Jurnal Elektronika dan Telekomunikasi (JET), Vol. 24, No. 2, December 2024, pp. 112-119
Accredited by KEMDIKBUDRISTEK, Decree No: **158/E/KPT/2021**
doi: 10.55981/jet.653

Enhancing Remote Sensing Image Resolution Using Convolutional Neural Networks

Julian Supardi [a, *], Samsuryadi Samsuryadi [a], Hadipurnawan Satria [a], Philip Alger M. Serrano [b], Arnelawati Arnelawati [a]

[a] *Department of Informatics Engineering*
Sriwijaya University
Jl. Srijaya Negara-Bukit Besar,
Palembang, Indonesia 30129
[b] *College of Computer Studies*
Camarines Sur Polytechnic Colleges,
San Miguel, Nabua, Camarines Sur, Philippines 4434

Abstract

Remote sensing imagery is a very interesting topic for researchers, especially in the fields of image and pattern recognition. Remote sensing images differ from ordinary images taken with conventional cameras. Remote sensing images are captured from satellite photos taken far above the Earth's surface. As a result, objects in satellite images appear small and have low resolution when enlarged. This condition makes it difficult to detect and recognize objects in remote-sensing images. However, detecting and recognizing objects in these images is crucial for various aspects of human life. This paper aims to address the problem of remote sensing image quality. The method used is a convolutional neural network. Our proposed method consists of two main parts: the first part focuses on feature extraction, and the second part is dedicated to image reconstruction. The feature extraction component includes 25 convolutional layers, whereas the reconstruction component comprises 75 convolutional layers. To validate the effectiveness of our proposed method, we employed the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) as evaluation metrics. The test datasets consisted of Landsat-8 images, which were segmented into three regions of interest (ROI) of sizes 16×16 pixels, 24×24 pixels, and 32×32 pixels. The experimental results demonstrate that the PSNR/SSIM values achieved were 28.94/0.822, 30.24/0.089, and 33.24/0.925 for each respective ROI. These results indicate that the proposed method outperforms several state-of-the-art techniques in terms of PSNR and SSIM.

Keywords: remote sensing, convolutional neural network, image enhancement, deep learning, object recognition.

I. INTRODUCTION

Artificial Intelligence (AI) primarily focuses on developing computerized systems that enable software to work like living creatures in solving problems. Regarding decision-making techniques, AI algorithms can be divided into two broad categories: algorithms that imitate animal behavior [1] and algorithms based on human thinking [2]. The first group includes Ant Colony Optimization [3], Particle Swarm Optimization [4], Genetic Algorithms [5], Bee Colony Optimization [6], and others. Meanwhile, algorithms that imitate humans in solving problems include fuzzy logic[7], Support Vector Machines (SVM) [8], Expert Systems [9], Artificial Neural Networks (ANNs) [10], [11], and more.

One branch of AI that has developed rapidly in the past decade is Deep Learning (DL), which is an extension of ANNs [12]. This field gained significant attention following the success of several ANN models in the ILSVRC competition, including AlexNet (2012) [13], Clarifai (2013) [14], GoogLeNet (2014) [15], and ResNet (2015) [16]. Building on this success, deep learning has been widely applied in various fields, such as classification, forecasting, image enhancement, remote sensing, and more.

On the other hand, the problem of detecting and recognizing objects in remote-sensing images has been a major focus for researchers over the last three decades. The main goal of object detection and recognition in remote-sensing images is to quickly and accurately locate and identify objects of interest to survey within the vast expanse of remote-sensing images.

Remote sensing technology has advanced significantly, enabling the capture of intricate details such as contours, colors, textures, and other distinctive attributes [17]. Nevertheless, object detection algorithms face numerous formidable challenges. This complexity arises from the differences in acquisition methods employed for remote optical sensing imagery compared to those used for natural imagery. Remote sensing imagery utilizes sensors, including optical, microwave, or laser devices, to gather data about the Earth's surface by detecting and recording radiation or reflections across various spectral ranges. In contrast, natural images are captured using electronic devices, such as cameras, or sensors that capture visible light, infrared radiation, and other forms of radiation present in the natural environment to obtain everyday image data. Unlike natural images captured horizontally by ground cameras, satellite images are obtained from an aerial perspective,

* Corresponding Author.
Email: julian@unsri.ac.id
Received: July 08, 2024 ; Revised: September 02, 2024
Accepted: November 21, 2024 ; Published: December 31, 2024

providing extensive imaging coverage and comprehensive information about the Earth's surface in the areas where the images are acquired.

Given those characteristics, detecting and recognizing objects in remote-sensing images represents one of the most complex tasks in pattern recognition. This is due to the satellite's distant position, causing the object to appear very small. Despite efforts that have been made to enlarge the remote sensing image, the resulting image of the object still has low resolution. These low-resolution object images present a challenge in object detection and recognition based on remote sensing images. This is because a subtle difference between pixels in low-resolution images makes it difficult for computers to distinguish between individual objects effectively.

