Smart Nanoscopy: A Review of Computational Approaches to Achieve Super-Resolved Optical Microscopy

SHIRAZ S. KADERUPPAN1,2, EUGENE WAI LEONG WONG1,2,3, (Member, IEEE), ANURAG SHARMA1,2,3, (Member, IEEE), AND WAI LOK WOO2,4, (Senior Member, IEEE)

1Newcastle Research and Innovation Institute Pte Ltd, (NewRIIS), Devan Nair Institute for Employment and Employability, Singapore 609607
2Faculty of Science, Agriculture and Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.
3Newcastle University in Singapore, SIT Building @ Nanyang Polytechnic, Singapore 567739
4Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K.

Corresponding author: Anurag Sharma (anurag.sharma@newcastle.ac.uk)

ABSTRACT The field of optical nanoscopy, a paradigm referring to the recent cutting-edge developments aimed at surpassing the widely acknowledged 200nm-diffraction limit in traditional optical microscopy, has gained recent prominence & traction in the 21st century. Numerous optical implementations allowing for a new frontier in traditional confocal laser scanning fluorescence microscopy to be explored (termed super-resolution fluorescence microscopy) have been realized through the development of techniques such as stimulated emission and depletion (STED) microscopy, photoactivated localization microscopy (PALM) and stochastic optical reconstruction microscopy (STORM), amongst others. Nonetheless, it would be apt to mention at this juncture that optical nanoscopy has been explored since the mid-late 20th century, through several computational techniques such as deblurring and deconvolution algorithms. In this review, we take a step back in the field, evaluating the various in silico methods used to achieve optical nanoscopy today, ranging from traditional deconvolution algorithms (such as the Nearest Neighbors algorithm) to the latest developments in the field of computational nanoscopy, founded on artificial intelligence (AI). An insight is provided into some of the commercial applications of AI-based super-resolution imaging, prior to delving into the potentially promising future implications of computational nanoscopy. This is facilitated by recent advancements in the field of AI, deep learning (DL) and convolutional neural network (CNN) architectures, coupled with the growing size of data sources and rapid improvements in computing hardware, such as multi-core CPUs & GPUs, low-latency RAM and hard-drive capacities.

INDEX TERMS Super-resolution microscopy, computational nanoscopy, high-resolution microscopical imaging, optical microscopy, deep learning.

I. INTRODUCTION

Optical microscopy has proven to be a ubiquitous tool and a gold standard for biological, geological and materials science research, as well as industrial quality control processes. Nonetheless, traditional optical microscopy suffers from numerous limitations, including (but not being restricted to) blurring/haze, lateral and axial resolution limitations [1], poor signal-noise ratio (SNR) and poor contrast at higher magnifications. In particular, an oft-cited equation describing the inherent resolution limitation faced by the compound optical microscope is the Abbe equation (proposed by Ernst Abbe in 1873 [1]), which may be defined as follows:

\[
\text{Abbe Lateral Resolution, } d_{x,y} = \frac{\lambda}{2n \sin \theta} \tag{1}
\]

\[
\text{Abbe Axial Resolution, } d_z = \frac{2\lambda}{(n \sin \theta)^2} \tag{2}
\]

where \(\lambda\) is the wavelength of the irradiating light, \(n\) is the refractive index of the imaging medium and \(\theta\) is the aperture angle of the light cone [1]. The product \(n \sin \theta\) is also sometimes defined as the numerical aperture (NA) of the lens.

In 1896, Rayleigh extrapolated (1) to include an additional contribution by the condenser optics, thereby contrasting episcopic (traditionally epifluorescence) microscopy with...
As such, numerous researchers globally have sought to circumvent these limitations through the development and exposition of both optical and computational approaches, prominently exemplified through the emergence of super-resolution fluorescence microscopy (as an optical enhancement of existing fluorescence microscopy methods) which culminated in the Nobel Prize in Chemistry being awarded to its developers (Moerner, Betzig and Hell) in 2014 [6]. Nonetheless, in this succinct (yet desirably impactful) review, we seek to evaluate some of the recently-employed computational advancements in the field of optical microscopy, with the intent that researchers worldwide would be inspired to address some of the existing limitations through further advancements in these in silico methodologies. This would inadvertently aid in potentially pushing the envelope of optical microscopy into the nanoscopy domain. In doing so, it would also be imperative to highlight the need for exploring the principles of image deconvolution, which is exemplified within the present review as well. In this light, Fig. 2 depicts a general timeline plot of publications which illustrates the advancements made in the field of nanoscopy over the decades:

At this juncture, it would be noteworthy to mention that optical nanoscopy procedures (such as STED [7], GSDIM [8], dSTORM [9], Lattice SIM [10], etc) are not discussed in the present review, as the focus of this study is to assess the computational aspects of nanoscopy. Nonetheless, as optical approaches are complementary to the computational aspects in most nanoscopy applications, the interested reader is encouraged to explore the afore-mentioned references (or [11]) for a detailed discussion on each of these techniques. The review is thus structured in the following manner: Section II presents an overview of some popular deconvolution algorithms utilized in microscopical imaging today, Section III discusses some commonly-used noise removal methods, while Section IV delves into AI and its current role in computational optical nanoscopy. The use of deep learning for image denoising is further presented in Section V and this is coupled with an exploration of the commercialized applications of current AI-based image enhancement approaches in Section VI. Section VII details the present limitations and potential future advancements in the field of computational nanoscopy, with the study being concluded in Section VIII.

II. DECONVOLUTION IN OPTICAL MICROSCOPY

Deconvolution methods have long been a source of image refinement and sharpening, although it would be prudent to

conventional diascopic brightfield illumination [2]:

\[
\text{Rayleigh Criterion, } d_{x,y} = \frac{0.61\lambda}{\text{NA}_{\text{obj}}} \quad \text{(for fluorescence)} \quad \text{or} \quad \frac{1.22\lambda}{\text{NA}_{\text{cond}} + \text{NA}_{\text{obj}}} \quad \text{(for brightfield)}
\]

where \(\text{NA}_{\text{cond}}\) and \(\text{NA}_{\text{obj}}\) refer to the numerical apertures of the condenser and objective optics respectively [2].

