scientific reports

Check for updates

OPEN

Comparative performance assessment of deep learning based image steganography techniques

Varsha Himthani[1], Vijaypal Singh Dhaka[1], Manjit Kaur[2], Geeta Rani[1], Meet Oza[1] & Heung-No Lee[2]✉

Increasing data infringement while transmission and storage have become an apprehension for the data owners. Even the digital images transmitted over the network or stored at servers are prone to unauthorized access. However, several image steganography techniques were proposed in the literature for hiding a secret image by embedding it into cover media. But the low embedding capacity and poor reconstruction quality of images are significant limitations of these techniques. To overcome these limitations, deep learning-based image steganography techniques are proposed in the literature. Convolutional neural network (CNN) based U-Net encoder has gained significant research attention in the literature. However, its performance efficacy as compared to other CNN based encoders like V-Net and U-Net++ is not implemented for image steganography. In this paper, V-Net and U-Net++ encoders are implemented for image steganography. A comparative performance assessment of U-Net, V-Net, and U-Net++ architectures are carried out. These architectures are employed to hide the secret image into the cover image. Further, a unique, robust, and standard decoder for all architectures is designed to extract the secret image from the cover image. Based on the experimental results, it is identified that U-Net architecture outperforms the other two architectures as it reports high embedding capacity and provides better quality stego and reconstructed secret images.

Image encryption and image steganography are the most common ways to secure image data. In image encryption, the image is encoded using an encryption technique[1]. In image steganography, an image is embedded in some cover media such as image, audio, video, etc.[2]. The advantage of image steganography is that it is hard to distinguish that a secret image is hidden into the cover media[3]. Whereas in image encryption, encrypted images are noise-like that may attract an attacker.

The traditional steganography technique 'Least Significant Bit' (LSB) substitutes the secret data bits on LSBs of image pixel values[2,3]. But it leaves traces of hidden data that can be detected by steganalysis[4,5]. To enhance security, many improvements to the LSB technique were proposed[5–8]. However, there is no significant improvement observed in the essential properties of steganography in these techniques[5–8]. To provide better results than LSB, transform domain-based steganography techniques were proposed[9–13]. In these techniques, the secret data is embedded into coefficient values. However, these techniques suffer from low payload capacity and poor visual quality of stego and reconstructed images[9,10].

To improve the weak aspects of the above-discussed methods, machine learning-based steganography methods such as the Genetic algorithm[14–16] and fuzzy logic-based[17,18] were proposed. These techniques have significantly improved the visual quality of the stego and reconstructed image, but the flaws like high complexity and low payload capacity are not elucidated. To improve security, support vector machine-based steganography techniques[19,20] is proposed, but these techniques are not suitable for large datasets.

In recent years, image steganography based on convolutional neural network (CNN) has gained wide research attention due to its superior capabilities against traditional methods[21]. In these methods, the secret image is embedded into cover media by intelligent and accurate coefficient selection. It enhances the performance of steganography in all the aspects like payload capacity, imperceptibility, and reconstructed image visual quality, etc.[22].

A CNN-based image steganography technique proposed by Rehman et al. improved the visual quality of the stego image by hiding the gray-scale secret image in specific extracted features of the color cover image[23].

[1]Department of Computer and Communication Engineering, Manipal University Jaipur, Jaipur 303007, India. [2]School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, Korea. ✉email: Heungno@gist.ac.kr

References	Steganography method	Improved	Needs to be improved
[2,3,5–8]	Traditional LSB based	Easy implementation	Security, payload capacity, visual quality of stego image and recovered image
[9–13]	Transform domain based	Better security and payload capacity than traditional LSB	Visual quality of stego and reconstructed images
[14–18]	Machine learning based	Better visual quality of stego and reconstructed images	High complexity, payload capacity can be improved
[19,20]	Support vector machine based	Better security	Not suitable for large dataset
[23–29]	CNN based	High payload capacity, reconstruction quality	Computational cost, security from deep learning based steganalysis
[30–33]	GAN based	High visual quality stego and reconstructed images, low computation cost	Security from deep learning based steganalysis

Table 1. Summary of the existing literature.

Further, Baluja proposed an autoencoder and decoder scheme[24]. In this, three networks are prepared, first is the preparation network that transforms the RGB pixels of the secret image into features. The second is a hiding network that hides the features obtained by the preparation network into the cover image. The third is the reveal network, which extracts the secret image from the cover image. Here, the payload capacity and stego image visual quality are improved, but the visual quality of the reconstructed image is significantly compromised. Duan et al. increased the payload capacity by embedding two secret images in one cover image[25].

Further, Zhang et al. proposed the improvement in the stego image visual quality by converting the cover image in YCrCb format[26]. Only the 'Y' channel is used to hide the secret grayscale image without affecting the 'Cr' and 'Cb' channels. These two channels contain all the color information. Hence, the stego image quality is improved, but this method is limited to a secret grayscale image. The U-net architecture-based steganography is proposed to improve the payload capacity and reconstructed image quality[27,28]. Wu et al. proposed CNN-based steganography that enhanced the payload capacity and stego image quality[29]. For further improvements, steganography techniques based on the generative adversarial networks are proposed[30–33]. These networks generate high-quality stego and reconstructed images at a low computation cost. However, the security of these methods also needs improvement. Table 1 provides the summary of the existing literature discussed above.

In recent literature, Sharma et al. proposed an image steganography technique based on graph signal processing. In this method, the secret image is first scrambled through quantum scrambling to enhance the security. Then, both the cover image and secret image are transformed by graph wavelet transform that improved the visual quality of the stego image and recovered the secret image[34]. Shen et al. presented an image steganography technique for the applications based on wireless visual sensor networks by using partial preservation embedding algorithm[35]. Telli et al. proposed a multi-image steganography technique inspired by Baluja's scheme[24] and improved stego image visibility[36]. Peter et al. improved the payload capacity of the steganography by using the histogram shifting method and quick response decomposition method[37].

It is evident from the literature that CNN-based steganography has the potential of securing image data by hiding the secret image into a cover image. But, each CNN-based architecture requires its unique corresponding decoder to decode the secret image at the receiver end. Furthermore, there is enormous scope to enhance the quality of reconstructed images, improve the payload capacity and reduce the computation time.

In this work, performance analysis of CNN based on three deep learning architectures *i.e.*, U-Net, V-Net, and U-Net++ for steganography is carried out. U-Net architecture for image steganography is analyzed in the literature[27]. However, its efficacy is not envisaged with similar CNN based techniques like V-Net and U-Net++ architectures. In this paper, V-Net and U-Net++ architectures are first time implemented for image steganography and their performance compared to U-Net is analyzed.

The main contribution of this paper could be summarized as follows:

- A comparative assessment is carried out between U-Net, V-Net, and U-Net++ architecture-based steganography, and various performance parameters are evaluated.
- Three image-in-image steganographic techniques based on U-Net, V-Net, and U-Net++ are proposed for confidential communication and storage of data.
- Developed a deep learning-based decoder that can decode the stego images generated by either of the three proposed encoders.

The proposed architectures hide a secret image of dimension $N \times N$ into a cover image of the same dimensions. In contrast to the methods[23,24,26], this research employs U-Net, V-Net, and U-Net++ architectures as encoders to hide the secret image into the cover image and a common decoder architecture to extract secret image from the stego image generated by either of the used encoder architectures.

Best of the authors' knowledge, implementation of V-Net and U-Net++ architectures for image steganography and comparative performance assessment of U-Net, V-Net, and U-Net++ encoders are not reported in the literature and can be considered as the unique contribution of the proposed work.

In the remaining part of the paper, "Background" section describes the background of the CNN architectures; "Proposed methodology" section demonstrates the proposed methodology. "Evaluation metrics" section gives the details of the experimental details. "Results and discussion" section discusses the implications of the

