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.2019.DOI

Image Super-Resolution Using Capsule Neural Networks

JUI-TING HSU[1], CHIH-HUNG KUO[2], (MEMBER, IEEE), AND DE-WEI CHEN.[3]

[1]Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan(email: x739145682@gmail.com)
[2]Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan(email: chkuo@ee.ncku.edu.tw)
[3]Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan(email: z40203x@yahoo.com.tw)

Corresponding author: De-Wei Chen (e-mail: z40203x@yahoo.com.tw).

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grants MOST 107-2221-E-006-221, MOST 108-2634-F-006-002 and National Cheng Kung University and Qualcomm Collaborating Research.

ABSTRACT

CONVOLUTIONAL neural networks (CNNs) have been widely applied in super-resolution (SR) and other image restoration tasks. Recently, Hinton *et al.* proposed capsule neural networks to resolve the problem of viewpoint variations in image classification tasks. Each capsule is represented as either a vector or a matrix to encode more object information, such as position, size, direction, etc. Instead of detecting specific features, these capsule neural networks search for the most relevant features using an iterative process. Therefore, capsule neural networks require fewer parameters compared to traditional neural networks. Inspired by these advances, we make use of a capsule neural network to exploit more potential features for image SR. In this paper, we develop two frameworks: the Capsule Image Restoration Neural Network (CIRNN) and the Capsule Attention and Reconstruction Neural Network (CARNN), to incorporate capsules into image SR convolutional neural networks. The CIRNN takes advantage of the rich information encoded in the capsules to reconstruct accurate high-resolution images. The CARNN generates SR attention features by utilizing the robust segmentation capability of the capsules. Our experiments show that both frameworks can enhance SR for most testing datasets. The CIRNN performs better than the CARNN and can achieve better performance than other traditional CNN methods with a similar amount of parameters.

INDEX TERMS Super resolution, deep learning, convolutional neural network, capsule neural network.

I. INTRODUCTION

Image Super-Resolution (SR) techniques are aimed toward recovering high-resolution (HR) images from low-resolution (LR) ones. Interpolation is the simplest by which to do this by producing sub-pixels via combinations of adjacent pixels. However, the resulting images usually contain aliasing artifacts and over-smoothed regions.

A dictionary-based algorithm is one of the representative methods typically used to solve the SR problem. Yang *et al.* [1] proposed to jointly learn the non-linear functions that map from LR patches to HR patches with two coupled dictionaries. The sparse representation coefficients for each LR patch are identified from the LR dictionary and are used to reconstruct the HR patch from the HR dictionary. Then the HR image is composed by aggregating the HR patches.

Deep learning-based SR methods have been intensively studied in recent years. Dong *et al.* [2] first proposed a CNN-based algorithm by directly learning the end-to-end mapping between LR and HR images. Many other works have applied similar concepts to train their neural networks [3]–[5]. However, these methods scale up the input image before feeding it into CNN and waste unnecessary operations on redundant sub-pixels. Some SR systems [6] [7] replace the pre-scaling with post-scaling using convolutional upsampling layers. Lai *et al.* [8] addressed difficulties in learning the mapping for large scaling factors by progressively upsampling. Harris *et al.* [9] exploited many up- and down-sampling layers to generate deeper features.

While conventional neural networks have become dominant in many computer vision applications, Sabour *et al.* [10] introduced a new framework using a brand new type of neurons, called capsules, to represent more intricate features. An iterative algorithm based on the routing-by-agreement mechanism is utilized to find a relevance path connecting the input and output capsules [11]. Later, Hinton *et al.* [12] proposed another routing method based on the Expectation-

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.2020.2964292, IEEE Access

Maximization (EM) algorithm. Capsules can automatically encode various information and model instantiation parameters. Motivated by these advantages, we apply capsule neural networks to implement robust SR systems.

The remainder of this paper is organized as follows. Section II reviews CNN-based SR methods, capsule neural networks and attention networks. Section III presents the proposed image super-resolution frameworks. The experimental results are discussed in Section IV. We conclude this paper in Section V.

II. PRIOR WORKS

A. CNN-BASED IMAGE SUPER-RESOLUTION

The Super-Resolution Convolutional Neural Network (SRCNN) [2] learns the non-linear functions that map from LR images to HR images. This framework pre-scales up the input LR image to the target resolution. However, the computation complexity increases dramatically. The pre-upsampled operation may produce no high-frequency information, but new noise. To alleviate this problem, the Fast Convolution Neural Network for Super-Resolution (FSRCNN) [6] is trained to reconstruct the HR image without any pre-upsampling. Unlike in [2], the upsampling operation in FSRCNN is performed by a deconvolutional layer at the last stage. Therefore, the number of convolutional operations is significantly reduced since the processing area in the layers before the last stage is reduced. By setting stride s of the filters, the deconvolutional layer scales up the input s times. However, the reconstructed images may suffer from chessboard artifacts that are caused by the overlapping areas of sliding windows. Shi *et al.* [7] applied a sub-pixel convolutional layer that assembles several small LR feature maps into a large SR feature map through pixel-shuffling. The sub-pixel convolutional layer can generate more delicate patterns and thus the chessboard artifacts can be avoided.

B. CAPSULE NEURAL NETWORK

Hinton *et al.* [13] first introduced the concept of capsules to express an image object from different viewpoints. A capsule consists of a group of neurons, which can be formed either as a pose vector or as a pose matrix, along with an associated activation probability. A pose within a capsule may represent various types of information, including positions, sizes, orientations, deformations, etc. A capsule layer also contains transformation matrices and a routing algorithm. Capsules layers are concatenated with convolutional layers to form a capsule neural network. Low-level capsules usually represent some parts inside larger objects, which are also represented as capsules in the next higher level. The poses of low-level capsules are transformed by weight matrices to cast votes for high-level capsules. So far, there have been two methods, Dynamic Routing (DR) [10] and Expectation-Maximization Routing (EMR) [12], which were proposed to find high-level capsules among voted candidates.

