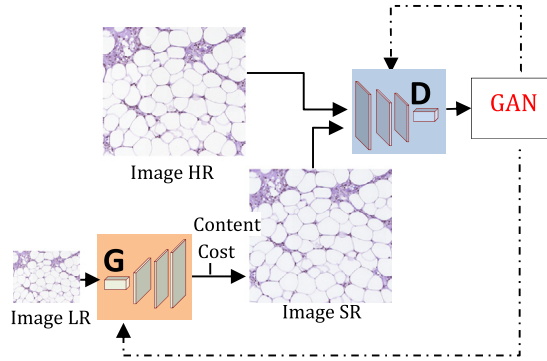


Fig. 2. SRGAN architecture [22].

$u(x, y)$  represents the pixel color value,  $x$  and  $y$  pixel position information. In order to obtain four equations for four different pixels, the algebraic solution of the system of equations is made and the coefficients ( $a$ ,  $b$ ,  $c$ ,  $d$ ) are calculated. Thus, the new color value at any  $x$  and  $y$  position can be calculated [6].

## 2.2. Bicubic interpolation

In bicubic interpolation method, horizontal, vertical and diagonal derivatives of the image are obtained first. A total of four different images occur with the derivatives. In the next stage, 16 equations of 16 unknowns are generated using the color values in each image of four different pixel points [5]. After the algebraic solution of the equation system, the unknown 16 coefficients  $a_{i,j}$  is calculated. Finally, the color value in the desired  $(x, y)$  coordinate is calculated using the following equation [6].

$$P(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j \quad (2)$$

## 3. Super-Resolution Generative Adversarial Network (SRGAN)

Generative adversarial networks (GANs) produce fake images that resemble (real) images in the training set. GANs consist of a Generator (G) and a discriminator (D) of the network. G takes the noise vector as input and tries to produce images similar to the actual images in the training set. D takes the actual images from the training set as input and the fake images produced by G and tries to distinguish which one is real/fake [13]. GAN architectures successful results in various fields such as image pattern transfer [14], noise removal and refilling [15], facial aging/rejuvenation [16], text image production [17], resolution enhancement [18], tissue transfer [19], production of financial time series [20]. The classical GAN architecture is given in Fig. 1.

SRGAN [18] uses GAN architecture to produce high resolution images. The low-resolution (LR) equivalents of high-resolution images (HR) are obtained before starting the training process.

G, takes LR images as input and produces super resolution images (SR). D is used to distinguish between HR and SR images. The GAN cost to train G and D is spread backwards. The architecture of SRGAN is shown in Fig. 2.

**SRGAN algorithm:**1. **For epoch do**

2. Obtain higher dimensional  $I^{SR}$  form  $I^{LR}$ :  $I^{SR} = G(I^{LR})$
3. Export the high-dimensional real ( $I^{HR}$ ) and fake ( $I^{SR}$ ) image as input to  $D$ .
4. Train  $D$  (update weights):

$$\nabla_{\theta_d} = \frac{1}{m} \sum_{i=1}^m [\log D(I^{HR}) + \log(1 - D(I^{SR}))]$$

5. Train  $G$  (update weights):

$$\nabla_{\theta_g} = 10^{-3} * \nabla_{\theta_{adversarial}} + \nabla_{\theta_{content}}$$

$$\nabla_{\theta_{adversarial}} = \frac{1}{m} \sum_{i=1}^m -\log(D(I^{SR}))$$

$$\nabla_{\theta_{content}} = \|VGG19(I^{HR}) - VGG19(I^{SR})\|_2$$

6. **endFor**

The feature that distinguishes SRGAN from the classical GAN architectures is the update logic of the weights in the  $G$  network. In classic GAN architectures, only the adversarial costs ( $\nabla_{\theta_{adversarial}}$ ) are used to update the weights in the  $G$  network in order for the  $G$  to fool  $D$ . However, the cost of content ( $\nabla_{\theta_{content}}$ ) is added in the SRGAN algorithm in addition to the cost of adversarial. The content cost refers to the perceptual similarity between the images  $I^{HR}$  and  $I^{SR}$  given as input to the  $D$  network. The VGG19 convolutional network is used to calculate the perceptual similarity values of the images. Accordingly, Euclidean distance is calculated between new images obtained by filtering both images from the VGG19 network.

