Journal of Biomedical Optics

BiomedicalOptics.SPIEDigitalLibrary.org

Super-resolution recurrent convolutional neural networks for learning with multi-resolution whole slide images

Lopamudra Mukherjee
Huu Dat Bui
Adib Keikhosravi
Agnes Loeffler
Kevin W. Eliceiri

SPIE.

Lopamudra Mukherjee, Huu Dat Bui, Adib Keikhosravi, Agnes Loeffler, Kevin W. Eliceiri, "Super-resolution recurrent convolutional neural networks for learning with multi-resolution whole slide images," *J. Biomed. Opt.* **24**(12), 126003 (2019), doi: 10.1117/1.JBO.24.12.126003.

Super-resolution recurrent convolutional neural networks for learning with multi-resolution whole slide images

Lopamudra Mukherjee,[a,*] Huu Dat Bui,[a] Adib Keikhosravi,[b] Agnes Loeffler,[c] and Kevin W. Eliceiri[b,d]
[a]University of Wisconsin–Whitewater, Department of Computer Science, Whitewater, Wisconsin, United States
[b]University of Wisconsin–Madison, Department of Biomedical Engineering, Madison, Wisconsin, United States
[c]MetroHealth Medical Center, Department of Pathology, Cleveland, Ohio, United States
[d]Morgridge Institute for Research, Madison, Wisconsin, United States

Abstract. We study a problem scenario of super-resolution (SR) algorithms in the context of whole slide imaging (WSI), a popular imaging modality in digital pathology. Instead of just one pair of high- and low-resolution images, which is typically the setup in which SR algorithms are designed, we are given multiple intermediate resolutions of the same image as well. The question remains how to best utilize such data to make the transformation learning problem inherent to SR more tractable and address the unique challenges that arises in this biomedical application. We propose a recurrent convolutional neural network model, to generate SR images from such multi-resolution WSI datasets. Specifically, we show that having such intermediate resolutions is highly effective in making the learning problem easily trainable and address large resolution difference in the low and high-resolution images common in WSI, even without the availability of a large size training data. Experimental results show state-of-the-art performance on three WSI histopathology cancer datasets, across a number of metrics.
[DOI: 10.1117/1.JBO.24.12.126003]

Keywords: image super-resolution; convolutional neural networks; pathology; whole slide imaging; machine learning.

Paper 190318R received Sep. 17, 2019; accepted for publication Nov. 20, 2019; published online Dec. 13, 2019.

1 Introduction

Many computational problems in medical imaging can be posed as a transformation learning problem, in that they receive some input image and transform it into an output image under domain-specific constraints. The image super-resolution (SR) problem is a typical problem in this category, where the goal is to reconstruct a high-resolution (HR) image given only a low-resolution (LR), typically degraded, image as input. Such problems are challenging to solve due to their highly ill-posed and underconstrained nature, since a large number of solutions exist for any given LR image, and the problem is especially magnified when the resolution ratio between the HR and LR images is large. Until recently, convolutional neural networks (CNN) have been the main driving tool for SR in computer vision applications, since they have the ability to learn highly nonlinear complex functions that constitute the mapping from the LR to the HR image. Several recent results have shown state-of-the-art results for the SR problem.[1–3] Since such CNN frameworks involve a large number of parameters, empirical evidence has shown that the corresponding models need to be trained on large datasets to show reproducible accuracy and avoid overfitting. This is not a problem for most applications in computer vision, where datasets in order of millions or larger (e.g., ImageNet,[4] TinyImages,[5] to name a few) are readily available. But for other application domains, particularly microscopic or medical imaging, such large sample sizes are hard to acquire, given that each image dataset has to be acquired individually, with significant human involvement. In this paper, we study an important SR application in the context of digital pathology and discuss how the limitations inherent to CNN-based SR methods can be addressed effectively in this context. This is described next.

1.1 *Application Domain*

The type of imaging modality we are interested in is called whole slide imaging (WSI) or virtual microscopy. WSI is a recent innovation in the field of digital pathology in which digital slide scanners are used to create images of entire histologic sections. Traditionally, the use of the optical capabilities of a microscope to "focus" the lens on a small subsection of the slide (based on the field of view of the device) to review and evaluate the specimen is often carried out by a trained pathologist. This process may need to be repeated for other sections of the slide depending on the scientific/clinical question of interest, toward obtaining consistently well-focused digital slides. WSI essentially automates this procedure for the whole slide. The ability to do so, automatically for a large number of slides, ideally capturing as much information as the pathologist may have been manually able to glean from the histological specimen via a light microscope, offers an immensely powerful tool with many immediate applications in clinical practice, research, and education.[6–8] For instance, WSI makes it feasible to solicit subspecialty disease expertise regardless of the location of the pathologist, integration of patient medical records in their health portfolio, pooling data between research institutions, and reducing long-term storage costs of histological specimens. However, given the relatively recent advent of WSI technology, there are several barriers that still need to be overcome, before it is widely accepted in clinical practice. The chief among these are the fact

[*] Address all correspondence to Lopamudra Mukherjee, E-mail: mukherjl@uww.edu

that HR WSI scanners, which have been shown to match images obtained from light microscopy in terms of quality for diagnostic capability, are typically very expensive, even for LR usage. In addition, the size of the files produced also generates a bottleneck. Typically, a virtual slide acquired at HR is about 1600 to 2000 megapixels, which results in a file size of several gigabytes. Such files are typically much larger that image files used by other clinical imaging specialties such as radiology. If one has to transport, share, or upload multiple such files or 3D stack, it results in a consequential increase of storage capacity and network bandwidth. Notwithstanding these issues, WSI offers tremendous potential and numerous advantages for pathologists, which is why it is important to find a way to alleviate the existing issues with WSI, such that it can be widely applicable. One potential way to address these issues is to use images from low magnification slide scanners, which are widely available, easy to use, comparatively inexpensive, and can also quickly produce images with smaller storage requirements. However, such LR images can increase the chance of misdiagnosis and false treatment if used as the primary source by a pathologist. For example, cancer grading normally requires identifying tumor cells based on size and morphology assessments,[9] which can be easily distorted in low magnification images. If such images were indeed to be used for research, clinical, or educational purposes, we need a way to convert such LR data and produce an output that closely resembles images acquired by a top-of-the-line WSI scanner, without substantial increase in storage and computational requirements.

1.2 *Solution Strategies and Associated Challenges*

One way to address the above issues is to dynamically improve the quality and resolution of LR images to render them comparable in quality to those acquired from HR scanners, as and when needed by the end user. Such a proposed workflow would need a fraction of the time and can yield near real-time quantifiably accurate results at a fraction of the setup cost of a standard WSI system. The implementation of such a system would require an SR approach that works well for WSI images. But still, we find that there is no off-the-shelf deep network architecture that can be used for our application directly. The reasons for this are numerous. First, most existing methods have been trained on databases of natural images. However, the WSI images under consideration do not have the same characteristics as natural images, such as, textures, edges, and other high-level exemplars, which are often leveraged by the SR algorithms. Second, such deep learning models are often trained using large training datasets (usually consisting of synthetic/resized HR–LR pairs), in the order of millions. This does not directly translate to our application for two reasons: (a) large training datasets are typically harder to acquire since each image pertains to a unique acquisition that requires significant human involvement, and (b) in our case, the LR images are acquired from a different scanner and is not just a resized version of HR image. Third and perhaps the most important limitation is that existing deep SR methods typically only reconstruct successfully up to a resolution increase factor of 4, whereas in case of WSI, the resolution (from a low-cost scanner to an expensive HR scanner) can increase up to a factor of 10×, since there can be wide variance in resolution between an LR scanner (4×) to HR scanners, which typically scan at 20× or 40×. We discuss this issue in detail in the following paragraph.

1.3 *Our Contribution*

Existing CNN-based methods have shown limited performance in scenarios when the resolution difference is high. The reason for this is that the complexity of the transformation that morphs the LR image to the HR one increases greatly in such situations, which in turn manifests in the CNN models taking longer to converge, or learning a function that generalizes poorly to unseen data. A typical way to address this issue is to make the CNN models deeper, by adding more layers (>5 layers) or increasing the number of examples required to learn a task to a given degree of accuracy while still keeping the network shallow. Both these directions pose challenges for our application: (a) a deeper network is associated with far more parameters, increasing the computational and memory footprint, to the extent that model may not be applicable in a real-time setup and (b) increasing the number of samples the extent needed would be impractical, due to associated time and monetary costs.

Our approach to solving this problem draws upon the inherent nature of the SR problem. While it is hard to acquire a large training dataset in this scenario, it is much more time and cost efficient to obtain the WSI images at different resolutions by varying the focus of the scanner. In this paper, we study how such multi-resolution data can be used effectively to make the transformation learning problem more tractable. Suppose I_1 and I_h represent a particular LR and HR image pair. If I_h is a significant resolution ratio higher than I_1, learning their direct transformation function $f(I_1) = I_h$ can be challenging leading to a overparameterized system. But if we had access to some intermediate resolutions say $I_2, \dots I_{h-1}$ (with a smaller resolution change between consecutive images), it makes intuitive sense that transformation that converts an image of a given resolution into the closest HR image would be roughly the same across all the resolutions considered, if we assume that resolution changes vary smoothly across the sequence. Having more image pairs (I_{k-1}, I_k) for $k = 2 \dots h$ to train, it may be computationally easier to learn a smooth function $\hat{f}$, such that $\hat{f}(I_{k-1}) \approx I_k$ for all k. In this paper, we formalize this notion and develop a recurrent convolutional neural network (RCNN) [Note that the acronym RCNN is also used to refer to region-based CNNs,[10] but in the context of this paper, we use it to refer to recurrent convolutional neural network.] to learn from multi-resolution slide scanner images. Our main contributions are as follows. (1) We propose a new version of SR problem motivated from this problem, multi-resolution SR (MSR), where the aim is to learn the transformation function, given a sequence of resolutions, rather than simply the LR and HR images. To the best of our knowledge, this is new problem scenario for SR that has not been studied before. (2) We propose an RCNN model to solve the MSR problem. (3) We demonstrate using experimental results on three WSI cell lines that the MSR idea can indeed reduce the need for large sample sizes and still learn the transformation that generalizes to unseen data.

