Sensors and Materials, Vol. 37, No. 3 (2025) 1153–1169
MYU Tokyo

S & M 3978

Image Compression Transmission of Embedded Device Based on Depthwise Separable Convolutional Autoencoder

Ming-Tsung Yeh,[1] Nian-Tang Wu,[2] Yen-Ting Lua,[1]
Neng-Sheng Pai,[1*] and Wei-Yin Lo[2]

[1]Department of Electrical Engineering, National Chin-Yi University of Technology,
57, Sec. 2, Zhongshan Rd., Taiping Dist, Taichung 411030, Taiwan
[2]Department of Electrical Engineering, National Changhua University of Education,
No. 1, Jinde Rd., Changhua City, Changhua County 50007, Taiwan

(Received June 21, 2024; accepted March 11, 2025)

Keywords: image compression, depthwise separable convolutional autoencoder, image transmission, image encryption, embedded device

Embedded devices, crucial components in various industries, often operate independently, executing specific tasks efficiently. Their compact size, low maintenance, and energy consumption make them highly desirable. With the ability to connect to networks, these devices facilitate communication with other devices, forming a robust computing system. However, image transmission on these devices poses a challenge, requiring a delicate balance between efficiency and cybersecurity. In this paper, we propose a novel solution, a depthwise separable convolutional autoencoder (DSCAE) network model, which is unique in its ability to address image compression and encryption simultaneously. This model incorporates the high-efficiency depthwise separable convolution (DSC) of the Xception network into the convolutional autoencoder (CAE) model, optimizing image transmission. It also utilizes the Xception middle flow structure to synthesize more features, thereby enabling the decoder to reconstruct the predicted image with greater accuracy and enhancing the model's performance. The output of the encoder is in ciphertext format to ensure the confidentiality of transmitted images, effectively safeguarding them and reducing the risks associated with unauthorized access during image communication. The experimental results demonstrate the efficacy of this approach, with the original photos transmitted by the proposed deep learning image encoding method retaining image quality and encryption by transferring only one-sixtieth of the original image size. On the receiver site, the reconstructed images can achieve an average peak signal-to-noise ratio (*PSNR*) of 29 dB compared with the actual image at the transmitter, thereby significantly improving the efficiency and security of image transmission on embedded devices.

*Corresponding author: e-mail: pai@ncut.edu.tw
https://doi.org/10.18494/SAM5204

ISSN 0914-4935
https://myukk.org/

1. Introduction

Embedded devices are widely implemented in various industries and independently execute a particular task or a series of tasks. They are designed with small size and require less maintenance and lower energy consumption in comparison to other computational equipment. Nowadays, most embedded devices have the capability of network connection to provide communication with other devices or powerful equipment and form a more extensive computing system. Developing embedded devices and network technologies has enabled real-time image transmission to be widely used in various applications. However, in image transmission, two critical aspects need to be considered: image transmission efficiency and cybersecurity. These two items have a trade-off relationship in device-sharing resources. Balancing them is an essential subject, particularly in embedded devices. With the increasing concern regarding data security and privacy, it is necessary to ensure that images are encrypted before they are transmission to prevent unauthorized access. Moreover, the efficiency of image transmission plays a pivotal role in real-time applications, exerting a direct effect on the delivery time and quality of the transmitted images. However, the embedded device can only execute specific tasks without complex computing compared with a full-featured computer because their key drawbacks are their designed size and limited capabilities. Hence, the limited computing resources must be well used to improve the performance of image transmission and security.

Traditionally, some methods have been employed to improve the efficiency of image transmission. Images can be compressed first, and many approaches can be adopted to reduce the transmitted size. Secondly, downscaling the image resolution can also help reduce the amount of data transmitted. Lastly, optimizing the Internet Protocol (IP) used for transmission can further enhance the efficiency of the transmission process. The image size can be reduced by downscaling its spatial resolution. Still, the restored image may produce noise, be intolerably blurred, and have jagged effects after resizing to the original size at the receiver site. Some researchers even apply super-resolution approaches to reconstruct the higher-resolution image, but the process is more complex and not guaranteed to restore the original image.[1–3] Transmission performance can be improved by optimizing network-level accelerated progress protocols, such as enlarging the transmission window size and adopting advanced traffic congestion control. However, these optimizations the require modification of the network interface firmware and must obey the same rules in the routing path. Image compression is a prevalent technique for implementing data compression on digital images, with the objective of decreasing image size. This method, including traditional computer vision approaches and learning-based models with lossy or lossless algorithms, is the most effective and can instantly provide better outcomes. Many researchers have proposed numerous approaches to compress images for transmission and storage.[4–7] Nevertheless, conventional compression methods using open standard algorithms without encryption are easy to reveal and steal. Methods based on the deep learning network model are applied to improve image transmission performance and address data security problems due to the image being encoded by its feature maps. This type of codec has been used increasingly recently.

Learning-based image compression, designed using deep learning neural networks, is an emerging technology to transform image pixels from high dimensions to low dimensions. Many sample images are used to train the networks and learn the feature selection knowledge applied to generate a compressed code of the original image that reduces the transmission size. On the other hand, the receiver site uses the trained reconstructive network, which is part of the whole network, to produce a predicted image that approximates the original image. Learning-based compression models are broadly divided into four types: convolutional neural network (CNN), recurrent neural network (RNN), generative adversarial network (GAN), and autoencoder network.[7–9] The CNN-based image compression includes two flexible nonlinear components: the analysis encoder and synthesis decoder modules. The analysis module uses convolution, division, and down-sampling stages to produce downscaled feature maps. These maps are then adjusted and normalized using a subsequent normalization layer. Generalized divisive normalization (GDN) is often applied to normalize feature maps and has shown superior results. The synthesis decoder uses inverse modules (e.g., iGDN, up-sampling) to reconstruct the image.[10,11] Recent approaches have focused on developing deeper layers to enhance network performance.[12] However, this significantly increases network complexity and computing resource usage.