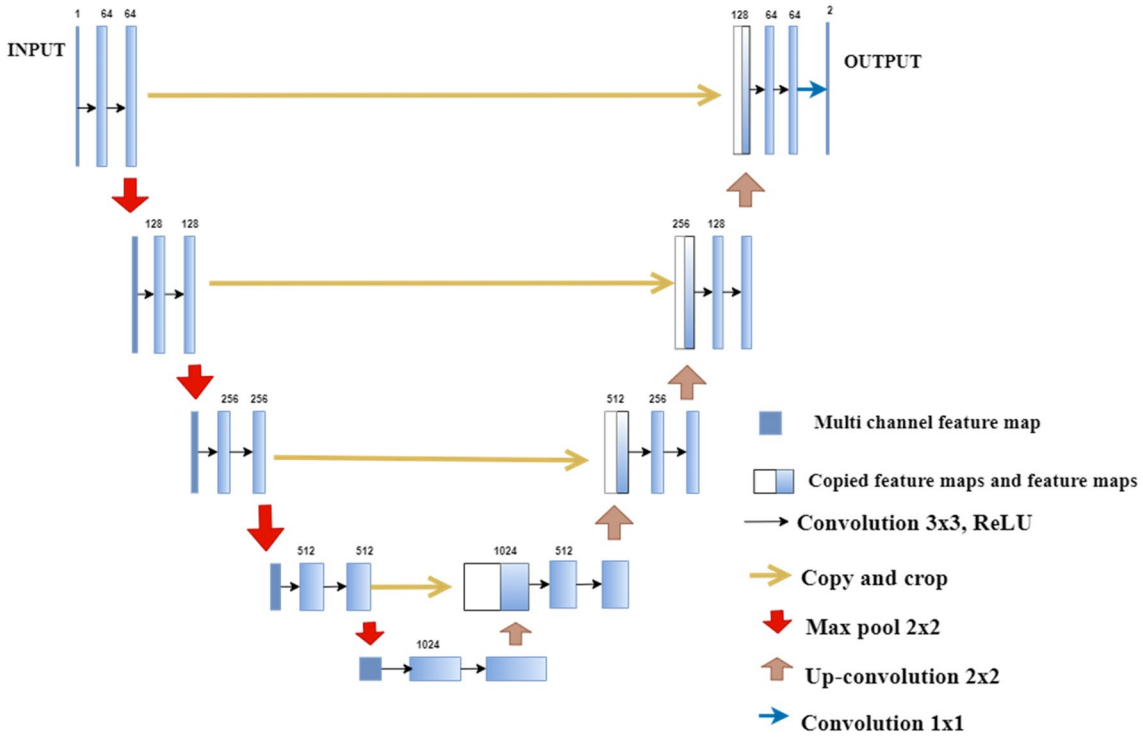

Figure 1. U-net architecture.

proposed work and its performance supremacy over the related works. Finally, "Conclusions" section presents the conclusions of the proposed work.

Background

CNN architectures gained popularity due to automatic feature extraction, reduced feature map, high accuracy, and versatile application areas[38]. The potential of CNN architectures is proved in the applications of pattern recognition[39], classification[40], object recognition[41], and image segmentation[42,43]. In recent literature, the applications of these networks are also observed in steganography[9]. U-Net, V-Net, and U-Net++ are widely known architectures used for image segmentation. In this paper, the applications of these architectures are extended in image steganography.

U-Net. U-Net is a fully convoluted neural network that provides enhanced performance with fewer training images[42]. Figure 1 shows the example of U-Net architecture[42]. The blue-colored boxes represent the multi-channel feature maps. This architecture consists of contraction (convolution) and expansion (deconvolution) paths. Each path comprises 23 convolutional layers. Each layer of the contraction path contains two filters of dimensions 3×3 that repeatedly perform unpadded convolutions. The feature channels are doubled in each convolution layer.

Further, each convolution operation is followed by the 2×2 max-pooling operation. The expansion path of the encoder performs deconvolution operations by using the filter size of 2×2. The feature channels are halved in each deconvolution layer. Thus, horizontal connections from the left to the right path to forward extracted features at the early stages. This improves the quality of the final reconstructed image by providing spatial information lost during contractions[42].

V-Net. The V-Net convolutional network is specially designed to take volumetric inputs. This architecture is similar to U-Net based architecture except that the contraction path has 1–3 convolution layers in each stage and substitutes max pooling operations with the convolution operations. Figure 2 represents the example of V-Net architecture. The network is divided into phases to work with volumetric inputs in the contraction path, and extracted features are expanded in the expansion path. In contrast to U-Net, the residual function is learned at each stage of contraction and expansion path that ensures convergence of the architecture[43].

U-Net++. The U-Net++ architecture is the tailored version of the U-Net. In this architecture, convolution layers are on skip pathways that tie the semantic gap among encoder and decoder feature maps. The count of convolution layers is dependent on the skip pathways. The number of skip pathways is calculated by using Eq. (1)[44]. These skip pathways connect the two sub-networks for deep supervision. A concatenation layer follows each convolution layer in the dense convolution block. The concatenation layer provides the output obtained by the fusion of the current and its previous convolution layers' outputs.

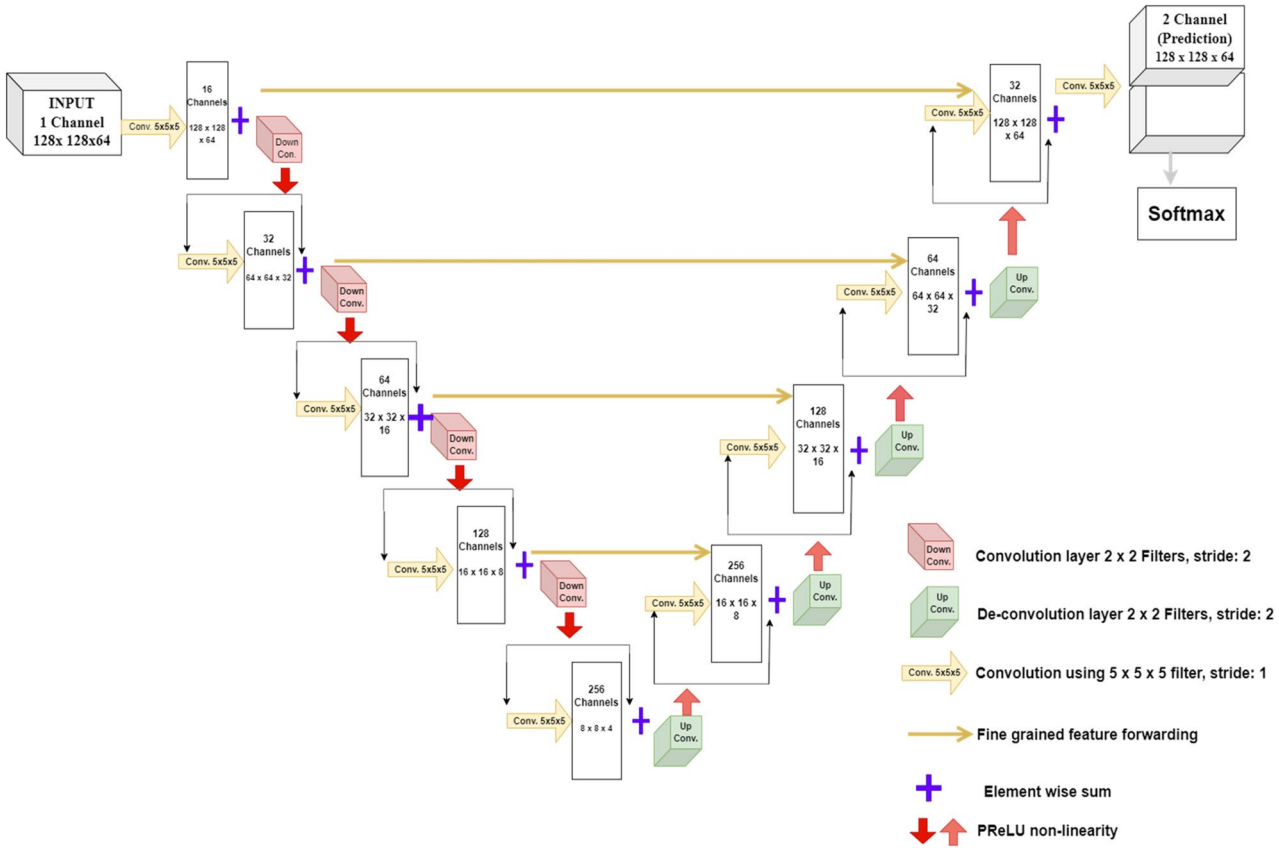

Figure 2. V-net architecture.

$$S^{i,j} = \begin{cases} Con\left(S^{i-1,j}\right), & j = 0 \\ Con([\left[S^{i,k}\right]_{k=0}^{j-1}, De(S^{i+1,j-1})]), & j > 0 \end{cases} \quad (1)$$

In Eq. (1), the function *con* is the convolution operation, function *De* is the deconvolution operation. Here, $S^{i,j}$ denotes the stack of feature map of the node $s^{i,j}$, i is the index of the contraction layer, j, and k are the indices of the convolution layers of the dense block. The dense skip connections on skip pathways enhances the gradient flow.

Also, the U-Net++ allows flexible network depth and is free from unnecessary limiting skip connections. Here, the merging of same-scale feature maps is considered. Further, the U-Net++ architecture allows compact feature proliferation through the compactly associated skip connections. Therefore, at the decoder nodes, more flexible feature fusion is obtained. The multiscale feature aggregation leads to deep supervision, high accuracy, and fast convergence. Figure 3 illustrates the example of U-Net++ architecture[44]. The red lines show the original U-Net architecture. In U-Net++ convolution layers are on skip pathways, that draw the semantic gap between encoder and decoder feature maps. The blue and green lines represent the dense skip connections on skip pathways.

Proposed methodology

In this work, deep learning-based image-in-image steganography techniques are implemented and their performance is assessed. The architecture shown in Fig. 4 demonstrates three deep learning architectures, viz. U-Net, V-Net, and U-Net++ based encoders that are employed to hide secret image into the cover image.

A unique decoder architecture is designed to extract hidden secret image from the stego image. The architectural details of the encoders and decoder are illustrated in the subsequent subsections 3.1 and 3.2, respectively.