2 Related Work

We summarize the current literature on three main aspects, pertaining to our model in this section: (a) deep network models for SR, (b) recurrent neural networks, and (c) CNN architecture for small sample size training. We discuss them briefly next.

2.1 Deep Network Models for SR

Stacked collaborative local autoencoders are used[11] to construct the LR image layer by layer. Reference 12 suggested a method for SR based on an extension of the predictive convolutional sparse coding framework. A multiple layer CNN, similar to our model, inspired by sparse-coding methods, is proposed in Refs. 1, 2, and 13. Chen and Pock[14] proposed to use multistage trainable nonlinear reaction diffusion as an alternative to CNN where the weights and the nonlinearity are trainable. Wang et al.[15] trained a cascaded sparse coding network from end to end inspired by learning iterative shrinkage and thresholding algorithm[16] to fully exploit the natural sparsity of images. Recently, Ref. 17 proposed a method for automated texture synthesis in reconstructed images by using a perceptual loss focusing on creating realistic textures. Several recent ideas have involved reducing the training complexity of the learning models using approaches, such as Laplacian pyramids,[18] removing unnecessary components of CNN,[19] and addressing the mutual dependencies of LR and HR images using deep back-projection networks.[20] In addition, generative adversarial networks (GAN) have also been used for the problem of single image SR, these include Refs. 21–24. Other deep network-based models for image SR problem include Refs. 25–28. We also briefly review SR approaches for sequence data such as videos. Most of the existing deep learning-based video SR methods using motion information inherent in video to generate a single HR output frame from multiple LR input frames. Kappeler et al.[29] warp video frames from the preceding and subsequent LR frames onto the current one using the optical flow method and pass them through a CNN that produces the output frame. Caballero et al.[30] followed the same approach but replaced the optical flow model with a trainable motion compensation network. Huang et al.[31] used a bidirectional recurrent architecture for video SR with shallow networks but do not use any explicit motion compensation in their model. Other notable works include Refs. 32 and 33.

2.2 Recurrent Neural Networks

A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit temporal dynamic behavior for a time sequence. Unlike feedforward neural networks such as CNNs, the input and outputs are not considered independent of each other, rather such models recompute the same/similar function for each element in the sequence, with the intermediate and final output of subsequent elements in the network being dependent on the previous computations on elements occurring earlier in the sequence. RNNs have most frequently been used in applications for language modeling,[34] speech recognition,[35] and machine translation.[36] But it can be applied to many learning tasks applied to sequence data, for more details, see survey paper by Ref. 37.

2.3 CNN Architectures for Small Sample Size Training

In this regard, Erhan et al.[38] devised unsupervised pretraining of deep architecture and showed that such weights of the network generalize better than randomly initialized weights. Mishkin and Matas[39] have proposed layer-sequential unit-variance that utilizes the orthonormal matrices to initialize the weights of CNN layer. Andén and Mallat[40] proposed scattering transform network (ScatNet), which is a CNN-like architecture where predefined Morlet filter bank is used to extract features. Other notable architectures in this regard include PCANet,[41] LDANet,[42] kernel PCANet,[43] MLDANet,[44] DLANet,[45] to name a few.

2.4 Deep Network Models in Microscopy

Since the application domain of this paper is in microscopy, we briefly review related papers that have used deep networks in this area. Most similar to our work is Rivenson et al.,[46] who showed how to enhance the spatial resolution of LR optical microscopy images over a large field of view and depth of field. But unlike our model, this framework is meant for single-image SR, where the model is trained on pairs of HR and LR images and provided a single LR image at test time. The design of this model includes a single CNN architecture, though they show that feeding the output of the CNN, back to the input, can improve results further. Wang et al.[47] proposed a GAN-based approach for super-resolving confocal microscopy images to match the resolution acquired with a stimulated emission depletion microscopes. Grant-Jacob et al.[48] designed a neural lens model based on a network of neural networks, which is used to transform optical microscope images into a resolution comparable to a scanning electron microscope image. Sinha et al.[49] used deep networks to recover phase objects given their propagated intensity diffraction patterns. Other related methods that have used deep learning-based reconstruction approaches for various applications in microscopy include Nehme et al.,[50] Wu et al.,[51] and Nguyen et al.[52] A more detailed survey of deep learning methods in microscopy can be found in Ref. 53.

3 Main Model

Here, we discuss our main model for obtaining HR images from LR counterparts. First, we briefly outline the problem setting. Let H and L denote the HR and LR image sets, respectively. In addition, we use two more intermediate resolutions of all images, we call these sets I^1 and I^2, respectively. For notational ease, we refer to L as I^0 and H as I^3, respectively. These image sets can be ordered in terms of increasing resolution, that is, $I^0 \leq I^1 \leq I^2 \leq I^3$ w.r.t. to image size. For training/learning, we assume image to image correspondence among these four sets are known.

For any pair of images (I^j, I^{j+1}), we need to learn the transformation $\mathbf{f}^j$ that maps I^j to the corresponding higher resolution image I^{j+1}. This can be done using a CNN architecture, with a number of intermediate convolutional layers, which we discuss shortly. The CNN pipeline then needs to be replicated for each of the three pairs (I^0, I^1), (I^1, I^2), and (I^2, I^3). However, the main premise of this work is that each CNN subarchitecture can be informed by outputs of other CNN subarchitectures, since they are implementing similar functions. To do this, we propose an RCNN that uses three CNN subarchitectures to map each LR images to next HR one. These three CNN subnetworks are then interconnected in a recurrent manner. Furthermore, we impose that the CNN pipelines share similar weights, to ensure that function learned for each pair of images is roughly the same. We describe the details of our model next. First, we discuss the components of the CNN architecture in Sec. 1, followed by the motivation and design choices for the RCNN framework in Sec. 2.

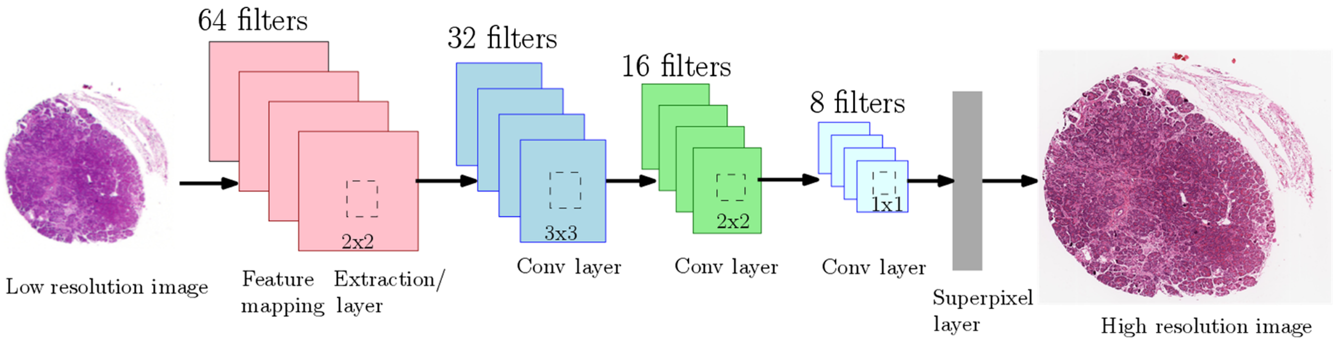

Fig. 1 Architecture of the proposed CNN for image super-resolution.

3.1 *CNN Subnetwork*

Here, we describe the basic structure of each CNN subnetwork and its constituent layers briefly. A more detailed description can be found in Sec. 7. Note that the components/layers of each CNN pipeline is kept the same, see Fig. 1. The first layer is a feature extraction-mapping layer that extracts features from LR images (denoted by Y_1^j for the j'th pipeline), which are then served as input to the next layers. This is followed by three convolutional layers. We briefly elaborate on this, since they are useful to understand the RCNN terminology in the next section.

3.2 *Convolutional Layers*

The feature extraction layer is followed by three additional convolutional layers. We also refer to these as hidden layers, denoted by H_i^j, which is the i'th hidden layer $(i \in \{2,3,4\})$ in the j'th CNN pipeline. The input to this layer is referred to as Y_{i-1}^j, and the output is denoted by Y_i^j. The filter functions in these intermediate layers can be written as

$$Y_i^j = \sigma(\theta_i^j \times Y_{i-1}^j + b_i^j) \quad i \in 2{,}3{,}4, \quad j \in 0,1,2,$$

where θ_i^j and b_i^j represent the weights and biases of each layer, respectively. Each of the weights θ_i^j is composed of n_i filters of size $n_{i-1} \times f_i \times f_i$. n_2 is set at 32 and $n_i = \frac{n_{i-1}}{2}$ for $i \in 3{,}4$. This progressive reduction in the number of filters leads to computational benefits as observed in numerical experiments. The filter sizes f_i are set to $\{3{,}2,1\}$ for each of the three layers, respectively, similar to hierarchical CNNs.

The last layer of the CNN architecture consists of a subpixel layer that upscales the LR feature maps to the size of the HR image.

3.3 *Recurrent Convolution Network*

The recurrent convolution network is built by interconnecting the hidden units of the CNN subarchitectures, see Fig. 2. We index the CNN pipeline components with superscript $j \in 0,1,2$ with the j'th pipeline being given image I^j as input and reconstructing the image I^{j+1}. The basic premise of our RCNN model is that the hidden units processing the each of the images can be informed by the outputs of the hidden units in other CNN pipelines. We use one directional dependence (low to high) as it is more challenging to reconstruct HR images compared to lower resolution ones. We can introduce bidirectional dependencies as well, but in practice, this increases the number of parameters substantially and contributes to an increase in training time for the model.

Besides the feedforward connections already discussed as a part of the CNN subarchitectures, we introduce two additional connections to encode the dependencies among the various hidden units, see Fig. 2. These are as follows.

