Received November 10, 2020, accepted December 6, 2020, date of publication February 5, 2021, date of current version March 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3057497

Breast Cancer Histopathology Image Super-Resolution Using Wide-Attention GAN With Improved Wasserstein Gradient Penalty and Perceptual Loss

FAEZEHSADAT SHAHIDI
Department of Informatics (FTIR), Universiti Teknologi Malaysia, Kuala Lumpur 54100, Malaysia
e-mail: faezeh.shahidi.sh@gmail.com

ABSTRACT In the realm of image processing, enhancing the quality of the images is known as a super-resolution problem (SR). Among SR methods, a super-resolution generative adversarial network, or SRGAN, has been introduced to generate SR images from low-resolution images. As it is of the utmost importance to keep the size and the shape of the images, while enlarging the medical images, we propose a novel super-resolution model with a generative adversarial network to generate SR images with finer details and higher quality to encourage less blurring. By widening residual blocks and using a self-attention layer, our model becomes robust and generalizable as it is able to extract the most important part of the images before up-sampling. We named our proposed model as wide-attention SRGAN (WA-SRGAN). Moreover, we have applied improved Wasserstein with a Gradient penalty to stabilize the model while training. To train our model, we have applied images from Camylon 16 database and enlarged them by 2×, 4×, 8×, and 16× upscale factors with the ground truth of the size of $256 \times 256 \times 3$. Furthermore, two normalization methods, including batch normalization, and weight normalization have been applied and we observed that weight normalization is an enabling factor to improve metric performance in terms of SSIM. Moreover, several evaluation metrics, such as PSNR, MSE, SSIM, MS-SSIM, and QILV have been applied for having a comprehensive objective comparison with other methods, including SRGAN, A-SRGAN, and bicubial. Also, we performed the job of classification by using a deep learning model called ResNeXt-101 (32×8d) for super-resolution, high-resolution, and low-resolution images and compared the outcomes in terms of accuracy score. Finally, the results on breast cancer histopathology images show the superiority of our model by using weight normalization and a batch size of one in terms of restoration of the color and the texture details.

INDEX TERMS SRGAN, Wasserstein gradient penalty, weight and batch normalization, perceptual loss, breast cancer histopathology medical images, classification.

I. INTRODUCTION

Due to the cost of hardware and storage to acquire the high resolution (HR) images from low resolution (LR) medical images are advisable. To tackle the cost of hardware, image super-resolution methods are drawing attention to the reconstruction of poor-quality images with missing pixels on them. Since it is vital to enlarge the pictures in a way that the texture details and feature information are indistinct and clear by using the quantitative performance of the applied SR methods, generative adversarial networks are gaining attention from researchers as they are able to reconstruct the images in a realistic manner [1].

Therefore, in this paper, we propose an SR solution method based on generative adversarial networks (SRGAN) with the aim of reconstructing histopathology breast cancer images. We used a wide residual block instead of residual blocks. Moreover, in both generator and discriminator models, we utilized a self-attention layer to catch the most important features.

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Callico.

Also, as training the GAN models is difficult and they are prone to divergence, we have applied Wasserstein extension with the gradient penalty that is added to the adversarial loss criterion to improve the stability of the model during the training phase.

Furthermore, perceptual loss in this study is calculated from a combination of the image loss, adversarial loss, perception loss, and total variation (TV) loss to generate SR images that match human perceptions of structure, luminance, and contrast. As pre-trained models focus on natural images, we have retrained a VGG-19 model on the medical images prior to using this model to calculate the perception loss. Similarly, the top and bottom gradients for both generator and discriminator have been recorded to monitor the performance of the model. In this work, we have been monitoring several quantitative metrics, including Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), Structural Similarity Index (SSIM), Multiscale Structural Similarity Index (MS-SSIM), and Quality Index based on Local Variance (QILV). During the evaluation, we focused on the SSIM metric as the higher score we have gained from this metric, the better result we have obtained in terms of preserving context and color information.

Consequently, we have prioritized an epoch with the highest SSIM results since medical images will be used for visual observation by doctors, and this metric is more in line with human subjective visual perception. Finally, we have done experiments on SRGAN [1] and A-SRGAN [2] models and trained them on the same database. These models have been examined with the same methods, including improved Wasserstein gradient penalty and perceptual loss as our proposed model to have an unbiased comparison. Also, two normalization methods, including batch normalization and weight normalization were applied and it has been observed that weight normalization outperforms batch normalization in terms of SSIM performance. After training our proposed model by using the weight normalization technique, the weights of our model were stored and then reused in the pre-processing phase of the classification to enlarge the LR images and obtain their corresponding SR ones. Afterward, the job of classification was performed for SR, HR, and LR images to obtain an accuracy score for each and then compare them. According to a comparison study conducted by Shahidi *et al.* [3], a deep learning model named ResNeXt-101 (32 × 8d) [4] gained a satisfactory result for the classification by using BreaKHis database [5]. This model is made up of in-built wide residual networks along with identity connections. Also, the width of the ResNeXt network is known as cardinality and it can improve accuracy score and ability to withstand the complexity. So, in this study, ResNeXt-101 (32 × 8d) was utilized to classify the images.

The comparison of the results shows the positive impact of our proposed method, WA-SRGAN, in the preprocessing phase as accuracy for the classification through the SR images is almost the same as the results obtained for HR images.

Overall, the contributions of this paper are mainly in eight aspects.

1) We proposed a novel generator model equipped with the combination of wide residual blocks and self-attention layers before up-sampling. This model can improve image quality by learning more prior information of sample data and reducing the difference between reconstructed image blocks.
2) We applied a combination of image loss, adversarial loss, perception loss, and TV loss as the perceptual loss to have a wide-ranging effect while generating SR images.
3) We implemented a discriminator model whose architecture has one self-attention layer to extract more features from image patches and end up fairly correcting the generator.
4) We used the Wasserstein extension with a gradient penalty in addition to independent pixel-wise losses in the adversarial loss function.
5) We retrained a pre-trained VGG-19 model to extract the most valuable texture details to compute the perception loss.
6) We have applied two methods of in-built normalizations, including batch normalization and weight normalization in our proposed architecture to compare these two methods using objective performance in terms of SSIM.
7) We applied a comprehensive combination of the performance metrics, including PSNR, MSE, SSIM, MSSSIM, QILV to monitor and assess the quality of the images while training.
8) We implemented our proposed model, WA-SRGAN, in the pre-processing phase before classification to enlarge the LR images and gain their corresponding SR images. The same process has been performed for A_SRGAN, and SRGAN models as well. A ResNeXt-101 (32 × 8d) model was utilized to do the job of classification while using LR, SR, and HR images. Then, a comparison study was made after gaining the results for LR, SR, and HR separately in terms of accuracy score and the loss value.

II. RELATED WORKS

A. SUPER-RESOLUTION (SR)

Super-resolution (SR) is a challenging task by which we are supposed to estimate a high-resolution image from its low-resolution image to fill the absent details in the images [1]. SR methods can be applied in various applications, such as medical images. As high-resolution medical images like histopathology images can be gained with the cost of expensive microscopes and maintaining these images can be achieved with the cost of storage, LR images can be acquired much faster and more cheaply. After acquiring LR images, they can be enlarged with the aid of SR methods. The early classical SR prediction-based methods, like the study conducted by Irani and Peleg [6] used the back-projection method applied in tomography to compute and fill the unknown pixels by using image sequence. Also, Duchon [7] applied the Lanczos filtering method, and Keys [8] introduced Cubic convolution interpolation for increasing the size of the images. Among all edge-based methods, Freedman and Fattal [9] up-scaled images by local self-similarity method, and Sun *et al.* [10] applied Gradient Profile prior to tackle the SR issue.

Also, the current CNN-based SR algorithms by Dong *et al.*, [11], and Wang *et al.*, [12] received attention due to their excellent performance. Moreover, Dong *et al.* [13] proposed a model in which the author applied bicubic interpolation to enlarge the small-sized picture, two convolutional layers to extract the features, and non-linear mapping to map the LR to the HR. In the end, the author calculated the loss function between the small picture and its HR one.

Furthermore, Dong *et al.* [13] introduced another model entitled fast super-resolution by CNN(FRCNN) with some improvements, including replacing bicubic interpolation by the decoder for up-sampling, increasing the number of convolutional layers, and utilizing smaller filter size. Moreover, by employing ReLU activation, residual units, and a deeper network, Wang *et al.*, [14] could improve the performance even more. Thus, for SR problems, deeper layers were helpful to improve learning capability and performance.

B. GENERATIVE ADVERSARIAL NETWORK (GAN)

Based on the taxonomy of the generative models presented by Goodfellow [15], models that are learned based on the Maximum Likelihood are divided into two groups, including explicit density and implicit density. Moreover, explicit models generate a value pixel based on the probability of the previous pixels. These models can be based on traceable density like PixelCNN [16] or approximate density, like variational autoencoder [17]. On the contrary, the generative adversarial network (GAN) falls into the implicit density category. This model was introduced by Goodfellow *et al.*, [18], which is a novel way of generating data [19]. GAN models provide us with generating acceptable-looking images [1]. This network is comprised of two parts, including a generative network that is in charge of generating images and the discriminative part that estimates the probability of a realistic created image [18]. In this model, the generator creates a picture whose distribution is in alignment with the real images (or the generated image is converged to the sample data), so that the discriminator part cannot distinguish that the generated image is not real and the model is converged.