Architecture of encoders. In the proposed steganography techniques, three fully connected distinct CNN architectures viz. U-Net[42], V-Net[43], and U-Net++[44] are implemented to generate a stego image that hides the secret image into the cover image.

Architecture of decoder. As a part of this research, a CNN-based unique and robust decoder is designed. The purpose of the decoder is to extract secret images from the stego images generated by any of the U-Net, V-Net, and U-Net++ based encoders.

As shown in Fig. 5, the decoder contains 11 convolution layers with different kernel sizes of 3×3, 4×4, and 5×5. Each kernel has multiple filters for enhancing the feature extraction capabilities of the decoder network. The

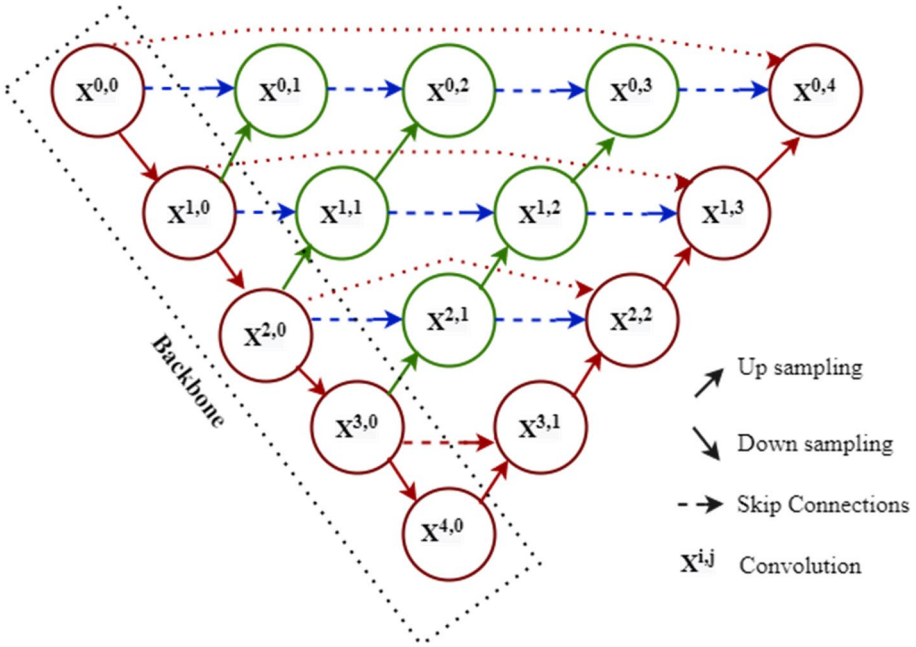

Figure 3. U-Net++ architecture.

convolution (CL) layers provide the feature maps as their outputs. The concatenation (CAT) of convolutional layers is performed to extract essential and useful semantic features from the feature maps[45]. These features improve the learning of the model. The details of the output size and number of parameters at each layer are demonstrated in Table 2. In this, the first column gives the details of each layer of the network where *conv* is the convolution, and *level* shows the level of the convolution layer. The second column shows the output size obtained corresponding to each input layer. The last column indicates the number of parameters at each layer of the network.

Training. In this research, the training and testing of the proposed architecture are carried out on a machine with RTX 2,080 Graphics Processing Unit (GPU) with 96 GB RAM and a 2 TB hard disk. The GPU runs with Ubuntu 16.04 operating system.

Dataset preparation. The dataset used for training the encoders and decoder is available online[46,47]. The dataset comprises 6616 color images with 3 channels and dimensions of 256×256. To ensure robustness, the model is trained on the imagery dataset of different types.

The test dataset[46,47] comprising 250 images of various kinds is used to evaluate the model performance. While training, each of the U-Net, V-Net, and U-Net++ based encoders individually take a cover and a secret image as inputs and provides a corresponding stego image as an output. The decoder network is trained simultaneously to extract the secret image from the stego image. Equation (2) shows the convolution operation performed at each convolution layer of the decoder.

$$Y_L = \sum_{x=0}^{N-1} S_x K_{L-x} \tag{2}$$

where, Y_L denotes the output of each convolution operation, S is stego image data bits, K is the kernel size, and N is the number of elements in S. Multiple convolution operations are performed at each convolution layer to give a feature map (f_L) as output. Now, f_L of all the n convolution layers $L(L = 1, \ldots n)$ are concatenated as given in Eq. (3).

$$CAT(i) = \frac{1}{w_o \times h_o} \sum f_L(.,.,i), \mathrm{i} = 1, 2, \ldots \mathrm{c_o} \tag{3}$$

Here, c_o denotes the number of channels in layer L, w_o and h_o are the width and height of the channel, respectively. Each convolutional layer is employed with Rectified Linear Unit (ReLU) activation function[48]. This function returns '0' for the negative input and the same value 'v' for any positive input value 'v'[44]. It is defined in Eq. (4).

$$\mathrm{ReLU\ (v)} = \mathrm{Maximum(0,v)} \tag{4}$$

Further, the Adam optimizer is employed for providing computationally fast and efficient learning. This optimizer reduces the memory requirements in comparison to the classical stochastic gradient descent approach[49]. To carry out the training, the batch size of 32, 16, and 8 images are selected for the U-Net, V-Net, and U-Net++

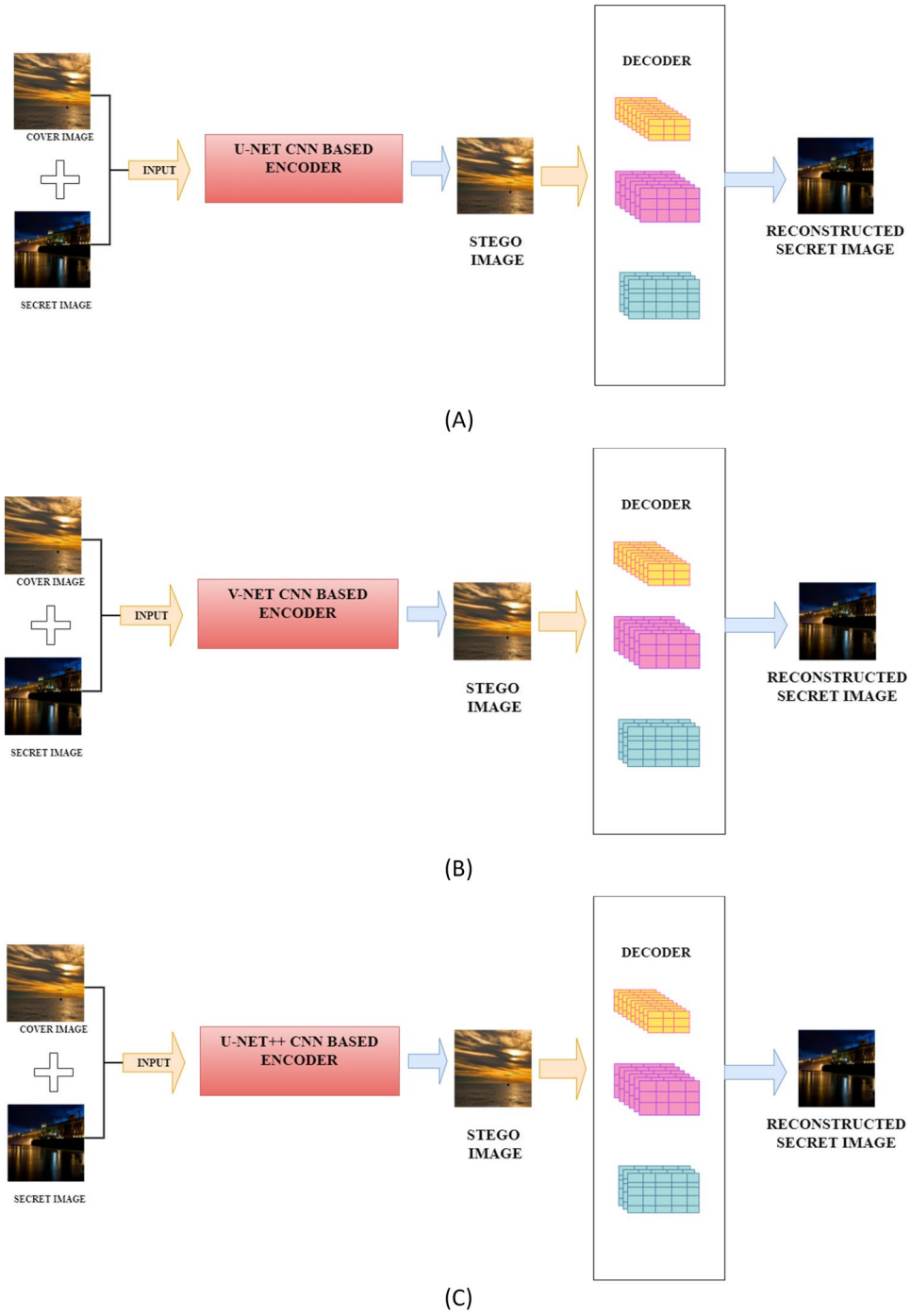

Figure 4. Block diagram of the proposed steganography techniques (**A**) U-Net architecture based encoder; (**B**) V-Net architecture based encoder; (**C**) U-Net++ architecture based encoder.

based encoders, respectively. The batch size is selected based on the CNN models to maximize GPU utilization with minimum overhead.

