Cascaded Deep Networks with Multiple Receptive Fields for Infrared Image Super-Resolution

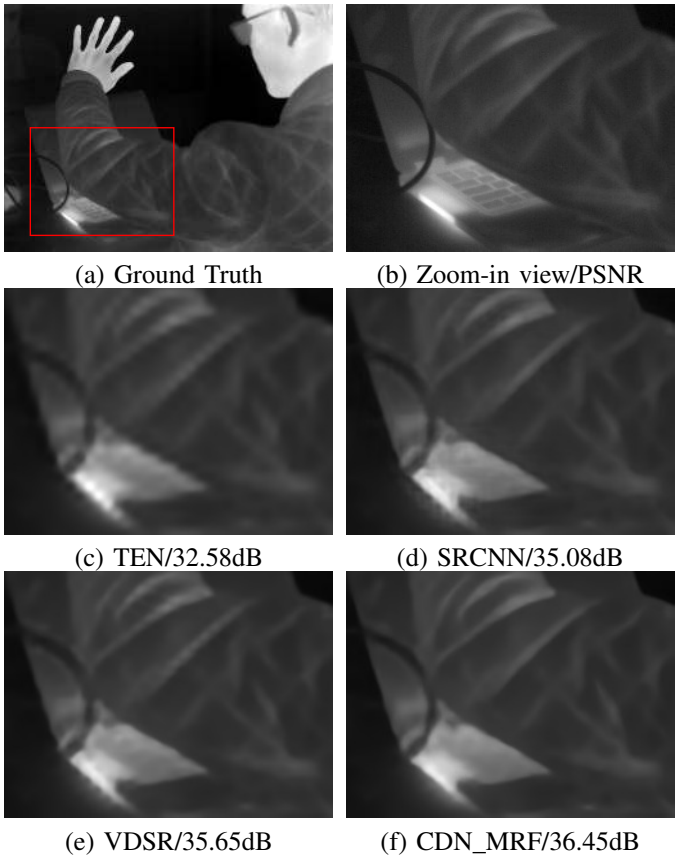
Zewei He, *Student Member, IEEE,* Siliang Tang, *Member, IEEE,* Jiangxin Yang, Yanlong Cao, Michael Ying Yang, *Senior Member, IEEE,* and Yanpeng Cao, *Member, IEEE*

***Abstract*—Infrared images have a wide range of military and civilian applications including night vision, surveillance and robotics. However, high-resolution infrared detectors are difficult to fabricate and their manufacturing cost is expensive. In this work, we present a cascaded architecture of deep neural networks with multiple receptive fields to increase the spatial resolution of infrared images by a large scale factor ($\times 8$). Instead of reconstructing a high-resolution image from its low-resolution version using a single complex deep network, the key idea of our approach is to set up a mid-point (scale $\times 2$) between scale $\times 1$ and $\times 8$ such that lost information can be divided into two components. Lost information within each component contains similar patterns thus can be more accurately recovered even using a simpler deep network. In our proposed cascaded architecture, two consecutive deep networks with different receptive fields are jointly trained through a multi-scale loss function. The first network with a large receptive field is applied to recover large-scale structure information, while the second one uses a relatively smaller receptive field to reconstruct small-scale image details. Our proposed method is systematically evaluated using realistic infrared images. Compared with state-of-the-art Super-Resolution methods, our proposed cascaded approach achieves improved reconstruction accuracy using significantly less parameters.**

***Index Terms*—infrared imaging, super-resolution, cascaded architecture, deep networks, receptive fields.**

I. Introduction

INFRARED imaging technology provides valuable thermal information to facilitate a wide range of important applications including thermal analysis, video surveillance, medical diagnosis, and remote sensing. To achieve high-accuracy thermal measurement, infrared detectors are encapsulated in individual vacuum packages which is a time-consuming and expensive process [1]. As the consequence, infrared sensors are significantly more expensive than visible ones of similar resolution. Given low-resolution (LR) infrared images, we focus on developing effective algorithms to restore thermal details which are essential to enable reliable target detection and recognition tasks but only available in high-resolution (HR) infrared images.



(a) Ground Truth (b) Zoom-in view/PSNR

(c) TEN/32.58dB (d) SRCNN/35.08dB

(e) VDSR/35.65dB (f) CDN_MRF/36.45dB

Fig. 1. Comparative results of a number of deep-learning-based SR methods including TEN [2], SRCNN [3], VDSR [4] and our proposed CDN_MRF method. It is observed that CDN_MRF method more accurately restores original image information without causing undesirable artifacts. Compared with state-of-the-art SR method (VDSR), our proposed CDN_MRF achieves higher Peak Signal-to-Noise Ratio (PSNR) value using significantly less parameters.

Single image based super-resolution (SR) is a promising technique to increase the spatial resolution of optical sensors [5]–[8]. Given a LR image, SR aims at reconstructing a higher resolution image through solving an ill-posed inverse problem [9], [10]. Due to the great performances achieved by deep learning based methods for many computer vision applications such as image classification, target detection, and object recognition, researchers start to design deep neural networks (DNNs) for learning the mapping relationship between LR

This work was supported in part by National Natural Science Foundation of China (No. 51605428, 51575486 and U1664264), and in part by the Fundamental Research Funds for the Central Universities. (Corresponding author: Yanpeng Cao, Email: caoyp@zju.edu.cn)

Zewei He, Jiangxin Yang, Yanlong Cao and Yanpeng Cao are with State Key Laboratory of Fluid Power and Mechatronic Systems and Laboratory of Advanced Manufacturing Technology of Zhejiang Province, School of Mechanical Engineering, Zhejiang University, Hangzhou, 310027, China.

Siliang Tang is with College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, China.

Michael Ying Yang is with Scene Understanding Group, ITC, Universiteit Twente, 3230 Enschede, Overijssel, Netherlands.

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/TCSVT.2018.2864777, IEEE Transactions on Circuits and Systems for Video Technology

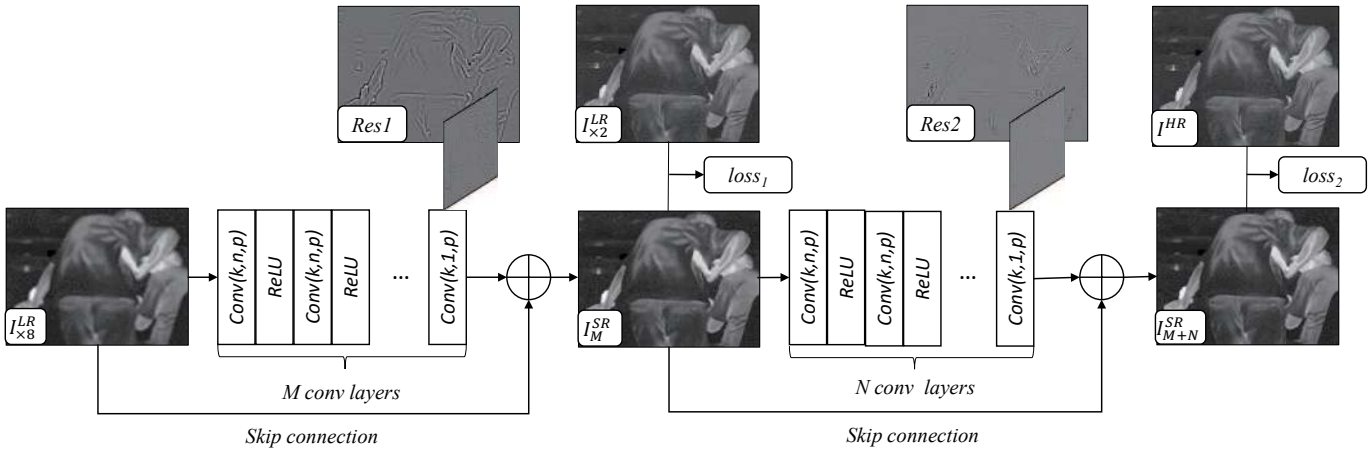


Fig. 2. Our proposed cascaded architecture of deep networks. I^{LR} is the LR input (upscaled to the size of HR output via bi-cubic interpolation), I^{SR}_{M} is the SR result of the first network, and I^{SR}_{M+N} denotes the final output. $Conv(k, n, p)$ indicates that the convolution uses n kernels of size $k \times k$ on the images/feature maps with padding p. Given a LR input, M convolutional layers and one skip connection are firstly applied to recover structure information and then another N convolutional layers and one skip connection are used to restore fine details. These two networks are jointly trained as an ensemble by minimizing the combination of multiple loss terms ($loss_1$ and $loss_2$).

and HR images [3], [11]–[16]. Although many successful SR methods have been proposed to increase resolution of visible images, the development of a DNN-based SR method working well for infrared images still remains an untackled problem. The major challenge is twofold. First, visible and infrared images present very different visual characteristics and it is not clear what is the optimal strategy to migrate a deep-learning-based SR method from visible spectrum to infrared one. Applying a SR model, which is trained in visible domain, to process infrared images cannot achieve satisfactory reconstruction results [2]. Second, existing deep-learning-based solutions [2]–[4], [12], [14] only demonstrate good reconstruction results for small scale factor SR ($\times 2$ or $\times 4$) which might not be sufficient for LR infrared images (e.g., 80×60). Typically, SR performance drops significantly when the scale factor increases since the original information in HR image has little or no evidence in its LR version [17].

