*

A Hardware-Aware Network for Real-World Single Image Super-Resolutions

Rui Ma, Xian Du*

***Abstract*—Most single image super resolution (SISR) methods are developed on synthetic low resolution (LR) and high resolution (HR) image pairs, which are simulated by a predetermined degradation operation, such as bicubic downsampling. However, these methods only learn the inverse process of the predetermined operation, which fails to super resolve the real-world LR images, whose true formulation deviates from the predetermined operation. To address this, we propose a novel SR framework named hardware-aware super-resolution (HASR) network that first extracts hardware information, particularly the camera degradation information. The LR images are then super resolved by integrating the extracted information. To evaluate the performance of HASR network, we build a dataset named Real-Micron from real-world micron-scale patterns. The paired LR and HR images are captured by changing the objectives and registered using a developed registration algorithm. Transfer learning is implemented during the training of Real-Micron dataset due to the lack of amount of data. Experiments demonstrate that by integrating the degradation information, our proposed network achieves state-of-the-art performance for the blind SR task on both synthetic and real-world datasets.**

Impact Statement— The proposed HASR method has significant impact on various areas, such as enhancing the accurate inspection of manufactured products for quality control and enhancing the resolution of medical images to enable more accurate diagnosis and healthcare. Current SR solutions neglect the uniqueness of each imaging system, hence cannot produce accurate HR images across the different systems. Taking advantage of the known hardware information, HASR can differentiate low-resolution images across different imaging systems and produce HR images that are closer to the real-world scenario. Given sufficient training images, the proposed HASR method can overcome the physical optical limitation and generate higher quality images. The proposed method improves the overall performance by about 0.2 dB and 0.5 dB on the synthetic and the real-world datasets, respectively.

***Index Terms*—Single image super-resolution (SISR), blind super-resolution, transfer learning**

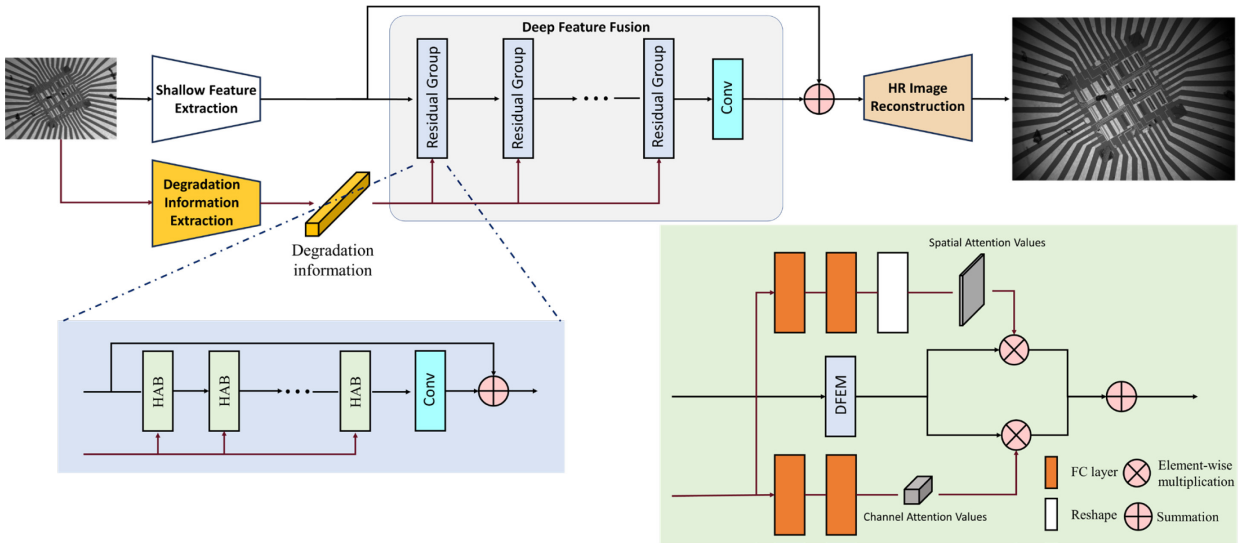
I. INTRODUCTION

High-resolution digital images are consistently preferred, whether for human satisfaction or for various downstream industrial applications. However, there are instances where obtaining images with the desired resolution is challenging due to limitations in imaging hardware. Factors like low-resolution (LR) cameras or unstable imaging conditions can result in a loss of image resolution. To address this issue, image super-resolution (SR) techniques are frequently employed. These SR techniques are designed to reconstruct high-resolution (HR) images from their LR counterparts. Image SR not only has the potential to enhance image details and realism [1] but also to overcome the limitations of imaging systems [2]. Recently, deep learning has paved the way for the development of numerous advanced SR algorithms that leverage large-scale datasets [3]–[5]. While these methods excel with artificially degraded LR images, like those created through techniques such as bicubic downsampling, they face challenges when dealing with real-world LR images. This decline in performance results from a domain gap between the training data and the data encountered during inference, particularly when the degradation kernel of real-world LR images differs from the one used for training.

There are typically two approaches to address the SR issue mentioned: (1) generating LR images through multiple degradation models during training [6]–[8], and (2) learning the degradation kernel first and then using it for SR [9]–[11]. The first approach struggles with complex real-world degradations, while the second approach is more practical, but it often overlooks a critical piece of prior knowledge: the hardware information of image acquisition devices.

Real-world degradations, stemming from factors like camera blur, sensor noise, sharpening artifacts, and image compression [6], are closely tied to the specific imaging system (camera) in use. Therefore, we posit that possessing prior knowledge of image acquisition system can significantly enhance real-world

- Correspoinding author: Xian Du
- Rui Ma and Xian du are with Center for Personalized Health Monitoring, Institute for Applied Life Sciences, University of Massachusetts, Amherst, MA 01003, USA
- Rui Ma is with Electrical and Computer Engineering Department, University of Massachusetts, Amherst, MA 01003, USA
- Xian Du is with Mechanical and Industrial Engineering Department, University of Massachusetts, Amherst, MA 01003, USA



(a) Residual Group. (b) Hardware-Aware Block (HAB).

Fig. 1. The architecture of HASR network.

SR, a common scenario in industry where known camera models and lenses are typically used to for image acquisition. Leveraging this prior knowledge and the supervised contrastive learning (SupCon) method [12], we can generate hardware representations and employ them to enhance the generation of SR images.

Our proposed hardware aware super-resolution (HASR) network consists of two steps. In the first step, we aim to extract hardware representations. We hypothesize that, in relatively stable capture environments, images taken by the same camera share similar blur kernels, while those from different cameras exhibit distinct blur kernels. Initially, we considered querying the specifications like pixel resolution and sensor type and encoding this information into vectors. However, for efficient differentiation of images from different hardware setups, we adopted contrastive learning. This method groups image patches from the same camera and separates patches from different cameras, implicitly embedding the camera's hardware information. In the second step, we integrate this hardware information into the SR network using our proposed hardware-aware block (HAB), incorporating spatial and channel attention mechanisms. Detailed structures of the HASR provided in Fig. 1 and Section III.

Furthermore, obtaining real-world LR-HR image pairs is challenging, resulting in limited large-scale real-world SR datasets. We address this in two ways. First, we apply transfer learning to the HASR network by initially training the network on publicly available synthetic datasets and fine-tune it with a small number of real-world datasets. These synthetic datasets simulate degradation processes using isotropic Gaussian filters with additive Gaussian noise. Second, we introduce the Real-Micron dataset, containing micron-scale patterns and captured using three Basler CMOS cameras with objectives of various high magnification factors (see details in Section IV).

The contributions of are as follows:

- Pioneering the utilization of hardware information to enhance SR generation.
- Introducing a novel supervised contrastive learning method for learning unknown degradation processes in various image acquisition systems.
- Empirically demonstrating that integrating prior hardware information significantly enhances SR generation.
- Presenting a real-world dataset featuring micron-scale patterns and containing precisely aligned HR and LR image pairs with different scale factors.

II. Related Work

This section is divided into three parts: The first part surveys current solutions for the blind SR problem, the second part introduces contrastive learning and its variants, and the third part explores feature fusion methods.

A. Blind SR Methods

As discussed in the first section, there are two categories of blind SR methods. The first category includes methods that incorporate multiple degradation models in the network. For example, in [8], the authors proposed to concatenate an LR input image with its degradation map as a unified input to the SR model, allowing for feature adaptation according to the specific degradation and covering multiple degradation types in a single model. In [7], a kernel modeling super-resolution network (KMSR) was proposed, where the simulated LR images were generated by applying a specific blur kernel to HR images, which was chosen from a predetermined kernel pool. Other methods, such as [6], [13], [14], built more generic training datasets with more kinds of realistic blur kernels. However, these methods had a significant drawback: they relied on predefined blur kernel pools and could not provide satisfactory results for images with degradations not covered in their pools.

The second category is to estimate the degradation kernel first and then to super resolve the LR images with the learned degradation kernel information. For instance, Iterative kernel correction (IKC) [10] proposed to correct kernel estimation in an iterative way to gradually approach a satisfactory result. In [9], the authors introduced "KernelGAN", an image-specific Internal-GAN that estimated the SR kernel (downscaling kernel) that best preserved the distribution of patches across scales of the LR image. However, these methods were time-consuming due to the numerous iterations during inference. In [15], unsupervised contrastive learning was used to estimate the degradation process. The authors first learned abstract representations to distinguish the various degradations in the representation space rather than explicitly estimating the exact degradations. They then introduced a Degradation-Aware SR (DASR) network with flexible adaptation to various degradations based on the learned representations. A contrastive loss was used to conduct unsupervised degradation representation learning by contrasting positive pairs against negative pairs in the latent space. However, the degradation representation highly relied on the contents of the LR images because of the assumption that each image had a unique degradation kernel. In [16], an unsupervised way to imitate real-world LR images of an unknown downsampling process was proposed. The authors implemented generative adversarial network [17] to generate the LR images that had similar distribution to the real-world LR images. Furthermore, to keep the generation process stable, low-frequency loss (LFL) and adaptive data loss (ADL) were utilized to keep the content consistency between the generated LR and the real-world LR images. However, balancing the data loss and the adversarial loss needed to be very careful. They also did not consider the kernel variances from the training data. The estimated degradation kernel was just an average from all the training data, which would be inaccurate if the training data came from different acquisition systems.