This study aims to improve the quality of object images in remote sensing images. Improving image quality is essential for addressing the challenges associated with object detection in remote-sensing images. This enhancement is typically evaluated using two standard metrics in image processing: peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). In this context, higher PSNR and SSIM values indicate superior image quality.

Several methods, including bicubic interpolation, SRCNN, and DCSCN, have been proposed to address the problem of increasing image resolution. However, the results still require improvement, especially when dealing with extremely low-resolution images, such as object images in remote sensing data.

The main contribution of this research is a relatively simple convolutional neural network (CNN) architecture that uses convolutional layers to improve the quality of remote-sensing images. This architecture can be combined with various architectures to recognize objects in remote-sensing images.

The rest of this paper is structured as follows. Section 1 introduces the introduction and the motivation. Section 2 discusses the proposed method in detail. Section 3 presents the experiments, and the final section provides the concluding remarks.

II. METHODS

A. Datasets

The datasets used in this study are of two types, training data and testing data. Data for training comes from Yang et al. [18] and the Berkeley Segmentation Database [19]. Both databases contain high-resolution images, and the data sizes vary. Both databases are commonly used in image resolution improvement research, such as in [20], [21], [22].

The next data set is for testing. It is obtained from remote sensing images produced by the Lansat 8 Satellite, downloaded from the official website of GIS Geography (https://gisgeography.com/landsat/). The illustration of the image for the dataset is shown in Figure 1.

The image can be downloaded by following these steps:

Step 1. Set your area of interest in the "Search Criteria" tab

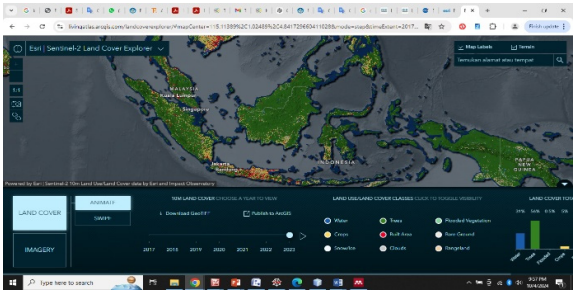

Figure 1. Capture of the remote sensing image from Landsat 8.

Step 2. Select your data to download in the "Datasets" tab

Step 3. Filter your data in the "Additional Criteria" tab

Step 4. Download free Landsat imagery in the "Results" tab.

B. Architecture of Proposed Method

To solve the challenge of detecting small objects in remote sensing images, Gan et al. [23] proposed a method that employed a novel edge-enhanced super-resolution GAN (EESRGAN) to enhance the quality of remote sensing images. The method integrated various detector networks in an end-to-end approach. The detector loss was backpropagated into the EESRGAN to optimize detection performance. Furthermore, Zhao et al. [24] proposed a method consisting of two parts of architecture: a degraded reconstruction-assisted enhancement branch and a detection branch. Hereinafter, Chung, et al [25] proposed a method using a bicubic and generative adversarial network (BLG-GAN).

In this research, we propose a method consisting of two main parts: feature extraction and reconstruction. Both parts consist of deeply convolutional layers. The purpose of the feature extraction network is to extract the most relevant features of the image, while the reconstruction network aims to enhance image resolution through deconvolution. Overall, Figure 2 shows the framework of the proposed method, with details of the first and second parts shown in Figures 3 and 4, respectively.

1) Bicubic Interpolation

Bicubic interpolation is employed to enlarge an image by a specified scale factor prior to its processing by a CNN. For instance, a low-resolution image can be upsampled to a higher resolution using this method. This step provides CNN with a larger input image, allowing it to concentrate on enhancing the details and overall quality of the interpolated image.

In cases where a low-resolution image is directly input into the CNN without prior interpolation, the network may require additional layers or greater complexity to effectively learn from the data and produce a high-resolution output. Bicubic interpolation alleviates this challenge by offering an image with an initially higher resolution, thus enabling CNN to focus on refining quality aspects, such as texture details and object edges, rather than merely enlarging the image.

In summary, the function of bicubic interpolation is to furnish a larger image as a foundation, thereby allowing CNN to prioritize the improvement of image quality over simple image enlargement.