In this paper, we present a cascaded architecture of deep networks with multiple receptive fields (CDN_MRF) to address the problem of single infrared image SR with a large scale factor ($\times 8$). It is observed that thermal information lost from scale factor $\times 1$ to $\times 2$ (fine details) is different from those lost from scale $\times 2$ to $\times 8$ (major structures). With this finding, we propose a two-level cascaded architecture of deep networks, as illustrated in Fig. 2, to gradually recover image information from scale $\times 8$ to $\times 2$ first and from scale $\times 2$ to $\times 1$ then. Instead of recovering all information lost from scale $\times 1$ to $\times 8$ using a single complex deep network, our approach set up a mid-point (scale $\times 2$) between scale $\times 1$ and $\times 8$ to divide lost information into two components. Lost information within each component has similar patterns thus can be more accurately recovered even using a simpler deep network. In addition, a multiple receptive fields strategy is adopted to deal with lost information of different scales. The first network uses a large receptive field to recover large-scale structure information from scale $\times 8$ to $\times 2$, while the second one considers information from a relatively smaller receptive field to reconstruct small-scale image details from scale $\times 2$ to $\times 1$. This strategy leads to further reduction of the complexity of our networks and higher SR reconstruction accuracy. Our experiments demonstrate that the proposed cascaded deep networks, using a significantly smaller number of parameters (1/10), can still achieve better performance compared with state-of-the-art deep-learning-based SR methods (VDSR). Some comparative results are shown in Fig. 1. The contributions of our work are two-fold.

- First, we build up a HR infrared image dataset (in total we captured 120 images of 640×480 pixels) covering a wide range of scenarios (e.g., vehicle, machine, pedestrian and building), and further present a DNN model to learn mapping relationship between LR and HR infrared images. To the best of our knowledge, this is the first attempt to make use of infrared data to solve its SR problem instead of using a model trained in visible spectral domain [2].
- Second, a cascaded architecture of deep networks with multiple receptive fields is proposed to achieve large scale factor ($\times 8$) infrared SR. The first network with a large receptive filed recovers most of the structure information and the second one uses relatively smaller receptive field to restore image fine details. Compared with the state-of-the-art deep-learning-based methods, our proposed CDN_MRF approach can achieve better SR accuracy with significantly less model parameters.

The remainder of the paper is organized as follows. We start by reviewing some existing classic and learning-based SR works in Sec. II. The details of our cascaded architecture CDN_MRF are presented in Sec. III. Extensive experimental results are presented in Sec. IV, and Sec. V concludes this paper.

II. Related work

Single image based SR is an under-determined inverse problem due to the fact that one LR image can correspond to multiple HR images. Classic machine learning methods such as neighbor embedding (NE) [18], [19], anchored neighborhood regression (ANR) [7] and sparse coding (SCSR) [8], [20], [21] attempt to constrain the solution space with prior information. In [18], SR is performed via a neighbor embedding algorithm with the assumption that the low-dimensional non-linear manifolds in LR and HR feature space have a similar local geometry. With enough samples, patches in the HR feature domain can be recovered as a weighted average of local neighbors using the weights calculated in the LR feature domain. To improve computational efficiency, Timofte *et al.* [7] utilized a number of linear regressors to anchor the neighborhood embedding of a LR patch to the nearest atom in the dictionary and to pre-compute the corresponding embedding matrix. Then the same authors proposed an improved variant of ANR which is built on the features and anchored regressors from ANR but uses the full training material [22] and summarized seven ways to improve SR performance [23]. Yang *et al.* [8], [20] assumed that LR patches share the same sparse representation with corresponding HR counterparts. After learning the LR and HR dictionaries, the sparse coefficients solved with the LR dictionary are then passed to corresponding HR dictionary for reconstructing HR patches. Several methods [9], [24] exploited the self-similarity prior that patches in a natural image tend to recur within and across scales of the same image. According to the fractal nature, an internal LR-HR patch dataset is built using the scale-space pyramid of the input image itself. However, the internal dictionary obtained from the dataset is not sufficient to handle large textural appearance variations. To overcome this drawback, SelfExSR method proposed by Huang *et al.* [5] expands the internal patch search space by allowing geometric variations. Although self-similarity based approaches do not require an training process, they involve time-consuming internal patch searching processes.

In recent years, deep learning has been successfully applied in various computer vision tasks (e.g., object classification [25], pedestrian detection [26], and image de-nosing [27]) and achieves breakthrough improvements. Many researchers attempt to solve the SR problem through the training of DNN models [3], [4]. An effective convolutional neutral network model (SRCNN) is proposed to learn the mapping relationship between LR and HR visible images [3]. With a three-layer lightweight structure, SRCNN still outperforms other learning-based methods (e.g., neighbor embedding (NE) [18], [19], Anchored Neighborhood Regression (ANR) [7] and sparse coding (SCSR) [8], [28]). It is noted that SRCNN directly learns the mapping relationship between LR/HR pairs and its training process takes a long time to converge. The same authors also developed a fast version (FSRCNN) to accelerate SRCNN [13] which achieves a real-time speed. VDSR [4] proposed a highly accurate SR method based on a very deep convolutional network (20 layers). VDSR firstly reconstructs the residual information and then adds it back to the LR image to generate the final SR output which is proven effective in achieve high SR accuracy. Using a large number of parameters, VDSR outperforms the other SR methods by a large margin. However, VDSR contains a large number of model parameters which are difficult to train and impractical for real-time implementation. Many state-of-the-art SR methods are reviewed and their performances are systematically evaluated in [29]. Choi et al. present the first deep learning based SR method for infrared images in which a SR model trained using visible spectral data is applied to enhance the spatial resolution of infrared images [2]. However the achieved improvement is quite limited even compared with the traditional bi-cubic interpolation method. Moreover, it is noticed most existing SR methods are designed for small scale factors (e.g., $\times 2$ and $\times 4$) [3], [15] which is not sufficient to process LR infrared images. It is common to capture LR images using 80×60 infrared detectors, thus it is desirable to develop a SR method with large scale factor ($\times 8$) which can convert LR images to 640×480 VGA resolution.

Previously, cascading strategy has been effectively applied to boost SR performance [15], [23], [30]. However, multiple stacked networks require to train/deploy more model parameters. Since information to restore within each network has similar patterns, it is possible to significantly reduce the number of network parameters without sacrificing SR accuracy. In addition, a multiple receptive fields strategy is adopted to deal with lost information of different scales. The first network uses a large receptive field to recover structure information from scale $\times 8$ to $\times 2$, while the second one considers information from a relatively smaller receptive field to reconstruct small-scale image details from scale $\times 2$ to $\times 1$. This strategy leads to further reduction of the complexity of our cascaded networks. As a result, our proposed CDN_MRF approach can achieve higher SR accuracy with significantly less model parameters.

III. Approach

In this paper we present a cascaded architecture of deep networks with multiple receptive fields (CDN_MRF) to address the challenging problem of large scale factor ($\times 8$) infrared image SR. For this purpose, we build up an infrared image dataset which consists of 120 HR images of 640×480 resolution. 100 images are utilized to train our cascaded deep networks and another 20 images are used for SR performance evaluation. More details of our captured infrared dataset are provided in Sec. IV-A.

A. Network architecture

In Fig. 3, we show a HR infrared image, its $\times 2$, $\times 4$ and $\times 8$ scale LR versions, and the residual images between different scales ($\times 1 \rightarrow \times 2$, $\times 2 \rightarrow \times 4$, $\times 4 \rightarrow \times 8$ and $\times 2 \rightarrow \times 8$). It is observed that the $\times 2$ LR image is visually similar to the original image. Structure information is still well preserved in $\times 2$ LR image and only some insignificant fine details are removed. The underlying reason for this phenomenon is that infrared images contains limited amount of fine details [31]. With the increase of the scale factor, structure information starts to disappear from scale $\times 2$ to $\times 4$ and is further reduced

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/TCSVT.2018.2864777, IEEE Transactions on Circuits and Systems for Video Technology