Evaluation of the Abbe and Rayleigh equations above yield a minimum lateral resolution of \(\sim 174\text{nm}\) (for Rayleigh) and 143nm (for the Abbe equation), when considering the shortest possible wavelength of visible light (400nm) and the highest possible NA attainable by most microscope objectives and condenser lenses today (\(\sim 1.4\)) [3] (lenses having NAs beyond this value are known to be manufactured but are excluded from the current context, as their applications are rather constrained and specialized in nature, e.g. TIRF [4]). A third method for lateral resolution computation (the Sparrow criterion) may be defined as \(d_{x,y} = \frac{0.472\lambda}{\text{NA}_{\text{obj}}^2}\), which evaluates to a value much closer to that obtained from the Abbe lateral resolution formula [1]. A diagram illustrating each of these 3 methods of resolution computation (and their differences) is depicted in Fig. 1:

As such, numerous researchers globally have sought to}
mention that deconvolution differs from sharpening, in that the former is intended to increase the resolution of an image, while the latter enhances edges in an image. Nonetheless (in some instances), the consequence of a resolution enhancement may very well be a sharpened image. Initially developed for temporal analysis by Wiener [12], deconvolution has gained prominence in the field of optical microscopy due to its ability to reassign out-of-focus light spots to their centroid maxima – made plausible due to the symmetric nature of the point spread function (PSF) of the light rays, as shown in Fig. 3:

Generally, the emergence of light rays through the optical train may be expressed by the mathematical convolution operation (\( \ast \)) as described in (4):

\[
(h \ast x) + \varepsilon = y
\] (4)

where \( h \) refers to the PSF (convolved with \( x \) as a consequence of the optical limitations of the system in question), \( x \) is the actual signal to be recovered, \( y \) is the detected signal/impulse response of the system and \( \varepsilon \) is the noise which has interacted with the convolved signal \((h \ast x)\) (adapted from [13]).

Notably, the convolution operator would be translated to multiplication in Fourier space, hence deconvolution may be perceived as division following a Fourier transform (FT) of the acquired impulse response \( y \), where the FT of the PSF is known as the Optical Transfer Function (OTF) [1]. Nonetheless (and as expected in most instances), the introduction of noise \( \varepsilon \) makes it exceedingly difficult to accurately obtain the unaltered signal \( x \), although numerous techniques (both optical and computational) have been proposed to counter and mitigate the effects of \( \varepsilon \) on the detected impulse response \( y \).

In this regard, optimal performance in deconvolution has been reported for thin (<50\( \mu m \)) sections, or optically-transparent material with little fluorescence, posing a challenge for live cell imaging applications due to motion blurring and enhanced spherical aberration effects [14]. Fig. 4 illustrates an image subjected to a Gaussian blur kernel/PSF and its corresponding deconvolved image, while Fig. 5 shows the diffraction limitations imposed by the OTF (referred to as the Abbe limiting frequency) [1]:

As such, it would be noteworthy to explore the various current computational approaches to resolving this dilemma, although one should be aware that none of these methods provide a perfect solution in reality. Moreover, the problem is further exacerbated by spatiotemporally-variant PSFs (partially elucidated in [15] which describes a non-linear variation in the PSF across different axial planes), coupled with complications such as the variable optical density of the specimen and mountant at different locations. In this regard, it would be essential to holistically evaluate the various deconvolution and denoising algorithms being utilized today, emphasizing on their underlying principles, advantages and shortcomings (where appropriate). Past studies (such as [16]) have attempted to assay and categorize the numerous deconvolution algorithms presently available into 3 main classes, namely (i) deblurring, (ii) inverse filtering and (iii) constrained iterative protocols, with both (ii) and (iii) being regarded as image restoration algorithms [14]. Further details on each of these algorithms and their respective classes are as depicted in Table 1 below (information adapted from [14] and [17]):
A. DEBLURRING

Deblurring algorithms (as the precursor to deconvolution) seek to remove haze (or blurring) in the image, although this might not result in a significant improvement in image resolution. Hence, such algorithms are often utilized for quick image inspections (e.g. defect identification), rather than in-depth analysis. Common deblurring approaches include the No Neighbors or Nearest Neighbors algorithms [18] – the former referring to the projection of the blur kernel from the image plane itself, while the latter utilizes the impulse response from a point source above and below the image plane as a reference to predict the extent of blurring introduced in the image. In addition, the Unsharp mask [19] (an image sharpening tool) also represents a popular dehazing tool employed in professional photography and image-editing applications, such as Adobe Photoshop. Further details on the mathematical principles underpinning each of these methods are discussed in the paragraphs which follow.

The No Neighbors deconvolution algorithm assumes that the blurring is caused by out-of-focus light originating from the same image plane as the object being imaged and is described by lower spatial frequencies [18]. Hence, the No Neighbors deconvolution algorithm seeks to eliminate this out-of-focus light by implementing high-pass filters, thereby amplifying the proportion of high spatial frequencies in the image. Mathematically, this may be expressed as follows (adapted from [18]):

$$\widetilde{I} = I^m \ast (I^m \ast H_g)$$

where $\widetilde{I}$ is the sharpened image, $I^m$ is the original blurred image, $H_g$ is the Gaussian blur kernel (derived from the PSF) and $\ast$ is the 2D convolution operator.

In addition, [18] also states how the afore-mentioned equation may be improved through the incorporation of a weighted factor (herein represented by $\beta$) so that the above equation becomes:

$$\widetilde{I} = \beta \cdot I^m \ast \left[ (1 - \beta) \cdot (I^m \ast H_g) \right]$$

where the range of $\beta = [0.6, 0.85]$ to yield optimal results for $\widetilde{I}$ in most instances [18].

In contrast, the Nearest Neighbors algorithm utilizes the impulse response from the optical planes both immediately before and after the image plane in an acquired Z-stack to be convolved with a suitable blur kernel and the result subtracted from the blurred image plane to obtain the sharpened image [18]. In this respect, the Nearest Neighbors algorithm is based on the premise that the out-of-focus blur in the image plane is primarily generated from the optical planes both immediately above and below the image plane, so that elimination of the defocused optical planes from the image plane would yield the sharpened image. As previously, this may be expressed mathematically by the following equation (adapted from [18]):

$$\widetilde{I} = I^m \gamma \ast \left[ (I^m_{-1} \ast H_g) + (I^m_{+1} \ast H_g) \right]$$
where $I_{m}^{1}$ and $I_{m}^{2}$ are the images acquired in the optical planes immediately before and after the evaluated image plane within a Z-stack, $\gamma$ is the additional weighting factor. In this respect, an apparent drawback of the Nearest Neighbors algorithm refers to the step-size in the acquired Z-stack, which may be perceived as a modulated function of the depth-of-field (DoF) of the microscope objective lens being used. An axial step-size greater than the DoF would result in an inaccurate approximation of the blur introduced by the image matrices $I_{m}^{1}$ and $I_{m}^{2}$, which would consequently result in the computation of a non-optimal raw image $I_{m}$.