B. Contrastive Learning

Contrastive learning is a self-supervised learning method widely utilized in computer vision, natural language processing, and other domains. Intuitively, contrastive learning can be considered as learning by comparing. To learn the representations of the samples, contrastive learning compares the similarities among the samples: it aims to embed similar samples (positive examples) close to each other while trying to push different samples (negative examples) away. In [18], a simple framework for contrastive learning of visual representations (SimCLR) was presented. SimCLR learned representations by maximizing agreement between differently augmented views of the same data example via a contrastive loss in the latent space. The paper showed that the authors' methods significantly outperformed previous techniques for self-supervised and semi-supervised learning on ImageNet. However, the batch size for SimCLR training was limited by the hardware constraints such as GPU memory. To address this issue, MoCo [19] introduced a dynamic dictionary with a queue and a moving-averaged encoder, allowing for the creation of a large and consistent dictionary on-the-fly, which facilitated contrastive unsupervised learning. MoCo-V2 [20] built upon this approach by incorporating SimCLR's stronger data augmentation and MLP projection head, enabling it to achieve better results than SimCLR on a typical 8-GPU machine. Additionally, if additional labels were provided, they could be integrated into the contrastive framework's similarity and dissimilarity definitions. The authors of [12] extended the self-supervised batch contrastive approach to the fully-supervised setting with two possible versions of the supervised contrastive (SupCon) loss. The SupCon loss offered benefits for robustness to natural corruptions and was more stable to hyperparameter settings such as optimizers and data augmentations.

C. Feature fusion

As deep learning continues to evolve in handling multimodal data, the effective fusion of information across multiple modalities is extensively explored. Multimodal information fusion is typically categorized into three main approaches: early (feature-based), late (decision-based), and hybrid fusion [21]. In the context of this paper, we exclusively focus on early fusion, where hardware information is treated as a supplementary component rather than an independent modality. Within early fusion, one straightforward technique involves the use of adaptive instance normalization (AdaIN) [22] to align the mean and variance of features from one modality with those from another. Attention mechanisms, widely employed in image super-resolution (SR) networks, have played a pivotal role in early fusion. In [23], a channel attention mechanism was proposed to adaptively rescale channel-wise features by considering interdependencies among channels. Additionally, in [24], the authors introduced the holistic attention network (HAN) to model the comprehensive interdependencies among layers, channels, and positions. In [25], an SR network based on graph attention network (SRGAT) fully leveraged internal patch-recurrence within natural images. With the increasing adoption of transformer backbones, self-attention mechanisms are making their way into SR tasks as well. In [26], a multiscale hierarchical design, incorporating efficient Transformer blocks, was introduced to capture long-range pixel interactions, even for large images. This approach divides images into multiple patches that interact with each other through self-attention mechanisms within the transformer blocks. This paper focuses on investigating whether the fusion of hardware information improves SR performance. Thus, our exploration has been primarily centered on the application of attention mechanisms. We remain open to considering additional fusion methods in the future, with the anticipation that more effective solutions will be uncovered.

III. METHOD

This section begins by elucidating the rationale behind the use of hardware information. It then proceeds to offer a comprehensive overview of the HASR network, as illustrated in Fig. 1.

A. Motivation of using hardware information

Digital image acquisition systems play a pivotal role in

myriad of applications, capturing continuous real-world objects and generating sampled image, denoted by f_{LR}. In these systems, a physical camera can be conceptually modeled as a continuous-space filter, followed by sampling on a lattice [27]. If a higher-resolution camera capable of producing the desired HR image f_{HR} exists, the transformation between the HR image and the LR images can be defined as a function, represented as:

$$f_{LR} = D(f_{HR}), \tag{1}$$

where $D(\cdot)$ is a degradation function that amalgamates both filtering and down-sampling processes. The essence of SR problem is to derive an estimated HR image $\hat{f}_{HR}$ from f_{LR}, effectively inverting transformation in (1). Note the SR problem is inherently ill-posed because multiple different HR images can yield the same LR result. To address this, it is transformed into an optimization problem.

Previous SR methods either predefined the degradation function [6], [7], [13], [14] or learned a degradation model for each LR image [15], [16]. However, in real-world scenarios, the degradation function is often more complex than the predefined ones, such as bicubic downsampling with anti-aliasing filter. Additionally, training a degradation prediction model to estimate the degradation function for each LR image heavily relies on the patterns within the LR images. Consequently the estimation may become inaccurate when applied to LR images with unseen patterns, which can deteriorate the SR results [28].

Considering that the degradation process originates from the image acquisition system, if we have knowledge that the images in the dataset come from similar image acquisition systems, it logically follows that these images should induce the same degradation process. Furthermore, if we possess a dataset containing information about the image acquisition system for each image, we can harness the contrastive learning method to extract information about these image acquisition systems, inherently representing various degradation processes. Our hypothesis posits that incorporating this learned information into the SR generation network will enhance SR performance. This approach eliminates the need for manually defining inaccurate degradation functions. Moreover, this approach defines different types of degradation functions based on the diversity of hardware information, rather than relying solely on individual LR images [15], [16], aligning it more closely with real-world scenarios. Therefore, the proposed SR algorithm can be represented as:

$$f_{SR} = HASR(f_{LR}, h), \tag{2}$$
$$h = F_D(f_{LR}), \tag{3}$$

where h is the feature map representing the degradation information of the current LR image acquisition system, acquired by the Degradation Information Extraction network F_D. Hence, two parts of the loss functions are included in the training process, with its optimization represented by:

$$HASR(f_{LR}) = \underset{HASR,F_D}{\operatorname{argmin}}\{\mathcal{L}_1(f_{SR}, f_{HR}) + \lambda\mathcal{L}_{sup}(F_D(f_{LR}))\}, \tag{4}$$

where $\mathcal{L}_1$ represents the pixel loss, $\mathcal{L}_{sup}$ represents the supervised contrastive loss, and λ is a hyperparameter that controls the tradeoff between $\mathcal{L}_1$ and $\mathcal{L}_{sup}$.

B. Network architecture

Our proposed SR algorithm has two stages: the Degradation Information Extraction stage and the hardware-aware super-resolution (HASR) stage. The first stage aims to extract a discriminative feature map from each LR image, while the second stage is responsible for performing the SR operation. The first stage is facilitated by a pretrained Degradation Information Extraction network, represented as the yellow block on the left side of Fig. 1. Within this initial stage, we use a simple 6-layer convolutional neural network as an encoder and SupCon method to extract the degradation information. Then, we omit the Two-layer Fully Connected (FC) projection part and employ the encoded feature map as the degradation representation. The complete procedure for Degradation Information Extraction is illustrated in Fig. 2, and we will delve into it shortly. The degradation representation obtained from the first stage and the LR feature map from the Shallow Feature Extraction block are combined within the Deep Feature Fusion block. The fusion operation is primarily executed by the proposed HAB. Finally, the super-resolved image is generated through the HR Image Reconstruction block, with the guidance of the hardware information. A detailed description of both stages is presented below.

1) Degradation Information Extraction: The goal of the degradation information learning is to extract a discriminative feature map from each LR image. Building on our previous hypothesis, feature maps originating from different acquisition systems will exhibit dissimilarity, whereas those from the similar acquisition system will manifest similarity.

In this context, we construct our degradation information learning based on the framework of MoCo V2 [20]. The presence of a large dictionary containing a diverse set of negative samples plays a critical role in contrastive learning, as underscored in existing contrastive learning methods [18], [19]. MoCo V2 offers a spacious and consistent dictionary that decouples the dictionary size from the mini-batch size. This feature enriches the pool of negative samples during training, and the size of the dictionary is not limited by the GPU memory.

Furthermore, we introduce positive examples not only by augmenting the anchor image, but also by augmenting images taken from the same acquisition system. Consequently, the LR image datasets in our model are distinctively labeled with corresponding acquisition systems. The SupCon loss function used is as follows:

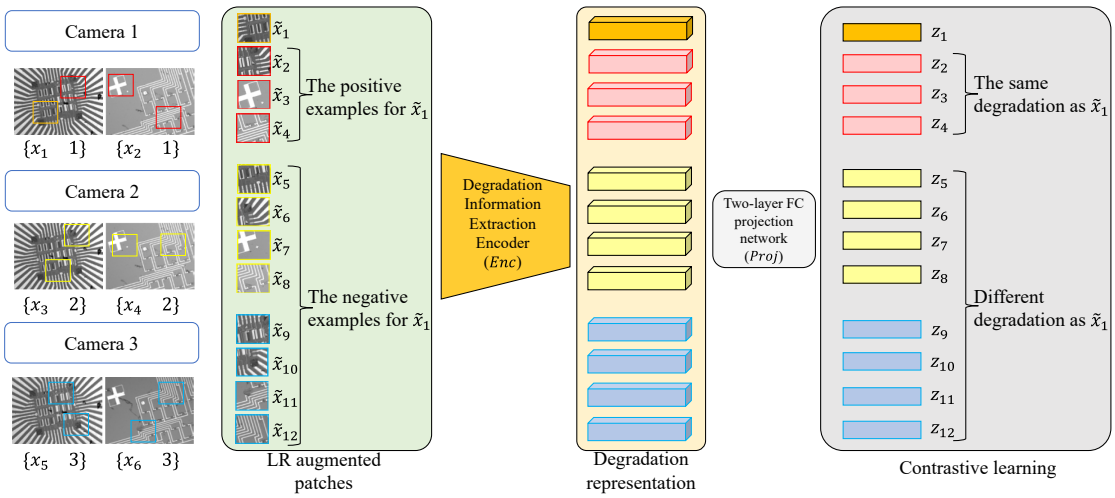


Fig. 2. Illustration of the Degradation Information Extraction.

$$L_{sup} = \sum_{i\in I} L_{sup,i} = \sum_{i\in I} \frac{-1}{|P(i)|} \sum_{p\in P(i)} log \frac{\exp\left(z_i \cdot z_p/\tau\right)}{\sum_{a\in A(i)} \exp\left(z_i \cdot z_a/\tau\right)}. \quad (5)$$

In this equation, $i \in I \equiv \{1 \cdots 2N\}$ represents the index of an arbitrary augmented sample, $z_i = Proj(Enc(\tilde{x}_i))$ represents the feature map generated by the Degradation Information Extraction Encoder and the projection network, the $\cdot$ symbol denotes the inner product, $\tau \in \mathcal{R}^+$ is a scalar temperature parameter, $A(i) \equiv I\backslash\{i\}$ represents all the indices except i, $P(i) \equiv \{p \in A(i): \tilde{y}_p = \tilde{y}_i\}$ represents all the indices that have the same label as the ith augmented sample, and $|P(i)|$ is its cardinality.