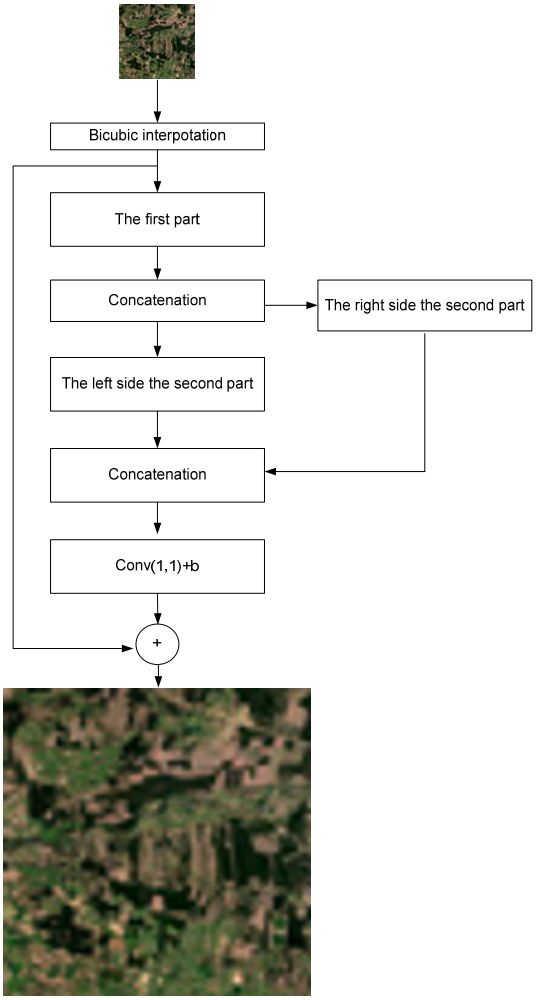

Figure 2. Framework of the enhancing remote sensing image resolution using CNN.

2) Feature Extraction Layers

The feature extraction network consists of 25 convolutional layers. Each layer employs a kernel size of 3×3, but the number of kernels per layer varies. Specifically, the first layer contains 139 kernels, and each subsequent layer decreases by 3 kernels. Table 1 shows the kernel and bias used on feature extraction layers, and Figure 3 shows the architecture of CNN in the first part of the proposed method.

3) Reconstruction Layer

In the reconstruction network, the feature maps generated in the first part are manipulated to enhance image resolution. See Figure 4, which comprises two convolutional neural network segments: the first segment (the left segment) contains a single convolutional layer, while the second segment (the right segment) consists of seventy-five convolutional layers. Additionally, the second segment concludes with a convolutional layer featuring a 1×1 kernel size. The architecture of the CNN in the second part of the proposed method is detailed in Figure 4. Here, OP-1 is output from the feature extraction layer. Table 2 shows the kernel and bias used in the feature extraction part.

TABLE 1
DETAILED CONVOLUTIONAL LAYER ON FEATURE EXTRACTION NETWORK

No. Layers	Size of Kernel	Number of Kernels	Number of Biases
1	*3×3*	139	139
2	*3×3*	136	136
3	*3×3*	133	133
4	*3×3*	130	130
5	*3×3*	127	127
6	*3×3*	124	124
7	*3×3*	121	121
8	*3×3*	118	118
9	*3×3*	115	115
10	*3×3*	112	112
11	*3×3*	109	109
12	*3×3*	106	106
13	*3×3*	103	103
14	*3×3*	100	100
15	*3×3*	97	97
16	*3×3*	94	94
17	*3×3*	91	91
18	*3×3*	88	88
19	*3×3*	85	85
20	*3×3*	82	82
21	*3×3*	79	79
22	*3×3*	76	76
23	*3×3*	73	73
24	*3×3*	70	70
25	*3×3*	67	67

TABLE 2
THE DETAILED KERNEL SIZE OF CONVOLUTIONAL LAYER ON RECONSTRUCTION NETWORKS

Layers	L1	R1	R2	…	R75	L2
Size of Kernel	1×1	1×1	3×3	…	3×3	1×1
Number of Kernels	32	32	32	…	32	1
Number of Biases	32	32	32	…	32	0

Furthermore, the detailed steps of the proposed method are outlined in Algorithm 1.

Algorithm 1:

Step 1: Input Image Enlargement: Enlarge the small input image using the bicubic interpolation method based on the desired scale.

Step 2: Perform feature extraction by running all convolution operations in the first part of the architecture.

Step 3: Combine all features generated by all channels through a concatenation operation to form a single image.

Step 4: (a) Run convolution operations on the left segment of the image in the second part of the architecture.
(b) Run convolution operations on the right segment of the image in the second part of the architecture.

Step 5: Combine the results of the left and right segment operations into a single image.

Step 6: Apply a 1x1 convolution to transform the combined output image from the second part of the architecture.

Step 7: Add the initial bicubic interpolation image to the transformed image from Step 6 to finalize the image reconstruction.

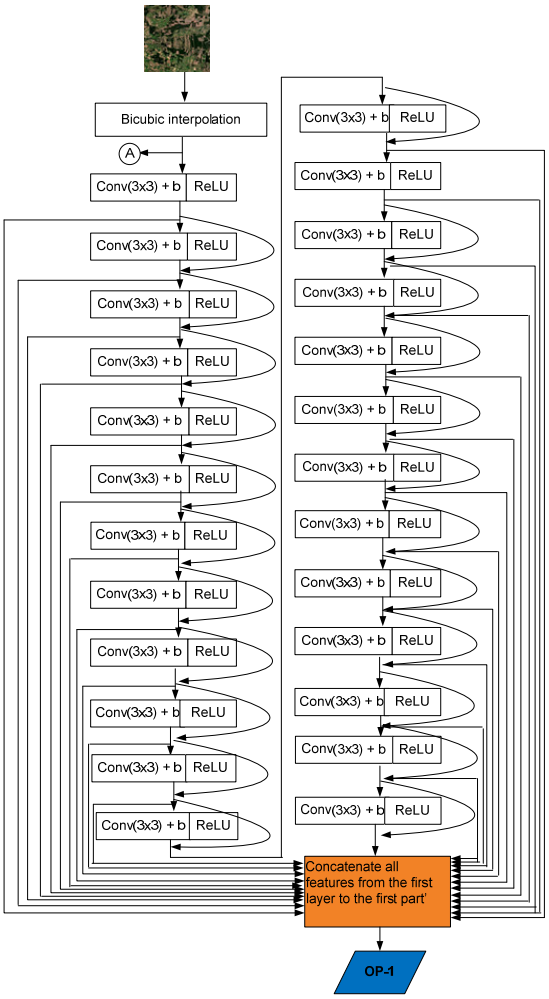

Figure 3. The architecture of CNN in the first part of the proposed method. OP-1 is output from the feature extraction layer.