3.3.1 *Recurrent convolutional connection*

The first type of connection, called recurrent convolutions, is denoted by red lines and aims to model the dependency across images of different resolutions at the same hidden layer i. These convolutions connect adjacent hidden layers of successive images (ordered by resolutions), that is, the current hidden layer H_i^j is conditioned on the feature maps learned from the hidden layer at the previous LR image H_i^{j-1}.

3.3.2 *Prelayer convolutional connections*

The second type of connections, called prelayer convolutions, is denoted by blue lines. This is used to model the dependency of a given hidden layer of the current image H_i^j on the hidden layer at the immediate previous layer corresponding to LR image H_{i-1}^{j-1}. This endows the hidden layer with not only the output of the previous layer but also information about how a lower resolution image has evolved in the previous layer.

Since the image sizes differ at each CNN pipeline, when implementing the dependence, we resize the higher-order images to match the size of images processed in the current pipeline. This resizing can be denoted by a function $\eta(.)$.

Note that the first CNN pipeline ($j = 0$), which processes I^0 as input, contains only feedforward connections, hence is identical to the network in Fig. 1. We incorporate the three types of connections (feedforward, recurrent, and prelayer) in the next two CNN pipelines ($j \in 1{,}2$). Let the output of hidden layer H_i^j be denoted by $\mathbf{Y}_i^j$. Then, we can rewrite functions learned and the outputs at the hidden layers of the CNN pipelines $j \in 1{,}2$ as follows. We start with the functions learned at first hidden layer ($i = 2$), which can be written as

$$\mathbf{Y}_2^j = \sigma[Y_1^j + \boldsymbol{\theta}_2^j \times \eta(\mathbf{Y}_2^{j-1}) + \hat{\boldsymbol{\theta}}_2^j \times \eta(Y_1^{j-1}) + \mathbf{b}_1^j] \quad j \in 1{,}2. \tag{1}$$

For the subsequent hidden layers ($i \in 3{,}4$), the function can be written as

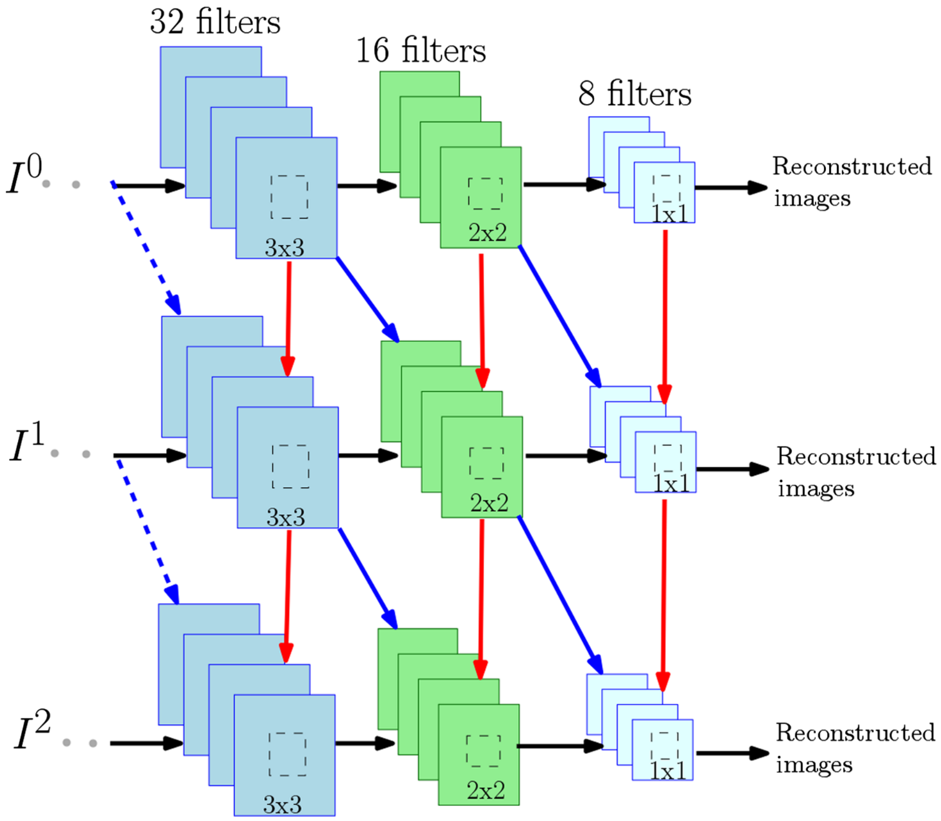

Fig. 2 Connections between hidden units of the three CNN subarchitectures: feedforward connections (in black), recurrent connections (in red), and prelayer connections (in blue).

$$\mathbf{Y}_i^j = \sigma[\mathbf{Y}_{i-1}^j + \boldsymbol{\theta}_i^j \times \eta(\mathbf{Y}_i^{j-1}) + \hat{\boldsymbol{\theta}}_i^j \times \eta(\mathbf{Y}_{i-1}^{j-1}) + \mathbf{b}_i^j]$$
$$j \in 1{,}2, \quad i \in 3{,}4. \tag{2}$$

The variables $\boldsymbol{\theta}_i^j$ and $\hat{\boldsymbol{\theta}}_i^j$ represent the weights of the recurrent and prelayer connections, respectively, whereas $\mathbf{b}_i^j$ represents the biases at the i'th layer of the j'th pipeline. Note that the $\eta(.)$ may be replaced by the subpixel layer, but this contributes to an increase in the training time. Therefore, we implemented the $\eta(.)$ as a simple bicubic interpolation.

3.4 *Training and Loss Function*

The complete architecture of our network can be seen in Fig. 3. The output from Eq. (2) (for pipelines $j = 1$ and $j = 2$) is then passed on as an input to the subpixel layer [described in Eq. (4)], which outputs the desired prediction (let this be denoted by R^j). For the pipeline ($j = 0$), the prediction is simply $R^0 = Y_5^0$. This network is learned by minimizing a weighted function of mean square error (MSE) and structured similarity metric (SSIM) between the predicted HR and the ground truth at each pipeline

$$O(H,R) = \sum_{j=0}^{2} \{\rho \mathrm{MSE}(I^{j+1}, R^j)$$
$$+ (\rho - 1)[1 - \mathrm{SSIM}(I^{j+1}, R^j)]\}, \tag{3}$$

where $\mathrm{MSE}(I^{j+1}, R^j) = ||I^{j+1} - R^j||^2$ and the structured similarity objective is defined as $\mathrm{SSIM}(I^{j+1}, R^j) = L(I^{j+1}, R^j)^\alpha C(I^{j+1}, R^j)^\beta S(I^{j+1}, R^j)^\gamma$, where $L(I^{j+1}, R^j)$ is the luminance-based comparison, $C(I^{j+1}, R^j)$ is a measure of contrast difference, and $S(I^{j+1}, R^j)$ is the measure of structural differences between the two images I^{j+1} and R^j. α, β, and γ are kept constant and ρ is set to .75.

The objective/loss function in Eq. (3) is optimized using stochastic gradient descent. During optimization, all the filter weights of recurrent and prelayer convolutions are initialized by randomly sampling from a Gaussian distribution with mean 0 and standard deviation 0.001, whereas the filter weights of feedforward convolution are set to 0. Note that one can also initialize these weights by pretraining the CNN pipeline on a small sized dataset. We experimentally find that using a smaller learning rate ($1e - 5$) for the weights of the filters is crucial to obtain good performance.

4 Experiments

4.1 *Datasets*

We performed experiments to evaluate our RCNN approach on three previously published tissue microarray (TMA) cancer datasets, a breast TMA dataset consisting of 60 images,[54] and a kidney TMA dataset with 381 images,[55] and a pancreatic TMA dataset with 180 images.[56] The datasets are summarized in Table 1.

4.2 *Imaging Systems*

In the datasets we analyze, highest resolution images were acquired and digitalized at 40× using an Aperio CS2 digital pathology scanner (Leica Biosystems),[57] with 4 pixels/μm, and

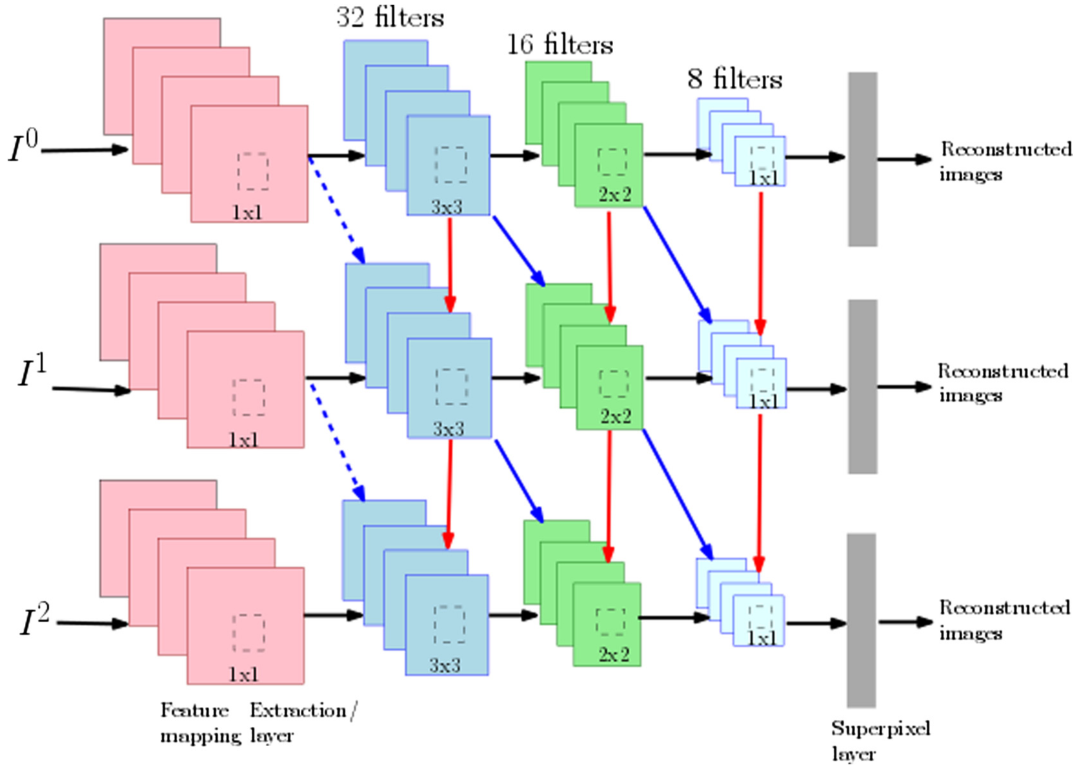

Fig. 3 Architecture of our proposed RCNN for image super-resolution.

