IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 June 2023, accepted 13 July 2023, date of publication 20 July 2023, date of current version 27 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3297144

RESEARCH ARTICLE

ESC-PAN: An Efficient CNN Architecture for Image Super-Resolution

ADNAN HAMIDA[1], (Member, IEEE), MOTAZ ALFARRAJ[1,2], (Member, IEEE), AND SALAM A. ZUMMO[1,3], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S3G8, Canada
[2]SDAIA-KFUPM Joint Research Center for Artificial Intelligence, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[3]Center for Communications and Sensing, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Corresponding author: Adnan Hamida (adnan.hamida@mail.utoronto.ca)

The authors would like to acknowledge the support received from the Saudi Data and AI Authority (SDAIA) and King Fahd University of Petroleum and Minerals (KFUPM) through the SDAIA-KFUPM Joint Research Center for Artificial Intelligence.

ABSTRACT Deep Learning models, based on Convolutional Neural Network (CNN) architecture, have proven to be useful and effective in many image processing tasks, and have recently been shown to be effective for image Super-Resolution (SR). Common trends in SR improve the quality of the reconstructed image by increasing the depth and complexity of the CNN model. While this approach produces superior performance in objective image quality metrics (IQA), such as Peak-Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) index, having the number of parameters in the order of millions sacrifices the practicality of model deployment. This is especially true for applications that require real-time processing, such as online conferencing. In this paper, a CNN-based SR model architecture that integrates an attention mechanism while maintaining low complexity is proposed. The number of parameters of the model is reduced by adopting depthwise-separable convolution (DSC) throughout the model. Multiply-accumulate operations (MACs) are reduced by adopting a late upsampling scheme to operate only on low-dimensional features maps. Experimental results show that the proposed model architecture has better performance in terms of objective IQA metrics, such as PSNR and SSIM, and subjective IQA. This improved performance is achieved at a reduced complexity. We also showcase the scalability of the proposed CNN architecture by increasing the model complexity slightly to gain better desired performance.

INDEX TERMS Real-time image super-resolution, depth-wise separable convolution, self-calibrated convolution, pixel attention, image quality assessment.

I. INTRODUCTION

Image super-resolution (SR) is a classical image processing problem that aims to produce a high resolution (HR) image $\mathbf{I}^{HR}$ from a low resolution (LR) one $\mathbf{I}^{LR}$. SR is sometimes referred to in the literature as image upsampling, image upscaling, zooming, and/or enlargement. Image SR is a well-known problem that has been shown to be ill-posed, sometimes referred to as a singular problem, meaning that there is a many-to-one mapping between $\mathbf{I}^{LR}$ and $\mathbf{I}^{HR}$. To alleviate the singularity of the problem, SR methods try to utilize additional, internal or external, information of the input image. Convolutional Neural Network (CNN)-based models have recently significantly outperformed classical methods for the image SR task. The application of CNNs in SR was first proposed in the Super-Resolution Convolutional Neural Network (SRCNN) model [1]. SRCNN, which used a shallow three-layer CNN, outperformed other methods at that time.

Many image processing tasks, including SR, benefited from the design of deep and complex CNN-based models with millions of parameters. These deep complex models achieve improved HR image reconstruction and produces superior performance in objective image quality assessment

The associate editor coordinating the review of this manuscript and approving it for publication was Andrea F. Abate.

(IQA) metrics such as Peak-Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) index [2]. However, these deep complex models lack practicality in implementation for real-time applications. Hung et al. [3] proposed a light SR model that uses recursive depthwise separable convolution (DSC). While this improvement maintains a low number of parameters, multiply-accumulate operations (MACs) grow rapidly due to the recursive block. Most recently, Li et al. [4] proposed s-LWSR, an efficient SR model with a U-Net structure as its backbone. The authors showcased the flexibility of s-LWSR by varying the number of propagated channels through the model, resulting in different complexities. While s-LWSR achieves similar performance to that of deep complex models with a more cost-effective implementation, the efficient version, s-LWSR$_8$, experiences a rapid degradation in performance. In addition, none of the real-time SR models utilize attention mechanisms, which are becoming staple blocks in SR models due to their improved feature representation while maintaining low complexity [5].

In this work, we introduce an SR model architecture that integrates an attention mechanism while maintaining low parameters and MACs for real-time SR. We achieve this by reducing the number of parameters of the model by adopting DSC throughout the model. We also adopt a late upsampling scheme to operate on low-dimensional features maps and reduce MACs. We also introduce an Efficient Self-Calibrated block with Pixel Attention (ESC-PA) that improves the extracted features' representation while maintaining a low number of parameters and MACs. The proposed model architecture is referred to as the Efficient Self-Calibrated Pixel-Attention Network (ESC-PAN). Experimental results show that the proposed ESC-PAN architecture has better performance in terms of PSNR, SSIM, and subjective quality when compared with real-time models of similar complexity. This improved performance is illustrated on two different versions of ESC-PAN with different model complexities, showcasing the scalability of the proposed model architecture.

The rest of the paper is organized as follows, Section II illustrates related works in the literature. Section III discusses limitations of the considered real-time models. Section IV introduces the proposed SR model architecture, ESC-PAN. The architecture is described, along with an explanation for each module in the model. In Section V, experimental results are illustrated that compare two versions of the proposed architecture with existing real-time models in three categories, objective IQA, subjective IQA, and computational complexity. An ablation study is also presented. Section VI concludes the paper and discusses future directions.

II. RELATED WORK

Earlier CNN architectures [1], [6], [7], [8] used for SR choose to do image upsampling at the very beginning. The input to the model would be an upsampled coarse approximation using an interpolation-based method. We refer to this as the *early upsampling scheme*. In this scheme, the network learns to refine and induce high-frequency information to the coarse approximation and obtain the SR image. An advantage of this approach is that only one network needs to be trained for multiple upscale factors. However, having a high-dimensional input is computationally expensive, and MACs grow rapidly as the resolution of the input image increases. This is because most of the operations will be performed on high-dimensional feature maps [9].

The alternative is to have the upsampling as a learnable task for the network rather than using interpolation-based methods. The input to the network will be the downsampled LR image and all operations will be done on low-dimensional feature maps to reduce MACs. We refer to this as the *late upsampling scheme*. In the literature, there are two prominent ways for the network to upsample. One way is by using a transposed convolutional layer, sometimes referred to as a deconvolutional layer. The other way is by using a sub-pixel convolution layer [10]. Dong et al. redesigned the architecture of SRCNN in [11] to perform upsampling using a transposed convolutional layer. This resulted in better performance with significant reduction in computational complexity, naming the network Fast SRCNN (FSRCNN). However, transposed convolutional layers suffer from what is known as the checkerboarding artifacts [12]. Shi et al. introduce a sub-pixel convolution layer to perform the upsampling in their network, Efficient Sub-Pixel Convolutional Network (ESPCN) [10], which achieved comparable results with a light architecture that eliminates checkerboarding artifacts [13]. Sub-pixel convolution layer has been widely adopted in the SR literature, including SSNet [3] and s-LWSR [4].

Recently, there has been a shift towards the incorporation of DSC layers instead of standard convolution in SR models. In the SR literature, the authors of [3] introduce a model based on recursive DSC layers. The recursive DSC block resulted in better performance while maintaining the same number of parameters as the baseline ESPCN [10]. Furthermore, the input image is over-sampled to higher dimensions than the target, then adaptively downsampled to the target dimension. The authors refer to this technique as Super Sampling (SS) and the model is named Super Sampling Network (SSNet). The authors also proposed a variant of the model, named SSNet-M, with fewer layers, and without the recursive block to reduce the model parameters and MACs.

