IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Enhanced Dense Space Attention Network for Super-Resolution Construction from Single Input Image

Yoong Khang Ooi, Haidi Ibrahim, Senior Member, IEEE, Muhammad Nasiruddin Mahyuddin, Senior Member, IEEE

School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Pulau Pinang, Malaysia

Corresponding author: Haidi Ibrahim (e-mail: Haidi_ibrahim@ieee.org).

This research is supported by the Universiti Sains Malaysia, under Research University Grant 1001/PELECT/8014052.

ABSTRACT In some applications, such as surveillance and biometrics, image enlargement is required to inspect small details on the image. One of the image enlargement approaches is by using convolutional neural network (CNN)-based super-resolution construction from a single image. The first CNN-based image super-resolution algorithm is the super-resolution CNN (SRCNN) developed in 2014. Since then, many researchers have proposed several versions of CNN-based algorithms for image super-resolution to improve the accuracy or reduce the model's running time. Currently, some algorithms still suffered from the vanishing-gradient problem and relied on a large number of layers. Thus, the motivation of this work is to reduce the vanishing-gradient problem that can improve the accuracy, and at the same time, reduce the running time of the model. In this paper, an enhanced dense space attention network (EDSAN) model is proposed to overcome the problems. The EDSAN model adopted a dense connection and residual network to utilize all the features to correlate the low-level feature and high-level feature as much as possible. Besides, implementing the convolution block attention module (CBAM) layer and multiscale block (MSB) helped reduce the number of layers required to achieve comparable results. The model is evaluated through peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) metrics. EDSAN achieved the most significant improvement, about 1.42% when compared to the CRN model using the Set5 dataset at a scale factor of 3. Compared to the ERN model, EDSAN performed the best, with a 1.22% improvement made when using the Set5 dataset at a scale factor of 4. In terms of overall performance, EDSAN performed very well in all datasets at a scale factor of 2 and 3. In conclusion, EDSAN successfully solves the problems above, and it can be used in different applications such as biometric identification applications and real-time video applications.

INDEX TERMS Computational and artificial intelligence, image processing, image resolution, image quality, machine learning algorithms

I. INTRODUCTION

Image super-resolution is referred to as a process to obtain an image of a high-resolution (HR) image from a low-resolution (LR) image [1]–[4]. Image super-resolution is also known as interpolation, scaling, upsampling, zooming, or enlarging [2]. The purpose of image super-resolution is to obtain a high pixel density and refined details from low-resolution (LR) image(s) that cannot be seen with the naked eye. Image super-resolution is very useful in applications that required recognition or detection purposes.

In the past, many image processing-based techniques have been used in image super-resolution before deep learning-based methods have been introduced. One of the image processing-based techniques is the interpolation-based method. The most common interpolation techniques used for this purpose are linear interpolation [5], bilinear interpolation [6], and bicubic interpolation [7], [8]. These methods are simple, but the high-frequency details of the image are not restored adequately, so more sophisticated insight may be required to recover the image [9], [10].

The reconstruction-based method is another traditional method that includes the sharpening of edge details [11], regularization [12], [13], and deconvolution [14] techniques. Researchers also have used these techniques for image reconstruction. The third traditional reconstruction method is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

the learning-based method that has an advantage over the interpolation-based method by restoring the missing high-frequency details through a relationship between the LR image and the HR image. One of the examples of the learning-based method is the sparse coding method. Yang et al. [15], [16] have developed a sparse coding network to train a joint dictionary to find a highly sparse and over-complete coefficient matrix. The drawbacks of this method are that it requires significant memory usage and a long processing time.

With the development of machine learning technologies, research is now being extended to a new learning-based approach, which is the deep learning method. Deep learning is often applied to image processing applications, such as medical image processing [17], noise removal [18], and object detection [19]. In 2014, the first convolutional neural network (CNN)-based image super-resolution algorithm, which is the super-resolution CNN (SRNCNN), was developed by Dong et al. [20]. Since then, many researchers have developed CNN-based algorithms, inspired by the idea of SRCNN.

The efforts put into the algorithm developments are to solve the model's performance issues, such as gradient-vanishing problems, reducing the number of memory resources consumed, and reducing the running time of the model. From these efforts, various network design techniques have been proposed. These techniques include residual learning, recursive learning, multi-path learning, dense connection, and attention network [21].

However, among the reviewed algorithms, it was observed that a vanishing-gradient problem might contribute to the low accuracy. The vanishing-gradient problem is a kind of deep learning scenario where the gradient of the loss value in the neural network became small. It got diminished gradually during the backpropagation to the initial layer, which results in the network's weights not being updated effectively [22]. Besides, most CNN-based networks obtained better results by adding more layers, relying heavily on computation time, limiting their real-world application, such as camera surveillance systems. Therefore, the problems mentioned above have motivated us to propose a new single input image super-resolution algorithm.

The main contribution in the papers are as follows:

i) To improve the accuracy of the model by adopting dense connection between local wider space attention (LWSA) to utilize all the features to improve the model performance.

ii) Deploying residual learning between local wider dense space attention block (LWDSAB) to utilize low-level information for better correlation learning.

iii) To reduce running time by reducing the number of layers with the help of convolution block attention module (CBAM) and multiscale block (MSB) to utilize channel information better while using the small number of depth layers.

This paper is divided into five sections. Section II presents related works to this research. Then, the proposed method is presented in Section III. The results and discussions are presented in Section IV, whereas Section V concludes our findings.

II. RELATED WORKS

SRCNN was the pioneering use of CNN in image super-resolution construction [20], [23]. SRCNN has been proven to perform better than traditional methods, such as sparse coding-based super-resolution methods. However, Dong et al. [24] later proposed a fast SRCNN (FSRCNN) to overcome the drawback in SRCNN that requires more convolutional layers in non-linear mapping to get a better result. Besides, an efficient sub-pixel CNN (ESPCN) [25] is also being proposed to overcome the complexity issue in SRCNN as it grew quadratically.

Kim et al. [26] have proposed a very deep super-resolution (VDSR) that introduced residual learning between input and output of the final feature mapping layer. Residual learning has shown its good contribution to image reconstruction. The idea of residual learning was then inspired by Lim et al. [27] to create a new algorithm called enhanced deep residual network (EDSR). A multi-connected convolutional network for super-resolution (MCSR) developed by Chu et al. [28] also deployed residual learning in the network. The most recent algorithm, cascading residual network (CRN) [29], was also developed and utilized residual learning. Besides, enhanced residual network (ERN) [29] also introduced residual learning into it. Zou et al. [30] also applied residual learning to both local level and global learning in the model design. Du et al. [31] introduced cross residual learning in expectation-maximization attention cross residual network (EACRN) to allow features from each branch to cross over and fused.

Deep-recursive convolutional network (DRCN) developed by Kim et al. [32] is the first algorithm that applies the recursive method for image super-resolution. DRCN solved the problem of SRCNN that requires many mapping layers to achieve better performance. Deep-recursive residual network (DRRN) [33] then adopts both recursive and residual learning to solve the burdening process in DRCN that requires supervision on every recursion. Han et al. [34] combined the basic design from SRCNN with recursive learning and residual learning, which creates global learning residual learning network (GLRL). Han et al. [35] then later extended their work to modify the network structure in GLRL to develop fast global learning residual learning network (FGLRL). Deep residual dense network [36] modifies the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

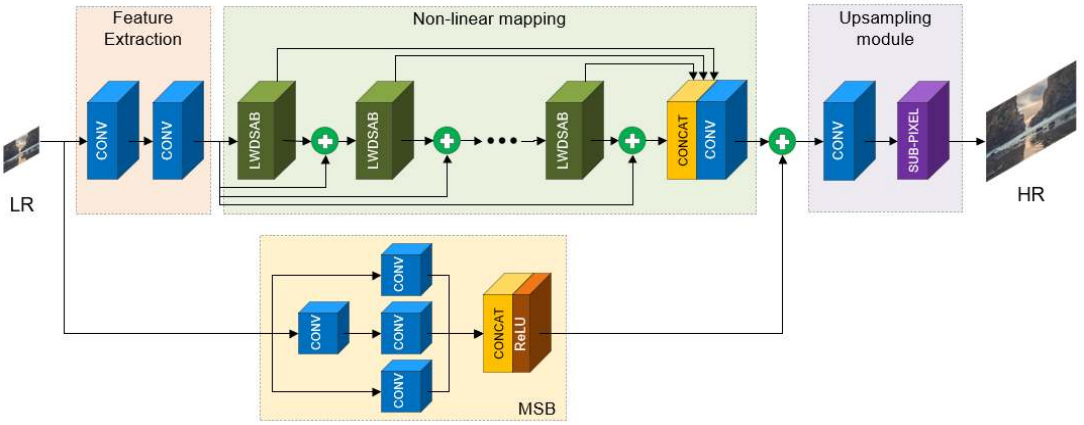


FIGURE 1. Proposed EDSAB Network

network in DRRN by brought all the input features to all inputs of each convolutional layer.

Other than residual learning and recursive learning, dense connection strategy is also found in image super-resolution development. Super-resolution dense connected convolutional network (SRDenseNet) [37] adopted dense connection by sharing the input feature to each input of dense block. The combination of residual learning and dense connection strategy developed in the residual dense network (RDN) [38] outperformed SRDenseNet. Shamsolmoali et al. [39] combined the idea of SRDenseNet and RDN to create a dilated residual dense network (Dilated-RDN) which has better performance than the previous two algorithms. A recent algorithm, the dense space attention network (DSAN) [40], introduces the idea from RDN and convolution block attention module (CBAM) into the network. Liu et al. [41] introduced residual feature aggreation (RFA) network that fused all the features from residual block through dense connection to gather all the information without losing it. A similar technique was also found in the densely connected residual network (DCRN) proposed by Hsu et al. [42].

The multibranch strategy is also being used by researchers in image super-resolution development. The dual-branch convolutional neural network (DBCN) [43] is one of the algorithms that conducted the multibranch strategy. The advantage observed is the reduction of the running time because the complexity of the model is reduced. Single image convolutional neural network (SICNN) [44] is another algorithm that deployed a multibranch strategy. Wang et al. [45] developed an enhanced multi-scale residual network (EMRN) that using a multibranch strategy, and each branch consists of different layer settings to extract features from different aspects.

Among the related works studied, a total of five common strategies used by many researchers can be summarized. The first one is the linear network which is also the most simple network structure. However, this kind of network is not utilizing all the feature information from the input feature. The next strategy used is residual learning. Major advantages observed from this strategy are it helped the model to achieve convergence faster and alleviate the degradation problem caused by deeper network depths. Besides that, recursive learning is also widely used, which deploying shared layer concept to increase the model layer without increasing the number of parameters. This concept reduces memory consumption redundancy. However, the computation time will be increased when the layer number increased. Dense connection, which is another type of strategy, can help solve gradient vanishing problems and reduce the model size without affecting the model performance. Last but not least, dual-branch learning is also another kind of strategy used to reduce the complexity of the model and allow the model to get more information from a different aspect of features.