C. SUPER-RESOLUTION GENERATIVE ADVERSARIAL NETWORK (SRGAN)

During the process of changing the small-sized images to the large ones, the whole pixels are needed to be filled, so these pixels can be generated by GAN models. Thus, the super-resolution GAN (SRGAN) model has been proposed by Ledig *et al.*, [1] to generate the missing points. As perceptually based algorithms can distinguish the realistic generated images [20] and perceptual loss function shows more appraisable results than per-pixel loss [21], a crucial perceptual similarity has been defined by Ledig *et al.*, [1] that tailors SR problems. This model leverages the state-of-the-art VGG-19 model [22] to acquire perceptual (or content) loss. Moreover, the definition of the adversarial loss in this model is based on the probabilities of the discriminator's overall training samples. Also, with the aid of residual blocks [23] in the generator, this model can extract the most important features before passing them to the up-sampling layers. Furthermore, by using residual in residual dense blocks [24], these blocks have been further enhanced by Wang *et al.*, [25] and gained better results.

D. SRGAN FOR MEDICAL HISTOPATHOLOGY BREAST CANCER IMAGING

Due to the poor-quality of histopathology images received from scanners and microscopes, super-resolution problems received attention to deal with this issue. For instance, Mukherjee *et al.*, [26] embraced the convolutional neural network (CNN) to convert low-resolution slide scanner images of cancer data into a high-resolution image. Then, the SRGAN model was applied by Çelik and Talu [27] to increase the resolution of breast cancer histopathology images. Moreover, this model was enhanced by the researchers. Thus, by using a feature similarity index for image quality assessment (FISM) instead of original mean square error (MSE), Huang *et al.*, [28] improved the SRGAN model in terms of perceptual loss and increased the histopathology image resolution. Also, SRGAN-SQE was proposed by Upadhyay and Awate [29] by adding an autoencoder in the pre-processing phase to intensify the resolution of the breast cancer histopathology images and employing a heavy-tailed non-Gaussian distribution probability density loss function on the residuals. Various research conducted on the SRGAN model shows this model is robust in terms of improving the quality of the images and can be applied for histopathology images to improve the resolution.

E. SELF-ATTENTION SRGAN

By introducing an encoder and a decoder by using recurrent neural networks, long short-term memory (LSTM), and gated recurrent neural networks, in particular, Vaswani *et al.*, [30] has introduced a state of the art method based on attention function that is able to map a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. Then, Zhang *et al.*, [31] introduced a self-attention generative adversarial network with an in-built attention function. This function is based on the non-local mean [32]. Unlike convolutional layers that extract local neighborhoods, the attention function is a filtering algorithm that calculates the weighted mean of all pixels in an image. By this algorithm, the model can learn relationships between widely separated spatial regions and distant pixels based on patch appearance similarity.

F. METRIC PERFORMANCE

There are two categories of the image quality evaluation method, including subjective and objective [33], [34] by which researchers are able to assess the quality of the images after enlarging the images. The subjective method is a mean opinion score (MOS) that is performed by a human being [1]. Image enhancement or improving the visual quality of a digital image based on the perceptual assessment of a human viewer can be subjective. For this reason, it is necessary to establish quantitative/empirical measures to compare the effects of image enhancement algorithms on image quality [35]. So, the following quantitative measures are listed to establish an objective assessment for the results.

1) MEAN SQUARE ERROR (MSE)

This is the modest and most commonly used quality metric and is computed by the average of the squared intensity differences of the original image and improved image [34], [35]. The equation of the MSE formula is defined as follows:

$$MSE\left(f,g\right)=\frac{1}{M\times N}\sum\nolimits_{i=1}^{M}\sum\nolimits_{i=1}^{N}\left(f_{ij}-g_{ij}\right)^{2} \qquad (1)$$

where, f (i, j) is the original image, g (i, j) is the enhanced image and M×N is the size of the image concerning rows and columns.

2) PEAK SIGNAL-TO-NOISE RATIO (PSNR)

Is the ratio between the maximum possible value and the power of distorting noise value that affects the quality of the image that is expressed in terms of a logarithmic decibel scale [34].

$$PSNR\left(f,g\right)=10\times\log_{10}\left(\frac{MAX^{2}}{MSE\left(f,g\right)}\right) \qquad (2)$$

3) STRUCTURAL SIMILARITY INDEX MEASURE (SSIM)

This is developed by Wang *et al.*, [36] and is considered to be allied with the quality perception of the human visual system (HVS). This metric is designed by a combination of three factors that are loss of correlation, luminance distortion, and contrast distortion. The SSIM is calculated as:

$$SSIM\left(f,g\right)=\left[l\left(f,g\right)\right]^{\alpha}.\left[c\left(f,g\right)\right]^{\beta}.\left[s\left(f,g\right)\right]^{\gamma} \qquad (3)$$

where,

$$\begin{aligned} l\left(f,g\right)&=\frac{2\mu_{f}\mu_{g}+C_{1}}{\mu_{f}^{2}+\mu_{g}^{2}+C_{1}},\\ c\left(f,g\right)&=\frac{2\sigma_{f}\sigma_{g}+C_{2}}{\sigma_{f}^{2}+\sigma_{g}^{2}+C_{2}},\\ s\left(f,g\right)&=\frac{\sigma_{fg}+C_{3}}{\sigma_{f}\sigma_{g}+C_{3}} \end{aligned} \qquad (4)$$

The closeness of the mean is calculated in the first term in (4), which shows the luminance comparison between the two images. In case this factor is equal to one, it illustrates that the means of the two images are equal. In the second term, the contrast comparison will be measured by using the standard deviation of two images. As the first term, the maximal number for contrast comparison function is one and that shows the two images are close in terms of standard deviation. Finally, by calculating the correlation coefficient or covariance between two images in the third term, we are able to compare the two images in terms of structure. The value of 0 in this term shows no correlation and the value of one means two images are equal in terms of structure. Moreover, in this equation, the positive values of C1, C2, and C3 are applied to dodge the zero denominators.

4) MULTI-SCALE SSIM (MS-SSIM)

Is proposed by Wang *et al.* [37]. This metric is evolved from the SSIM index that derives from the original image at different scales as [1, M-1]. As shown in (5), the luminance comparison function is calculated only for scale M and the rest, including contrast comparison and structural comparison functions, are computed in different scales. The overall MS-SSIM assessment is found by joining the measurement at different scales using:

$$\begin{aligned} &MS-SSIM\left(f,g\right)\\ &\quad=\left[l_{M}\left(f,g\right)\right]^{\alpha M}.\prod\nolimits_{j=1}^{M}\left[c_{j}\left(f,g\right)\right]^{\beta j}.\left[s_{j}\left(f,g\right)\right]^{\gamma j} \end{aligned} \qquad (5)$$

5) QUALITY INDEX BASED ON LOCAL VARIANCE (QILV)

This is another metric performance that has been introduced by Aja-Fernandez *et al.,* [38]. In order to match properly with a subjective judgment based on the visual information, we need QILV as SSIM has minimally taken some sources of degradation, like blurriness. Moreover, SSIM weighs noise over blur and blurred images affect further structural processing and interpretation of the image for the human eye. Thus, one should rely on different structural information to reduce this bias and alternative quality measures should be conceived.

$$QILV\left(f,g\right)=\frac{2\mu v_{f}\mu v_{g}}{\mu v_{f}^{2}+\mu v_{g}^{2}}\cdot\frac{2\sigma v_{f}\sigma v_{g}}{\sigma v_{f}^{2}+\sigma v_{g}^{2}}\cdot\frac{\sigma v_{f}v_{g}}{\sigma v_{f}\sigma v_{g}} \qquad (6)$$

Unlike SSIM, QILV is more based on the distribution of the local variance in the images than global quality. This way, this metric can better compare the non-stationarity of the images. There are three factors, including the mean of the local variance distributions, the standard deviation of the local variances, and spatial coherence in this equation by which the comparison between two images is performed.

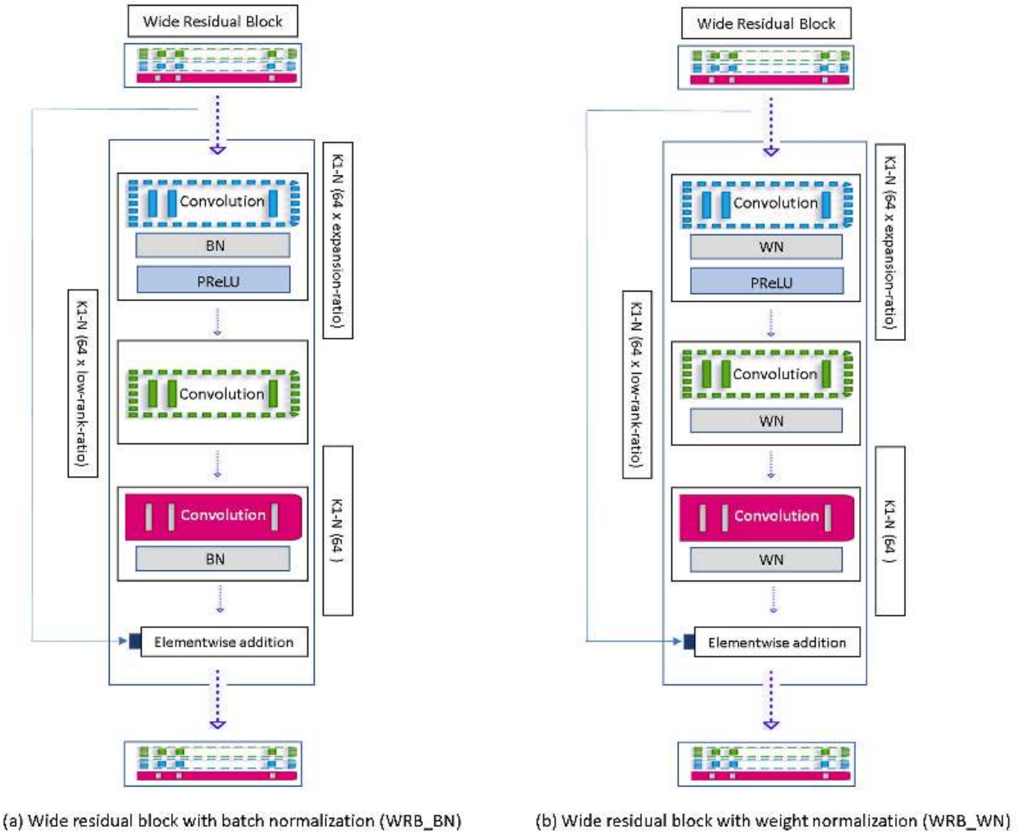
III. METHODS

In this part, we introduce our proposed networks Wide-Attention SRGAN (WA-SRGAN) for enlarging the breast cancer histopathology images. In this part, the architecture, learning algorithms, and loss functions will be explained.