A third deblurring algorithm often utilized in image processing is the Unsharp mask (which is akin to the No Neighbors deconvolution algorithm). However, in Unsharp masking, the edge-enhancing kernel is obtained by subtracting a smoothed/blurred version of the original image from the original image [19]. This kernel is then added to the original image to obtain a sharpened image [19]. Mathematically, this may be illustrated by the following matrix computation (adapted from [19]):

$$M = I_{m} - (I_{m} \ast H) = I_{m} - \tilde{I}_{m}$$  \hspace{1cm} (8)

where $H$ is the smoothing kernel, $\tilde{I}_{m}$ is the smoothed (blurred) version of $I_{m}$, $M$ is the mask image and $\tilde{I}_{m} = \min[(I_{m} + \alpha \cdot M), 255]$  \hspace{1cm} (9)

where $\tilde{I}_{m}$ is the sharpened image of $I_{m}$ and $\alpha$ is the weight controlling the extent of sharpening by the mask $M$.

Visually, an image deblurred through implementation of the Unsharp mask algorithm is presented in Fig. 6 below:

**FIGURE 6.** A Raw image of an anther. B Unsharp mask applied to the raw image in A using Adobe PhotoShop CS5 (© 1990-2010 Adobe Systems Inc.). Notice the highlighted structural details evident upon application of the Unsharp mask algorithm. Filter parameters used are 200% (Amount), 33.6 pixels (Radius), 2 levels (Threshold).

B. INVERSE FILTERING

Inverse filtering represents another set of frequently employed image deconvolution algorithms. However, as the principle of inverse filtering is simple (to introduce a deconvolution kernel $h^{-1}$ which restores the blurred impulse response $y$ to its original state $x$, where $h \ast x = y$, $\ast$ representing the convolution operator), the effective use of inverse filtering is determined by the level of noise (both photon and detector noise) in the image – the higher the noise, the less effective the deconvolution. Commonly used inverse filters include Naive inverse filtering (NIF), the Tikhonov-Miller filter (an example of a linear inverse filter) and the Wiener filter (a regularized inverse filter). An additional approach couples Tikhonov regularization with NIF, minimizing the latter’s contribution of noise [20]. The specific details on each of these inverse filtering algorithms are outlines in the following sub-sections for the interested reader.

1) NAIVE INVERSE FILTERING (NIF)

NIF represents one of the most basic deconvolution protocols, which seeks to minimize a least-squares cost function [20]. However, in so doing, NIF also accentuates measurement noise, resulting in an increased number of high-frequency waveforms detected in the impulse response of the system. Mathematically, NIF may be represented by the following equation:

$$\hat{x} = \arg \min_{x} \xi \left( x \right)$$  \hspace{1cm} (10)

where $\xi \left( x \right) = \|y - Hx\|^2$, $y$ is the observed data, $H$ is the PSF matrix and $x$ is the underlying fluorescence signal [20].

When Gaussian noise is present, NIF reduces to the concept of maximum-likelihood estimation, which may be determined via a quotient of coefficients in Fourier space as follows:

$$\hat{x} = \frac{\hat{y}}{\max(H, \epsilon)}$$  \hspace{1cm} (11)

where $\hat{x}$, $\hat{y}$ and $\hat{H}$ represent the discrete FT coefficients of $\hat{x}$, $y$ and the PSF, max represents the element-wise maximum and $\epsilon$ is a constant term, included to circumvent divisions by zero (adapted from [20]).

Subsequently, the inverse FT of $\hat{x}$ is used to obtain the final solution, where a regularization parameter defined as the squared Euclidean norm of $x$ ($\|x\|^2$) may be added to $\xi \left( x \right)$ to accord large values with a penalty (a procedure known as Tikhonov-regularized NIF/TRNIF) as follows [20]:

$$\xi \left( x \right) = \|y - Hx\|^2 + \lambda \|x\|^2$$  \hspace{1cm} (12)

where $\lambda$ is a weighting factor determining the contribution of the 2 terms in the equation [20].

The above equation may then be minimized by applying the following relation, which may be considered as a maximum a posteriori (MAP) model, since $\lambda$ is used to introduce prior information about $x$ to facilitate its estimation:

$$x = \left( H^TH + \lambda I \right)^{-1} H^Ty$$  \hspace{1cm} (13)

where $I$ is the identity matrix [20].

2) TIKHONOV-MILLER (TM) AND ITERATIVELY CONSTRAINED TIKHONOV-MILLER (ICTM) FILTER

Tikhonov-Miller (TM) filtering (as an example of linear inverse filtering) is often utilized as a preliminary strategy for image deconvolution prior to iterative deconvolution. This may be attributed to it being computationally economical and rapid, although TM filtering is simultaneously susceptible to artifact generation, allowing noisy effects to be effectively
transferred and thus unsuited for enhancing an image’s resolution [14]. The TM algorithm uses an iterative gradient descent approach to minimize the regularized inverse filter cost, allowing the implementation of a positivity constraint at each iteration [17]. An alternative form of the TM filter refers to the iteratively constrained Tikhonov–Miller (ICTM) filter, which is formed with repeated incremental projections of TM onto the set \((\mathbb{R}^+)^K\), where \(K\) represents the dimensionality of the fluorescence signal \(x\) (i.e. \(x \in \mathbb{R}^K\)) [20]. Mathematically, the iterative construct of ICTM may be expressed by the following equation from [20]:

\[
x^{(k+1)} = P_{(\mathbb{R}^+)^K} \left\{ x^{(k)} + \gamma \left( H^T y - \left( H^T H + \lambda L H \right) x^{(k)} \right) \right\}
\]

where \(P_{(\mathbb{R}^+)^K} \{ a \} = \max(a, 0)\) and indicates the component-wise projection of \(a\) onto the set \((\mathbb{R}^+)^K\), \(\gamma\) is a weighting factor and \(L\) represents the matrix corresponding to the discretization of a differential operator [20].