III. PROPOSED APPROACH

In this section, a new network design is proposed to overcome the problems mentioned above. The proposed network is known as Enhanced Dense Space Attention Network (EDSAB). Inspired by CRN and ERN [29], the local wider residual block and residual learning used in CRN and ERN significantly show the results outperformed EDSR in terms of accuracy as well as low memory consumption and running time. Besides, the convolutional block attention module (CBAM) adopted in DSAN [40] that used the dense connection technique helped the network to give more attention to the useful channel of the features. This provided an advantage of achieved better accuracy while using less layer depth. Fig. 1 showed the design of the proposed EDSAB network structure. It consists of four major components, which are feature extractions, non-linear mapping, multiscale block (MSB), and upsampling modules. In terms of algorithm flow, the pseudocode is present as in Algorithm 1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

Algorithm 1: EDSAN's algorithm

function EDSAN (I_{LR}, r, B, G):
Input: Image, I_{LR} with size $\left(\frac{96}{r}\times\frac{96}{r}\times 3\right)$
Output: Image with size $(96\times 96\times 3)$
$X = H_1(I_{LR})$
$X = H_2(X)$
$X_{temp} = X$
for b = 0 to B **do**
  $X = \text{LWDSAB}(X, G)$
  LWDSAB_collection.append(X)
  $X = X + X_{temp}$
end for
$X = \text{Concat(LWDSAB\_collection)}$
$X = H_{map}(X)$
$X = H_{Output}(X)$
$X = \text{Upsample}(X)$
return X

A. FEATURE EXTRACTION

The feature extraction component extracts the raw features from the LR image. In this component, there will be two convolutional layers, as shown in Fig. 2, that plays the role of feature extraction. Let denoted input image as I_{LR}, the first convoluted feature, F_1 can be expressed in Equation (1), whereas the second convoluted feature, $F_{feature}$ is expressed in Equation (2).

$$F_1 = H_1(I_{LR}) \quad (1)$$

$$F_{feature} = H_2(F_1) \quad (2)$$

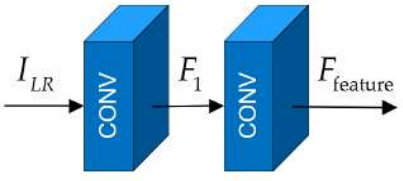


FIGURE 2. Structure of feature extraction component

where H_1 and H_2 refer to convolutional filters of first and second convolutional layers, respectively. Using two convolutional layers instead of a single convolutional layer, will allow information from the images to be extracted more completely and effectively.

B. NON-LINEAR MAPPING

The non-linear mapping component is one of the most important parts of this network. It learns the mapping between LR and HR images and computes each layer's optimized weight value within the non-linear mapping component. A series of local wider dense space attention blocks (LWDSAB) chains together, forming this component. The design of LWDSAB is inspired by the local wider residual block (LWRB) in CRN and ERN [29], and the dense space attention block (DSAB) in DSAN [40]. LWRB is able to help a network to achieve performance as close as EDSR while using fewer memory resources and running time, whereas DSAB helps to pay attention to the useful channel of the features, which help in both accuracy and running time of the model. Therefore, LWDSAB is designed to improve the accuracy of the model and reduce the running time. The LWDSAB, as shown in Fig. 3 or represented with pseudocode in Algorithm 2, is made up of a chained series of local wider space attention modules (LWSA).

Algorithm 2: LWDSAB's algorithm

function LWDSAB (x, G):
Input: Features, x with 64 channels
Output: Features with 64 channels
depth_collection = [x]
for g = 0 to G **do**
  $X = \text{Concat(depth\_collection)}$
  $X = H_{W,m}(X)$
  $X = H_{X,m}(X)$
  $X = \text{CBAM}(X)$
  depth_collection.append(X)
end for
$X = \text{Concat(depth\_collection)}$
$X = H_{ext}(X)$
return X

LWSA consists of a wider channel of convolutional layer followed by a ReLU activation, a narrow channel of convolutional layer, and a convolution block attention module (CBAM). A dense connection is adopted in LWDSAB by concatenating all the features from the preceding layer in the channel layer as the input of the current layer. The concatenated features are convoluted at the end of LWSA before it is added with the input feature through skip connection. Equation (3) is used to express the output feature of LWDSAB, F_{LWDSAB}.

$$F_{LWDSAB} = F_{input} + H_{ext}(F_{ext}) \quad (3)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

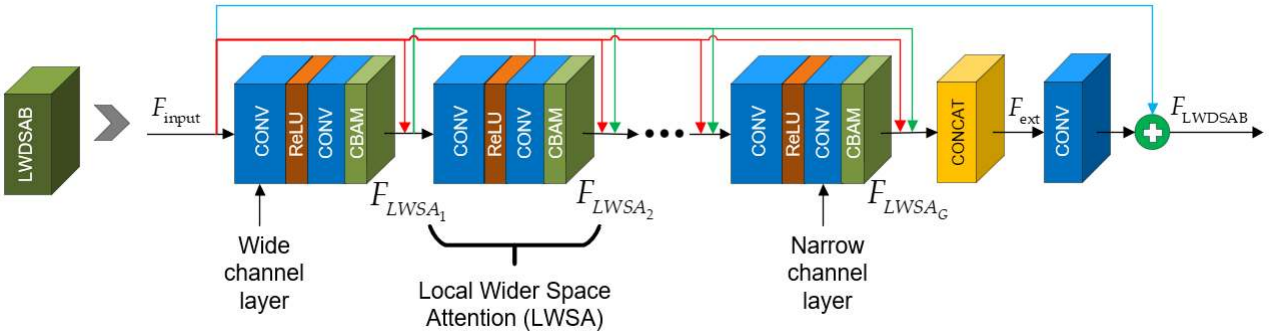


FIGURE 3. Structure of LWDSAB

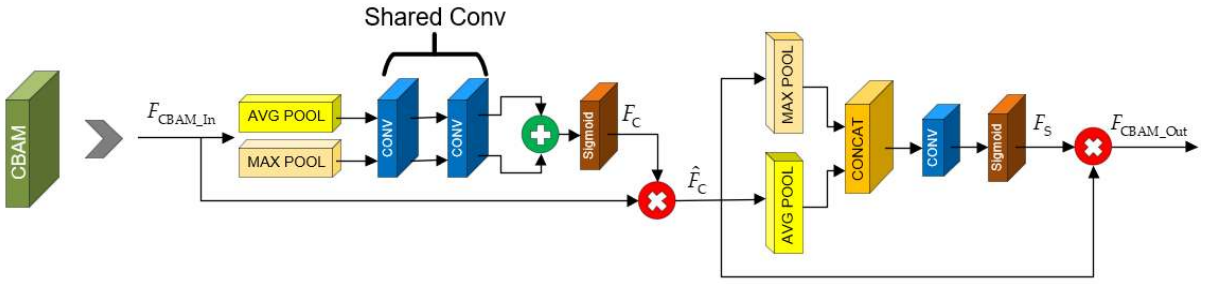


FIGURE 4. Structure of CBAM

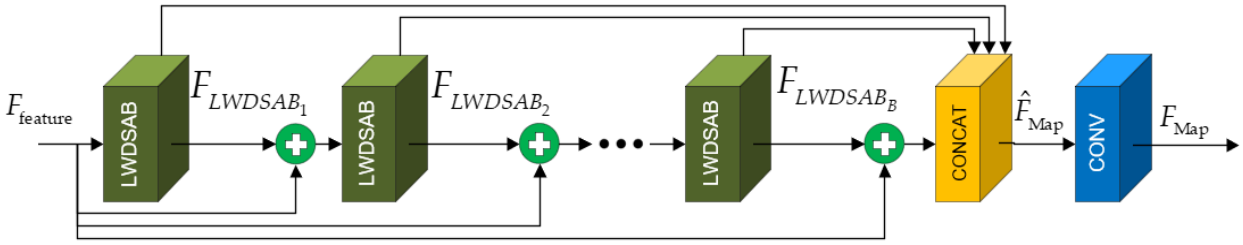


FIGURE 5. Structure of non-linear mapping component

where F_{input} is the input feature feed into the LWDSAB, H_{ext} is the filters for the final concatenated feature and F_{ext} is the concatenated output features from all LWSA which can also be expressed as in Equation (4).

$$F_{\text{ext}} = \left[F_{\text{input}}, F_{LWSA_1}, \cdots, F_{LWSA_G} \right] \tag{4}$$

where F_{LWSA_G} is the output features at *G*-th LWSA, *G* is the number of local wider space attention group (LWSA), and [] is the concatenation of layers in the channel axis. Each of the output features from LWSA can be represented by Equation (5).

$$F_{LWSA_G} = H_{CBAM}\left(H_{X,G}\left(H_{W,G}\left(\left[F_{\text{input}}, F_{LWSA_1}, \cdots, F_{LWSA_{G-1}}\right]\right)\right)\right) \tag{5}$$

where $H_{W,m}$ and $H_{X,m}$ are the filters of wide channel convolutional layer and narrow channel convolutional layer, respectively. H_{CBAM} represents the process of the feature that undergoes CBAM.

The purpose of CBAM, as shown in Fig. 4, is to amplify or shrink features on each channel so that the important features can be extracted faster, such as inner dependency information can be captured while outer dependency information can be reduced. The input feature, $F_{\text{CBAM\_In}}$ is extracted through a global maximum pooling and a global average pooling followed by a convolution process through shared convolutional layers. The convoluted features are added element-wise, followed by a sigmoid activation function, and a channel attention feature, F_{C} is produced. The channel attention feature, F_{C} is undergoing element-wise multiplication with the input feature, $F_{\text{CBAM\_In}}$ and forming feature, $\hat{F}_{\text{C}}$ as expressed in Equation (6).

$$\hat{F}_{\text{C}} = F_{\text{CBAM\_In}} \otimes F_{\text{C}} \tag{6}$$

where $\otimes$ represents the element-wise multiplication or the Kronecker product and F_{C} can be expressed by Equation (7).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

$$F_{\mathrm{C}} = \sigma\left(H_{\mathrm{C1}}\left(H_{\mathrm{C2}}\left(\mathrm{AvgPool}\left(F_{\mathrm{CBAM\_In}}\right)\right)\right) + H_{\mathrm{C1}}\left(H_{\mathrm{C2}}\left(\mathrm{MaxPool}\left(F_{\mathrm{CBAM\_In}}\right)\right)\right)\right) \quad (7)$$

where H_{C1} and H_{C2} are the filter of shared convolutional layers, σ represents the sigmoid activation function, AvgPool represents the average pooling and MaxPool represents the max pooling.

With the feature $\hat{F}_{\mathrm{C}}$ obtained, global average pooling and a global maximum pooling are applied to it, followed by concatenation on each other. A convolutional process is then applied to the fused feature followed by sigmoid activation and the spatial attention feature, F_S is produced. The spatial attention feature, F_S can be expressed as in Equation (8).

$$F_{\mathrm{S}} = \sigma\left(H_S\left(\left[\mathrm{AvgPool}\left(\hat{F}_{\mathrm{C}}\right), \mathrm{MaxPool}\left(\hat{F}_{\mathrm{C}}\right)\right]\right)\right) \quad (8)$$

where H_{S} is the filter of spatial attention. Finally, the output feature of CBAM is obtained by multiply the spatial attention feature, F_{S} with the feature $\hat{F}_{\mathrm{C}}$ element-wise. Equation (9) expresses the formula for the output feature of CBAM.

$$F_{\mathrm{CBAM\_Out}} = \hat{F}_{\mathrm{C}} \otimes F_{\mathrm{S}} \quad (9)$$

Algorithm 3 has represented the pseudocode for CBAM.