In GAN-based image compression, the GAN structure is introduced into the decoder of the CNN-based method to improve the restored image quality. This model comprises an encoder, a decoder, and a discriminator. The encoder is used to capture input image features, and then the decoder becomes the GAN generator applied to reconstruct an image close to the original from the input features. The discriminator performs the loss function to determine the difference between the decoder output and the ground truth to update the decoder weight parameters. The output of the generator is optimized to restrain some visual artifacts, such as blurring or blocking, making it close to the natural image. This compression method performed better than the CNN-based approach, which improves the restored image quality, especially at a low bit rate. However, the compression artifacts produced by the decoder have been shown to reduce the realism of the restored image, even if it appears natural. Galteri *et al.* applied an ensemble of conditional GAN and selected an appropriate module using the quality predictor of the compressed image to reconstruct a sharper and more realistic image; they efficiently reduced the mosquito noise and ringing artifacts.[13] Kudo *et al.* used a regularization method that optimizes relationships between the coding features and the restored images to maximize mutual information and obtain a reconstructed image that retains the same appearance as the original image.[14] Wu *et al.* proposed a masking algorithm to generate the importance map for compensating restored image distortion at a low bit rate.[15] They applied a symmetrical multiscale structure, which is adaptable to different object sizes, to the decoder and discriminator. Wang *et al.* designed a semantic-perceptual residual compensation block and a U-shaped encoder–decoder structure with a dense residual connection on the GAN framework to obtain higher visual quality.[16]

The RNN-based model performs better in video stream compression and variable bit-rate coding structures. In this approach, a pixel-level RNN, long short-term memory (LSTM), or gated recurrent unit (GRU) network was applied to find continuously adjacent pixel relationships

of images and reduce the coded feature map size. Toderici *et al.* used repetitive pixel-level RNN and binary RNN structures to form the entropy encoder and induce variable bit-rate compression. The obtained results were superior to those of competing methods in the field of full-resolution lossy image compression.[17] Ororbia *et al.* applied a nonlinear RNN estimator for iterative decoding that uses spatial context information to generate images from a decoder with low reconstruction error and high perceptual quality. Still, their method only focuses on single-channel grayscaled images.[18] Hu *et al.* proposed a progressive spatial RNN for video codecs to deal with intraprediction learning. The bit-rate reduction can be improved, and the image restoration quality can be the same as that of high-efficiency video coding.[19] Islam *et al.* applied a LSTM network as an encoder to reduce the unnecessary image data for variable rate compression and used the pixel-level RNN for quantization.[12] However, the RNN-based model uses the iterative method to train networks using sequential pixels in the sample images. This model relies heavily on CPU computation processing and memory usage, and hence requires a longer training time and a complicated inference process.

The existing learning-based image compression methods need extra network layers designed on the encoder and decoder to reduce artifact effects and improve restored image quality. These methods make the neural network larger and require additional information added to the bit stream for the decoder. However, these methods do not adapt to embedded devices with limited resources. In this paper, we propose a depthwise separable convolutional autoencoder (DSCAE) network model for simultaneously processing image compression and encryption. The DSCAE applies the DSC structure of Xception to perform spatial and channel learning processes, which reduce weight parameters and improve network efficiency, respectively. The image encryption approach bears a similarity to visual cryptography. Traditional visual cryptography can be decrypted using human vision. The DSCAE model is decoded using variable compact feature maps. Furthermore, this model can extract the image to a small feature map size. It can effectively achieve the desired image compression effect to improve image transmission efficiency. Our proposed approach also ensures the confidentiality of transmitted images, effectively safeguarding them.

2. Proposed Method

In this section, we present our proposed DSCAE image compression method and introduce a data augmentation approach to train networks to enhance the performance and efficiency of image transmission for embedded devices. We also describe the cropped augmentation image method, which focuses on augmenting the training dataset by cropping image regions, to improve network model generalization and robustness.

2.1 System framework

Figure 1 shows the overall system framework. Our proposed DSCAE network model compresses pictures taken by a camera set in the embedded device. This reduces the image size to improve transmission efficiency and decrease storage space. The DSCAE model is divided

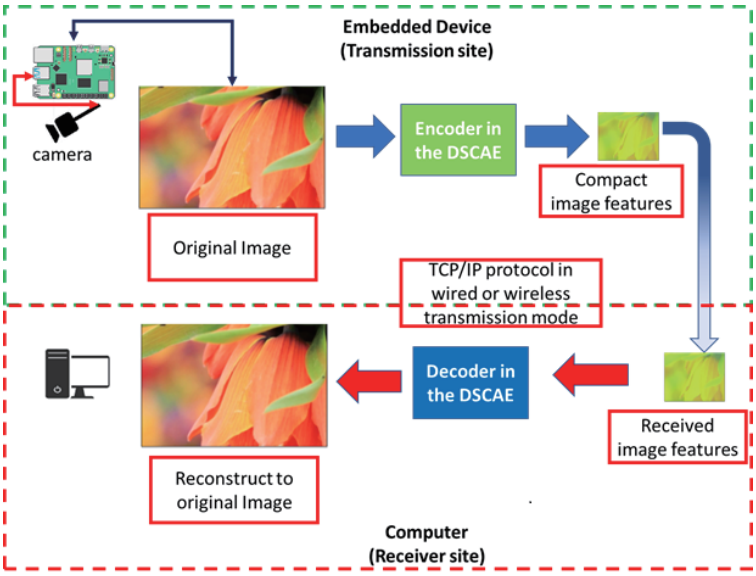


Fig. 1. (Color online) System framework of the proposed DSCAE network.

into two parts: encoder and decoder. The encoder module of DSCAE is presented in the embedded device of the transmission site and used as a codec to extract and encode the feature maps of images, which reduces the coded length. The receiver site applies the decoder module to reconstruct the actual pictures as close to the original image as is feasible using the compact feature codes.

2.2 Proposed network

The traditional convolution neural network applies a convolutional kernel to extract object features of cross-channel and spatial correlations simultaneously. For more complex images, where there is a greater variety of subtler visual characteristics, it is necessary to include more layers of the CNN in order to extract additional features for processing at subsequent layers. This results in more parameters in the network and introduces additional complexities.

The DSC structure can completely segregate the channel and spatial feature extraction processes. The Xception network, constructed by the DSC structure, exhibits superior performance in image classification compared with traditional convolutional networks.[20] It separates the feature extraction processes into two distinct procedures. First, depthwise convolution is used to capture spatial information, and then, pointwise convolution is used to obtain the correlation information of the channels. This network structure reduces the number of weight parameters and speeds up the process compared with the traditional CNN model.