The training parameters such as the exponential decay of the first and second moments of the gradients are set to 0.9 and 0.999, respectively. The value of another training parameter, 'epsilon,' is set to 1e-07[49]. In the contrast, the learning rate (alpha) is set to 0.0001, which is smaller than the value (0.001) used in the reference[49]. The value of the learning rate is decided based on the experiments conducted in this research. By varying the learning rate from 1 to 0.0001, it is observed that the model reported the minimum value of loss function and highest accuracy at the 0.0001 value of the learning rate. Further, it is witnessed that there is a slight decrease in the accuracy when the learning rate is increased from 0.0001 to 0.0005, but a sharp decline is observed on further increasing its value. The loss functions defined in Eqs. (5) and (6) are employed to train the encoder and

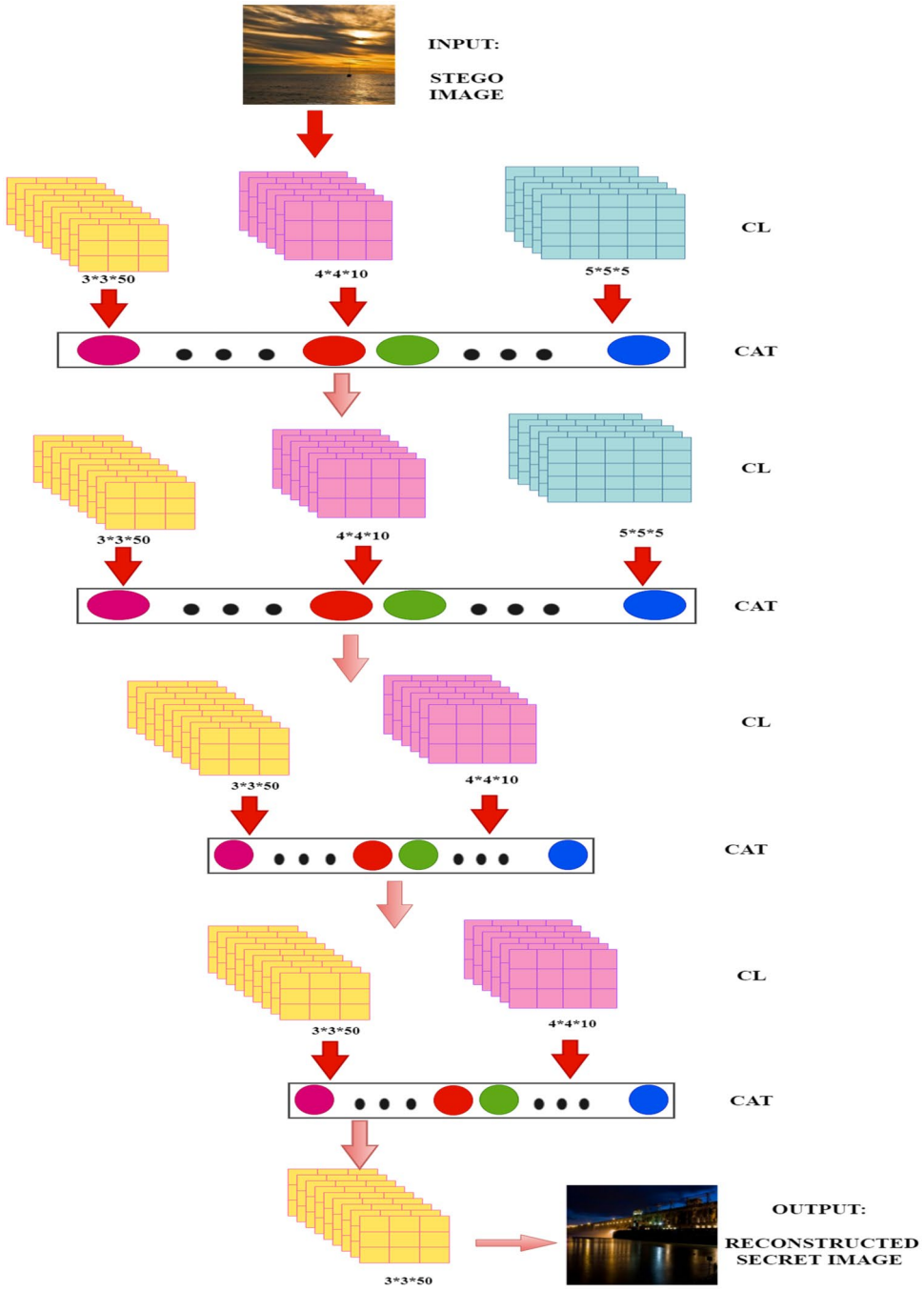

Figure 5. The architecture of the decoder network.

decoder networks. This loss function is effective in reducing the training error in the process of embedding the secret image into the cover image.

In Eq. (5), c, c' are the cover, and stego images, respectively. In Eq. (6), s, and s' are the secret, and reconstructed secret images, respectively. Here, E is the image reconstruction error as defined in reference[31].

$$lossencoder = ||c - c'|| \tag{5}$$

$$lossdecoder = E||s - s'|| \tag{6}$$

At each epoch of training, the value of the loss function is computed. The weights of the network dynamically change until the value of the loss function approaches a minimum value. This ensures that the network is trained to generate a high-quality stego image. The stego image contains a secret image hidden into it. Still, it appears

Layer (type)	Output size	Parameters
Input Layer	256, 256 ,3	0
conv_level0_3 × 3 (Conv2D)	256, 256, 50	1400
conv_level0_4 × 4 (Conv2D)	256, 256, 10	490
conv_level0_5 × 5 (Conv2D)	256, 256, 5	380
Concatenate_conv_level0	256, 256, 65	0
conv_level1_3 × 3 (Conv2D)	256, 256, 50	29,300
conv_level1_4 × 4 (Conv2D)	256, 256, 10	10,410
conv_level1_5 × 5 (Conv2D)	256, 256, 5	8130
Concatenate_conv_level1	256, 256, 65	0
conv_level2_3 × 3 (Conv2D)	256, 256, 50	29,300
conv_level2_4 × 4 (Conv2D)	256, 256, 10	10,410
Concatenate_conv_level2	256, 256, 60	0
conv_level3_3 × 3 (Conv2D)	256, 256, 50	27,050
conv_level3_4 × 4 (Conv2D)	256, 256, 10	9610
Concatenate_conv_level3	256, 256, 60	0
conv_level4_3 × 3 (Conv2D)	256, 256, 50	27,050
Output Layer	256, 256, 3	1353

Table 2. Structure of the decoder network .

indistinguishable from the cover image. Simultaneously, the decoder is trained to extract the secret image from the stego image generated by any of the encoders. The loss function for the decoder is computed as the difference in the secret image and its corresponding reconstructed secret image.

The training procedure of the encoder and decoder network is illustrated in Algorithm 1. In step 1, the encoder receives the cover image c and the secret image s as inputs and hides the secret image s into cover image c to generate the stego image c'. Now, at the second step, the value of the loss function is calculated as the difference between the original cover image c and the stego image c', as defined in Eq. (5). In step 3, this loss function is backpropagated to the encoder network and the weights of the encoder are updated. Then, the encoder iterates steps 2 to 3 until the value of the loss function becomes negligible and the encoder is trained enough to generate the imperceptible stego image. Now, the decoder receives the stego image c' and extracts the secret image s' at the next step. Next, in step 6, the value of the loss function is computed as per Eq. (6). Now, in step 7, this value of the loss function is backpropagated to the decoder network and the weights of the decoder are updated. The procedure followed in steps 6 to 7 is repeated until the value of the loss function calculated for the decoder network becomes negligible. The decoder is trained enough to extract the secret image without degrading its quality. Now, the reconstructed secret image s' is obtained as the output image.

Algorithm 1. Training Procedure of Encoder and Decoder

Input: c, s
Output: s'
Initialize random weights of convolution for the encoder as well as decoder networks
For i =1 to N epochs do
For j=1 to B batches do
(1) $c' = \text{encode}(c \oplus s)$ *// secret image is embedded in cover image*
(2) Compute $\text{loss}_{\text{encoder}} = \| c - c' \|$ as per equation (5)
(3) Back propagate the loss computed in step 2 and update the weights of Encoder *// loss function is backpropagated to the encoder network and the weights of the encoder are updated*
(4) Repeat step 2 to 3 untill $\text{loss}_{\text{encoder}} \approx 0$ *// the encoder iterates steps 2 to 3 until the value of the loss function becomes negligible*
(5) $s' = \text{decode}(c')$ *// the decoder receives the stego image c' and extracts the secret image s'*
(6) Compute $\text{loss}_{\text{decoder}} = \| s - s' \|$ as per equation (6)
(7) Back propagate the loss computed at step 6 and update the weights of the decoder *// this value of the loss function is backpropagated to the decoder network and the weights of the decoder are updated.*
(8) Repeat step 6 to 7 until $\text{loss}_{\text{decoder}} \approx 0$. *// The procedure followed in steps 6 to 7 is repeated until the value of the loss function calculated for the decoder network becomes negligible*
End inner for loop
End outer for loop
Output: s'

Evaluation metrics

For evaluating the performance of the steganography techniques based on U-Net, V-Net, and U-Net++, the following metrics are used. These metrics are the measure of image quality as discussed in[50–52]. Thus, these are important for comparison in the quality of the images reconstructed by the steganography techniques.