**4. Noise removal methods**4.1. *Median filter*

When moving a mask over the image, the corresponding pixels of the mask are sorted from large to small and the median pixel is assigned to the center pixel, which corresponds to the mask [23]. Median filter for a given  $I$  image [24]:

$$I_{mf}(i, j) = \text{median}\{I(i + r, j + s), (r, s) \in W\} \quad (3)$$

where  $(i, j) \in (1, 2, \dots, H) \times (1, 2, \dots, L)$ ,  $H$  ve  $L$  gives the width and height of the given image.  $W$  shows the coordinates of the mask.

4.2. *Mean filter*

In the mean filter, the original density value of a pixel in the image is changed by the average of neighboring pixels. This causes soft pixels to soften. For this reason,  $3 \times 3$ ,  $5 \times 5$  single number masks are convoluted with the image. As the mask size increases, the softening effect will increase, resulting in a blurred image [25].

4.3. *Adaptive filter*

The difference of the adaptive filter from the mean and median filters is that they automatically adjust the filter coefficients according to the local content in the image [23]. Accordingly, it reduces the filter coefficients when it reaches an area where the edge density is high, increases the filter coefficient when it reaches a flat area without edge information. In this way, it protects both edges and performs more powerful noise cleaning than the mean and median filter.

**5. Preparation of data set**

In this study, the images of the data set Camelyon17 [26] were used. Camelyon17 contains approximately 3TB of histopathological images of breast cancer. These images were obtained by digital pathology technique by five medical centers (RUMC, UMCU, RST, CWZ and LPON) based in the Netherlands. In the digital pathological imaging process, the microscope produces high resolution images (160 nm per pixel), while the microscope is automatically moved. This allows the entire surface to be photographed on the preparation. Consecutive images are combined to obtain a single image (whole slide image-WSI) of enormous size ( $200,000 \times 100,000$ ). The recorded high-resolution image is considered to be the level 0 of the pyramid, and images up to level 6 are recorded so that each top-level image is twice as small as the lower level. It is possible to copy images of any desired size to any size (see Fig. 3). Such images can be viewed with the help of special purpose programs such as ASAP, Open Slide [26,27]. In this study, a small data set was constructed by copying 5682 images from WSI images using Python programming language.

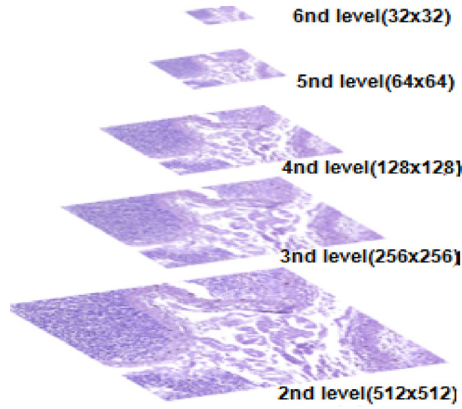


Fig. 3. Pyramid structure of WSI [27]. The image at each level is twice as small as the previous one.

## 6. Experimental application

Experimental studies A 20-core CPU consisting of 2 CPUs was performed on a server computer with NVIDIA QUARD 4000 GPU and 64 GB RAM components. Python and Matlab programming languages are used for applications.

In the first experimental study, the performances of the current method (SRGAN) with the classical method (BICUBIC) in the field of image magnification were observed. For this reason, 5682 high-resolution (HR) images of  $512 \times 512$  dimensions were obtained from the 2nd level of WSI images. In the second stage, each HR image was reduced to 2 times ( $256 \times 256$ ), 4 times ( $128 \times 128$ ), 8 times ( $64 \times 64$ ), 16 times ( $32 \times 32$ ) respectively, and low resolution (LR) images were obtained. In the third stage, LR images were magnified by algorithms based on magnification and similarities were calculated to the original image. The SRGAN architecture is trained in 30,000 iterations for each magnification. HR and LR images were scaled to  $[-1, 1]$ . For optimization algorithm Adam ( $lr = 0.0002$ ,  $\beta_1=0.5$ ) [28] optimization algorithm was used. This method combines the advantages of AdaGrad and RMSProp optimization algorithms. These are different convergence approaches. It varies according to the approach to the optimal value of the weights in the artificial system [28]. For G architecture, 16 residual blocks [29] were used. In deep networks, adding an extra layer can result in a higher education error and higher validation error. Residual blocks solve such problems by adding shortcut links collected with the output of convolution layers. Also for G architecture, 64 feature maps per block,  $3 \times 3$  kernel and convolution layer with stride 1, batch normalization (momentum = 0.8) [30] layer and RELU activation function were used. At the same time, depending on the magnification ratio, the up-sampling layer and 256 feature maps, the  $3 \times 3$  mask, the convolution layer with stride 1 and the RELU activation function were used. For the D architecture, 8 convolution layers, LeakyReLU ( $\alpha = 0.2$ ) activation function after each convolution layer, sigmoid activation function for the vectorization process and finally sigmoid activation function were used. The results are shown in Fig. 4. When the results are examined, the superiority of SRGAN result from BICUBIC is understood more clearly with the increase in magnification. While the results of SRGAN magnification are clearer and more comprehensible, the BICUBIC result is becoming more and more turbulent.