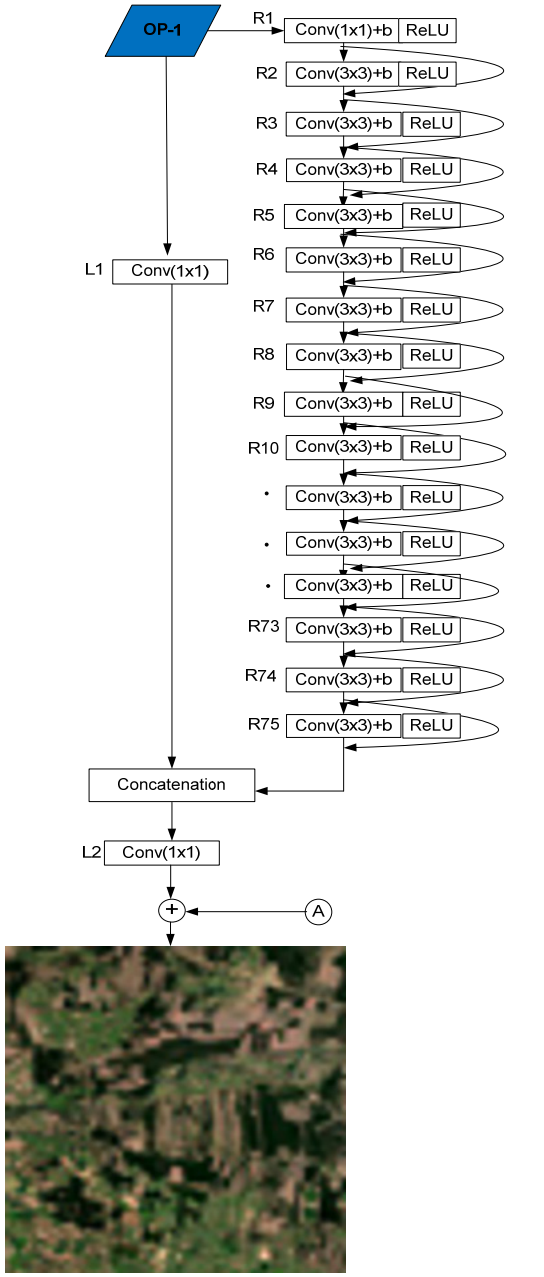

Figure 4. The architecture of CNN in the second part of the proposed method.

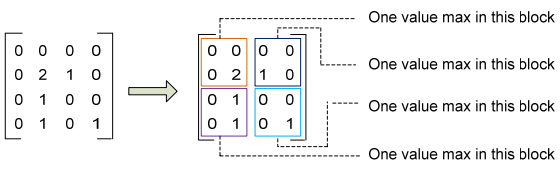

Figure 5. Illustration of max pooling.

4) Convolution Layer

Let I be the input of the convolution layer, K the kernel, and B the bias. The output of the convolution layer $l+1$ can be calculated using (1) and (2) [11],

$$Y_{r,s}^{(l)} = B^{(l)} + \sum_{u=-H_1}^{H_1} \sum_{v=-H_2}^{H_2} \sum_{d=0}^{D} K_{u,v}^{(l)} * I_{r+u,s+v}^{(l)} \quad (1)$$

$$I_{r,s}^{(l+1)} = \varphi(Y_{r,s}^{(l)}). \quad (2)$$

where H_1 and H_2 are the sizes of the kernel K, D is the number of kernels K, $r=0, 1,..., m$ and $s=0, 1,..., n$, and φ is the sigmoid function, defined as: $\varphi(x) = \frac{1}{1+e^{-x}}$.

5) Pooling Layer

A pooling layer (a subsampling layer) aims to reduce the feature resolution to make the features more resistant to noise and distortion. There are two primary methods of pooling: maximum pooling and average pooling. Both methods start by dividing the pixel matrix into several two-dimensional matrices (see Figure 5). Maximum pooling selects the highest value from each region, whereas average pooling computes the average value from each region [11].

6) Training Phase

Training is a very crucial stage in deep learning. The purpose of training is to determine the best model to solve the problem. Training calculations are carried out by minimizing the loss function. In this study, to minimize

the error in the training phase, we use the loss function L_2 as given by (3),

$$\xi = (\sum_{i=1}^{n} \|h(x) - t(x)\|)^2 \quad (3)$$

where ξ is the loss function, *h(x)* is the image output from the network, and *t(x)* is ground truth images.