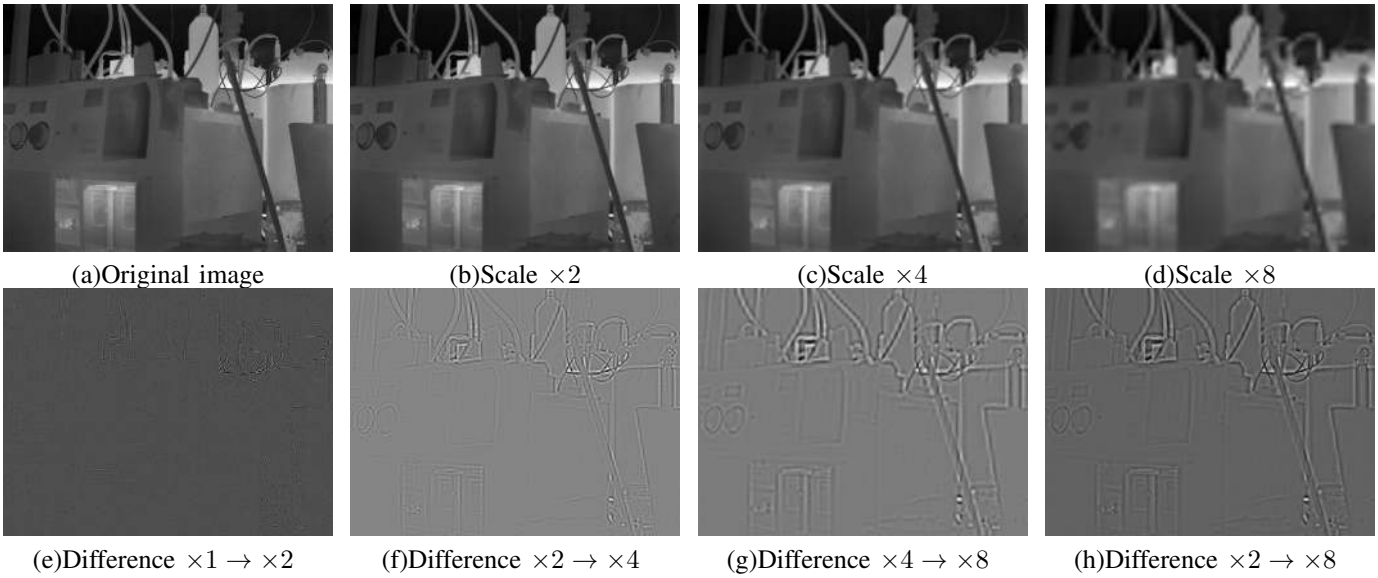


Fig. 3. Down-sampling images with different scale factors and information lost between different scales. Note all down-sampling images are resized through bi-cubic interpolation (using the image resize function - $imresize()$ provided in Matlab) to its original size and normalized to 0-1 value range for visualization.

from scale $\times 4$ to $\times 8$. As illustrated in Fig. 3, thermal information lost from scale $\times 2$ to $\times 4$ (Fig. 3 (f)) and from scale $\times 4$ to $\times 8$ (Fig. 3 (g)) both consist of major image structures, which are significantly different from fine details lost from scale factor $\times 1$ to $\times 2$ (Fig. 3 (e)). In addition, we calculate the histogram of the residual images between different scales ($\times 1 \rightarrow \times 2$, $\times 2 \rightarrow \times 4$, $\times 4 \rightarrow \times 8$ and $\times 1 \rightarrow \times 8$), and the comparative results on 100 training images are shown in Fig.4. It is observed that the residual images $\times 2 \rightarrow \times 4$ and $\times 4 \rightarrow \times 8$ have very similar data distributions and both contain a certain number of large value components (corresponding to high-variance structure edges). In comparison, the difference between $\times 1$ and $\times 2$ images is insignificant and the residual image $\times 1 \rightarrow \times 2$ mostly consists of small value components (corresponding to low-variance image details).

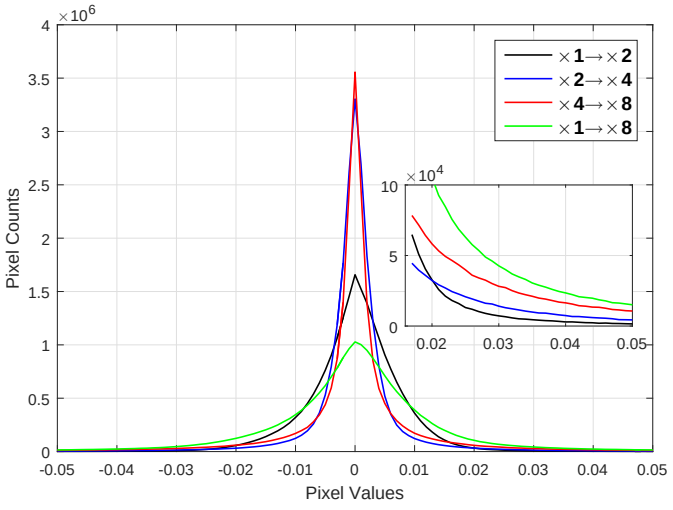


Fig. 4. The histograms of the residual images $\times 1 \rightarrow \times 2$, $\times 2 \rightarrow \times 4$, $\times 4 \rightarrow \times 8$ and $\times 1 \rightarrow \times 8$ calculated on 100 training images.

With this interesting observation, we argue that performance of large scale factor SR can be substantially improved by applying the divide-and-conquer philosophy. Instead of directly learning the mapping relationship between LR (scale $\times 8$) and HR (scale $\times 1$) images which is difficult to achieve high reconstruction accuracy and requires a complex deep network, a mid-point (scale $\times 2$) is set up between scale $\times 1$ and $\times 8$ to divided lost information into two components of different characteristics. Accordingly we deploy two different deep network models to recover major structures ($\times 2 \rightarrow \times 8$) and fine details ($\times 1 \rightarrow \times 2$) respectively. Since similar patterns exist within each component, they can be more accurately restored even using two simpler deep networks. We systematically evaluate the effectiveness of the proposed cascaded architecture ($\times 8 \rightarrow \times 2 \rightarrow \times 1$) in Sec. III-C.

Inspired by the popular VGG-net [25] and Residual net [32], we use two deep networks and cascade them as an ensemble to gradually reconstruct HR image I^{SR} from a LR image I^{LR}. Our cascaded architecture is illustrated in Fig. 2. Given a LR input, we firstly upscale it to the size of HR image through bi-cubic interpolation for deep network training [12]. To avoid confusion, we still call it a "low-resolution" image, although it has the same size of the HR image. For low-contrast infrared images, the differences between LR and HR images is insignificant and only a small numbers of pixels have non-zero differences. Therefore, computing the residual images is a more efficient way to depict their differences, and can lead to much faster convergence with higher accuracy [4].

The first deep network consists of M convolutional layers and one skip connection. The first deep network take LR image (scale $\times 8$) $I^{LR}_{\times 8}$ as input and predict its scale $\times 2$ version as

$$I^{SR}_1 = max(0, w_1 * I^{LR}_{\times 8} + b_1), \quad (1)$$

$$I^{SR}_n = max(0, w_n * I^{SR}_{n-1} + b_n), n \in \{2, 3...M-1\}, \quad (2)$$

$$I^{SR}_M = (w_M * I^{SR}_{M-1} + b_M) + I^{LR}_{\times 8}, \quad (3)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2018.2864777, IEEE Transactions on Circuits and Systems for Video Technology

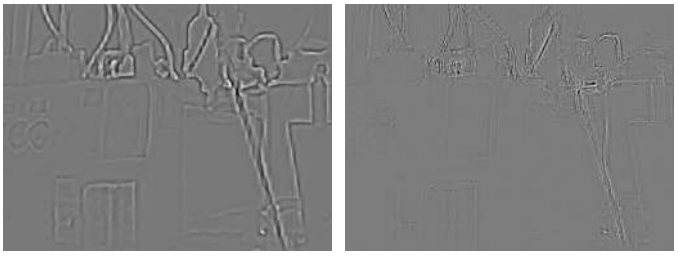
where w_n and b_n represent the filtering weights and biases respectively, $*$ denotes the convolution operation. We apply Rectified Linear Unit (ReLU) activation function (i.e., max(0,x)) [33] on the results of the convolutions. We use the first network to learn a mapping from LR image $I^{LR}_{\times 8}$ (scale $\times 8$) to a scale $\times 2$ intermediate image. In this step, major image structures are recovered as shown in Fig. 5 (a).

Consecutively, we deploy another deep network to learn a mapping function from I^{SR}_M to the original image I^{HR}. The second network consists of N layers and one skip connection. It takes the computed image I^{SR}_M (output from the first deep network) as the input and computes I^{SR}_{M+N} as

$$I^{SR}_n = max(0, w_n * I^{SR}_{n-1} + b_n), n \in \{M+1...M+N-1\}, \quad (4)$$

$$I^{SR}_{M+N} = (w_{M+N} * I^{SR}_{M+N-1} + b_{M+N}) + I^{SR}_M, \quad (5)$$

The restored I^{SR}_{M+N} should be as similar as possible to the ground truth image I^{HR}. Different from the first deep network, the second model attempts to recover some image fine details as shown in Fig. 5 (b). Please note these two deep networks are not identical and have different receptive fields. We will discuss possible techniques to optimize cascaded deep networks in Sec. III-C.



(a) Res1 (b) Res2

Fig. 5. Learned residual images of our proposed cascaded deep networks. (a) The first residual image contains rich structure information and (b) The second residual image consists of some texture details. Both of the images are normalized to 0-1 value range for better visualization.

B. Network Training

To train our cascaded deep networks, we need a large number of infrared image patches of low- (scale $\times 8$), middle- (scale $\times 2$) and high-resolutions (original image). We randomly crop a large number of image patches from HR infrared images and then apply some standard data augmentation methods (e.g., rotation and flip) to expand the training dataset. For each HR image patch I^{HR}, we perform down-sampling by a factor of 2 and 8 to get its LR version $I^{LR}_{\times 2}$ and $I^{LR}_{\times 8}$ and then upscale them to the size of HR image through bi-cubic interpolation.