Deep Learning SR models have recently started to equip their architectures with attention mechanisms. The essence behind attention mechanisms is that features vary in their importance and improving SR performance can be achieved if we focus the *attention* of the model on important features. Usually, this is done by weighting features according to their importance. Some of the popular attention mechanisms include channel attention, spatial attention, and Pixel Attention (PA). Residual Channel Attention Network (RCAN) [14] leverages the channel attention mechanism and introduces global and local residual connections. The results from RCAN showed significant perceptual and objective quality improvements with high computational complexity. Zhao et al. introduced PA [5] which alleviates the design of

complex attention blocks by using 1×1 convolution to generate 3D PA feature maps. PA is integrated with the newly introduced Self-Calibrated (SC) convolution block [15], which uses a nested convolution structure. The authors showed through a study that PA is more suitable to improve the performance of lighter models rather than complex ones.

III. LIMITATIONS OF REAL-TIME SR MODELS

Existing state-of-the-art real-time SR models that are considered in this work are SRCNN [1], FSRCNN [11], ESPCN [10], s-LWSR$_8$ [4], SSNet-M, and SSNet [3]. These models are considered for real-time applications due to their relatively light complexity. Complexity considers the number of parameters of the model and the required MACs given a certain target image dimensions. To illustrate this, Table 1 describes the complexity of the real-time SR models for the case of $\times 4$ SR and a target image of size 540×360 to calculate the MACs. G stands for 10^9 MACs, and k stands for 10^3 parameters.

TABLE 1. Complexity of real-time SR models for scale $\times 4$.

Model	Parameters	MACs
SRCNN [1]	57.3k	11.1G
FSRCNN [11]	12.6k	0.980G
ESPCN [10]	24.8k	0.301G
s-LWSR$_8$ [4]	36.7k	0.794G
SSNet-M [3]	7.8k	0.096G
SSNet [3]	23.1k	1.262G

To have some perspective on how light these real-time SR models are, consider deeper more complex models; namely, DnCNN [7], VDSR [6], RCAN [14], and EDSR [16]. Table 2 describes the complexity of these models with the same $\times 4$ SR and target image of size 540×360. In Table 2, G stands for 10^9 MACs, and M stands for 10^6 parameters. Comparing the complexities in Table 1 and Table 2 and it is immediately clear how light the selected models are.

TABLE 2. Complexity of complex SR models for scale $\times 4$.

Model	Parameters	MACs
DnCNN [7]	556.0k	108.1G
VDSR [6]	664.7k	129.2G
RCAN [14]	15.9M	191.9G
EDSR [16]	37.8M	918.2G

State-of-the-art real time SR models that are considered in this work suffer from certain limitations. These include the use of tanh activation function, the use of standard convolution, carrying the low-frequency information through the model, SS technique scalability, increased MACs for the recursive DSC block, and limited capability of shallow U-Net. We will describe each limitation below.

ESPCN, SSNet-M, and SSNet use tanh as the activation function. The authors of ESPCN argued using tanh instead of the conventional ReLU because their experimental results showed superior performance. However, tanh, similar to sigmoid, suffers from the vanishing gradient problem. This is because the derivative of tanh has limited values (0 to 1) that cause vanishing gradients and hinders the training process.

SRCNN, FSRCNN, and ESPCN use standard convolution throughout their models. In the case of the three-layer ESPCN, the second layer alone requires around 18.5k parameters. This constitutes around 75-87% of ESPCN's total parameters depending on the upscale factor. Furthermore, all of these real-time SR models are similar in the sense that the model serves two purposes. First, the model learns to carry the low-frequency information from the input to the output. Second, the model has to reconstruct the residual image, which is the high-frequency information. However, both LR and HR share the same low-frequency information. Kim et al. [6] showed that passing the low-frequency can be done using a skip connection. This skip connection contains an upsampled coarse approximation using any interpolation method on LR. This way, the model would serve one purpose, which is to reconstruct high-frequency information. Kim et al. [6] showed that using this skip connection allows the model to converge faster and further improves the reconstruction.

In [3], the SS technique is selected to double the target image dimensions for SSNet-M and SSNet. This way, the 1×1 convolutional layer before upsampling would have $(2r)^2$ output channels rather than the conventional r^2 in [10]. For input channels C, the number of parameters in that layer will be $(C+1)\cdot(2r)^2$, rather than $(C+1)\cdot(r)^2$. This multiplier by 4 becomes more prominent as the scale factor r increases. Thus, limiting the scalability of the model for larger scaling factors (i.e. $\times 8$, $\times 16$, etc.). Furthermore, SSNet matches the number of parameters of ESPCN by using the recursive DSC layers and reusing parameters. While this approach maintains low parameters, the required MACs increase significantly as illustrated in Table 1, where SSNet requires more than four times the MACs of ESPCN.

In [4], the authors propose a flexible and efficient architecture for SR based on U-Net. While the core of the architecture remains U-Net, the authors make modifications to adapt it for SR. However, using a shallow U-Net restricts its capability for higher feature extraction and reconstruction. As a result, there is a rapid degradation in performance when reducing the complexity of the model, as in the case of s-LWSR$_8$ [4]. We consider s-LWSR$_8$ here since its complexity is in the same order as the real-time SR models we consider as shown in Table 1. This limitation will be shown in Section V.

IV. PROPOSED MODEL ARCHITECTURE

In light of the aforementioned limitations of the existing real-time SR models, an efficient model in terms of number parameters and MACs that achieves superior objective and subjective performance is proposed. The proposed model consists of three modules, *feature extraction module*,

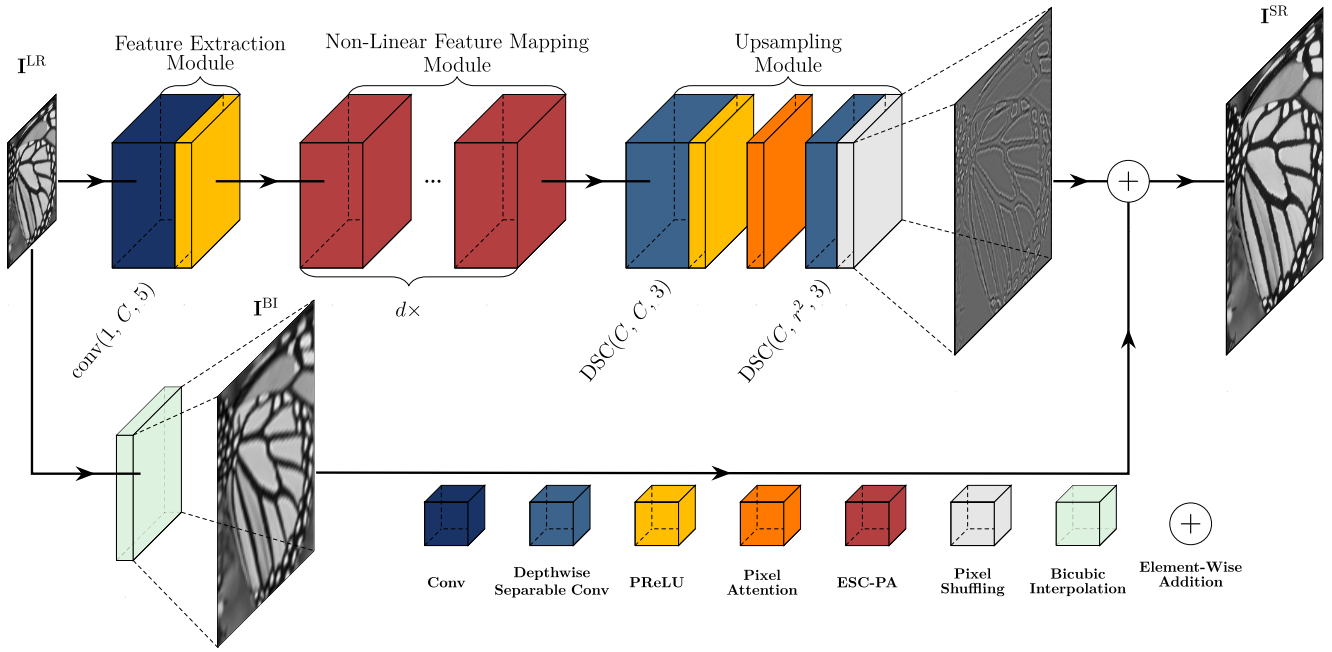

FIGURE 1. Proposed model architecture.

non-linear feature mapping module, and *upsampling module*. The architecture of the model is presented in Figure 1. Padding is used in all convolutional layers to maintain the dimensions of the input image. In this work, we used "replicate" padding, which repeats the edge pixels. The number of padded pixels depends on the kernel size at that layer. If the kernel size at a certain layer is $k \times k$, then the added pixels will be $\frac{(k-1)}{2}$.