A. ARCHITECTURE

1) BATCH NORMALIZATION

Batch Normalization (BN) is a technique that is applied by the researchers to curve the speed limitation barriers while



(a) Wide residual block with batch normalization (WRB_BN)

(b) Wide residual block with weight normalization (WRB_WN)

FIGURE 1. The architecture of our wide residual block (WRB) consists of three convolutional layers. (a) WRB_BN shows the wide residual block with batch normalization and PReLU. (b) WRB_WN identifies the wide residual block using weight normalization and PReLU.

training deep neural models and it is observed that this technique is effective in various applications. This method is based on two fundamental concepts, i.e. normalization and distributions [39], [40]. BN takes the values of $\mathrm{x_i}$ as the input for the mini-batch B of size m and computes the following:

$$\mu\sigma y_i \leftarrow \frac{x_i - {}_{B}}{\sqrt{{}^{2}_{B} + \epsilon}}.\gamma + \beta \equiv BN_{\gamma,\beta}(x_i) \tag{7}$$

where μ_{B} is the mean over the specific mini-batch, σ is the mini-batch variance, and $\mathrm{y_i}$ is the output corresponding to each input in the batch. Gamma (γ) and beta (β) are known as scale and shift parameters respectively and they have to be trained.

2) WEIGHT NORMALIZATION

Weight Normalization (WN) has recently been proposed by Salimans and Kingma [41] which is one of the most cutting-edge techniques. For a linear layer, this technique computes the following:

$$y = W^T x + b \tag{8}$$

WN is a reparameterization method which decouples or recomputes the weight tensor shown by W by its norms that are magnitude vector known as g, and direction vector identified as v before every forward call as follows:

$$w_i = \frac{g_i}{||v_i||_2} \cdot v_i \tag{9}$$

where w_i and v_i are the i-th column of W and V, respectively.

Although deep learning models show better results for training results by means of weight normalizations, this method decreases the test accuracy by almost 0.6 percent for the test dataset with the aim of classifications [42]. However, in the realm of generative adversarial networks, the batch normalization affects the quality of the generated images negatively [43]. Thus, for generating better images through GAN models, weight normalization is proposed.

3) WIDE RESIDUAL BLOCK (WRB)

Fig. 1 illustrates the architecture of a wide residual block (WRB) in our study. Using this block in our generative model has been motivated by a study conducted by Yu *et al.*, [44]. In this study, the author shows wider channels before activation in residual blocks will improve the performance of image super-resolution networks. Moreover, instead of using a convolutional layer with a size of three for the kernel before the activation function, the author applied a one-by-one conv layer that was introduced by Lin *et al.* [45]. Using a one-by-one conv layer helps the network to modify the number of channels with no computational complexes.

Thus, in this work, the author could use the expansion ratio of 2× to 4× and even wider 6× to 9× with ease. By gaining this idea, we have improved the generator by using wide residual blocks with three convolution layers. The first layer with the kernel size of 1 × 1 enables the layer to increase the dimensions. The expansion ratio that we have applied was 4×. Unlike Yu *et al.*, [44] who applied ReLU activation after increasing the number of the channels, we have applied Parametric ReLU (PReLU) activation function [46] in our WRB blocks as it is proved to work better than the ReLU activation function for deeper and wider networks due to the fact that the slopes factor for negative value is a learnable parameter. After decreasing the number of the dimensions in the second layer by the low-rank ratio of 0.8, this number decreased to 51, and finally, at the third layer, the number of dimensions returned to 64. This way, the result of the Generator has been improved in terms of not only the performance metrics but also the visual perception.

Moreover, weight normalization is suggested by Yu *et al.*, [44] to improve the results. We have applied both batch and weight normalization in WRB blocks that are shown in Fig 1. Fig 1. (a) is WRB_BN that is the wide residual block with batch normalizations. In this block, three conv layers have been applied and two batch normalizations were used after the first and the third block. Also, Fig 1. (b) is WRB_WN which is the wide residual block with weight normalization. In this block, three weight normalizations have been applied after each conv layer. This block is the same as the one that has been proposed by Yu *et al.*, [44] yet with the activation function difference.

Since we applied both generator and discriminator with both batch and weight normalizations, two WRB blocks have been proposed. Thus, WRB_BN was applied by the WA-SRGAN model with an in-built batch normalization method and WRB_WN was used with the same model by an in-built weight normalization method and they have been compared and explained in terms of objective results.

4) SELF-ATTENTION LAYER

The self-attention layer has been applied in both generator and discriminator networks of the SRGAN model by Pathak *et al.*, [2] for large-scale images. In this study, the author employed two pooling layers with kernel size and stride of two before and after the self-attention layer to build a flexible attention layer that is suitable for big scaled images, and then by using a bicubic function the output of this layer was returned to the size of the input image. However, in our study, the pooling layers have not been applied as the sizes of the images are not huge.

Thus, the plain attention function layer applied in our proposed model is based on three weighted functions, quarry, keys, and values called, f, g, and h respectively. Each of these functions maps the input image batch X with the size of B (number of batches), C (number of channels), W (width), and H (height) to their corresponding feature spaces with the aid of a one-by-one convolutional layer. The size of each feature size is changed to B, C/k (where k = 8 (i.e., $\bar{C} = C/8$) due to memory efficiency) and N (is gained by the multiplication of the W and H of the input images). After acquiring these functions, f(X) will be transposed and multiplied by g(X) and then passed through a sigmoid function to calculate the attention, where,

$$\beta_{j,i} = \frac{exp(s_{ij})}{\sum_{i=1}^{N} exp(s_{ij})} \quad Where,\ s_{ij} = f(x_i)^T g(x_j) \tag{10}$$

Hence, $\beta_{j,i}$ indicates the extent to which the model attends to the ith location when synthesizing the jth region. Then, the output of the attention function is calculated by a product between the value function h(X), and the attention. The output of the attention layer is calculated as follows:

$$o_j = v\left(\sum_{i=1}^{N} \beta_{j,i} h(x_i)\right), \quad h(x_i) = W_h x_i,\ v(x_i) = W_v x_i \tag{11}$$

In the above formulation, W's are the learned weight matrices, which are implemented as 1 × 1 convolution. The final output is given by:

$$y_i = \gamma o_i + x_i \tag{12}$$

Finally, the output of the attention layer that is o_i multiplied by γ that is a learnable scale parameter and then added by the input feature. The learnable scale parameter γ is initialized as 0 to rely on the local neighborhood and then gradually learned to assign more weight to the non-local evidence.

5) GENERATOR STRUCTURE

Fig. 2 shows the architecture of the generator of our proposed WA-SRGAN. In this model, we have applied eight wide residual blocks (WRB). WRB_BN blocks have been applied by the generator with an in-built batch normalization method and on the contrary, WRB_WN blocks have been applied in the generator with an in-built weight normalization technique.

Similarly, the numbers of kernels, channels, strides, and padding are illustrated as K, N, S, and P respectively for each layer. After passing the input batch images by the first layer of the generator, the number of the channels increases to 64.

Then, the output of this layer is added elementwise to the output of WRB and another convolutional layer. Afterward, a self-attention layer is applied to gain attention function. So, the most important features are gained through three functions, including key, query, and value and the attention output will be passed to the final convolution layer. Eventually, the output of the generator network is scaled to [−1,1] by a Tanh activation function, and by adding them to one and then divide them by 2 the output range changes to [0,1]. Once the pixels of the generated images are in the range of [0,1], it will make it comfortable for the discriminator network to distinguish between the generated and real images. Also, there is no need

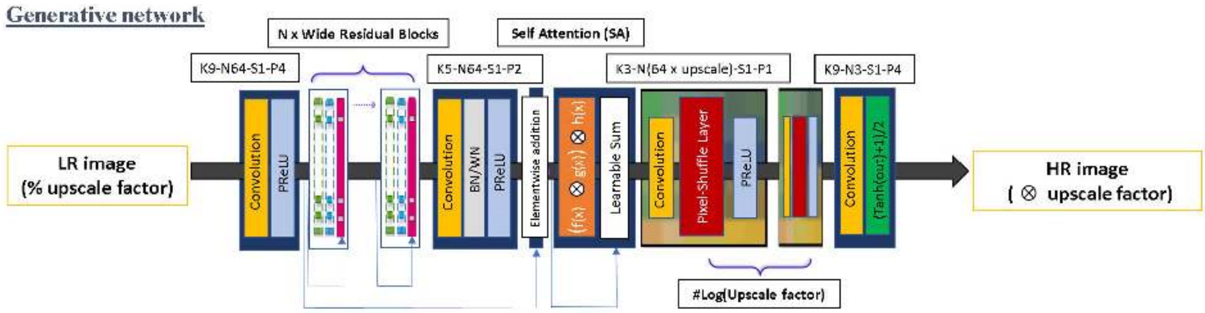


FIGURE 2. An overview of the generator of our proposed Wide-Attention SRGAN (WA-SRGAN) that consists of wide residual blocks and self-attention layer before up-sampling. In this network, both batch normalization, and weight normalization have been applied separately.