Low-contrast infrared images usually contain limited amount of textures [31]. As a result, many cropped image patches cover a homogeneous region and contain pixels of similar gray values as shown in Selection A in Fig. 6. If the training dataset contains lots of such sample images, the deep network will be tuned to learn mapping relationships between these homogeneous regions instead of recovering lost high-frequency signals. As a simple yet effective solution, we compute the standard variation of pixels within an image patch to decide whether this patches is suitable for training. A threshold θ is set to selection patches with high intensity variations. Only image patches from Section B in Fig. 6 will be used for deep network training. We evaluate the effectiveness of this strategy using two different deep networks including SRCNN [3] and VDSR [4]. As shown in Tab. I, this verification strategy is an effective technique to generate valid training patches and leads to SR performance boost for both models.

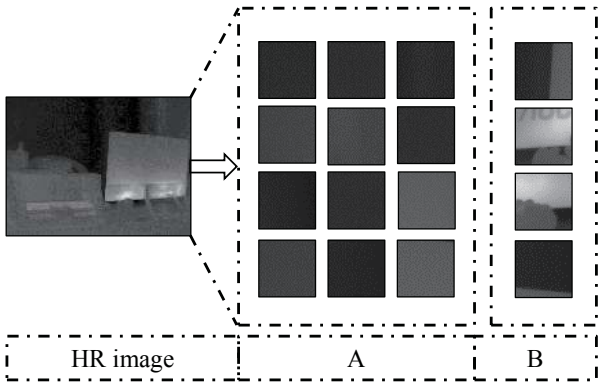


Fig. 6. A HR infrared image is cropped into a number of small patches. Selection A: image patches contain pixels of similar intensity values. Selection B: image patches contain image edges and textures. We only consider patches from selection B to train our deep networks.

TABLE I
THE SR RESULTS WITH AND WITHOUT TRAINING IMAGE PATCH SELECTION. THE PSNR AND SSIM VALUES ARE CALCULATED ON OUR TESTING DATASET WHICH CONTAINS 20 INFRARED IMAGES.

	SRCNN [3]		VDSR [4]	
	w/o	with	w/o	with
PSNR (dB)	35.22	35.32	35.51	35.65
SSIM	0.9154	0.9157	0.9186	0.9198

Given training images $I^{LR}_{\times 8}$, $I^{LR}_{\times 2}$ and I^{HR}, weights $w_{1:M+N}$ and the biases $b_{1:M+N}$ are computed by jointly minimizing the weighted sum of two loss functions $loss_1$ and $loss_2$ as

$$\arg\min_{w_n, b_n} = \alpha loss_1 + \beta loss_2, \quad (6)$$

$$loss_1 = \frac{1}{2}\sum_i^a \sum_j^b ||I^{SR}_M(i,j) - I^{LR}_{\times 2}(i,j)||_2^2, \quad (7)$$

$$loss_2 = \frac{1}{2}\sum_i^a \sum_j^b ||I^{SR}_{M+N}(i,j) - I^{HR}(i,j)||_2^2, \quad (8)$$

where a and b denote image width and height respectively, M and N denote the M_{th} and N_{th} convolutional layer, I^{SR}_M is the SR result of the first network, I^{SR}_{M+N} is the final output, (i,j) denotes the image coordinates, and α and β are the weights of the first and second loss function, respectively. It is noted that label $I^{LR}_{\times 2}$ is down-sampled from I^{HR} by a factor of 2 and then upscaled to the size of I^{HR} through bicubic interpolation to perform pixel-wise operation. Different with

previous works [2]–[4], our cascaded deep networks contain two training labels (I^{HR} and $I^{LR}_{\times 2}$) and two loss functions ($loss_1$ and $loss_2$). Due to the splitting operation, $loss_1$ is the structure loss function and $loss_2$ is the texture loss function. During the training process, we back-propagate $loss_1$ from M_{th} layer to the first layer to adjust the weights and biases for recovering structure information in the first deep network. Similarly, we back-propagate $loss_2$ from $(M+N)_{th}$ to $(M+1)_{th}$ layer to learn the weights and biases for recovering texture information in the second deep network. To realizing the isolation operation, we set 'propagate_down' to false in the $(M+1)_{th}$ layer on Caffe [34].

Training is carried out using "Adam" optimizer [35] with a mini-patch of 64 sub-images. We train our model using Caffe[1] [34], a deep learning framework developed by Jia Yangqing et al. and implement this model through MatConvNet[2] package [36]. The weights $w_{1:M+N}$ are initialized using the method described in [37] and the biases $n_{1:M+N}$ are initialized using a constant (zero). The learning rate for weights is set to 10^{-4} and decreased by a factor 10 every 40 epochs and the training is regularized by weight-decay (L_2 penalty multiplied by 0.0001). We empirically train our model by 80 epochs.

C. Network optimization

VDSR [4] is a very deep convolutional network (20 layers) for high-accuracy image SR. Increasing the depth of network enables better ability to model complicated image patterns and leads to performance gain of SR. VDSR deep network consists of 20 layers and each convolutional layer contains 64 filters. The size of filter is set to 3×3 to make the deep network thin as suggested by Simonyan et al. [25]. As the baseline, two standard VDSR deep networks are cascaded to restore major structures ($\times 2 \rightarrow \times 8$) and fine details ($\times 1 \rightarrow \times 2$) individually. Here we use $M(n) + N(n)$ to depict the configuration of the cascaded deep networks. The baseline model can be indicated by $20(64) + 20(64)$, where $M = N = 20$ denotes the number of convolutional layers and $n = 64$ denotes the width[3] of a convolutional layer. Cascading strategy can been effectively applied to boost SR performance [15], [23], [30]. However, such practice will double the number of parameters. Moreover, the computational cost and the chance to fall into local minimum both increase. In this section, we firstly evaluate the effectiveness of the proposed cascaded architecture ($\times 8 \rightarrow \times 2 \rightarrow \times 1$) and then present a number of techniques to optimize the baseline cascaded deep networks ($20(64) + 20(64)$). As a result, the proposed CDN_MRF approach achieves higher SR accuracy using less model parameters.

We compare our proposed cascaded architecture ($\times 8 \rightarrow \times 2 \rightarrow \times 1$) with four other alternatives including (1) without network cascading ($\times 8 \rightarrow \times 1$), (2) three cascaded networks with two mid-points at scale 2 and 4 ($\times 8 \rightarrow \times 4 \rightarrow \times 2 \rightarrow \times 1$), (3) two cascaded networks with a mid-point at scale 4 ($\times 8 \rightarrow \times 4 \rightarrow \times 1$), and (4) two cascaded networks with a mid-point at odd scale 3 ($\times 8 \rightarrow \times 3 \rightarrow \times 1$). In each cascaded network, we make use of a standard VDSR model for fair comparison. SR results of different cascaded architectures are shown in Tab. II. It is noted that employing a cascaded architecture can always achieve SR performance gain as the mapping function from scale $\times 8$ to original scale $\times 1$ is difficult to learn through a single deep network. Although the cascaded architecture of three networks ($\times 8 \rightarrow \times 4 \rightarrow \times 2 \rightarrow \times 1$) contains the largest number of parameters, it does not produce the best SR performance since the complex model becomes difficult to train and over-fitting is likely to happen. It is worth mentioning that scale $\times 2$ provides a better middle point than scale $\times 3$ or $\times 4$ to separate the lost information to structural edges and fine details, and our proposed architecture ($\times 8 \rightarrow \times 2 \rightarrow \times 1$) achieves the highest PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structure SIMilarity) [38].

TABLE II
THE SR RESULTS WITH DIFFERENT CASCADED ARCHITECTURES. THE PSNR AND SSIM VALUES ARE CALCULATED ON OUR TESTING DATASET WHICH CONTAINS 20 INFRARED IMAGES.

Different Architectures	PSNR (dB)	SSIM
$\times 8 \rightarrow \times 1$	35.65	0.9198
$\times 8 \rightarrow \times 4 \rightarrow \times 2 \rightarrow \times 1$	35.87	0.9214
$\times 8 \rightarrow \times 4 \rightarrow \times 1$	35.91	0.9223
$\times 8 \rightarrow \times 3 \rightarrow \times 1$	35.93	0.9220
$\times 8 \rightarrow \times 2 \rightarrow \times 1$	**35.96**	**0.9224**

TABLE III
THE COMPARATIVE RESULTS USING DIFFERENT WIDTH CONFIGURATIONS.