Mean square error (MSE). This is the difference between the pixel values of secret and reconstructed images[53] as defined in Eq. (7). In Eq. (7), a and b are image pixel coordinates with the size of $M \times N$ pixels. Here, I_1 and I_2 are original and reconstructed images, respectively[50]. Thus, the minimum value of MSE favors the better quality of the reconstructed image.

$$\mathrm{MSE} = \frac{\sum_{a=1}^{a=M}\sum_{b=1}^{b=N}[I_1(a,b) - I_2(a,b)]^2}{M \times N} \tag{7}$$

Peak signal to noise ratio (PSNR). It is the peak signal-to-noise ratio between the secret and reconstructed images. In Eq. (8), R_I is the maximum variation in the input image data type, and MSE is the mean square error. The value of PSNR is used to measure the visual quality difference between two images[51]. Its high value indicates the better quality of the reconstructed image.

$$\mathrm{PSNR} = 10.\log_{10}\left(\frac{R_I^2}{\mathrm{MSE}}\right) \tag{8}$$

Structural similarity Index (SSIM). This metric is used to measure the deterioration in the image quality caused due to processing. It also measures the difference in the perceptual quality of the secret and reconstructed images. SSIM is the combined evaluation for the luminance (l), contrast (c), and the structure (s) of two images (a and b)[51].

$$\mathrm{SSIM}(a,b) = [l(a,b)]^{\alpha}.[c(a,b)]^{\beta}.[s(a,b)]^{\gamma} \tag{9}$$

In Eq. (9),

$$l(a,b) = \frac{2\mu_a\mu_b + C_1}{\mu_a^2 + \mu_b^2 + C_1}, c(a,b) = \frac{2\sigma_a\sigma_b + C_2}{\sigma_a^2 + \sigma_b^2 + C_2}, s(a,b) = \frac{\sigma_{ab} + C_3}{\sigma_a\sigma_b + C_3} \tag{10}$$

where, μ_a and μ_b are the average of original and reconstructed images,σ_a and σ_b are standard deviations, and $\sigma_{a.b}$ is covariance for images a and b. If $\alpha = \beta = \gamma = 1$ and, $C_3 = \frac{C_2}{2}$, SSIM can be simplified as:

$$SSIM = \frac{(2\mu_a\mu_b + C_1)(2\sigma_{ab} + C_2)}{(\mu_a^2 + \mu_b^2 + C_1)(\sigma_a^2 + \sigma_b^2 + C_2)} \tag{11}$$

Entropy. Entropy ($H(K)$) is the degree of uncertainty present in an image as defined in Eq. (12). In this equation, p_i is the occurrence probability of the pixel i in the image K. Entropy is used to quantify the information available in the image. More amount of information indicates better quality of image[52].

$$H(K) = -\sum_{i=1}^{n} p_i log_2 p_i \tag{12}$$

Blind/reference less image spatial quality evaluator (BRISQUE SCORE). This is used to estimate the perceptual quality of an image using the locally normalized luminance coefficients. In this manuscript, the BRISQUE SCORE as defined in the reference[48] is used. It provides the no-reference image quality score by comparing the image to default natural scene images with similar distortions. The mean score is assigned between 0 and 100. A low score signifies better perceptual quality[53].

Results and discussion

Image quality measures. In this section, the sample results obtained by employing U-Net, V-Net, U-Net++ encoders and the decoder, designed in this research, are shown in Figs. 6, 7 and 8. For proving the efficacy of the encoders and decoder, the difference (diff cover) between the original and encoded cover image is calculated. Also, the difference (diff secret) between the original and decoded secret image is calculated. Both the differences are approximate to zero. Thus, plotting the pixels of the difference gave the black color image as shown in Figs. 6, 7 and 8.

Further, the visual quality of the stego and reconstructed images is demonstrated in Tables 3, 4, 5, 6, and Fig. 9, respectively.

It is evident from the values of MSE shown in the first row of Table 3 that there is a difference of merely 0.0001 in the MSE of cover and stego images generated by the U-Net encoder and 0.0003 in the original and reconstructed secret images by the decoder. Further, it is clear from the results shown in the second row that the difference is 0.0019 for cover and stego images and 0.0010 for secret and reconstructed images when V-Net is employed as an Encoder. It is apparent from the third row that the difference for cover and stego image is 0.007

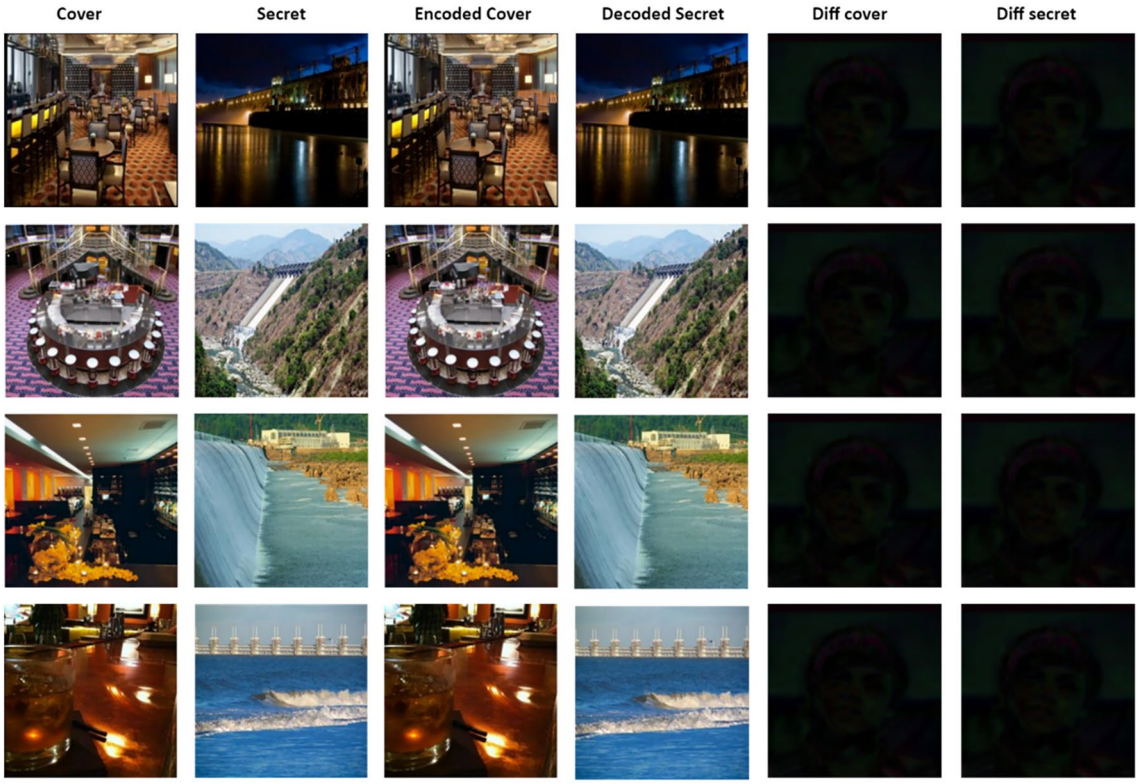

Figure 6. Test samples of U-Net encoder model.