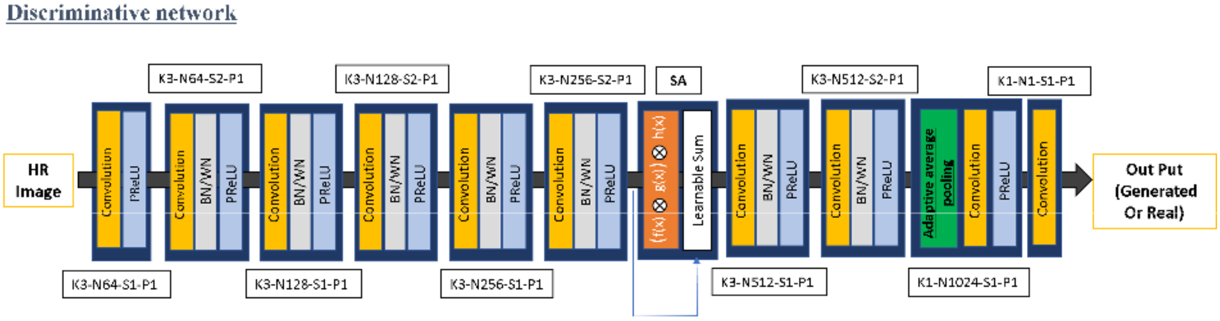


FIGURE 3. The discriminator of the WA-SRGAN model with an in-built self-attention layer.

to rescale the generated samples before feeding them to the discriminator or loss function.

6) DISCRIMINATOR

Fig. 3 shows the architecture of the discriminator network which is implemented in our proposed WA-SRGAN model. The discriminator of our proposed model has a VGG-like structure that gradually decreases the size of the feature maps and expands the depth of channels since each layer contains a similar amount of information. Unlike the vanilla discriminator in the SRGAN model, we have implemented a plain self-attention layer in the middle of the layer that can gain the most important feature by the attention function.

The final output of the discriminator network is a single value that indicates whether the input image is generated or real.

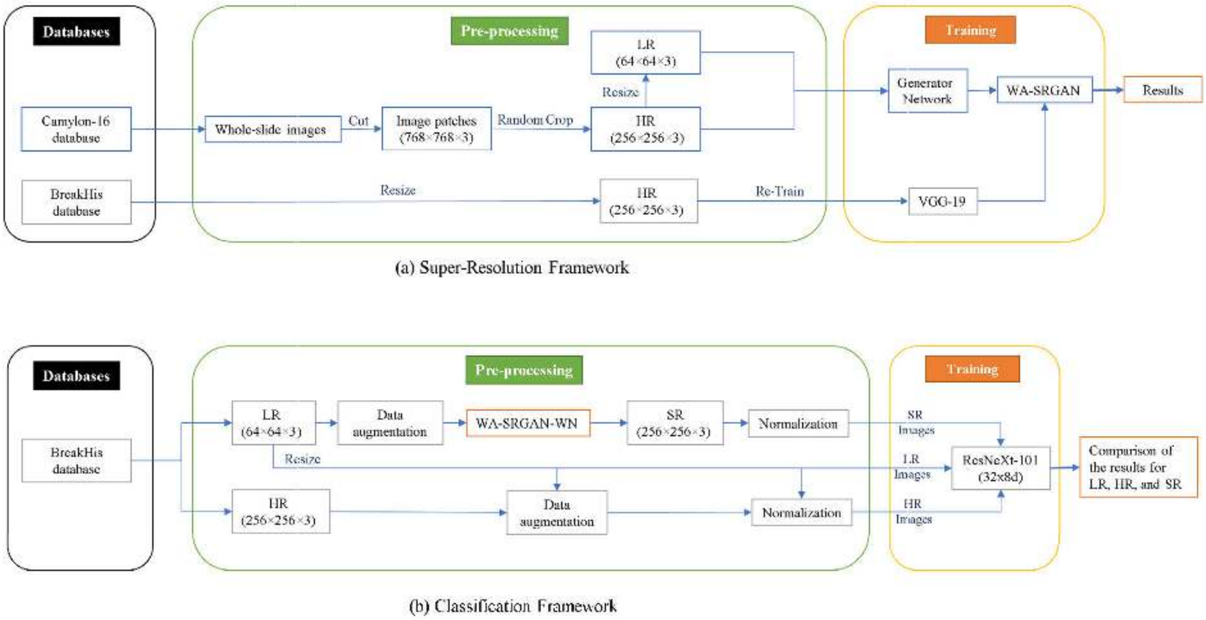
B. IMPROVED WASSERSTEIN GRADIENT DESCENT

During training the GAN models, the values of the gradients clarify how to update the parameters. Large gradients will lead the parameters to undergo bigger changes. In case the loss surface is deep around the searching location, large gradient values will lead the model to surpass the region without converging. Thus, the model has to search for a new region at the next iteration for the optimal solutions. There are several techniques to stabilize the training and solve the big value of the gradient. The most recommended method is gradient clipping [47], and setting limitations that is one optimal solution for this issue. Consequently, setting minimum and maximum for the gradient values will help the model to make small changes while updating the parameters without jeopardizing the search results in the current region area while spending ages for training.

Moreover, weight clipping was proposed by Arjovsky *et al.* [48] to build the Wasserstein GAN model is another way to stabilize the GAN training. However, according to Hany and Walters [49], weight clipping is an indirect way to perform the gradient clipping by which it is difficult to train deep models since most of the gradients or weights stick to the minimum and maximum values [-c, c] and only a few of them gain the values between these two parameters. Also, this method causes vanishing or explosion gradients. To solve this issue, Gulrajani *et al.*, [50] improved the Wasserstein GAN by adding a gradient penalty to the discriminator loss function. The gradient penalty is calculated based on the random interpolation between a pair of real and generated data that can be defined as follows:

$$\mathbb{E}_{fake}\left[D\left(x\right)\right]-\mathbb{E}_{real}\left[D\left(x\right)\right]+\lambda.\mathbb{E}_{\hat{x}}[||\frac{\partial D(x)}{\partial \hat{x}}||_2-1]^2,$$
$$\hat{x}=\alpha.x_{real}+\left(1-\alpha\right).x_{fake},\alpha\sim U(0,1) \qquad (13)$$

In this method, the author applied a way to enforce the Lipschitz by constraining the gradient norm of the critics' output with respect to its input random samples $\hat{x}$. The gradient penalty contains the norm two of the second derivatives of the network with activation functions. The point x, used to



(a) Super-Resolution Framework

(b) Classification Framework

FIGURE 4. The general framework of our experiment is divided into two parts. Both parts consist of three fragments, including databases, pre-processing, and training. (a) This is the first part of our framework that shows the flow of the learning phase of our proposed WA-SRGAN model. Likewise, part (b) illustrates the framework for performing the job of classification. In this part, a ResNeXt-101 model was used to do the job of classification for LR, HR, and SR and compare the results respectively. for obtaining the SR images, the WA-SRGAN with the weight normalization was used. Moreover, in the pre-processing stage, data augmentation and normalization techniques were applied.

calculate the gradient norm, is any point sampled between the real data distribution P_r and the generated (fake) data distribution P_g.

The implementation of the improved Wasserstein function in the discriminator architecture is performed based on the following techniques:

1) The sigmoid function has been removed from the last layer of the discriminator.
2) The logarithmic function does not apply in this for the results while computing the loss function.
3) Using the gradient penalty and adding it to the loss criterion.
4) Using Adam as an optimization parameter.
5) The value of the λ is set to 10.

C. PERCEPTUAL LOSS

In this work, we have applied a combination of several loss functions to calculate the perceptual loss function for the generator criterion as follows:

1) Adversarial loss, which is based on the output probability of the improved discriminator.

$$l_{adv}^{SR} = -\mathbb{E}_{fake}\left[D\left(x\right)\right] \tag{14}$$

2) Image loss or pixel-wise content loss l_{pixel}^{SR}, which is the MSE loss between the SR and the HR images.
3) Perception loss or VGG loss l_{vgg}^{SR}, that is the MSE loss between the last feature maps of a retrained VGG network by breast cancer histopathology image database called BreaKHis from SR and HR images.
4) TV loss or regularization loss l_{tv}^{SR}, that is the sum of average L2-norm or the pixel gradients in horizontal and vertical directions. Since the TV loss makes the images blurry, we added a strong restraint $(2e-8)$ to the pixel gradients.

The final perceptual loss that is defined as follows is able to take both pixel-wise and high-level similarities into account once discriminating between SR and HR images.

$$l_{pixel}^{SR} + 1e-3.l_{adv}^{SR} + 6e-3.l_{vgg}^{SR} + 2e-8.l_{tv}^{SR} \tag{15}$$

Despite Upadhyay and Awate [29], who have applied a negative sum of structural similarity (sSSIM) in their work to calculate the perceptual loss for the generator, Wang *et al.*, [36] stated that using SSIM index in the design of some algorithms is not an easy task as it is mathematically more cumbersome than MSE. Thus, in our proposed work we used MSE for the pixel-wise and perception loss functions.

IV. EXPERIMENTS

A. GENERAL FRAMEWORK

The overall general framework of our experiment is shown in Fig.4 that is divided into two parts, including the super-resolution framework, and the classification framework. Each part contains three parts, including databases, pre-processing, and training. In the following, we will explain each part based on our framework. In our experiments, Pytorch, an open-source machine learning library based on the Torch library [51], had been utilized.

B. SUPER-RESOLUTION EXPERIMENTS

1) DATABASE

In this experiment, we have applied two databases. The first database was Breast Cancer Histopathological Image Classification (BreakHis) which is a pathology dataset that consists of 7,909 breast cancer histopathology images from 82 patients with different magnification factors, including 40×, 100×, 200×, and 400× [5]. The 7,909 images include 2,480 benign and 5,429 malignant sample images with all the subtypes [52].