Methods	PSNR(dB)	Number of Parameters
20(64)	35.65	664704
20(64)+20(64)	35.96	1329408
20(32)+20(32)	**35.97**	**332928**
20(16)+20(16)	35.72	83520

A feasible solution to reduce the number of network parameters is to decrease the width parameter n. We set up experiments to investigate how width n influence the SR performance. To compare with VDSR ($n = 64$), we set our network width to two different values: (1) $n = 32$; (2) $n = 16$. The testing dataset contains 20 infrared images captured by a commercial long-wave[4] infrared camera. The comparative SR results are illustrated in Tab. III. Compared with VDSR ($20(64)$), the simplified cascaded deep networks ($20(16)$+$20(16)$) still achieve better SR results (higher average PSNR and SSIM) using only $1/8$ parameters. This is mainly because a single deep network (VDSR) needs more parameters to describe the mapping function working well for the reconstruction of both fine details and structural edges. In contrast, our cascaded architecture uses two models to separately recover high-frequency signals with similar patterns. As the result, lots of parameters can be reduced by using two simpler deep networks. The performance will be further boosted with a wider width ($n = 32$). When further expanding the width ($n = 64$), we do not observe performance improvement while the number of parameters significantly increases, so we set

[1]http://caffe.berkeleyvision.org/

[2]http://www.vlfeat.org/matconvnet/

[3]We use width to term the number of filters in a layer, following [12].

[4]Note that wavelengths of long-wave infrared ranges from 8 to 14um.

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/TCSVT.2018.2864777, IEEE Transactions on Circuits and Systems for Video Technology

TABLE IV
THE COMPARATIVE RESULTS USING DIFFERENT LAYER CONFIGURATIONS.

Methods	PSNR(dB)	Number of Parameters
20(64)	35.65	664704
20(32)+20(32)	35.97	332928
15(32)+20(32)	35.35	286848
10(32)+20(32)	34.85	240768
20(32)+15(32)	35.98	286848
20(32)+10(32)	**36.02**	**240768**
20(32)+5(32)	35.94	194688

Fig. 7. 20 selected training infrared images (from training dataset) covering a wide range of scenarios.

$n = 32$ in our implementation and the architecture of our cascaded deep networks is simplified to $20(32) + 20(32)$.

The number of convolutional layers (M or N) determines how many neighboring pixels (i.e., receptive fields) are considered to recover the lost information. In our cascaded deep networks, a mid-point (scale $\times 2$) is set up between scale $\times 1$ and $\times 8$ to divided lost information into large-scale structures ($\times 2 \rightarrow \times 8$) and small-scale details ($\times 1 \rightarrow \times 2$). Accordingly, a multiple receptive fields strategy is adopted to deal with lost information of different scales. The first network uses a large receptive field to recover structural information from scale $\times 8$ to $\times 2$, while the second one consider information from a relatively smaller receptive field to reconstruct image details from scale $\times 2$ to $\times 1$. A number of different layer configurations are considered and their comparative results are shown in Tab. IV. It is observed that SR results drop when we reduce the number of layers (M) in the first deep network which is used to restore lost information from scale $\times 2$ to scale $\times 8$. The experimental results demonstrate that more neighborhood information considered through a large receptive field is essential to restore large-scale structural information. It is also worth mentioning that decreasing the number of layers in the second deep network ($N = 15$ or $N = 10$), which is used for fine detail restoration, leads to not only reduction

Fig. 8. 20 testing infrared images covering a wide range of scenarios. From left to right, top to bottom: testing 1 to testing 20.

of parameters but also improvement on SR accuracy. Since receptive field is the minimum unit for restoration, irrelevant information within a large receptive field will provide false training samples for the second deep network and decrease SR accuracy. When further decreasing the number of layers ($N = 5$), the performance will drop significantly. In this case, the receptive field is too small to provide sufficient neighborhood information for SR. Based on above analysis, we set $M = 20$, $N = 10$ and the architecture of our proposed CDN_MRF is represented as $20(32) + 10(32)$.

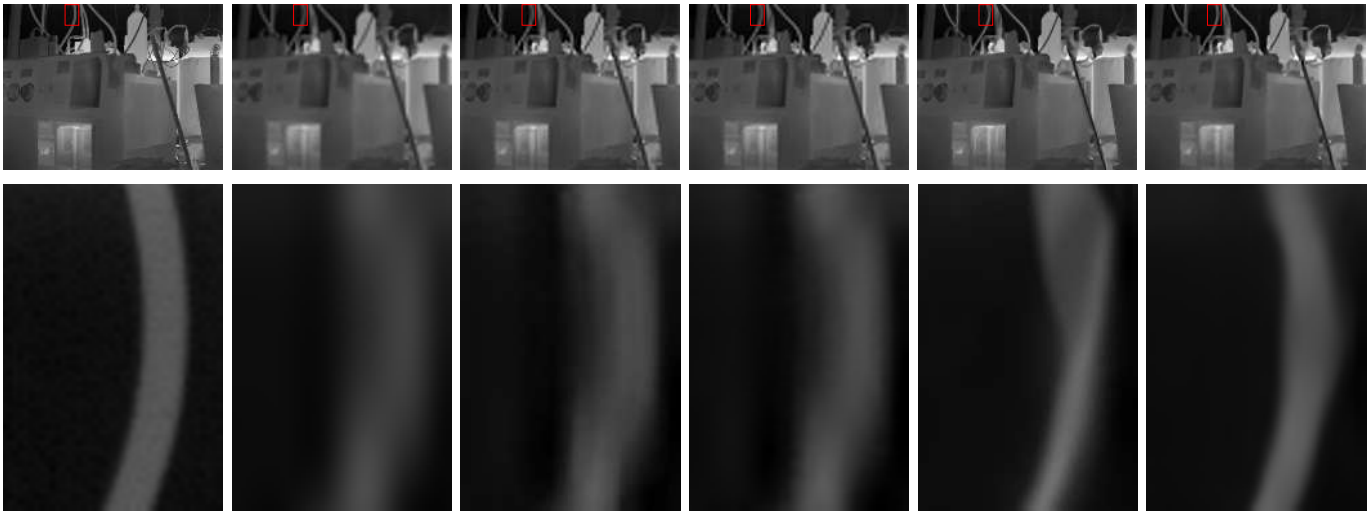
IV. EXPERIMENTS

In this section, we compare our CDN_MRF with several state-of-the-art SR methods using realistic infrared images. The same training and testing dataset are used for fair comparison. Firstly, datasets for training and testing are introduced, and then we outline our implementation details. The comparative results are also illustrated in this section.

A. Datasets

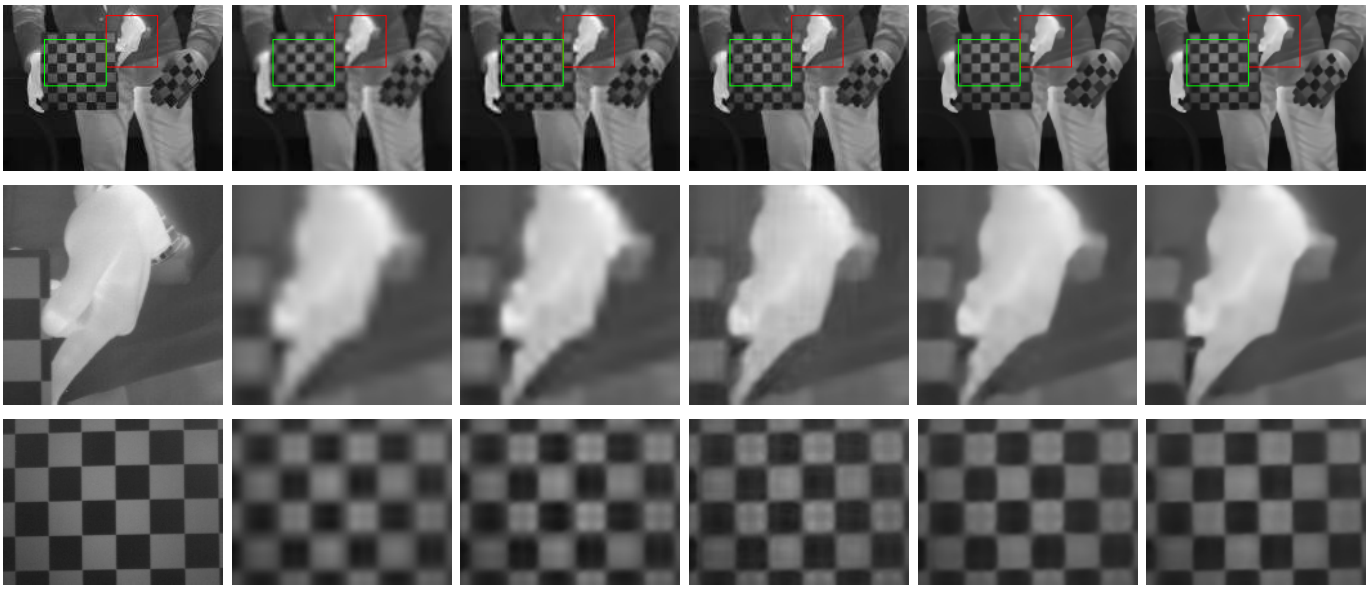
The performances of deep learning methods depend heavily on the training data. For fair comparison, the same training and testing dataset are used for our evaluation. We use a commercial uncooled long ware infrared camera to capture 100 HR infrared images (640×480 resolution) to form the training dataset and another 20 images as the testing dataset. In the training phase, we set the size of training patches to 41×41 and data augmentation (flip and rotation) is used to avoid over-fitting and further improve accuracy. In total 165120 valid (pass the training patch selection described in Sec. III-B) sample sub-images are cropped from the original 640×480 images with a stride of 29. Fig. 7 and Fig. 8 show some sample images from our training and testing datasets. It is observed that our captured images cover a wide range

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/TCSVT.2018.2864777, IEEE Transactions on Circuits and Systems for Video Technology



Ground Truth	Bicubic x8	SCSR-512 [8]	SCSR-1024 [8]	SelfExSR [5]	**CDN_MRF**
PSNR/SSIM	28.99/0.8510	29.67/0.8539	29.74/0.8556	30.15/0.8710	**30.66/0.8864**

Fig. 9. SR results of **Testing 19** using some classic methods (SCSR-512, SCSR-1024, SelfExSR) and our method. The first row exhibits the ground truth and some processing results. The second row are the zoom-in views of the highlighted regions. The SCSR methods generate blurry SR results and the SelfExSR method causes undesired artifacts that distort the original shape of electrical cable.



