Super-resolution using 3d convolutional neural networks in CT scan image of COVID19

Rafaa Amen Kazem*[1], Jamila H.Suad[2], Huda Abdulaali Abdulbaqi[3]

[(1,2,3)]Mustansiriyah University, College of Science, Computer Science Dept., Baghdad, Iraq.
[1]rafah.amen.kazem@gmail.com

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract

In the medical field, the accuracy and quality of the image led to a rapid increase in using deep learning models-based architectures convolutional nerve CNN where three-dimensional convolutional neural networks (3D) CNNs utilized to analyze the medical images. This paper provides the pre-processing steps for the medical images then fed to the CNNs 3D. The objective of the proposed model is to operate from low-resolution to reach high-resolution images by using Keras with TensorFlow as wallpaper for training; this training is work separately on the different scaling factors for reconstruction. Two datasets of lung tomography data are used in this paper, the first dataset is Kaggle.coved2020 for Corona patients, and the second is the Iraqi dataset for Corona patients 2020. The proposal method that demonstrated improvements over the bicubic upsampling and FSRCNN peak signal-to-noise ratio (PSNR) 37.10 dB with 40.47 dB respectively.

Keywords: 3D FSRCNN, scaling factors, MSE, PSNR, SSIM**.**

1. Introduction

Over the past decades, Artificial intelligence takes inspiration from the elements of statistics, computer science, and mathematics [1]. Machine Learning (ML) is a sub branch of AI; it has a role in creating algorithms that learn from and make predictions on data [2]. In addition, it has been developed due to the continuous capability of machines in handling massive input information, while a few machines might even identify the hidden patterns as well as complex relations for making reliable and appropriate decisions in which human beings couldn't. ML and AI, particularly in biomedical imaging, improve the capabilities of researchers and doctors in understanding the way for analyzing generic variations which are going to cause disease, for instance, in X-ray CT and MRI [3]. Those approaches are composed of traditional algorithms with no learning, such as SVM, NN, KNN, and so on. also, the deep learning algorithms like CNN, RNN, LSTM, ELM, GAN, and so on. [4]. Commonly, the word deep indicates the existence of various layers in ANN, yet such meaning was changed over time [5]. Standard CNN rely developed on the learned convolutional filters that allow them to be adjusted flexibly to the available data via considering the two ideas; preserving the image's spatial locality (and, generally, in any information type) and learning through abstraction's progressive levels: with a single layer, just learn simple patterns might be utilized; with multilayer, learn multiple patterns might be utilized.

The interplay between CNN and image super-resolution (SR) approaches, which indicates the process of the recovery of the high resolution images from a sequence of low resolution ones (noisy) is considered as a significant class concerning the techniques of image processing in image processing and computer vision [6][7]. The super resolution CNN (SRCNN) as a simple network structure and excellent restoration quality doesn't need any engineered features, which were commonly required in other approaches and showing excellent performance [8, 9]. Generally, the algorithms of SR utilize deep learning methods are different in the next main aspects: various network architecture types, various loss function types, various forms of learning strategies and principles [10].

Many algorithms are accelerating SRCNN via achieving convolution on LR images as well as replacing predefined up-sampling operators with sub pixel convolution or transposed convolution (de-convolution). These approaches utilize the fairly small networks and could not learn complicated mappings because of limited network capacity [11]. The rest of the paper includes, section 3 Methodology of Proposed 3DFSRCNN, section 4 Pre-process of Training-Set, section 5. Evaluation of Performance, section 6 Experimental Results, section 7 3DFSRCNN Performance, section 8 Conclusions.

2. Related Work

SRCNN can be defined as a state for a representative of the art approach in deep learning based SR approach; thus, we analyzed and comparison in our proposed methods with appropriate research. 3DFSRCNN models are composed of six layers.

Dong *et al.*, [12, 13] were elucidated two types of specifies, one of the original LR image require bicubic interpolation by up-sampling to the required size for forming the input; therefore, grow to spatial size related to HR image. On the other hand, the non linear mapping step needs a high-cost value so investigate to shrink the network scale, whereas keeping the previous accuracy mediated via adding adding a shrinking and expanding layer separately then place the de-convolution layer at the network's end for restricting mapping in a low-dimensional feature space, after that, the computational complexity is just proportionate with the spatial size that is associated with original LR image [14].