Table 1 Summary of datasets.

Dataset	Number of images	Source
Breast	60	53
Kidney	381	54
Pancreas	180	55

lowest resolution images were acquired and digitized using PathScan Enabler IV[58] with 0.29 pixels/μm (4×). Besides these, we have acquired images at two different resolutions (10× and 20× from the Aperio CS2 digital pathology scanner, which served as our intermediate resolutions). These images are then resized (in the order of resolution) to 256×256, 512×512, 1024×1024, and 2048×2048, which provides us the dataset with four different resolutions.

4.3 *Evaluations*

We evaluate various aspects of our RCNN model to determine the efficacy of our method. First, we evaluate the quality of reconstruction of our RCNN model compared to a single CNN pipeline (which is a baseline for our model) and other comparable methods for SR. We show both qualitative and quantitative results in this regard. Second, we study the advantage of having intermediate resolutions, by varying the number of resolutions available. We also analyze how useful our obtained reconstructions are toward end-user applications, such as segmentation and perform a user study, done by a pathologist to evaluate the utility of the reconstructed images for cancer grading. In addition, we study how the reconstruction accuracy is affected as a function of the training set size. Finally, we discuss the parameters used and the computational time requirements of our model. We discuss these issues next.

4.3.1 *Quality of reconstruction*

Metrics. We evaluate the reconstruction quality of the obtained images by our approach by evaluating it relative to HR ground truth image and calculating seven different metrics: (1) root mean square error (RMSE), (2) signal-to-noise ratio (SNR), (3) SSIM, and (4) mutual information (MI), (5) multi-scale structured similarity (MSSIM), (6) noise quality measure (NQM),[59] and (7) weighted peak signal-to-noise ratio (WSNR).[60] RMSE should be as low as possible, whereas SNR, SSIM (1 being the maximum), MSSIM (1 being the maximum), and the remaining metrics, should be high for good reconstruction.

Experimental setup. Note that our model was trained by providing three sets of images (of three different resolutions) as input. However, our testing experiments can be done in the following two settings: (a) in the first case, we provide the images of the same three resolutions I^0, I^1, and I^2 as input to the trained model, which then outputs the reconstructed highest resolution image I^3. We call this setup RCNN(full). (b) In the second case, we see how our model behaves given only the lowest resolution image I^0. In this case, we first generate the two intermediate resolutions as follows. First, we only activate pipeline $j = 0$, which outputs I^1. Then, use this as input and activate both pipelines $j = 0$ and $j = 1$, which then reconstructs I^2 as well. Using I^0 and the reconstructed I_1 and I_2 as input, we activate all three pipelines and reconstruct I_3. We call this setup RCNN(1 input).

Table 2 Quantitative results from reconstructed breast images.

			Breast TMA			
Metric	SRGAN	ESCNN	FSRCNN	CNN	RCNN (full)	RCNN (1 input)
RMSE	48.12	42.86	46.64	39.57	**15.64**	31.27
SNR	14.63	15.37	14.95	16.37	**24.36**	18.37
SSIM	0.40	0.35	0.42	0.34	**0.98**	0.51
MI	0.05	0.09	0.01	0.08	**0.31**	0.36
MSSIM	0.42	0.39	0.19	0.38	**0.95**	0.53
NQM	0.37	0.39	0.28	1.09	**20.33**	2.48
WSNR	13.83	14.61	14.78	15.77	**26.59**	18.04

Note: Best values are highlighted in bold.

Comparable methods. To the best of our knowledge, we do not know of any other neural network framework to study the MSR problem presented in this paper. Therefore, to compare our method to other state-of-art methods, we choose other SR approaches that work in the two resolution setting (low and high). Our default baseline is the CNN architecture shown in Fig. 1. We refer to this method as CNN. In addition, we also compare with the following methods: (i) the CNN-based framework (FSRCNN) by Dong et al.,[13] (ii) a CNN model that uses a subpixel layer (ESCNN), and (iii) a GAN-based approach for SR.[21]

Results. Results for the breast, kidney, and pancreatic datasets are shown in Tables 2–4, respectively. In each case, we see that the RCNN(full) setting outperforms all other methods, giving a significant improvement in all the metrics calculated. A qualitative analysis of the reconstructed images in Figs. 4–6 shows that the reconstructed images are indeed very similar to the HR images. The comparatively poorer performances of comparable methods, such as FSRCNN or SRGAN, are expected since these methods are not designed to learn a resolution ratio of 8 used in our experiments. Still we find that RCNN(1 input), which is trained on all three input resolutions but tested by providing the lowest resolution only, outperforms the other baselines, showing that the weights learned our model generalizes better than other neural network models for this difficult learning task. The qualitative results showing the performance of RCNN(1 input) is shown in Fig. 7.

Table 3 Quantitative results from reconstructed kidney images.

			Kidney TMA			
Metric	SRGAN	ESCNN	FSRCNN	CNN	RCNN (full)	RCNN (1 input)
RMSE	28.75	25.48	38.90	29.15	**11.60**	22.92
SNR	19.00	20.06	16.35	18.96	**28.31**	21.03
SSIM	0.82	0.72	0.39	0.76	**0.98**	0.85
MI	0.11	0.16	0.07	0.13	**0.35**	0.31
MSSIM	0.70	0.70	0.41	0.68	**0.97**	0.78
NQM	7.77	4.61	0.45	6.80	**11.15**	10.30
WSNR	20.55	20.34	15.71	19.58	19.75	**21.89**

Note: Best values are highlighted in bold.

Table 4 Quantitative results from reconstructed pancreatic images.

			Pancreas TMA			
Metric	SRGAN	ESCNN	FSRCNN	CNN	RCNN (full)	RCNN (1 input)
RMSE	84.59	37.30	39.56	35.39	**20.32**	33.50
SNR	10.0	16.78	16.26	17.26	**22.07**	17.78
SSIM	0.39	0.52	0.42	0.64	**0.96**	0.79
MI	0.07	0.16	0.16	0.16	**0.29**	0.33
MSSIM	0.39	0.50	0.42	0.56	**0.93**	0.69
NQM	0.14	7.10	5.95	10.24	**16.94**	10.97
WSNR	7.93	16.27	15.57	16.99	**24.79**	17.88

Note: Best values are highlighted in bold.

4.3.2 *Effect of the number of intermediate resolutions*

Here we study the effect of having intermediate resolution images toward the quality of reconstruction. For this purpose, besides the RCNN(full), which uses two intermediate resolution, we also trained a model with only one intermediate resolution I^2, besides the LR and HR images I^0 and I^3. That is, the RCNN models have only two pipelines with inputs I^0 and I^2, respectively. We call this model RCNN(1 layer). We train and evaluate our model on each of the three datasets, see Table 5. The results show RCNN(full) shows superior performance compared to RCNN(1 layer), showing that each additional intermediate resolution adds toward the quality of the reconstructions produced.

4.3.3 *Segmentation results*

Pathological diagnosis largely depends on nuclei localization and shape analysis. We used a simple color segmentation method to segment the nuclei using K-means clustering to segment the image into four different classes based on pixel values in Lab color space.[61] Following this, we use the Hadamard product of each class with the gray level image of the original brightfield image, computed average of pixel intensities in each class, and assigned the lowest value to the cell nuclei. To evaluate our results, we compare the segmentation of the reconstructed images with the results from HR images (ground truth) for 20 samples from each group by computing the misclassification error, which calculates the percentage of pixels misclassified. Results show that number of pixels misclassified from images generated using our RCNN(full) method generates segmentation masks with lower number of pixels misclassified, followed by RCNN(1 input) (Table 6).