Fig. 2 serves as an illustration of (5). At the beginning of each training batch, a set of N randomly sampled {image, acquisition system label} pairs $\{x_n, y_n\}_{n=1\cdots N}$, are selected. The corresponding training data comprises $2N$ pairs, $\{\tilde{x}_i, \tilde{y}_i\}_{i=1\cdots 2N}$, where $\tilde{x}_{2n}$ and $\tilde{x}_{2n-1}$ represent two random augmentations or "views" of x_n $(n = 1\cdots N)$, and $\tilde{y}_{2n-1} = \tilde{y}_{2n} = y_n$. Fig. 2 presents an example with $N = 6$, $i = 1$, $P(1) = \{2,3,4\}, A(1) = \{2,3,\ldots,12\}$, and the labels for the three acquisition systems (different cameras in Fig. 2) are respectively {1,2,3}. Intuitively, for the ith augmented sample, all the other augmented samples with the same label are expected to be positive samples, while the remaining augmented samples are expected to be negative samples. This equation is simply an extension of the classical self-supervised contrastive loss that enables multiple positive examples in a batch of training data.

When the training is completed, like classical contrastive learning methods [18], [20], the degradation representation h_i is used for the SR algorithm in this paper.

Discussion. The proposed degradation information learning does not require the ground-truth degradation process. Its goal is to learn the hidden distinctive characteristics of degraded images taken from the different acquisition systems for distinguishing. Such a good degradation representation can improve the SR network performance, as shown in section IV.

2) HASR network: Given the degradation information extracted from LR images we can integrate this information into an SR network backbone through deep feature fusion. As shown in Fig. 1, our proposed HASR network mainly contains three components: shallow feature extraction, deep feature fusion, and the HR image reconstruction.

A convolution layer is first utilized to extract the *shallow feature map* F_0 from f_{LR}, which can be represented by:

$$F_0 = W_3^{(3,mid)}(f_{LR}), \quad (6)$$

where $W_3^{(3,mid)}$ denotes a convolution layer with filter size 3×3, input channel 3, and output channel mid. mid is a hyper parameter that decides the number of filters of the shallow feature extraction convolution layers. Next, the feature map F_0 and the degradation representation h will go through multiple blocks of the residual group for the *deep feature fusion*. Each residual group takes both the feature map from the previous residual group and the degradation representation h as inputs, and outputs the fused feature map F_i,

$$F_i = H_{ResG}^i(F_{i-1}, h), \quad (7)$$

where H_{ResG}^i represents the ith residual group. More details of the residual group will be presented later. Then, after the last residual group, the fused feature map F_{last} will go through a convolution layer and make the summation with F_0 (see (6)) to create the dense feature map F_{DF} by the global residual learning:

$$F_{DF} = W_3^{(mid,mid)}(F_{last}) + F_0. \quad (8)$$

Finally, the dense feature map F_{DF} will go through the *HR reconstruction* decoder. To effectively upscale the dense feature map F_{DF}, the decoder utilizes efficient sub-pixel CNN (ESPCNN) [29] followed by a single convolution layer to output the three-channel SR images:

$$f_{SR} = W_3^{(mid,3)}\left(H_{\text{ESPCN}}(F_{DF})\right), \tag{9}$$

$$H_{\text{ESPCN}} = \begin{cases} PS\left(W_3^{(mid,4*mid)}(\cdot)\right) \ if\ upscale = 2, \\ PS(W_3^{(mid,4*mid)}\left(PS\left(W_3^{(mid,4*mid)}(\cdot)\right)\right)) \ if\ upscale = 4, \end{cases} \tag{10}$$

where PS represents the pixel-shuffle operation with the scale factor of 2.

Residual Group: The Residual Group serves as a crucial component in *deep feature fusion*. The incorporation of multi-level skip connections allows abundant low-frequency information to be bypassed, enabling the main network to focus on learning high-frequency information. As shown in Fig. 1 (a), each residual group comprises multiple HABs. The current residual group i takes the previous fused feature map F_{i-1} from the previous residual group and the degradation information h as inputs. Then, F_{i-1} and h go through d HABs. Finally, the residual group outputs the fused feature map F_i with the long skip connection. It can be formulated as:

$$F_i = H_{HAB}^d((\cdots H_{HAB}^1(F_{i-1}, h)\cdots), h) + F_{i-1}, \tag{11}$$

where H_{HAB}^d represents the dth HAB. d is a hyper parameter that determines the number of HABs in each residual group.

Hardware-Aware Block: The detailed structure of the HAB is illustrated in Fig. 1 (b). The current HAB j takes the fused feature map from previous HAB and the degradation information h as inputs. It involves a deep feature extraction module (DFEM) and a dual-path attention mechanism. The DFEM can be either CNN based or Transformer based feature extraction layers. For more details of the structure with DFEM, readers can refer to our supplemental materials. The dual-path attention mechanism involves both channel attention (CA) and spatial attention (SA) paths. The output of the current HAB, H_{HAB}^j can be inferred by:

$$F_i^j = DFEM\left(F_i^{j-1}\right) \otimes Rs\left(L(h)\right) + DFEM(F_i^{j-1}) \otimes L(h), \tag{12}$$

where F_i^j represents the output feature map of the jth HAB of the ith residual group. $j \in \{1 \ldots d\}$, $F_i^0 = F_i$. L represents the two-layer multilayer perceptron (MLP), Rs represents the reshape operation, $\otimes$ represents the element-wise multiplication. If the feature map F_i^{j-1} has the dimension of $\mathbb{R}^{C\times H\times W}$, the degradation information will travel through dual paths before implementing element-wise multiplication with the feature map. The first path contains two fully connected (FC) layers and a reshape operation that projects the dimension of the degradation information to $\mathbb{R}^{1\times H\times W}$ as the spatial attention values. The second path contains two FC layers that project the dimension of the degradation information $\mathbb{R}^{C\times 1\times 1}$ as the channel attention values. During element-wise multiplication, the attention values are broadcasted accordingly: spatial attention values are broadcasted (copied) along the channel dimension, and vice versa. This parallel attention mechanism enables the network to extract more informative features from the degradation information.

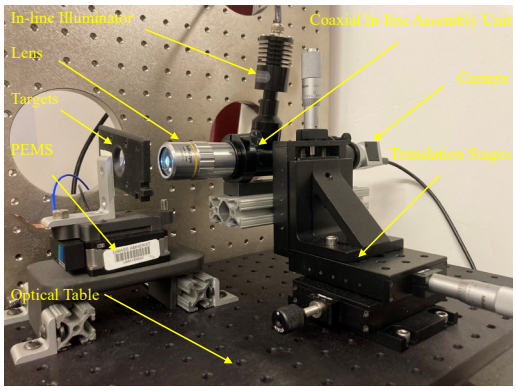


Fig. 3. Image acquisition system.

Discussion. Current SR networks designed to handle multiple degradations, as seen in [8], [30], often combine degradation information with image feature maps and directly input them into the SR network. However, this direct integration using convolution may introduce interference due to the inherent domain gap between degradation information and image features, as highlighted in [10], [15]. In our approach, we utilize degradation information as attention values within dual paths, allowing us to effectively harness this information to adapt to specific degradation scenarios. The spatial attention path focuses on optimizing the connections between adjacent pixels in the image, guided by the degradation information. Meanwhile, the channel attention path is dedicated to optimizing the relationships between feature channels, again guided by degradation information. Subsequently, by optimizing through these two attention paths, we combine their results to achieve the fusion of degradation information and deep feature maps. In Section IV, we also conduct an ablation study on our fusion method to empirically demonstrate its effectiveness.

IV. Experiments

In this section, we first introduce the super-resolution dataset named Real-Micron created from the real-world micron-scale patterns. We then present the experiment details and results based on open-source synthetic datasets, real-world datasets including DRealSR [31], ImagePairs [32], and Real-Micron dataset. Ablation study is presented at last.

A. Real-Micron Datasets

We collected sets of LR and HR images at multiple resolutions with the combination of three Basler cameras and three Mitutoyo objectives to build a dataset for learning and evaluating the super-resolution models of the real-world micron-scale patterns.