Fig. 2 illustrates the routing mechanism for a capsule $\mathbf{v}_j$ in the $(l+1)$th layer. The capsules $\{\mathbf{u}_i\}$ in layer l are first transformed into candidates $\{\hat{\mathbf{u}}_{j|i}\}$ by multiplying with trained weight matrices $\mathbf{W}_{ij}$. Then, they are combined with coupling coefficients $\{c_{ij}\}$, which can be refined with different routing strategies. For Dynamic Routing (DR), a non-linear squashing function is applied to restrict the length of output capsule $\mathbf{v}_j$ to be under a unit. The coupling weights c_{ij} are then iteratively refined with an agreement that is proportional to the inner product between $\mathbf{v}_j$ and $\hat{\mathbf{u}}_{j|i}$. On the other hand, for Expectation-Maximization Routing (EMR), each output capsule $\mathbf{v}_j$ is modeled as a Gaussian random matrix with a mean μ_j and a variance σ_j. The agreement is gauged using a probabilistic cost function, and then used to iteratively update the coupling coefficients c_{ij} through an EM-like process. Interested readers are referred to references [10] and [12] for more details. The geometric relations between objects and their parts can be more accurately identified with these routing processes. The activations of capsules in the last layer usually represent the probabilities of the objects. Thus, a capsule neural network can learn to detect objects hierarchically.

C. ATTENTION NETWORKS

Attention networks can recalibrate the important components of the input features and have been applied in various fields, such as image generation [14], image description [15] [16], image recognition [17] [18] and image restoration [19] [20]. The attention weights are generated by a separate subnetwork from the input feature maps and are usually applied in two different ways, channel attention and spatial attention. For channel attention, each channel is multiplied with the corresponding coefficient. For spatial attention, feature elements in the same spatial position of different maps are multiplied with a shared weight. This may make it possible to obtain more information in certain important regions.

In [21] [19], attention networks are proposed for an SR task that achieves state-of-the-art performance. Inspired by the remarkable performance of the capsule neural network in the segmentation task [10], we also designed capsule attention networks to extract more SR features than would otherwise be possible.

III. IMAGE SUPER-RESOLUTION USING CAPSULE NEURAL NETWORK

In contrast to conventional neural networks, a capsule neural network encodes object features into vectors or matrices to represent richer information, which can be exploited to reconstruct finer images in SR applications. Therefore, we propose two series of SR methods using capsule neural networks, the Capsule Image Restoration Neural Network (CIRNN) and the Capsule Attention and Reconstruction Neural Network (CARNN). The receptive fields and channel numbers for these architectures are intentionally designed to be comparable to those of SRCNN [2]. In this work, the SR effects of applying capsules are investigated rather than merely using convolutional neurons.

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.2020.2964292, IEEE Access

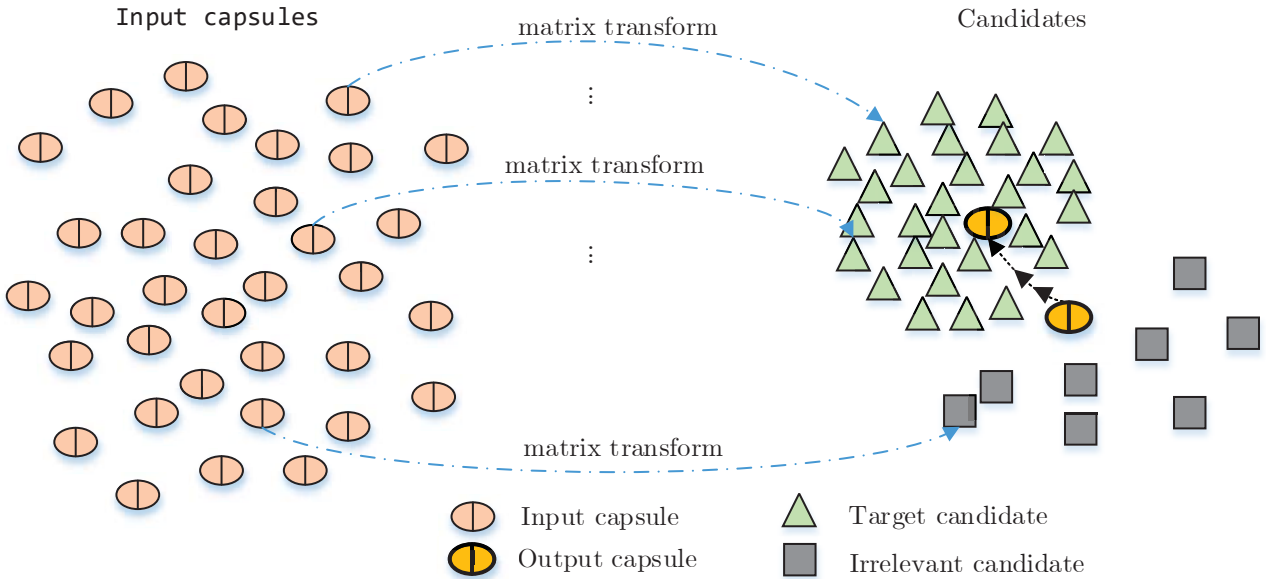

FIGURE 1. The mechanism of routing algorithm in the CIRNN. The broken line shows that the routing algorithm produces the output capsule toward the center of the target capsules with each routing iteration.

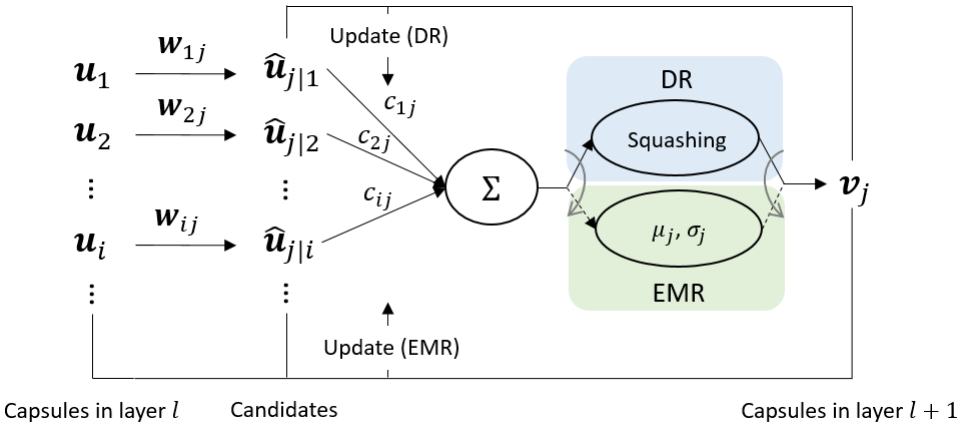

FIGURE 2. The mechanism of routing algorithm.