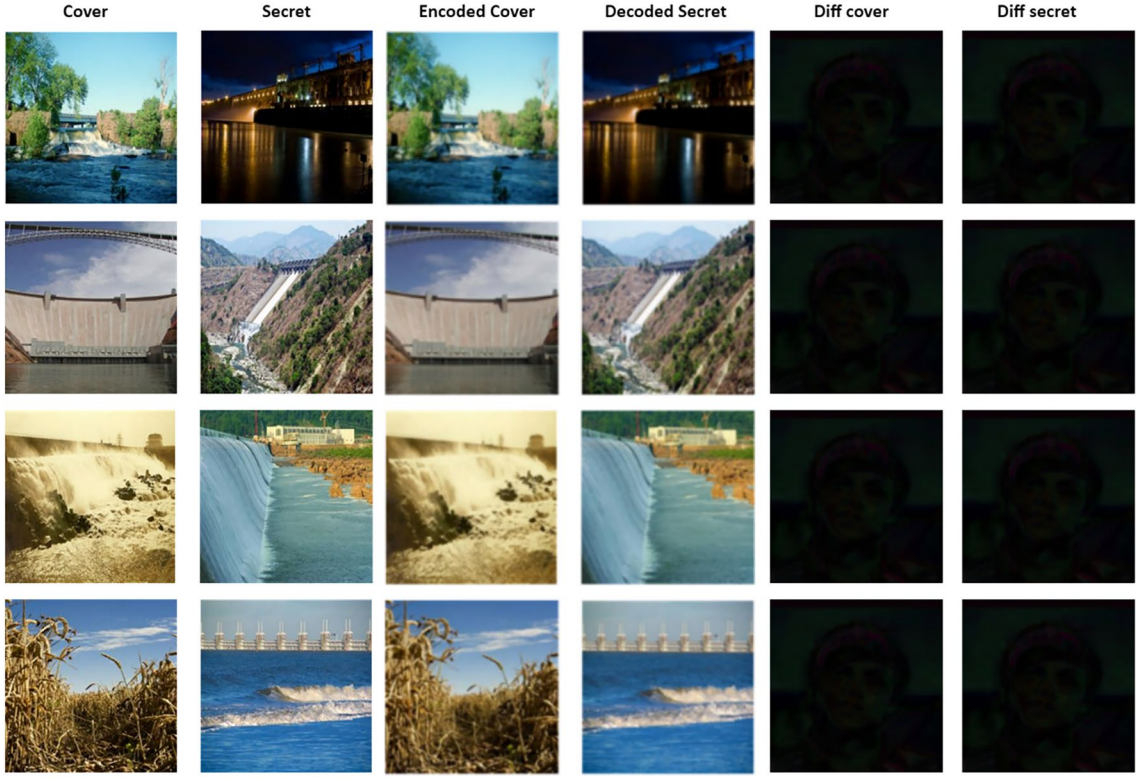

Figure 7. Test samples of V-Net encoder model.

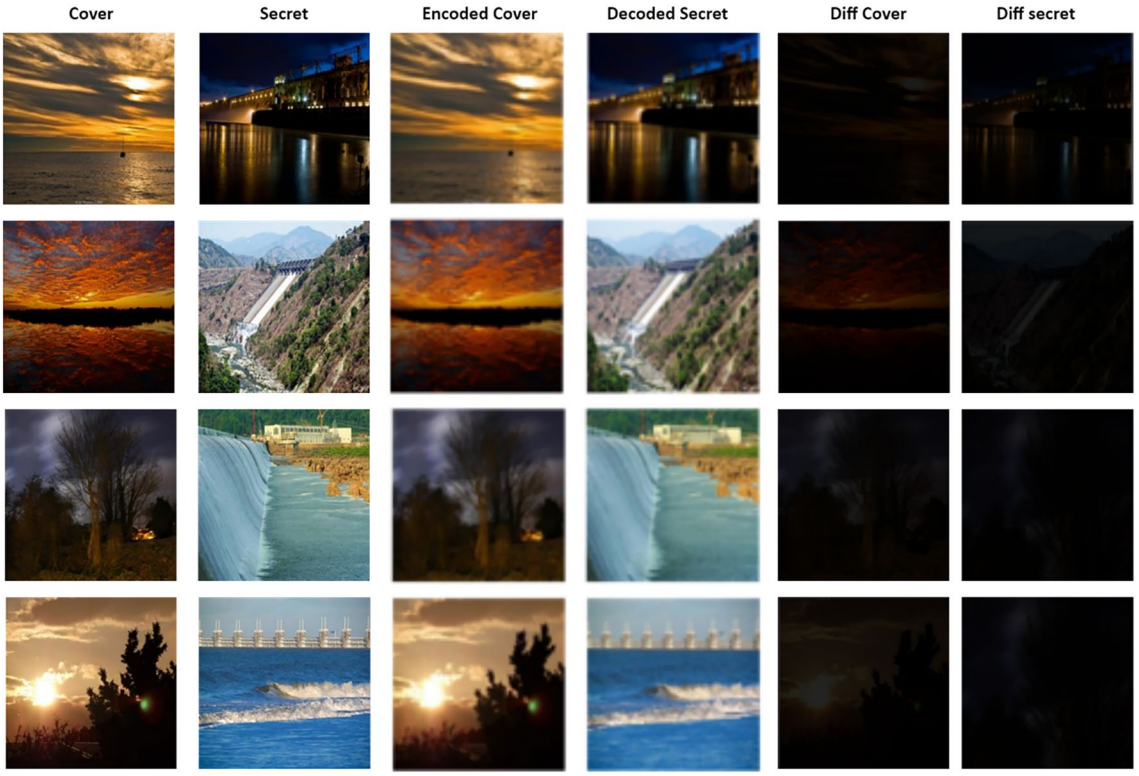

Figure 8. Test samples U-Net++ encoder model.

Encoder model	MSE	
	Cover and Stego image	Secret and reconstructed secret image
U-Net Encoder	0.0001	0.0003
V-Net Encoder	0.0019	0.0010
U-Net++ Encoder	0.007	0.006

Table 3. Mean square error of Stego image and reconstructed secret image.

Encoder model	PSNR					
	Cover and Stego image			Secret and reconstructed secret image		
	Minimum	Maximum	Mean	Minimum	Maximum	Mean
U-Net encoder	35.00	41.02	38.00	29.00	38.94	38.00
V-Net encoder	27.80	31.20	30.00	30.20	34.40	33.00
U-Net++ encoder	18.50	29.00	24.00	21.00	33.40	27.00

Table 4. Peak signal to noise ratio of Stego and reconstructed secret image.

Encoder model	SSIM (%)					
	Cover and Stego image			Secret and reconstructed secret image		
	Minimum	Maximum	Mean	Minimum	Maximum	Mean
U-Net Encoder	90.00	99.40	98.75	89.74	99.89	98.69
V-Net Encoder	93.00	97.10	96.80	92.20	98.30	98.10
U-Net++ Encoder	88.00	95.40	91.00	85.00	95.50	93.00

Table 5. Structure similarity index of Stego and reconstructed secret image.

Encoder model	Entropy								
	Original image			Stego image			Reconstructed secret image		
	Minimum	Maximum	Mean	Minimum	Maximum	Mean	Minimum	Maximum	Mean
U-Net Encoder	6.31	7.94	7.32	6.10	7.94	7.45	5.79	7.91	7.47
V-Net Encoder	6.31	7.94	7.32	7.00	7.91	7.59	5.3	7.85	7.40
U-Net++ Encoder	6.31	7.94	7.32	6.80	7.90	7.54	5.69	7.80	7.40

Table 6. The entropy of Stego and reconstructed secret image.

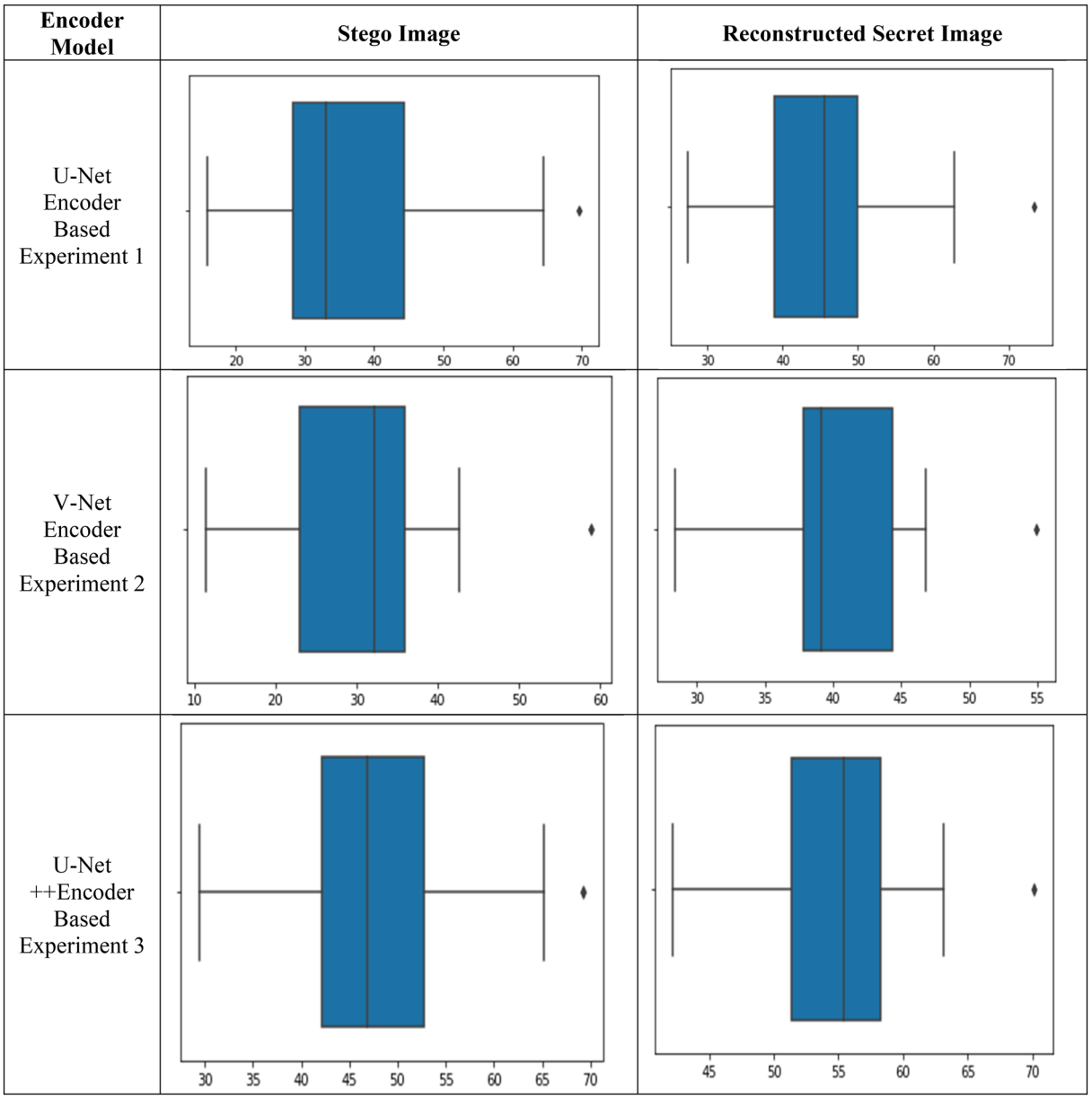

Figure 9. BRISQUE score scaling.

and 0.006, for secret and reconstructed images in the U-Net++ encoder model. Based on the comparison of the values of MSE presented in Table 3, it is observed that the U-Net encoder reports the minimum MSE. This proves the supremacy of U-Net based encoders over the V-Net and U-Net++ based encoders.