3) REGULARIZED INVERSE FILTERING (RIF) AND WIENER FILTERS

Yet another regularized approach to inverse filtering (other than TRNIF) involves subjecting \(x\) to a smoothness constraint through reducing the impact of its – a procedure known as regularized inverse filtering (RIF). The cost function for RIF may be described by the following expression [20]:

\[
\xi(x) = \left\| y - H \left( H^T H + \lambda L^T L \right)^{-1} H^T y \right\|^2 + \lambda \left\| (H^T H + \lambda L^T L)^{-1} H^T y \right\|^2
\]

The above expression may also be effectively reduced to (12) when the explicit minimizer \(x = \left( H^T H + \lambda L^T L \right)^{-1} H^T y\) [20], thereby proving (15) to be a reformulation of TRNIF.

According to [20], defining \(\lambda\) as the inverse of the noise variance and coupling it with \(L^T L\) filtering poses a whitening effect on \(x\), thereby effectively converting RIF to Wiener filtering. Wiener filtering shall be discussed in greater detail in a subsequent section in this review (Noise removal), where it is being used to both deconvolve and restore a blurred, noisy image. Nonetheless, [21] has also represented the Wiener filter via the mathematical relationship below and regarded it as the “golden linear deconvolution trade-off”:

\[
y = \frac{H^* x}{|H|^2 \cdot |x| + \varepsilon}
\]

where \(H^*\) is the complex conjugate of \(H\) and \(|H|\) refers to the magnitude of \(H\) (adapted from [21]).

For illustration purposes, a single image deconvolved using ICTM and RIF is depicted in Fig. 7 below:

**FIGURE 7.** A Raw image of an oesophagus section; B The raw image in Adconveolved using ICTM \((N = 20, y\text{-step } 0.7, \lambda = 1.0000E-05)\); C The same image in A processed using RIF (\(\lambda = 5\)). All deconvolution processes were performed using DeconvolutionLab2 (c 2018 EPFL) [17].

**C. NON-BLIND DECONVOLUTION**

Non-blind deconvolution (NBD) algorithms utilize an empirically determined PSF to deconvolve an image [22]. Most suited for constant blurring across a specified region of interest, NBD algorithms require the user to image a sample containing sub-diffraction-sized beads [22] (often \(\approx 80\)-150nm in diameter, or 0.61\(\lambda\)/3 NA, where \(\lambda = 550\)nm) to compute the PSF of the optical train and subsequently factor the FT of this value (as a divisor) into the FT of the impulse response to obtain the FT of the deconvolved image. Ideally, the sample containing the beads should be the same sample imaged under identical conditions to minimize deviations in the acquired PSF caused by differences in refractive index, optical density and spherical aberration [22]. A sample image of the PSF measured using this mode of deconvolution (as well as the associated images generated via application of the NBD algorithm) is shown in Fig. 8 as follows:

NBD algorithms (generally) have a number of constraints, including the need for sample clarity and absence of dust, the axial focal range used for acquiring the Z-stack, the objective type and matching mountant being used, the temperature of the sample being imaged, etc [22]. Despite satisfying these conditions, one may still encounter setbacks through the employment of a NBD algorithm, due to the varying PSF caused by sample-coverslip distance deviations, noise or inherent differences between the imaging conditions of the beads and the sample [22]. However, NBD algorithms are still widely utilized today in popular scientific imaging application suites such as Huygens Professional (c Scientific Volume Imaging B.V.) [23] and AutoQuant X3 (c Media Cybernetics, Inc.) [24], as they minimize the num-
ber of iterations required to attain an optimally-deconvolved image while accounting for individual setup-specific deviations/aberrations [22]. In this regard, NBD algorithms are particularly suited for imaging fixed samples, where most of the confounders, such as varying sample-coverslip distances due to the movement of the sample (for live cell imaging) are generally absent.

D. BLIND DECONVOLUTION

Blind deconvolution (as its name implies) refers to the deconvolution of an image without an exact understanding of the blur kernel/PSF employed. Sporadically, this problem may be further amplified with the introduction of noise in the blurred image. Blind deconvolution (BD) may be performed in either an iterative or a non-iterative manner (as expounded in [25]), with non-iterative methods such as SeDDaRA [26] estimating the PSF through a comparison of the spatial frequency of the blurred vs the target image. In contrast, a common iterative approach adopted by numerous researchers ([3], [4], [5]) refers to the maximum a posteriori (MAP) distribution as described mathematically in [27] below (expectation-maximization algorithms are also exploited as iterative BD methods):

$$\arg \min_{u,k} \sum_{x \in F} \left( (k \circ u) [x] - f [x] \right)^2$$ (19)

where $k \geq 0$, $\|k\|_1 = 1$

Here, the authors of [27] have defined $o$ to be the valid convolution (which is also employed in MATLAB), with $k \circ u \neq u \circ k$ ($u \circ k$ being undefined if the support of $k$ is large) [27].

By doing so and utilizing the findings of [37], [38] and [39] through implementation of the AM algorithm for a 1D signal domain, [27] has derived an expression for $\lambda$ based on a clear (but linearly-transformed) $u^0$ as the solution of the following denoising problem expressed mathematically:

$$\hat{u} [x] = \arg \min_{u} \frac{1}{2} \sum_{x=-L+1}^{L-1} (u [x] - f [x])^2$$

$$+ \lambda \sum_{x=-L}^{L-1} |u [x+1] - u [x]|$$ (20)

Nonetheless, the scaling of $u^0$ (induced by the use of the total variation regularizer) was claimed by [27] to be the primary factor accounting for the non-convergence of the AM algorithm towards a global minimum, resulting in the postulation of PAM (an iterative construct similar to AM in $u$, but alternating with an unconstrained convex problem in $k$) being described by the following equations:

$$k^{t+\frac{1}{3}} \leftarrow \arg \min_{k} \sum_{x \in F} ((k \circ u) [x] - f [x])^2$$ (21)

$$k^{t+\frac{2}{3}} \leftarrow \max \left\{ k^{t+\frac{1}{3}}, 0 \right\}, k^{t+1} \leftarrow \frac{k^{t+\frac{2}{3}}}{\|k^{t+\frac{2}{3}}\|_1}$$ (22)

PAM implementation (based on the following protocol from [27]) and its results are depicted in Fig. 9:

The significant improvement of images deconvolved using PAM (over traditional TV-based regularizers in blind deconvolution) may thus hold promise in the utilization (and potential further development) of PAM to achieve nanoscopic super-resolution in the field of optical microscopy.