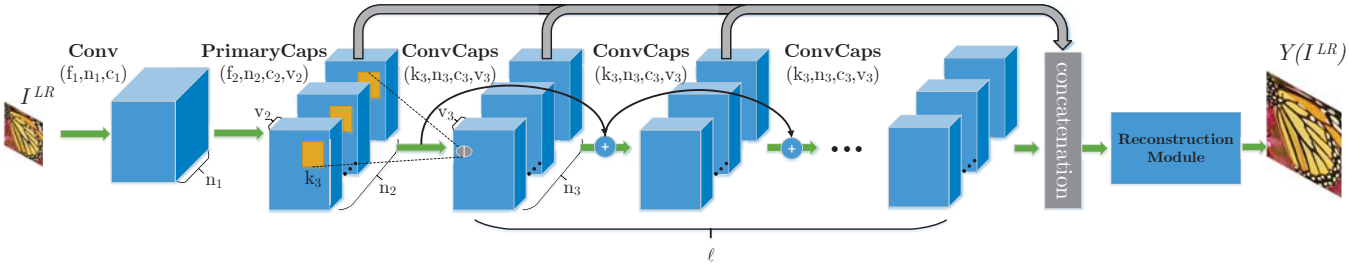

FIGURE 3. The framework of the proposed CIRNN with an upsampling layer.

A. CAPSULE IMAGE RESTORATION NEURAL NETWORK (CIRNN)

The CIRNN is designed to reconstruct HR images directly from capsules. Fig. 1 illustrates the routing mechanism for SR applications. Input capsules may contain either feature matrices or vectors. They are transformed into another capsule space that contains both target candidates and irrelevant candidates. We assume that the target candidates contain

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.2020.2964292, IEEE Access

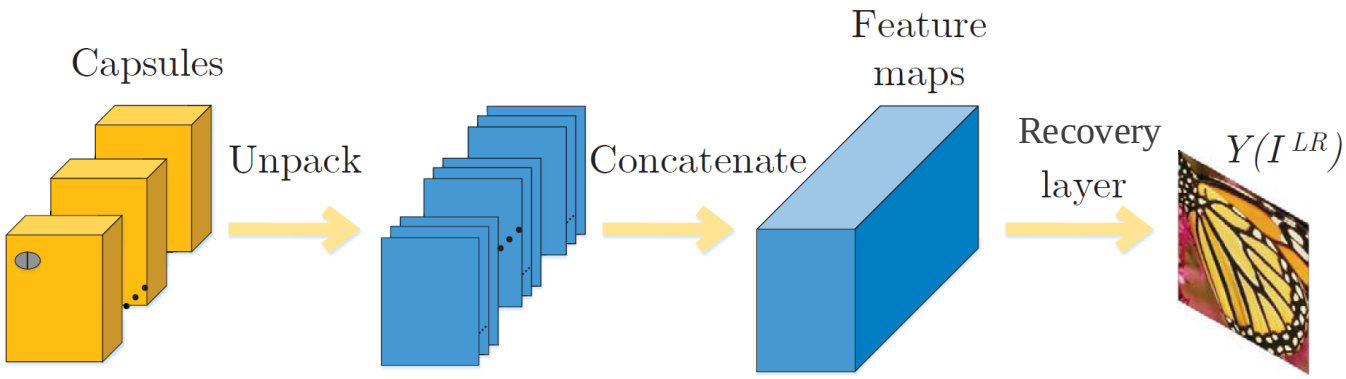

FIGURE 4. Reconstruction module for DR in the CIRNN.

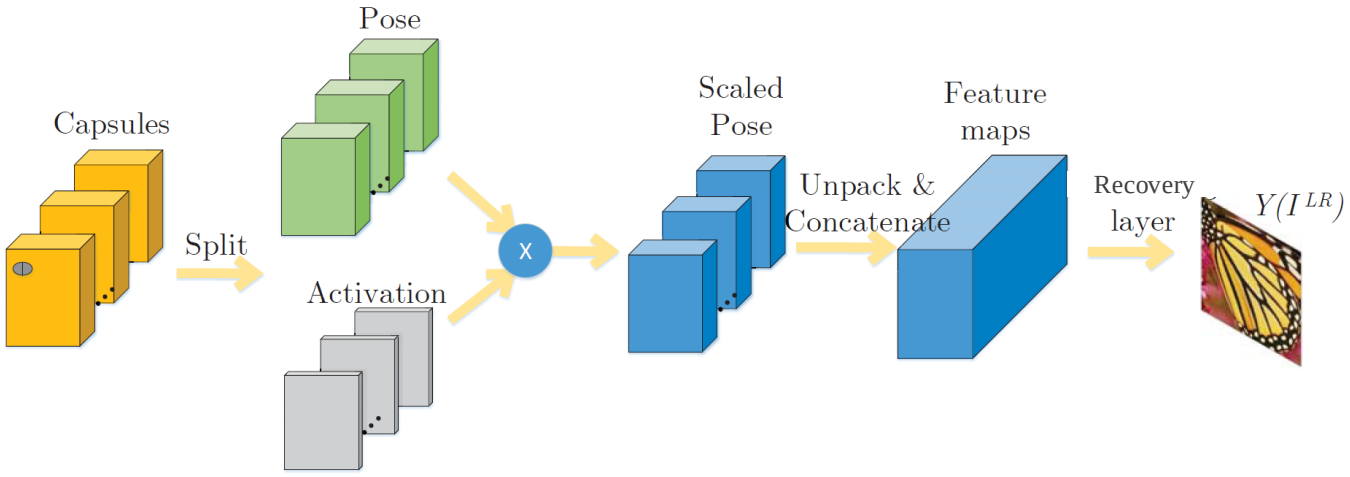

FIGURE 5. Reconstruction module for EMR in the CIRNN.

some common HR features, while the irrelevant candidates contain less related or even noisy features. The output capsule in the next layer is identified from target candidates through a routing algorithm. Fig. 1 shows an example where the output capsule is iteratively updated to approach the cluster center of the target candidates. This mechanism can be operated using either DR [10] or EMR [12].

Fig. 3 gives an overview of the CIRNN framework. The convolutional layer $Conv(f_1, n_1, c_1)$ generates n_1 feature maps from c_1 input channels with filters of size $f_1 \times f_1$. The primary capsule layer $PrimaryCaps(f_2, n_2, c_2, v_2)$ is basically a convolutional layer that generates $n_2 \times v_2$ feature maps from c_2 feature maps with filters of size $f_2 \times f_2$. The $n_2 \times v_2$ feature maps are then grouped into n_2 capsule maps, in which each capsule is formed by a vector or a matrix containing v_2 elements.