Hereafter, to optimize the training phase, we employed the Adam Optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\in = 1e-8$. The optimizer and RMSprop momentum were both set to a value of 0.9. The learning rate started at 0.002 and increased to 0.005. The training process would terminate upon reaching the final learning rate. If the loss remained constant for 10 consecutive epochs, we reduced the learning rate by a factor of 2 until the final learning rate was achieved. We implemented a technique to create high resolution based on instructional techniques, as referenced in [26], [27]. This method aims to improve prediction accuracy [28]. Additionally, we applied the strategy proposed by Wang et al. in [29] to the self-ensemble. During this training phase, a cross-validation ensemble of five was utilized.

In addition, the calculation steps for each layer's feed-forward phase are derived from [30], and those for the feed-backward phase are derived from [31]. The weight update rule follows the classic backpropagation method [32] and employs the Adam Optimizer [33]. To mathematically update the weights *w* and bias *b* at time *t*, we use (4) and (5), respectively [11],

$$w(t+1) = w(t) - \alpha \frac{\hat{m}_{t_w}}{\sqrt{v_{t_w}} + \in}, \text{for } \in > 0 \quad (4)$$

$$b(t+1) = b(t) - \alpha \frac{\hat{m}_{t_b}}{\sqrt{v_{t_b}} + \in}, \text{for } \in > 0 \quad (5)$$

$$m_{t_w} = \beta_1 m_{t_w - 1} + (1 - \beta_1)\, g_{t_w}$$

$$m_{t_b} = \beta_1 m_{t_b - 1} + (1 - \beta_1)\, g_{t_b}$$

$$v_{t_w} = \beta_2 v_{t_w - 1} + (1 - \beta_2)\, g_{t_w}^2$$

$$v_{t_b} = \beta_2 v_{t_b - 1} + (1 - \beta_2) g_b^2$$

$$\hat{m}_{t_w} = \frac{m_{t_w}}{(1-\beta_1^t)};\ \hat{m}_{t_b} = \frac{m_{t_b}}{(1-\beta_1^t)}$$

$$\hat{v}_{t_w} = \frac{v_{t_w}}{(1-\beta_2^t)};\ \hat{v}_{t_b} = \frac{v_{t_b}}{(1-\beta_2^t)}$$

where m_{t_w} is the first moment of weight *w*, v_{t_w} is the second raw moment of weight *w*, m_{t_b} is the 1st moment of bias *b*, v_{t_b} is 2nd raw-moment of bias *b*, $\hat{m}_{t_w}$ is the weight-corrected 1st moment, $\hat{v}_{t_w}$ is the weight-corrected 2nd raw moment, $\hat{m}_{t_b}$ is the bias-corrected 1st moment, $\hat{v}_{t_b}$ is the bias-corrected 2nd raw-moment, α is learning rate, β_1 *and* β_2 are hyperparameters, $g_{t_w} = \frac{\partial \mathcal{L}}{\partial w}$ is the partial derivative of the loss function with respect to *w*, and $g_{t_b} = \frac{\partial \mathcal{L}}{\partial b}$ is the partial derivative of the loss function with respect to *b*.

7) Measurement and Validation

To measure the effectiveness of the proposed method, we employ standard metrics commonly used to assess the quality of transformed images, specifically the PSNR and SSIM [34]. PSNR compares the maximum signal level of the original image with the noise that appears after the transformation process (output image). Meanwhile, SSIM evaluates the structural and visual information between the output and original images. Mathematically, PSNR is calculated using (6), while SSIM is determined by (8).

$$PSNR = 20\, Log_{10}\left(\frac{Max_f}{\sqrt{MSE}}\right) \quad (6)$$

$$MSE = \frac{1}{mn} \sum_0^{m-1} \sum_0^{n-1} \|f(i,j) - g(i,j)\|^2 \quad (7)$$

Here, f denotes the pixel matrix of the original image, while g represents the pixel matrix of the resulting image. The variable m indicates the number of rows of pixels in the images, with i as the index of a specific row. Additionally, n signifies the number of columns of pixels in the image, and j represents the index of a specific column. Furthermore, Max_f represents the maximum signal value present in the original image.

$$SSIM_{(x,y)} = \frac{(2\mu_x\mu_y + C_1)(2\tau_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\tau_x^2 + \tau_y^2 + C_2)} \quad (8)$$

Where μ_x and μ_y are the average brightness of images *x* and *y*, τ_x^2 and τ_y^2 are the variants of image *x* and image *y* that is contrast, τ_{xy} covariance of image *x* and image *y* that is structure measure, and C_1 and C_2 are small constants to stabilization numerical.

III. Results and Discussion

A. Training Model

For the training phase, we utilized databases commonly used to train CNN models for generating high-resolution images, i.e. the database from Yang et al. [18] and the Berkeley Segmentation Database [19]. The Yang database consists of 96 nature images, while the Berkeley database (BSD200) contains 200 images. An illustration of some images from the BSD200 database [19] is shown in Figure 6. Furthermore, we initialized the weights using random numbers generated by a Gaussian distribution with a mean of zero and a standard deviation of 0.001, while setting the biases to zero for every part.