Kim *et al.*, [9, 15], SRCNN is learning an end-to-end mapping directly between HR and LR images and distinctive from conventional learning-based approaches, resulting in accurate and fast inference when reached too, respectively.

Wang *et al.* [16] replaced the mapping layer via a set of sparse coding sub networks and suggested a sparse coding based network (SCN), which is superior to SRCNN with a small model size. Furthermore, all previous networks in [9, 15, 16] needed to process bicubic upscaled LR images, and FSRCNN needs different de-convolution layer which proposed not just perform on original LR image, yet also containing more efficient and simpler mapping layer with faster upscale of the image in different sizes.

Where another approach in 2018 and 2020 [17] [18] reported extends SRCNN to MRI or CT imaging, they demonstrate that 3D SRCNN and FSRCNN, respectively, effectively keeping the contours and structure in 3D CT and MRI scans. The limitations of SRCNN more time requires and FSRCNN making them less realistic and consumed a memory with massive expense respectively.

3. Methodology of Proposed 3DFSRCNN

In the present study, we present a three-dimensional Fast SR Convolutional Neural Network (3D FSRCNN) architecture consisting of 6-layers of 3D-CNN as shown in Fig.1.

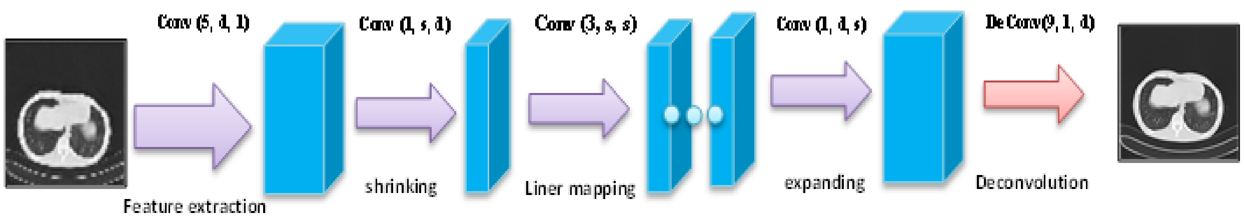


Fig.1: Proposed Architecture of 3DFSRCNN

Figure1 explains the suggested architecture of 3DFSRCNN phases as follows:

3.1 Network Structure

For volumetric super-resolution, a 3D grid structure is proposed, called 3D FSRCNN, to attain SR volumetric CT images, as has been illustrated in Fig.2. The whole network contains dozens of variables that cannot analyze all of them. Thus, leave them to be tuned by sensitive variables, which is also mirroring a few significant catalytic scaling factors in SR; also, in these phases, the proposal is consist of six layers as shown:

1. **The first layer (Feature Extraction):** in this layer, the LR image is directly used as input to a system rather than on interpolated image. Filter size or kernel size (f_1) is 5x5 used for appropriate image size. Thus, at this stage f_1 = 5, c_1 = 1 and output of filters is n_i. Let n_i at this layer be represented via d. The filter number has been considered to be a sensitive variable; thus, it is able to perform the first layer as Conv (5, d, 1) as the specific formula is as follows:

$$F_1(X) = \max(0, W_1 * X + B_1) \text{ -------- } (1)$$

Where X is a low resolution input image,W_1 is a convolution kernel, and B_1 is a bias

2. **The second layer (Shrinking)**: FSRCNN has a shrinkage layer to lower the dimensionality of d. In addition, the linear regression d features are quite big, and for the 3D CT scan, it will be even bigger. Therefore, a single filter is used to reduce math costs. Thus, $f_2 = 1$ for reducing the dimensionality of d. Also, the filter number n_2 = s has to be less than d for minifying LR feature dimension; which is why, the 2nd layer might appear as Conv (1, s, d). The specific formula is as follows:

$F_2(X) = \max(0, W_2 * F_1(X) + B_2)$ ------- (2)