E. RICHARDSON-LUCY (RL) DECONVOLUTION

In 1972 and 1974, 2 researchers ([40], [41]) independently proposed an approach to address the issue of deblurring images captured off the sensing plane with the addition of noise. This approach came to be known as the Richardson-Lucy (RL) model for image deconvolution, which is still being widely used today.

Fundamentally, the RL deconvolution algorithm seeks to solve for $x$ given the values of $h$ and $y$ in (4) previously. Elucidation of $x$ is made possible through an understanding of the Poissonian nature of photoelectron distribution and image formation [42] exemplified by the Poisson probability mass function $P(n)$ as follows (adapted from [43] for non-mathematically-inclined readers):

$$P(n) = \frac{\nu^n e^{-\nu}}{n!}$$ (23)

where $P(n)$ (in the current context) refers to the probability of a single pixel receiving $n$ photoelectrons and $\nu$ is the expected (average) number of photoelectrons collected per pixel.
From [42], one can thus deduce the probability of the collected photoelectrons contributing to the formation of the entire image $I^m$ as follows (adapted from [42]):

$$p(I^m|o) = \prod_a \left( \frac{(h \ast x)^Y e^{-(h \ast x)}}{y!} \right)$$  \hspace{1cm} (24)

where an object $o$ is projected as an image $I^m$ through an optical system having a PSF $h$ and subject to Poisson noise. Notice that the Poisson parameter $\nu$ in (23) has been replaced by the convolution $(\ast)$ of $x$ with $h$, so that $p(I^m|o) = \prod_a [P_a(n)],$ where $P_a(n)$ refers to $P(n)$ at pixel $a$ (a being a 2-D or a 3-D vector, represented as $(X,Y)$ or $(X,Y,Z)$ respectively [42]).

In order to maximize the value of $p(I^m|o)$ [a process known as maximum likelihood estimation (MLE) which represents a common paradigm employed in machine learning] so that the image $I^m$ closely resembles $o$, we may reduce (24) to the following (also from [42]) which we seek to minimize:

$$J_l(o) = -\ln[p(I^m|o)]$$

$$= \int_J [h \ast x - y \cdot \ln (h \ast x) + \ln (y!)] \, dx$$

$$\Leftrightarrow \int_J [h \ast x - y \cdot \ln (h \ast x)] \, dx$$  \hspace{1cm} (25)

where $J_l(o)$ is the first-order Bessel function while $\ln(y!)$ is a constant (with respect to o).

[43] then further proceeds to elaborate on how (25) may be converted (through a series of steps not described here) to the following final equation, which defines the RL deconvolution algorithm:

$$y_k+1 = y_k \left[ h \ast \left( \frac{y}{(h \ast x_k)} \right) \right]$$  \hspace{1cm} (26)

where $k$ is the iteration count, $y_{k+1}$ $= 1$ at convergence and $h$ represents the PSF $h$ at a complementary point about the origin.

Notably, (26) above represents the multiplicative form of the RL deconvolution algorithm. [42] also provides an additive gradient-descent variant of the RL deconvolution algorithm as follows:

$$o_{k+1} = o_k + \delta t \left( 1 - h \ast \frac{y}{(h \ast x_k)} \right)$$  \hspace{1cm} (27)

where $\delta t$ represents the gradient descent step size.

Nonetheless, [42] has observed that the vanilla RL algorithm may not always converge to a particular solution, as no prior information on $o$ has been supplied to it, highlighting the need for a prior model to be constructed on $o$ (a process termed regularization) [42]. [42] further described 2 common regularizers commonly-utilized in conjunction with RL deconvolution – (i) TM regularization [indicated in (28) below] and (ii) Total Variation (TV) regularization [indicated in (29) below], both of which are sourced from [42]:

$$x_{k+1} = \left[ \left( \frac{y}{(x_k \ast h)} \right) \ast h \right] \cdot \frac{x_k}{1 + 2 \lambda_{TM} \Delta x_k}$$  \hspace{1cm} (28)

where $\lambda_{TM}$ is the TM regularization parameter.

$$x_{k+1} = \frac{x_k}{1 - \lambda \cdot \text{div} \left( \frac{\nabla o_k}{|\nabla o_k|} \right)} \left[ h \ast \left( \frac{y}{(h \ast x_k)} \right) \right]$$  \hspace{1cm} (29)

where $\lambda$ is the TV regularization parameter and $\text{div}(F)$ is the divergence of $F$.

A sample image implementing the RL deconvolution algorithm is depicted in Fig. 10 below:

As with PAM [27] discussed in the previous section, further improvements made in the RL deconvolution algorithm seem promising for achieving nanoscopic resolution in optical microscopy in the near future.
F. JANSSON-VAN CITTERT DECONVOLUTION

The Jansson-Van Cittert deconvolution (JVCD) algorithm was well-described by Sastry [45] and illustrated in the flow diagram (adapted from [45]) as follows:

Generally, an initial guess \( x^0 \) is first provided of the actual signal \( x \) and subjected to the convolution kernel \( h \), yielding a convolved signal \( y^0 \). This is then compared with the actual acquired signal \( y \) and the difference between these two impulse responses computed [45]. If this difference evaluates to 0, the prior used \( x^0 \) is the desired ‘deconvolved’ signal, else the difference \( (y - y^0) \) is evaluated and a weight \( \alpha \) applied to this difference to be added to the input signal \( x^0 \) to produce \( x^1 \) (the first iteration) [45]. This process repeats itself, until \( (y - y^i) \) approaches 0 for the \( i \)th iteration. A sample of an image (though not a photomicrograph) implementing the JVCD algorithm with a large \( p_{11} \) psf on \( O_x \) and \( O_y \) respectively, \( p_{11} = [17^\circ \times 17^\circ, N(8, 2.6), \text{fwhm} = 6.6^\circ ]; p_{31} = [31^\circ \times 31^\circ, N(15, 4), \text{fwhm} = 9.4^\circ ] \). Figures reprinted with permission (and adapted) from [46], copyright Lunar and Planetary Institute.