The conventional CNN methodology determines the number of weight parameters by applying Eq. (1). For comparison, it is assumed that the DSC model employs an identical kernel size for depthwise convolution and an equivalent filter number for pointwise convolution in relation to the traditional CNN network. Equations (2) and (3) are employed to calculate the

parameters associated with the depthwise and pointwise convolution structures, respectively. The DSC parameter can be estimated by summing the values obtained from Eqs. (2) and (3). Equation (4) presents the method of calculating the DSC parameters. The DSC structure has fewer weight parameters than the traditional convolution structure, as illustrated in Eq. (5). Consequently, the DSC model can perform better and has a lighter module with faster computation.

$$\text{Traditional CNN parameter sizes} = D_f \cdot D_f \cdot M \cdot N \tag{1}$$

$$\text{Depthwise convolution parameter sizes} = D_f \cdot D_f \cdot M \tag{2}$$

$$\text{Pointwise convolution parameter sizes} = 1 \cdot 1 \cdot M \cdot N \tag{3}$$

$$\text{DSC parameter sizes} = D_f \cdot D_f \cdot M + M \cdot N \tag{4}$$

$$\frac{D_f \cdot D_f \cdot M + M \cdot N}{D_f \cdot D_f \cdot M \cdot N} = \frac{1}{N} + \frac{1}{D_f^2} \tag{5}$$

Here, D_f represents the width and height of the kernel size in the convolutional layer and depthwise convolution. M denotes the number of input channels or depth for the input feature map, and N refers to the number of output channels or depth for the output feature map.

In this paper, we propose a novel network, DSCAE, which is constructed by transferring some DSC components of the Xception network into the CAE structure. The DSCAE mainly provides image compression to enhance transmission efficiency and encryption for the embedded device simultaneously. Figure 2 shows the DSCAE network structure. The CAE is commonly used to rebuild the target image from the input. It extracts features through the encoder and decrypts information via the decoder. The CAE operation is defined by Eqs. (6) and (7). Also, the encoder and decoder processes are calculated as shown in Eq. (8). In this study, we use the mean-squared error (*MSE*) as the loss function for backpropagation during the CAE training stage. The *MSE* loss function is shown in Eq. (9).

$$\psi, \varphi = \arg\min_{\psi,\varphi} \left\| X' - (\varphi \circ \psi) \right\| X^2, \tag{6}$$

$$\psi: X \to F, \varphi: F \to X'. \tag{7}$$

The input vector is denoted as X, the feature vector is F, and the target vector is X'. The extraction of feature vectors from the input images is represented by ψ, while the reconstruction of the feature vectors into target images is represented by φ. The $\varphi \circ \psi$ calculation expression means the total CAE calculation process.

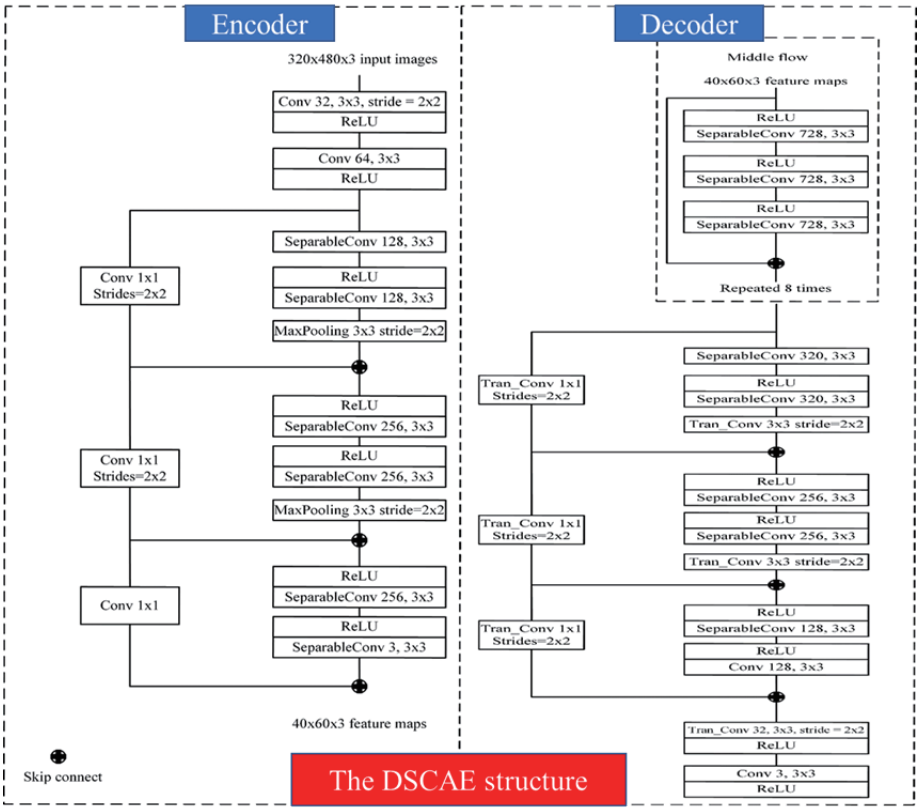


Fig. 2. (Color online) DSCAE structure.

$$\text{If } x \in \mathbb{R}^d = X, \text{If } h \in \mathbb{R}^p = F, \begin{cases} h = \sigma(Wx + b), \\ X' = \sigma'(W'h + b'). \end{cases} \tag{8}$$

$$L(x, X') = \|x - X'\|^2 = \|x - \sigma'(W'(\sigma(Wx + b)) + b'\|^2 \tag{9}$$

Here, x represents the input vector, situated within a d-dimensional real space designated as $\mathbb{R}^d$. Similarly, h denotes the output vector located within a p-dimensional real space denoted as $\mathbb{R}^p$. The encoder activation function is denoted by σ, the decoder activation function by σ', the encoder weight by W, the decoder weight by W', the encoder weighting bias by b, and the decoder weighting bias by b'. The input x is linearly transformed by the weight matrix W and bias b and then activated by σ to produce the output h of the hidden layer, followed by linear transformation by the weight matrix W' and bias b' and activation by σ' to produce the final output X'.

Our proposed model applies the high-efficiency DSC of Xception to the autoencoder module and improves the performance. The optimized Xception Entry flow is imported into the encoder and the output is then quantized into compact feature maps, which are used to compress images for transmission or storage. The decoder part consolidates the Xception Middle flow to the top

layers, which can synthesize finer feature information. Multiple DSC-structured convolution layers with skipped transpose convolutions are then concatenated to restore the predicted image close to the truth.

2.3 Data augmentation method

In this paper, we propose a cropped augmentation image method, which utilizes a particular cropping strategy combined with image processing technologies to enhance the training samples based on the DIV2k dataset. This method ensures the extraction of clear and relevant image regions while maximizing the utilization of available data.