3. **The third layer (Hidden layer):** maps the features of the LR image to HR images. It has the biggest share in the SR performance overall analysis of the deep networks has been carried out in that keeping the filter size as 3x3 is a big tradeoff among pretty performance and network scale. Our study complies with their analysis and preserves Convolution layers f_3 =3 and $n_3 = s$, Conv (3, s, s) as the specific formula is as follows:

$$F_3(X) = \max(0, W_3 * F_2(X) + B_3) \ldots\ldots (3)$$

4. **The fourth layer (Hidden layer):** In this step, we maintain the filter size $f_3 = f_4$= 3 and $n_4 > n_3$. This layer can refer to be Conv (3, s1, s). The specific formula is as follows.

$$F_4(X) = \max(0, W_4 * F_3(X) + B_4) \text{ -------- } (4)$$

5. **Fifth layer (Expanding):** This layer nullifies the shrink layer, reducing the dimensions of the LR feature for computational capacity. Creating images without expansion may create poor quality images. Filter size used f_5 = 1, as for the shrink layer. The layer has been shrinking Conv (1, s, d), so Conv (1, d, s) would be the expanding layer. The specific formula is as follows.

$$F_5(X) = \max(0, W_5 * F_4(X) + B_5) \text{ ------ } (5)$$

6. **Sixth layer (De-convolution):** This layer is accountable for upgrading of level and assembly of previous features with the use of de-convolution filters, which bypass inversion process convolution. A stride of the first five layers will be equal to its default value of one. As in a convolution process stride, k to outputs 1/k represents the number of input times, and due to the fact that de-convolution is considered as a reverse process, stride k raises input by a k factor. we can take an interest in this and set the phase k = n as a scaling factor, leading to re-constructed, upscaled HR image. This study assumes that the filter size f_6 must be 9 × 9. Therefore, the layer might be specified as DeConv (9, 1, d) and stride of n that has been set to the required scaling. The specific formula is as follows.

$$F_6(X) = W_6 * F_5(X) + B_6 \text{ -------- } (6)$$

3.2 Activation Functions

i. **Rectified Linear Unit (ReLU)** is used to activate after the first five convolution layers, while the sixth layer of de-convolution can use the linear function. ReLU can be defined as an activation function utilized in NNs, majorly in CNNs. A simple calculation, as there aren't complicated calculations. The FSRCNN 3D model might consume little time in each stage of training or testing, and it converges more quickly.

$Re\,LU\ (X) = max(0, W_X + B)$ ------- (7)

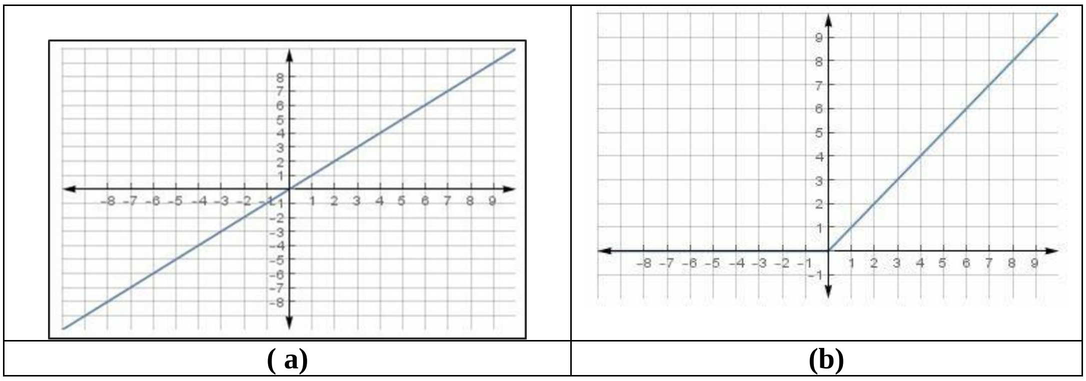
Where **X** and **W** are denoted as input and weight parameters, respectively, b is bias. It's illustrated in figure (2b).

ii. **The linear function** is considered as a polynomial of the degree of 1 or less, involving zero polynomial (the latter doesn't have degree 0). When the function is of one variable only, it will be of the form:

$$f(x) = ax + b\ ---- (8)$$

Where **a** and **b** were constants, sometimes real numbers. The graph of such a function of one variable is non vertical line and often indicated as the line slop, and b is referred to as intercept. In addition, the linear means that **x** gets grows, the downhill does not "slake" or plateau (Rounding to values closer to pixel values), as in the figure below (2a). For a function $f(x1, x2, \ldots\ldots.., xk)$ of any finite number of variables, the general formula is:

$$f(x_1, x_2, \ldots., x_k) = b + a_1x_1 + a_2x_2 + \ \ldots\ldots + a_kx_k\ ----- (9)$$