G. ARTIFACT-FREE DECONVOLUTION

Artifact-free deconvolution was proposed by [48] for use with the light field microscope (LFM) – an imaging modality developed by [49] which has been utilized in numerous biomedical and cellular imaging applications (see [50], [51], [52], [53] for details). Here, the authors of [48] claim that the LFM has a depth-dependent sampling pattern, resulting in variable lateral resolution at different optical planes across the sample. According to [48], the LFM utilizes micro-lenses for 3D fluorescent microscopical imaging, thereby eliminating the inherent scanning present in traditional confocal laser-scanning fluorescence microscopy, while allowing the vectorial light field to be acquired in a single exposure for topographical reconstruction of the imaged sample [48]. The LFM was thus conceived by [48] to belong to a family of plenoptic devices [54] which allow 3D imaging [49] and refocusing of the acquired images ([55, [56, [57]). In this respect, [48] highlights the numerous past studies seeking to improve the LFM resolution (namely via multi-view reconstruction, ray-based and wave-based methods), with a particular study [58] depicting laterally well-resolved images. Nonetheless, [58] also reported that the improvement rate was inconsistent at different imaged depths – a phenomenon attributed to the variable sampling patterns at different axial planes coupled with aliasing effects. As such, [48] sought to resolve these artifacts by proposing “depth-dependent anti-aliasing filters” [48, p. 31645] applied to an iterative “aliasing-aware deconvolution method for artifact-free 3D reconstruction” [48, p. 31645] via an expectation maximization protocol. A sample image of the USAF 1951 target imaged using a light field microscope (left), a comparative deconvolution method discussed in [58] and used for image reconstruction (center), the anti-aliased image following application of the presently reviewed approach (right). Notice the image produced in [48] depicting reduced aliasing (increased smoothing) between neighboring pixels. Figures adapted with permission from [48], © The Optical Society.

\[
\gamma_z = \left| \frac{d_{\text{m}l}}{d_{\text{ml}a}} \right| \frac{z'''}{d_{\text{ml}a}''} - z'''
\]

(30)

\[
b_z = r_{\text{ml}} \left| \frac{1}{z'''} - \frac{d_{\text{m}l}}{d_{\text{ml}a}} \right|
\]

(31)
that under a microlens, $d_{\text{mla}}$ is the distance between the sensor and the MLA planes, $d_{\text{mla}}$ is the distance between the tube lens and the MLA planes, $z$ is the conjugate image plane formed by the MLA, $r_{\text{ml}}$ is the micro-lens radius and $b_z$ is the blur radius (Adapted from [48]).

Subsequently, the authors of [48] proceeded to evaluate a probable radius of the ideal filter kernel, $w_{\text{obj}} = \frac{w_{\text{sen}}}{P_{\text{sen}}}$, prior to subsequently proposing the depth-dependent ideal filter $h_{\text{obj}}$, with individual kernels at different depths (defined by $w_{\text{obj}}$ as previously) and the use of a Lanczos2-constrained sinc kernel as a non-ideal anti-aliasing filter (further details of these are expounded in [48] for the interested reader). Following this, the authors of [48] then derived an equation describing a 3D aliasing deconvolution schema (based on an expectation-maximization-smoothing approach) which depicted significant improvement in artifact removal when compared with a different study [58] as shown in Fig. 14:

**3D Aliasing Schema (From [48]):**

$$v^{q+1} = h_{\text{obj}} \odot \frac{v^q}{A^T} \left[ A^T m / A^T v \right]$$ (32)

where $v$ refers to the imaging volume to be constructed, $q$ is the iteration count, $h_{\text{obj}}$ is the axially-based anti-aliasing filter, $A$ defines the forward operation of the LFM and $m$ refers to the noisy measurements acquired through the LFM [48].

## H. LEAST SQUARES DECONVOLUTION

The Least squares deconvolution algorithms category comprises of 3 main classes, namely (i) Linear Least Squares (Landweber), (ii) Non-Negative Least Squares and (iii) Bounded-Variable Least Squares deconvolution. In this regard, we seek to evaluate each of these 3 algorithms in greater depth as follows:

The Landweber deconvolution (LD) approach utilizes similar principles as NIF, i.e. it seeks to minimize the same least squares cost function $C(x) = \| y - Hx \|^2$ (adapted from [20]), although unlike the latter, it does this in an iterative fashion using gradient descent ([59] from [20]). Specifically, [20] utilizes the iterative feature of LD as a means of ensuring each iteration yields positive results, which may be expressed mathematically as follows (sourced from [20]):

$$x^{(k+1)} = P_{\mathbb{R}^+} \left\{ x^{(k)} + \gamma H^T \left( y - Hx^{(k)} \right) \right\}$$ (33)

where $\gamma$ is the step size, $k$ is the iteration count, $y$ is the observed volume, $H$ is the PSF matrix and $P_{\mathbb{R}^+} \{ x \} = \max (x, 0)$, which represents the projection of $x$ onto the domain $\mathbb{R}^+$ in a component-wise fashion.

Notably, [20] mentions that the number of iterations is only suggestive – iterative convergence of the computed solution results in over-fitting of the noise in the input data, while an optimal SNR may be achieved through premature truncation of the algorithm. Additionally, [20] highlights that the number of iterations is a pseudo regularizer common to all MLE algorithms, of which the LD is an example. Other studies ([60] and [61]) have also sought to encapsulate the Landweber algorithm with Shannon wavelets [60] or Shannon sheartlets [61], the former demonstrating an accelerated algorithmic convergence (by a single order of magnitude) while the latter exhibited improvements for motion deblurring.

The Non-Negative Least Squares (NNLS) algorithm extends the LD algorithm to include a constraint of non-negativity, while seeking to minimize $C(x)$ [17]. Similarly, the Bounded Variable Least Squares (BVLS), also known as the Stark-Parker (SP) [17], algorithm represents an alternative approach to NNLS, by imposing limits (both an upper- and a lower-bound active set) as an additional constraint on $C(x)$ [62]. Mathematically, the problem which the BVLS algorithm seeks to solve may be expressed as follows:

$$\min_{\ell \leq x \leq b} \| Ax - b \|_2$$ (34)

where $A$ is a 2-dimensional $(m \times n)$ matrix, $l$, $x$, $u \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $\| x \|_2$ refers to the Euclidean ($\ell_2$) norm of $x$.

It would be prudent to highlight at this juncture that BVLS incorporates some differences from NNLS, including (but unrestricted to) a re-computation of the QR decomposition each time the following calculation is performed (in order to improve numerical stability) [62]:

$$z = \arg \min_z \| A'z - b' \|_2^2$$ (35)

for some variable $z$, where $A'$ is a matrix of the columns in $A$ whose indices are ‘free’ components of $x$ between $l$ and $u$ and $b'$ is a data vector excluding the bound variable predictions [62]. By doing so, the BVLS algorithm allows a return of the last free and bound sets in a sequence of relations (where the upper and lower limits are well-defined) in a computationally-efficient manner.