Convolutional capsule layers play a vital role in finding the relationship between LR and HR features. Similar to the capsule scheme demonstrated in [12], we use ℓ convolutional capsule layers $ConvCaps(k_3, n_3, c_3, v_3)$ that route c_3 input capsule maps, inside a $k_3 \times k_3$ sliding window, to n_3 output capsule maps, wherein each capsule contains v_3 elements. The parameter set (k_3, n_3, c_3, v_3) is empolyed for all ℓ convolutional capsule layers. The moving stride of the sliding window is set to 1 to keep the spatial size unchanged. We set the number of routing iteration for the convolutional capsule layer to 3, as in [10] [12]. A capsule is represented as a vector with v_3 elements, which can be processed directly in the DR scheme. To apply the EMR scheme, we reshape the vector capsule into a $\sqrt{v_3} \times \sqrt{v_3}$ matrix capsule.

The *Reconstruction Module* in Fig. 3 generates SR images from capsule maps. It involves different processes to match for different routing schemes. To reconstruct for the purpose of the DR scheme, capsule maps are directly unpacked into independent maps, which are then concatenated into feature maps, as shown in Fig. 4. To reconstruct for the purpose of

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.2020.2964292, IEEE Access

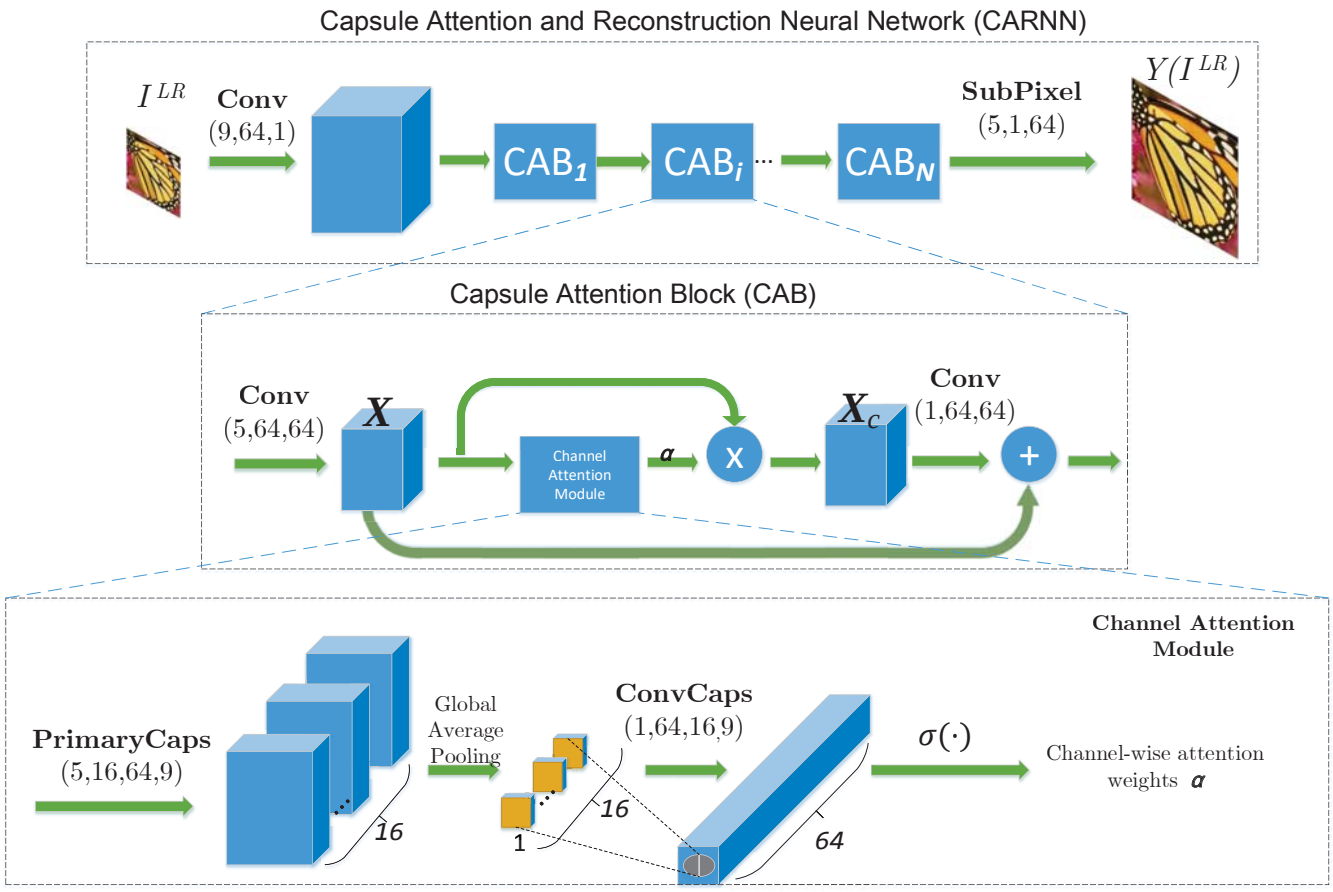

FIGURE 6. The framework of the proposed CARNN.

the EMR scheme, each capsule is first split into a pose matrix and a scalar activation. Each element in the pose matrix is scaled by the activation as presented in Fig. 5. The feature maps are formed by unpacking and concatenating the scaled pose maps. Then, the SR image is generated by applying an additional recovery layer for the feature maps. The adopted recovery layer is discussed in Section IV.

Given M taining sample pairs, the proposed networks are trained to minimize the cost function

$$L(\theta) = \frac{1}{M}\sum_{i=1}^{M} \|Y(I_i^{LR}) - I_i^{HR}\|_2^2 + \lambda\|\theta\|_2^2 \qquad (1)$$

where I_i^{LR} is the ith input image, I_i^{HR} is the corresponding ground truth data, λ is the weight decay parameter, $Y(I_i^{LR})$ is the output of the network, and the trainable parameter θ includes the filter weights in the convolutional layers and the transform matrices in the capsule layers.