A. FEATURE EXTRACTION MODULE

The first module of the model is the *feature extraction module*. There are many definitions of what image features are, but broadly speaking, feature extraction computes representative features or attributes that best represent the input image. There exist predefined bases for feature extractions, such as discrete cosine transform (DCT), Haar wavelet bases, and many others. In the *feature extraction module*, these bases are learned. The *feature extraction module* consists of one standard convolutional layer, conv(1, C, 5), where 1 is the number of input channels, C is the number of output channels, 5 is the kernel size (or 5×5). The input channels are set to 1 since all the SR operations will be carried out on a grayscale image, or the Y channel of the YCbCr color space.

All layers in the proposed model structure are activated with Parametric Rectified Linear Unit (PReLU) activation function [17]. PReLU is a generalized version of the common ReLU activation function, where a_i is a trainable parameter that corresponds to the slope in the non-positive domain. For ReLU, a_i is set to 0, and for Leaky ReLU, a_i is set to a fixed value. In PReLU however, a_i is a trainable parameter. PReLU is similar to Leaky ReLU in the sense that it avoids the "dead features" issue in conventional ReLU caused by zero gradients when the input to ReLU is in the negative range. Leaky ReLU sets the negative slope to a fixed value introducing a hyperparameter. We choose PReLU since it is flexible and sets the negative slope as a trainable parameter.

B. NON-LINEAR FEATURE MAPPING MODULE

The second module is the *non-linear feature mapping module*, which plays an important role in the design of the model architecture since it contains most of the model parameters. It is responsible for learning the mapping function between the $\mathbf{I}^{LR}$ image features and $\mathbf{I}^{HR}$ image features. The *non-linear feature mapping module* consists of a stack of d ESC-PA blocks. Before delving into ESC-PA, we will briefly explain DSC layers and the PA attention mechanism.

1) DEPTHWISE SEPARABLE CONVOLUTION

The idea behind DSC layers is to factorize the convolution kernels that are in standard convolutional layers. Following the formulation in [18], consider the three cases in Figure 2.

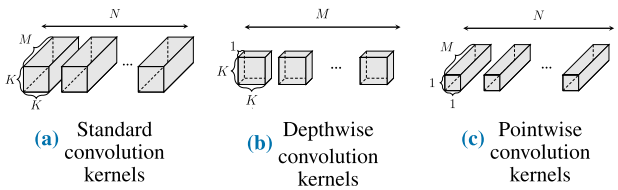

FIGURE 2. Convolutional layer kernels.

In Figure 2a, standard convolution kernels are illustrated for a certain conv(M, N, K) layer. Each kernel in a standard

convolution layer is of size $K \times K \times M$, where M is the number of input channels to the layer. There are N output channels, thus there is a total of N kernels available in the layer. Note that each kernel has one additional bias parameter. Then, the total number of parameters in the standard convolutional layer will be

$$\text{conv}(M, N, K) \text{ parameters} = K \times K \times M \times N + \overbrace{N}^{\text{biases}}. \tag{1}$$

DSC layers, on the other hand, factorizes the kernel into two stages. The first stage is *depthwise* convolution illustrated in Figure 2b, where M 2D kernels of size $K \times K \times 1$ are used. Each kernel in this stage is convolved with one of the input channels. The output from the first stage is fed into the second stage. The second stage is *pointwise* convolution illustrated in Figure 2c, where N 1×1 kernels are used. Both stages constitute a DSC layer block. The DSC layer is denoted similar to standard convolution as DSC(M, N, K), where M is the input channels to the *depthwise* stage, N is the output channels from the *pointwise* stage, and K is the kernel size at the *depthwise* stage. Then, the number of parameters in DSC layer will be

$$\begin{aligned}\text{DSC}(M, N, K) \text{ parameters} = & \underbrace{K \times K \times M + \overbrace{M}^{\text{biases}}}_{\substack{\text{depthwise convolution} \\ \text{parameters}}} \\ & + \underbrace{M \times N + \overbrace{N}^{\text{biases}}}_{\substack{\text{pointwise convolution} \\ \text{parameters}}}.\end{aligned} \tag{2}$$

To illustrate the complexity reduction of DSC from standard convolution, consider the layers conv(32, 32, 3) and DSC(32, 32, 3), and MACs are calculated for input image with dimensions $540 \times 360 \times 32$, where 540 is the height, 360 is the width, and 32 is the number of channels from the previous layer. The number of parameters reduce from 9,248 in conv(32, 32, 3) to 1,376 in DSC(32, 32, 3). MACs also reduce from 1.798G to 0.267G. DSC reduces the complexity by more than 85% from the standard convolution parameters and MACs in this case.

2) PIXEL ATTENTION MECHANISM

Figure 3 illustrates the block diagram of PA proposed in [5]. Consider an input feature maps cuboid, $\mathbf{X}$, of dimensions $(H \times W \times C)$. $\mathbf{X}$ is passed into a conv(C, C, 1) layer. The output from the convolutional layer is fed into a sigmoid activation to normalize the values between [0,1] and produce the normalized cuboid $F(\mathbf{X})$. This normalized cuboid is used as weights and it is element-wise multiplied with input cuboid $\mathbf{X}$ to produce the output $\tilde{\mathbf{X}}$. The output cuboid $\tilde{\mathbf{X}}$ is a weighted

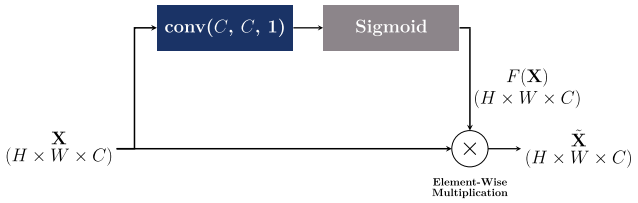

FIGURE 3. Pixel attention block diagram.

version of $\mathbf{X}$, and the weights in $F(\mathbf{X})$ focus the "attention" of the network into important features that are present in the input cuboid $\mathbf{X}$.

3) EFFICIENT SELF-CALIBRATED BLOCK WITH PIXEL ATTENTION

ESC-PA block diagram is shown in Figure 4. The input feature maps cube $\mathbf{X}$ of dimensions $(H \times W \times C)$ is split into two using 1×1 convolutional layers. Two branches are formed, one for $\mathbf{X}'$, and one for $\mathbf{X}''$, each with dimensions $(H \times W \times \frac{C}{2})$. The upper branch contains the PA component, which is the sub-branch that contains the K2 layer followed by the sigmoid activation. Having the PA component, along with K3 and K4 layers allows for higher-level feature manipulation on $\mathbf{X}'$. The lower branch is responsible for maintaining the input original information, which consists of one DSC layer. The outputs from the two branches are concatenated and summed with the input $\mathbf{X}$ to form the output $\mathbf{Y}$ of the same dimensions as the input.