Further, the values of PSNR presented in Table 4 also showcase the error of reconstruction. In strong contrast to the MSE, higher PSNR indicates better quality of image reconstruction. For the stego image, the U-Net, V-Net, and U-Net++ report the highest PSNR of 41.02, 31.20, and 29.0 decibels, respectively. For the reconstructed

Encoder model	BRISQUE Score					
	Stego image			Reconstructed secret image		
	Minimum	Maximum	Mean	Minimum	Maximum	Mean
U-Net encoder	15.87	69.656	36.02	27.34	73.45	45.36
V-Net encoder	11.40	58.91	31.30	28.39	54.95	40.46
U-Net++ encoder	29.44	69.27	47.83	42.18	70.13	54.71

Table 7. BRISQUE Score of Stego and reconstructed secret image.

Encoder model	Secret image size (absolute capacity)	Cover image size	Relative payload capacity
U-Net encoder	256 × 256 × 3	256 × 256 × 3	1
V-Net encoder	256 × 256 × 3	256 × 256 × 3	1
U-Net++ encoder	256 × 256 × 3	256 × 256 × 3	1

Table 8. Comparisons of steganographic payload capacity.

secret image, U-Net, V-Net, and U-Net++ based architectures report the maximum values of 38.94, 34.40, and 33.40, respectively. It is apparent from these values that U-Net based architecture generates better quality images than V-Net and U-Net++ based architectures.

Now, the values of SSIM shown in Table 5, demonstrate the quality degradation caused during image reconstruction. The U-Net, V-Net, and U-Net++ based architectures report the highest similarity of 99.89%, 97.10%, and 95.40%, respectively, between stego and cover images. Similarly, for the reconstructed secret image, the values of SSIM are 99.40%, 98.30%, and 95.50% for the U-Net, V-Net, and U-Net++ based architectures, respectively.

It is clear from the values of SSIM that the U-Net based architecture generates the stego and reconstructed secret images with the highest degree of similarity. Thus, the generated images are approximately indistinguishable from their corresponding original images.

As shown in Table 6, the values of entropy are calculated to showcase the degree of randomness in the generated images. For the stego image, the U-Net, V-Net, and U-Net++ based architectures report the mean entropy of 7.94, 7.91, and 7.90, respectively. For the reconstructed secret images, these architectures give the mean entropy of 7.91, 7.85, and 7.80, respectively. It is evident from the given values that all the three networks generate images with a similar degree of randomness as the original images. These architectures generate images that retain the maximum information. Further, it is also observed that U-Net-based architecture gives the highest entropy values for both the stego and reconstructed secret images. Therefore, it outperforms the V-Net and U-Net++ based architectures in terms of retaining the information.

Now, the values of the BRISQUE score, as shown in Fig. 9, are calculated to prove the perceptual quality of the generated images. The lower values of the BRISQUE score favor the better perceptual quality of images. For the stego images, the U-Net, V-Net, and U-Net++ based architectures give the lowest BRISQUE score of 15.87, 11.40, and 29.44, respectively. For the reconstructed images the lowest values are 27.34, 28.39, and 42.18 for the U-Net, V-Net, and U-Net++ based architectures, respectively. These values indicate that the V-Net architecture outperforms the U-Net, and U-Net++ architectures in terms of the perceptual quality of generated images.

It is observed from the experimental results obtained that the U-Net based encoder generates high-quality stego and reconstructed secret images as compared to the other two encoder models. Further, the V-Net based encoder regenerated the images with good perceptual quality, Still, it lacks information preserving and maintaining the structural similarity between the generated images and their corresponding input images. It is also evident from the results shown in Tables 3, 4, 5, 6 and 7 and Fig. 9 that the U-Net++ encoder is a poor performer than U-Net and V-Net architectures in image steganography.

Steganographic payload capacity. An efficient steganographic technique aims to embed maximum information into cover media without affecting the visual quality so that an attacker cannot percept it as a target image. The payload capacity is the embedding rate at which the number of secret data bits is embedded in the cover image, Table 8 depicts the comparison of payload capacity between proposed and existing techniques, here second and third column shows the size of the secret and cover image, respectively. The fourth column represents the relative payload capacity, calculated as per Eq. (13).

$$\text{Relative Payload Capacity} = \frac{\text{Absolute Capacity}}{\text{Cover Image Size}} \tag{13}$$

Here, a secret color image of size 256 × 256 is embedded in the cover image of the same size. Hence, the relative payload capacity of all the three steganography techniques is 1 byte/pixel. In CNN-based steganographic methods[22,24], a gray-scale secret image is embedded in the color cover image to maintain stego image quality. It can be observed from Tables 3, 4, 5, and 8, that the proposed techniques improve the steganographic payload capacity without compromising the image quality.

Conclusions

In this paper, performance parameter assessment of deep learning-based image steganography techniques U-Net, V-Net, and U-Net++ based encoders are carried out. The encoder architectures generate the stego image that hides the secret image into the cover image. The unique and robust decoder is designed that effectively extracts the secret image from the stego image. The visual quality of the secret image reconstructed by the decoder is evaluated in terms of MSE, SSIM, PSNR, Entropy, and Brisque Score. It is observed from the comparative performance analysis of U-Net, V-Net, and U-Net++ based architectures that the U-Net architecture outperforms the other two architectures. This architecture ensures the high payload capacity without compromising the visual quality of reconstructed and stego images. Thus, it is useful for securing the data of real-life applications such as healthcare, defense, scientific documents, etc. Further, there is a vast scope of integrating the encryption and steganography techniques for enhancing security.

Ethical statement. All methods were carried out in accordance with relevant guidelines and regulations.

Data availability

The Datasets analyzed during the current study are available for public access through Labeled Faces in the Wild[46] and Know Your Data[47].

Received: 24 March 2022; Accepted: 25 July 2022
Published online: 07 October 2022