B. CAPSULE ATTENTION AND RESTORATION NEURAL NETWORK (CARNN)

Unlike the CIRNN that directly generates an SR image from capsules, CARNN uses capsule layers to capture the channel attention. In a conventional attention network, attention weights are generated by a separate neural sub-network to recalibrate the input feature maps. In this work, we take the advantage of rich information encoded in capsules to obtain more accurate channel weights. The activation component of the routed capsule is assigned as an attention weight since it represents the importance of a specific channel. Fig. 6 illustrates the CARNN framework, which consists of a feature extraction convolutional layer, N Capsule Attention Blocks (CABs) and a sub-pixel convolutional layer. Each CAB contains a channel attention module, which takes feature maps $\boldsymbol{X} \in \mathbb{R}^{H\times W\times 64}$ as the input, and output $\boldsymbol{\alpha} \in \mathbb{R}^{64}$ as the channel attention weights. The residual feature map $\boldsymbol{X}_c$ is the scaled version of $\boldsymbol{X}$ by

$$\boldsymbol{X}_c = f_c(\boldsymbol{X}, \boldsymbol{\alpha}) = [\boldsymbol{x}_1 \circ \alpha_1, \boldsymbol{x}_2 \circ \alpha_2, ..., \boldsymbol{x}_{64} \circ \alpha_{64}] \qquad (2)$$

where $\boldsymbol{x}_i \circ \alpha_i$ denotes scaling the i-th channel $\boldsymbol{x}_i \in \mathbb{R}^{H\times W}$ by the i-th scalar element α_i of the vector $\boldsymbol{\alpha}$. The residual learning method is applied to stablize the training convergence. After the serial operations of N CAB modules, the SR image is reconstructed using a sub-pixel convolutional layer [7].

In each channel attention module, the primary capsule layer $PrimaryCaps$ and convolutional capsule layer

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.2020.2964292, IEEE Access

$ConvCaps$ function similarly to the CIRNN. The EMR [12] algorithm is applied in each convolutional capsule layer. In the module, the primary capsule layer uses 5×5 convolutional kernels to generate 16 capsule maps, wherein each capsule contains 9 elements. These 16 capsule maps are down-sampled through the use of global average pooling to generate exact 16 capsules as the input of the following convolutional capsule layer. If the candidates have more features in common, which means they are located closer in the capsule space, the output capsule of the routing process will gain a larger value in terms of its activation component. By applying the sigmoid activation function $\sigma(\cdot)$, the channel attention module outputs the channel weights $\boldsymbol{\alpha}$.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed networks, we implement CIRNN and CARNN using the publicly available package Matrix-Capsules-EM-Tensorflow [22]. The details of the dataset, training procedure and testing results are provided in the following subsections.

A. DATASET

We chose DIV2K [23] to be the training dataset. Each image in DIV2K is down-sampled and patched into 19×19 sub-images. The entire training dataset comprises approximately 1.3 million samples. We set the batch size to 16. We used General-100 [6] and Yang91 [1] as the validation datasets. To compare with other state-of-the-art approaches, Set14 [24] and BSD100 [25] are used as the testing datasets. We only considered the luminance component for the performance comparison, since the human visual system is more sensitive to it.

B. TRAINING METHOD

We empirically set both the learning rate and the weight decay parameter λ to 10^{-4} for all neural network layers. The filters of convolutional layers and transform matrices of capsule layers in the CIRNN and CARNN are initialized using the Xavier method [26]. All parameters are updated through the ADAM optimizer [27].

C. NETWORK ANALYSIS

We first conducted a small experiment by replacing the first two convolutional layers of the FSRCNN with two capsule layers, including one primary layer and one convolutional layer. We managed to keep the same number of parameters by setting the capsule layers as $PrimaryCaps(1, 3, 56, 4)$ and $ConvCaps(3, 3, 3, 4)$, respectively. The results showed that the modified network with capsules was higher than the original FSRCNN by 0.06 dB in PSNR.

To further explore the potential enhancement, we investigate various settings for the CIRNN. The combinations are listed in Table 1. The bicubic pre-upsampling setting in the second column refers to interpolating the input samples to match the target scaling size before feeding them into the neural network. Since the input had already been upsampled, we simply applied a convolutional layer as the Recovery Layer in the Reconstruction Module as shown in Fig. 4 and Fig. 5. As for the $Deconv$ and $SubPixel$ Recovery Layer settings in the third column, we directly input the LR samples into the neural network, and then recovered the HR image using a single deconvolutional or sub-pixel convolutional layer, respectively, with 5×5 filters in the last stage. The DR and EMR settings in the fourth column indicates the routing algorithm adopted by the convolutional capsules. It should be noted that for the DR in the convolutional capsule layer, we implemented a sliding window exactly like the EMR [12] scheme.

In this work, we set the CIRNN for the first convolution layer with parameter $(f_1, n_1, c_1) = (9, 32, 1)$, the primary capsule layers with $(f_2, n_2, c_2, v_2) = (1, 4, 32, 9)$, and the other convolutional capsule layers with $(k_3, n_3, c_3, v_3) = (3, 4, 4, 9)$. We first evaluated different recovery and routing algorithms for the network with only the $\ell = 1$ convolutional capsule layer. This model did not include either skip-connections or dense-connections, as presented in Fig. 3, since it is the simplest model by which to evaluate the recovery and routing algorithms. Comparing the second and the third rows in Table 1, it can be seen that the model using EMR performs better than the model using DR based on the peak-signal-to-noise ratio (PSNR) gain of 0.17 dB. This result shows that EMR generates more accurate capsules as target HR candidates.

We further tested different Recovery layers to be operated with EMR. The results are shown in the third to the fifth rows of Table 1. The model with the sub-pixel layer [7] outperformed the models with the convolutional and deconvolutional layers by 0.12 and 0.11 dB, respectively. We thus concluded that the combination of the sub-pixel convolutional layer and EMR had the best performance. Therefore, based on the results, we adopted the above settings to design deeper models.

For the deeper models, the primary capsules and the convolutional capsules were densely connected and fed to the $Reconstruction\ Module$. In this manner, we could reuse the output from previous layers. Skip-connection was applied between the convolutional capsule layers for training stability and convergence. We denoted the deeper models with the skip- and dense-connections as CIRNN+SDC to distinguish them from the previous model without such connections. In CIRNN+SDC, the convolutional layer was set to extract more features with $n_1 = 64$,and convolutional capsule layers were set for different values of n_2, n_3 and ℓ in order to evaluate the effects. The results are shown in the last three rows of Table 1, where it can be seen that the combination $(n_2 = n_3 = 4, \ell = 5)$ performed best. The concatenation with more convolutional capsule layers provided richer representations to CIRNN-SDC, and which in turn led to better performance.