(a) **(b)**

Fig2: Activation function type: a) Linear function and b) ReLU function

3.3 Loss Function and Optimizer

FSRCNN uses MSE as a cost function; the major aim is reducing MSE between the original and recovered image. Adam optimizer might be used for minimizing MSE loss. This is considered the MSE between real values and predictions. Mathematically, it is a vector of n predictions, and Y represents the vector of n spotted values, in this case they satisfy the equation:

$$MSE = \frac{1}{n}\sum_{0}^{n}\left(\mathrm{Y^{i}} - \mathrm{Yi}\right)^{2} \quad ----(10)$$

Where ($\mathrm{Y^i}$) recovered image and (Yi) original image.

3.4 Adjustable Learning Rate

It has been observed in the FSRCNN where the rally is extremely slow in the case when there is a small learning rate. The learning rate helps reinforce training but may trigger an explosion of gradients. To avoid bursting gradation and speed up training, we will use an adjustable learning rate strategy. In early epochs, for speeding up the process of training, the learning rate is going to be fairly high. As the training epoch continues, the learning rate is decreased with the next rules.

$$\mathrm{lr} = \mathrm{lr} * 0.1^{\frac{(epoch)}{step}} \quad ----- (13)$$

Where epoch counts current training times and step is pre-defined for controlling the drop of the rate of the learning.

4. Pre-process of Training-Set

prior to training, normalization function, mod cropping into 3D-sub-blocks to produce the training set and rotating by 90° original CT-image sample should be used in a suitable format. Also, the original image should be downsampling by using scaling factor=2, 3, 4. We called the original CT- datasets {Y} in various samples, and these are utilized as LR images{X}. In Figure 3 HR{Y} images are shown as a classification to compute the loss function, and the LR {X} images are fed into the grid.

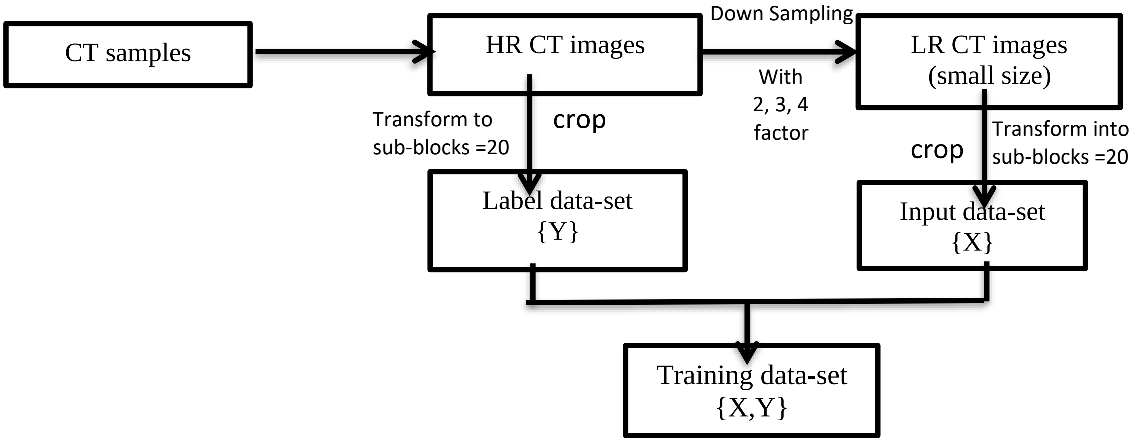


Fig3: Procedure of the production of the training set.