Algorithm 3: CBAM's algorithm

function CBAM (x):
Input: Features, x with 64 channels
Output: Features with 64 channels
$X_{\mathrm{mean}} = \mathrm{Mean}(x)$
$X_{\mathrm{mean}} = H_{C1}\left(X_{\mathrm{mean}}\right)$
$X_{\mathrm{mean}} = H_{C2}\left(X_{\mathrm{mean}}\right)$
$X_{\mathrm{max}} = \mathrm{Max}(x)$
$X_{\mathrm{max}} = H_{C1}\left(X_{\mathrm{max}}\right)$
$X_{\mathrm{max}} = H_{C2}\left(X_{\mathrm{max}}\right)$
$X = X_{\mathrm{mean}} + X_{\mathrm{max}}$
$X = \mathrm{Sigmoid}\left(X\right)$
$X = x \bullet X$
$Y_{\mathrm{mean}} = \mathrm{Mean}(X)$
$Y_{\mathrm{max}} = \mathrm{Max}(X)$
$Y = \mathrm{Concat}\left(Y_{\mathrm{mean}}, Y_{\mathrm{max}}\right)$
$Y = H_{\mathrm{S}}\left(Y\right)$
$Y = X \bullet Y$
return Y

The non-linear mapping component adopted both residual learning and dense connection techniques, as shown in Fig. 5. Residual learning combines the extracted feature from the feature extraction component to each of the output features of LWDSAB through skip connection. Residual learning can speed up the convergence of the loss function since low-level and high-level features are highly correlated. The dense connection concatenates all the features from preceding layers so that it utilizes all the features for accuracy improvement before it is being upsampled. The feature of the entire non-linear mapping component, denoted as F_{Map} is expressed as in Equation (10).

$$F_{\mathrm{Map}} = H_{\mathrm{Map}}\left(\hat{F}_{\mathrm{Map}}\right) \quad (10)$$

where $\hat{F}_{\mathrm{Map}}$ is the concatenated output features from all LWDSAB as shown in Equation (11).

$$\hat{F}_{\mathrm{Map}} = \left[F_{LWDSAB_1}, F_{LWDSAB_2}, \ldots, F_{LWDSAB_{B-1}}, F_{LWDSAB_B} + F_{\mathrm{feature}}\right] \quad (11)$$

where B is the number of local wider dense space attention blocks (LWDSAB).

C. MULTISCALE BLOCK (MSB)

Multiscale block (MSB) is used to exploit the low-level information from the input LR image. MSB extracts the feature from the LR image through three convolutional layers with different kernel sizes, as shown in Fig. 6. This allows different aspects of features to be extracted from the same input so that it extends the capability of the model to correlate input and output features.

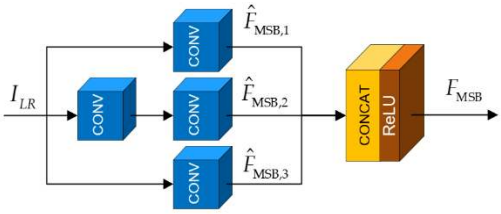


FIGURE 6. Structure of multiscale block (MSB) component

By denoting $\hat{F}_{\mathrm{MSB},1}$, $\hat{F}_{\mathrm{MSB},2}$ and $\hat{F}_{\mathrm{MSB},3}$ as an extracted feature from different kernels, their equations are expressed as in Equations (12), (13) and (14).

$$\hat{F}_{\mathrm{MSB},1} = H_{\mathrm{MSB},1}\left(I_{LR}\right) \quad (12)$$

$$\hat{F}_{\mathrm{MSB},2} = H_{\mathrm{MSB},2b}\left(H_{\mathrm{MSB},2a}\left(I_{LR}\right)\right) \quad (13)$$

$$\hat{F}_{\mathrm{MSB},3} = H_{\mathrm{MSB},3}\left(I_{LR}\right) \quad (14)$$

where $H_{\mathrm{MSB},1}$, $H_{\mathrm{MSB},2a}$, $H_{\mathrm{MSB},2b}$ and $H_{\mathrm{MSB},3}$ are the filter with different sizes and channels. The output of MSB, as shown in Equation (15) is obtained by concatenating all extracted features through channel level before the ReLU activation function is applied to it. Algorithm 4 presents the pseudocode for MSB.

$$F_{\mathrm{MSB}} = \mathrm{ReLU}\left(\left[\hat{F}_{\mathrm{MSB},1}, \hat{F}_{\mathrm{MSB},2}, \hat{F}_{\mathrm{MSB},3}\right]\right) \quad (15)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

Algorithm 4: MSB's algorithm

function MSB (x):
Input: Features, x with 64 channels
Output: Features with 64 channels
$X_1 = H_{\text{MSB},1}(x)$
$X_2 = H_{\text{MSB},2a}(x)$
$X_2 = H_{\text{MSB},2b}(X_2)$
$X_3 = H_{\text{MSB},3}(x)$
$X = \text{Concat}(X_1, X_2, X_3)$
return X

D. UPSAMPLING MODULE

Upsampling module is used to reconstruct the HR image from the processed features. In this project, sub-pixel convolution is used as an upsampling module because preceding works [25], [40], [46] showed that it creates better performance than the traditional upsampling method, such as bicubic interpolation. Besides, it performed faster than bicubic interpolation in running time, and less computation cost is required.

Before upsampling take place, the feature from non-linear mapping, F_{Map} and feature from MSB, F_{MSB} are added through skip connection and the resulting output feature F_{Output} is fed into the upsampling module. The resulting feature is then brought into the convolutional layer to create a new feature, $\hat{F}_{\text{Output}}$ as shown in Fig. 7, that has the shape of width W, height H, and r^2C channels through the formula of Equation (16).

$$\hat{F}_{\text{Output}} = H_{\text{Output}}\left(F_{\text{Output}}\right) \tag{16}$$

where H_{Output} is the filter of the convolutional layer applied to F_{Output}. The sub-pixel layer will then transformed the feature into the shape of width rW, height rH and C channels through a process called pixel shuffle. Mathematically, the operation of pixel shuffle, PS can be described using Equation (17).

$$PS\left(\hat{F}_{\text{Output}}\right)_{x,y,c} = \hat{F}_{\text{Output}\left[\frac{x}{r},\frac{y}{r},cr\bmod(y,r)+c\bmod(x,r)\right]} \tag{17}$$

where x, y and c are pixel coordinate of the image, mod is the modulus operation.

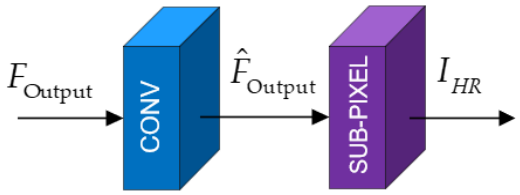


FIGURE 7. Sub-pixel Upsampling Components

E. EXPERIMENT DESIGN

The experiment design includes the process details of datasets used for training and testing and the data preprocessing step. Besides, the type of loss function, optimizer, and evaluation metrics for both the training and testing process is also described. Last but not least, the hyperparameters settings that will be used for the training process will be defined.

1) DATASETS PREPARATION

In this project, the DIV2K dataset is chosen as a training dataset [47] because this dataset has been widely used by many researchers in recent research due to its high resolution and large quantity. DIV2K consists of 800 training images and 200 validation images that including different categories, such as environment, flora, fauna, handmade object, people, and scenery. All the images are color images in png format. They have an average resolution of 1972×1437 pixels.

Since the image in the dataset is huge and memory-consuming when the full image is feeding into the model for training, each of the images has to be pre-processed. Each image is being cropped randomly into many small patches of size 96 × 96 pixels. The patch size is following the standard size used by ERN and CRN models [29]. A total of 10 patches are cropped from each image randomly without overlapping each other. Each of the patch images is further augmented through rotation and flipping to make the network more robust to different cases. Rotation of 90° and horizontal flipping are applied on every patch image, resulting in two additional patch images for each. As a result, a total of (1+2)×10×800=24000 patch images are available now.

The corresponding LR images are obtained by resizing the patch images to a different scale, such as 2×, 3×, and 4× using bicubic interpolation. With limited hardware resources of the graphical processing unit (GPU) that are available, the training process not able to include all patch images that had been prepared. Therefore, a total of 14000 pairs of HR and LR patch images are randomly selected as the training dataset.

For state-of-art comparison purposes, the testing datasets that will be used are Set5 [48], Set14 [49], Urban100 [50], and BSDS100 [51], which are used by most of the algorithms, including CRN [29], ERN [29], DSAN [40], and DRDN [36]. The same preparation flow is applied to the testing dataset. For testing datasets, all images in question will be used for evaluation since the hardware resource is capable of handling a designated quantity of datasets. Set5 and Set14 datasets contain 150 images and 420 images, respectively, whereas both Urban100 and BSDS100 contain 3000 images each. Therefore, a total of 6570 images are used as the testing images.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

2) LOSS FUNCTION, OPTIMIZER AND EVALUATION METRICS

The loss value will be used by the optimizer for parameter tuning. An optimizer is an algorithm used to change the weight values of each filter layer so that the weight is able to achieve optimum value for helping the model achieve the best possible accuracy. For a fair comparison with the previous work, this project adopted the same optimizer, which is the ADAM optimizer. The parameter settings used for the optimizer are also set the same as in previous work [29], [40]. TABLE I shows the parameter settings for the ADAM optimizer.

Evaluation metrics are used to evaluate the performance of the model in terms of its accuracy of image reconstruction. For the state-of-art comparison purpose, the peak signal-to-noise ratio (PSNR) and structural similarities (SSIM) will be used as the evaluation indicator. The formula of the PSNR value can be expressed as in Equation (18).

$$PSNR = 10\log_{10}\left(\frac{M^2}{\frac{1}{N}\sum_{i=n}^{N}\left[\frac{1}{hwc}\sum_{i,j,k}\left(\hat{I}_{HR,n(i,j,k)} - I_{HR,n(i,j,k)}\right)\right]^2}\right) \quad (18)$$

where M was the maximum pixel value (equals to 255 when 8-bit pixel value was used), N was the number of images, h is the height of image, w is the width of image and c is the channel of the image.

TABLE I
PARAMETER SETTINGS FOR THE ADAM OPTIMIZER

Parameters	Value
Learning rate, LR	1×10^{-04}
Exponential decay rate for the 1st-moment estimates, β_1	0.9
Exponential decay rate for the 2nd-moment estimates, β_2	0.999
Constant for numerical stability, ε	1×10^{-08}
Learning scheduler	Reduce LR by half for every 50 epochs

Another measure that will be used is the structural similarity index measure (SSIM). The value of SSIM can be obtained through Equation (19).

$$\mathrm{SSIM}\left(I_{HR},\hat{I}_{HR}\right) = \left[C_l\left(I_{HR},\hat{I}_{HR}\right)\right]^{\alpha}\left[C_c\left(I_{HR},\hat{I}_{HR}\right)\right]^{\beta}\left[C_s\left(I_{HR},\hat{I}_{HR}\right)\right]^{\gamma} \quad (19)$$

where α, β and γ are the control parameters for adjusting the importance of luminance, contrast and structure respectively. $C_l\left(I_{HR},\hat{I}_{HR}\right)$, $C_c\left(I_{HR},\hat{I}_{HR}\right)$ and $C_s\left(I_{HR},\hat{I}_{HR}\right)$ are the luminance, contrast and structures measurement respectively.

3) HYPERPARAMETERS

Hyperparameters such as the number of filters and kernel size are applied differently to each of the filter layers in the model. As for filter strides and padding mode, they are set to [1, 1] and 'same'. TABLE II summarizes the hyperparameters settings for all filter layers in the designed model.