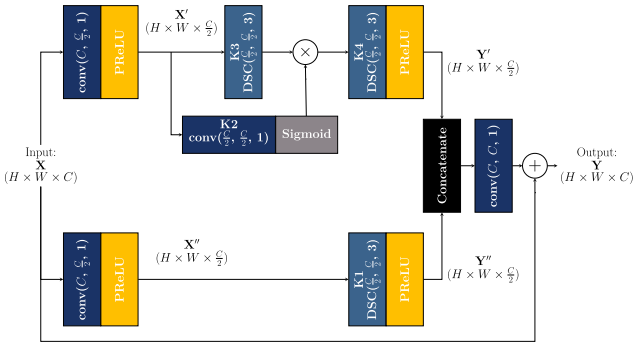

FIGURE 4. Efficient self-calibrated block with pixel attention block diagram.

Comparing ESC-PA with SC-PA in [5], ESC-PA replaces all activation functions with PReLU, and all the 3×3 standard convolutional layers with 3×3 DSC layers to reduce the parameters needed. To illustrate the scale of complexity reduction in the proposed block, consider the case of an input image with dimensions 540×360 and the input channels are set to $C = 32$. In this case, the number of parameters of the block reduces from 9,232 in SC-PA to 3,524 in ESC-PA, and MACs in this block will be reduced from 113M to 43M and effectively reducing the complexity by more than 60%.

C. UPSAMPLING MODULE

The *upsampling module* is the last module of the model, and it is responsible for reconstructing the upsampled SR image to the target dimension. This module consists of DSC(C, C, 3), followed by a PA block, followed by DSC(C, r^2, 3), followed by a pixel shuffling operator for upsampling. The value of r is the desired upscale factor (2, 3, and so on). The last DSC layer and the the pixel shuffling operator constitute the sub-pixel convolutional layer introduced in [10]. To briefly explain how the sub-pixel convolutional layer perform the upsampling, consider an input $\mathbf{I}^{\mathrm{LR}}$ of dimensions ($H \times W \times 1$). Then the output of the last DSC layer will be ($H \times W \times r^2$). The pixel shuffling operator rearranges the dimensions of the output of the last layer to ($rH \times rW \times 1$) and produce the upsampled image.

Finally, a global skip connection is established that upsamples the input $\mathbf{I}^{\mathrm{LR}}$ to the desired dimension using bicubic interpolation. This coarse approximation is referred to as $\mathbf{I}^{\mathrm{BI}}$, and is added to the output of the model to finally produce the super-resolution output image $\mathbf{I}^{\mathrm{SR}}$. The role of this branch in the network is pass the low-frequency information outside the model. The low-frequency information is shared between $\mathbf{I}^{\mathrm{LR}}$ and $\mathbf{I}^{\mathrm{HR}}$, and since bicubic does not induce any external information, $\mathbf{I}^{\mathrm{BI}}$ carries the same low-frequency information.

Throughout the paper, we will denote the proposed model architecture with ESC-PAN(r, C, d), where r is the desired upscale factor, C are the propagated channels through the model, and d is the number of blocks in the stack of ESC-PA blocks in the *non-linear feature mapping module*.

V. EXPERIMENTS

In this section, the adopted datasets for training, validation, and testing are described. An ablation study for the adopted techniques and the architecture scalability is illustrated. Two versions of the proposed model architecture are used to compare with real-time SR models. The comparison is three-fold, performance on objective IQA metrics PSNR and SSIM, visual outputs for subjective IQA, and computational complexity in terms of the number of parameters and MACs.[1]

A. DATASETS

For training and validating, DIV2K [19] is used. DIV2K composed of 1000 high-resolution images, with splits 80/10/10 for training, validation, and testing. DIV2K images have minimal noise and contain at least 2K pixels in either vertical or horizontal dimension. DIV2K was chosen because it contains images with large dimensions, which allows for more extracted patches, and with data augmentation, more images can be generated. In addition, the available images are rich with details and sharp edges. The 800 training images and 100 validation images are used.

For testing, we evaluate the model on the five benchmark SR testing datasets. Set5 [20], which contains five uncompressed images for a baby, bird, butterfly, head, and a woman.

[1]The code is available at: https://github.com/AdnanHamida/ ESC-PAN

TABLE 3. Ablation study summary for scale ×4.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Convolution layer type	Standard	DSC	DSC	DSC	DSC	DSC
Pixel attention	With	Without	With	With	With	With
Upsampling scheme	Late	Late	Early	Late	Late	Late
Skip connection interpolation	Bicubic	Bicubic	Bicubic	Nearest	Bicubic	Bicubic
Propagated channels C	32	32	32	32	32	16
# of ESC-PA blocks d	1	1	1	1	1	10
Parameters (k)	25.0	6.3	7.1	7.6	7.6	21.4
MACs (G)	0.304	0.077	22.199	0.093	0.093	0.260
Valid. Set [19] PSNR (dB)	27.65	27.45	27.37	27.54	27.56	27.83

Set14 [21], which is a nine image extension of Set5 that contains a larger variety of images in terms of content, where natural and unnatural images are included. BSD100 [22], which contains 100 images that are mostly natural scenery, food, and people. Urban100 [23], which contains 100 images for urban scenery of man-made structures that are rich with details and edges. Manga109 [24] which contains 109 manga volume hand-drawn cover images.

B. ABLATION STUDY

Table 3 presents an ablation study that investigates the effectiveness of the adopted techniques, namely DSC, PA, and late upsampling. The comparison includes model complexity and PSNR performance on the DIV2K validation set [19] for ×4 SR. The complexity comparison includes the number of parameters and MACs required for a target image size of 540 × 360. Note that the column labeled "Model 5" corresponds to ESC-PAN(4, 32, 1), which was proposed in our previous work [25]. The motivation behind setting $C = 32$ and $d = 1$ is to reduce the number of parameters and MACs, making the model comparable to the most efficient real-time SR model in Table 1 (SSNet-M [3]).

To assess the effectiveness of other factors influencing our proposed model architecture, we vary the number of propagated channels C and the number of ESC-PA blocks in the *non-linear feature mapping module* d, introducing "Model 6". Following our notation, Model 6 is denoted as ESC-PAN(4, 16, 10). This version is included to demonstrate the scalability of the proposed architecture and compare it with SSNet [3].

Note that for standard convolution in Model 1, validation PSNR is higher than ESC-PAN(4, 32, 1) by 0.1dB. However, this comes at the cost of three times the complexity of ESC-PAN(4, 32, 1) in terms of both the number of parameters and required MACs. Notice also that the newly proposed ESC-PAN(4, 16, 10) achieves the highest PSNR and at lower complexity than Model 1. Additionally, removing PA in Model 2 reduces ESC-PAN(4, 32, 1) parameters and MACs by approximately 16-17%, but at the cost of 0.1dB performance. The adoption of late upsampling is beneficial in both reducing MACs and improving performance. Early upsampling in Model 3 increases MACs significantly and produces the worst performance out of all cases. However, late upsampling comes at a small cost of increasing the number of parameters due to the convolutional layer before the pixel shuffling operator. Moreover, changing the interpolation method in the skip

connection does not matter much when comparing Model 4 with ESC-PAN(4, 32, 1), as using bicubic interpolation rather than nearest neighbor interpolation improves the validation PSNR by 0.02dB at no additional complexity. Recall that the skip connection's goal is to pass the low-frequency information, and any interpolation-based upsampling is sufficient.

In summary, ESC-PAN, with its main components, provides a compromise between slightly increasing the complexity while maintaining good performance. Furthermore, increasing the depth of the model archives the highest PSNR at lower complexity when comparing ESC-PAN(4, 16, 10) with Model 1.