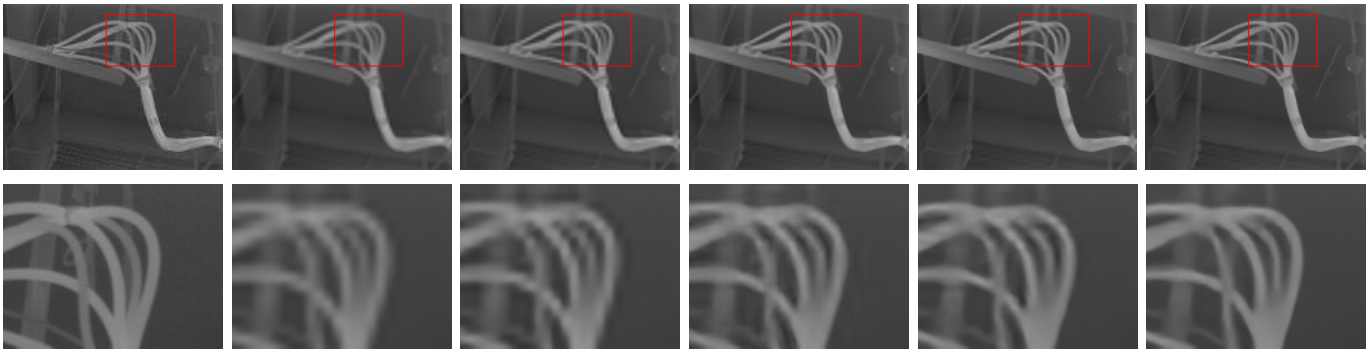
Ground Truth	Bicubic x8	TEN [2]	SRCNN [3]	VDSR [4]	CDN_MRF
PSNR/SSIM	30.44/0.8751	31.37/0.8866	33.32/0.9086	33.92/0.9217	**34.61/0.9287**

Fig. 10. Original image and comparative SR processing results of **Testing 10**. The first row shows processing results using different methods. The second and third rows visualize the highlighted regions in the first row. It is observed that the edges in the red highlighted region restored by our SR method appear much sharper. As well, our method suppresses the artifacts as shown in the green highlighted region. PSNR value of our method is significantly higher than the second best performing SR method VDSR ($> 0.69dB$).

of contents (e.g., vehicle, machine, pedestrian and building) and the training and testing datasets are significantly different from each other. These infrared images will be made publicly available in the future.

B. Implementation details

In our implementation, we use a cascaded network of a total depth of 30. Training is carried out by optimizing the objective function using "Adam" optimizer with a mini-patch of 64 sub-images. Weight decay and threshold θ are set to 0.0001 and 0.0005 respectively. We utilize the method described in [37] for weights initialization and the biases are initialized to zeros. The model is trained for 80 epochs. The learning rate for weights is set to 10^{-4} and decreased by a factor of 10 every 40 epochs. For each layer, we set (k, n, p) to $(3, 32, 1)$ except the 20th and 30th layers where we set (k, n, p) to $(3, 1, 1)$ to reconstruct the output image. The weights of the two loss functions are set to the same value: $\alpha = \beta = 1$. We train our models on a single GPU of NVIDIA TITAN X.



Ground Truth	Bicubic x8	TEN [2]	SRCNN [3]	VDSR [4]	CDN_MRF
PSNR/SSIM	34.45/0.8951	35.30/0.8998	36.67/0.9107	36.81/0.9128	**37.25/0.9171**

Fig. 11. Some comparative SR results of **Testing 5**. Cable contours are well restored by our method while they are either blurred or distorted in SR results of other methods.

Ground Truth	Bicubic x8	TEN [2]	SRCNN [3]	VDSR [4]	**CDN_MRF**
PSNR/SSIM	35.10/0.8991	35.63/0.9019	36.65/0.9087	36.92/0.9108	**37.33/0.9139**

Fig. 12. SR results of **Testing 19** using TEN, SRCNN, VDSR and our method. Please zoom in to check details highlighted in red rectangles.

C. Comparisons with state-of-the-art SR methods

We perform quantitative and qualitative experiments to compare our proposed method with state-of-the-art SR approaches including classic methods (e.g., SCSR [8] and SelfExSR [5]) and deep-learning-based methods (e.g., SRCNN [3], VDSR [4] and TEN [2]). The source codes of SCSR[5], SelfExSR[6] and SRCNN[7] methods are provided by their authors. For SCSR method, two dictionaries of size 512 and 1024 are trained respectively. The implementation of VDSR model is also publicly available[8]. We re-implement TEN method which consists of the 4-layer CNNs on Caffe and apply the same parameter setting described in the paper [2]. This re-implementation achieves very similar SR results reported in the original paper. All of these methods are trained using the same dataset described in Sec. IV-A to ensure fair comparison.

First of all, we show comparative results of a number of classic methods including SCSR [8] and SelfExSR [5] in Fig. 9. SCSR-512 and SCSR-1024 denote SCSR model with dictionary size of 512 and 1024, respectively. It is observed that both SCSR-512 and SCSR-1024 methods output blurry SR results. SelfExSR method causes undesired artifacts that distort the original shape of objects. Another limitation of SelfExSR is that it will fail to recover the fine details when the input image does not contain obvious planes and similar texture patterns [5]. A noticeable disadvantage of these classic methods is they are time-consuming and not suitable for real-time applications. Time comparisons are provided in Sec. IV-D.

Then, comparative results of some deep-learning-based methods are illustrated in Fig. 10. Overall our SR method based on cascaded deep networks can achieve better image restoration results. It is observed that contour edges in red highlighted region processed by our SR method are much sharper and clearer than results of other methods. As well, our method can effectively suppress undesired artifacts as shown in the green highlighted region. With such improvement, small objects (e.g., finger tip) can be easily identified in our SR result. More SR results can be found in Fig. 11 and Fig. 12.

We quantitatively evaluate SR performances of our CDN_MRF method and state-of-the-art methods (SCSR [8], SelfExSR [5], TEN [2], SRCNN [12], VDSR [4]). We make use of PSNR and SSIM as our evaluation metrics and the comparative results are shown in Tab. V. When calculating the metrics, we ignore certain amount of borderpixels according to the work presented in [39]. On average, our method outperforms other state-of-the-art SR methods by large margins (VDSR: $> 0.37dB$, SRCNN: $> 0.7dB$, SelfExSR: $> 0.85dB$, SCSR-1024: $> 1.78dB$, SCSR-512: $> 1.84dB$,

[5]http://www.ifp.illinois.edu/ jyang29/ScSR.htm
[6]SelfExSR: https://sites.google.com/site/jbhuang0604/publications/struct_sr
[7]SRCNN: http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html
[8]https://github.com/huangzehao/caffe-vdsr

TEN: $> 1.87dB$). Moreover, the performance of our method is very stable and it achieves the best SR results and produces the highest PSNR values for all 20 testing images.

Another advantage of our proposed CDN_MRF architecture is that it can achieve better SR accuracy with significantly less parameters. As illustrated in Tab. VI and Fig. 13, using significantly less parameters $(1/10)$, our $20(16) + 10(16)$ model still achieves more accurate SR result compared with the VDSR method $(20(16) + 10(16)$: 35.71 dB vs. VDSR: 35.65 dB). If we further decrease the parameters (width = 12), our lightweight $20(12)+10(12)$ model with 34128 parameters achieves an averaged PSNR value of $35.60dB$ which is better than the performances of SRCNN (57184 parameters and 35.32 dB PSNR) and TEN (63840 parameters and 34.15 dB PSNR).

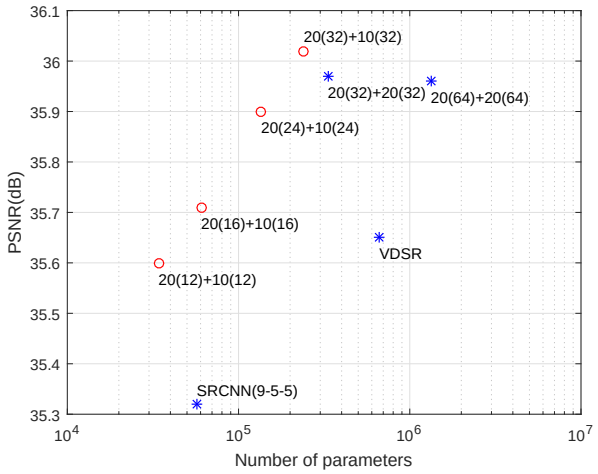


Fig. 13. Graph of PSNR vs. Number of Parameters. Our method $20(32) + 10(32)$ achieves the highest PSNR value. TEN is not included in this comparison since it cannot produce comparable PSNR values.