5. Evaluation of Performance

Many metrics are utilized for comparing various models' performances.

i. **PSNR**: this is the most prevalent approach utilized for determining the results' quality; it might be directly evaluated from MSE by means of eq.14, in which L represents the maximum pixel value possible (255 for an 8-bit image).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(I(i) - \hat{I}(i))^2$$

$$PSNR = 10.\log_{10}\left(\frac{L^2}{MSE}\right)$$

ii**. SSIM**: This metric is utilized for comparing the perceptual quality regarding 2 images with the use of eq.15, with the variance (σσ), mean (μμ) and correlation (c) of both images.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} - - - - - - (15)$$

6. Experimental Results

Our search provides evaluation metrics PSNR and SSIM that used to evaluate image quality, then investigated the impact of the important parameters on the accuracy of the reconstruction, as well as to conduct extensive experiments. The network was trained on two sets of Corona lung patients tomography data (CT scan); the first dataset group was from the Kaggle as a globally site (Kaggle.coved2020) [19] and the locally from AL- Kindi and AL- Qanaat diagnostic centers/Iraq.

Training Implementation

3D FSRCNN model is implemented with the use of Keras with TensorFlow as wallpaper and using some others image processing libraries like OpenCV and others. They have been trained for epoch = 10, learning rate = 0.001, Batch size = 64, and the size of the filter has an effect on the velocity of spread. If it is a large size, it will be increasing the complexity and the memory cost for training, but the performance may improve. Because of this trade-off and high data complexity of 3D models where table 1 explains the layers (types) with their output shape and Parameters (parm).

Table1: Training Implementation for each layer.

Layers (types)	Output shape	Parm#
Conv2d-1 (Conv2D)	**(None, None, None,48)**	**1248**
Conv2d-2 (Conv2D)	**(None, None, None, 16)**	**784**
Conv2d-3 (Conv2D)	**(None, None, None, 16)**	**2320**
Conv2d-4 (Conv2D)	**(None, None, None, 216)**	**31320**
Conv2d-5 (Conv2D)	**(None, None, None, 48)**	**10416**
Conv2d-transpose-**1 (conv2DTr**)	**(None, None, None, 1)**	**3889**
Total parms: 49,977 **Trainable parms: 49,977**		

7. 3DFSRCNN performance

The effect of the number of epochs used to train the network 3DFSRCNN, and its effect on performance measures are investigated. The implementations of results are used: epochs = 10, 20, and 30 with the learning rate 0.001 and batch size = 64 with the scale factor = 2 and the data used are local data and noticed a slow progressive in values of PSNR, SSIM, and MSE. Clearly, CNN network-based methods achieve better results in performance than interpolation-based methods. Also, using 3DCNN will achieve better performance than the network 2D. Learning rate of 0.00001, epochs=10, scaling factor =2 are used in kaggel dataset, and the results of PSNR=35.72, SSIM =0.9408, MSE= 0.18. But if the measurement is used factor = 4, the results of PSNR=27.60, SSIM=0.8140, MSE=1.18. A rapidly increase can be seen in all values, but if it is compared with the results of

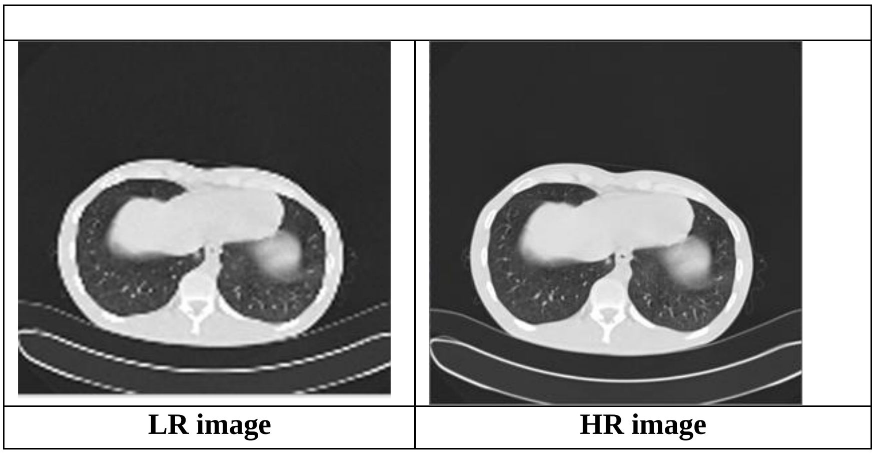
Table2, we notice that it is less, so the change of the learning rate to 10^{-3} leads to increase the values. This same experience has been applied to local data, and the results were PSNR =39.48, SSIM= 0.9513, MSE=0.08 in the case of the scale factor=2 and if the scaling factor = 3, the results are PSNR=34.01, SSIM=0.8939, MSE=0.28. Thus, we notice a clear increase in metric values.