In the second experimental study, SRGAN method was compared with classical approaches in noise elimination process. The same architecture was used as the network architecture except for the magnification layer of the previous experimental work. Architecture has been trained in 30,000 iterations. In this experiment,  $256 \times 256$  images were taken from the 3rd level of the WSI images. These images were obtained by adding gauss noise with variance  $\sigma = 0.015$  and  $\sigma = 0.02$  to obtain SRGAN cleaning results with classical noise removal approaches (see Fig. 5). It was observed that SRGAN's performance in noise removal is better than classical methods.

Three different similarity metrics were used to compare the magnification and noise elimination results with the original image. These are the structural similarity metric SSIM [31], the signal-to-noise ratio PSNR [32] and the average square error of the frequency is FFT\_MSE. The SSIM similarity coefficient is expressed in Eqs. (4)–(7). The actual image is expressed as  $sr$  in SRGAN of the resulting image.

$$SSIM(hr, sr) = [I(hr, sr)]^\alpha \cdot [C(hr, sr)]^\beta \cdot [S(hr, sr)]^\gamma \quad (4)$$

$$I(hr, sr) = \frac{2\mu_{hr}\mu_{sr} + c_1}{\mu_{hr}^2 + \mu_{sr}^2 + c_1} \quad (5)$$

$$C(hr, sr) = \frac{2\sigma_{hr}\sigma_{sr} + c_2}{\sigma_{hr}^2 + \sigma_{sr}^2 + c_2} \quad (6)$$

$$S(hr, sr) = \frac{\sigma_{hrsr} + c_3}{\sigma_{hr}\sigma_{sr} + c_3} \quad (7)$$