For the CARNN, we first evaluated the performance of the channel attention module in the CARNN with only one CAB. We turned off the channel attention module by setting every element of the channel attention weights $\boldsymbol{\alpha}$ to 1. Table 2

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.2020.2964292, IEEE Access

TABLE 1. Average PSNR values on the validation datasets with a scale factor of 3 for the CIRNN with different settings.

	Pre-upsampling	Recovering layer	Routing	Average PSNR (dB)
CIRNN($n_1 = 32, n_2 = n_3 = 4, \ell = 1$)	Bicubic	$Conv$	DR	31.16
	Bicubic	$Conv$	EMR	31.33
	None	$Deconv$	EMR	31.34
	None	$SubPixel$	EMR	31.45
CIRNN+SDC($n_1 = 64, n_2 = n_3 = 3, \ell = 3$)	None	$SubPixel$	EMR	31.81
CIRNN+SDC($n_1 = 64, n_2 = n_3 = 4, \ell = 3$)	None	$SubPixel$	EMR	31.88
CIRNN+SDC($n_1 = 64, n_2 = n_3 = 4, \ell = 5$)	None	$SubPixel$	EMR	31.94

TABLE 2. Average PSNR values on the validation datasets with a scale factor of 3 for the CARNN with N CAB

	Average PSNR (dB)
CARNN ($N = 1$) w/o channel attention	31.54
CARNN ($N = 1$)	31.61
CARNN ($N = 3$)	31.70
CARNN ($N = 5$)	31.92

shows that the model with the channel attention module outperformed the model without a channel attention module by 0.07 dB, which demonstrated the robustness of the proposed capsule attention network. Then, we investigated the influence of the number of blocks N. As shown in Table 2, the CARNN with more CABs led to better performance since the deeper network was able to extract finer features for the purpose of reconstructing HR images. Therefore, we chose the CARNN($N = 5$) as the test model for the subsequent experiments.

D. COMPARISON WITH STATE-OF-THE-ART METHODS

Table 3 provides the quantitative comparisons on the commonly used dataset. We compared the proposed CIRNN and CARNN methods with state-of-the-art methods, including A+ [28], SRCNN [2], RFL [29] and FSRCNN [6]. PSNR and the Structural Similarity (SSIM) indices were used as the quality measurement. In addition, we present the qualitative results in Figs. 7, 8 and 9. The CIRNN performed better on Set5 in terms of both the PSNR and SSIM indices, while the CARNN outperformed the others on Set14 in the PSNR index. In addition, the channel attention mechanism may not benefit all datasets on super-resolution tasks. The same results can be found in [30]. However, when testing on the BSD100, the proposed CIRNN and CARNN methods performed slightly worse than the FSRCNN, but were still better than many other SR methods.

Table 3 also lists the number of parameters for each model with a scaling factor of 3. The CIRNN contained around 63K parameters, of which 77% were used on the Recovery layer in the $Reconstruction\ Module$. The feature maps of the previous layers were densely concatenated and resulted in a significant amount of parameter costs. Compared to the SRCNN [2], the proposed CIRNN method greatly improved on Set5 and Urban100 by only 6K more parameters. The CARNN achieved similar performance to that of the FSRCNN and CIRNN. However, the CARNN used much more parameters since we did not apply techniques such as expanding and shrinking, as proposed in [6]. The shrinking layer reduces the input feature dimension before mapping for the sake of computational efficiency, and the expanding layer expands the HR feature dimension in order to improve the restoration quality. The design of the CARNN is still rudimentary, yet it shows the effectiveness of the proposed capsule attention module.

We noted that the proposed methods performed less effectively than the FSRCNN on some testing datasets,such as the BSD100. This might have been because this kind of testing dataset contains too many different themes. The limited size of the transform matrices in the capsule layers may result in failure to generate the target candidates necessary for the routing process to select an accurate capsule. A possible solution for this issue would be to significantly increase the number of convolutional capsule channels to raise the probability of producing target candidates.

V. CONCLUSION

In this paper, we proposed applying a capsule neural network to CNN-based image super-resolution algorithms. The CIRNN employs rich information contained in the capsules to directly reconstruct the images. Different from the CIRNN, the CARNN applies the attention module to recalibrate the feature maps. Both of the proposed methods could achieve results comparable to recent CNN-based methods.

We plan in the future to speed up the proposed framework with more capsule channels. Compared to a conventional CNN, a capsule neural network requires significantly more extra computational power for performing iterative routing. Some faster methods other than the EM algorithm should be developed for efficient capsule voting to implement a deeper capsule neural network for SR.

REFERENCES

[1] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," IEEE Transactions on Image Processing, vol. 19, no. 11, pp. 2861–2873, 2010.

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.2020.2964292, IEEE Access

TABLE 3. Quantitative comparison with the state-of-the-art methods for the testing dataset (PSNR(dB)/SSIM). The best and the second best results are marked in red and blue, respectively. The parameters of the models are also listed for comparison.

Dataset	Scale	Bicubic	A+	RFL	SRCNN	FSRCNN	CIRNN (Ours)	CARNN (Ours)
Set5	x2	33.66/0.9288	36.54/0.9544	36.54/0.9537	36.66/0.9542	37.00/0.9558	37.15/0.9563	37.06/0.9558
	x3	30.39/0.8682	32.58/0.9088	32.43/0.9057	32.75/0.9090	33.16/0.9088	33.37/0.9149	33.22/0.9126
	x4	28.42/0.8104	30.28/0.8603	30.14/0.8548	30.48/0.8628	30.71/0.8646	31.34/0.8757	31.10/0.8725
Set14	x2	30.24/0.8688	32.28/0.9056	32.26/0.9040	32.42/0.9063	32.63/0.9086	32.76/0.9091	32.78/0.9089
	x3	27.55/0.7742	29.13/0.8188	29.05/0.8164	29.28/0.8209	29.42/0.8243	29.37/0.8230	29.44/0.8226
	x4	26.00/0.7027	27.32/0.7491	27.24/0.7451	27.49/0.7503	27.59/0.7539	27.57/0.7545	27.61/0.7547
BSD100	x2	29.56/0.8403	31.21/0.8863	31.16/0.8840	31.36/0.8855	31.50/0.8908	31.44/0.8907	31.47/0.8909
	x3	27.21/0.7349	28.29/0.7835	28.22/0.7806	28.41/0.7863	28.51/0.7900	28.32/0.7877	28.39/0.7881
	x4	25.96/0.6577	26.82/0.7087	26.75/0.7054	26.90/0.7101	26.96/0.7138	26.74/0.7110	26.81/0.7122
Urban100	x2	26.88/0.8403	29.20/0.8938	29.11/0.8904	29.50/0.8946	29.85/0.9011	30.12/0.9041	30.06/0.9033
	x3	24.46/0.7349	26.03/0.7973	25.86/0.7900	26.24/0.7989	26.41/0.7353	26.46/0.8068	26.49/0.8064
	x4	23.14/0.6577	24.32/0.7183	24.19/0.7096	24.52/0.7221	24.60/0.7267	24.62/0.7271	24.69/0.7294
Parameters		-	-	-	57K	12K	63K	1.88M