Visually, image deconvolution performed using LD, NNLS and BVLS in DeconvolutionLab2 may be described by Fig. 15:

## I. WAVELET DECONVOLUTION

Wavelet deconvolution refers to a field of image processing which primarily involves the use of wave kernels (often a single oscillation) to convolve with the detected impulse response of a system so as to identify the presence of a frequency corresponding to the mother wavelet in the said impulse response. The popularity of wavelets has resulted in the development of a large family of wavelets, which may
wavelets [63] refer to a square wave kernel, having a mother (which refers to the Haar wavelet) or continuous wavelets being orthogonal and having an unbound (non-compact) support [64], with some spline wavelets (as discussed in [65]) being orthogonal and having an unbound (non-compact) support. Nonetheless, the term ‘spline wavelets’ has been popularly characterized (in most instances) to refer to a specific class of B-spline wavelets which have a compact support and are not orthogonal, termed B-spline wavelets or cardinal B-spline wavelets ([66], [67]). Cardinal B-spline wavelets have numerous properties, although they all seek to solve the following mathematical equation (where $N_m$ refers to the B-spline wavelets of order $m$ having knots in the set $\mathbb{Z}$) from [66]:

$$N_1 (x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \quad (37)$$

$$N_m (x) = \int_0^1 N_{m-1} (x-t) \, dt, \quad \text{for } m > 1, m \in \mathbb{Z}^+ \quad (38)$$

Essentially, such cardinal B-splines may span multiple orders, from $m = 1$ (for constant B-splines) to $m = 5$ (for quintic B-splines), as an example.

Conversely, compactly supported B-spline wavelets having a support of $[0, m]$ are based on the following generalized scaling and B-spline wavelet functions [represented as $\varphi_m (x)$ and $\psi_m (x)$ in (39) – (40) respectively], exhibiting up (but not being restricted) to the sextic order [68]:

$$\varphi_m (x) = \sum_{k=0}^m p_k \varphi_m (2x - k) \quad (39)$$

where the two-scale sequence $p_k = 2^{1-m} \binom{m}{k}$ (for $0 \leq k < m$), and

$$\psi_m (x) = \sum_{k=0}^{3m-2} q_k \varphi_m (2x - k) \quad (40)$$

where $q_k = (-1)^k 2^{1-m} \sum_{l=0}^m \binom{m}{l} \varphi_{2m} (k-l+1)$.

Here, [68] defines the order $m$ of a B-spline wavelet as being 1 more than the highest exponent variable in its scaling function $\varphi_m (x)$. The interested reader would henceforth be encouraged to refer to [68] for a further detailed mathematical treatment of the decomposition function $\varphi_m (2x - k)$.

On a separate note, a visual representation of the real part of a complex frequency B-Spline wavelet having order $m = 3$, a bandwidth $FB$ of 0.5 and center frequency $FC$ of 1 is depicted in the following diagram (Fig. 17):

A third class of B-spline wavelets is the Battle-Lemarie (BL) wavelets [69] which is defined from the cardinal B-spline wavelets, but have their expressions derived in the Fourier (i.e. frequency) domain of the mother wavelet. Here, the BL wavelet of $m^{th}$ order ($\psi_{BL,m} (t)$) has the following FT (denoted as $\hat{\psi}_{BL,m} (\omega)$ and adapted from [69]):

$$\hat{\psi}_{BL,m} (\omega) = \frac{e^{-i\omega/2} \varphi_m (\omega + 2\pi) \varphi_m (\omega - 2\pi)}{\varphi_m (\omega + \pi) \varphi_m (\omega - \pi)} \quad (41)$$
where $\hat{\varphi}_{m}^{\omega}(\omega)$ is the FT of the scaling function of $\psi_{BL,m}(t)$ (i.e., $\varphi_{m}(t)$) and $\bar{x}$ represents the complex conjugate of $x$ [69]. Here too, the interested reader is encouraged to refer to [69] for a complete treatment and proof of the BL wavelets through a series of fast FT (FFT) computations.

1) ITERATIVE SHRINKAGE THRESHOLDING ALGORITHM (ISTA) AND FAST ISTA (FISTA)

The ISTA and FISTA algorithms represent 2 other methods employing wavelets which are coupled with an expectation-maximization (EM) algorithm that is founded on a likelihood penalization approach in the wavelet domain for the derivation of a maximum penalized likelihood estimator (MPLE) [70]. Here, both ISTA and FISTA utilize the property of sparsity of wavelets to preserve image details and discontinuities [20], being particularly adapted to evaluating linear inverse problems [71]. Nonetheless, the use of a non-smooth Manhattan ($l_{1}$) norm for FISTA allows efficient solving of the cost function up to several orders of magnitude faster than ISTA [71]. Mathematically, the equations defining ISTA are indicated as follows (adapted from [70]):

**Expectation (E)-Step:**

$$Q(\theta, \hat{\theta}^{(t)}) = E \left[ \log p(y, z | \theta) | y, \hat{\theta}^{(t)} \right]$$

(42)

**Maximization (M)-Step:**

$$\hat{\theta}^{(t+1)} = \arg \min_{\theta} \left\{ Q(\theta, \hat{\theta}^{(t)}) - \text{pen}(\theta) \right\}$$

(43)

where pen($\theta$) refers to the penalized value of $\theta$.

Essentially, the EM algorithm in ISTA alternates between an FFT-derived E-step and a M-step centered on the Discrete Wavelet Transform (DWT), thereby increasing the operational efficiency of the iteration to achieve a linear logarithmic complexity, i.e., O(NlogN) [70]. In addition, [70] has also shown ISTA to be convergent towards a global maximum under suitable conditions, while potentially out-performing other leading deconvolution methods in benchmark tests. Conversely, FISTA may be employed with a (i) constant or (ii) backtracking step-size, the former of which utilizes an iterative shrinkage operator $p_{L}()$ on $y_{k}$ (a linear combination of the previous 2 points $x_{k-1}$ and $x_{k-2}$) [71]. In this regard, FISTA is able to achieve an improved computational complexity of $O(1/k^{2})$ as opposed to $O(1/k^{2})$ attainable by ISTA [71]. Further details underlying the FISTA algorithm may be accessed from [71], for the interested reader.