C. TRAINING DETAILS

The training images are sliced into 120 × 120 patches and downsampled using bicubic interpolation to generate $\mathbf{I}^{\text{LR}}$ input image. We apply vertical and horizontal flipping for training data augmentation. ℓ_1 loss is adopted during training. To define the ℓ_1 loss, consider $\mathbf{I}^{\text{SR}}$ prediction produced by the SR model and the ground truth $\mathbf{I}^{\text{HR}}$, where both have the dimensions $m \times n$. ℓ_1 loss can be described as follows

$$\ell_1 = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|\mathrm{I}^{\text{SR}}(i,j) - \mathrm{I}^{\text{HR}}(i,j)\right|, \tag{3}$$

where $|\cdot|$ here denotes the absolute value. The validation data is used in full dimensions with no augmentation. PSNR of validation set is monitored and the model with the highest validation PSNR is saved and used for testing. Adam optimizer is used with $\alpha_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ and an initial learning rate of 10^{-3}. The learning rate reduces by a factor of 0.75 if the validation PSNR plateaus for 20 epochs, and would stop decaying when it reaches 10^{-4}. Training stops when validation loss stops decreasing for 100 epochs.

D. RESULTS AND DISCUSSION

The summary of the performance comparison of both versions of ESC-PAN with the considered real-time SR models from Table 1 on the test datasets as well as model complexity is presented in Table 4. Results are reported for scales ×2, ×3, and ×4. MACs in the second column are calculated for a target image of size 540 × 360.

Note that the reported SSNet-M and SSNet MACs for ×4 SR in [3] does match what we tried to reproduce from the authors public code.[2] Therefore, in Table 4, we include the generated values for the parameters and MACs of SSNet-M and SSNet from the available public code. For ESPCN, we reproduce the results following the training details provided in [10] since its evaluation is not reported for the adopted scales and testing datasets in the original manuscript. Similarly, s-LWSR$_8$ trained version is not available in the authors public repository,[3] so we reproduced the results for all the adopted scales and testing datasets. Note that our training of s-LWSR$_8$ achieved slightly better performance than the ones originally reported in [4] for the case of ×4 SR.

We covered the comparison between ESC-PAN(r, 32, 1) with the rest of the real-time SR models in our work [25], but excluded SSNet and s-LWSR$_8$ from the comparison. This is because of the clear higher complexity of SSNet and s-LWSR$_8$ when compared with ESC-PAN(r, 32, 1) as seen in the complexity column of Table 4 for all the adopted scales. However, it is interesting to point out that with significantly less complexity in both parameters and MACs, ESC-PAN(r, 32, 1) performance is slightly less than s-LWSR$_8$. ESC-PAN(r, 32, 1) actually slightly outperforms s-LWSR$_8$ in case of PSNR score of Set14 [21] and Urban100 [23] for scale ×3.

In this work, we introduce ESC-PAN(r, 16, 20) that has a similar complexity to SSNet and s-LWSR$_8$ for a fair comparison. Looking at the complexity in the second column of Table 4, notice that similar to the case of SSNet-M, SSNet grows around 3.1k parameters from the case of ×2 to ×4. On the other hand, ESC-PAN(r, 16, 20) grows only 0.2k parameters and has less complexity for the case of ×4, and same number of parameters for the case of ×3 when comparing with SSNet. Regarding MACs, ESC-PAN(r, 16, 20) always has significantly less required MACs than SSNet because SSNet utilizes recursive DSC layers as described in Section III. Also, ESC-PAN(r, 16, 20) complexity is always less than s-LWSR$_8$ for all cases in both the number of parameters and MACs.

For objective performing using PSNR and SSIM metrics, ESC-PAN(r, 16, 20) produces the best performance in most of the datasets for the considered scale factors. ESC-PAN(r, 16, 20) only falls short in three instances. The first instance is the case of Set14 for scale ×2, where it performs marginally worse than SSNet in terms of PSNR, while producing higher SSIM. The second instance is the case of BSD100 for scale ×2, where SSNet-M actually produces the highest SSIM, but ESC-PAN(r, 16, 20) produces the best PSNR. The final instance is the case of BSD100 for scale ×4, where SSNet produces the highest SSIM, but ESC-PAN(r, 16, 20) produces the higher PSNR with less complexity. In all cases however, ESC-PAN(r, 16, 20) outperforms s-LWSR$_8$ with a noticeable margin and with less complexity. This highlights the limitation of shallow U-Net discussed in Section III.

It is also important to point out that both SSNet-M and SSNet structures were determined in an ablation study in [3]. The deciding metric was chosen to be PSNR performance at Set5. This way, Set5 is used as a validation set rather than a testing set, as it was used to evaluate the model multiple times. This approach explains the high PSNR of Set5 for both SSNet and SSNet-M for all scales. Furthermore, the high correlation between Set5 and Set14 also explains the high PSNR of SSNet-M and SSNet that are comparable, or even perform better in one case, to our ESC-PAN models.

To have a combined comparison of both objective scores and complexity, we plot the average PSNR score of the five testing datasets for each model versus the required MACs in

[2]https://github.com/kwokwaihung/Real_Time_SR
[3]https://github.com/Sudo-Biao/s-LWSR

TABLE 4. Objective evaluation for real-time SR models for scales ×2, ×3, and ×4. bold is for the best performing model.

Method	Scale	Complexity Params./MACs	Set5 [20] PNSR/SSIM	Set14 [21] PNSR/SSIM	BSD100 [22] PNSR/SSIM	Urban100 [23] PNSR/SSIM	Manga109 [24] PNSR/SSIM
Bicubic Interpolation	×2	-/-	33.68/0.930	30.24/0.869	29.56/0.844	26.88/0.841	31.05/0.935
SRCNN [1]		57.3k/11.135G	36.66/0.954	32.45/0.907	31.36/0.888	29.51/0.895	35.72/0.968
FSRCNN [11]		12.6k/1.276G	36.98/0.956	32.62/0.909	31.50/0.890	29.85/0.901	36.62/0.971
ESPCN [10]		21.3k/1.034G	37.04/0.957	32.62/0.909	31.50/0.891	29.82/0.900	36.31/0.970
s-LWSR$_8$ [4]		33.8k/1.629G	37.44/0.959	32.91/0.912	31.74/0.894	30.43/0.910	37.27/0.974
SSNet-M [3]		6.2k/0.303G	37.08/0.956	32.78/0.908	31.46/**0.900**	29.92/0.900	36.47/0.970
ESC-PAN(2, 32, 1) [25]		7.2k/0.352G	37.40/0.959	32.84/0.911	31.70/0.894	30.31/0.908	36.93/0.973
SSNet [3]		20.0k/4.892G	37.51/0.958	**33.19**/0.912	31.79/0.894	30.76/0.914	37.55/0.974
ESC-PAN(2, 16, 20)		21.2k/1.0G	**37.73/0.960**	33.15/**0.914**	**31.98**/0.897	**31.10/0.918**	**37.91/0.976**
Bicubic Interpolation	×3	-/-	30.40/0.869	27.54/0.774	27.21/0.739	24.46/0.735	26.95/0.856
SRCNN [1]		57.3k/11.135G	32.75/0.909	29.29/0.822	28.41/0.786	26.24/0.799	30.48/0.912
FSRCNN [11]		12.6k/1.057G	33.16/0.914	29.42/0.824	28.52/0.789	26.41/0.806	31.10/0.921
ESPCN [10]		22.7k/0.491G	33.01/0.913	29.39/0.823	28.45/0.788	26.28/0.801	30.77/0.914
s-LWSR$_8$ [4]		36.7k/0.794G	33.35/0.918	29.56/0.828	28.63/0.793	26.63/0.816	31.61/0.927
SSNet-M [3]		6.9k/0.150G	33.24/0.915	29.51/0.824	28.42/0.788	26.44/0.805	31.01/0.918
ESC-PAN(3, 32, 1) [25]		7.4k/0.160G	33.35/0.918	29.58/0.828	28.62/0.792	26.65/0.814	31.34/0.926
SSNet [3]		21.3k/2.204G	33.89/0.922	29.83/0.830	28.69/0.795	27.09/0.826	32.14/0.932
ESC-PAN(3, 16, 20)		21.3k/0.460G	**33.90/0.923**	**29.88/0.833**	**28.85/0.798**	**27.23/0.830**	**32.40/0.936**
Bicubic Interpolation	×4	-/-	28.43/0.811	26.00/0.702	25.96/0.668	23.14/0.657	25.15/0.789
SRCNN [1]		57.3k/11.135G	30.48/0.863	27.50/0.751	26.90/0.710	24.52/0.723	27.66/0.858
FSRCNN [11]		12.6k/0.980G	30.70/0.866	27.59/0.754	26.96/0.713	24.60/0.726	27.89/0.859
ESPCN [10]		24.8k/0.301G	30.69/0.866	27.58/0.754	26.95/0.714	24.54/0.722	27.72/0.853
s-LWSR$_8$ [4]		36.1k/0.530G	31.03/0.876	27.81/0.763	27.11/0.720	24.81/0.739	28.36/0.873
SSNet-M [3]		7.8k/0.096G	31.01/0.873	27.61/0.755	26.84/0.710	24.62/0.725	27.87/0.858
ESC-PAN(4, 32, 1) [25]		7.6k/0.093G	31.02/0.876	27.78/0.762	27.11/0.719	24.81/0.738	28.30/0.873
SSNet [3]		23.1k/1.262G	31.54/0.884	28.01/0.764	27.08/**0.738**	25.08/0.746	28.90/0.879
ESC-PAN(4, 16, 20)		21.4k/0.260G	**31.56/0.886**	**28.13/0.770**	**27.31**/0.726	**25.24/0.755**	**29.21/0.890**