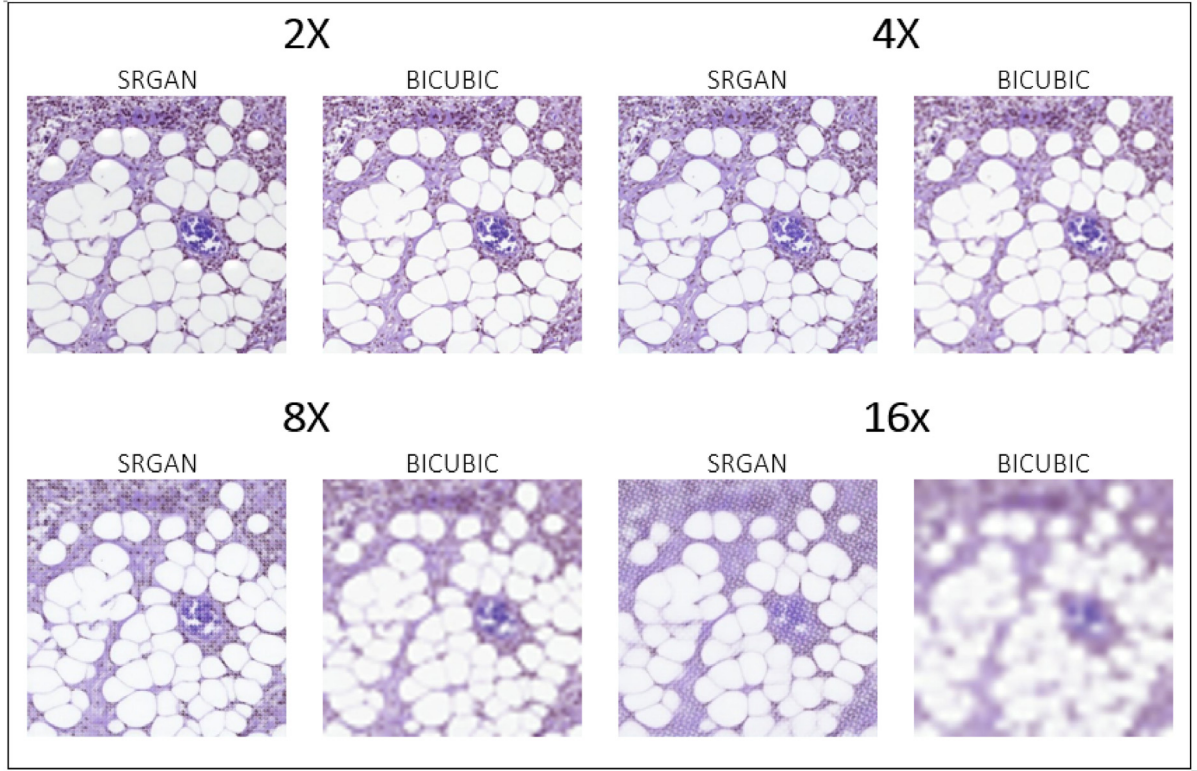


Fig. 4. Image magnification results.

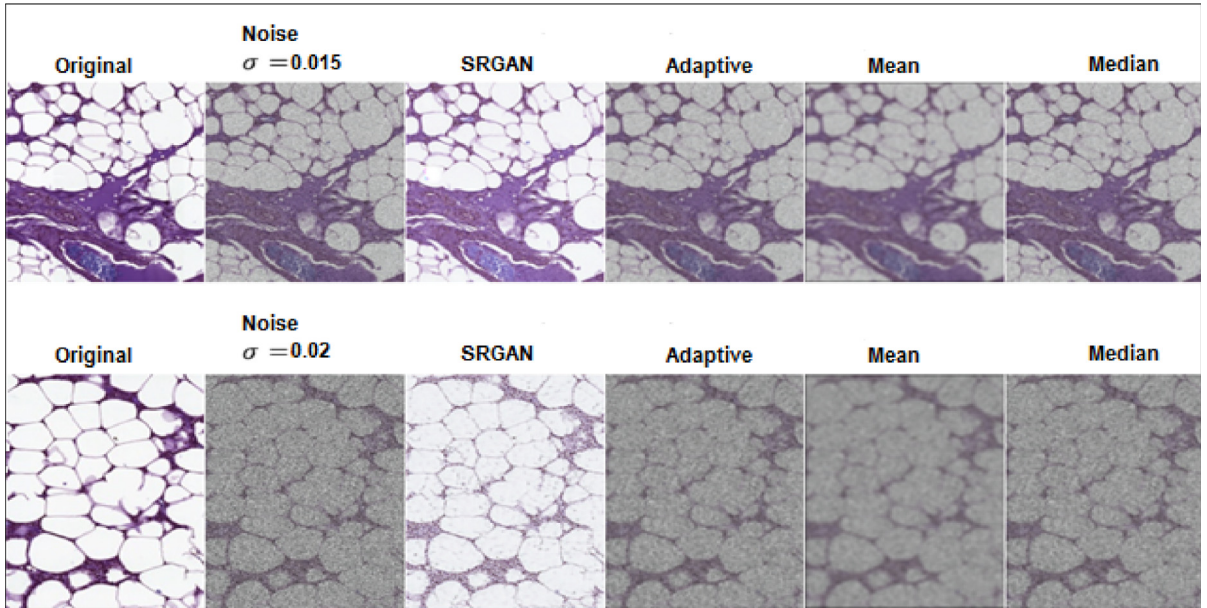


Fig. 5. Noise elimination results.

for  $hr$  and  $sr$  images,  $\mu_{hr}$ ,  $\mu_{sr}$  represents color mean,  $\sigma_{hr}$ ,  $\sigma_{sr}$  standard deviation and  $\sigma_{hrsr}$  covariance. The coefficients  $c_1$ ,  $c_2$ ,  $c_3$  are small scalar values to prevent zero-error. We use  $\alpha$ ,  $\beta$ ,  $\alpha$  coefficients to weight each term.