All the model training process was performed through the Kaggle kernel service which using Nvidia P100 hardware. Each model will be run for 300 epochs at a batch size of 14. By denoting the total number of local wider dense space attention blocks (LWDSAB) as B and the total number of local wider space attention modules (LWSA) as G, this project will conduct a few experiments to investigate the relationship between B and G towards PSNR value. As an overview of the entire experiment setup, Fig. 8 illustrated the experiment flow of this project.

TABLE II
PARAMETER SETTINGS FOR THE ADAM OPTIMIZER

Components	Parts	Layers	Number of filters	Kernel size
Feature Extraction	-	H_1	64	3 × 3
		H_2	64	3 × 3
Non-linear Mapping	LWDSAB	$H_{W,m}$	512	3 × 3
		$H_{X,m}$	64	3 × 3
		H_{ext}	64	3 × 3
	CBAM	H_{C1}	32	1 × 1
		H_{C2}	64	1 × 1
		H_S	1	7 × 7
	-	H_{Map}	64	3 × 3
Multiscale Block (MSB)	-	$H_{MSB,1}$	24	3 × 3
		$H_{MSB,2a}$	32	1 × 1
		$H_{MSB,2b}$	24	3 × 3
		$H_{MSB,3}$	16	1 × 1
Upsampling Module	-	H_{Output}	$64 \times r^2$	3 × 3

V. RESULTS & DISCUSSIONS

In this section, the results of this project will be discussed. There are several subsections to be discussed in this section. First, the effect on the performance and running time of the model at different model settings will be studied. Next, the quantitative results corresponding to the accuracy, running time, and the number of parameters will be discussed. Last but not least, the qualitative results through visual observation will be reviewed.

A. MODEL SETTINGS

In this subsection, two different experiments are conducted to observe the effect of different settings in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

model on its performance and running time. The first experiment will investigate the effect of network depth on the performance and running time of the model. Second, another experiment is conducted to study the effect of deploying MSB and CBAM on the performance and running time of the model.

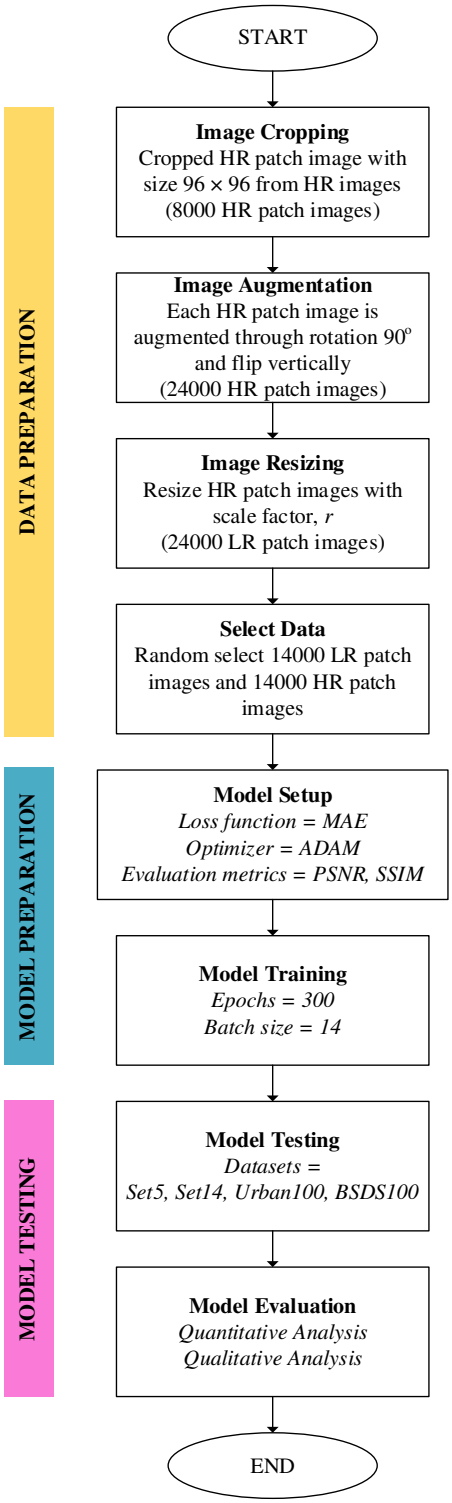


FIGURE 8. Experiment flow of this work

1) EFFECT OF NETWORK DEPTH ON PERFORMANCE & RUNNING TIME

A few experiments had been conducted to investigate the effect of variation in the number of local wider dense space attention blocks (LWDSAB), B and the number of local wider space attention groups (LWSA), G towards the PSNR value. These experiments are conducted with a scale factor of 3. The first experiment was conducted by fixing the value of G at 3 and varying the value of B. Fig. 9 showed the result of the training process for 300 epochs by varying B value at constant G value. The results showed that when the number of groups increases, the PSNR value can be increased. Another experiment is conducted by fixing the value of B at 3 and varying the value of G. Fig. 10 showed that across 300 epochs, the higher the G value, the better the PSNR value. For a fair comparison with CRN and ERN [29], B and G values follow the same settings as tested in CRN and ERN model.

The results obtained from the two experiments are combined into one graph as shown in Fig. 11 to see the relationship between PSNR with the variation of B and G. From the observation in Fig. 11, the model with $B = 3$, $G = 4$ and $B = 4$, $G = 3$ had a higher PSNR value. The lower PSNR value is obtained at $B = 2$, $G = 3$ and $B = 3$, $G = 2$. As for $B = 3$, and $G = 3$, it possesses the average performance compared to the other. The difference in PSNR value can be attributed due to the network depth being introduced. The network depth of the model can be obtained from Equation (20). The bigger the network depth, the higher the PSNR value. This is because when network depth is larger, there will be more filter layers to store more weight. Therefore, more weight values can be more varied to correlate input image and output image as closely as possible.

$$\text{Network depth} = B \times G \tag{20}$$

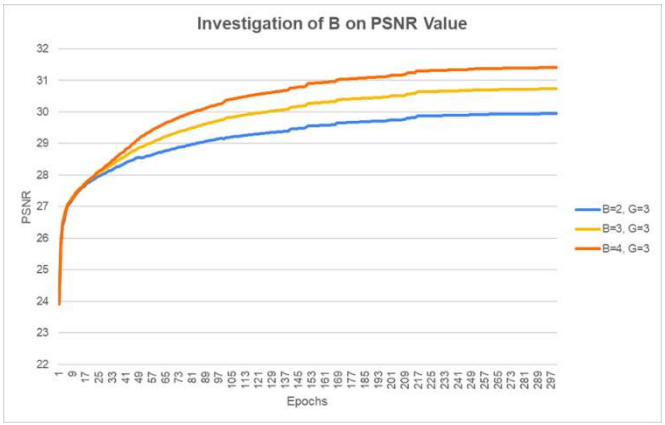


FIGURE 9. Training results on variation of LWDSAB number, *B* towards PSNR value

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

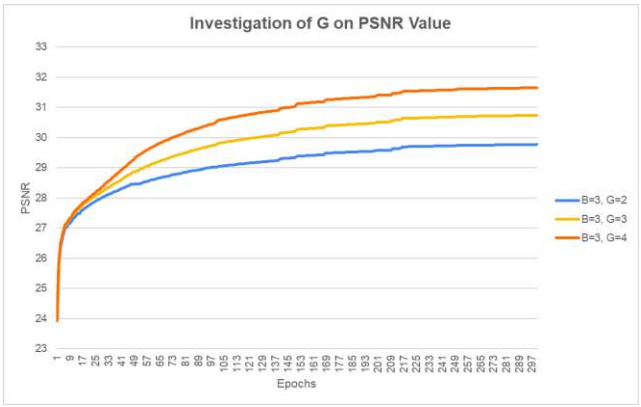


FIGURE 10. Training results on variation of LWSA number, *G* towards PSNR value

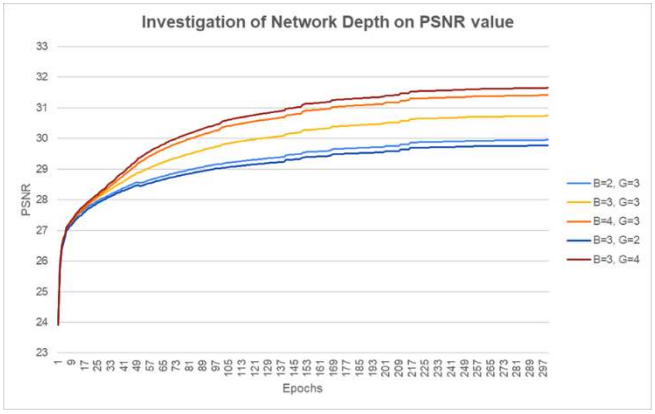


FIGURE 11. Training results of various network depth towards PSNR value

By comparing $B = 3$, $G = 4$ and $B = 4$, $G = 3$, both of them are having the same network depth. However, $B = 3$, $G = 4$ has a better PSNR value than $B = 4$, $G = 3$. In a single LWDSAB, many features being utilized to each LWSA through skip connection, and the input feature of each LWDSAB is added to the end output of LWSA through residual learning. However, only residual learning is adopted in between LWDSAB. Therefore, the model with a high number of G will contain more features to be extracted at the end of the LWSA. This explains that $B = 3$, $G = 4$ obtain a higher PSNR value since the model has a larger number of parameters compared to $B = 4$, $G = 3$. The same observation is also obtained from $B = 2$, $G = 3$ and $B = 3$, $G = 2$ that has the same network depth, but the model with higher G performed better.

Although using a high number of B and G gives a better result, it will have an impact on the memory resources and running time. Table III showed the total number of parameters and the running time required for each image corresponding to B and G. It is observed that the model with $B = 3$, $G = 4$ had the highest PSNR value among the 5 models consumes a lot of memory resource and requires long running time. Therefore, this information is very useful for different application cases that prioritize high accuracy or fast running time.

TABLE III
NUMBER OF PARAMETERS AND RUNNING TIME WITH RESPECT TO *B* AND *G*

B	*G*	Number of Parameters (M)	Average Running Time (s)
3	4	13.03	0.128
4	3	11.30	0.112
3	3	8.62	0.074
2	3	5.86	0.070
3	2	5.05	0.065

2) EFFECT OF MSB AND CBAM ON PERFORMANCE AND RUNNING TIME

Fig. 12 demonstrates the effect of MSB and CBAM on the accuracy and running time. The first model is built according to the proposed design with both B and G set to 3. The second model is built by removing MSB and CBAM from the proposed design network with both B and G also set to 3. The last model is also created without MSB and CBAM but with a more depth layer, in which both B and G are set to 4.

When the MSB and CBAM are removed from the model at $B = 3$ and $G = 3$, the PSNR value dropped as indicated by the yellow line in Fig. 12. From this observation, it is noticeable that both MSB and CBAM play an important role in PSNR improvement. CBAM works to help the model to pay more attention to the useful channel of the feature. Therefore, it allows the model to obtain optimal weight easily. Besides, the use of MSB is to extract input features from different scales so that the low-level feature and high-level feature can correlate with each other from a different perspective.