1) Setup of Image Acquisition: The image acquisition system was mounted on an optical table to keep it as stable as possible, as shown in Fig. 3. Auto-focus algorithm [33] was applied during the acquisition process. The cameras and objectives

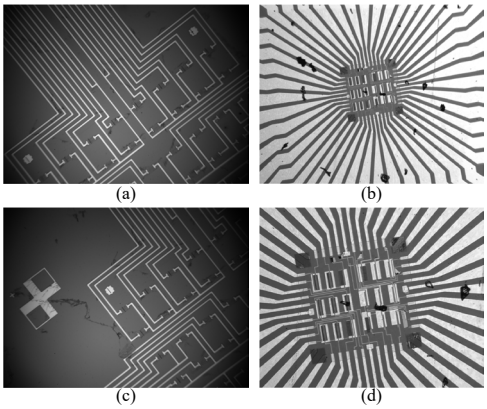


Fig. 4. Sample images from our acquisition system. (a) and (c) are capturing the same target using the same acA4112 - 30 μm camera. (a) uses 10× Objective and (c) uses 20× Objective. (b) and (d) are capturing the same target using the same acA1300 - 200 μm camera. (b) uses 5× Objective and (d) uses 10× Objective.

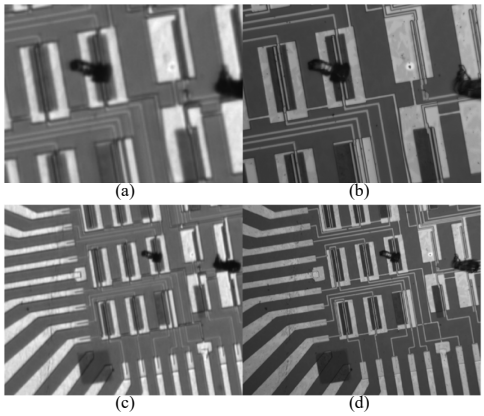


Fig. 5. Registered image pairs. (a) and (b) are captured by the same camera, Basler acA640 - 750 μm, different objectives, 5 × and 20 × , respectively. (c) and (d) are captured by the same camera, Basler acA1300 - 200 μm, different objectives, 5× and 20× , respectively.

could be easily unscrewed from the coaxial in-line assembly unit. The working distance could be adjusted by the translation stages and fine-tuned by the piezoelectric motion stage (PEMS).

Four different samples were captured by the acquisition system, including the US Air Force Hi-Resolution target and three different micro-scale circuits as shown in Fig. 4. Different parts of each sample were captured by three different cameras. For each camera, images with three different resolutions were captured using the objectives with 20, 10, and 5 magnifications (20×, 10×, and 5×). After image pair registration, the images captured by 20×, 10× and 5× objectives were respectively ground-truth (GT), two times downsampled LR images (LR-×2), and four times downsampled LR images (LR-×4) as the super-resolution dataset. Furthermore, each LR image was labeled by the camera number, showing which camera it came from.

To reduce sensor noise, we captured $L(L = 10)$ consecutive images for each scene as [34] did. Therefore, the raw images are computed by:

$$X_{raw} = \frac{1}{L}\sum_{l=1}^{L} X_l, \qquad (13)$$

where X_l represents the lth consecutive image. Each of these L consecutive images was captured under constant illumination and without interframe motion.

2) Image Pair Registration: To create the pixel-wise aligned image pairs in different resolutions, we utilized the image pair registration algorithm. For the images acquired by each model of the camera, we implemented image registration algorithms between the 5 × and 10 × objectives, the 10 × and 20 × objectives as the two times downsampling pixel-wise aligned pairs, and the 5 × and 20 × objectives as the four times downsampling pixel-wise aligned pairs. However, obtaining pixel-wise aligned image pairs is not straightforward due to duplicate patterns and unstable luminance conditions in the circuit targets. As shown in Fig. 4, conventional image registration algorithms such as SIFT *[35]*, SURF *[36]*, and SuperGlue *[37]* cannot produce accurate results. To obtain accurate image pair registration of our dataset, we designed a coarse-to-fine registration algorithm that maximizes the structural similarity index measure (SSIM) between the transformed LR image and the HR image.

TABLE I Cameras and Lenses Used in Data Collection.

Cameras	Lenses
Basler acA640 - 750 μm	5×
Basler acA1300 - 200 μm	10×
Basler acA4112 - 30 μm	20×

Note: All lenses are Mitutoyo Plan Apo Infinity Corrected Long WD Objective

Denote I_{HR} and I_{LR} as the HR and the LR images to be registered. The final target of our algorithm is to maximize the objective function:

$$\max_{TransM} SSIM[Crop(TransM \cdot I_{LR}),\ \ I_{HR}], \qquad (14)$$

where $TransM$ is the affine transformation matrix, $Crop$ is the cropping operation to make the transformed I_{LR} the same size as I_{HR} , $\|\cdot\|_{SSIM}$ is the structural similarity index measure (SSIM).

To find the accurate $TransM$, point correspondences between I_{HR} and I_{LR} must be also accurate. We first implemented the registration algorithm in [38] to obtain the point correspondences since it solved the problem of duplicate and deformable patterns. Then, given the scale factor from the magnification of the lenses, other unknown parameters in $TransM$ can be calculated from the point correspondences using the least square method. Next, several cropped candidates will be proposed based on the inverse transformation of I_{HR}. Due to the stability of our acquisition system, scale and translation are the principal transformations. Therefore,

identifying four corners of $I_{HR} \cdot inverse(TransM)$ will be enough for proposing the candidates. Last, the SSIM values will be calculated to pick the best candidate.

The detailed registration algorithm is included in the supplemental materials. The code was written in MATLAB 2022b and is available at https://github.com/cucum13er/Hardware-Aware-Super-Resolution/tree/main/Matlab_github.

Fig. 5 shows examples of the registered image pairs from different cameras, and conspicuous field of view (FOV) differences can be observed between the two cameras. We will present the quantitative results in the next subsections to prove that the images taken from different cameras have different degradation processes. Note: it is difficult to observe the degradation differences among cameras by eyes.

B. Experimental setup

To train the Degradation Information Extraction network, we first synthesized LR images according to (1). The simulation of the degradation process included Gaussian blurring and bicubic downsampling.

To evaluate the performance of the degradation information, we implemented five different isotropic Gaussian kernels with bicubic downsampling on the HR images in the synthetic experiment. Five different image acquisition systems were simulated by the five 2D-Gaussian blurring kernels with σ^2 setting to $[0.5, 1.0, 2.0, 3.0, 4.0]$, respectively. Following [10], the size of the Gaussian kernels was fixed to 21×21. We also used LARS optimizer [39] to train the degradation information network with the SupCon loss with 128 batch size and 2 augmented views (see (5)). During training, each of the 128 LR image patches was randomly selected from different degradation processes and cropped into size 160×160. Data augmentation was then performed through random flipping and transposing. The start learning rate was 0.4, and we performed 1000 iterations of training. We separated the training images of the DIV2K [40] dataset into 70%, 10%, and 20% as the training set, validation set and one of the test sets respectively. We also included Flickr2K [41], BSD100 [42], Set5 [43], Set14 [44], and Urban100 [45] as the test sets.

We employed the same training process for the real-world datasets (DRealSR [31], ImagePairs [32], and Real-Micron) as for synthetic datasets, with the difference being that we already had real LR images and different camera labels in real-world datasets.

TABLE II The classification results for real-world datasets.

Method (backbone)	DRealSR	Real-Micron
Supervised (6-layer CNNs)	98.9%	85.9%
SimCLR + SupCon (6-layer CNNs)	95.7%	84.4%
MoCo-V2 + SupCon (6-layer CNNs)	**100%**	**93.8%**

We evaluated our HASR model using both the synthesized LR-HR image pairs with known blurring kernels and downsampling methods and the real-world LR-HR image pairs with unknown degradation processes.

For synthetic experiments, we used training images from the DIV2K and Flickr2K datasets as the training set and the Set5, Set14, and Urban100 benchmark datasets as the testing set. HR images were degraded into LR images using the same methods as we used to train the Degradation Information Extraction network. We trained HASR network with a combination of SupCon loss and $L1$ loss for $200K$ iterations, with the learning rate of 1×10^{-4} for the SR part, 1×10^{-9} for the degradation information part, and decaying half every $40K$ iterations. The hyperparameter λ was set to 0.1, and we used the Adam [46] optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$ for optimization.

We used the registered image pairs from DRealSR, ImagePairs, and Real-Micron datasets to conduct real-world experiments. DRealSR consisted of real-world LR and HR images collected by zooming DSLR cameras. The dataset included five DSLR cameras (Canon, Nikon, Olympus, Panasonic, and Sony), corresponding to five different acquisition systems of our Degradation Information Extraction network. ImagePairs used a beam-splitter to capture the same scene by a low resolution camera (LRC) and a high resolution camera (HRC). The LRC can be the sixth acquisition system for our Degradation Information Extraction network. For the $\times 2$ experiments, we combined the DRealSR and ImagePairs for training and testing. For the $\times 4$ experiments, we only used DRealSR for training and testing since ImagePairs does not have the ground-truth HR images. However, the Real-Micron dataset does not have enough training samples. Therefore, we implemented transfer learning to improve the model performance. We first separated the Real-Micron dataset into 80% and 20% as the training and testing datasets, respectively. Next, the best model we have trained on the Real-Micron dataset for extracting the hardware information (MoCo-V2+SuperCon) was selected to initialize the Degradation Information Extraction part of HASR network. Then, the other

TABLE III The results of 5-class classification for isotropic Gaussian kernels.