**Table 1**  
Image magnification results.

|      | 2X          |              |              |             |              |              | 4X          |              |              |             |              |         |
|------|-------------|--------------|--------------|-------------|--------------|--------------|-------------|--------------|--------------|-------------|--------------|---------|
|      | SRGAN       |              |              | BICUBIC     |              |              | SRGAN       |              |              | BICUBIC     |              |         |
| Imge | SSIM        | PSNR         | FFT_MSE      | SSIM        | PSNR         | FFT_MSE      | SSIM        | PSNR         | FFT_MSE      | SSIM        | PSNR         | FFT_MSE |
| Img1 | <b>0,91</b> | 27,24        | 16,95        | 0,87        | <b>27,88</b> | 16,83        | <b>0,78</b> | <b>24,88</b> | <b>16,99</b> | 0,71        | 24,09        | 17,79   |
| Img2 | <b>0,90</b> | 27,51        | 16,92        | 0,89        | <b>29,52</b> | <b>16,44</b> | <b>0,82</b> | <b>27,01</b> | <b>16,54</b> | 0,77        | 25,75        | 17,40   |
| Img3 | <b>0,91</b> | 27,92        | 16,86        | 0,91        | <b>31,32</b> | <b>15,99</b> | <b>0,84</b> | <b>28,20</b> | <b>16,19</b> | 0,82        | 27,28        | 17,00   |
| Img4 | <b>0,94</b> | 26,82        | 17,08        | 0,92        | <b>28,47</b> | <b>16,62</b> | <b>0,84</b> | <b>24,64</b> | <b>17,04</b> | 0,80        | 23,53        | 17,87   |
| Img5 | 0,92        | 26,42        | 17,30        | <b>0,93</b> | <b>32,29</b> | <b>15,69</b> | <b>0,90</b> | <b>28,98</b> | <b>16,23</b> | 0,86        | 27,09        | 17,01   |
| Img6 | <b>0,94</b> | 27,39        | 17,09        | 0,92        | <b>30,54</b> | <b>16,16</b> | <b>0,88</b> | <b>28,24</b> | <b>16,32</b> | 0,82        | 26,02        | 17,29   |
| Img7 | <b>0,95</b> | 26,41        | 17,35        | 0,94        | <b>32,10</b> | <b>15,73</b> | <b>0,88</b> | <b>27,57</b> | <b>16,46</b> | 0,86        | 26,73        | 17,09   |
|      | 8X          |              |              |             |              |              | 16X         |              |              |             |              |         |
|      | SRGAN       |              |              | BICUBIC     |              |              | SRGAN       |              |              | BICUBIC     |              |         |
| Imge | SSIM        | PSNR         | FFT_MSE      | SSIM        | PSNR         | FFT_MSE      | SSIM        | PSNR         | FFT_MSE      | SSIM        | PSNR         | FFT_MSE |
| Img1 | <b>0,61</b> | 20,83        | <b>17,84</b> | 0,59        | <b>21,63</b> | 18,41        | 0,50        | 19,46        | <b>18,38</b> | <b>0,55</b> | <b>20,18</b> | 18,77   |
| Img2 | <b>0,69</b> | 22,92        | <b>17,34</b> | 0,67        | <b>23,32</b> | 18,01        | 0,62        | 21,75        | <b>17,79</b> | <b>0,64</b> | <b>21,98</b> | 18,36   |
| Img3 | <b>0,72</b> | 23,65        | <b>17,24</b> | 0,72        | <b>24,41</b> | 17,74        | 0,65        | 21,94        | <b>17,59</b> | <b>0,68</b> | <b>22,70</b> | 18,17   |
| Img4 | <b>0,70</b> | 20,18        | <b>18,20</b> | 0,65        | <b>20,38</b> | 18,69        | <b>0,59</b> | 18,31        | <b>18,57</b> | 0,58        | <b>18,69</b> | 19,12   |
| Img5 | <b>0,82</b> | <b>24,61</b> | <b>17,05</b> | 0,74        | 23,19        | 18,00        | <b>0,72</b> | <b>22,02</b> | <b>17,66</b> | 0,67        | 20,70        | 18,62   |
| Img6 | <b>0,80</b> | <b>24,15</b> | <b>17,16</b> | 0,72        | 23,07        | 18,05        | <b>0,71</b> | <b>22,06</b> | <b>17,65</b> | 0,67        | 21,25        | 18,50   |
| Img7 | <b>0,79</b> | <b>23,18</b> | <b>17,39</b> | 0,74        | 22,84        | 18,07        | <b>0,68</b> | <b>20,87</b> | <b>17,90</b> | 0,66        | 20,51        | 18,64   |

Eqs. (8)–(9) was used to calculate the PSNR [32,33] signal to noise ratio.

$$MSE(hr, sr) = \frac{1}{N} \sum_{i=1}^N (hr_i - sr_i)^2 \quad (8)$$

$$PSNR(hr, sr) = 10 \log_{10} \frac{L^2}{MSE} \quad (9)$$

where N is the total number of pixels, L is the maximum color density value (for example, L value is  $L = 2^8 - 1 = 255$  for 8 bit grayscale images).