D. Time comparison

The training and testing times of our CDN_MRF method and SCSR [8], SelfExSR [5], TEN [2], SRCNN [12], VDSR [4] methods are systematically evaluated. The SelfExSR method does not require a training process based on external datasets, so its training time is negligible. The training process of SCSR takes $\sim$4 hours to converge on a CPU. The rest deep-learning-based methods are trained on a single GPU of NVIDIA TITAN X. The lightweight SRCNN and TEN methods directly learn the mapping relationship between LR/HR pairs and their training process takes a long time (several days) to converge. VDSR embeds residual learning and gradient clipping strategies to significantly reduce the training process to $\sim$5 hours. In comparison, the training times of our CDN_MRF models $(20(32)+10(32)$ and $20(16)+10(16))$ are $\sim$3.5 hours and $\sim$2 hours, respectively.

Although the training process of deep-learning-based methods is time-consuming, the trained model can be efficiently deployed during the testing phase which is critical for practical applications. In addition, deep-learning-based methods do not need to fine tune the hyper-parameters to achieve good performance. For fair comparison, above mentioned SR methods are conducted to process a 640×480 resolution image in Matlab R2015b without GPU or parallel implementation on a PC with an Inter Core i7-6820HK CPU (2.7GHz) and 16 GB memory. Each SR method is executed for 100 times and the averaged testing time is provided in Fig. 14.

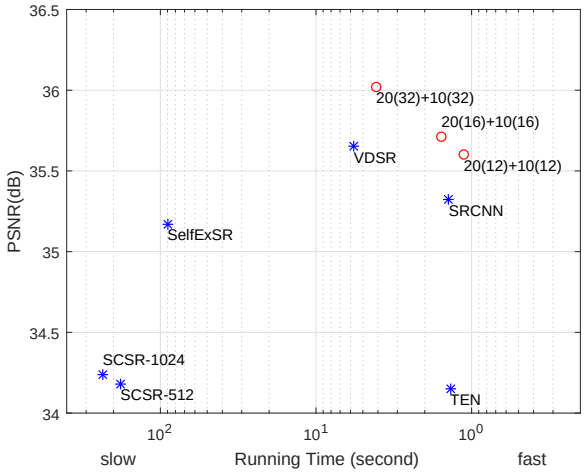


Fig. 14. Graph of PSNR vs. running time. All SR methods are conducted in Matlab R2015b without GPU or parallel implementation on a PC with an Inter Core i7-6820HK CPU (2.7GHz) and 16 GB memory.

It is noted that classic SCSR and SelfExSR methods require $\sim$100 seconds to restore HR outputs since the optimization of sparse representation and internal patch searching are extremely time-consuming. In comparison, our $20(32) + 10(32)$ produce a significantly higher PSNR value using less testing time compared with VDSR method. It is worth mentioning that our $20(12) + 10(12)$ model achieves faster speed than the lightweight SRCNN (3 layers) and TEN (4 layers) methods, and its SR results are comparable with ones of VDSR.

V. Conclusion

Infrared images have a wide range of military and civilian applications including night vision, surveillance and robotics. However, due to hardware limitation, existing thermal cameras can only produce LR infrared images. In our proposed CDN_MRF, residual information could be divided into two components: major structures and fine details. Our method contains two consecutive networks to gradually recover the high-frequency information. The first network restores most of the structure information and the second one tries to recover image fine details. Our experiments demonstrate that the proposed cascaded architecture of deep networks, with a significantly smaller number of parameters $(1/10)$, can still achieves better performance compared with state-of-the-art deep-learning-based SR methods (VDSR).

In the future, we plan to further optimize the number of parameters for real-time implementation without compromising SR accuracy. Another feasible solution to reduce computational cost is take the down-sampled LR image without bi-cubic interpolation as input. Also, applicability of this

TABLE V
THE PSNR (dB) AND SSIM VALUES OF OUR CDN_MRF METHOD AND STATE-OF-THE-ART SR METHODS. IT IS OBSERVED THAT OUR CDN_MRF METHOD ACHIEVES THE BEST SR PERFORMANCES FOR ALL 20 TESTING IMAGES. NOTE **BLOD FONT** AND UNDERLINE INDICATE THE BEST AND THE SECOND BEST SR RESULTS, RESPECTIVELY.

PSNR(dB)/SSIM	Bi-cubic $\times 8$	SCSR-1024 [8]	SelfExSR [5]	TEN [2]	SRCNN [3]	VDSR [4]	CDN_MRF
Testing 1	31.36 / 0.8413	31.85 / 0.8426	32.09 / 0.8435	31.82 / 0.8442	32.24 / 0.8478	32.31 / 0.8499	**32.58 / 0.8524**
Testing 2	32.95 / 0.8947	34.09 / 0.8980	35.71 / 0.9101	34.04 / 0.9016	35.95 / 0.9119	36.44 / 0.9173	**36.82 / 0.9205**
Testing 3	31.55 / 0.9122	32.90 / 0.9150	34.28 / 0.9231	32.58 / 0.9199	35.08 / 0.9265	35.65 / 0.9326	**36.45 / 0.9366**
Testing 4	30.71 / 0.8890	31.75 / 0.8919	33.87 / 0.9128	31.74 / 0.8980	33.86 / 0.9151	34.44 / 0.9225	**35.21 / 0.9268**
Testing 5	34.45 / 0.8951	35.43 / 0.8986	36.46 / 0.9037	35.30 / 0.8998	36.67 / 0.9107	36.81 / 0.9128	**37.25 / 0.9171**
Testing 6	36.87 / 0.9276	37.32 / 0.9277	37.30 / 0.9248	37.30 / 0.9290	37.90 / 0.9318	38.08 / 0.9329	**38.22 / 0.9336**
Testing 7	28.99 / 0.8510	29.74 / 0.8556	30.15 / 0.8710	29.69 / 0.8614	30.37 / 0.8749	30.47 / 0.8800	**30.66 / 0.8864**
Testing 8	30.86 / 0.8797	31.78 / 0.8827	32.80 / 0.8929	31.80 / 0.8886	33.19 / 0.8993	33.52 / 0.9056	**33.88 / 0.9111**
Testing 9	30.85 / 0.8783	31.81 / 0.8798	33.13 / 0.8936	31.78 / 0.8855	33.41 / 0.8946	33.75 / 0.9016	**34.30 / 0.9062**
Testing 10	30.44 / 0.8751	31.45 / 0.8789	34.56 / 0.9273	31.37 / 0.8866	33.32 / 0.9086	33.92 / 0.9217	**34.61 / 0.9287**
Testing 11	37.86 / 0.9250	38.45 / 0.8253	38.39 / 0.9247	38.46 / 0.9267	39.19 / 0.9290	39.38 / 0.9303	**39.60 / 0.9314**
Testing 12	35.12 / 0.9685	35.88 / 0.9690	37.50 / 0.9763	35.77 / 0.9700	36.83 / 0.9752	37.27 / 0.9771	**37.55 / 0.9788**
Testing 13	34.20 / 0.8796	34.68 / 0.8822	34.33 / 0.8721	34.52 / 0.8819	34.88 / 0.8859	35.04 / 0.8878	**35.27 / 0.8903**
Testing 14	34.30 / 0.9380	34.89 / 0.9381	35.14 / 0.9427	34.79 / 0.9400	35.44 / 0.9438	35.74 / 0.9469	**36.10 / 0.9495**
Testing 15	33.07 / 0.8947	33.82 / 0.8974	34.00 / 0.9005	33.74 / 0.8989	34.58 / 0.9049	34.73 / 0.9081	**34.91 / 0.9106**
Testing 16	35.14 / 0.9027	35.61 / 0.9028	35.77 / 0.9010	35.59 / 0.9040	36.29 / 0.9067	36.50 / 0.9087	**36.66 / 0.9094**
Testing 17	31.90 / 0.9055	32.55 / 0.9063	33.45 / 0.9104	32.46 / 0.9088	33.18 / 0.9125	33.47 / 0.9156	**33.76 / 0.9195**
Testing 18	36.86 / 0.9721	38.28 / 0.9719	40.16 / 0.9750	38.02 / 0.9743	39.46 / 0.9747	40.36 / 0.9792	**40.80 / 0.9807**
Testing 19	35.10 / 0.8991	35.71 / 0.9012	36.47 / 0.9029	35.63 / 0.9019	36.65 / 0.9087	36.92 / 0.9108	**37.33 / 0.9139**
Testing 20	36.10 / 0.9476	36.77 / 0.9478	37.80 / 0.9450	36.70 / 0.9488	37.94 / 0.9520	38.24 / 0.9541	**38.37 / 0.9559**
Average	33.43 / 0.9038	34.24 / 0.9056	35.17 / 0.9128	34.15 / 0.9085	35.32 / 0.9157	35.65 / 0.9198	**36.02 / 0.9230**

TABLE VI
THE COMPARATIVE SR RESULTS OF BI-CUBIC INTERPOLATION, VDSR, AND OUR PROPOSED CASCADED DEEP NETWORKS USING DIFFERENT WIDTHS. IN OUR CDN_MRF METHOD, $M = 20$ AND $N = 10$. OUR CDN_MRF CAN BE INDICATED AS $20(n) + 10(n)$.