The second database was Cancer Metastases in Lymph Nodes (Camelyon) which was established based on a research challenge dataset competition in 2016. This database comprises 400 whole slide images with a size of 218,000 × 95,000 pixels. Whole-slide images are stored in a multi-resolution structure, including 1×, 10×, 40× magnifying factors. It also has both benign and malignant images [53]. The training dataset in this database provides access to 270 whole slide images, 160 of which are normal slides and 110 slides contain metastases [54]. As this database has been published in a whole-slide image format, the size of the patches can be defined by the individual researchers using this database [55].

2) PRE-PROCESSING

The main database for training our proposed method in our study is Camelyon 16, which is the whole slide images (WSI) in *.tif format from the Camelyon16 challenge. This database has been applied by other researchers like Upadhyay and Awate [29] to train their proposed SRGAN model. Although the original 400 WSI files contain all the necessary information, they are not directly applicable to train our model.

Therefore, we had to sample much smaller image patches that a typical deep learning model can handle. Efficiently, informative and representative patches are two of the most critical parts to achieve good tumor detection performance. To ease this process, we have applied the coordinates of pre-sampled patches used by Li and Ping [56]. Employing the Openslide library in python, we could generate patches of the size of 768 × 768 × 3 at level 0 using 1 process, where the center of each patch corresponds to the coordinates. Then, the patches from the Camylon-16 database were cropped randomly to gain the HR input images with a size of 256 × 256 × 3. Next, the LR input images with the size of 64 × 64 × 3 were obtained by resizing their corresponding HR images. Similarly, in this phase, the images from the BreaKHis database were applied and resized to 256 × 256 × 3 to make them suitable for training the VGG-19 model.

3) TRAINING

The training in this experiment had two phases. The first phase was retraining the VGG-19 model by BreaKHis database with 90 percent of the database for training and 10 percent of them applied for evaluation.

VGG-19 has been retrained for 100 epochs and in epoch 69, we have received the lowest loss error as 0.026. Moreover, the accuracy score that has been gained in this epoch is 98.84 percent for the test dataset and 99.57 for the training dataset. Afterward, the weights of this model with the lowest loss error have been saved as a checkpoint file to calculate the perception loss in the generator network in the next phase. Furthermore, the training VGG model was conducted on the google Colabetory (Colab) service with the Pro version. With Google Colab pro, we had access to the fast GPUs, such as T4 or P100, and a high-memory virtual machine, which is 27.4 gigabytes of available RAM.

In the second phase, three different models, including SRGAN, A-SRGAN, and our proposed model, WA-SRGAN, have been trained through Camelyon 16. We have trained all the networks by using random-vector sample pair of (X^{LR}, X^{HR}) whose X^{LR} is the random image batch of low-resolution images and X^{HR} is its corresponding high-resolution batch images. Since we had enough data images by using databases like Camylon 16, about 10,520 patches with a volume of 10.8 GB out of all have been applied to train the WA-SRGAN model. Also, 301 patches are employed to evaluate the model in each epoch. We have trained the model by random cropping the HR images with the ground truth images with the size of 256 × 256 × 3. All the models have been trained by using the same techniques as follows:

1) At the beginning of the training, the generator has been updated for two epochs.
2) Adam optimizer with the learning rate of 1e-4 has been applied while training.
3) The MSE loss function was utilized to calculate the criterion for the perceptual loss and end up performing the backpropagation.
4) An improved Wasserstein gradient penalty has been applied to train all the models.
5) All the models have been trained for 60 epochs and the performance metrics, and the gradient of the top and the bottom of the network for both generator and discriminator models were observed
6) The sigmoid function has been applied at the end to score the real and generated images by the discriminator to input them in the range of $[1, -1]$ while observing the score of the HR and SR images

To compare our proposed model with the other two models, we have trained the three models, including SRGAN, A-SRGAN, and WA-SRGAN with the ground truth with the size of 256×256×3 and enlarged them with an upscale factor of 4×. Thus, the size of the LR images was 64 × 64 × 3. All of the evaluation results have been saved in a csv file through the Pandas library during training for each epoch. The results of each epoch with the best SSIM metric performance has been selected for each model for our comparison. Some of the models have been trained on google Colab service and some of them have been trained on a server with the Hardware settings that are listed as follows:

1) RAM:125.8GiB
2) Processor: Intel, registered: Xeon(R), CPU E5-2683 v4 @ 2.10GHz × 52
3) Graphics: Quadro P6000/PCIe/SSE2 with 24GB Graphics Card Memory
4) OS: ubuntu 16.04 LTS/ 64-bit

We have tried to train our proposed model, WA-SRGAN, by the CPU as we have 125.8 GB memory RAM. Although we could train the model with a batch size of 64 by CPU, the speed of the training was ten times faster by GPU even with eight batch sizes due to the GPU RAM size (24 GB). Therefore, we have performed all the experiments using the GPU with a batch size of 8. Training the WA-SRGAN by the aforementioned hardware settings has taken about 5 days for 335 epochs. Nevertheless, after epoch 60 we have not observed any improvement; therefore, in this study, we will discuss all the SR experiments for 60 epochs.

C. CLASSIFICATION EXPERIMENTS

1) DATABASE

In the classification phase of our experiment, the BreaKHis was applied as the main database.

2) PRE-PROCESSING

In this phase, we resized the images to 256×256×3 as the HR images and 64 × 64 × 3 as LR images. Then, two methods of the data augmentation techniques, including random rotation 45 degrees and vertical and horizontal flipping were applied. Also, we normalized the input images between 0 and 1. Since we have applied a pre-trained ResNeXt model on the ImageNet data set, in order to abide by the normalization method applied for this data set, each color channel was normalized separately; the means were [0.485, 0.456, 0.406] and the standard deviations were [0.229, 0.224, 0.225]. For the SR images, we used the LR images and normalized each pixel between 0 and 1. After performing the data augmentation techniques, we applied three models, including WA-SRGAN_WN, A_SRGAN, and SRGAN to enlarge the LR images by four times to reach the size of 256 × 256 × 3. Then each channel of the SR images was normalized.

3) TRAINING

In this phase, we performed the classification utilizing a cutting-edge ResNeXt-101(32 × 8d) model. We had fine-tuned this model with three fully connected layers with a size of 1024. The number of parameters of the last three fully connected layers was 3.149 M that were set to be trainable. By using a fine-tuning technique, 3.14 M of the parameters were dedicated to these trainable layers. During the training, we have applied the forward method returning the log-SoftMax for the output and exponential for the class probability [57]. Then, negative log loss as our criterion has been employed to calculate the loss function [58]. We also chose to use Adam optimizer with a learning rate of 0.0002. Furthermore, we used ReLU activations and a dropout of 0.5 for the fine-tuning layers. we applied a 90/10 rate of division for all the examinations. We trained this model on 7,118 images and evaluated the models on 790 images in each epoch.

V. RESULTS AND DISCUSSIONS

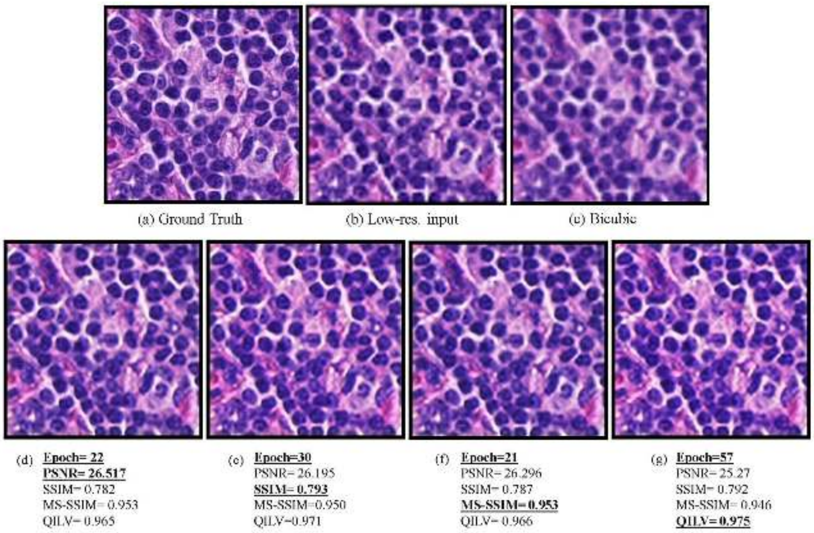
A. SUPER-RESOLUTION RESULTS

In this section, the results of our experiments for the super-resolution part will be discussed and explained. All the models have been trained by 10,520 patches from Camelyon 16 database in each epoch. Consequently, using a batch size of 8 leads to 1,315 input batch images to be fed to the models. After training the models, they were evaluated by 301 patches with a batch size of 1. The following results are based on the evaluation of 301 input images.

During our evaluation, we have observed each model through its objective performance metrics, including MSE, PSNR, SSIM, MS-SSIM, and QILV. Since the lowest MSE result led to the highest PSNR score as these two metrics are correlated, we removed the MSE score and proceeded with the PSNR. Since in each epoch one of the aforementioned performance metrics acquired the highest score, we have illustrated the evaluation results of each epoch along with the scores gained by the WA-SRGAN model with batch normalization technique in Fig. 5 to choose the best performance metric. Fig. 5 part (e) shows the results in epoch 30, in which the model gained the best score for the SSIM metric. As we can observe, in this epoch we gained acceptable results for other metrics as well in comparison with other epochs. Thus, we compared the results of each model based on the epoch with the highest SSIM results.

We performed other experiments on our model by using batch normalization through different upscale factors. All the experiments have been done by the batch size of eight except the factor of two because of the size of input images, which is 128 × 128 × 3 and it is bigger than the input size while using an upscale factor four times or more than that. Thus, we had to use one batch size to cope with the RAM size limitation for the GPU. The results of our experiments on our model with different upscaling factor are shown in Table. 1. As we can see, SSIM metric performance is correlated with other metrics as the bigger the SSIM is, the bigger the other metrics are (Except for the MSE which has to be smaller).