Figure 5. The number of parameters is represented through the radius of each circle. Figure 5 is generated for ×4 SR from the results in Table 4 and it offers a combined method to see the objective performance along with model complexity. Notice how our previously proposed ESC-PAN(4, 32, 1) [25] stands out as the most complexity-efficient model, outperforming more complex methods such as FSRCNN, ESPCN, SSNet-M, and SRCNN. Furthermore, notice that ESC-PAN(4, 32, 1) performance is slightly lower than s-LWSR$_8$, while providing massive gains in reducing computational complexity. Moreover, Figure 5 clearly demonstrates how ESC-PAN(4, 16, 20) attains the best average objective score. This is achieved at lower complexity than SSNet, s-LWSR$_8$, ESPCN, FSRCNN, and SRCNN. Results in Figure 5 are inline with aforementioned observations on Table 4, but also provides a combined method to compare performance and complexity.

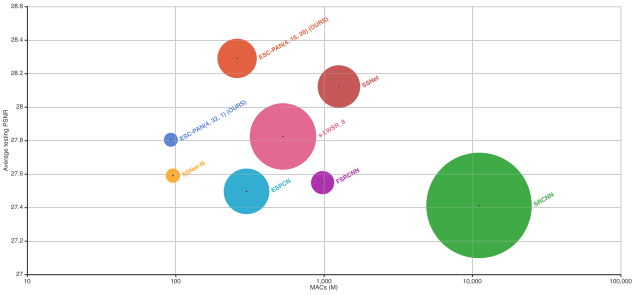

FIGURE 5. Benchmarking real-time SR models for ×4 SR. The number of parameters is proportional to circle radius.

For subjective evaluation, Figure 6, Figure 7, and Figure 8 illustrate visual examples of the considered models. Figure 6 illustrates visual examples for a sample image from Urban100 [23] testing dataset. Let us first compare ESC-PAN(4, 32, 1) with similar complexity models; SRCNN, FSRCNN, ESPCN, and SSNet-M in figures 6c-6h. Both ESC-PAN(4, 32, 1) and SSNet-M produce the same higher SSIM score than the rest of the considered models, and ESC-PAN(4, 32, 1) produces the higher PSNR. As for the zoomed-in patch, we chose this patch as the frequency of the intensities increased in the diagonal direction requiring high frequency reconstruction at the top right edge in this specific test image. ESC-PAN(4, 32, 1) produces sharper edges when compared with SRCNN, FSRCNN, ESPCN, and SSNet-M in the zoomed-in patch with the white lines showing much more detail and better contrast with dark background. It is notable to note that ESC-PAN(4, 32, 1) has the least complexity at this upscale factor when compared with all of the considered models. As for the ESC-PAN(4, 16, 20) in Figure 6j, SSNet in

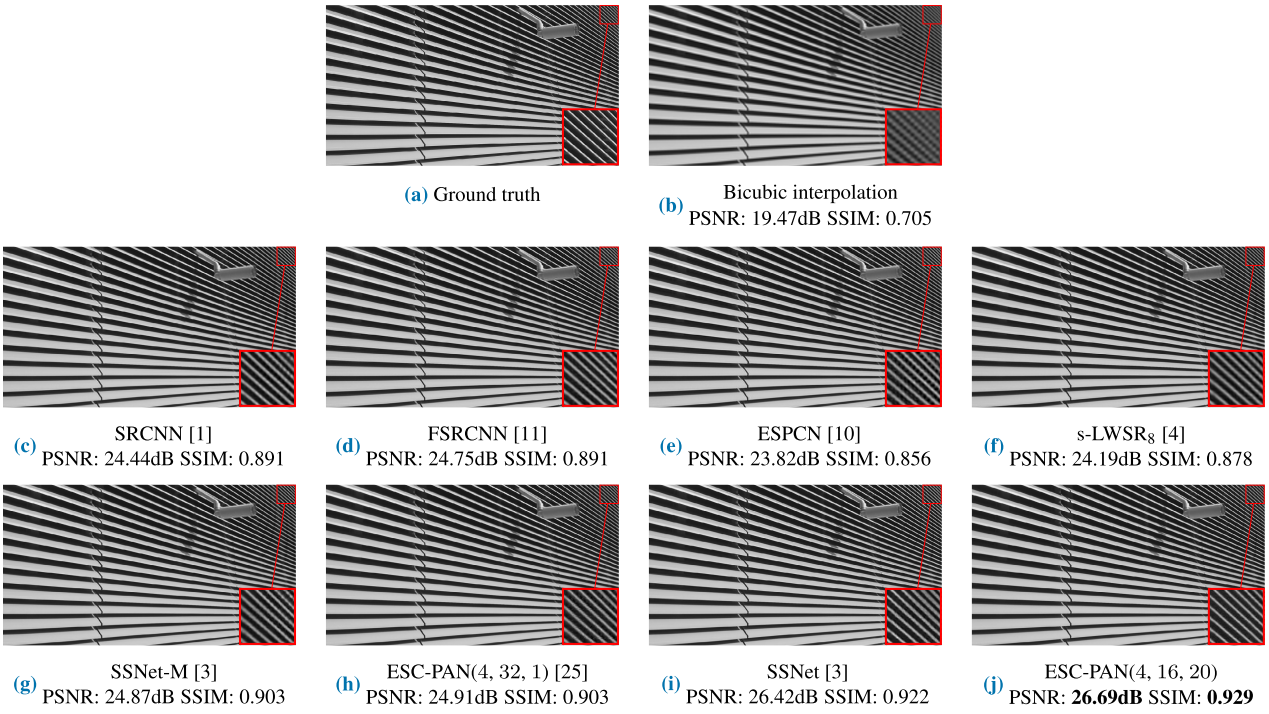

FIGURE 6. Testing sample from Urban100 [23] for scale ×4.