Method (backbone)	DIV2K	Flickr2K	BSD100	Set5	Set14	Urban100
MoCo-V2 (ResNet-18)	71.8%	76.4%	68%	56%	65.7%	72.8%
Supervised (ResNet-18)	89.4%	85.6%	90.4%	76%	85.7%	83.4%
Supervised (6-layer CNNs)	95.9%	**96.1%**	92.6%	84.0%	91.4%	93.2%
SimCLR + SupCon (ResNet-18)	87.9%	86.6%	87.2%	72.0%	78.6%	82.2%
SimCLR + SupCon (6-layer CNNs)	94.3%	95.6%	93.8%	88.0%	95.7%	93.8%
MoCo-V2 + SupCon (ResNet-18)	89.2%	91.1%	88.0%	72%	84.3%	84.8%
MoCo-V2 + SupCon (6-layer CNNs)	**96.1%**	95.5%	**94.6%**	**88.0%**	**95.7%**	**94.8%**

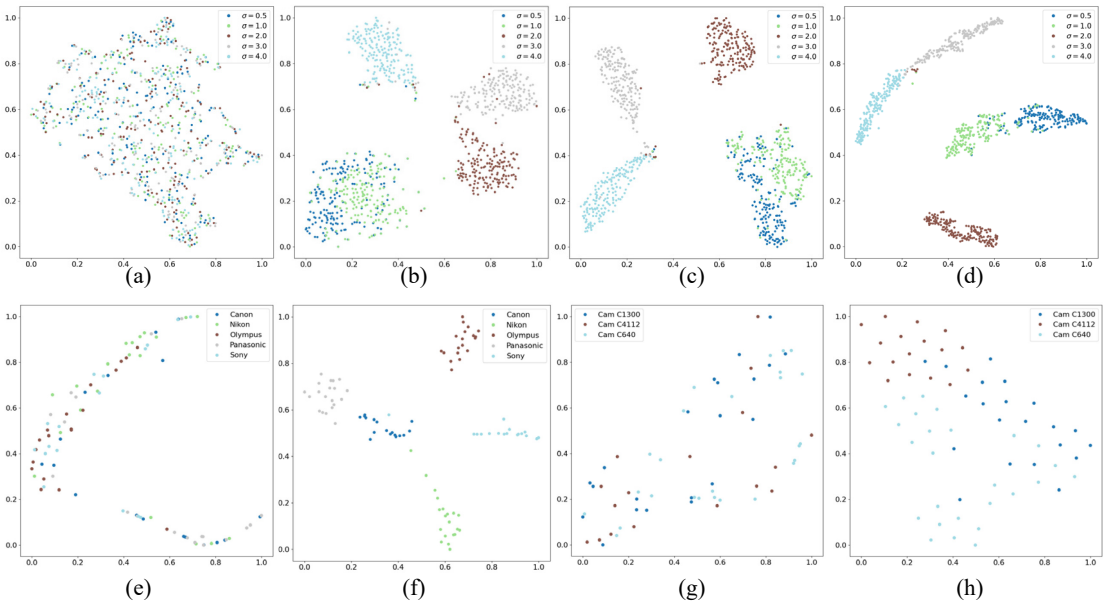


Fig. 6. Visualization of the degradation information. (a) – (d), visualization of the degradation information from the synthetic dataset DIV2K with different kernel width $\boldsymbol{\sigma}$. The methods used from left to right: before training, SimCLR+SupCon, Supervised, MoCo-V2+SupCon. (e) – (h), visualization of the degradation information from real-world datasets. (e) DRealSR dataset, before training. (f) DRealSR dataset, MoCo-V2+SupCon. (g) Real-Micron, before training. (h) Real-Micron, MoCo-V2+SupCon.

part of HASR network was initialized by the model we trained on the synthetic experiments (DIV2K and Flickr2K datasets). Finally, we trained the HASR network on the Real-Micron dataset for 20K iterations by freezing the Degradation Information Extraction part and partially freezing the residual groups. Specifically, we experimentally quantified the generality versus specificity of neurons in each residual group of the network by freezing the trainable parameters of different residual groups during the fine-tuning. Further analysis and more details are included in the next subsections and supplemental materials.

We conducted experiments using PyTorch and MMediting [47]. NVIDIA RTX3090 and RTX2080ti GPUs were used for training and testing. The source code and pre-trained models is available at https://github.com/cucum13er/mmagic/tree/0.x.

C. Experiments on The Degradation Information Extraction Network

To evaluate the performance of the degradation information, we compared the supervised contrastive methods to unsupervised contrastive methods, including SimCLR [18] and MOCO V2 [20], and the supervised method. To be fair, we used the same backbones ResNet-18 [48] and 6-layer CNNs [15] to compare different methods. For the performance evaluation, we added a classification head (a supervised linear classifier: two fully connected layers followed by SoftMax) to the backbone

TABLE IV PSNR and SSIM comparison of CNN based models on open-source synthetic datasets.

Kernel width (σ)		**1.0**	**2.0**	**3.0**	**4.0**	**1.0**	**2.0**	**3.0**	**4.0**	**1.0**	**2.0**	**3.0**	**4.0**
Method	scale	Set5 (PSNR/SSIM)				Set14 (PSNR/SSIM)				Urban100 (PSNR/SSIM)			
RDN	× 2	30.63	26.13	23.90	22.56	27.74	24.21	22.53	21.52	24.58	21.05	19.56	18.71
		0.878	0.748	0.659	0.602	0.808	0.653	0.565	0.515	0.803	0.600	0.491	0.434
Real-ESRGAN		27.94	27.03	25.78	24.78	26.00	25.31	24.24	23.21	23.08	22.24	20.97	19.99
		0.839	0.812	0.770	0.731	0.747	0.711	0.656	0.609	0.751	0.705	0.633	0.571
DASR		35.17	32.64	25.30	23.19	30.66	28.64	23.38	21.81	**28.95**	26.32	20.46	19.09
		0.934	**0.902**	0.732	0.640	0.875	0.820	0.624	0.544	**0.908**	0.840	0.553	0.454
CNN HASR (MoCo)		35.27	32.89	30.48	28.89	30.98	29.12	27.14	**25.95**	28.60	26.46	24.00	22.89
		0.928	0.896	0.850	0.811	0.874	0.824	0.749	**0.698**	0.900	0.845	0.749	0.691
CNN HASR (SimCLR)		**35.40**	**32.95**	**30.57**	**28.95**	**31.19**	**29.34**	**27.22**	25.75	28.80	**26.72**	**24.21**	**22.99**
		0.932	0.896	**0.851**	**0.813**	**0.880**	**0.827**	**0.751**	0.685	0.905	**0.852**	**0.759**	**0.697**
RDN	× 4	29.10	25.96	23.86	22.54	26.22	24.02	22.50	21.50	23.75	21.42	20.09	19.25
		0.824	0.736	0.656	0.601	0.716	0.634	0.562	0.514	0.733	0.616	0.535	0.487
Real-ESRGAN		26.16	25.63	24.56	23.33	24.37	24.13	23.18	22.06	21.77	21.26	20.23	19.14
		0.721	0.704	0.668	0.616	0.665	0.648	0.604	0.550	0.677	0.654	0.600	0.530
DASR		29.98	29.91	29.28	28.05	26.34	26.29	25.87	25.03	24.21	24.06	23.61	22.81
		0.859	**0.856**	**0.840**	**0.804**	0.736	**0.731**	**0.709**	**0.670**	**0.750**	**0.742**	**0.719**	**0.676**
CNN HASR (MoCo)		**30.66**	**30.05**	**29.83**	**29.73**	**26.99**	**26.55**	**26.25**	**25.45**	**25.36**	**24.95**	**24.33**	**23.64**
		0.847	0.826	0.819	0.803	**0.749**	0.726	0.706	0.653	0.735	0.709	0.704	0.657
CNN HASR (SimCLR)		29.96	30.03	29.35	27.97	26.58	26.53	26.08	25.18	23.98	23.86	23.46	22.61
		0.842	0.842	0.824	0.791	0.727	0.723	0.700	0.660	0.739	0.735	0.711	0.666

TABLE V PSNR and SSIM comparison of Transformer based models on open-source synthetic datasets.

Kernel width (σ)		**1.0**	**2.0**	**3.0**	**4.0**	**1.0**	**2.0**	**3.0**	**4.0**	**1.0**	**2.0**	**3.0**	**4.0**
Method	scale	Set5 (PSNR/SSIM)				Set14 (PSNR/SSIM)				Urban100 (PSNR/SSIM)			
DiffBIR	× 2	27.24	26.24	25.54	24.14	23.98	23.60	23.02	22.18	22.28	21.76	20.76	19.70
		0.785	0.756	0.736	0.681	0.632	0.624	0.583	0..544	0.685	0.654	0.585	0.508
HAT		30.64	26.14	23.91	22.57	27.81	24.22	22.54	21.52	24.66	21.05	19.56	18.72
		0.891	0.767	0.675	0.613	0.817	0.660	0.572	0.522	0.808	0.599	0.487	0.429
Restormer		31.45	29.57	27.50	25.84	28.75	26.87	24.88	23.57	25.45	23.49	21.66	20.56
		0.891	0.843	0.777	0.717	0.838	0.766	0.672	0.608	0.825	0.735	0.620	0.544
SwinIR		32.31	27.71	26.95	23.53	29.41	25.81	24.81	23.66	26.26	22.27	21.11	19.96
		0.916	0.812	0.704	0.699	0.825	0.656	0.593	0.540	0.810	0.589	0.483	0.420
Swin-Transformer		**37.34**	**34.29**	**31.70**	**30.55**	**31.23**	**30.39**	**28.74**	**26.03**	**28.55**	**26.45**	**24.46**	**22.89**
HASR (MoCo)		**0.924**	**0.901**	**0.861**	**0.828**	**0.872**	**0.823**	**0.758**	**0.658**	**0.882**	**0.821**	**0.715**	**0.621**
DiffBIR	× 4	24.55	24.66	23.84	22.32	22.53	22.30	21.78	21.43	20.67	20.54	20.24	19.76
		0.703	0.703	0.672	0.611	0.566	0.543	0..514	0.495	0.587	0.573	0.549	0.521
HAT		29.36	25.98	23.87	22.54	26.39	24.05	22.51	21.51	24.18	21.47	20.09	19.26
		0.850	0.757	0.672	0.613	0.732	0.643	0.569	0.521	0.756	0.621	0.536	0.486
Restormer		26.98	26.59	26.26	25.32	24.59	24.26	23.99	23.37	21.82	21.43	21.06	20.70
		0.749	0.738	0.722	0.689	0.658	0.640	0.624	0.591	0.628	0.605	0.582	0.557
SwinIR		30.14	26.63	24.15	22.45	26.94	24.39	22.50	22.83	24.90	22.69	20.64	20.15
		0.855	0.763	0.677	0.594	0.726	0.638	0.549	0.533	0.725	0.608	0.513	0.450
Swin-Transformer		**32.18**	**31.17**	**30.10**	**29.49**	**27.55**	**26.97**	**26.19**	**25.27**	**26.21**	**25.54**	**25.08**	**24.08**
HASR (MoCo)		**0.868**	**0.863**	**0.839**	**0.819**	**0.744**	**0.743**	**0.703**	**0.656**	**0.723**	**0.725**	**0.715**	**0.665**

and loaded the pretrained weights into the backbone. We then froze the weights of the backbone and trained the whole network for a small number of epochs.