Furthermore, Fig. 6 shows the corresponding results of the WA-SRGAN algorithm with that shown in Table. 1 through different input image sizes and scale factors. In this figure, the ground truth is illustrated in part (a). Also, part (b) is the low-resolution input image with a size of 128 × 128 × 3, and (f) is its corresponding image that has been enlarged by two times. Also, part (c) shows the low-resolution input image with the size of 64 × 64 × 3, and part (g) is the result of our algorithm while increasing the size of the input image by the factor of 4 to reach the size of 256 × 256 × 3. To increase the size of the images by the factor of 8 and 16, the input sizes of the images are considered as 32 × 32 × 3 and 16 × 16 × 3 that are shown in part (d) and (e) respectively. The results of our



(a) Ground Truth

(b) Low-res. input

(c) Bicubic

(d) **Epoch= 22**
PSNR= 26.517
SSIM= 0.782
MS-SSIM= 0.953
QILV= 0.965

(e) **Epoch=30**
PSNR= 26.195
SSIM= 0.793
MS-SSIM=0.950
QILV=0.971

(f) **Epoch=21**
PSNR= 26.296
SSIM= 0.787
MS-SSIM=0.953
QILV= 0.966

(g) **Epoch=57**
PSNR= 25.27
SSIM= 0.792
MS-SSIM= 0.946
QILV=0.975

FIGURE 5. The results of WA-SRGAN in each epoch with the highest result of each metric performance. (a) Is the ground truth of the image with the size of 256 × 256 × 3. (b) Is the low resolution (LR) input image with the size of 64 × 64 × 3. (c) Is the result of the bicubic method. (d) Is the result of epoch 22 of the model that received the highest result for the PSNR. (e) Is the results for epoch 30 with the highest result for SSIM metric performance. (f) Shows the results of the model in epoch 21 in which we have gained the best result for MS-SSIM. (g) Is the results for epoch 57 with the highest score for the QILV metric.

TABLE 1. The Results Of WA-SRGAN By Using Four Different Upscale Factors For The Ground Truth Of 256 × 256 × 3. We Have Performed All The Experiments By Using The Batch Normalization Technique.

Model	Factorial upscale	Input size	Output Size	Batch size	Epoch	MSE	PSNR	**SSIM**	MS-SSIM	QILV
WA-SRGAN	×2	128×128×3	256×256×3	1	25	0.001272	28.7401	**0.9620**	0.9711	0.9960
	×4	64×64×3	256×256×3	8	30	0.00239	26.195	**0.793**	0.9504	0.9716
	×8	32×32×3	256×256×3	8	24	0.007929	21.5559	**0.499**	0.7923	0.7815
	×16	16×16×3	256×256×3	8	10	0.019562	17.8483	**0.3161**	0.5133	0.5499

method after enlarging the images are illustrated in part (h) for 8 times scale factor and part (i) for 16 times scale factor.

Moreover, as the weight normalization technique gained attention for GAN models, we have used both batch and weight normalization with our proposed model to observe the impact of each method on our proposed model. Then, we compared our proposed model, WA-SRGAN (with weight and batch normalization) with other models, including bicubic, SRGAN, and A-SRGAN.

As illustrated in Table. 2, the best results of each model have been compared with others, and our proposed model, WA-SRGAN, with the weight normalization gained the best results in comparison with other models in terms of SSIM, MSE, PSNR, MS-SSIM metric performances. Moreover, our proposed model reached the highest QILV result by using Batch normalization. Additionally, using weight normalization helped us to train the model faster as our model with weight normalization took almost three minutes less than our model with batch normalization. In this examination, the size of input images is 64 × 64 × 3 and the size of output images is 256 × 256 × 3. Fig. 7 shows the evaluation results of five super-resolution methods, including bicubic, SRGAN, A-SRGAN, WA-SRGAN_BN, and WA-SRGAN_WN that are compared through Table. 2.

Similarly, we have trained our WA-SRGAN model with both batch normalization and weight normalization methods with different batch sizes of one and eight to observe the best batch size that is tailored for each method.

Table. 3 shows four different experiments with different methods and batch sizes on our proposed model WA-SRGAN. In this experiment, the best results for SSIM, and MS-SSIM gained by our model with the weight normalization technique and batch size of one. Also, our model gained the best result for the QILV by using batch normalization.

Furthermore, Fig.8 shows the visual evaluation results related to Table. 3. In this figure, part (a) is the ground truth, part (b) shows the low-resolution input image, part (e) shows the results for the bicubic method, part (d) shows the results of WA-SRGAN while using weight normalization, finally, part (e) illustrates the results of WA-SRGAN by batch normalization method.

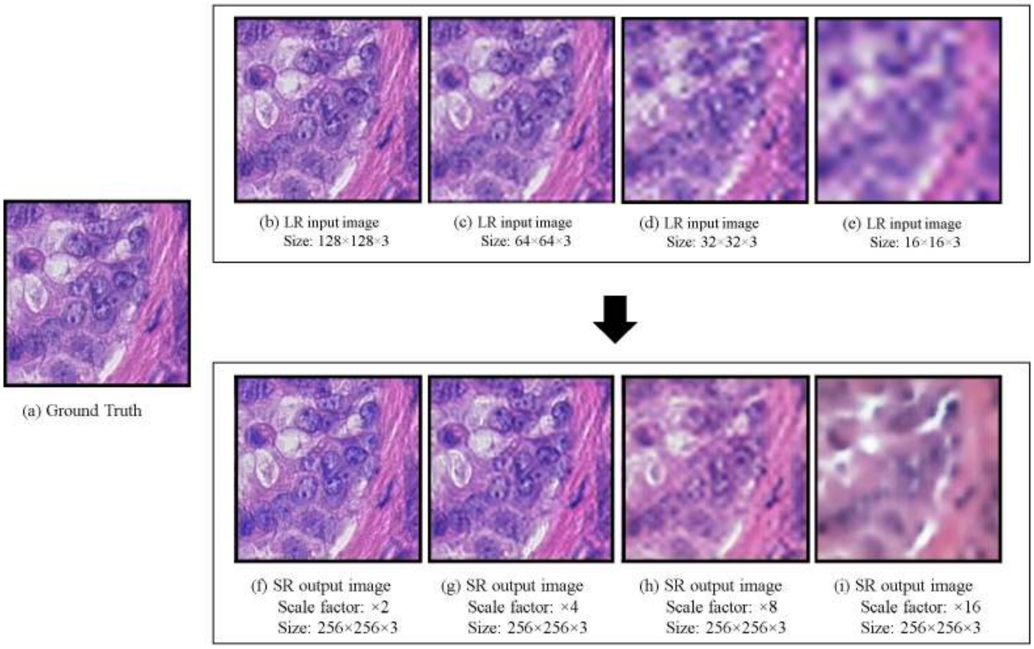


FIGURE 6. The results of the WA-SRGAN model with batch normalization technique to enlarge the images through different scale factors. In this figure, part (a) shows the ground truth of the input image. In the upper row, the low-resolution input images with different sizes, including (b) 128 × 128 × 3, (c) 64 × 64 × 3, (d) 32 × 32 × 3, and (e) 16 × 16 × 3 have been displayed. Also, the images that are located in the lower row are their corresponding enlarged images through different upscale factors, including (f) two scale factor, (g) four scale factor, (h) eight scale factor, and (i) sixteen scale factor.

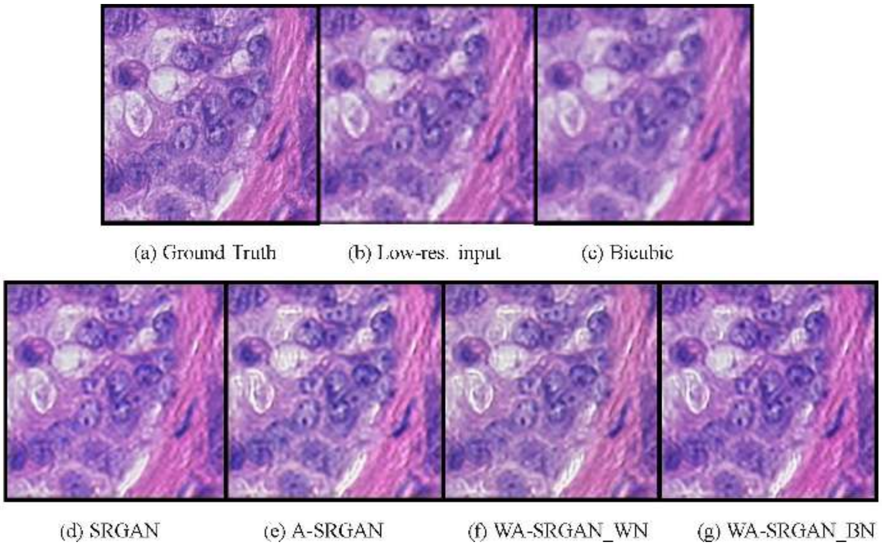


FIGURE 7. The results of all five examined methods, including bicubic, SRGAN, A-SRGAN, WA-SRGAN_WN, and WA-SRGAN_BN to enlarge the images from 64 x 64 x 3 to 256 x 256 x 3. In this figure, (a) shows the ground truth, (b) shows the LR input image, (c) displays the result of the bicubic method, (d) illustrates the result of SRGAN, (e) demonstrates the output gained from A-SRGAN, (f) shows the results gained by our proposed model, with weight normalization technique that is called WA-SRGAN-WN, and (g) is our proposed model by using batch normalization that is named WA-SRGAN-BN.