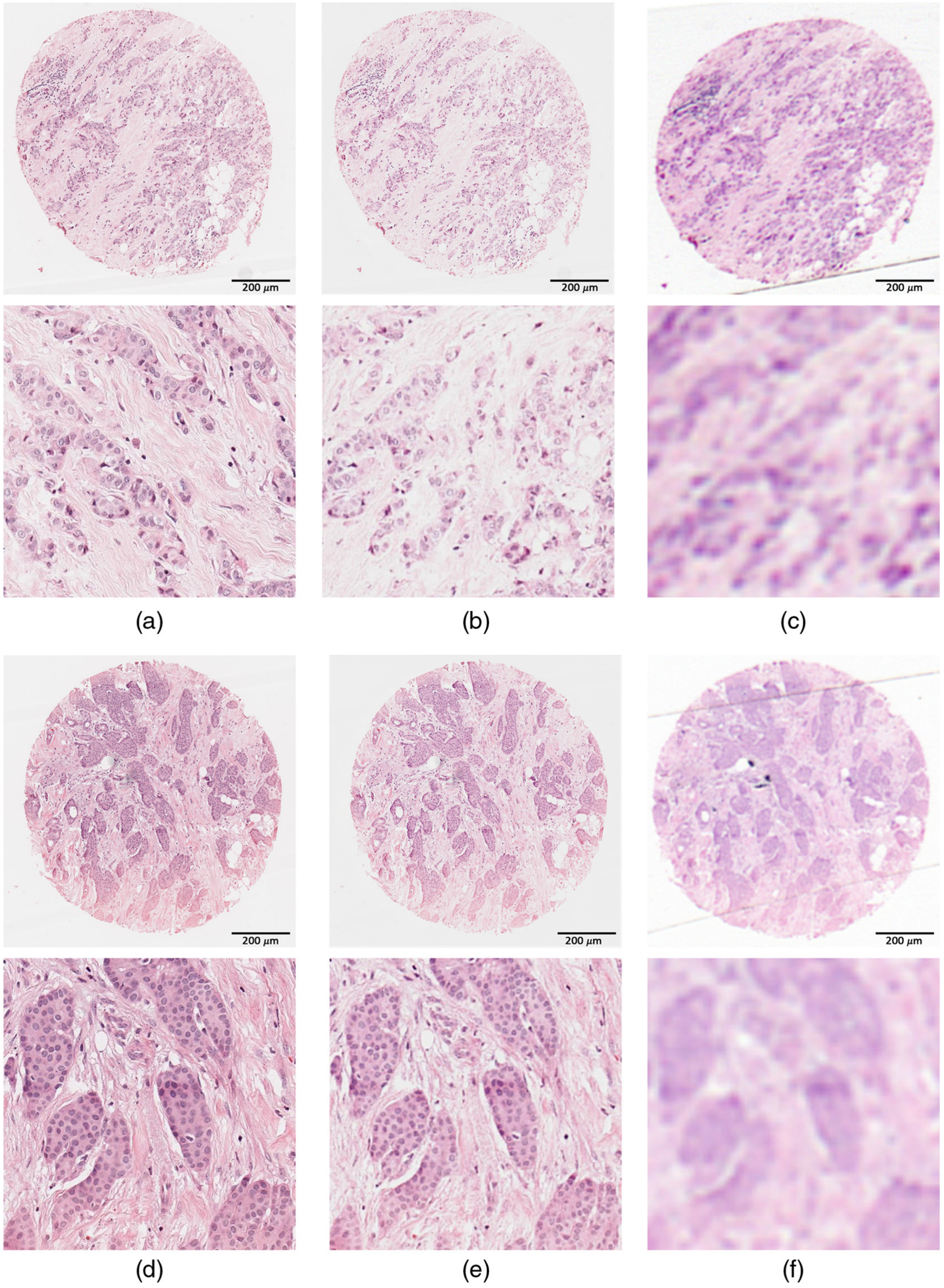

Fig. 4 Results of reconstruction of breast cancer TMA: columns 1 and 3 show HR and LR images and column 2 shows the reconstructed image. Rows 2 and 4 show a zoomed in region of interest from the corresponding images in row 1 and row 3, respectively.

4.3.4 *Grading user study by pathologists*

Pathological assessment of tissue samples is usually considered the gold standard that requires large magnification for microscopic assessment or HR images. For example, in different types of cancer patient prognosis and treatment plans are predicated on cancer grade and stage.[62] Tumor grade is based on pathologic (microscopic) examination of tissue samples, conducted by trained pathologists. Specifically, it involves assessment of the degree of malignant epithelial differentiation, or percentage of gland-forming epithelial elements, and does not take into consideration characteristics of the stroma surrounding the cells.[63–67] Accuracy of pathological assessment has a vital importance in clinical workflow since the downstream treatment plans mainly rely on that. Lack of interobserver and intraobserver agreement

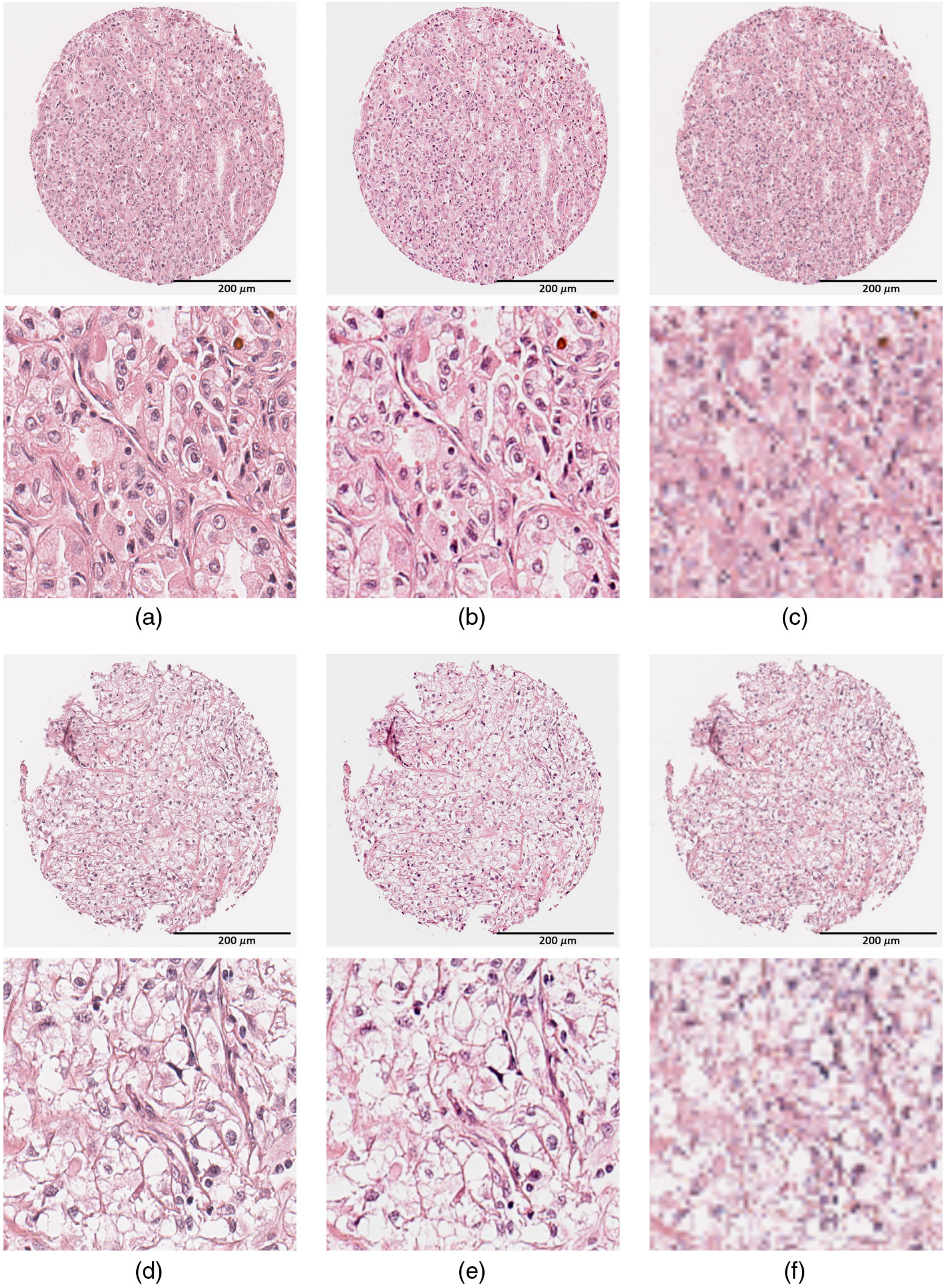

Fig. 5 Results of reconstruction of kidney cancer TMA: columns 1 and 3 show HR and LR images and column 2 shows the reconstructed image. Rows 2 and 4 show a zoomed in region of interest from the corresponding images in row 1 and row 3, respectively.

in pathological tissue assessments is still of major concern, which has been reported for many diseases including pancreatic cancer,[68] intraductal carcinoma of the breast,[69] malignant non-Hodgkin's lymphoma,[70] and soft tissue sarcomas,[71] among others. SR algorithms can mitigate this effect by making second opinion and collaborative diagnosis easily accessible. However, this is achievable if able to reconstruct fine morphological details of the tissue image. For this project, we used reconstructed images of 35 TMA cores randomly selected from a TMA slide (PA 2072, US Biomax), which was graded on HR images more than a year ago and was now graded on the reconstructed images by our collaborator pathologist. These were from different grades of cancer and normal tissue. Grading for 22 TMA cores matched the previous grading by same