TABLE III presents a comparison of the classification performance using different methods for the isotropic Gaussian kernels. The results demonstrate that the supervised contrastive and the classic supervised methods outperform the unsupervised methods in this classification task due to their use of label information. As noted in [12], supervised contrastive learning can improve classifier accuracy and robustness. We, therefore, select this method to extract degradation information, as supported by the results in TABLE I. Surprisingly, simple 6-layer CNNs outperform ResNet-18 in all the three methods because they can effectively represent degradation information, unlike the more complex ResNet-18, which has too many redundant trainable parameters. Additionally, limited training data and iterations can cause overfitting issues with ResNet-18.

Given the 6-layer CNNs perform well in synthetic experiments, we opt to use them to train the real-world datasets. TABLE II presents the classification results of these real-world datasets. The supervised contrastive method with MoCo-V2 structure achieves the best classification accuracy on average. We, therefore, finalized our Degradation Information Extraction network with 6-layer CNNs as the backbone, MoCo-V2 as the training algorithm, SuperCon loss as the loss function.

To further visualize the learned degradation information, we used the T-SNE method [49] to cluster LR images from both synthetic and real-world datasets. The degradation representations of those LR images were fed to the Degradation Information Extraction networks and then visualized. Fig. 6 shows the visualization results, where the first row includes the results of the synthetic dataset DIV2K with five different isotropic Gaussian blurring kernels and the second row includes the results of the DRealSR dataset with five DSLR cameras and the results of Real-Micron dataset with three different Basler cameras. The visualization results reveal the feature vectors are

TABLE VI PSNR and SSIM results on DRealSR and ImagePairs datasets.

Method	Scale	Canon		Nikon		Olympus		Panasonic		Sony		LRC
		× 2	× 4	× 2	× 4	× 2	× 4	× 2	× 4	× 2	× 4	× 2
RDN	Backbone: CNN	32.41	28.56	32.49	28.05	32.07	28.07	32.21	28.14	31.85	29.27	22.30
		0.893	0.834	0.885	0.804	0.872	0.771	0.865	0.788	0.845	0.821	0.694
Real-ESRGAN		27.53	24.83	29.68	26.95	29.66	26.31	29.54	26.03	26.68	26.53	21.86
		0.868	0.793	0.886	0.798	0.867	0.750	0.848	0.748	0.810	0.766	0.785
DASR		30.87	27.91	31.71	27.99	30.52	27.73	31.08	28.05	28.14	28.52	21.86
		0.898	0.844	0.901	0.831	0.881	0.796	0.873	0.806	0.831	0.826	0.735
CDC		32.61	30.43	33.12	29.84	31.58	29.31	32.43	30.18	28.63	29.93	22.10
		0.933	**0.898**	**0.930**	**0.874**	**0.909**	**0.832**	**0.903**	**0.847**	**0.851**	0.854	0.785
CNN HASR (MoCo)		34.10	30.78	**34.18**	29.73	33.63	29.77	33.70	30.92	31.61	31.50	25.26
		0.932	0.884	0.917	0.841	0.906	0.811	0.891	0.816	0.843	0.846	**0.829**
DiffBIR	Backbone: Transformer	26.99	26.99	27.51	26.98	27.27	27.17	27.63	27.22	25.84	27.20	21.73
		0.805	0.802	0.774	0.777	0.757	0.739	0.761	0.757	0.724	0.768	0.751
HAT		30.27	27.75	31.50	27.43	31.67	27.47	33.46	28.40	31.79	29.12	21.87
		0.874	0.822	0.892	0.810	0.886	0.778	0.885	0.789	0.875	0.818	0.756
Restormer		30.25	28.62	30.11	28.46	29.82	28.22	30.16	28.57	28.67	28.83	22.37
		0.895	0.855	0.874	0.827	0.862	0.796	0.862	0.806	0.787	0.818	0.759
SwinIR		31.30	28.97	31.68	27.47	30.33	28.02	30.50	27.95	30.82	28.39	21.68
		0.902	0.852	0.886	0.795	0.858	0.780	0.847	0.782	0.855	0.808	0.688
Swin-Transformer		**35.27**	**32.59**	33.88	**31.58**	**34.30**	**30.94**	**34.08**	**31.09**	**31.70**	**32.65**	**25.62**
HASR (MoCo)		0.929	0.893	0.911	0.867	0.908	0.821	0.892	0.839	0.849	**0.879**	0.807

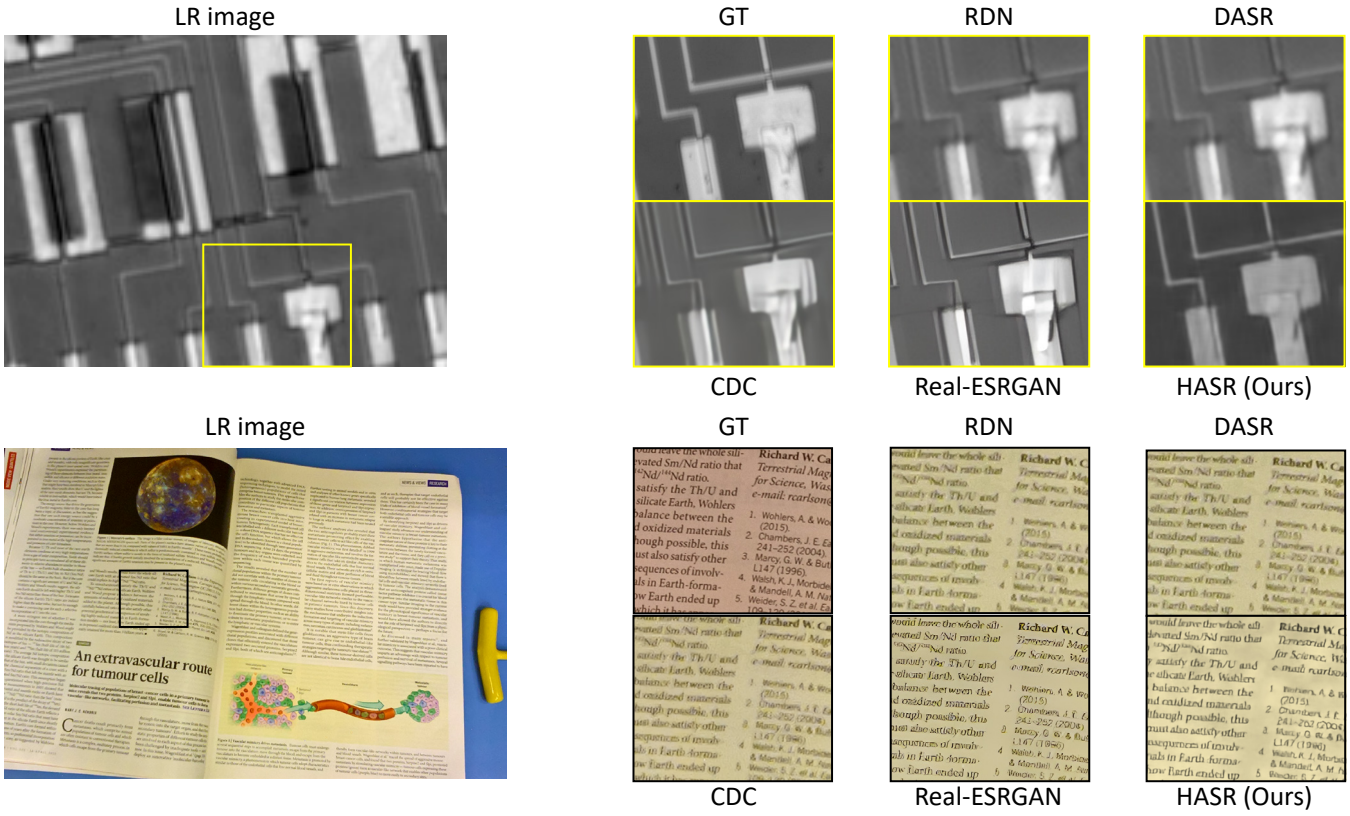


Fig. 7. Qualitative comparison of our model with other works on × **4** super-resolution on the Real-Micron dataset (top) and × **2** super-resolution on the ImagePair dataset (bottom).

well clustered by different degradation kernels or different cameras. MoCo-V2 can distinguish different categories better than other algorithms, as demonstrated in TABLE III and TABLE II. Fig. 6 (h) is less distinguishable because the three Basler cameras have very similar specifications, making their degradation information quite similar.