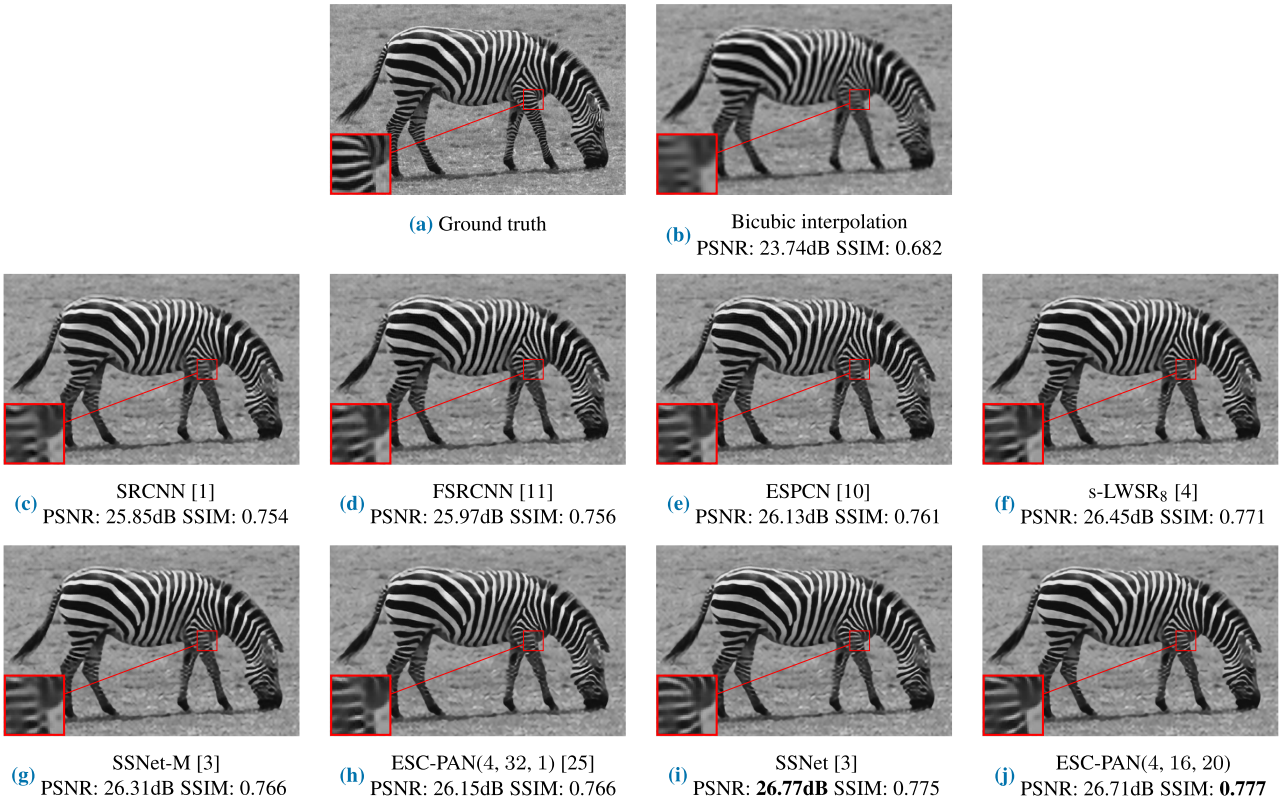

FIGURE 7. Testing sample from BSD100 [22] for scale ×4.

Figure 6i, and s-LWSR$_8$ in Figure 6f, our model ESC-PAN(4, 16, 20) produces the highest PSNR and SSIM scores when compared all the considered SR models. Furthermore, ESC-PAN(4, 16, 20) showcases a clear sharp reconstruction of the

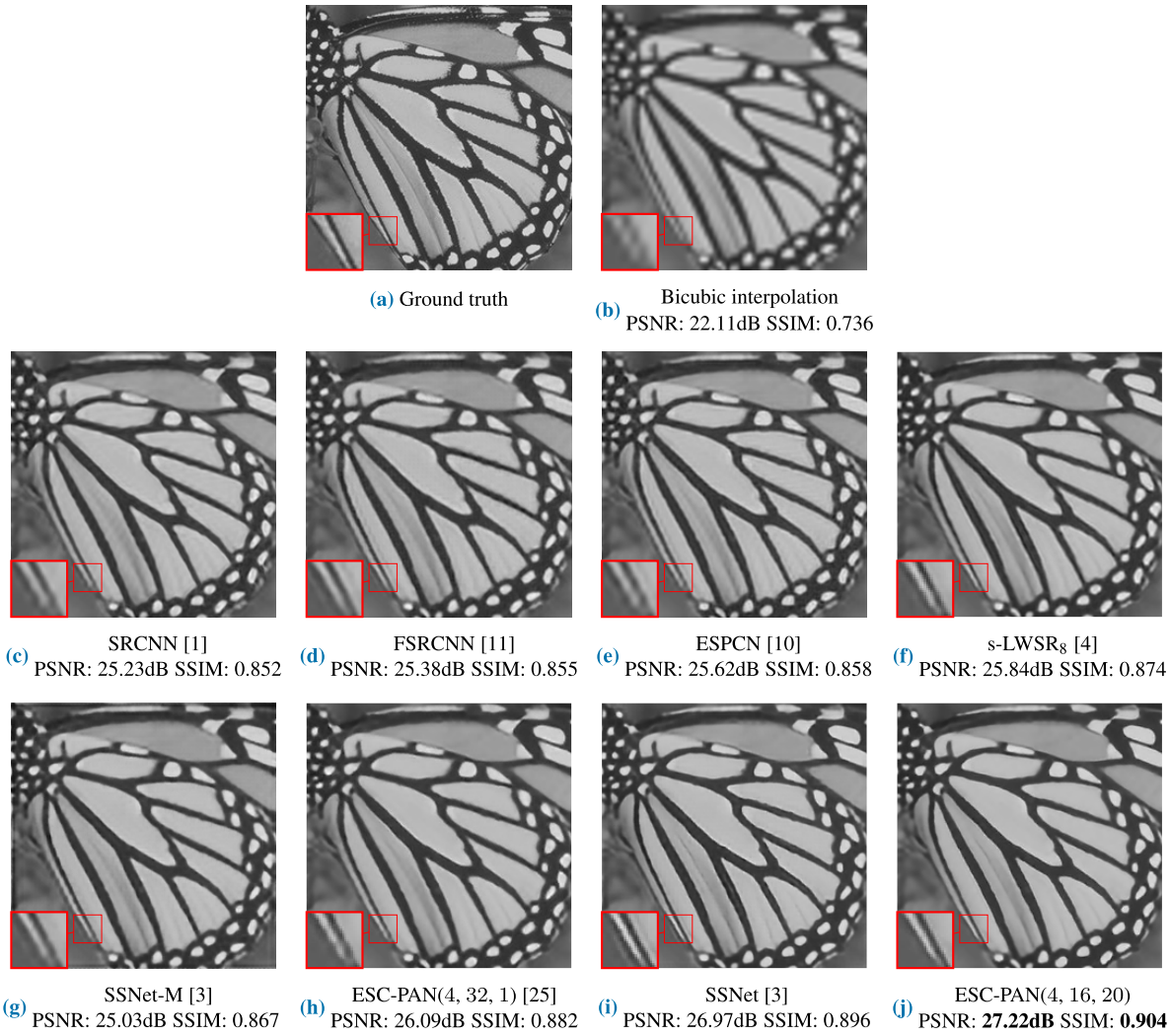

FIGURE 8. **Testing sample from Set5 [20] for scale ×4.**

white lines in the zoomed-in patch that is similar to the ground truth image in Figure 6a. This high performance from ESC-PAN(4, 16, 20) is achieved at lower complexity than SSNet and s-LWSR$_8$. Another important note, even though ESC-PAN(4, 32, 1) produces inferior objective metrics than SSNet, the zoomed-in patch of ESC-PAN(4, 32, 1) in Figure 6h illustrates a sharp reconstruction at the top right corner when compared with SSNet in Figure 6i. This is achieved with a drastic reduction in complexity when comparing ESC-PAN(4, 32, 1) and SSNet.