Every 2k pixel image in the DIV2k dataset is cropped to a designated size of 320 × 480 × 3. This size selection ensures consistency in the dimensions of the cropped images and facilitates efficient training and model interpretation. If the original picture is not divisible by the specific width or height size, the last cropped piece is compensated by the preceding section to gather a whole sample. Although this approach may result in some repeated image regions, it ensures that valuable data is well spent and provides additional features for network training. Figure 3 shows the augmentation results.

The original dataset size of 800 images in the DIV2k dataset is expanded to 19448 images by our proposed method, effectively diversifying the training samples. Then, these sample images undergo horizontal and vertical flipping to represent different orientations of objects or scenes to augment more training data. This augmentation expands the training dataset to 77792 instances. This augmentation process introduces variations in the visual appearance of the input images and diversifies the dataset, allowing the model to learn robust and invariant features. This not only improves the network generalization ability but also prevents the overfitting issue, resulting in better overall performance.

3. Experimental Results and Discussion

To assess the efficiency and capacity of the proposed approach, we present the training process results along with two alternative methods used to evaluate performance and conduct a comparative analysis. In this section, the peak signal-to-noise ratio (*PSNR*) is used to estimate the reconstruction fidelity of the output image for each trained model. The objective is to

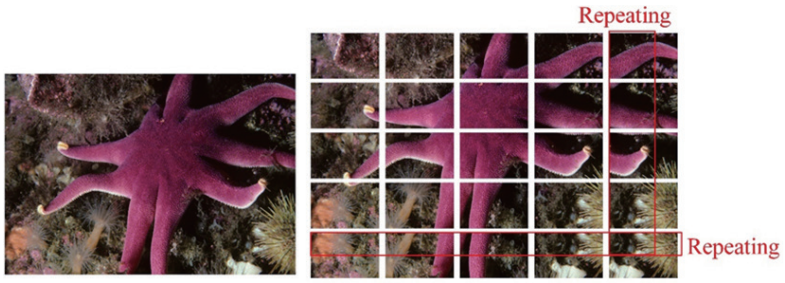


Fig. 3. (Color online) Data augmentation results.

identify the optimal model that can be implemented in an embedded device for image compression and transmission. Wireless transmission of image files generated by the above-mentioned methodologies is undertaken to validate this.

3.1 Given target map to the encoder output training method

Traditionally, U-net is an excellent choice for image reconstruction. This model applies a set of CNN structures with residual skip connections between the encoder and decoder components, demonstrating remarkable performance. However, this work focuses on reducing the image size for efficient transmission. U-net needs to send the output feature maps of many layers in the encoder to the corresponding decoder layer, even if it performs better. To achieve smaller-size image transmission, we intentionally designed all layer outputs of the CAE without residual connection between the encoder and the decoder.

In the first experiment, we utilize two target images to train the CAE network. The input image is resized to one-sixtieth of the original size as the desired output of the encoder. Simultaneously, the original input image is used as the target output for the final decoder. The encoder plays a crucial role in this approach by extracting features from the input image and resizing the input image to match the desired output of the encoder. The encoder architecture comprises 13 convolutional layers, three of which employ a stride of 2 to extract image features and generate smaller feature maps efficiently. Additionally, the encoder structure incorporates the internal skip connection architecture inspired by ResNet to improve feature extraction capabilities. The expected target output of the encoder is acquired by resizing the original image using the binary interpolation method.

The network model is constructed on the basis of the progressive upsampling architecture in the decoder component. It comprises 13 transposed convolutional layers, three of which utilize a stride of 2, to replace the upsampling layer and intelligently reconstruct the small-size feature maps. The schematic flow chart and structure of this experimental model structure are shown in Figs. 4 and 5, respectively. We evaluate the model performance and quality of the reconstructed images using *MSE* and *PSNR*. *MSE* is used as the indicator of the loss function. *PSNR* is the

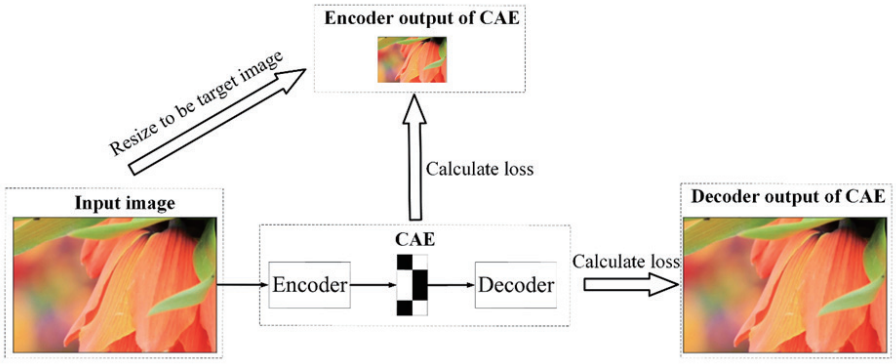


Fig. 4. (Color online) Schematic flow chart of the given target map method.