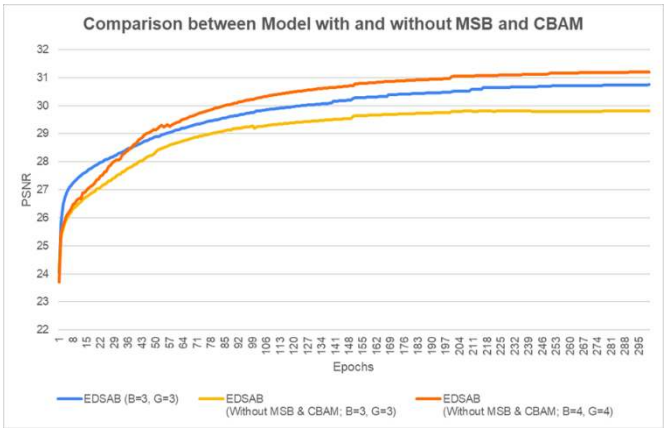


FIGURE 12. Training results of various network depth towards PSNR value

The model without MSB and CBAM has to extend its depth layer to obtain a result close to the proposed design network. The model with $B = 4$ and $G = 4$ is found to have close results to the proposed design model. However, this has an impact on the memory resources and running time. Table IV displays the total number of parameters featured by the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

model and the respective running time required to process an image. The EDSAN without MSB and CBAM but deeper depth will have about twice the amount of parameter compared to that of the originally proposed EDSAN network. In other words, about double of memory resources are required for the network. Besides, the running time also increases about 34.48% when a deeper network is used. In conclusion, MSB and CBAM not only help in PSNR but also reduce layer quantity to achieve a better PSNR.

TABLE IV
NUMBER OF PARAMETERS AND RUNNING TIME WITH RESPECT TO EACH MODEL

Model	Number of Parameters (M)	Average Running Time (s)
EDSAN (B=3, G=3)	8.62	0.074
EDSAN (without MSB & CBAM; B=3, G=3)	8.57	0.070
EDSAN (without MSB & CBAM; B=4, G=4)	17.20	0.117

B. QUANTITATIVE ANALYSIS

Quantitative analysis is the analysis of result that obtained through a mathematical formula. The PSNR value and SSIM value will be compared between the EDSAN model and with other models in this subsection. The quantitative analysis will be interpreted in terms of the percentage of model improvement to observe the significant value of the improvement. The PSNR value and SSIM value will be compared at three different scale factors, which are ×2, ×3 and ×4. The algorithms that will be used for comparison are CRN, ERN, DSAN, and DRDN because the idea of the new proposed algorithm is inspired by their network design.

1) ACCURACY

For state-of-art comparison purpose, these algorithms are evaluated with four different datasets, which are Set5 [50], Set14 [50], Urban100 [47] and BSDS100 [52]. Table V shows the results obtained for average PSNR value and average SSIM value corresponding to each model and testing datasets at different scale factors. The text in Table V with red color indicates the best performance achieved, whereas the blue color indicates the second-best performance that had been achieved. As for interpretation, the percentage of PSNR and SSIM improvement achieved by the EDSAN model relative to other individual models can be expressed as in Equation (21) and Equation (22), respectively.

$$\%\ \text{PSNR Improvement} = \frac{\text{PSNR}_{EDSAN} - \text{PSNR}_{X}}{\text{PSNR}_{X}} \times 100\% \quad (21)$$

$$\%\ \text{SSIM Improvement} = \frac{\text{SSIM}_{EDSAN} - \text{SSIM}_{X}}{\text{SSIM}_{X}} \times 100\% \quad (22)$$

TABLE V
QUANTITATIVE EVALUATION BETWEEN MODEL

Scale	Model	Parameters (M)	Set5		Set14		Urban100		BSDS100	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
2	CRN [29]	9.47	34.89	0.927	29.62	0.856	28.86	0.880	28.56	0.845
	ERN [29]	9.48	**34.89**	**0.928**	**29.67**	**0.856**	**28.97**	**0.882**	**28.66**	**0.847**
	DSAN [40]	3.19	34.11	0.922	29.23	0.855	28.71	0.878	28.51	0.844
	DRDN [36]	5.79	32.95	0.914	28.79	0.856	26.07	0.832	28.31	0.847
	EDSAN	8.44	**35.25**	**0.930**	**29.80**	**0.860**	**28.98**	**0.882**	**28.81**	**0.852**
3	CRN [29]	9.48	31.04	0.872	26.17	0.740	24.75	0.756	25.29	0.706
	ERN [29]	9.49	**31.24**	**0.874**	**26.30**	**0.745**	**24.86**	**0.759**	**25.39**	**0.709**
	DSAN [40]	3.20	30.77	0.867	25.92	0.738	24.73	0.754	25.25	0.705
	DRDN [36]	5.79	29.18	0.831	25.74	0.733	23.22	0.692	25.11	0.709
	EDSAN	8.62	**31.48**	**0.879**	**26.34**	**0.751**	**24.83**	**0.760**	**25.44**	**0.716**
4	CRN [29]	9.51	28.58	0.811	24.33	0.655	**22.70**	0.650	**23.76**	0.606
	ERN [29]	9.52	**28.63**	**0.812**	**24.43**	**0.657**	**22.79**	**0.654**	**23.83**	**0.607**
	DSAN [40]	3.19	28.42	0.808	24.27	0.653	22.66	0.648	23.72	0.605
	DRDN [36]	5.79	27.35	0.775	24.21	0.649	21.80	0.593	22.91	0.613
	EDSAN	8.58	**29.80**	**0.823**	**24.34**	**0.661**	22.68	**0.654**	23.75	**0.613**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

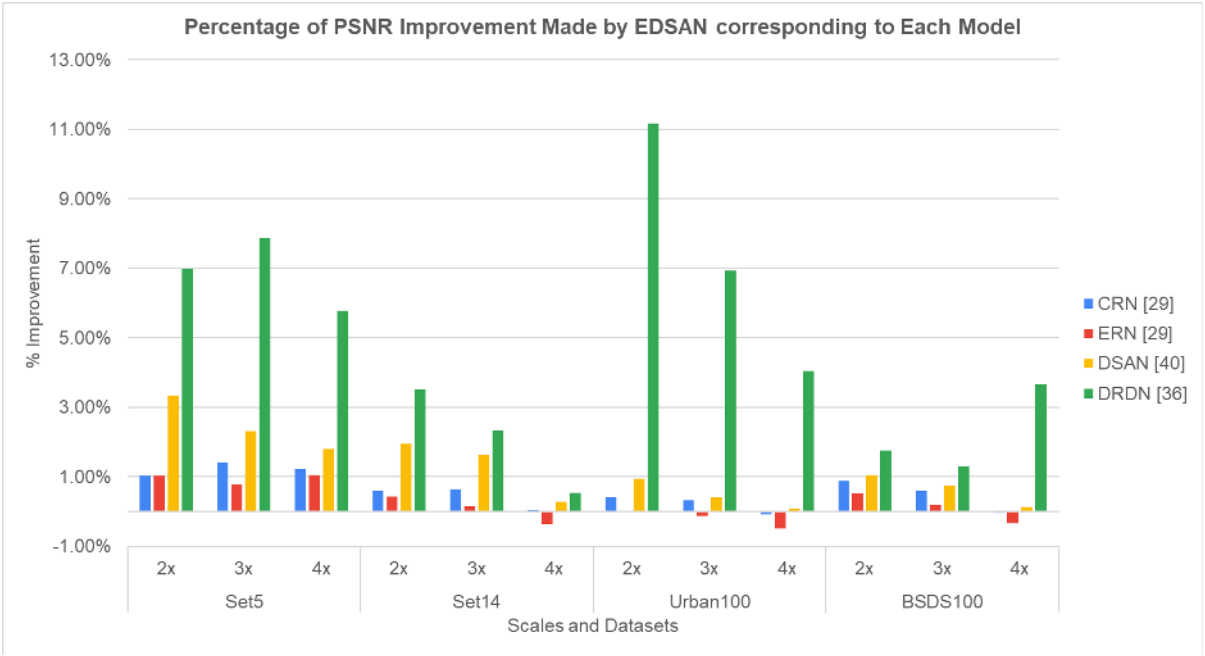


FIGURE 13. Percentage of PSNR Improvement for EDSAN model as compared to CRN [29], ERN [29], DSAN [40] and DRDN [36]

where X represents the model to be compared. The percentage of improvement can be illustrated as how much does accuracy improvement can be achieved by EDSAN relative to the model that will be compared.

Fig. 13 and Fig. 14 illustrate the percentage of PSNR and SSIM improvement achieved by EDSAN relative to other models and different datasets at different scale factors, respectively, which had been discussed earlier. For the positive bar chart facing upwards, it indicates the EDSAN model shows an improvement compared to the relative model. The longer the positive bar chart, the more significant is the amount of improvement made by the EDSAN relative to the model. However, the negative bar chart that facing downwards indicates the relative model is outperformed the EDSAN model.

By looking at the PSNR value, the EDSAN model outperformed the DRDN model most significantly. The highest improvement achieved was 11.16% on the Urban100 dataset at a scale factor of 2, whereas the smallest improvement achieved was 0.54% on Set14 at a scale factor of 4. When compared to the DSAN model, the EDSAN model also shows an improvement result, at which the most significant improvement observed is on Set5 and Set14 datasets. For Urban100 and BSDS100 datasets, the improvement was also observed but not as much as in Set5 and Set14 datasets. Although the EDSAN model does not obtain the top performance for all the datasets at all scales, the positive result is higher than the negative result. Therefore, the overall performance of the EDSAN model is still the best among all the compared models.

As compared with the CRN model, except for Urban100 and BSDS100 data at a scale factor of 4, the EDSAN model shows improvement results on different datasets at different scales. The CRN model outperforms the EDSAN model by as much as 0.09% and 0.04% on Urban100 and BSDS100 datasets, respectively. For the comparison with the ERN model, except on Set14, Urban100, and BSDS100 datasets at a scale factor of 4, the EDSAN model also shows improved results. The highest improvement achieved was 1.05% on the Set5 dataset at a scale factor of 4, whereas the lowest achievement was 0.03% on the Urban100 dataset at a scale factor of 2.

By looking at the SSIM value, the EDSAN model shows an overall improvement in all datasets at all scales. Among the results observed, the highest improvement achieved by the EDSAN model was about 10.29% when it is compared with the Urban100 dataset at a scale factor of 4. There is no lowest improvement made, but equal value with the relative is observed. The ERN model on the Urban100 dataset at a scale factor of 2 and 4, and the DRDN model on the BSDS100 dataset at a scale factor of 4 are having the same SSIM value as that of the EDSAN model.