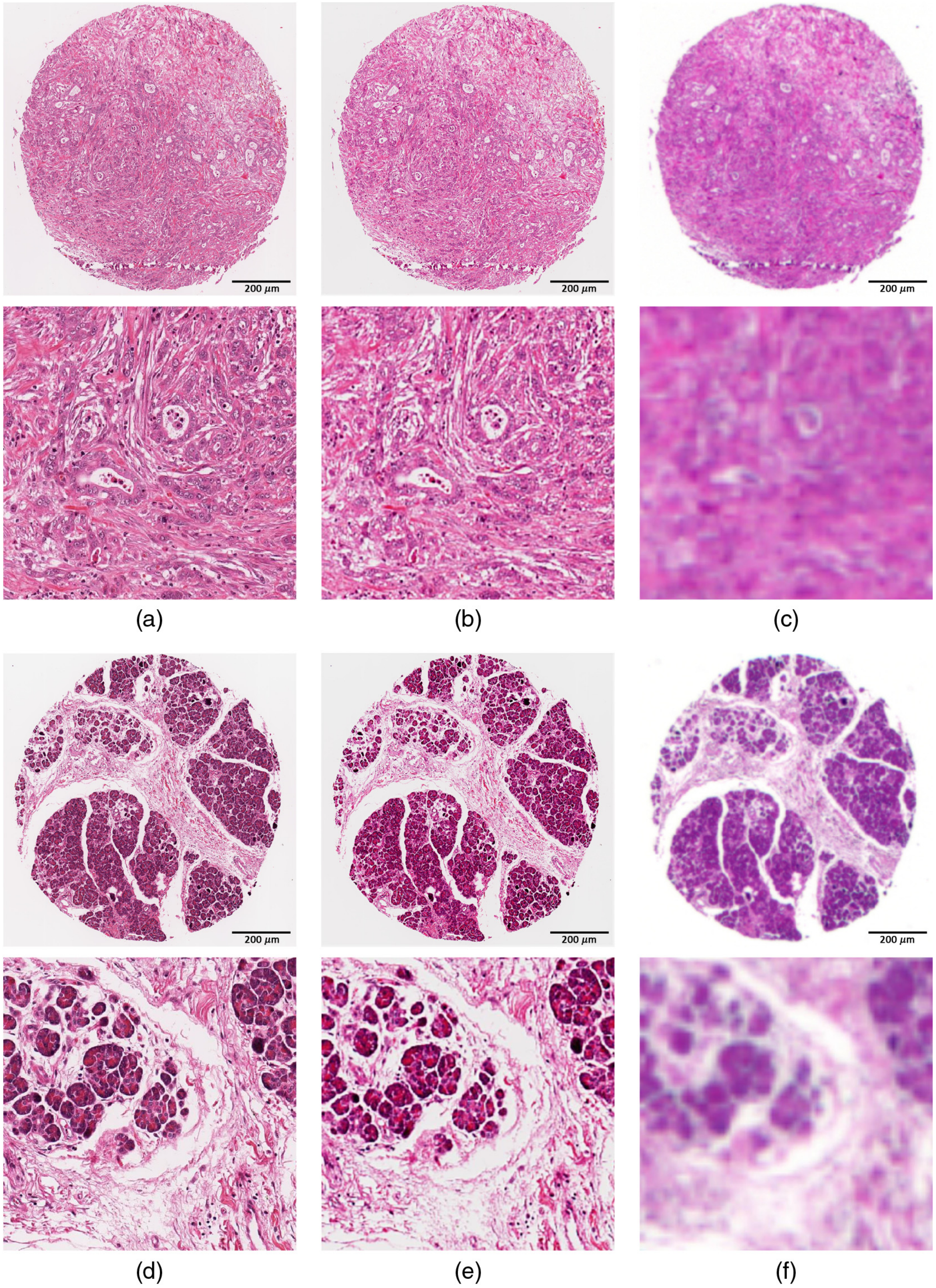

Fig. 6 Results of reconstruction of pancreatic cancer TMA: columns 1 and 3 show HR and LR images and column 2 shows the reconstructed image. Rows 2 and 4 show a zoomed in region of interest from the corresponding images in row 1 and row 3, respectively.

pathologist. In general, the overall structure of the pancreatic tissue was reconstructed good enough so that normal and grade one was easier to identify based on overall gland shapes and in case of grade one cancer the stroma surrounding the gland was identifiable too. However, it was observed that differentiation between grade 2 and 3 was more difficult since it requires visualization of infiltrating individual tumor cells. Grading results specific to individual cores is provided in Sec. 8.

4.3.5 *Effect of training set size*

The necessary size of the training set for a particular learning task is generally hard to estimate and depends mostly on the hardness of the learning problem as well as on the type of model being trained. Here, we provide empirical evidence of how our model behaves wrt to increasing the size of the training set. For this purpose, we use the Kidney dataset, since it is the largest

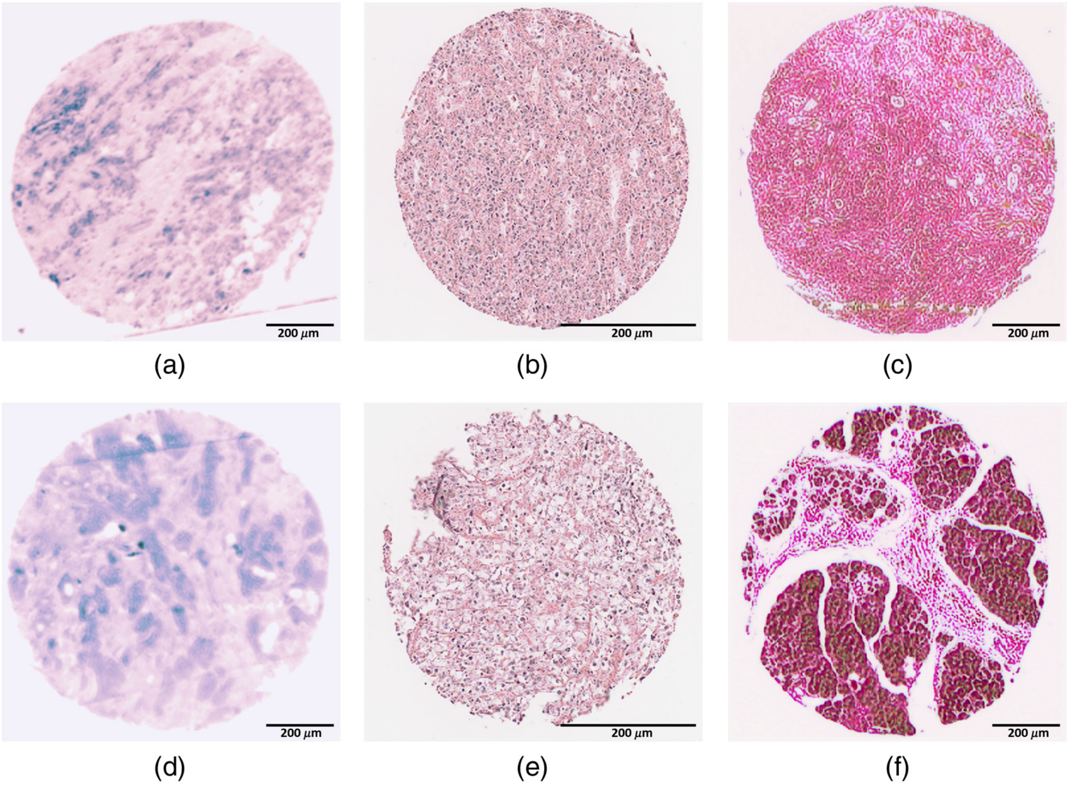

Fig. 7 Results of reconstruction of all three cell types using RCNN(1 input): column 1 shows breast cells; column 2 shows kidney cells; and column 3 shows pancreatic cells. The HR images for the breast cells [images (a) and (d) in this figure] are shown in Figs. 4(a) and 4(d) and the corresponding LR images are in Figs. 4(c) and 4(f), respectively. Similarly for the kidney [images (b) and (c) of this figure], the HR images are in Figs. 5(a) and 5(d) and LR images are in 5(c) and 5(f), respectively. Finally, for the pancreatic cells [(c) and (f) in this figure], the HR images are in Figs. 6(a) and 6(d) and LRn images are in 6(c) and 6(f), respectively.

Table 5 Quantitative results from varying the number of intermediate resolutions.

	Breast TMA		Kidney TMA		Pancreas TMA	
Metric	RCNN (1 layer)	RCNN (full)	RCNN (1 layer)	RCNN (full)	RCNN (1 layer)	RCNN (full)
RMSE	18.71	15.64	16.76	11.60	25.02	20.32
SNR	22.85	24.36	22.85	28.31	20.28	22.07
SSIM	0.85	0.98	0.95	0.98	0.94	0.96
MI	0.27	0.31	0.28	0.35	0.25	0.29
MSSIM	0.936	0.95	0.89	0.97	0.84	0.93
NQM	6.34	20.33	9.87	11.15	14.0	16.94
WSNR	24.91	26.59	25.63	30.60	22.18	24.79

dataset considered in this paper. We vary the training size from $\{50,100,150,200,300\}$. The test set is set to 50 in each case. We analyze the quality of reconstruction by comparing the SNR in each case.

As seen in Fig. 8, the SNR improves only slightly when the training set is increased, indicating that a higher training size does not significantly improve the image reconstruction metrics.

Table 6 Quantitative results from segmentation on the three datasets.

Misclass error	Breast	Kidney	Pancreas
RCNN (full)	0.1417	0.1758	0.1586
RCNN (1 input)	0.2358	0.1803	0.1630
CNN	0.2371	0.1908	0.1851

4.3.6 *Parameters and running time*

We implemented our model in TensorFlow using Python, which has inbuilt GPU utilization capability. We used a workstation with an AMD processor 6378 with a 2.4 GHz CPU, 48 GB RAM and NVIDIA GPU 1070 TI graphics card. All our experiments have been performed using GPU, which shows significant performance gains compared to CPU runtime. The training time of our models depends on various factors such as dataset volume, learning rate, batch size and number of training epochs. To report running times for training, we fix learning rate to 10^{-5}, dataset volume to 300 images, batch size to 2 and number of training epochs to 10^5. The training time of our model is approximately 20.9 hours. The time to generate a new HR image at 2048×2048 once the network is trained takes 1.4 minutes. The test-time speed of our model can be further accelerated by approximating or simplifying the trained networks with possible slight degradation in performance.

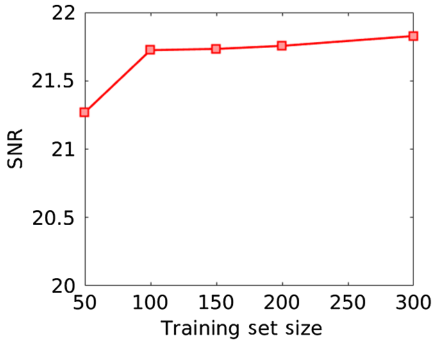

Fig. 8 SNR as a function of training size.

5 Future Directions

This paper provides an innovative approach to utilize multi-resolution images to generate a high quality reconstruction of slide scanner images. In addition, this work also leads to several interesting ideas that we will pursue as future work. We discuss these briefly next.

1. We will study a variation of this model that is inspired by mixed effects models popular in statistics. Here the fixed effects will be modeled as function of other higher resolution inputs and the random effects are modelled as residual connection on the LR input in each pipeline. This leads to a Residual variation of the Recurrent Convolutional Network. We will study this network in detail and analyze its computational properties.
2. One of our future goal is also aimed at making our RCNN model scalable to large datasets and higher resolution ratios. In order to do so, we need a way of speeding up the RCNN model to produce HR images at a reduced computational and memory footprint. To do this, we will adopt recent developments in deep learning that show that one can substantially improve the running time of deep CNNs by approximating by linear filters and other related ideas.[72]

6 Conclusion

This paper studies a new setup for the SR problem, where the input is multi-resolution slide scanner images, and the goal is to utilize these to make the learning problem associated with SR more tractable, so that it scales to an HR change in the target reconstruction. We propose a RCNN for this purpose. Results show that this model performs favorably when compared to state-of-the-art approaches for SR. This provides a key contribution toward the overarching goal of making use of LR scanners over their HR counterparts, which opens up new opportunities in histopathology research and clinical application.