In addition, the configuration of our training is divided into multiple scaling factors: 2, 4, 8, and 16. Each scale factor defines the desired improvement in image resolution. For instance, if the input image resolution is

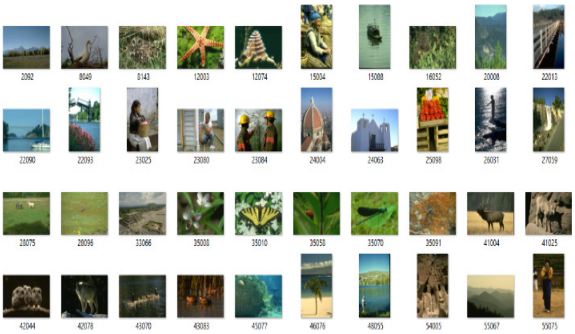

Figure 6. Sample images from Berkeley Segmentation [19].

to be increased by a factor of 2, the training scale factor is 2; if the resolution is to be increased by a factor of 4, the training scale factor is 4; and so on.

B. Testing Model

To verify the proposed method, we used a dataset originating from Landsat 8 imagery, which was downloaded from the Google Earth Engine platform (https://developers.google.com/earth-engine/datasets/catalog/landsat-8).

As we know, remote sensing images are taken from distant locations and cover large areas. For example, the Landsat 8 satellite has an imaging area of 185 km^2. Despite its wide coverage, the objects in the image are tiny. Enlarging the entire image directly is not the best solution, as it requires large resources and high computational complexity.

To overcome this problem, this research applies a partition technique based on area. In this case, we experimented with three different partition sizes: 16×16 pixels, 24×24 pixels, and 32×32 pixels. Next, each partition is increased to 128×128 pixels. An illustration of the image partitioning process is presented in Figure 7.

In this experiment, we compare the result from our proposed method with previous methods widely used to improve the quality of low-resolution images, i.e. Bicubic, SRCNN [35], SRCNN-IBP [36], DRL [37], DCSCN [38]. These five methods are considered very good and are commonly used in the wider world. The results of the comparison obtained can be seen visually in Figure 8, while mathematically, the comparison of PSNR and SSIM from each method is presented in Table 3.

Figure 7. The segmentation of image for input CNN process.

As illustrated in Figure 8, the images generated by our method exhibit superior visual quality compared to those produced by previously established methods. This observation is further substantiated by the quantitative results, specifically the SSIM and PSNR values presented in Table 3. The SSIM and PSNR values for all partition sizes (i.e. 16×16 pixels, 24×24 pixels, and 32×32 pixels) indicate that the quality of the images produced by the proposed method is consistently higher compared to

TABLE 3
THE PSNR AND SSIM COMPARISON OF THE OUTPUT OF SOME STATE OF THE ART

Methods	PSNR/SSIM		
	16×16 pixels	**24×24 pixels**	**32×32 pixels**
Bicubic	26.45/0.520	27.16/0.721	29.75/0.831
SRCNN [35]	26.74/0.632	27.66/0.722	30.84/0.856
SRCNN-IBP [36]	27.78/0.641	28.87/0.746	30.90/0.859
DRL[37]	28.77/0.779	29.83/0.841	30.38/0.896
DCSCN [38]	28.66/0.790	29.88/0.861	32.93/0.910
Our Method	**28.94/0.822**	**30.24/0.089**	**33.24/0.925**

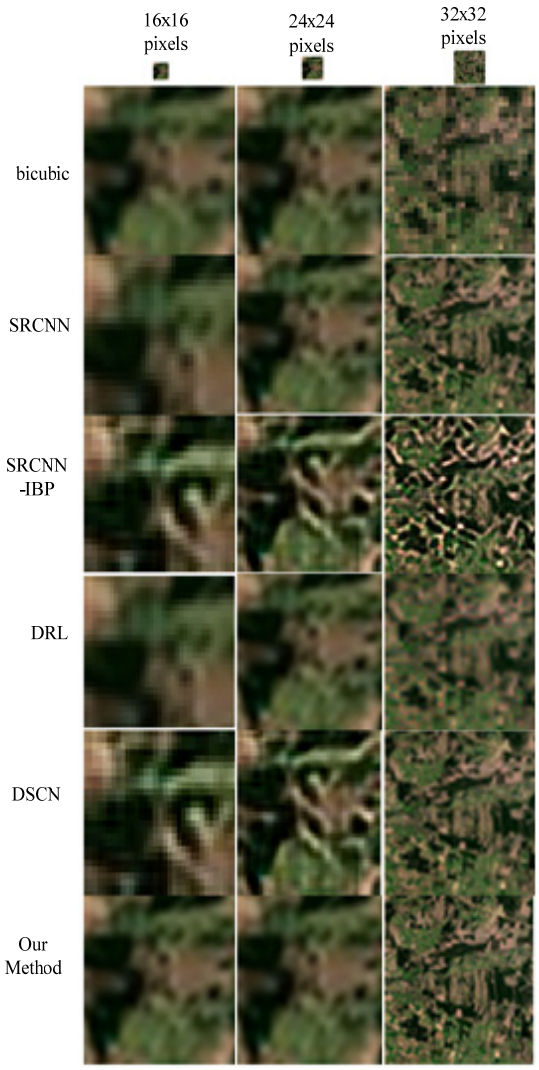

Figure 8. The comparison of some output from our proposed method with the existing methods for segment size areas 16×16 pixels, 24×24 pixels, and 32×32 pixels.

existing methods. Accordingly, it can be concluded that the proposed method outperforms existing approaches.