For the calculation of FFT\_MSE [34], *hr* and *sr* images in the spatial domain were converted to frequency domain with the help of *fft(hr)* and *fft(sr)* commands. *fft(.)* is a function that performs Fast Fourier Transform. The average quadratic differences of the images in the frequency domain are calculated.

In order to better understand the performances of classical and current image magnification algorithms, seven different images selected from Camellia17 were selected and reduced to 2, 4, 8, 16 times. Then, magnification was carried out by magnification algorithms. The similarities of the magnification results to the original image were calculated using three different similarity metrics (SSIM, PSNR and FFT\_MSE). The results are given in Table 1. The SSIM value is close to 1 and the PSNR values are high, while the similarity is high, while the lower FFT\_MSE value means better similarity. The colored values represent the best results.

In order to better understand the performances of classical and current noise removal approaches, 10 samples were taken randomly from the test data set and different rates of noise were added. The cleaning process was then performed using four different noise removal algorithms. The results are presented in Table 2. When the results were examined, it was observed that SRGAN model performed best in different metrics in 10 images.

In the last experimental study, the ability of a SRGAN, trained in medical images, to investigate the noise clearance in a different image was investigated. For this reason, an image shown in Fig. 6(a) ( $\sigma = 0.015$  with gauss noise added) was given as input to SRGAN, which was trained with Camelyon17 images, and was subjected to noise clearing. The result is shown in Fig. 6(b). In the result image, it was observed that the performance of the SRGAN method to clean an image with different characteristics from the images in the training data set was quite low.

## 7. Discussion and conclusion

In this study, the superiority of SRGAN method, a adversarial image generator, to classical methods in the field of image enhancement and cleaning was investigated. For this, histopathological breast cancer images in the Camelyon17 data set were used. Three different similarity metrics (SSIM, PSNR and FFT\_MSE) were used to calculate the similarity of the enlarged and cleaned images obtained by the algorithms to the original images. When BICUBIC method is better than 2 times magnified images, it is seen that SRGAN model performs better when magnification is increased (4 times, 8 times 16 times). Similarly, when the noise cleaning results were examined by adding gauss noise at different rates, SRGAN model was found to be more successful than the classical image processing techniques.

As a result, the superiority of the SRGAN approach to classical techniques was observed in size magnification and noise removal processes. Finally, SRGAN has been found to have a poor performance in cleaning an image other than a training set. The reason for this was interpreted as low data and noise diversity in the education set.