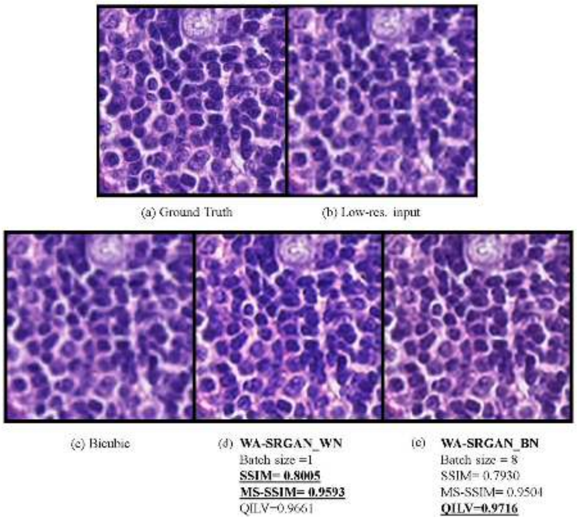
According to Fig.8 part (e), our model with batch normalization method and a batch size of eight could gain the highest QILV results. This metric has been highlighted by some researchers such as Upadhyay and Awate [29]. But we can observe that the color of the results in part (e) is less similar to the ground truth, part (a), than our model with weight normalization and a batch size of one, part (d). According to Fig. 8, better results were gained by our model by using weight normalization, and a batch size of one than using the same model using batch normalization and a batch size of eight. Moreover, the results are visually realistic in terms of color and context while gaining a better SSIM metric. This experiment confirms the importance of the SSIM metric in terms of gaining visually acceptable results.

TABLE 2. The Results Of Five Different Models For Upscaling The Images By Four Times. The Results For Each Epoch Based On The Highest SSIM Metric Have Been Selected. The Models Were Trained With The Same Techniques For 60 Epochs And A Batch Size Of Eight.

Model	Normalization method	epoch	MSE	PSNR	SSIM	MS-SSIM	QILV	Time per epoch (minutes)
Bicubic	-	-	0.00261	25.799	0.7260	0.9360	0.8578	-
SRGAN	Batch Normalization	60	0.00238	26.576	0.7890	0.9503	0.9589	≈16.30
A-SRGAN	Batch Normalization	21	0.00275	26.041	0.7870	0.9437	0.9647	≈18.30
WA-SRGAN_BN	Batch Normalization	30	0.00239	26.195	0.7930	0.9504	**0.9716**	≈21.30
WA-SRGAN_WN	Weight Normalization	55	**0.002188**	**27.2254**	**0.7936**	**0.9586**	0.9652	≈18.30

TABLE 3. The Results Of WA-SRGAN By Using Two Different Methods, Including Batch And Weigh Normalization With The Batch Sizes Of One And Eight.

Model	Method	Factorial upscale	BS	epoch	MSE	PSNR	SSIM	MS-SSIM	QILV	Time per epoch (min)
WA-SRGAN	**WN**	×4	1	57	0.002283	26.9026	**0.8005**	0.9593	0.9661	≈31
		×4	8	55	0.002188	27.2254	0.7936	0.9586	0.9652	≈18.30
	BN	×4	1	20	**0.002165**	**27.4372**	0.7918	0.9585	0.9436	≈31
		×4	8	30	0.00239	26.195	0.7930	0.9504	**0.9716**	≈ 21.30



(a) Ground Truth

(b) Low-res. input

(c) Bicubic

(d) **WA-SRGAN_WN**
Batch size =1
SSIM= 0.8005
MS-SSIM= 0.9593
QILV=0.9661

(e) **WA-SRGAN_BN**
Batch size = 8
SSIM= 0.7930
MS-SSIM= 0.9504
QILV=0.9716

FIGURE 8. The results of three examined methods, including bicubic, WA-SRGAN-WN, and WA-SRGAN-BN to enlarge the images from 64 × 64 × 3 to 256 × 256 × 3. In this figure, (a) shows the ground truth (b) demonstrates the low-resolution input image with the size of 64 × 64 × 3, (c) is the output of the bicubic method, (d) illustrates the result of WA-SRGAN_WN which is our proposed model with weight normalization and a batch size of one, and (d) shows the output that is gained from WA-SRGAN while using batch normalization with the batch size of eight.

TABLE 4. The Results of two-class Classification For LR, HR, And SR Images Separately That Are Gained By ResNeXt-101(32 × 8d) Model.

Model	Image Type	Epoch	Test Loss	Test Accuracy	SSIM
ResNeXt-101	HR	66	0.006	99.49 %	-
	SR (WA_SRGAN)	62	**0.022**	**99.11 %**	**0.8005**
	SR (A_SRGAN)	45	0.034	98.86 %	0.7890
	SR (SRGAN)	58	0.032	98.73 %	0.7870
	LR	47	0.098	95.82 %	0.7260

B. TWO-CLASS CLASSIFICATION RESULTS

According to the general framework that is shown in Fig.4, three types of images, including HR, LR, and SR were fed into the ResNeXt-101(32 × 8d) model. The input image size in this experiment was 256 × 256 × 3. This model was trained for about 70 epochs for each type. Furthermore, we have employed three types of models, including WA_SRGAN_WN, A_SRGAN, and SRGAN in the preprocessing phase of the classification to gain SR images from LR images. We kept the epoch with the lowest error loss score and the highest test accuracy score. The results of these experiments for each type of images were compared through Table. 4. As this table shows, the ResNeXt model gained a 99.49% of accuracy score and a 0.006 loss value for the test dataset in epoch 66 for HR images. Also, classification accuracy reached a value of 95.82% along with a loss value of 0.098 for LR images for the test dataset in epoch 47.

By using the WA_SRGAN model in the preprocessing phase to enlarge the LR images to SR images, and then training and evaluating the ResNeXt model, the results show a 99.11% of accuracy score and a 0.022 loss value for the test dataset in epoch 62. This process was completed for A_SRGAN and SRGAN models as well. ResNeXt model scored 98.86% of accuracy and 0.034 of loss in epoch 45 for the test dataset while using A_SRGAN. Subsequently, the ResNeXt model yielded a 98.73% accuracy score and a 0.032 loss value in epoch 58 for the test dataset while using the SRGAN model in the pre-processing phase.

According to Table. 4, there is a positive relationship between the accuracy results for the job of classification and metric performance in terms of SSIM. Moreover, among all of the SR images that were illustrated in Table. 4, our proposed model, WA_SRGAN with the best SSIM metric performance achieved the highest accuracy score and the lowest loss error while classifying.

Therefore, the results demonstrate the positive impact of using our model on the accuracy score and the loss value in the pre-processing phase before the job of classification.

VI. CONCLUSION

We proposed a novel WA-SRGAN framework for super-resolution that makes the model robust to learn prior information in the training set by using the wide residual blocks and plain self-attention layer before up-sampling in the generator network. The combination of several loss functions, including image loss, adversarial loss, perception loss, and TV loss have been applied in our model to compare high-level differences, like content and style discrepancies, between images. To compute the perception loss function, the pre-trained VGG-19 model has been retrained and applied, and in order to compare the SR and HR images, the MSE metric function has been used.

To fairly correct the generator, a self-attention layer has been applied in the discriminator. Also, the gradient penalty was added to the discriminator criterion through improved Wasserstein extension. This technique enabled us to train the model easily by clipping the gradients of the model in a small range. Similarly, our proposed model has been compared with other algorithms, including SRGAN and A-SRGAN that have been trained with the same techniques and our proposed model achieved the best results in terms of objective metrics in comparison with other models. Also, two different methods, including weight and batch normalizations have been applied in our architecture and compared in terms of several performance metrics, including PSNR, SSIM, MSSSIM, and QILV to evaluate our model. It has been observed that our model works the best through weight normalization with the batch size of one.

Although we could improve the objective metrics by our proposed model, WA-SRGAN-WN, we applied a pre-trained ResNeXt-101 (32 × 8d) model to compare the results of the classification while using our model in the pre-processing phase. The accuracy result gained by the SR images for the classification is 99.11% that is almost the same as the accuracy score obtained for the HR images that is 99.49%. Thus, the results show the positive impact of our model on the resolution of the images as the classification accuracy score can be improved from 96.83% for the low-quality images to 99.11% for the SR images.

ACKNOWLEDGMENT

The authors would like to acknowledge Mr. Mohsen Golshiripour for his support with financial aid. They also wish to acknowledge Dr. Sebelan Danishvar and Mr. Behnam Kiani Kalejahi for their support and assistance with resources. They would also like to acknowledge the help provided by Mr. Amirabbas Qassemi for editing and revising the language of the manuscript.

REFERENCES

[1] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4681–4690. [Online]. Available: https://arxiv.org/abs/1609.04802

[2] H. N. Pathak, X. Li, S. Minaee, and B. Cowan, "Efficient super resolution for large-scale images using attentional GAN," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Seattle, WA, USA, Dec. 2018, pp. 1777–1786, doi: 10.1109/BigData.2018.8622477.

[3] F. Shahidi, S. M. Daud, H. Abas, N. A. Ahmad, and N. Maarop, "Breast cancer classification using deep learning approaches and histopathology image: A comparison study," *IEEE Access*, vol. 8, pp. 187531–187552, 2020, doi: 10.1109/ACCESS.2020.3029881.

[4] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1492–1500. [Online]. Available: https://arxiv.org/abs/1611.05431v2

[5] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "A dataset for breast cancer histopathological image classification," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 7, pp. 1455–1462, Jul. 2016, doi: 10.1109/TBME.2015.2496264.

[6] M. Irani and S. Peleg, "Improving resolution by image registration," *Graph. Models Image Process.*, vol. 53, no. 3, pp. 231–239, May 1991, doi: 10.1016/1049-9652(91)90045-L.

[7] C. E. Duchon, "Lanczos filtering in one and two dimensions," *J. Appl. Meteorol.*, vol. 18, no. 8, pp. 1016–1022, 1979, doi: 10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO.2.

[8] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981, doi: 10.1109/TASSP.1981.1163711.

[9] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. Graph.*, vol. 30, no. 2, pp. 1–11, Apr. 2011, doi: 10.1145/1944846.1944852.

[10] J. Sun, Z. Xu, and H. Y. Shum, "Image super-resolution using gradient profile prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Anchorage, AK, USA, Jun. 2008, pp. 1–8, doi: 10.1109/CVPR.2008.4587659.