7 Appendix A: CNN Subnetwork

Here, we describe the basic structure of each CNN subnetwork and its constituent layers. The components/layers of each CNN pipeline are kept the same. The basic CNN architecture is similar to the model described in our earlier paper,[73] except it does not involve the nearest-neighbor-based enhancement, which was used to ensure that reconstructed image retains the finer details of the original HR image, which are otherwise lost using a CNN framework for learning the transformation. We avoid this step since it is also computationally expensive to search a large database of patches for nearest neighbors, especially for the output sizes of HR images at 2048×2048 we want to reconstruct. We also replace the ReLU function at the output of each convolutional layer with a leaky ReLU, which is known to have better performance in practice. We describe the layer of the CNN architecture next, see Fig. 1.

7.1 *Feature Extraction-Mapping Layer*

This layer is used as the first step to extract features from the LR input images (I^j) for the j'th CNN subarchitecture. The feature extraction process can be framed as a convolution operation and hence implemented as a single layer of the CNN. This can be expressed as

$$\hat{Y}_1^j = \sigma(\theta_1^j \times I^j + b_1^j) \quad j \in 0,1,2,$$

where I^j is the image of a given resolution and θ_1^j and b_1^j represent the weights and biases of the first layer of this CNN pipeline, respectively. The weights are composed of $n_1 = 64$ convolutions on each image patch, with each convolution filter being of size 2×2. Therefore, this layer has 64 filters, each of size 2×2. The bias vector is of size $b_i^j \in R^{n_1}$. We keep filter sizes small at this level, so as it extracts more fine grained features from each patch. The $\sigma(x)$ function implements a leaky ReLU function, which can be written as $\sigma(x) = \mathbb{1}(x < 0)(\alpha x) + \mathbb{1}(x >= 0)(x)$, where α is a small constant. This is followed by a sum pooling layer, to obtain a weighted sum pool of features across various feature-maps of the previous layer. The output of this layer is referred to as Y_1^j.

7.2 *Convolutional Layers*

The feature extraction layer is followed by three additional convolutional layers. We also refer to these as hidden layers, denoted by H_i^j, which is the i'th hidden layer ($i \in \{2,3,4\}$) in the j'th CNN pipeline. The input to this layer is referred to as Y_{i-1}^j and the output is denoted by Y_i^j. The filter functions in these intermediate layers can be written as

$$Y_i^j = \sigma(\theta_i^j \times Y_{i-1}^j + b_i^j) \quad i \in 2,3,4, \quad j \in 0,1,2,$$

where θ_i^j and b_i^j represent the weights and biases of each layer, respectively. Each of the weights θ_i^j is composed of n_i filters of size $n_{i-1} \times f_i \times f_i$. n_2 is set at 32 and $n_i = \frac{n_{i-1}}{2}$ for $i \in 3,4$. This progressive reduction in the number of filters leads to computational benefits as observed in numerical experiments. The filter sizes f_i are set to $\{3,2,1\}$ for each of the three layers, respectively, similar to hierarchical CNNs.

7.3 *Subpixel Layer*

In our CNN architecture, we leave the final upscaling of the learned LR feature maps to match the size of the HR image, to be done at the last layer of the CNN. This is implemented as a subpixel layer similar to Ref. 74. The advantage of this is that is that layers prior to the last layer operate on the reduced

Table 7 Results of grading done by pathologist on individual cells.

Cell number	Grading on reconstructed images	Grading on LR images
C13	Pathologist identified this as higher grade cancer, based on condensations that are likely strips of malignant epithelium infiltrating the stroma. She could not see the epithelial cell profiles well enough to judge whether it is G2 or G3.	There were some irregular gland-like spaces at 11:00 and 4:00. Pathologist suspected that there are higher-grade malignant cells infiltrating through the stroma, but could not resolve what is a vessel, stromal cell, inflammatory cell, or malignant cell.
D5	This picture gave better definition of high-grade cancer infiltrating the stroma, but pathologist still could not call G2 from G3	According to pathologist, gland-like spaces were present near the center of the image and at 1 to 2:00. She could not resolve the dark spots in the stroma, whether these strips of high-grade cancer, inflammatory cells, or capillaries.
D6	The top of the image contained G1 glands, and there was a large complex conglomeration near the center of the field that the pathologist called G2. Toward the bottom of the field, dispersed nuclei was concerning for isolated cells or clusters of cells (G3).	The gland outlines were irregular enough that pathologist called this G2 cancer, but she could not tell if there was also G3 in the background. She did not want to make a diagnosis of cancer off this slide, since the image does not show any nuclear or architectural detail (proximity of the glands to nerves, arteries, and remnant acinar tissue).
D7	A cluster of G2 was present in the bottom half of the field. The pathologist thought the top was necrotic.	She also called this as G2 cancer, but could not tell if there is G3 cancer present, as well. There was not enough nucelar definition to be able to tell the degree of nuclear atypia.
D9	G3. The nuclei at the very middle of the field was bizarre enough to identify as high-grade carcinoma, and clusters of infiltrating glands contain enough of the same nuclei to identify the cells as epithelial (as opposed to lymphocytes sitting in stroma)	Also G2 cancer. Necrosis could be seen in very irregularly shaped glands. The stroma could not be resolved at all, there may have been single cells in the background but she could not see them.
E3 and E9	These cancers were heavily infiltrated by lymphocytes, making identification of the malignant glands extremely difficult, even on tissue sections. According to pathologist, they were probably G2 tumor.	Irregularly shaped glands were present throughout the core. The background could not be resolved.

LR image rather than HR size, which reduce the computational and memory complexity substantially.

The upscaling of the LR image to the size of the HR image is implemented as a convolution with a filter θ^j_{sub} whose stride is $\frac{1}{r}$ (r is the resolution ratio between the HR and LR images). The size of the filter is denoted as f_{sub}. A convolution with stride of $\frac{1}{r}$ in the LR space with a filter θ^j_{sub} (weight spacing $\frac{1}{r}$) would activate different parts of θ^j_{sub} for the convolution. The patterns are activated at periodic intervals of $\text{mod}(a,r)$ and $\text{mod}(b,r)$ where a, b are the pixel position in HR space. This can be implemented as a filter θ^j_5, whose size is $n_4 \times r^2 \times f_5 \times f_5$, given that $f_5 = \frac{f_{\text{sub}}}{r}$ and $\text{mod}(f_{\text{sub}}, r) = 0$. This can be written as

$$Y^j_5 = \gamma(\theta^j_5 \times Y^j_4 + b^j_5) \quad j \in 0,1,2, \tag{4}$$

where γ is periodic shuffling operator that rearranges r^2 channels of the output to the size of the HR image.

8 Appendix B: Grading of Cancer TMAs

Here, we provide the individual grading results on the subset of the pancreatic TMA cores graded by the pathologist. Note that these results are representative of the results on a typical reconstruction. For comparison purposes, our patholoist also graded the LR images of the same TMAs. The results can be seen in Table 7. The first column refers to the identifier for each cell. To summarize the results, the overall structure of the pancreatic tissue was reconstructed well enough so that normal and grade one was easier to identify based on overall gland shapes. Specifically, in case of grade one cancer, the stroma surrounding the gland was identifiable too. However, it was observed that differentiation between grade 2 and 3 was more difficult since it requires visualization of infiltrating individual tumor cells.

Disclosures

The authors have no relevant financial interests in this article and no potential conflicts of interest to disclose.

Acknowledgments

This research was supported by UW Laboratory for Optical and Computational Instrumentation, the Morgridge Institute for Research, UW Carbone Cancer Center, NIH R01CA199996, and Semiconductor Research Corporation.[75]

References

1. C. Dong et al., "Learning a deep convolutional network for image super-resolution," *Lect. Notes Comput. Sci.* **8692**, 184–199 (2014).
2. C. Dong et al., "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(2), 295–307 (2016).
3. S. Gu et al., "Convolutional sparse coding for image super-resolution," in *IEEE Int. Conf. Comput. Vision*, pp. 1823–1831 (2015).
4. J. Deng et al., "ImageNet: a large-scale hierarchical image database," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, IEEE, pp. 248–255 (2009).