Figure 7 presents another visual example. The image is from BSD100 [22]. While SSNet in Figure 7i achieves the highest PSNR, our model ESC-PAN(4, 16, 20) achieves the highest SSIM, as shown in Figure 7j. Visually, both results are subjectively very similar. Additionally, when examining the zoomed-in patch, it is evident that the stripes in the zebra's body are reconstructed similarly in both cases. One could argue that the reconstruction of ESC-PAN(4, 16, 20) exhibits richer contrast, as previously discussed in Figure 6. Furthermore, it is worth noting the sharp distinction between the background and the zebra in the reconstruction of ESC-PAN(4, 16, 20) in Figure 7j. This sharp distinction is particularly emphasized in our model, which explains the higher SSIM score when compared with the rest of the models.

Another visual example for the subjective evaluation of the considered models is shown in Figure 8 for a sample image from Set5 [20] testing dataset. As seen from Figure 8h and Figure 8j, both versions of ESC-PAN provide superior performance by maintaining edge details. This is apparent in the zoomed-in patch at the edge between the butterfly wing and the background. This boundary between the wing and the background in the zoomed-in patch illustrates the sharp reconstruction of both ESC-PAN models, where models such as FSRCNN, SSNet-M, SSNet have induced somewhat of a halo effect. Furthermore, notice that ESC-PAN(4, 16, 20) reconstructs the image without inducing any artifacts, such

as in FSRCNN in Figure 8d, ESPCN in Figure 8e, SSNet-M in Figure 8g, and SSNet in Figure 8i. In addition, both versions of ESC-PAN produce a better contrast between the white spots on the butterfly wing and its dark background. Objective metric PSNR and SSIM are the highest for ESC-PAN(4, 16, 20) for this image.

VI. CONCLUSION

In this work, we introduce an efficient CNN architecture for the image super-resolution task. Efficiency of the model is achieved by utilizing depthwise separable convolution throughout the model to reduce the parameters. The model architecture limits all operations to the low resolution input image and adopts a late upsampling scheme, effectively reducing the required multiply-accumulate operations. Feature representation is improved by utilizing a pixel attention mechanism and by introducing an efficient self-calibrated block with pixel attention.

Two versions of the proposed model architecture are illustrated in the experimental evaluation that are compared with state-of-the-art real-time super-resolution models. The considered models are of similar complexity to both versions of the proposed model architecture. The carried comparison is three-fold, performance on objective image quality assessment metrics PSNR and SSIM, visual outputs for subjective image quality assessment, and computational complexity in terms of the number of parameters and multiply-accumulate operations. Experimental results showed that both versions of the proposed architecture produced superior performance in terms of PSNR and SSIM on the five benchmark super-resolution datasets. Visual results also showcase the superior subjective performance of the proposed architecture. Furthermore, we showcase the scalability of the proposed model architecture by showing improved objective and subjective reconstructed image quality with the least complexity at larger upscale factors.

In this work, only pixel-wise based loss function is adopted; namely ℓ_1 loss. In the super-resolution literature, there are trends of using a combined loss. Future direction of this work should focus on the combination of loss functions such as pixel-wise loss functions with perceptual loss [26], and loss function based on SSIM [27]. Incorporating the SSIM loss function will improve the SR reconstruction.

REFERENCES

[1] 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, Feb. 2016.
[2] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[3] K.-W. Hung, Z. Zhang, and J. Jiang, "Real-time image super-resolution using recursive depthwise separable convolution network," *IEEE Access*, vol. 7, pp. 99804–99816, 2019.
[4] B. Li, B. Wang, J. Liu, Z. Qi, and Y. Shi, "S-LWSR: Super lightweight super-resolution network," *IEEE Trans. Image Process.*, vol. 29, pp. 8368–8380, 2020.
[5] H. Zhao, X. Kong, J. He, Y. Qiao, and C. Dong, "Efficient image super-resolution using pixel attention," in *Computer Vision (ECCV)*, A. Bartoli and A. Fusiello, Eds. Cham, Switzerland: Springer, 2020, pp. 56–72.
[6] J. Kim, J. K. Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2016, pp. 1646–1654.
[7] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[8] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2808–2817.
[9] S. Anwar, S. Khan, and N. Barnes, "A deep journey into super-resolution: A survey," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, May 2020, doi: 10.1145/3390462.
[10] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2016, pp. 1874–1883.
[11] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 391–407.
[12] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard/
[13] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi, "Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize," 2017, *arXiv:1707.02937*.
[14] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 1–16.
[15] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10093–10102.
[16] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," 2017, *arXiv:1707.02921*.
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[19] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1122–1131.
[20] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L.-A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2012, pp. 1–10.
[21] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. curves Surf.* Cham, Switzerland: Springer, 2010, pp. 711–730.
[22] 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, May 2011.
[23] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Jun. 2015, pp. 5197–5206.
[24] A. Fujimoto, T. Ogawa, K. Yamamoto, Y. Matsui, T. Yamasaki, and K. Aizawa, "Manga109 dataset and creation of metadata," in *Proc. 1st Int. Workshop Comics Anal., Process. Understand.*, Dec. 2016, pp. 1–5.
[25] A. Hamida, M. Alfarraj, and S. A. Zummo, "Efficient self-calibrated convolution for real-time image super-resolution," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 1176–1180.
[26] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision (ECCV)*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 694–711.
[27] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Image*, vol. 3, no. 1, pp. 47–57, Mar. 2017.

ADNAN HAMIDA (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2019 and 2022, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada.

MOTAZ ALFARRAJ (Member, IEEE) received the B.Sc. degree in electrical engineering from the King Fahd University of Petroleum and Minerals (KFUPM), in 2013, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2016 and 2019, respectively.

He is currently an Assistant Professor with the Electrical Engineering Department, KFUPM. He is also the Director of the SDAIA-KFUPM Joint Research Center for Artificial Intelligence (JRC-AI). His research interests include machine learning, deep learning, computer vision, and image processing. His research focuses on the integration of physics in data-driven systems to enable effective learning from noisy data for applications in oil and gas exploration and production. He is a member of the Society of Exploration Geophysicists (SEG) and Society of Petroleum Engineers (SPE).

SALAM A. ZUMMO (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 1998 and 1999, respectively, and the Ph.D. degree from the University of Michigan, Ann Arbor, USA, in June 2003.

He was the Dean of the Graduate School, KFUPM, from 2008 to 2020, where he is currently a Professor with the Electrical Engineering Department. He has 18 issued U.S. patents and more than 200 papers published in reputable journals and conference proceedings. In addition to his vast research experience in the area of wireless communications, his current research interests include physical layer security, visible light communications, and the applications of artificial intelligence (AI) in different signal processing and wireless communication optimization problems. He was awarded the Award for Outstanding Contribution to Education, in 2016, the KFUPM Excellence in Research Award for the years 2012 and 2019, and the KFUPM Best Research Project Award, in 2016. In addition, he won the British Council/BAE Research Fellowship Awards, in 2004 and 2006, and the Saudi Ambassador Award for early Ph.D. completion, in 2003.