These results of 3DFSRCNN are illustrated in table 2, and figure **4** (**A, B, C**) used two types of datasets (globally and locally) with various values of scale factor (SF 2, 3, & 4) and the extent of their influence on the performance measures within 3DFSRCNN and Bicubic interpolation. Could record some conclusions, used SF 4 in LR image, also

Table2: the comparison between two types of datasets, epochs = 10 and learning rate = 0.001 values are fixed.

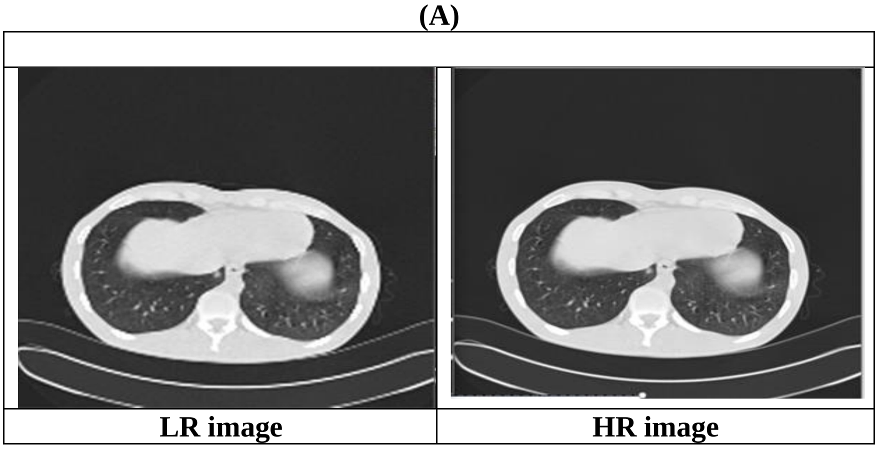
Dataset	Scaling Factor	Bicubic	FSRCNN
		PSNR/SSIM/MSE	PSNR/SSIM/MSE
Iraq.coved2020	2	37.10/0.9461/0.13	40.47/0.9536/0.07
Kaggle.coved2020	2	31.21/0.9356/0.49	36.87/0.9245/0.13
Iraq.coved2020	3	30.66/0.8742/0.57	35.51/0.9061/0.20
Kaggle.coved2020	3	25.26/0.8385/1.95	32.39/0.8713/0.40
Iraq.coved2020	4	27.19/0.8242/1.25	32.69/0.8757/0.37
Kaggle.coved2020	4	22.88/0.7853/3.37	30.21/0.8635/0.67

Figure 4 shows **the results of FSRCNN**, where image (**A)** led to noisier than SF 3 in the image (B) with epochs = 10 and lr = 0.001 for each case, also could increase the resolution to PSNR as in image (C) by making it equal to 20 epoch with use same SF 4 (epochs = 20, lr = 0.001 and SF4). As shown bellow



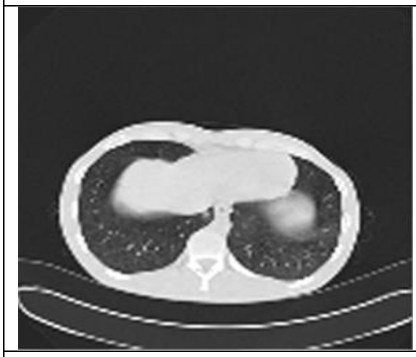
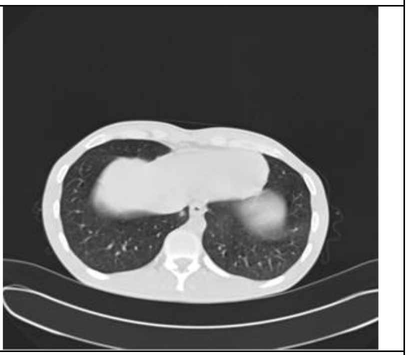
LR image **HR image**

(A)



LR image **HR image**

(B)

	
LR image	**HR image**

(C)