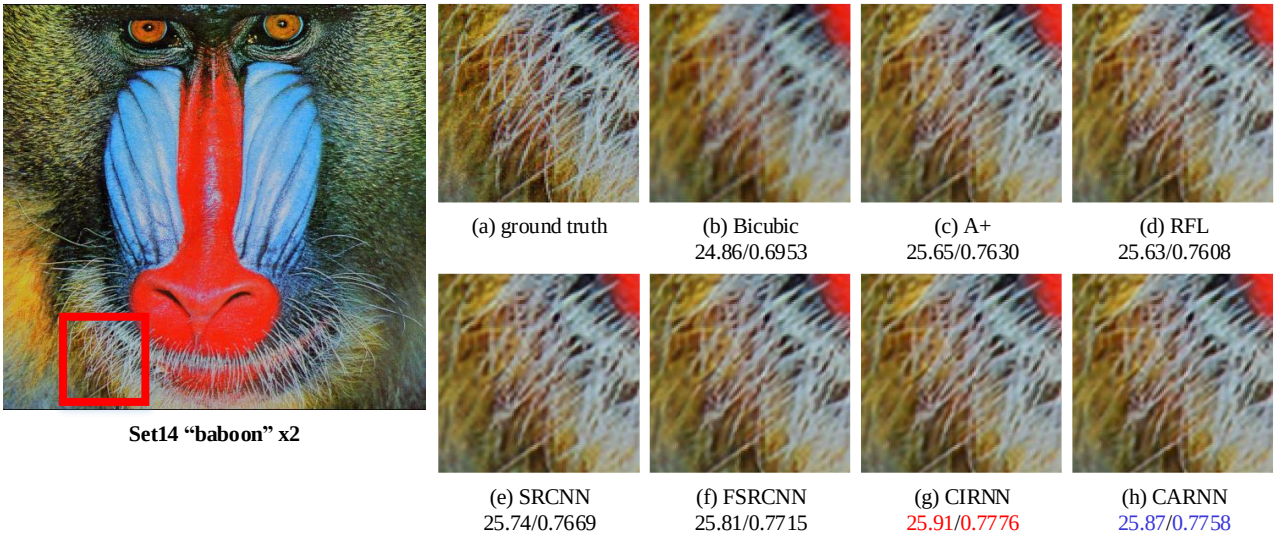

FIGURE 7. The reconstruction results of “baboon” image from Set14 with a scale factor of 2.

[2] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in European conference on computer vision, pp. 184–199, Springer, 2014.

[3] J. Kim, J. Kwon Lee, and K. Mu Lee, “Deeply-recursive convolutional network for image super-resolution,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1637–1645, 2016.

[4] Y. Tai, J. Yang, and X. Liu, “Image super-resolution via deep recursive residual network,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3147–3155, 2017.

[5] J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR Oral), June 2016.

[6] C. Dong, C. C. Loy, and X. Tang, “Accelerating the super-resolution convolutional neural network,” in European conference on computer vision, pp. 391–407, Springer, 2016.

[7] 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 Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1874–1883, 2016.

[8] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, “Fast and accurate image super-resolution with deep laplacian pyramid networks,” IEEE transactions on pattern analysis and machine intelligence, 2018.

[9] M. Haris, G. Shakhnarovich, and N. Ukita, “Deep back-projection networks for super-resolution,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1664–1673, 2018.

[10] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” in Advances in Neural Information Processing Systems, pp. 3856–3866, 2017.

[11] A. Shahroudnejad, P. Afshar, K. N. Plataniotis, and A. Mohammadi, “Improved explainability of capsule networks: Relevance path by agreement,” in 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 549–553, IEEE, 2018.

[12] G. E. Hinton, S. Sabour, and N. Frosst, “Matrix capsules with em routing,” in 6th International Conference on Learning Representations, ICLR, 2018.

[13] G. E. Hinton, A. Krizhevsky, and S. D. Wang, “Transforming auto-encoders,” in International Conference on Artificial Neural Networks, pp. 44–51, Springer, 2011.

[14] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov, “Generating images from captions with attention,” arXiv preprint arXiv:1511.02793, 2015.