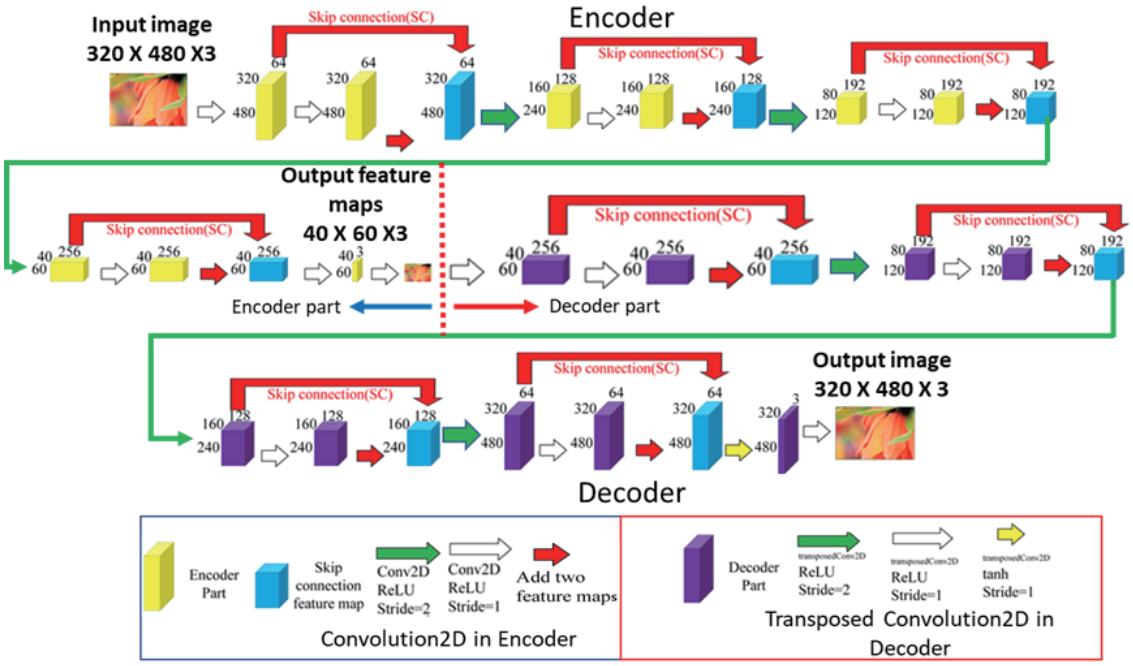


Fig. 5. (Color online) Experimental CAE structure.

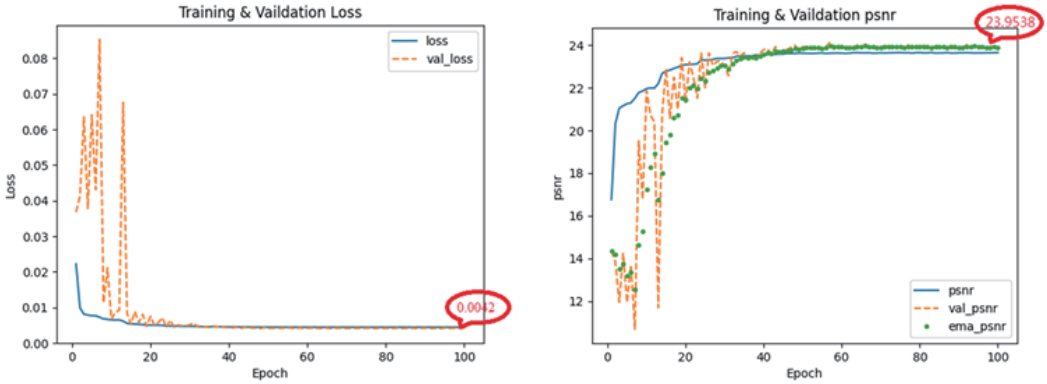
most commonly used metric for evaluating the performance of a reconstruction-type model. The *PSNR* and *MSE* values are obtained using Eqs. (10) and (11). Figure 6 shows the trajectory of the loss function and *PSNR* metrics during training; they become stable after approximately 50 epochs. The loss value gradually converges to a final value of 0.0042. In terms of image quality assessment, *PSNR* achieved a value of 23.9538 dB.

$$PSNR = 10 * \log_{10}\left(\frac{Max^2}{MSE}\right) = 20 * \log_{10}\left(\frac{Max^2}{\sqrt{MSE}}\right) \tag{10}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{11}$$

The variable *Max* represents the maximum value of the digital image, which is 1 in this case since the image has been normalized. Equation (11) for *MSE* introduces the following notations: n denotes the total number of pixel points, Y_i represents the original pixel value, and $\hat{Y}_i$ indicates the predicted pixel value of the image.

The result from the encoder serves as a specific representation of the image while preserving essential features in this model. The designed progressive upsampling architecture enables the decoder to effectively reconstruct the original image from the extracted feature maps. Nevertheless, this method has limitations regarding image reconstruction quality and raises security concerns owing to the visibility of the encoder outputs.



(a) (b)

Fig. 6. (Color online) Training trajectory for the first experiment. (a) Loss function. (b) *PSNR* metrics.

3.2 CAE direct training method

This approach directly trains the CAE network proposed in the last section. During training, the neural network automatically finds the optimum solution for the encoder outputs. These intricate features and patterns are extracted to be feature maps as a transmission object, equivalent to a shrunken image. Figure 7 shows the schematic flow chart of the direct training method. The output of the encoder, in the form of feature maps, is observed. It is notable that these maps, which are used for transmission purposes, appear as an unstructured vision that is invisible to the human eye. Figure 8 illustrates the encoder output feature maps in relation to the corresponding originals. The encoder output can achieve the encryption effect, thereby ensuring secure transmission.

The *MSE* and *PSNR* metric curves for this training method are presented in Fig. 9. While this model demonstrates successful encryption, the utilization of a mere three channels in the encoder output for the middle layer represents a limitation in terms of *PSNR*, with a value of approximately 24.684 dB. To address this limitation and further enhance the performance, the DSCAE method is proposed to improve the CAE network.

3.3 Direct training method for the proposed DSCAE network

The direct training method ensures that the output of the encoder is in ciphertext format to enhance secure transmission. This approach is highly suitable for training our proposed DSCAE method, which leverages the characteristics of the encoder architecture and defines different target outputs in the model layers for improved feature map representation and final image reconstruction. The DSCAE training results for the *MSE* and *PSNR* metric trajectories are shown in Fig. 10. The value curves show gradually stabilizing convergence after the 40th training epoch. Upon the completion of training, the DSCAE model exhibits a remarkably low loss of 0.0011, demonstrating its capacity to match the original input images during reconstruction with high fidelity. Moreover, the corresponding *PSNR* is a high level of 29 dB, indicating a substantial improvement in preserving image details and reducing distortion.