[11] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016, doi: 10.1109/TPAMI.2015.2439281.

[12] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 370–378, doi: 10.1109/ICCV.2015.50.

[13] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comp. Vis.* Cham, Switzerland: Springer, 2016, pp. 391–407, doi: 10.1007/978-3-319-46475-6_25.

[14] Y. Wang, L. Wang, H. Wang, and P. Li, "End-to-end image super-resolution via deep and shallow convolutional networks," *IEEE Access*, vol. 7, pp. 31959–31970, 2019, doi: 10.1109/ACCESS.2019.2903582.

[15] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," 2017, *arXiv:1701.00160*. [Online]. Available: http://arxiv.org/abs/1701.00160

[16] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with PixelCNN decoders," in *Proc. Adv. Neural Inf. Process. Syst.*, New York, NY, USA: Curran Associates, 2016, pp. 4790–4798. [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/b1301141feffabac455e1f90a7de2054-Paper.pdf

[17] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, New York, NY, USA: Curran Associates, 2014, pp. 2672–2680. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[19] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[20] J. A. Ferwerda, "Three varieties of realism in computer graphics," in *Proc. 8th Hum. Vis. Electron. Imag.* Bellingham, WA, USA: SPIE, 2003, pp. 290–297, doi: 10.1117/12.473899.

[21] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis. (CoRR)*. Cham, Switzerland: Springer, 2016, pp. 694–711. [Online]. Available: http://arxiv.org/abs/1603.08155

[22] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[23] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, ''Densely connected convolutional networks,'' in *Proc. 30th IEEE Conf. Comp. Vis. Patt. Recog. (CVPR)*, Jul. 2017, pp. 4700–4708. [Online]. Available: https://arxiv.org/abs/1608.06993.

[25] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, ''ESRGAN: Enhanced super-resolution generative adversarial networks,'' in *Proc. Eur. Conf. Comp. Vis. (ECCV)*, 2018.

[26] L. Mukherjee, A. Keikhosravi, D. Bui, and K. W. Eliceiri, ''Convolutional neural networks for whole slide image superresolution,'' *Biomed. Opt. Exp.*, vol. 9, no. 11, pp. 5368–5386, Nov. 2018, doi: 10.1364/BOE.9.005368.

[27] G. Çelik and M. F. Talu, ''Resizing and cleaning of histopathological images using generative adversarial networks,'' *Phys. A, Stat. Mech. Appl.*, vol. 554, Sep. 2020, Art. no. 122652, doi: 10.1016/j.physa.2019.122652.

[28] X. Huang, Q. Zhang, G. Wang, X. Guo, and Z. Li, ''Medical image super-resolution based on the generative adversarial network,'' in *Proc. Chinese Intel. Syst. Conf.* Singapore: Springer, Oct. 2019, pp. 243–253, doi: 10.1007/978-981-32-9686-2_29.

[29] U. Upadhyay and S. P. Awate, ''Robust super-resolution GAN, with manifold-based and perception loss,'' in *Proc. IEEE 16th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2019, pp. 1372–1376, doi: 10.1109/ISBI.2019.8759375.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst.* New York, NY, USA: Curran Associates, 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[31] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, ''Self-attention generative adversarial networks,'' in *Proc. Int. Conf. Mach. Learn.*, Long Beach, CA, USA, Jun 2019, pp. 7354–7363. [Online]. Available: http://proceedings.mlr.press/v97/zhang19d.html

[32] X. Wang, R. Girshick, A. Gupta, and K. He, ''Non-local neural networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803. [Online]. Available: https://arxiv.org/abs/1712.10158.

[33] I. Avcıbas, B. Sankur, and K. Sayood, ''Statistical evaluation of image quality measures,'' *J. Electron. Imag.*, vol. 11, no. 2, pp. 206–223, Apr. 2002, doi: 10.1117/1.1455011.

[34] A. Hore and D. Ziou, ''Image quality metrics: PSNR vs. SSIM,'' in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369, doi: 10.1109/ICPR.2010.579.

[35] G. R. Vidhya and H. Ramesh, ''Effectiveness of contrast limited adaptive histogram equalization technique on multispectral satellite imagery,'' in *Proc. Int. Conf. Video Image Process.*, Dec. 2017, pp. 234–239, doi: 10.1145/3177404.3177409.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, ''Image quality assessment: From error visibility to structural similarity,'' *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[37] Z. Wang, E. P. Simoncelli, and A. C. Bovik, ''Multiscale structural similarity for image quality assessment,'' in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, 2003, pp. 1398–1402, doi: 10.1109/ACSSC.2003.1292216.

[38] S. Aja-Fernandez, R. S. J. Estepar, C. Alberola-Lopez, and C.-F. Westin, ''Image quality assessment based on local variance,'' in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2006, pp. 4815–4818, doi: 10.1109/IEMBS.2006.259516.

[39] S. Ioffe and C. Szegedy, ''Batch normalization: Accelerating deep network training by reducing internal covariate shift,'' 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[40] N. Kumaran and A. Vaidya, ''Batch normalization and its optimization techniques: Review,'' *Int. J. Eng. Res. Comput. Sci. Eng.*, vol. 4, no. 8, pp. 211–215, Aug. 2017.

[41] T. Salimans and D. P. Kingma, ''Weight normalization: A simple reparameterization to accelerate training of deep neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.* New York, NY, USA: Curran Associates, 2016, pp. 901–909.

[42] I. Gitman and B. Ginsburg, ''Comparison of batch normalization and weight normalization algorithms for the large-scale image classification,'' 2017, *arXiv:1709.08145*. [Online]. Available: http://arxiv.org/abs/1709.08145

[43] S. Xiang and H. Li, ''On the effects of batch and weight normalization in generative adversarial networks,'' 2017, *arXiv:1704.03971*. [Online]. Available: http://arxiv.org/abs/1704.03971

[44] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, and T. Huang, ''Wide activation for efficient and accurate image super-resolution,'' 2018, *arXiv:1808.08718*. [Online]. Available: http://arxiv.org/abs/1808.08718

[45] M. Lin, Q. Chen, and S. Yan, ''Network in network,'' 2013, *arXiv:1312.4400*. [Online]. Available: http://arxiv.org/abs/1312.4400

[46] K. He, X. Zhang, S. Ren, and J. Sun, ''Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[47] J. Zhang, T. He, S. Sra, and A. Jadbabaie, ''Why gradient clipping accelerates training: A theoretical justification for adaptivity,'' 2019, *arXiv:1905.11881*. [Online]. Available: http://arxiv.org/abs/1905.11881

[48] M. Arjovsky, S. Chintala, and L. Bottou, ''Wasserstein GAN,'' 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875

[49] J. Hany and G. Walters, ''Best practices for model design and training,'' in *Hands-On Generative Adversarial Networks With PyTorch 1. X: Implement Next-Generation Neural Networks to Build Powerful GAN Models Using Python*. Birmingham, U.K.: Packt, 2019, pp. 56–73.

[50] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, ''Improved training of Wasserstein GANs,'' in *Proc. Adv. Neural Inf. Process. Syst.* New York, NY, USA: Curran Associates, 2017, pp. 5768–5778.

[51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, ''Pytorch: An imperative style, high-performance deep learning library,'' in *Proc. Adv. Neural Inf. Process. Syst.* New York, NY, USA: Curran Associates, 2019, pp. 8024–8035.

[52] M. Jannesari, M. Habibzadeh, H. Aboulkheyr, P. Khosravi, O. Elemento, M. Totonchi, and I. Hajirasouliha, ''Breast cancer histopathological image classification: A deep learning approach,'' in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2018, pp. 2405–2412, doi: 10.1109/BIBM.2018.8621307.

[53] P.-H.-C. Chen, K. Gadepalli, R. MacDonald, Y. Liu, S. Kadowaki, K. Nagpal, T. Kohlberger, J. Dean, G. S. Corrado, J. D. Hipp, C. H. Mermel, and M. C. Stumpe, ''An augmented reality microscope with real-time artificial intelligence integration for cancer diagnosis,'' *Nature Med.*, vol. 25, no. 9, pp. 1453–1457, Aug. 2019, doi: 10.1038/s41591-019-0539-7.

[54] (2016). *Camelyon16 Challenge on Cancer Metastases Detection in Lymph Node*. [Online]. Available: https://camelyon16.grand-challenge.org

[55] Z. Guo, H. Liu, H. Ni, X. Wang, M. Su, W. Guo, K. Wang, T. Jiang, and Y. Qian, ''A fast and refined cancer regions segmentation framework in whole-slide breast pathological images,'' *Sci. Rep.*, vol. 9, no. 1, pp. 1–10, Dec. 2019, doi: 10.1038/s41598-018-37492-9.

[56] Y. Li and W. Ping, ''Cancer metastasis detection with neural conditional random field,'' 2018, *arXiv:1806.07064*. [Online]. Available: http://arxiv.org/abs/1806.07064

[57] A. de Brébisson and P. Vincent, ''An exploration of softmax alternatives belonging to the spherical loss family,'' 2015, *arXiv:1511.05042*. [Online]. Available: http://arxiv.org/abs/1511.05042

[58] D. Zhu, H. Yao, B. Jiang, and P. Yu, ''Negative log likelihood ratio loss for deep neural network classification,'' 2018, *arXiv:1804.10690*. [Online]. Available: http://arxiv.org/abs/1804.10690

FAEZEHSADAT SHAHIDI received the B.Sc. degree in applied course of computer software after attending the University of Erfan Higher Education Institute (Sealed and signed by Selected the University of Kerman province), Iran, and the master's degree in master of science business intelligence and analytics from Universiti Teknologi Malaysia. Her research interests include computer vision, image processing, big data, cloud computing, social media Web, graph theory, machine/deep learning, generative adversarial networks, the IoT, and blockchain.