5. A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: a large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(11), 1958–1970 (2008).
6. R. S. Weinstein et al., "An array microscope for ultrarapid virtual slide processing and telepathology. Design, fabrication, and validation study," *Hum. Pathol.* **35**(11), 1303–1314 (2004).
7. D. Wilbur et al., "Whole-slide imaging digital pathology as a platform for teleconsultation: a pilot study using paired subspecialist correlations," *Arch. Pathol. Lab. Med.* **133**(12), 1949–1953 (2009).
8. L. Pantanowitz, M. Hornish, and R. A. Goulart, "The impact of digital imaging in the field of cytopathology," *Cytojournal* **6**, 6 (2009).
9. National Cancer Institute, "Tumor grade," 2013, https://www.cancer.gov/about-cancer/diagnosis-staging/prognosis/tumor-grade-fact-sheet.
10. R. Girshick, "Fast {R-CNN}," in *Proc. Int. Conf. Computer Vision (ICCV)* (2015).
11. Z. Cui et al., "Deep network cascade for image super-resolution," *Lect. Notes Comput. Sci.* **8693**, 49–64 (2014).
12. C. Osendorfer, H. Soyer, and P. Van Der Smagt, "Image super-resolution with fast approximate convolutional sparse coding," *Lect. Notes Comput. Sci.* **8836**, 250–257 (2014).
13. C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," *Lect. Notes Comput. Sci.* **9906**, 391–407 (2016).
14. Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1256–1272 (2017).
15. Z. Wang et al., "Deep networks for image super-resolution with sparse prior," in *IEEE Int. Conf. Comput. Vision*, pp. 370–378 (2015).
16. K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn.*, pp. 399–406 (2010).
17. M. S. Sajjadi, B. Scholkopf, and M. Hirsch, "Enhancenet: single image super-resolution through automated texture synthesis," in *Proc. IEEE Int. Conf. Computer Vision*, pp. 4491–4500 (2017).
18. W.-S. Lai et al., "Fast and accurate image super-resolution with deep Laplacian pyramid networks," *IEEE Trans. Pattern Anal. Mach. Intell.* (2018).
19. B. Lim et al., "Enhanced deep residual networks for single image super-resolution," in *IEEE Conf. Comput. Vision and Pattern Recognit. Workshops*, Vol. 1, p. 4 (2017).
20. M. Haris, G. Shakhnarovich, and N. Ukita, "Deep backprojection networks for super-resolution," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit.* (2018).
21. C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *IEEE Conf. Comput. Vision and Pattern Recognit.* (2016).
22. B. Wu et al., "SRPGAN: perceptual generative adversarial network for single image super resolution," arXiv:1712.05927 (2017).
23. J. Li et al., "Similarity-aware patchwork assembly for depth image super-resolution," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 3374–3381 (2014).
24. J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," *Lect. Notes Comput. Sci.* **9906**, 694–711 (2016).
25. J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1646–1654 (2016).
26. J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1637–1645 (2016).
27. R. Timofte, R. Rothe, and L. Van Gool, "Seven ways to improve example-based single image super resolution," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, IEEE, pp. 1865–1873 (2016).
28. S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 3791–3799 (2015).
29. A. Kappeler et al., "Video super-resolution with convolutional neural networks," *IEEE Trans. Comput. Imaging* **2**(2), 109–122 (2016).
30. J. Caballero et al., "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, Vol. 1, p. 7 (2017).
31. Y. Huang, W. Wang, and L. Wang, "Bidirectional recurrent convolutional networks for multi-frame super-resolution," in *Adv. Neural Inf. Process. Syst.*, pp. 235–243 (2015).
32. M. S. M. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-recurrent video super-resolution," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* (2018).
33. X. Tao et al., "Detail-revealing deep video super-resolution," in *Proc. IEEE Int. Conf. Comput. Vision*, Venice, Italy, pp. 22–29 (2017).
34. T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *IEEE Spoken Language Technol. Workshop*, Vol. 12, pp. 234–239 (2012).
35. A. Graves, "Generating sequences with recurrent neural networks," arXiv:1308.0850 (2013).
36. N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proc. Conf. Empirical Methods Nat. Language Process.*, pp. 1700–1709 (2013).
37. Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," arXiv:1506.00019 (2015).
38. D. Erhan et al., "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.* **11**, 625–660 (2010).
39. D. Mishkin and J. Matas, "All you need is a good init," arXiv:1511.06422 (2015).
40. J. Andén and S. Mallat, "Multiscale scattering for audio classification," in *Proc. 12th Int. Soc. Music Inf. Retrieval Conf.*, Miami, Florida, pp. 657–662 (2011).
41. T.-H. Chan et al., "PCANet: a simple deep learning baseline for image classification?" *IEEE Trans. Image Process.* **24**(12), 5017–5032 (2015).
42. D. Zhang et al., "Learning from LDA using deep neural networks," *Natural Language Understanding and Intelligent Applications*, pp. 657–664, Springer (2016).
43. Q. Li, J. Zhao, and X. Zhu, "A kernel PCA radial basis function neural networks and application," in *9th Int. Conf. Control, Autom., Rob. and Vision*, pp. 1–4 (2006).
44. R. Zeng et al., "Tensor object classification via multilinear discriminant analysis network," in *IEEE Int. Conf. Acoust., Speech and Signal Process.*, IEEE, pp. 1971–1975 (2015).
45. Z. Feng et al., "DLANet: a manifold-learning-based discriminative feature learning network for scene classification," *Neurocomputing* **157**, 11–21 (2015).
46. Y. Rivenson et al., "Deep learning microscopy," *Optica* **4**(11), 1437–1443 (2017).
47. H. Wang et al., "Deep learning enables cross-modality super-resolution in fluorescence microscopy," *Nat. Methods* **16**, 103–110 (2019).
48. J. A. Grant-Jacob et al., "A neural lens for super-resolution biological imaging," *J. Phys. Commun.* **3**, 065004 (2019).
49. A. Sinha et al., "Lensless computational imaging through deep learning," *Optica* **4**(9), 1117–1125 (2017).
50. E. Nehme et al., "Deep-storm: super-resolution single-molecule microscopy by deep learning," *Optica* **5**(4), 458–464 (2018).
51. Y. Wu et al., "Three-dimensional propagation and time-reversal of fluorescence images," arXiv:1901.11252 (2019).
52. T. Nguyen et al., "Deep learning approach for Fourier ptychography microscopy," *Opt. Express* **26**(20), 26470–26484 (2018).
53. E. Moen et al., "Deep learning for cellular image analysis," *Nat. Methods* **16**, 1233–1246 (2019).
54. M. W. Conklin et al., "Aligned collagen is a prognostic signature for survival in human breast carcinoma," *Am. J. Pathol.* **178**(3), 1221–1232 (2011).
55. S. L. Best et al., "Collagen organization of renal cell carcinoma differs between low and high grade tumors," *BMC Cancer* **19**(1), 490 (2019).
56. C. R. Drifka et al., "Highly aligned stromal collagen is a negative prognostic factor following pancreatic ductal adenocarcinoma resection," *Oncotarget* **7**(46), 76197 (2016).
57. Leica Biosystems, http://www.leicabiosystems.com/digital-pathology/aperio-digital-pathology-slide-scanners/products/aperio-cs2/.
58. Meyer Instruments Inc., "Pathscan enabler IV, digital pathology slide scanner," https://www.meyerinst.com/scanners/pathscan-enabler-iv/.
59. N. Damera-Venkata et al., "Image quality assessment based on a degradation model," *IEEE Trans. Image Process.* **9**(4), 636–650 (2000).
60. N. S. Bunker, "Optimization of weighted signal-to-noise ratio for a digital video encoder," US Patent 5,525,984 (1996).
61. C. Ledig et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 4681–4690 (2017).

62. S. Edge et al., *AJCC Cancer Staging Handbook: From the AJCC Cancer Staging Manual*, Springer Science & Business Media (2010).
63. N. Wasif et al., "Impact of tumor grade on prognosis in pancreatic cancer: should we include grade in AJCC staging?" *Ann. Surg. Oncol.* **17**(9), 2312–2320 (2010).
64. A. Neesse et al., "Stromal biology and therapy in pancreatic cancer," *Gut* **60**(6), 861–868 (2011).
65. R. F. Hwang et al., "Cancer-associated stromal fibroblasts promote pancreatic tumor progression," *Cancer Res.* **68**(3), 918–926 (2008).
66. L. M. Wang et al., "The prognostic role of desmoplastic stroma in pancreatic ductal adenocarcinoma," *Oncotarget* **7**(4), 4183–4194 (2016).
67. D. Xie and K. Xie, "Pancreatic cancer stromal biology and therapy," *Genes Diseases* **2**(2), 133–143 (2015).
68. J. Lüttges et al., "The grade of pancreatic ductal carcinoma is an independent prognostic factor and is superior to the immunohistochemical assessment of proliferation," *J. Pathol.* **191**(2), 154–161 (2000).
69. M. D. Lagios et al., "Mammographically detected duct carcinoma in situ. Frequency of local recurrence following tylectomy and prognostic effect of nuclear grade on local recurrence," *Cancer* **63**(4), 618–624 (1989).
70. K. Lennert, *Histopathology of Non-Hodgkin's Lymphomas: Based on the Kiel Classification*, Springer-Verlag, New York (2013).
71. V. Jensen et al., "Proliferative activity (MIB-1 index) is an independent prognostic parameter in patients with high-grade soft tissue sarcomas of subtypes other than malignant fibrous histiocytomas: a retrospective immunohistological study including 216 soft tissue sarcomas," *Histopathology* **32**(6), 536–546 (1998).
72. X. Zhang et al., "Accelerating very deep convolutional networks for classification and detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(10), 1943–1955 (2016).
73. L. Mukherjee et al., "Convolutional neural networks for whole slide image superresolution," *Biomed. Opt. Express* **9**, 5368–5386 (2018).
74. W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1874–1883 (2016).
75. Semiconductor Research Corporation, https://www.src.org/.

Lopamudra Mukherjee is an associate professor in the Department of Computer Science at the University of Wisconsin–Whitewater. She graduated with a PhD in computer science from the University at Buffalo in 2008. Her research interests include computer vision, machine learning, and their applications in biomedical image analysis; she has published a number of papers in these areas.

Huu Dat Bui is a senior software engineer at IBM. He received his master's degree in computer science from the University of Wisconsin–Whitewater in 2019. Currently, he is working in the field of weather data and weather forecasts. He specializes in a data lake system. He is interested in machine learning and image processing areas. His passion lies at the junction of academic research meeting the needs and expectations of industry.

Adib Keikhosravi received his MSc and PhD degrees in biomedical engineering from the University of Wisconsin–Madison. During this time, he was working at the Laboratory for Optical and Computational Instrumentation (LOCI) to develop state of the art optical and computational imaging systems, image processing and machine learning tools for extracting stromal biomarkers from a variety of optical microscopy modalities during cancer progression. He has published several book chapters and research articles in peer reviewed journals.

Agnes Loeffler is currently the chair of the Department of Pathology at the MetroHealth System, Cleveland, Ohio. She received her medical degree from the University of Illinois, Urbana-Champaign and completed her residency in anatomic pathology at Dartmouth-Hitchcock Medical Center in New Hampshire. Her research interests are primarily in gastrointestinal pathology and informatics.

Kevin W. Eliceiri is the Walter H. Helmerich Professor of Medical Physics and Biomedical Engineering at the University of Wisconsin at Madison and investigator in the Morgridge Institute for Research in Madison, Wisconsin. He is also associate director of the McPherson Eye Research Institute. He has published over 200 papers on optical imaging instrumentation, open source image informatics, and role of the cellular microenvironment in disease. He is a member of both OSA and SPIE.