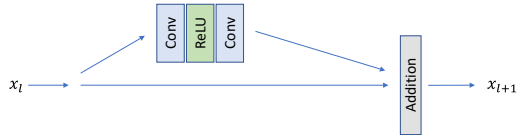
D. Experiments on the HASR Network

We conducted simulation experiments on LR-HR pairs with known blurring kernels and downsampling methods, i.e., isotropic Gaussian blurring kernels with bicubic downsampling method. We compared our CNN based HASR to several recent CNN based SR algorithms, including RDN [50], Real-ESRGAN [6] and DASR [15], using their pretrained models. Furthermore, the adoption of stronger Transformer backbones has gained significant traction recently. To validate that our proposed degradation information's impact on enhancing SR is not confined to the SR generation network's backbone, we conducted experiments using Transformer based backbones as well, including DiffBIR [51], HAT [52], SwinIR [53] with their pretrained models, fine-tuned Restormer [26], and Swin-Transformer based HASR. TABLE IV shows the PSNR and SSIM comparison results among the CNN based backbones, indicating that with the assistance of the degradation information, our CNN based HASR algorithm outperforms other algorithms, especially when the LR images are heavily blurred by a greater σ value. TABLE V presents a comparison of PSNR and SSIM results among the Transformer based backbones. Similar to TABLE V, we find that the inclusion of degradation information consistently enhances the quality of SR results. Taking advantage of both local self-attention mechanism and the shifted window scheme, the Swin-Transformer based HASR achieves the best performance across most test datasets.

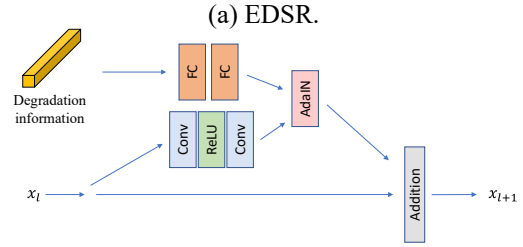
We also conducted experiments on real-world LR-HR image pairs using DRealSR and ImagePairs datasets for the × 2 experiments and DRealSR dataset for the × 4 experimens. We then used Real-Micron dataset for another real-world dataset evaluation. As shown in TABLE VI, our HASR algorithm consistently achieves higher PSNR and SSIM values compared

TABLE VII PSNR and SSIM results on Real-Micron dataset.

Method	Scale	Cameras		
		C640	C1300	C4112
RDN	× 2	22.06	22.01	15.01
		0.854	0.846	0.761
DASR		21.98	11.96	12.02
		0.854	0.560	0.740
CDC		21.83	21.50	12.13
		0.862	0.867	0.761
CNN HASR (MoCo)		**28.99**	**28.07**	**21.02**
		0.921	**0.900**	**0.841**
RDN	× 4	19.90	17.18	11.05
		0.839	0.823	0.725
DASR		19.89	17.17	11.02
		0.845	0.830	0.727
CDC		19.57	17.15	11.11
		0.836	0.825	0.726
CNN HASR (MoCo)		**27.79**	**25.02**	**21.54**
		0.904	**0.914**	**0.869**



(a) EDSR.



(b)EDSR+AdaIN.

Fig. 8. Comparison of residual blocks in original EDSR and EDSR with AdaIn fusion.

to most other algorithms. It's worth noting that CDC [31] exhibits higher SSIM values in certain cases, as it dissects an image into three components (flat, edges, and corners) and reconstructs each component individually. In contrast, our proposed method is designed to reconstruct the entire image as a whole. However, an interesting avenue for future research could involve adapting CDC to incorporate degradation information.

For the evaluation of the Real-Micron dataset, as stated in subsection IV.B, we initialized the HASR model by employing the pretrained Degradation Information Extraction network obtained from the Real-Micron dataset, along with the pretrained HASR network acquired from the synthetic experiments. We utilized CNN based HASR in the evaluation due to the relatively small scale of the Real-Micron dataset.

TABLE VI and TABLE VII shows the PSNR and SSIM results and confirms that our proposed HASR network achieved better quantitative evaluation results than other state-of-the-art algorithms. Additionally, Fig. 7 shows the SR visualization results on the Real-Micron and ImagePairs datasets, demonstrating that the proposed HASR network successfully reconstructs detailed textures and edges in the HR images, yielding better-looking SR outputs compared to other methods. While Real-ESRGAN produces sharper-looking details, it introduces some artifacts due to its adversarial model. The adversarial model prioritizes generating visually pleasing SR images over SR images closer to the input LR images, resulting in a tradeoff between the visual quality and the quantitative performance. Note the PSNR metric fundamentally disagrees with the subjective evaluation of human observers [1]. If users care more about the quantitative performance in SR applications, e.g., using the HASR for product pattern inspection and metrology in manufacturing processes, the SR results must be as close as possible to the ground-truth rather than guessing a more visually pleasing image.

E. Ablation Studies

We first evaluated the effectiveness of the degradation information in the network by conducting ablation experiments using three different backbones. Then, we evaluated the effectiveness of the dual-path attention mechanism by conducting an ablation experiment using different fusion methods. Finally, we evaluated the performance of transfer learning on Real-Micron dataset by training and evaluating various models.

1) Analysis on Degradation Information: The backbones we have implemented include CNN based HASR, Restormer based HASR, and EDSR [4] with Adaptive Instance Normalization (AdaIN) [22]. To disregard the degradation information, we set $\lambda = 0$ for $\mathcal{L}_{\text{sup}}$ of the HASR networks and compared the experiment results of these models to the results of previous HASR networks for the first two comparisons. To explore the generalizability of the degradation information, we conducted an experiment on another SR backbone with a different fusion method, EDSR with AdaIN fusion method. For this experiment, we made specific modifications to the residual blocks of EDSR. Specifically, we used the two-FC-layer projected degradation information as the style feature map for AdaIN, while the feature map from the original residual blocks served as the content feature map. These two feature maps were then combined using an AdaIN layer. Fig. 8 illustrated both original and modified residual blocks. Similarly, we trained two models for this architecture with $\lambda = 0.1$ and $\lambda = 0$, respectively.

TABLE IX displays the PSNR and SSIM results for these three models ($HASR^{CNN}$, $HASR^{RT}$, $EDSR$ respectively represent CNN HASR, Restormer HASR, and EDSR backbones). It is evident that the inclusion of degradation information enhances the performance of both SR networks, confirming the effectiveness of this approach.

2) Analysis on Feature Fusion: To evaluate the effectiveness of the dual-path attention mechanisms, we conducted experiments of different fusion approaches of the CNN based HASR network. Specifically, we compared the original HASR with single path attention (either only spatial or channel attention) and channel attention outside of RCAB [23]. Readers can refer to Supplemental Materials for more details. TABLE VIII shows the PSNR and SSIM comparison of different fusion methods.

TABLE VIII PSNR and SSIM results of different fusion methods.

Kernel width (σ)		**1.0**	**2.0**	**3.0**	**4.0**	**1.0**	**2.0**	**3.0**	**4.0**	**1.0**	**2.0**	**3.0**	**4.0**
Method	scale	Set5				Set14				Urban100			
SA only	× 4	29.87	29.66	28.39	28.05	26.41	26.11	25.71	24.54	23.53	23.94	22.95	22.81
		0.829	0.825	0.783	0.779	0.722	0.694	0.677	0.625	0.668	0.662	0.632	0.609
CA only		29.29	29.41	29.01	28.63	26.32	25.78	24.76	24.18	24.10	24.08	23.33	22.67
		0.803	0.810	0.794	0.764	0.714	0.702	0.663	0.634	0.687	0.667	0.645	0.617
CA outside of RCAB		29.33	28.59	28.44	27.71	25.79	25.08	25.33	24.16	23.38	23.26	22.40	21.89
		0.812	0.790	0.796	0.760	0.700	0.682	0.641	0.609	0.660	0.651	0.617	0.574
CNN HASR (MoCo)		**30.66**	**30.05**	**29.83**	**29.73**	**26.99**	**26.55**	**26.25**	**25.45**	**25.36**	**24.95**	**24.33**	**23.64**
		0.847	**0.826**	**0.819**	**0.803**	**0.749**	**0.726**	**0.706**	**0.653**	**0.735**	**0.709**	**0.704**	**0.657**

TABLE IX PSNR and SSIM comparisons with/without degradation information.

Method	Scale	Canon	Nikon	Olympus	Panasonic	Sony
$HASR^{CNN}_{\lambda=0}$	× 4	30.51	**29.79**	29.68	30.77	30.30
		0.874	0.830	0.805	**0.819**	0.839
$HASR^{CNN}_{\lambda=0.1}$		**30.78**	29.73	**29.77**	**30.92**	**31.50**
		0.884	**0.841**	**0.811**	0.816	**0.846**
$HASR^{RT}_{\lambda=0}$		28.62	28.46	28.22	28.57	28.83
		0.855	0.827	0.796	0.806	0.818
$HASR^{RT}_{\lambda=0.1}$		**30.50**	**29.56**	**30.00**	**29.31**	**30.46**
		0.896	**0.851**	**0.807**	**0.821**	**0.834**
$EDSR_{\lambda=0}$		30.32	29.32	29.46	29.74	29.63
		0.870	0.829	0.790	**0.816**	0.820
$EDSR_{\lambda=0.1}$		**31.69**	**29.55**	**29.73**	**30.28**	**30.09**
		0.883	**0.836**	**0.793**	0.814	**0.831**

The method which performed the worst was the one where fusion occurred outside of the RCAB. This outcome can be attributed to the absence of degradation information during the deep feature extraction process, which occurred inside RCAB. Similarly, methods employing a single path, be it the CA or SA path, exhibited worse performance. These single-path methods lack connections between adjacent pixels or feature channels, making them less effective compared to the proposed fusion method with dual-path.

3) Analysis on Transfer Learning: To evaluate the effectiveness of transfer learning on the Real-Micron dataset, we conducted two sets of experiments using the CNN based HASR network. Firstly, we trained the HASR network using only the Real-Micron training data, with the degradation information part pretrained and the HASR part randomly initialized. Secondly, we trained the network using the same training data with both pretrained degradation information and HASR parts. For the latter, we froze different residual groups in the models during training. Fig. 9 shows the PSNR and SSIM results of both transfer learning metrics. More training details are included in the supplemental materials.