[15] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation

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.2020.2964292, IEEE Access

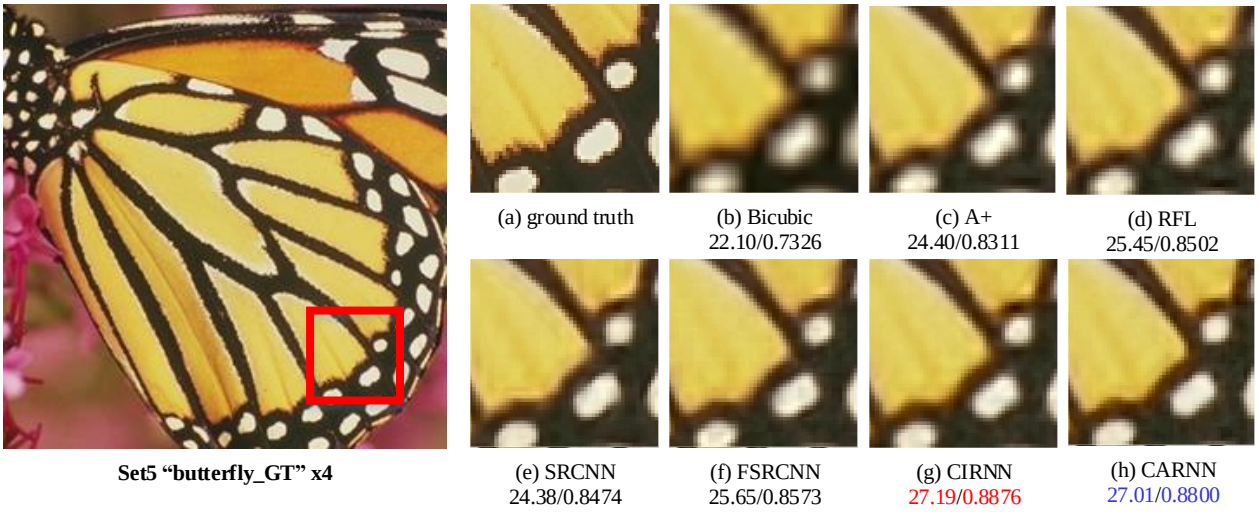

FIGURE 8. The reconstruction results of “butterfly_GT” image from Set5 with a scale factor of 4.

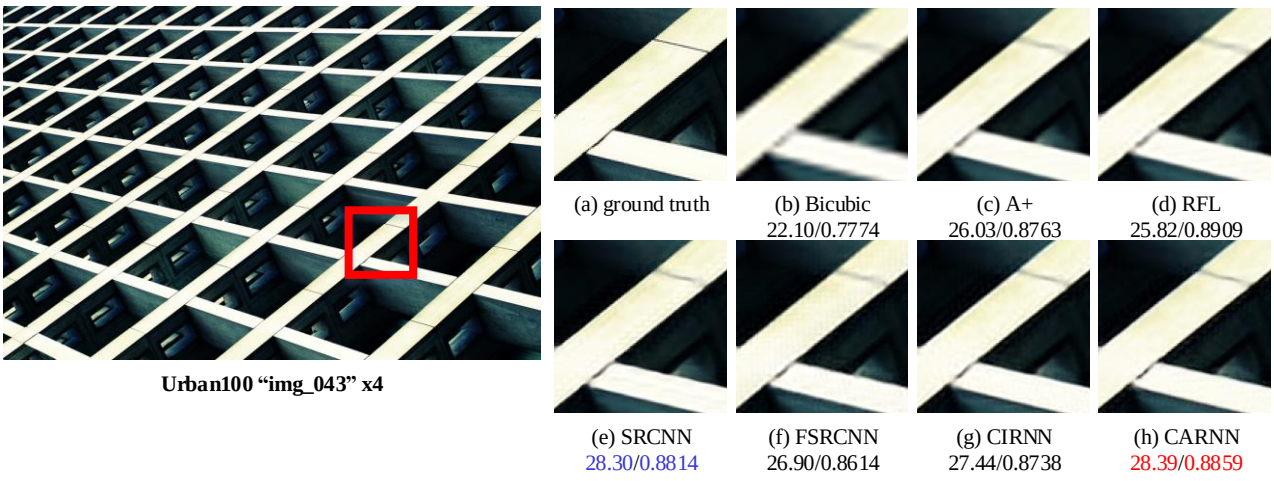

FIGURE 9. The reconstruction results of “img_043” image from Urban100 with a scale factor of 4.

with visual attention,” in International conference on machine learning, pp. 2048–2057, 2015.

[16] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, “Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5659–5667, 2017.

[17] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141, 2018.

[18] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in Proceedings of the IEEE conference on computer vision and pattern recognition. (CVPR), pp. 3156–3164, 2017.

[19] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image super-resolution using very deep residual channel attention networks,” in Proceedings of the European Conference on Computer Vision (ECCV), pp. 286–301, 2018.

[20] X. Wang, K. Yu, C. Dong, and C. Change Loy, “Recovering realistic texture in image super-resolution by deep spatial feature transform,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 606–615, 2018.

[21] Y. Hu, J. Li, Y. Huang, and X. Gao, “Channel-wise and spatial feature modulation network for single image super-resolution,” IEEE Transactions on Circuits and Systems for Video Technology, 2019.

[22] Matrix-Capsules-EM-Tensorflow, “https://github.com/www0wwwjs1/matrix-capsules-em-tensorflow,”

[23] E. Agustsson and R. Timofte, “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, vol. 3, p. 2, 2017.

[24] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using

sparse-representations," in International conference on curves and surfaces, pp. 711–730, Springer, 2010.

[25] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," IEEE transactions on pattern analysis and machine intelligence, vol. 33, no. 5, pp. 898–916, 2011.

[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249–256, 2010.

[27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[28] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in Asian conference on computer vision, pp. 111–126, Springer, 2014.

[29] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3791–3799, 2015.

[30] J.-H. Kim, J.-H. Choi, M. Cheon, and J.-S. Lee, "Ram: Residual attention module for single image super-resolution," 2018.

FIRST A. AUTHOR [Jui-Ting Hsu] was born in Kaohsiung, Taiwan in 1993. He received the B.S. degree in electrical engineering from National Central University, Taoyuan, Taiwan and the M.S. degree in electrical engineering National Cheng Kung University, Tainan, Taiwan. His research interests include deep learning, image processing, and so on.

SECOND B. AUTHOR [Chih-Hung Kuo] (S'01-M'04) received the B.S. and M.S. degrees from the National Tsing Hua University, Hsinchu, Taiwan, in 1992 and 1994, respectively, and the Ph.D. degree from the University of Southern California (USC), Los Angeles, USA, in 2003, all in Electrical Engineering.

He was with Computer and Communications Research Laboratories/Industrial Technology Research Institute (CCL/ITRI), Taiwan, as a DSP Design Engineer from 1996 to 1998. From March 2004, he was a Senior Engineer in Winbond Electronics, Taiwan. In August 2004, he joined the Department of Electrical Engineering of the National Cheng Kung University, Tainan, Taiwan, as an Assistant Professor. He becomes an Associate Professor since February 2010. His current research interests include system-level designs for video processing and multimedia communications.

THIRD C. AUTHOR [De-Wei Chen] was born in Changhua, Taiwan in 1994. He received the B.S. degree from National Central University, Taoyuan, Taiwan, in Electrical Engineering and did a M.S. degree in Electrical Engineering from National Cheng Kung University, Tainan, Taiwan. His research interests include deep learning and image processing.