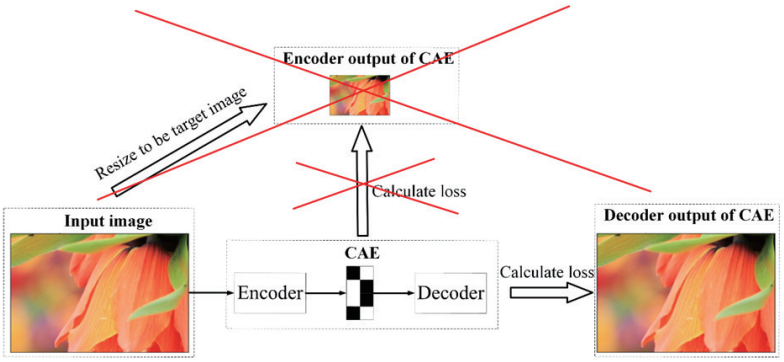
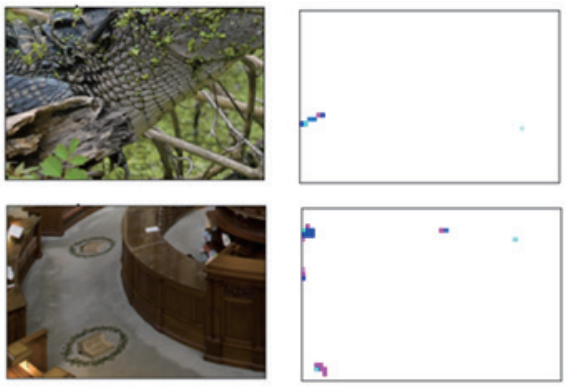
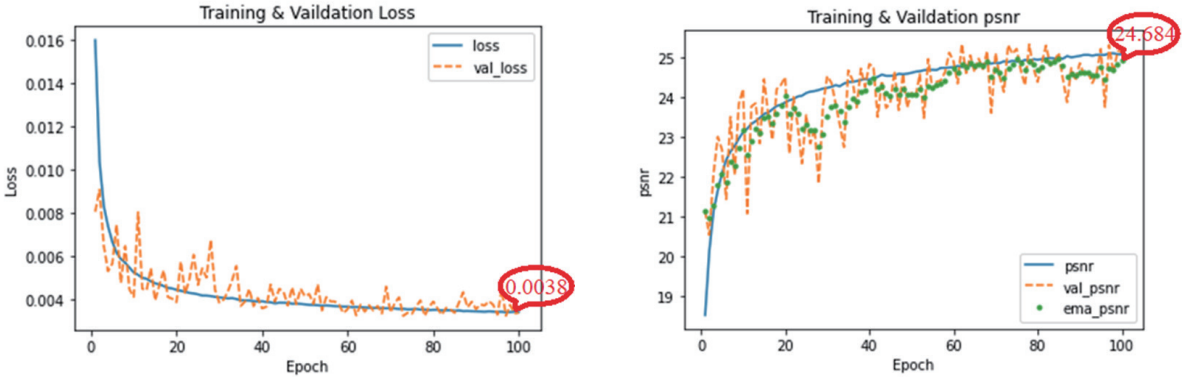


Fig. 7. (Color online) Schematic flow chart of the direct training method.



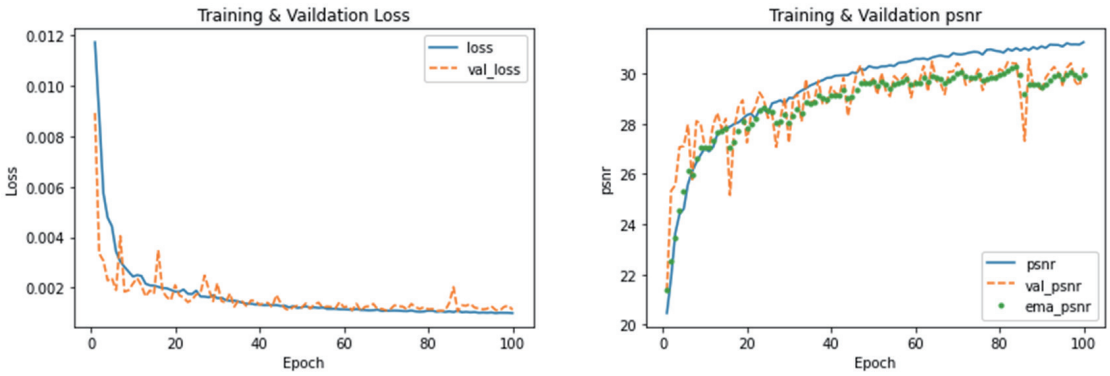
(a) (b)

Fig. 8. (Color online) Feature maps of the encoder output. (a) Original pictures. (b) Feature maps.



(a) (b)

Fig. 9. (Color online) Training trajectory for the second experiment. (a) Loss function. (b) *PSNR* metrics.



(a) (b)

Fig. 10. (Color online) Training trajectory throughout DSCAE training. (a) *MSE* loss function. (b) *PSNR* metrics.

3.4 Comparisons with different models and discussion

Two distinct training methods are employed to train two different types of network models. The efficacy of these methods is evaluated through three experiments in which the encoder compresses images that are then reconstructed by the decoder. Figure 11 presents decoder outcomes for different training models. The resulting graph illustrates that the DSCAE model demonstrates the most favorable performance, exhibiting the highest *PSNR*.

The results of the first experiment indicate that the lack of encryption in the encoder output of the proposed CAE structure results in security vulnerabilities and a low *PSNR*, and produces a final image of poor quality. In the second experiment, the encoder output is observed to contain coded feature maps that are invisible to the naked eye. In this experiment, the direct training method is employed to train the designed CAE structure. However, *PSNR* still needs improvement, and the decoder cannot generate an image of high quality. Our proposed DSCAE model is trained using the direct training method in the final experiment, and the results demonstrate satisfactory *PSNR* values. The output image synthesized by the decoder exhibits a remarkable resemblance to the original. Furthermore, the encoder output is utilized as an encrypted map for transmission and storage, ensuring secure transmission.

To facilitate comparison with the traditional image scaling, the original images are reduced to 1/64 of their original size using bilinear interpolation, with the resulting photos having the same size as the encoder output of DSCAE. These images are then enlarged to their original size using the same interpolation method. The results are shown in Fig. 12. The results of this test demonstrate that the output image produced by the traditional scaling method, after enlargement at the receiving side, exhibits the blurring and distortion of details and a relatively low *PSNR*.