References

1. Kaur, M. & Kumar, V. Comprehensive survey on image encryption techniques. *Arch. Comput. Methods Eng.* **27**, 15–43 (2018).
2. Deshpande, N., Kamalapur, S. & Jacobs, D. Implementation of LSB steganography and its evaluation for various bits. In *2006 1st International Conference on Digital Information Management*, 173–178 https://doi.org/10.1109/ICDIM.2007.369349 (2006).
3. Nolkha, A., Kumar, S. & Dhaka, V. S. Image steganography using LSB substitution: A comparative analysis on different color models. In *Smart Systems and IoT: Innovations in Computing. Smart Innovation, Systems and Technologies* Vol. 141 (eds Somani, A. *et al.*) (Springer, Singapore, 2020).
4. Subramanian, N., Elharrouss, O., Somaya, A. M. & Bouridane, A. Image steganography: A review of the recent advances. *IEEE Access.* https://doi.org/10.1109/ACCESS.2021.3053998 (2021).
5. Akhtar, N., Khan, S., & Johri, P. An improved inverted LSB image steganography. In *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 749–755 https://doi.org/10.1109/ICICICT.2014.6781374 (2014).
6. Singh, A. & Singh, H. An improved LSB based image steganography technique for RGB images. In *2015 IEEE International Conference on Electrical, Computer and Communication Technologies* (*ICECCT*), 1–4 https://doi.org/10.1109/ICECCT (2015).
7. Zhou, X., Gong, W., Fu, W. & Jin, L. An improved method for LSB based color image steganography combined with cryptography. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS),* 1–4 https://doi.org/10.1109/ICIS.2016.7550955 (2016).
8. Sugathan, S. An improved LSB embedding technique for image steganography. In *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 609–612 https://doi.org/10.1109/ICATCCT.2016.7912072 (2016).
9. Kadhim, I. J., Premaratne, P., Vial, P. J. & Halloram, B. Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research. *Neurocomputing* https://doi.org/10.1016/j.neucom.2018.06.075 (2018).
10. Kaur, H. & Rani, J. A survey on different techniques of steganography. *MATEC Web Conf.* https://doi.org/10.1051/matecconf/2016570 (2016).
11. Kumar, V. & Kumar, D. A modified DWT-based image steganography technique. *Multimed. Tools Appl.* **77**, 13279–13308. https://doi.org/10.1007/s11042-017-4947-8 (2018).
12. Patel, H. & Dave, P. Steganography technique based on DCT coefficients. *Int. J. Eng. Res. Appl.* **2**, 713–717 (2012).
13. Kumar, V. & Kumar D. Digital image steganography based on combination of DCT and DWT. In *Information and Communication Technologies. ICT 2010. Communications in Computer and Information Science*, Vol. 101 (Springer, Berlin) https://doi.org/10.1007/978-3-642-15766-0_102 (2010).
14. Raja, K. B., Kumar, K. K., Kumar, S. K., Lakshmi, M. S., Preeti, H., Venugopal, K. R. & Patnaik, L. M. Genetic algorithm based steganography using wavelets. In *International Conference on Information Systems Security ICISS* Vol. **4812** (Springer, Berlin) https://doi.org/10.1007/978-3-540-77086-2_5 (2007).
15. Nosrati, M., Hanani, A. & Karimi, R. Steganography in image segments using genetic algorithm. In *Fifth International Conference on Advanced Computing & Communication Technologies*, 102–107 https://doi.org/10.1109/ACCT.2015.57 (2015).
16. Khamrui, A. & Mandal, J. K. A genetic algorithm based steganography using discrete cosine transformation (GASDCT). *Procedia Technol.* **10**, 105–111 (2013).
17. Karakış, R., Güler, İ, Çapraz, İ & Bilir, E. A novel fuzzy logic-based image steganography method to ensure medical data security. *Comput. Biol. Med.* **67**, 172–183 (2015).
18. Vanmathi, C. & Prabu, S. Image steganography using fuzzy logic and chaotic for large payload and high imperceptibility. *Int. J. Fuzzy Syst.* **20**, 460–473 (2018).
19. Sun, W., Jia, M., Yu, S., Dong, B. & Li, X., An SVM based secural image steganography algorithm for IoT. In *Cyberspace Safety and Security* 11983 (Springer, 2019).
20. Tanwar, R. & Malhotrab, S. Scope of support vector machine in steganography. In *2017 IOP Conference Series: Materials Science and Engineering* 225 (2017).
21. Chahar, V., Laddha, S., Sharma, A. & Dogra, N. Steganography techniques using convolutional neural networks. *Rev. Comput. Eng. Stud.* **7**, 66–73 (2020).
22. Hussain, I., Zeng, J., Qin, X. & Tan, S. A survey on deep convolutional neural networks for image steganography and steganalysis. *Ksii Trans. Internet Inf. Syst.* **14**, 1228–1248 (2020).
23. Rehman, A., Rahim, R., Nadeem, S. & Hussain, S. End-to-end trained CNN encoder-decoder networks for image steganography. In *Computer Vision and Pattern Recognition, Cornell University, ECCV 2018 Workshop Paper*, arXiv:1711.07201 (2018).
24. Baluja, S. Hiding images within images. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**, 1685–1697 (2019).
25. Duan, X., Liu, N., Gou, M., Wang, W. & Qin, C. SteganoCNN: Image steganography with generalization ability based on convolutional neural network. *Entropy* https://doi.org/10.3390/e22101140 (2020).
26. Zhang, R., Dong, S. & Liu, J. Invisible steganography via generative adversarial networks. *Multimed. Tools Appl.* **78**, 8559–8575 (2019).

27. Duan, X. *et al.* Reversible image steganography scheme based on a U-Net structure. *IEEE Access.* https://doi.org/10.1109/ACCESS.2019.2891247 (2019).
28. Van, T. P., Dinh, T. H. & Thanh, T. M. Simultaneous convolutional neural network for highly efficient image steganography. In *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, 410–415 https://doi.org/10.1109/ISCIT.2019.8905216 (2019).
29. Wu, P., Yang, Y. & Li, X. StegNet: Mega image steganography capacity with deep convolutional network. *Fut. Internet.* https://doi.org/10.3390/fi10060054 (2018).
30. Tang, W., Li, B., Tan, S., Barni, M. & Huang, J. CNN-based adversarial embedding for image steganography. *IEEE Trans. Inf. Forensics Secur.* **14**, 2074–2087 (2019).
31. Yang, J., Liu, K., Kang, X., Wong, E. K. & Shi, Y. Q. Spatial image steganography based on generative adversarial network. In *Computer Vision and Pattern Recognition, Cornell University,* arXiv:1804.07939 (2018).
32. Hu, D., Wang, L., Jiang, W., Zheng, S. & Li, B. A novel image steganography method via deep convolutional generative adversarial networks. *IEEE Access* https://doi.org/10.1109/ACCESS.2018.2852771 (2018).
33. Yang, J., Ruan, D., Huang, J., Kang, X. & Shi, Y. Q. An embedding cost learning framework using GAN. *IEEE Trans. Inf. Forensics Secur.* **15**, 839–851 (2020).
34. Sharma, V. K., Sharma, P. C., Goud, H. & Singh, A. Hilbert quantum image scrambling and graph signal processing-based image steganography. *Multimed. Tools Appl.* **81**, 17817–17830 (2022).
35. Shen, Q., Jiang, T., Zhu, Y. & Wu, Y. An improved image steganography scheme based on partial preservation embedding algorithm for wireless visual sensor networks. *Math. Probl. Eng.* https://doi.org/10.1155/2021/6618134 (2021).
36. Telli, M., Othmani, M. & Ltifi, H. An improved image steganography model based on Deep Convolutional Neural Networks. *EasyChair* (2022).
37. Peter, G., Sherine, A., Teekaraman, Y., Kuppusamy, R. & Radhakrishnan, A. Histogram shifting-based quick response steganography method for secure communication. *Wirel. Commun. Mob. Comput.* (2022).
38. Alzubaidi, L. *et al.* Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data.* https://doi.org/10.1186/s40537-021-00444-8 (2021).
39. Niu, X. & Suen, C. Y. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recogn.* **4**, 1318–1325 (2012).
40. Jin, N., Wu, J., Ma, X., Yan, K. & Mo, Y. Multi-task learning model based on multi-scale CNN and LSTM for sentiment classification. *IEEE Access* **8**, 77060–77072. https://doi.org/10.1109/ACCESS.2020.2989428 (2020).
41. Singh, R. D., Mittal, A. & Bhatia, R. K. 3D convolutional neural network for object recognition: a review. *Multimed. Tools Appl.* **78**, 15951–15995. https://doi.org/10.1007/s11042-018-6912-6 (2019).
42. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional networks for biomedical image segmentation. *Computer Vision and Pattern Recognition, Cornell University*, arXiv:1505.04597 (2015).
43. Milletari, F., Navab, N. & Ahmadi, S. A. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *Computer Vision and Pattern Recognition, Cornell University* arXiv:1606.04797 (2016).
44. Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N. & Liang, J. UNet++: A nested U-Net architecture for medical image segmentation. In *Computer Vision and Pattern Recognition, Cornell University* arXiv:1807.10165 (2018).
45. Li, Y., Zhang, T., Liu, Z. & Hu, H. A concatenating framework of shortcut convolutional neural networks. In *Computer Vision and Pattern Recognition, Cornell University* arXiv:1710.00974 (2018).
46. http://vis-www.cs.umass.edu/lfw/.
47. https://knowyourdata-tfds.withgoogle.com/.
48. Ide, H. & Kurita, T. Improvement of learning for CNN with ReLU activation by sparse regularization. In *2017 International Joint Conference on Neural Networks (IJCNN)*, 2684–2691 https://doi.org/10.1109/IJCNN.2017.7966185 (2017).
49. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations, San Diego* arXiv:1412.6980 (2015).
50. Sara, U., Akter, M. & Uddin, M. Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study. *J. Comput. Commun.* https://doi.org/10.4236/jcc.2019.73002 (2019).
51. Hore, A. & Ziou, D. Image quality metrics: PSNR vs. SSIM. In *Proceedings of IEEE International Conference on Pattern Recognition* 2366–2369 (2010).
52. Tsai, D. Y., Lee, Y. & Matsuyama, E. Information entropy measure for evaluation of image quality. *J. Digit. Imaging* **21**, 338–347. https://doi.org/10.1007/s10278-007-9044-5 (2008).
53. Moorthy, A. K. & Bovik, A. C. Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE Trans. Image Process.* **20**, 3350–3364 (2011).

Author contributions

V.H. designed the study, analyzed data, and implemented the pipeline for model development; V.S.D and M.K. analyzed selected variants; G.R., M.O., and H.N.L., contributed towards proofreading the manuscript along with support in implementation.

Funding

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by the Korean Government through the Ministry of Science and ICT (MSIT) (Development of of decentralized consensus composition technology for large- scale nodes) under Grant 2021-0-00118; and in part by MSIT, South Korea, through the Information Technology Research Center (ITRC) Support Program supervised by IITP under Grant IITP-2021-0-01835.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to H.-N.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.