From the perspective view for the effect of image sizes changes on the model performance, the accuracy of image reconstruction is dropped when the image size is reduced, which can be observed in Table V. The amount of PSNR value dropped does not have much difference between the algorithms. Taking BSDS100 dataset as an example, the percentage of PSNR value dropped from scale ×2 to scale ×3 at a range of 11.30% to 11.70%. Meanwhile, from scale ×3 to scale ×4, the percentage dropped ranged from 6.05% to 8.76%. The PSNR value dropped as image size reduced because a lot of information had been missing when the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

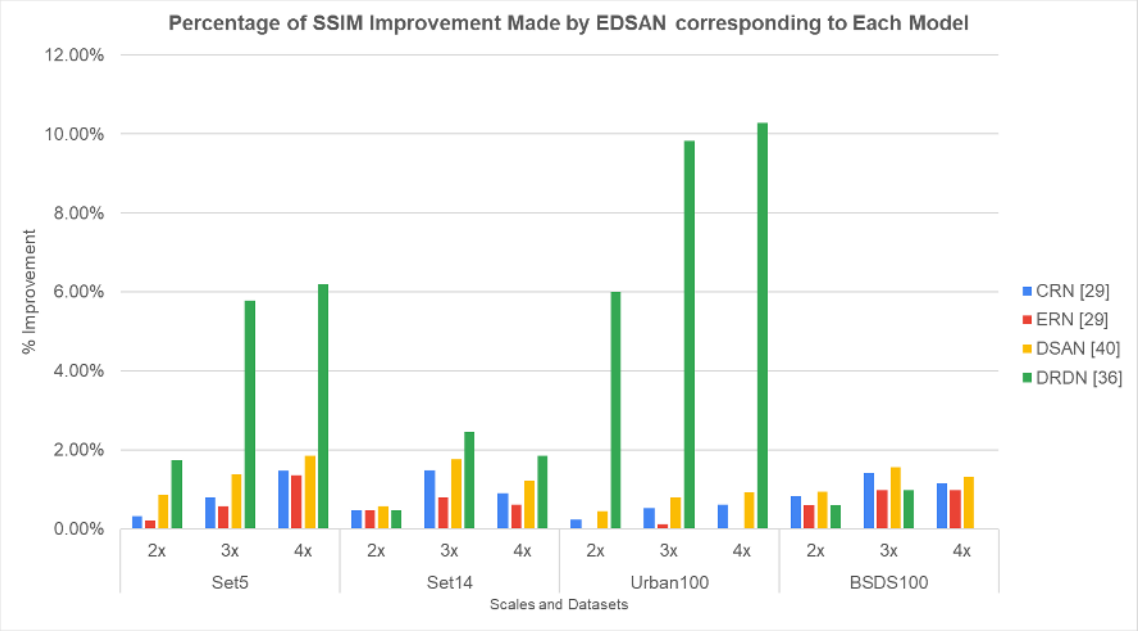


FIGURE 14. Percentage of SSIM Improvement for EDSAN model as compared to CRN [29], ERN [29], DSAN [40] and DRDN [36]

image is reduced to a smaller size. Therefore, it is difficult for the deep learning model to correlate the input feature with the output feature and not to obtain optimum weight settings.

2) RUNNING TIME & NUMBER OF PARAMETERS

Running time is one of the important aspects required in many applications. A shorter running time is not only required by the static image but also in a real-time application, such as a surveillance system. Therefore, the shorter the running time, the better it is in application usage. Therefore, the lower the running time, the better it is in application usage. In this project, the objective is to obtain a shorter running time without forsaking the PSNR value. By taking CRN and ERN as a benchmark since they had the highest PSNR value among the selected models, the EDSAN model is designed to have a shorter running time than CRN and ERN models. For a better observed in running time reduction, the result is interpreted in terms of percentage. The percentage of time reduction achieved by the EDSAN model relative to other individual models can be expressed as in Equation (22).

$$\%\ \text{Running Time Reduction} = -\frac{t_{EDSAN} - t_X}{t_X} \times 100\% \qquad (22)$$

where t_{EDSAN} is the running time of the EDSAN model, t_X is the running time of the model to be compared, and the negative sign is referred to as reduction.

Fig. 15 shows the comparison of average running time for each model at different scales. For a scale factor of 2, EDSAN shows a significant drop in the average running time compared to CRN and ERN. It is about 7.78% and 12.50% of running time reduction, respectively. However, EDSAN is still running slower than DSAN and DRDN models. As for a scale factor of 3, EDSAN is running the fastest among all models. The average running time dropped about 7.50% when compared to CRN, 16.85% when compared to ERN, and 38.33% when compared to DSAN. A huge average running time reduction is made between DRDN and the designed EDSAN model, in which EDSAN achieves about 50.67% drop in average running time. A similar observation is also obtained for as scale factor of 4. EDSAN shows a 12.00% drop in average running time compared to ERN and a 1.49% drop when compared to CRN. A huge drop is observed, about 29.79%, in running time between DSAN and EDSAN models. When compared to the DRDN model, a huge drop in average running time in the EDSAN model is observed, about 55.70%.

Memory resource utilization is another aspect that is concerned by researchers. When the memory resource consumed increases, the higher the cost needed to spend on hardware resources (e.g., graphical processing unit, GPU with high random access memory, RAM). Therefore, the lower the memory consumption required with no impact on PSNR value, the better the model. In this project, CRN and ERN are also taken as a benchmark since they had the highest PSNR to have a lower number of parameters than CRN and ERN models. The percentage of parameter number reduction achieved by the EDSAN model relative to other individual models can be expressed as in Equation (23).

$$\%\ \text{Parameter Time Reduction} = -\frac{N_{EDSAN} - N_X}{N_X} \times 100\% \qquad (23)$$

where N_{EDSAN} is the parameter number of the EDSAN model, N_X is the parameter number of the model to be compared, and the negative sign is referred to as reduction.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

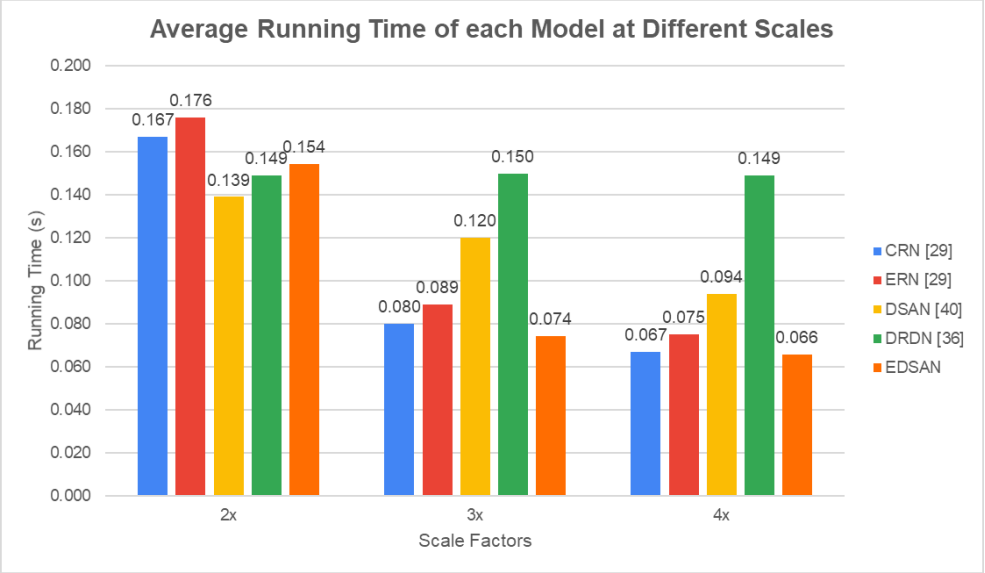


FIGURE 15. **Average Running Time of each model at different scales**

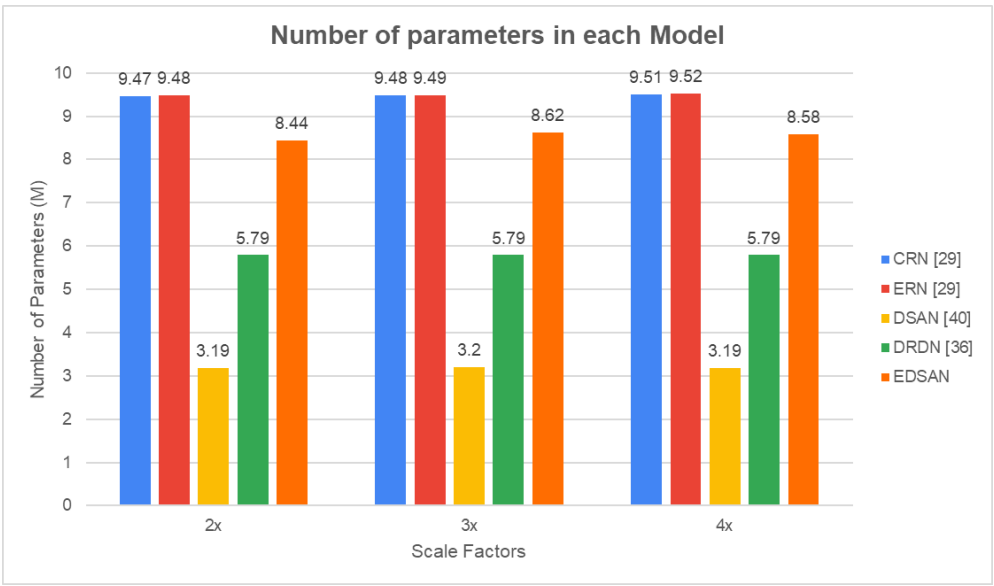


FIGURE 16. **Number of Parameters in each model**

Fig. 16 shows the comparison of the total number of parameters in each model at different scales. The results show that the number of parameters for the EDSAN model is lower than that in CRN and ERN models but higher than that in DSAN and DRDN models at all scales. On a scale factor of 2, about 10.97% and 10.88% reduction is made when compared to ERN and CRN models, respectively. About 9.17% and 9.07% reduction is observed when compared to ERN and CRN models, respectively, at a scale factor of 3. Last but not least, about 9.87% and 9.78% reduction is obtained in running time when compared to ERN and CRN models, respectively.

C. QUALITATIVE ANALYSIS

Qualitative analysis is the analysis of the image through visual observation directly. Fig. 17 and Fig. 18 are the sample image results obtained from Set5 and BSDS100 datasets, respectively. By observing the image from the Set5 dataset, the EDSAN model reconstructs the image very close to the HR image at a scale factor of 2. The CRN, ERN, and DSAN models also reconstruct the image very well, but their color is slightly dimmer than the original image. As for the DRDN model, the image produced is not as clear as compared to the HR image and with others. On a scale factor of 3, the EDSAN model is able to recover the shape of the object perfectly. However, when compared to that at a scale