IV. CONCLUSION

This research has successfully developed an architecture for convolutional neural networks (CNNs) to enhance the quality of remote-sensing images. The architecture, classified as a deep-CNN model, incorporates over 75 convolutional layers. Moreover, the proposed method outperforms existing methods based on peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM).

DECLARATIONS

Conflict of Interest

The authors have declared that there is no conflict of interest in this publication and research.

CRediT Authorship Contribution

Julian Supardi: Conceptualization, Methodology, Software, Visualization, Investigation, Writing-Original draft, Writing-Reviewing and Editing; Samsuryadi: Data curation, Writing-Reviewing and Editing; Hadipurnawan Satria: Software; Philip Alger M. Serrano: Writing-Reviewing and Editing; Arnelawati: Data curation.

Funding

This research supports funding from a research grant from Sriwijaya University by contract number: 0098.137/UN9/SB3.LP2M.PT/2024.

REFERENCES

[1] E. Turan and G. Çetin, "Using artificial intelligence for modeling of the realistic animal behaviors in a virtual island," *Comput. Standards Interfaces*, vol. 66, 2019, Art. no. 103361, doi: 10.1016/j.csi.2019.103361.

[2] Y. Xu et al., "Artificial intelligence: A powerful paradigm for scientific research," *Innovation*, vol. 2, no. 4, 2021, Art. no. 100179, doi: 10.1016/j.xinn.2021.100179.

[3] Y. Sun, W. Dong, and Y. Chen, "An improved routing algorithm based on ant colony optimization in wireless sensor networks," *IEEE Commun. Lett.*, vol. 21, no. 6, pp. 1317–1320, 2017, doi: 10.1109/LCOMM.2017.2672959.

[4] H. Liu, C. Li, S. He, W. Shi, Y. Chen, and W. Shi, "Simulated annealing particle swarm optimization for a dual-input broadband GaN Doherty like load-modulated balance amplifier design," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 69, no. 9, pp. 3734–3738, 2022, doi: 10.1109/TCSII.2022.3173608.

[5] A. Awad, A. Hawash, and B. Abdalhaq, "A genetic algorithm (GA) and swarm-based binary decision diagram (BDD) reordering optimizer reinforced with recent operators," *IEEE Trans. Evol. Comput.*, vol. 27, no. 3, pp. 535–549, 2023, doi: 10.1109/TEVC.2022.3170212.

[6] M. Xu, Y. Zhang, Y. Fan, Y. Chen, and D. Song, "Linear spectral mixing model-guided artificial bee colony method for endmember generation," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 12, pp. 2145–2149, 2020, doi: 10.1109/LGRS.2019.2961502.

[7] J. M. Mendel and D. Wu, "Critique of "a new look at type-2 fuzzy sets and type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 3, pp. 725–727, 2017, doi: 10.1109/TFUZZ.2017.2648882.

[8] Y. Chen, Q. Mao, B. Wang, P. Duan, B. Zhang, and Z. Hong, "Privacy-preserving multi-class support vector machine model on medical diagnosis," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 7, pp. 3342–3353, 2022, doi: 10.1109/JBHI.2022.3157592.

[9] D. B. Strydom, "Industrial application of a real- time expert system," *Trans. South African Inst. Elect. Eng.,* vol. 81, no. 2, pp. 1–6, 1990.

[10] J. Supardi and S.-J. Horng, "Very small image face recognition using deep convolution neural networks," *J. Phys.: Conf. Ser*, vol. 1196, 2019, Art. no. 012020, doi: 10.1088/1742-6596/1196/1/012020.

[11] S. J. Horng, J. Supardi, W. Zhou, C. T. Lin, and B. Jiang, "Recognizing very small face images using convolution neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2103–2115, 2022, doi: 10.1109/TITS.2020.3032396.

[12] M. M. Taye, "Understanding of machine learning with deep learning: architectures, workflow, applications and future directions," *Comput.,* vol.12, no. 5, 2023, Art. no. 91, doi: 10.3390/computers12050091.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 1097–1105, 2017, doi: 10.1145/3065386.

[14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. 2014 European Conf. Comput. Vision*, *Lecture Notes in Computer Science*, vol. 8689, pp. 818–833, 2014, doi: 10.1007/978-3-319-10590-1_53.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc.* 2015 *IEEE Conf. Computer Vision and Pattern Recognition,* 2015, doi: 10.1109/CVPR.2015.7298594.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition,* 2016, doi: 10.1109/CVPR.2016.90.