**Table 2**  
Noise cleaning results (10 sample images selected from the test data set were used).

| $\sigma = 0.015$ |       |       |         |          |       |         |      |       |         |        |       |         |
|------------------|-------|-------|---------|----------|-------|---------|------|-------|---------|--------|-------|---------|
| Image            | SRGAN |       |         | Adaptive |       |         | Mean |       |         | Median |       |         |
|                  | SSIM  | PSNR  | FFT_MSE | SSIM     | PSNR  | FFT_MSE | SSIM | PSNR  | FFT_MSE | SSIM   | PSNR  | FFT_MSE |
| Img1             | 0,86  | 21,33 | 16,94   | 0,62     | 14,01 | 18,89   | 0,54 | 13,25 | 19,07   | 0,62   | 14,01 | 18,89   |
| Img2             | 0,86  | 18,44 | 17,71   | 0,62     | 13,77 | 18,95   | 0,58 | 13,09 | 19,11   | 0,62   | 13,77 | 18,95   |
| Img3             | 0,87  | 20,74 | 17,12   | 0,59     | 14,08 | 18,87   | 0,51 | 13,18 | 19,08   | 0,59   | 14,08 | 18,87   |
| Img4             | 0,87  | 21,97 | 16,79   | 0,60     | 13,16 | 19,10   | 0,56 | 12,65 | 19,22   | 0,60   | 13,16 | 19,10   |
| Img5             | 0,90  | 25,50 | 15,84   | 0,59     | 11,63 | 19,47   | 0,57 | 11,13 | 19,58   | 0,59   | 11,63 | 19,47   |
| Img6             | 0,86  | 19,86 | 17,41   | 0,62     | 14,21 | 18,85   | 0,54 | 13,41 | 19,03   | 0,62   | 14,21 | 18,85   |
| Img7             | 0,87  | 21,08 | 17,00   | 0,62     | 13,86 | 18,93   | 0,53 | 13,06 | 19,12   | 0,62   | 13,86 | 18,93   |
| Img8             | 0,89  | 28,02 | 15,03   | 0,55     | 11,81 | 19,42   | 0,54 | 11,40 | 19,52   | 0,55   | 11,81 | 19,42   |
| Img9             | 0,84  | 18,55 | 17,73   | 0,62     | 14,31 | 18,82   | 0,55 | 13,60 | 18,98   | 0,62   | 14,31 | 18,82   |
| Img10            | 0,87  | 25,26 | 15,95   | 0,58     | 12,74 | 19,20   | 0,54 | 12,17 | 19,34   | 0,58   | 12,74 | 19,20   |
| $\sigma = 0.02$  |       |       |         |          |       |         |      |       |         |        |       |         |
| Image            | SRGAN |       |         | Adaptive |       |         | Mean |       |         | Median |       |         |
|                  | SSIM  | PSNR  | FFT_MSE | SSIM     | PSNR  | FFT_MSE | SSIM | PSNR  | FFT_MSE | SSIM   | PSNR  | FFT_MSE |
| Img1             | 0,54  | 14,67 | 18,52   | 0,34     | 10,50 | 19,70   | 0,35 | 10,30 | 19,76   | 0,34   | 10,50 | 19,70   |
| Img2             | 0,59  | 13,59 | 18,88   | 0,35     | 10,27 | 19,76   | 0,38 | 10,08 | 19,82   | 0,35   | 10,27 | 19,76   |
| Img3             | 0,58  | 13,20 | 18,95   | 0,33     | 10,73 | 19,64   | 0,35 | 10,52 | 19,71   | 0,33   | 10,73 | 19,64   |
| Img4             | 0,58  | 15,33 | 18,44   | 0,34     | 10,34 | 19,74   | 0,37 | 10,19 | 19,79   | 0,34   | 10,34 | 19,74   |
| Img5             | 0,64  | 17,16 | 17,97   | 0,34     | 9,27  | 20,00   | 0,42 | 9,10  | 20,05   | 0,34   | 9,27  | 20,00   |
| Img6             | 0,58  | 14,42 | 18,63   | 0,35     | 10,83 | 19,62   | 0,37 | 10,61 | 19,69   | 0,35   | 10,83 | 19,62   |
| Img7             | 0,54  | 14,66 | 18,52   | 0,33     | 10,09 | 19,80   | 0,33 | 9,90  | 19,86   | 0,33   | 10,09 | 19,80   |
| Img8             | 0,48  | 19,64 | 17,06   | 0,31     | 7,67  | 20,38   | 0,39 | 7,57  | 20,41   | 0,31   | 7,67  | 20,38   |
| Img9             | 0,55  | 13,78 | 18,80   | 0,35     | 10,93 | 19,59   | 0,36 | 10,74 | 19,65   | 0,35   | 10,93 | 19,59   |
| Img10            | 0,51  | 18,70 | 17,32   | 0,31     | 8,16  | 20,26   | 0,39 | 8,04  | 20,30   | 0,31   | 8,16  | 20,26   |



**Fig. 6.** Noise cleaning with SRGAN. (a) An image selected from outside the training cluster (b) SRGAN cleaning result.

## References

- [1] L. Zhao, H. Bai, J. Liang, B. Zeng, A. Wang, Y. Zhao, Simultaneous color-depth super-resolution with conditional generative adversarial networks, *Pattern Recognit.* 88 (2019) 356–369.
- [2] S. Wei, X. Zhou, W. Wu, Q. Pu, Q. Wang, X. Yang, Medical image super-resolution by using multi-dictionary and random forest, *Sustain. Cities Soc.* 37 (2018) 358–370.
- [3] C.E. Duchon, Lanczos filtering in one and two dimensions, *J. Appl. Meteorol.* 18 (8) (1979) 1016–1022.
- [4] R.G. Keys, Cubic convolution interpolation for digital image processing, *IEEE Trans. Acoust. Speech Signal Process.* 29 (6) (1981) 1153–1160.
- [5] A.B. Gavader, V.S. Rajpurohit, Single frame image resolution enhancement using super resolution, *Int. J. Emerg. Technol. Comput. Sci. Electron.* 14 (2) (2015) 293–298.
- [6] P.R. Rajarapolu, V.R. Mankar, Bicubic interpolation algorithm implementation for image appearance enhancement, *Int. J. Comput. Sci. Technol.* 8 (2) (2017) 23–26.
- [7] M.A. Mazurowski, M. Buda, A. Saha, M.R. Bashir, Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI, *J. Magn. Reson. Imaging* 153 (2018) 1–27.
- [8] C. Dong, Y. Deng, C.C. Loy, X. Tang, Compression artifacts reduction by a deep convolutional network, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 576–584.
- [9] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (2) (2016) 295–307.
- [10] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, in: Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, 2017, pp. 105–114.
- [11] D. Mahapatra, B. Bozorgtabar, R. Garnavi, Image super-resolution using progressive generative adversarial networks for medical image analysis, *Comput. Med. Imaging Graph.* 71 (2019) 30–39.