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

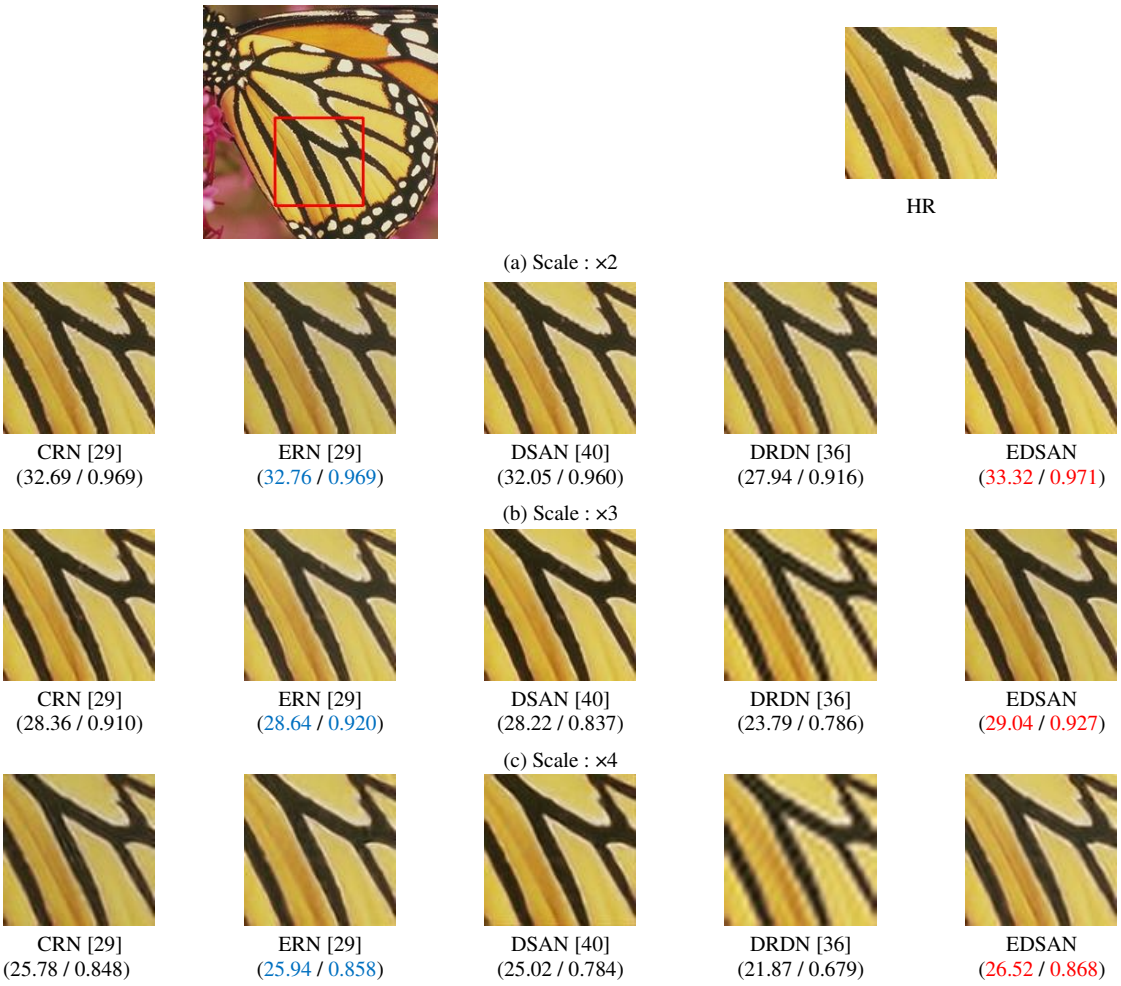


FIGURE 17. Sample results obtained from Set5 datasets at scale ×2, ×3 and ×4

factor of 2, its color is slightly dimmer as well. Besides, the surface of the black line also got smoothen, which has slightly different from the HR image. The EDSAN model is also able to reconstruct the LR image at a scale factor of 4 as close as to the HR image. The edge corner of the image contains some blur visual is observed.

An image from BSDS100 datasets is observed on its character reconstruction. On a scale factor of 2, all models manage to reconstruct a very clear character as similar to the HR image. When coming to the scale factor of 3, except for the DSAN model and DRDN model, which have some difficulty in recover the round shape on characters '9' and '6', the rest of the models, including EDSAN, manage to see the reconstructed character. However, by comparing those images with that in scale factor of 2, the texture of the character produced at a scale factor of 3 is not so perfect. By observing the images reconstructed at a scale factor of 4, the characters are not so clear as compared to other scale factors. However, among the give images produced by each of the models, the character '9' and '6' are still able to recognize by a bare eye from the EDSAN model. Character '0' are more likely to see in the image reconstructed by the CRN, ERN, and DSAN models.

As a summary of the findings, EDSAN performed the best in both PSNR and SSIM values compared to the other four models. The performance improvement made for EDSAN can be summarized to the following points. First, the adoption of dense connection between LWSA utilized all the features from preceding layers and fused them before feeding into the next layer, which had enhanced the correlation between features. Besides, the residual learning is connected between the input, and the output of the last layer of LWDSAB allows the low-level feature to be utilized since low-level feature and high-level feature are highly correlated.

EDSAN model also achieved lower average running time and memory consumption as compared to CRN and ERN. The achievement is made with the help of MSB and CBAM. MSB extracts the input feature from different scales and fused with a non-linear mapping feature that helps the model

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

FIGURE 18. Sample results obtained from BSDS100 datasets at scale ×2, ×3 and ×4

to have visibility on the correlation between a low-level feature and a high-level feature. CBAM helps the model to pay attention to useful channels of the feature. Although the running time and the number of parameters for the EDSAN model are higher than that in DRDN and DSAN, their accuracy performance is significantly lower than that in EDSAN.

Although EDSAN had achieved the best performance among the compared algorithms, EDSAN still has poor little performance to reconstruct images with very smaller scale factor. Reconstruction of the image with a very small scale factor is still an essential technique for many applications that may have small scale images, especially for object images from the surveillance system [53], biometric information like face recognition [54], fingerprint recognition [55] and iris recognition [56].

D. DIFFERENCE WITH OTHER ALGORITHMS

The difference in network design between EDSAN and other algorithms will be compared. The algorithms that will be compared are CRN [29], ERN [29], DSAN [40] and DRDN [36].

1) DIFFERENCE WITH RESPECT TO CRN

The locally sharing group (LSG) in CRN is replaced by a local wider dense space attention block (LWDSAB). In LSG, the individual small group is known as the local wider residual block (LWRB), which is made up of a wider convolutional layer, a narrow convolutional layer, and a multiplication layer. The small group in LWDSAB is known as local wider space attention (LWSA). It is made up of a wider convolutional layer, a narrow convolutional layer, and a convolution block attention module (CBAM). The major difference between the two small groups is the use of CBAM in the EDSAN network to allow the network to pay more

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access

attention to important channel features so that the model is able to reduce the network depth without affecting the performance.

Another difference is that CRN uses residual learning between the small group, whereas EDSAN uses the dense connection to pass all features to the input of each layer to utilize all the features evenly. This enhances the model to correlate input feature and output feature and obtains an optimized weight for the model. Besides, the residual learning adopted in CRN is combining the output feature from the second previous layer and the current layer. In EDSAN, residual learning combines the feature from extracted feature component with the output feature of each LWDSAB because the low-level feature highly correlates to the high-level feature. The output feature from each block is also concatenated before feeding into the upsampling module, in which CRN does not make it.

2) DIFFERENCE WITH RESPECT TO ERN

The major difference between ERN and EDSAN is the use of local wider residual block (LWRB) in ERN and local wider space attention (LWSA) in CRN. LWSA includes CBAM in the group to allow the model to pay more attention to important channel features so that the model is able to reduce the number of layers without affecting the model performance. Besides, both dense connection and residual learning are adopted in LWSA, but only residual learning is found in LWRB. This helps EDSAN utilizes more features evenly and get a better correlation between a low-level feature and a high-level feature to obtain an optimized weight.

3) DIFFERENCE WITH RESPECT TO DSAN

Instead of using dense space attention block (DSAB) in DSAN, EDSAN deployed a local wider dense space attention block (LWDSAB). The difference between them is that DSAB only uses a single convolutional layer, whereas LWSA uses two convolutional layers with different filter sizes. The use of two convolutional layers is used because it was proven that a model could achieve a comparative performance by using a small depth layer of the model with a model that has deeper depth layers. This may significantly reduce the running time and memory resource consumption.

4) DIFFERENCE WITH RESPECT TO DRDN

One of the major differences is that DRDN uses the interpolated image as input, whereas EDSAN uses the original LR image as input and being upsampled after the non-linear mapping process. Besides, the dense block in DRDN only uses a series of convolutional layers with the same filter size. As for EDSAN, it includes a series of the small group, LWSA that has a wider convolutional layer and narrow convolutional layer. This allows more information to be extracted and able to improve the network. Another difference observed in the dense block is all the output features are not concatenated before passing to the next dense block. In contrast, LWSA utilized all the features by fusing them through concatenation before passing to the next block.

VI. CONCLUSION

The proposed EDSAN models had successfully solved the vanishing-gradient problems and the high running time mentioned. The adoption of dense connection between local wider space attention (LWSA) utilized all the features from preceding layers. LWSA fused them before feeding them into the next layer for further feature extraction. Such a method helps the model to correlate features to each other so that the weight of each layer can be optimized easily. Besides, residual learning is also adopted between input and the output of the last layer group. This gives an additional ability to the model to correlate the features and enhance the model performance since low-level feature and high-level feature is highly correlated. Residual learning is also applied in between each of the local wider dense space attention blocks (LWDSAB).

The use of MSB and CBAM also shows that they not only improve the PSNR value but also facilitate in reducing the average running time. Furthermore, memory resources are lowered by reducing the number of layers required by the model. CBAM helps the model by allowing the model to focus on the useful channel of the feature. MSB extracts the input feature from different scales and fuses with the non-linear mapping feature. This allows the model to have visibility on the correlation between a low-level feature and a high-level feature.

The development of EDSAN may help in a few applications. First, biometric information identification, such as face recognition [54], fingerprint recognition [55], and iris recognition [56] that required image super-resolution techniques to enlarge the image for better recognition purposes. Besides, the model can also help in the surveillance system because the video from the surveillance system is sometimes unclear due to the small image size or poor quality of closed-circuit television (CCTV) [53].

The development of image super-resolution algorithms is not limited to EDSAN. There are a few enhancement can be done in future. First of all, these algorithms can be developed with a more lightweight layer that is able to give high PSNR value and is compatible with embedded systems. Embedded systems had become one of the useful platforms nowadays because of their convenience for application use compared to computer system platforms [57]. Besides, transformation techniques, such as fast Fourier transform (FFT) [58] can be added to the model pipeline for another perspective feature extractions to obtain more accessible information respective to the frequency domain. Last but not least, a transfer learning technique can also be applied to the model by transferring knowledge from other datasets, such as natural images into a pre-trained model. This will help the model to learn more details and be robust to a different scenario.

REFERENCES

[1] K. Li, S. Yang, R. Dong, X. Wang, and J. Huang, "Survey of single image super-resolution reconstruction," *IET Image Process.*, vol. 14, no. 11, pp. 2273–2290, 2020, doi: 10.1049/iet-ipr.2019.1438.

[2] S. Anwar, S. Khan, and N. Barnes, "A deep journey into super-resolution: A survey," *ACM Comput. Surv.*, vol. 53, pp. 1–34, 2020.

[3] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020, doi: 10.1109/tpami.2020.2982166.

[4] W. Wang, Y. Hu, Y. Luo, and T. Zhang, "Brief survey of single image super-resolution reconstruction based on deep learning approaches," *Sens. Imaging*, vol. 21, no. 1, pp. 1–20, 2020, doi: 10.1007/s11220-020-00285-4.

[5] C. S. Tong and K. T. Leung, "Super-resolution reconstruction based on linear interpolation of wavelet coefficients," *Multidimens. Syst. Signal Process.*, vol. 18, no. 2–3, pp. 153–171, 2007, doi: 10.1007/s11045-007-0023-2.

[6] N. Sun and H. Li, "Super resolution reconstruction of images based on interpolation and full convolutional neural network and application in medical fields," *IEEE Access*, vol. 7, pp. 186470–186479, 2019, doi: 10.1109/ACCESS.2019.2960828.

[7] J. Liu, Z. Gan, and X. Zhu, "Directional bicubic interpolation: A new method of image super-resolution," in *Proceedings of 3rd International Conference on Multimedia Technology(ICMT-13)*, 2013, pp. 470–477, doi: 10.2991/icmt-13.2013.57.

[8] G. Kumar and K. Singh, "Image super resolution on the basis of DWT and bicubic interpolation," *Int. J. Comput. Appl.*, vol. 65, no. 15, pp. 1–6, 2013, [Online]. Available: papers://b6c7d293-c492-48a4-91d5-8fae456be1fa/Paper/p8210%5Cnfile:///C:/Users/Serguei/OneDrive/Documents/Papers/Image Super Resolution on the-2013-03-13.pdf.

[9] X. Li, Y. Wu, W. Zhang, R. Wang, and F. Hou, "Deep learning methods in real-time image super-resolution: a survey," *J. Real-Time Image Process.*, vol. 17, no. 6, pp. 1885–1909, 2020, doi: 10.1007/s11554-019-00925-3.

[10] K. Nasrollahi and T. B. Moeslund, *Super-resolution: A comprehensive survey*, vol. 25, no. 6. 2014.