[17] J. Zhang, Z. Chen, G. Yan, Y. Wang, and B. Hu, "Faster and lightweight: An improved YOLOv5 object detector for remote sensing images," *Remote Sensing*, vol.15, no. 20, 2023, Art. no. 4974, doi: 10.3390/rs15204974.

[18] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, 2010, doi: 10.1109/TIP.2010.2050625.

[19] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011, doi: 10.1109/TPAMI.2010.161.

[20] W. Fan and D.-Y. Yeung, "Image hallucination using neighbor embedding over visual primitive manifolds," in *Proc. 2007 IEEE Conf. Comput. Vis. Pattern Recognit.,* MN, USA, 2007, pp. 1-7, doi: https://doi.org/10.1109/CVPR.2007.383001.

[21] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1637–1645, doi: 10.1109/CVPR.2016.181.

[22] J. Sun, N.-N. Zheng, H. Tao, and H.-Y. Shum, "Image hallucination with primal sketch priors," in *2003 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Proc.*, Madison, WI, USA, 2003, doi: https://doi.org/10.1109/CVPR.2003.1211539.

[23] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury and D. Chao, "Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network," *Remote Sensing*, vol. 12, no. 9, 2020, Art. no. 1432, doi: 10.3390/rs12091432.

[24] Y. Zhao, H. Sun, and S. Wang, "Small object detection in medium–low-resolution remote sensing images based on degradation reconstruction," *Remote Sensing*, vol. 16, no. 14, 2024, Art. no. 2645, doi: 10.3390/rs16142645.

[25] M. Chung, M. Jung, and Y. Kim, "Enhancing remote sensing image super-resolution guided by bicubic-downsampled low-resolution image," *Remote Sensing*, vol. 15, no. 13, 2023, Art. no. 3309, doi: 10.3390/rs15133309.

[26] C. Li, D. He, X. Liu, Y. Ding, and S. Wen, "Adapting image super-resolution state-of-the-arts and learning multi-model ensemble for video super-resolution," in *Proc. 2019 IEEE/CVF Conf. Comput. Vis. and Pattern Recognit. Workshops*, Long Beach, CA, USA, 2019, pp. 2033-2040, doi: 10.1109/CVPRW.2019.00255.

[27] R. Timofte, R. Rothe, and L.Van Gool, "Seven ways to improve example-based single image super resolution," in *Proc. 2016 IEEE Conf. Comput. Vision and Pattern Recognit.,* Las Vegas, NV, USA, 2016, pp. 1865-1873, doi: 10.1109/CVPR.2016.206.

[28] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Multiple Classifier Systems*. *MCS 2000. Lecture Notes in Computer Science*, vol 1857, Springer, Berlin, Heidelberg, 2000, doi: 10.1007/3-540-45014-9_1.

[29] L. Wang, Z. Huang, Y. Gong, and C. Pan, "Ensemble based deep networks for image super-resolution," *Pattern Recognit.*, vol. 68, pp. 191–198, 2017, doi: 10.1016/j.patcog.2017.02.027.

[30] J. Wu, "Introduction to convolutional neural networks," National Key Lab for Novel Software Technology, Nanjing University, China, 2017. [Online]. Available: https://pdfs.semanticscholar.org/450c/a19932fcef1ca6d0442cbf52fec38fb9d1e5.pdf (accessed Dec. 10, 2019).

[31] Z. Zhang, "Derivation of backpropagation in convolutional neural network (CNN)," University of Tennessee, Knoxvill, TN, 2016. [Online]. Available: https://github.com/ZZUTK/An-Example-of-CNN-on-MNIST-dataset/blob/master/doc/Derivation_of_Backpropagation_in_CNN.pdf (accessed Dec. 12, 2024).

[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. the IEEE*, vol. 86, no. 11, 1998, pp. 2278–2324, doi: 10.1109/5.726791.

[33] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. for Learning Representations,* San Diego*,* 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[34] K.-H. Thung and P. Raveendran, "A survey of image quality measures," in *Proc. 2009 Int. Conf. Tech. Postgraduates*, Kuala Lumpur, Malaysia, 2009, doi: 10.1109/TECHPOS.2009.5412098.

[35] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Analysis Mach. Intell.,* vol. 38, no. 2, pp. 295–307, 2015, doi: 10.1109/TPAMI.2015.2439281.

[36] D. Huang and H. Liu, "Face hallucination using convolutional neural network with iterative back projection," in *Proc. 11th Chinese Conf. Biometric Recognition, Lecture Notes in Computer Science*, vol. 9967, 2016, pp. 167–175, doi: 10.1007/978-3-319-46654-5_19.

[37] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li, "Attention-aware face hallucination via deep reinforcement learning," in *Proc. 2017 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1656–1664, doi: 10.1109/CVPR.2017.180.

[38] J. Yamanaka, S. Kuwashima, and T. Kurita, "Fast and accurate image super resolution by deep CNN with skip connection and network in network," in *Proc. 24th Int. Conf. Neural Information Processing, Lecture Notes in Computer Science,* vol. 10635, 2017, pp. 217-225, doi: https://doi.org/10.1007/978-3-319-70096-0_23.