Methods	Bi-cubic	VDSR [4]	SRCNN [3]	TEN [2]	$20(32) + 10(32)$	$20(24) + 10(24)$	$20(16) + 10(16)$	$20(12) + 10(12)$
# Parameters	–	664704	57184	63840	240768	135648	60480	34128
PSNR (dB)	33.43	35.65	35.32	34.15	36.02	35.90	35.71	35.60
Improvement	–	2.22	1.89	0.72	2.59	2.47	2.28	2.17
SSIM	0.9038	0.9198	0.9157	0.9085	0.9230	0.9221	0.9207	0.9196
Improvement	–	0.0160	0.0119	0.0047	0.0192	0.0183	0.0169	0.0158
Performance	8	4	6	7	1	2	3	5

cascaded architecture for other spectral images will be investigated in the future. Moreover, we plan to implement the proposed method in our hardware device to improve the quality of infrared images for other high-level computer vision applications such as video stabilization, stereo matching, image stitching, target detection and tracking.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China (No. 51605428, 51575486 and U1664264), and in part by the Fundamental Research Funds for the Central Universities. We thank Dr. Christel-Löic Tisse (ULIS, ZI Les Iles Cordees BP27, 38113 Veurey-Voroize, France) for his insightful discussions. Also, we would like to thank the anonymous reviewers for their valuable suggestions.

REFERENCES

[1] A. Rogalski, P. Martyniuk, and M. Kopytko, "Challenges of small-pixel infrared detectors: a review," *Reports on Progress in Physics*, vol. 79, no. 4, p. 046501, 2016.

[2] Y. Choi, N. Kim, S. Hwang, and I. So, "Thermal Image Enhancement using Convolutional Neural Network," in *Int. Conf. Intell. Robot. Syst.*, 2016.

[3] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *IEEE Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.

[4] J. Kim, J. K. Lee, and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[5] J. B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.

[6] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 29, no. 6, pp. 1153–1160, 1981.

[7] R. Timofte, V. De, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1920–1927.

[8] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image Super-Resolution Via Sparse Representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, 2010.

[9] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *IEEE Int. Conf. Comput. Vis.*, 2009, pp. 349–356.

[10] K. Zeng, J. Yu, R. Wang, C. Li, and D. Tao, "Coupled Deep Autoencoder for Single Image Super-Resolution," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 27–37, 2017.

[11] L. Ding, Z. Wang, B. Wen, and J. Yang, "Robust single image super-resolution via deep networks with sparse prior," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3194–3207, 2016.

[12] 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, 2015.

[13] C. Dong, C. C. Loy, and X. Tang, "Accelerating the Super-Resolution Convolutional Neural Network," in *IEEE Eur. Conf. Comput. Vis.*, 2016.

[14] Y. Wang, L. Wang, H. Wang, and P. Li, "End-to-End Image Super-Resolution via Deep and Shallow Convolutional Networks," in *arXiv preprint arXiv:1607.07680*, 2016.

[15] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *IEEE Int. Conf. Comput. Vis.*, 2015, pp. 370–378.

[16] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken1, R. Bishop, D. Rueckert, and Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883.

[17] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," in *IEEE Eur. Conf. Comput. Vis.*, 2016.

[18] H. Chang, D. Y. Yeung, and Y. Xiong, "Super-Resolution through Neighbor Embedding," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 275–282.

[19] M. Bevilacqua, A. Roumy, C. Guillemot, and A. Morel, "Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding," in *Br. Mach. Vis. Conf.*, 2012.

[20] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008.

[21] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution." *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3467–3478, 2012.

[22] R. Timofte, V. D. Smet, and L. V. Gool, "A+: Adjusted Anchored Neighborhood Regression for Fast Super-Resolution," in *Asian Conf. Comput. Vis.*, 2014.

[23] R. Timofte, R. Rothe, and L. V. Gool, "Seven ways to improve example-based single image super resolution," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[24] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–10, 2010.

[25] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Int. Conf. Learn. Represent.*, 2015, pp. 1–14. [Online]. Available: http://arxiv.org/abs/1409.1556

[26] W. Ouyang and X. Wang, "Joint Deep Learning for Pedestrian Detection," in *IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2056–2063.

[27] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," in *IEEE Int. Conf. Comput. Vis.*, 2013, pp. 633–640.

[28] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International Conference on Curves and Surfaces*, 2010, pp. 711–730.

[29] R. Timofte, E. Agustsson, L. V. Gool, M. H. Yang, L. Zhang, B. Lim, S. Son, H. Kim, S. Nah, K. M. Lee, X. Wang, Y. Tian, K. Yu, Y. Zhang, S. Wu, C. Dong, L. Lin, Y. Qiao, C. C. Loy, W. Bae, J. Yoo, Y. Han, J. C. Ye, J. S. Choi, M. Kim, Y. Fan, J. Yu, W. Han, D. Liu, H. Yu, Z. Wang, H. Shi, X. Wang, T. S. Huang, Y. Chen, K. Zhang, W. Zuo, Z. Tang, L. Luo, S. Li, M. Fu, L. Cao, W. Heng, G. Bui, T. Le, Y. Duan, D. Tao, R. Wang, X. Lin, J. Pang, J. Xu, Y. Zhao, X. Xu, J. Pan, D. Sun, Y. Zhang, X. Song, Y. Dai, X. Qin, X. P. Huynh, T. Guo, H. S. Mousavi, T. H. Vu, V. Monga, C. Cruz, K. Egiazarian, V. Katkovnik, R. Mehta, A. K. Jain, A. Agarwalla, C. V. Praveen, R. Zhou, H. Wen, C. Zhu, Z. Xia, Z. Wang, and Q. Guo, "NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results," in *IEEE CVPR Workshop*, vol. 2017-July, 2017, pp. 1110–1121.

[30] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep Network Cascade for Image Super-resolution," in *IEEE Eur. Conf. Comput. Vis.*, 2014, pp. 1–16.

[31] H. Pfister, W. Matusik, N. J. W. Morris, and S. Avidan, "Statistics of Infrared Images," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–7.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016. [Online]. Available: http://arxiv.org/pdf/1512.03385v1.pdf

[33] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[34] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[35] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," in *Int. Conf. Learn. Represent.*, 2015.

[36] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *IEEE Int. Conf. Comput. Vis.*, 2015.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[39] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," in *CVPR workshop*, 2017.

Zewei He now is a Ph.D. candidate in the School of Mechanical Engineering, Zhejiang University, Hangzhou, China. He graduated with B.E in Mechanical Engineering and Automation from University of Science and Technology Beijing (USTB) (2014). His research interests include infrared imaging, multi-sensor image fusion, super-resolution and image de-noising.

Siliang Tang is currently an associated professor with the College of Computer Science, Zhejiang University. He received the BSc from Zhejiang University, Hangzhou, China and the Ph.D. from the National University of Ireland, Maynooth, Ireland. His research interests include multimedia analysis, text mining and statistic learning.

Jiangxin Yang now is a full-time professor in the State Key Laboratory of Fluid Power and Mechatronic Systems and Laboratory of Advanced Manufacturing Technology of Zhejiang Province, School of Mechanical Engineering, Zhejiang University, Hangzhou, China. His research interests are quality engineering, infrared imaging and measurement.

Yanlong Cao now is a full-time professor in the State Key Laboratory of Fluid Power and Mechatronic Systems and Laboratory of Advanced Manufacturing Technology of Zhejiang Province, School of Mechanical Engineering, Zhejiang University, Hangzhou, China. His research interests are precision design, quality engineering and measurement.

Michael Ying Yang is currently an Assistant Professor with University of Twente (the Netherlands), heading a group working on scene understanding. He received the Ph.D. degree (summa cum laude) from University of Bonn (Germany) in 2011. From 2008 to 2012, he worked as Researcher with the Department of Photogrammetry, University of Bonn. From 2012 to 2015, he was a Postdoctoral Researcher with the Institute for Information Processing, Leibniz University Hannover. From 2015 to 2016, he was a Senior Researcher with Computer Vision Lab Dresden, TU Dresden. His research interests are in the fields of computer vision and photogrammetry with specialization on scene understanding and semantic interpretation from imagery and videos.

Yanpeng Cao is a Research Fellow in the School of Mechanical Engineering, Zhejiang University, Hangzhou, China. He graduated with MSc in Control Engineering (2005) and Ph.D. in Computer Vision (2008), both from the University of Manchester, UK. He worked in a number of R&D institutes such as Institute for Infocomm Research (Singapore), Mtech Imaging Ptd Ltd (Singapore), and National University of Ireland Maynooth (Ireland). His major research interests include infrared imaging, sensor fusion, image processing, and 3D reconstruction.