[11] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong, "Soft edge smoothness prior for alpha channel super resolution," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2007, doi: 10.1109/CVPR.2007.383028.

[12] K. Chang, P. L. K. Ding, and B. Li, "Single image super resolution using joint regularization," *IEEE Signal Process. Lett.*, vol. 25, no. 4, pp. 596–600, 2018, doi: 10.1109/LSP.2018.2815003.

[13] L. Yu, S. Cao, J. He, B. Sun, and F. Dai, "Single-image super-resolution based on regularization with stationary gradient fidelity," *Proc. - 2017 10th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2017*, vol. 2018-Janua, pp. 1–5, 2018, doi: 10.1109/CISP-BMEI.2017.8301942.

[14] Q. Shan, Z. Li, J. Jia, and C. K. Tang, "Fast image/video upsampling," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 1–8, 2008, doi: 10.1145/1409060.1409106.

[15] 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.

[16] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," *26th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR*, 2008, doi: 10.1109/CVPR.2008.4587647.

[17] A. Maier, C. Syben, T. Lasser, and C. Riess, "A gentle introduction to deep learning in medical image processing," *Z. Med. Phys.*, vol. 29, no. 2, pp. 86–101, 2019, doi: 10.1016/j.zemedi.2018.12.003.

[18] M. A. Khan, F. A. Dharejo, F. Deeba, S. Ashraf, J. Kim, and H. Kim, "Toward developing tangling noise removal and blind inpainting mechanism based on total variation in image processing," *Electron. Lett.*, doi: https://doi.org/10.1049/ell2.12148.

[19] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image Vis. Comput.*, vol. 97, p. 103910, 2020, doi: 10.1016/j.imavis.2020.103910.

[20] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8692 LNCS, no. PART 4, pp. 184–199, 2014, doi: 10.1007/978-3-319-10593-2_13.

[21] Y. K. Ooi and H. Ibrahim, "Deep learning algorithms for single image super-resolution: A systematic review," *Electron.*, vol. 10, no. 7, 2021, doi: 10.3390/electronics10070867.

[22] S. Basodi, C. Ji, H. Zhang, and Y. Pan, "Gradient amplification: An efficient way to train deep neural networks," *Big Data Min. Anal.*, vol. 3, no. 3, pp. 196–207, 2020, doi: 10.26599/BDMA.2020.9020004.

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

[24] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9906 LNCS, pp. 391–407, 2016, doi: 10.1007/978-3-319-46475-6_25.

[25] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 1874–1883, 2016, doi: 10.1109/CVPR.2016.207.

[26] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 1646–1654, 2016, doi: 10.1109/CVPR.2016.182.

[27] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 1132–1140, 2017, doi: 10.1109/CVPRW.2017.151.

[28] J. Chu, J. Zhang, W. Lu, and X. Huang, "A novel multiconnected convolutional network for super-resolution," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 946–950, 2018, doi: 10.1109/LSP.2018.2820057.

[29] R. Lan *et al.*, "Cascading and enhanced residual networks for accurate single-image super-resolution," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 115–125, 2021, doi: 10.1109/TCYB.2019.2952710.

[30] Y. Zou, L. Zhang, C. Liu, B. Wang, Y. Hu, and Q. Chen, "Super-resolution reconstruction of infrared images based on a convolutional neural network with skip connections," *Opt. Lasers Eng.*, vol. 146, no. January, p. 106717, 2021, doi: 10.1016/j.optlaseng.2021.106717.

[31] X. Du, J. Niu, and C. Liu, "Expectation-maximization attention cross residual network for single image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021, pp. 888–896.

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

[33] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2790–2798, 2017, doi: 10.1109/CVPR.2017.298.

[34] J. Hou, Y. Si, and L. Li, "Image super-resolution reconstruction method based on global and local residual learning," *2019 IEEE 4th Int. Conf. Image, Vis. Comput. ICIVC 2019*, pp. 341–345, 2019, doi: 10.1109/ICIVC47709.2019.8981305.

[35] J. Hou, Y. Si, and X. Yu, "A novel and effective image super-resolution reconstruction technique via fast global and local residual learning model," *Appl. Sci.*, vol. 10, no. 5, p. 1856, 2020, doi: 10.3390/app10051856.

[36] W. Wei, J. Yongbin, L. Yanhong, L. Ji, W. Xin, and Z. Tong, "An advanced deep residual dense network (DRDN) approach for image super-resolution," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 2, pp. 1592–1601, 2019, doi: 10.2991/ijcis.d.191209.001.

[37] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 4809–4817, 2017, doi: 10.1109/ICCV.2017.514.

[38] J. Xu, Y. Chae, B. Stenger, and A. Datta, "Dense bynet: Residual dense network for image super resolution," *Proc. - Int. Conf. Image Process. ICIP*, pp. 71–75, 2018, doi: 10.1109/ICIP.2018.8451696.

[39] P. Shamsolmoali, X. Li, and R. Wang, "Single image resolution enhancement by efficient dilated densely connected residual network," *Signal Process. Image Commun.*, vol. 79, no. June, pp. 13–23, 2019, doi: 10.1016/j.image.2019.08.008.

[40] C. Duanmu and J. Zhu, "The image super-resolution algorithm based on the dense space attention network," *IEEE Access*, vol. 8, pp. 140599–140606, 2020, doi: 10.1109/ACCESS.2020.3013401.

[41] J. Liu, W. Zhang, Y. Tang, J. Tang, and G. Wu, "Residual feature aggregation network for image super-resolution," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, pp. 2356–2365, 2020, doi: 10.1109/CVPR42600.2020.00243.

[42] C. C. Hsu and C. H. Lin, "Dual reconstruction with densely connected residual network for single image super-resolution," *Proc. - 2019 Int. Conf. Comput. Vis. Work. ICCVW 2019*, pp. 3643–3650, 2019, doi: 10.1109/ICCVW.2019.00449.

[43] X. Gao, L. Zhang, and X. Mou, "Single image super-resolution using dual-branch convolutional neural network," *IEEE Access*, vol. 7, pp. 15767–15778, 2019, doi: 10.1109/ACCESS.2018.2889760.

[44] J. Liu, Y. Xue, S. Zhao, S. Li, and X. Zhang, "A convolutional neural network for image super-resolution using internal dataset," *IEEE Access*, vol. 8, pp. 201055–201070, 2020, doi: 10.1109/access.2020.3036155.

[45] M. J. Wang, X. Yang, M. Anisetti, R. Zhang, M. K. Albertini, and K. Liu, "Image super-resolution via enhanced multi-scale residual network," *J. Parallel Distrib. Comput.*, vol. 152, pp. 57–66, 2021, doi: 10.1016/j.jpdc.2021.02.016.

[46] J. T. Barron, "A general and adaptive robust loss function," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 4326–4334, 2019, doi: 10.1109/CVPR.2019.00446.

[47] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 1122–1131, 2017, doi: 10.1109/CVPRW.2017.150.

[48] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," *BMVC 2012 - Electron. Proc. Br. Mach. Vis. Conf. 2012*, no. Ml, pp. 1–10, 2012, doi: 10.5244/C.26.135.

[49] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6920 LNCS, no. 1, pp. 711–730, 2012, doi: 10.1007/978-3-642-27413-8_47.

[50] J. Bin Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 5197–5206, 2015, doi: 10.1109/CVPR.2015.7299156.

[51] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, 2001, vol. 2, pp. 416–423 vol.2, doi: 10.1109/ICCV.2001.937655.

[52] 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.

[53] L. Yue, H. Shen, J. Li, Q. Yuan, H. Zhang, and L. Zhang, "Image super-resolution: The techniques, applications, and future," *Signal Processing*, vol. 128, pp. 389–408, 2016, doi: 10.1016/j.sigpro.2016.05.002.

[54] P. Rasti, T. Uiboupin, S. Escalera, and G. Anbarjafari, "Convolutional neural network super resolution for face recognition in surveillance monitoring," in *International Conference on Articulated Motion and Deformable Objects*, 2016, vol. 1, pp. 175–184, doi: 10.1007/978-3-319-41778-3.

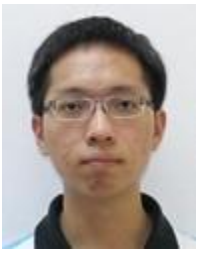
[55] K. Nguyen, C. Fookes, S. Sridharan, M. Tistarelli, and M. Nixon, "Super-resolution for biometrics: A comprehensive survey," *Pattern Recognit.*, vol. 78, pp. 23–42, 2018, doi: 10.1016/j.patcog.2018.01.002.

[56] F. Alonso-Fernandez, R. A. Farrugia, J. Bigun, J. Fierrez, and E. Gonzalez-Sosa, "A survey of super-resolution in iris biometrics with evaluation of dictionary-learning," *IEEE Access*, vol. 7, pp. 6519–6544, 2019, doi: 10.1109/ACCESS.2018.2889395.

[57] B. H. Ha, Y. shin Kim, and P. W. Kim, "Image super-resolution for heterogeneous embedded smart devices and displays in smart global village," *Sustain. Cities Soc.*, vol. 47, no. March, p. 101496, 2019, doi: 10.1016/j.scs.2019.101496.

[58] Varsha Nair; Moitrayee Chatterjee; Neda Tavakoli; Akbar Siami Namin; Craig Snoeyink, "Fast Fourier Transformation for Optimizing Convolutional Neural Networks in Object Recognition," *Comput. Vis. Pattern Recognit.*, pp. 1–10, 2020.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3111983, IEEE Access



YOONG KHANG OOI was born in Penang, Malaysia in 1994. He received his B. Eng degree in electronic engineering from the Universiti Sains Malaysia in 2018. He is currently studying PhD in the field of digital image processing at Universiti Sains Malaysia. His research interest lies in the area of artificial intelligence and software development.

HAIDI IBRAHIM (M'07-SM'18) received his B. Eng degree in electrical and electronic engineering from Universiti Sains Malaysia, Malaysia. He received his Ph.D degree in image processing from Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, United Kingdom in 2005. His research interest includes digital image and signal processing, and analysis.

DR. MUHAMMAD NASIRUDDIN MAHYUDDIN (Senior Member IEEE) is currently an Associate Professor at the School of Electrical and Electronics Engineering, Universiti Sains Malaysia and also an Honorary Visiting Fellow in Faculty of Engineering, University of Bristol. He received his PhD in 2014 from University of Bristol in the area of Mechanical Engineering, specializing on Control and Robotics. He received his M.Eng with high distinction in Mechatronic and Automatic Control from the Universiti Teknologi Malaysia in 2006 and obtained his B.Eng with Honours in Mechatronic Engineering from the International Islamic University of Malaysia in 2004. Soon after graduation, he started his industrial career in 2004 as an Application Engineer at Agilent Technologies working with Motion Control product. He was appointed as a Senior Associate Teacher by University of Bristol via contract, giving lecture in Nonlinear Control with Application to Robotics from October 2011 to July 2012 and involved in a research project funded by Jaguar Land Rover. He was also invited as a Visiting Professor at MIS Lab, Universite de Picardie Jules Verne, France in March and April 2018. He was also attached to Continental Automotive Components during his Sabbatical in 2019 working on closed-loop vehicle instrument cluster test. He received a Secondment International Grant (S0419_01) for September, 2019, February 2020 research visit from R.A.I.N. Programme (Robotics and A.I. research) hosted by The University of Manchester, UK. His current research interests include nonlinear control, distributed adaptive control, cooperative control, and parameter estimation involving mechatronics system and robotics.