- [12] D. Mahapatra, B. Bozorgtabar, S. Hewavitharanage, R. Garnavi, Image super resolution using generative adversarial networks and local saliency maps for retinal image analysis, in: *Medical Image Computing and Computer Assisted Intervention - MICCAI 2017*, 2017, pp. 382–390.
- [13] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *Generative adversarial networks*, *Vet. Immunol. Immunopathol.* 155 (4) (2014) 270–275.
- [14] C. Li, M. Wand, *Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks*, in: *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) LNCS*, vol. 9907, 2016, pp. 702–716.
- [15] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, H. Li, High-resolution image inpainting using multi-scale neural patch synthesis, in: *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, 2017, pp. 4076–4084.
- [16] G. Antipov, M. Baccouche, J.L. Dugelay, Face aging with conditional generative adversarial networks, in: *Proc. - Int. Conf. Image Process. ICIP*, 2018, pp. 2089–2093.
- [17] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D. Metaxas, StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks, in: *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5908–5916.
- [18] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [19] J.Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.
- [20] S. Takahashi, Y. Chen, K. Tanaka-Ishii, *Modeling financial time-series with generative adversarial networks*, *Physica A* 527 (2019) 121261.
- [21] T. Silva, A beginner's guide to generative adversarial networks (GANs). <https://skymind.ai/wiki/generative-adversarial-network-gan>. (Accessed 28 December 2018).
- [22] J. Langr, V. Bok, *GANs in Action*, Manning Publications, MEAP. Shelter Island, NY / USA, 2018.
- [23] I. Aydın, Görüntü restorasyonu, [http://web.firat.edu.tr/iaydin/bmu357/bmu\\_357\\_bolum5.pdf](http://web.firat.edu.tr/iaydin/bmu357/bmu_357_bolum5.pdf). (Accessed 15 January 2019).
- [24] H. Gao, M. Hu, T. Gao, R. Cheng, Robust detection of median filtering based on combined features of difference image, *Signal Process., Image Commun.* 72 (2019) 126–133.
- [25] N.R. Kumar, J.U. Kumar, A spatial mean and median filter for noise removal in digital images, *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.* 4 (1) (2015) 246–253.
- [26] G. Litjens, P. Bandi, B.E. Bejnordi, O. Geessink, M. Balkenhol, P. Bult, A. Halilovic, M. Hermesen, R. van der Loo, R. Vogels, Q.F. Manson, N. Stathonikos, A. Baidoshvili, P. v. Diest, C. Wauters, M. v. Dijk, J. van der Laak, 1399 h & e-stained sentinel lymph node sections of breast cancer patients: the CAMELYON dataset, *Gigascience* 7 (6) (2018).
- [27] Camelyon17: Breast cancer metastases in lymph nodes and the TNM classification system. <https://camelyon17.grand-challenge.org/Background/>. (Accessed 22 February 2019).
- [28] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: *International Conference on Learning Representations*, 2014.
- [29] S. Gross, M. Wilber, Training and investigating residual nets, <http://torch.ch/blog/2016/02/04/resnets.html>. (Accessed 28 February 2019).
- [30] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015, *arXiv:1502.03167*.
- [31] A.O. Daoud, A.A. Tsehayae, A.R. Fayek, A guided evaluation of the impact of research and development partnerships on university, industry, and government, *Can. J. Civil Eng.* 44 (4) (2017) 253–263.
- [32] Z. Wang, A.C. Bovik, Mean squared error: Lot it or leave it? A new look at signal fidelity measures, *IEEE Signal Process. Mag.* 26 (1) (2009) 98–117.
- [33] A.B. Ishak, Choosing parameters for rényi and tsallis entropies within a two-dimensional multilevel image segmentation framework, *Physica A* 466 (2017) 521–536.
- [34] O. Sarbishei, K. Radecka, Analysis of mean-square-error (MSE) for fixed-point FFT units, in: *Proc. - IEEE Int. Symp. Circuits Syst.*, 2011, pp. 1732–1735.