Fig. 4: illustrated the results of FSRCNN. (A) epochs = 10, lr = 0.001, scale = 4, (B) epochs = 10, lr = 0.001, scale = 3 (C) epochs = 20, lr = 0.001, scale = 4

8. Conclusions

The presented study suggested a 3D CNN enhance resolutions of CT images of people lung injured with COVID-19 disease. Our experiments showed better performance results by learning the proposed model parameters on medical images. In our proposed model, we converted the low resolution of gathering dataset images to super high resolution images to assist the decision-makers in diagnosing the disease without any error that may be corrupted. A comparison of the results between the proposed 3DFSRCNN and Bicubic interpolation where the training required a unique character as in our case when used more than 8 GB memory due to the huge time-consuming in the CPU, an approach which uses CUDA to call up GPU resources.

Acknowledgments: Authors thank the Dept. of Computer Science, College of Science, Mustansiriya Univ. for supporting this work.

References:

[1] Russell, S. J., & Norvig, P. (2020). Instructor's solution manual for artificial intelligence: a modern approach.4th edition. *Prentice Hall.* ISBN 0-13-461099-7. OCLC 359890490 (https://www.worldcat.org/oclc/359890490). Website (http://aima.cs.berkeley.edu).

[2] Bishop, C. M. (2006). Pattern recognition and machine learning. *Springer-Verlag.* 1st edition. ISBN: 978-0-387-31073-2.

[3] Ni, D., Xiao, Z., & Lim, M. K. (2019). A systematic review of the research trends of machine learning in supply chain management. International Journal of Machine Learning and Cybernetics, 1-20.

[4] Razzak, M. I., Naz, S., & Zaib, A. (2018). Deep learning for medical image processing: Overview, challenges and the future. In Classification in BioApps (pp. 323-350). Springer, Cham.

[5] Gulli, A., Kapoor, A., & Pal, S. (2019). Deep learning with TensorFlow 2 and Keras: regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API. Packt Publishing Ltd.

[6] Bruna, J., & Mallat, S. (2013). Invariant scattering convolution networks. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1872-1886.

[7] Ulicny, M., Krylov, V. A., & Dahyot, R. (2020). Harmonic Convolutional Networks based on Discrete Cosine Transform. arXiv preprint arXiv:2001.06570.

[8] Timofte, R., De Smet, V., & Van Gool, L. (2014). A+: Adjusted anchored neighborhood regression for fast super-resolution. In Asian conference on computer vision (pp. 111-126). Springer, Cham.

[9] Kim, J., Kwon Lee, J., & Mu Lee, K. (2016). Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1646-1654).

[10] Sajjadi, M. S., Scholkopf, B., & Hirsch, M. (2017). Enhancenet: Single image super-resolution through automated texture synthesis. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4491-4500).

[11] Lai, W. S., Huang, J. B., Ahuja, N., & Yang, M. H. (2017). Deep laplacian pyramid networks for fast and accurate super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 624-632).

[12] Dong, C., Loy, C. C., He, K., & Tang, X. (2014). Learning a deep convolutional network for image super-resolution. In European conference on computer vision (pp. 184-199). Springer, Cham

[13] Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence, 38(2), 295-307

[14] Dong, C., Loy, C. C., & Tang, X. (2016). Accelerating the super-resolution convolutional neural network. In European conference on computer vision (pp. 391-407). Springer, Cham.

[15] Kim, J., Kwon Lee, J., & Mu Lee, K. (2016). Deeply-recursive convolutional network for image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1637-1645).

[16] Wang, Z., Liu, D., Yang, J., Han, W., & Huang, T. (2015). Deep networks for image super-resolution with sparse prior. In Proceedings of the IEEE international conference on computer vision (pp. 370-378).

[17] Wang, Y., Teng, Q., He, X., Feng, J., & Zhang, T. (2018). Ct-image super resolution using 3d convolutional neural network. arXiv preprint arXiv:1806.09074.

[18] Mane, V., Jadhav, S., & Lal, P. (2020). Image Super-Resolution for MRI Images using 3D Faster Super-Resolution Convolutional Neural Network architecture. In *ITM Web of Conferences* (Vol. 32). EDP Sciences.

[19] https://www.kaggle.com/datasets?search=covid+19.