Table 1 shows the different image transmission and processing times for the various methodologies presented in this paper. The configuration of the test environment is illustrated in Fig. 13. Raspberry Pi 4 functions as a server for image transmission and is equipped with 4 GB memory and Raspberry Pi OS version 11. The computer is configured as a client to receive images and utilizes an Intel$^{(R)}$ Core$^{(TM)}$ i5-10400 CPU, 16 GB of RAM, and the Microsoft Windows 11 operating system. A D-Link AX5400 dual-band wireless access point is a packet-

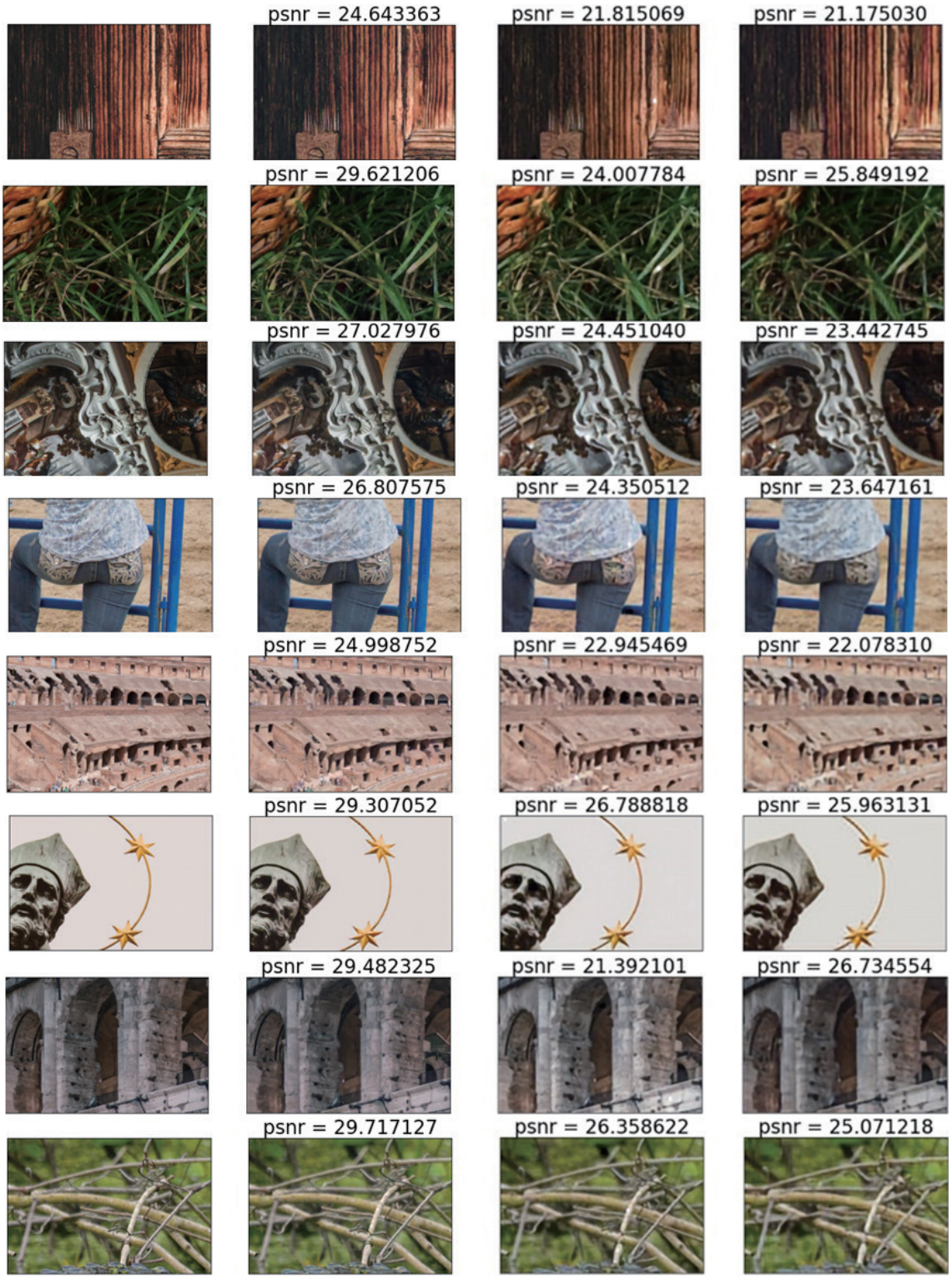


Fig. 11. (Color online) Reconstruction images of decoder output in different training models. (a) Ground truth. (b) DSCAE outputs. (c) Direct training method outputs. (d) Given target method outputs.

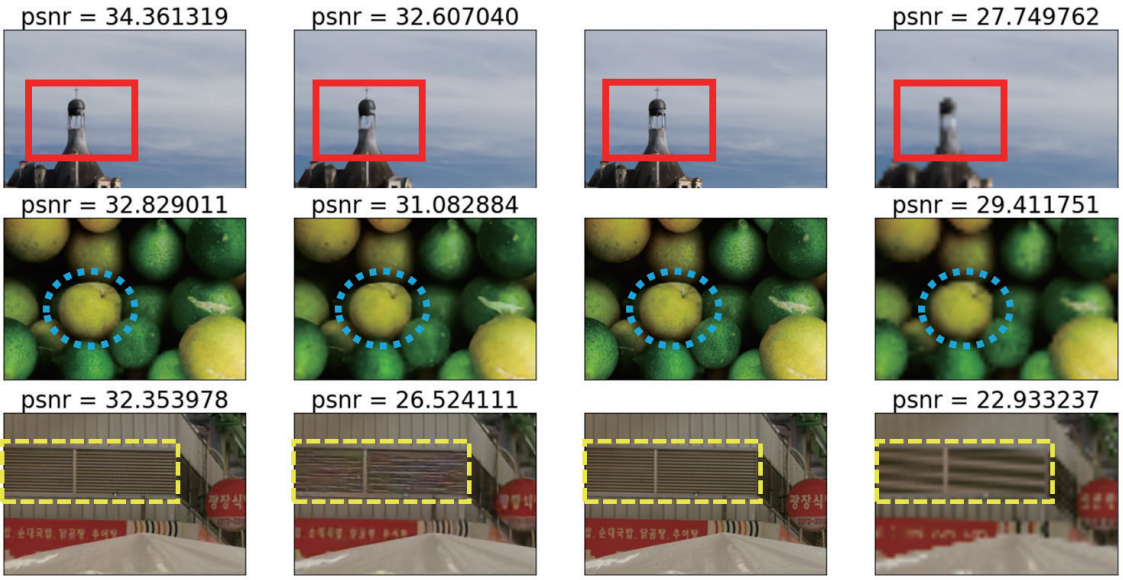


Fig. 12. (Color online) Reconstruction images obtained with various resizing methods. (a) DSCAE method. (b) CAE method. (c) Ground truth. (d) Traditional resizing method.

Table 1
Transmission and processing times.