The results indicate that transfer learning outperforms direct training from scratch when the weights of the first one, two or three residual groups are frozen. This is reasonable due to two factors. Firstly, the Real-Micron dataset has fewer LR-HR image pairs than other public datasets like ImagePairs and DRealSR, making overfitting a potential issue during training from scratch. Secondly, by using the pretrained model (DIV2K+Flirck2K) to initialize the HASR, the SR performance can be improved. However, since the pretrained model has domain gaps with the Real-Micron dataset, the best performance was achieved when unlocking the weights of the last and penultimate residual groups. This approach locks in the learned generic features from pretrained model, while providing enough learnable parameters for learning the unique features of the Real-Micron dataset.

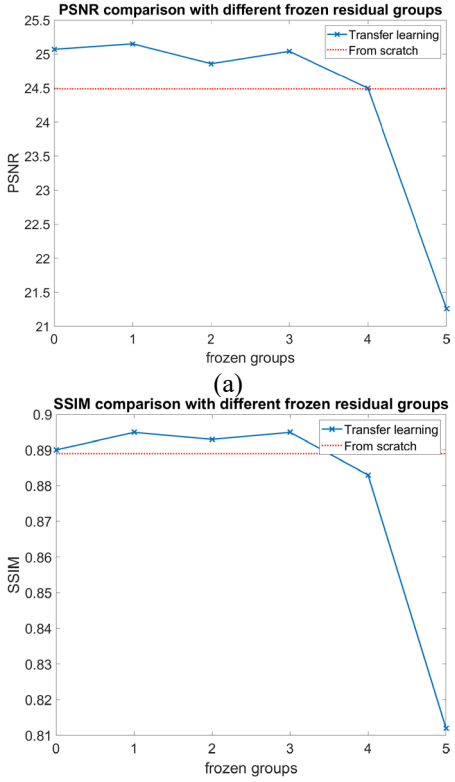


Fig. 9. PSNR and SSIM comparison of transfer learning on Real-Micron dataset.

V. Conclusion

In this study, we propose a blind SR method that can handle various degradation processes of different image acquisition systems by extracting and integrating the prior hardware information. By the inclusion of HAB, both Transformer based and CNN based HASR networks outperform conventional approaches by not relying on predefined or ground-truth degradation kernels. Results from both synthetic and real-world datasets demonstrate the effectiveness of the proposed method in handling blind SR problems. Future work will extend our method to more state-of-the-art SR frameworks such as CDC and verify the effectiveness of the degradation information in these frameworks. Additionally, the effective utilization of prior hardware knowledge to enhance image quality represents a promising avenue for exploration. Algorithms developed on the basis of such hardware information hold significant potential for practical applications.

However, our HASR method may have limitations when handling input LR images acquired from hardware that significantly deviates from the training data. In such cases, the HASR network cannot accurately predict the unknown hardware degradation, resulting in a decline of SR performance. Moreover, obtaining labeled device sources to use as training data for the HASR method can be challenging, which adds to the difficulty of acquiring the necessary data.

Acknowledgment

This work is supported in part by the National Science Foundation (CMMI #1916866, CMMI #1942185, and CMMI #1907250). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] C. Ledig *et al.*, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 105–114. doi: 10.1109/CVPR.2017.19.

[2] B. Wronski *et al.*, "Handheld multi-frame super-resolution," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–18, 2019.

[3] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[4] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA: IEEE, Jul. 2017, pp. 1132–1140. doi: 10.1109/CVPRW.2017.151.

[5] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.

[6] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-esrgan: Training real-world blind super-resolution with pure synthetic data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1905–1914.

[7] R. Zhou and S. Susstrunk, "Kernel Modeling Super-Resolution on Real Low-Resolution Images," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 2433–2443. doi: 10.1109/ICCV.2019.00252.

[8] K. Zhang, W. Zuo, and L. Zhang, "Learning a Single Convolutional Super-Resolution Network for Multiple Degradations," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Jun. 2018, pp. 3262–3271. doi: 10.1109/CVPR.2018.00344.

[9] S. Bell-Kligler, A. Shocher, and M. Irani, "Blind super-resolution kernel estimation using an internal-gan," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[10] J. Gu, H. Lu, W. Zuo, and C. Dong, "Blind super-resolution with iterative kernel correction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1604–1613.

[11] J. Liang, K. Zhang, S. Gu, L. Van Gool, and R. Timofte, "Flow-based kernel prior with application to blind super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10601–10610.

[12] P. Khosla *et al.*, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18661–18673, 2020.

[13] H. Ren, A. Kheradmand, M. El-Khamy, S. Wang, D. Bai, and J. Lee, "Real-world super-resolution using generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 436–437.

[14] J. Cai, H. Zeng, H. Yong, Z. Cao, and L. Zhang, "Toward Real-World Single Image Super-Resolution: A New Benchmark and a New Model," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 3086–3095. doi: 10.1109/ICCV.2019.00318.

[15] L. Wang *et al.*, "Unsupervised Degradation Representation Learning for Blind Super-Resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10581–10590.

[16] S. Son, J. Kim, W.-S. Lai, M.-H. Yang, and K. M. Lee, "Toward Real-World Super-Resolution via Adaptive Downsampling Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021, doi: 10.1109/TPAMI.2021.3106790.

[17] I. Goodfellow *et al.*, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[18] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, PMLR, 2020, pp. 1597–1607.

[19] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[20] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[21] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

[22] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[23] 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)*, 2018, pp. 286–301.

[24] B. Niu *et al.*, "Single image super-resolution via a holistic attention network," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, Springer, 2020, pp. 191–207.

[25] Y. Yan, W. Ren, X. Hu, K. Li, H. Shen, and X. Cao, "SRGAT: Single Image Super-Resolution With Graph Attention Network," *IEEE Transactions on Image Processing*, vol. 30, pp. 4905–4918, 2021, doi: 10.1109/TIP.2021.3077135.

[26] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5728–5739. Accessed: Sep. 28, 2023. [Online]. Available: http://openaccess.thecvf.com/content/CVPR2022/html/Zamir_Restormer_Efficient_Transformer_for_High-Resolution_Image_Restoration_CVPR_2022_paper.html

[27] H. A. Aly and E. Dubois, "Specification of the observation model for regularized image up-sampling," *IEEE Transactions on Image Processing*, vol. 14, no. 5, pp. 567–576, May 2005, doi: 10.1109/TIP.2005.846019.

[28] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin, "Accurate Blur Models vs. Image Priors in Single Image Super-resolution," in *2013 IEEE International Conference on Computer Vision*, Sydney, Australia: IEEE, Dec. 2013, pp. 2832–2839. doi: 10.1109/ICCV.2013.352.

[29] W. Shi *et al.*, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 1874–1883. doi: 10.1109/CVPR.2016.207.

[30] Y.-S. Xu, S.-Y. R. Tseng, Y. Tseng, H.-K. Kuo, and Y.-M. Tsai, "Unified dynamic convolutional network for super-resolution with variational degradations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12496–12505.

[31] P. Wei *et al.*, "Component Divide-and-Conquer for Real-World Image Super-Resolution," in *Computer Vision – ECCV 2020*,

vol. 12353, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., in Lecture Notes in Computer Science, vol. 12353. , Cham: Springer International Publishing, 2020, pp. 101–117. doi: 10.1007/978-3-030-58598-3_7.

[32] H. R. V. Joze *et al.*, "Imagepairs: Realistic super resolution dataset via beam splitter camera rig," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 518–519.

[33] P. DiMeo, L. Sun, and X. Du, "Fast and Accurate Autofocus Control Using Gaussian Standard Deviation and Gradient-based Binning," *Optics Express*, vol. Revision, 2021.

[34] T. Kohler, M. Batz, F. Naderi, A. Kaup, A. Maier, and C. Riess, "Toward Bridging the Simulated-to-Real Gap: Benchmarking Super-Resolution on Real Data," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2019, doi: 10.1109/TPAMI.2019.2917037.

[35] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[36] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, Springer, 2006, pp. 404–417.

[37] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching With Graph Neural Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4938–4947. Accessed: Mar. 24, 2022. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Sarlin_SuperGlue_Learning_Feature_Matching_With_Graph_Neural_Networks_CVPR_2020_paper.html

[38] R. Ma and X. Du, "Closed-loop feedback registration for consecutive images of moving flexible targets," *Appl Intell*, Aug. 2022, doi: 10.1007/s10489-022-04068-0.

[39] Y. You, I. Gitman, and B. Ginsburg, "Large batch training of convolutional networks," *arXiv preprint arXiv:1708.03888*, 2017.

[40] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 126–135.

[41] J. Cai, S. Gu, R. Timofte, and L. Zhang, "NTIRE 2019 Challenge on Real Image Super-Resolution: Methods and Results," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0. Accessed: Dec. 24, 2020. [Online]. Available: https://openaccess.thecvf.com/content_CVPRW_2019/html/NTIRE/Cai_NTIRE_2019_Challenge_on_Real_Image_Super-Resolution_Methods_and_Results_CVPRW_2019_paper.html

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

[43] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding," in *British Machine Vision Conference (BMVC)*, 2012.

[44] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*, Springer, 2010, pp. 711–730.

[45] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5197–5206.

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

[47] MMEditing Contributors, "MMEditing: OpenMMLab Image and Video Editing Toolbox." 2022. [Online]. Available: https://github.com/open-mmlab/mmediting

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[49] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[50] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2472–2481.

[51] X. Lin *et al.*, "DiffBIR: Towards Blind Image Restoration with Generative Diffusion Prior." arXiv, Aug. 29, 2023. doi: 10.48550/arXiv.2308.15070.

[52] X. Chen, X. Wang, J. Zhou, Y. Qiao, and C. Dong, "Activating More Pixels in Image Super-Resolution Transformer," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, Jun. 2023, pp. 22367–22377. doi: 10.1109/CVPR52729.2023.02142.

[53] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image Restoration Using Swin Transformer," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada: IEEE, Oct. 2021, pp. 1833–1844. doi: 10.1109/ICCVW54120.2021.00210.