Processing time	Image processing model			
	Without encoding	CAE model	DSCAE model	Traditional scaling
Encoding (shrinking) time (s)	X	1.8453	0.7847	0.531
Decoding (enlarging) time (s)	X	1.3509	1.0181	0.467
Transmission image scale	320 × 480	40 × 60	40 × 60	40 × 60
Transmission time (s)	0.0320	0.0133	0.0133	0.0141

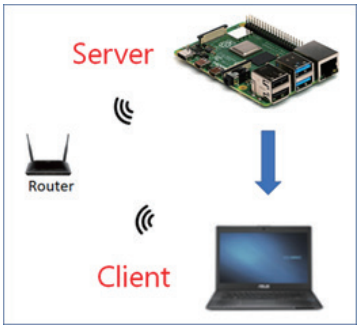


Fig. 13. (Color online) Architecture of image transmission experiment.

forwarding router. A comparison is performed by sending images from the Raspberry Pi module to a computer via wireless transmission. The transfer and processing times displayed in this table have been calculated by averaging over 100 images. The DSCAE model exhibits the

shortest transmission time. Despite the longer processing time for the encoding and decoding of the two neural network models, when the original image is considered together with the encryption processing time, it can be shown that the deep learning model offers a relatively advantageous solution. This is due to the fact that the deep learning model optimizes the transmission bandwidth and ensures the security of the image transmission, which is achieved through an end-to-end procedure.

The experimental results demonstrate that the proposed DSCAE model markedly enhances its performance in image transmission tasks. The DSCAE network shows promise in improving the image reconstruction quality, supporting encrypted transmission, and offering computational efficiency. Consequently, it is a potentially valuable component for enhancing various image processing and transmission systems.

4. Conclusions

We proposed a novel DSCAE network model that incorporates the DSC structure of the Xception network into the CAE model for embedded devices. The proposed model was designed to reduce the number of weight parameters and enhance the efficiency of the network, thereby overcoming the significant limitations of embedded devices concerning computing resources. The feature map of the encoder output was compressed to one-sixtieth of the original image size, resulting in a shorter transmission time and an optimized transmission efficiency. Furthermore, the reconstructed image produced by the decoder showed a high degree of resemblance to the original image and achieved an average *PSNR* of more than 29 dB. The experimental results demonstrated that the DSCAE model achieves superior performance, effectively preserving the quality of transmitted images with encryption transmission. Moreover, the DSCAE model exhibited superior computational efficiency compared with other traditional networks, rendering it suitable for deployment in embedded devices with limited computational resources.

References

1 C. Ren, X. He, Q. Teng, Y. Wu, and T. Q. Nguyen: IEEE Trans. Image Process. **25** (2016) 2168. https://doi.org/10.1109/TIP.2016.2542442
2 M. Shaoshuo, Z. Yanhua, Q. Xiaolan, and J. Yanbing: Proc. 2021 IEEE Int. Conf. Multimedia and Expo (ICME, 2021) 1–6. https://doi.org/10.1109/ICME51207.2021.9428084
3 M. T. Yeh, W. Y. Lo, Y. N. Chung, and H. Y. Cai: Sens. Mater. **36** (2024) 91. https://doi.org/10.18494/SAM4531
4 J. Kumar, and M. Kumar: Proc. 2015 Int. Conf. Advances in Computer Engineering and Applications (ICACA, 2015) 114–118. https://doi.org/10.1109/ICACEA.2015.7164678
5 M. Alzahrani and M. Albinali: Proc. 2021 Int. Conf. Women in Data Science at Taif University (WiDSTaif, 2021) 1–6. https://doi.org/10.1109/WiDSTaif52235.2021.9430242
6 M. I. Patel, S. Suthar, and J. Thakar: Proc. 2019 Int. Conf. Intelligent Computing and Control Systems (ICCS, 2019) 1103–1105. https://doi.org/10.1109/ICCS45141.2019.9065473
7 D. Mishra, S. K. Singh, and R. K. Singh: EURASIP Signal Process. **191** (2022) 108346. https://doi.org/10.1016/j.sigpro.2021.108346
8 L. Cavigelli, P. Hager, and L. Benini: Proc. 2017 Int. Joint Conf. Neural Networks (IJCNN, 2017) 752–759. https://doi.org/10.1109/IJCNN.2017.7965927
9 H. Wang, K. Deng, Y. Duan, M. Yin, Y. Wang, and F. Meng: Proc. Neural Information Process. 2023 (ICONIP, 2023) 37–52. https://doi.org/10.1007/978-981-99-8132-8_4

10 J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston: Proc. The Sixth Int. Conf. Learning Representations (ICLR, 2018) 1–47. https://doi.org/10.48550/arXiv.1802.01436
11 D. Minnen, J. Ballé, and G. D. Toderici: Proc. 32nd Ann. Conf. Neural Information Processing Systems (NIPS, 2018) 10771-10780. https://doi.org/10.48550/arXiv.1809.02736
12 K. Islam, L. M. Dang, S. Lee, and H. Moon: Proc. 2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW, 2021) 1875–1879. https://doi.org/10.1109/CVPRW53098.2021.00209
13 L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo: Proc. 2017 IEEE Int. Conf. Computer Vision (ICCV, 2017) 4836–4845. https://doi.org/10.1109/ICCV.2017.517
14 S. Kudo, S. Orihashi, R. Tanida, and A. Shimizu: Proc. 2019 Picture Coding Symp. (PCS, 2019) 1–5. https://doi.org/10.1109/PCS48520.2019.8954548
15 L. Wu, K. Huang, and H. Shen: Proc. 2020 IEEE Winter Conf. Applications of Computer Vision (WACV, 2020) 2323-2331. https://doi.org/10.1109/WACV45572.2020.9093387
16 R. Wang, Z. Sun, and S. Kamata: Proc. 2020 25th Int. Conf. Pattern Recognition (ICPR, 2021) 9030–9037. https://doi.org/10.1109/ICPR48806.2021.9412655
17 G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell: Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR, 2017) 5435–5443. https://doi.org/10.1109/CVPR.2017.577
18 A. G. Ororbia, A . Mali, J. Wu, S. O'Connell, W. Dreese, D. Miller, and C. L. Giles: Proc 2019 Data Compression Conf. (DCC, 2019) 3–12. https://doi.org/10.1109/DCC.2019.00008
19 Y. Hu, W. Yang, M. Li, and J. Liu: IEEE Trans. Multimedia **21** (2019) 3024. https://doi.org/10.1109/TMM.2019.2920603
20 F. Chollet: Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR, 2017) 1800–1807. https://doi.org/10.1109/CVPR.2017.195