A photomicrograph of a copepod, individually subjected to each of these wavelet deconvolution algorithms (ISTA and FISTA) as implemented within DeconvolutionLab2 (c EPFL) [17], is shown in Fig. 18:

2) MULTI-WIENER SURE-LET AND PURE-LET DECONVOLUTION

In 2013, Multi-Wiener SURE-LET Deconvolution was proposed by [72] as a new approach to image deconvolution integrating Wiener filters with undecimated Haar-wavelet thresholding so as to minimize the regularized Stein’s unbiased risk estimate (SURE) as an unbiased mean squared error (MSE) estimate (assuming Gaussian noise), was proposed by [72] and named [72]. Here, SURE (being solely dependent on empirical data) has been noted by [72] as a feasible parameter for solving linear problems, with potential use-cases including (i) parameter optimization (as in Tikhonov regularization [73], (ii) non-local means (NLM) denoising [74], (iii) monitoring PSNR increments during IST iterations [72], or (iv) being utilized as a minimization measure for denoising algorithms implementing the linear expansion of thresholds (LET) approach, leading to the integrated procedure named SURE-LET ([75], [76]).

Mathematically, SURE may be expressed by $\epsilon$ in (44) as follows:

$$\epsilon = \frac{1}{N} \left( ||f(y)||^{2} - 2y^{T}H^{-T}f(y) + 2div_{y} \left\{ CH^{-T}f(y) \right\} + ||x||^{2} \right)$$

(44)

where $N$ is the number of pixels in the image, $y$ is the convolved impulse response of the system, $H$ is a square matrix representing a linear distortion, $x$ is the uncorrupted and unknown signal, $C$ is the covariance matrix and $div_{y}$ is the divergence of $y$ [72]. Here, $\epsilon$ represents the unbiased estimate of the MSE $= \frac{1}{N} E \left\{ ||f(y) - x||^{2} \right\}$ where $E(x)$ refers to the expectation of $x$, which translates into the following expression:

$$\epsilon = \frac{1}{N} \left( ||f(y)||^{2} - 2y^{T}H^{-T}f(y) + 2\sigma^{2} div_{y} \left\{ CH^{-T}f(y) \right\} + ||x||^{2} \right)$$

(45)

with variance $\sigma^{2}$ and $C = \sigma^{2}I$ (where $I$ is the identity matrix).

Further details underlying the mathematical proof for SURE, as well as visual representations of images deconvolved using SURE-LET are discussed in [72], although generally, a significant improvement was noted in the results obtained from SURE (as expressed in [72]). Notably, the average PSNR (over 10 cycles) of the 256 $\times$ 256 House image deconvolved via SURE-LET was 25.20, as compared to other leading deconvolution methods such as ForWaRD,
which exhibited an average PSNR of 24.27 (Gaussian blurred image with variance, $\sigma = 30$) [72].

An alternative approach (named PURE-LET) was proposed by [77] in 2017 to resolve 3D epifluorescence microscopy images, which sought to extend SURE-LET to Poisson and mixed Poisson-Gaussian noise cases. Here, PURE-LET utilizes a Poisson unbiased risk estimate (hence its name, PURE) to solve a set of linear equations both rapidly and accurately, with promising results reported for different convolution kernels and noise levels [78]. Mathematically, the PURE-LET model is based on the following equation [78]:

$$y = \alpha P \left( \frac{H x}{\alpha} \right) + N \left( 0, \sigma^2 I_d \right) \quad (46)$$

where $y$ represents the distorted representation of the true $d$-dimensional image $x$, $N = \prod_{i=1}^{d} N_i$, $H \in \mathbb{R}^{N \times N^d}$ represents a convolution matrix of the PSF $h$ (where $Hx$ is used to impose a positivity constraint on the Poisson noise intensities), $P(\cdot)$ represents the effect of Poisson noise, $\alpha$ is the scaling factor controlling the strength of the noise and $\sigma^2$ is the variance of the additive-white-Gaussian-noise (AWGN) [78].

PURE-LET seeks to resolve this problem by considering 2 cases – (i) the Poisson Noise case and (ii) the mixed Poisson-Gaussian Noise case (full mathematical details are discussed in [78]). In so doing, [78] proposed defining the fundamental deconvolution functions using Wiener filtering followed by transform-domain denoising. The Haar wavelet transform was also employed by [78] following prior studies confirming its efficacy in minimizing numerous types of noise degradations [78]. Visual realization of the improvements garnered through the utilization of PURE-LET for image deconvolution (as compared to other techniques, e.g. PIDAL or GILAM) are depicted in [78], where (for instance) a 256 $\times$ 256 Galaxy image deconvolved with PURE-LET had a PSNR of 27.74 (for $\alpha = 4$) while PIDAL and GILAM facilitated deconvolutions reported a PSNR of 27.38 and 26.97 respectively. Separately, [77] has demonstrated a further use case of PURE-LET for deconvolving autofluorescence microscopical images of mature pollen grains, with a similar stellar performance reported for PURE-LET as compared to other techniques (such as ParallIterDecon and MitivDecon) [PSNR$_{\text{PURE-LET}} = 28.77$, PSNR$_{\text{ParallIterDecon}} = 26.11$, PSNR$_{\text{MitivDecon}} = 27.56$ for $\alpha = 0.5$], while being significantly faster than these methods as well (t$_{\text{PURE-LET}} = 11.01$s, t$_{\text{ParallIterDecon}} = 67.39$s, t$_{\text{MitivDecon}} = 27.29$s) [77]. An image deconvolved using PURE-LET (from [77]) is indicated in Fig. 19 below:

III. NOISE REMOVAL

Noise in microscopical imaging represents an inherent problem faced by most algorithm developers today in the never-ending quest to push the envelope of optical resolution to greater heights. Noise may arise from numerous optical components (such as the field lens, condenser and objective optics, the tube lens, the slide on which the specimen is mounted, etc), although primary sources of noise are stray background light, delaminated microscope optics (for older scopes), as well as dark current of the sensor or camera. Hence, it would be prudent to mention that out-of-focus blur (generated by optical planes other than image plane) is not considered as noise in this context, although the blurring contributes to the background signal to be removed in the deconvolution of an acquired image. A prior in-depth analysis on numerous popular cutting-edge denoising algorithms was conducted by [79], and is as highlighted in Table 2 below:

Despite the wide array of denoising algorithms holistically described in Table 2, a key emphasis in the current context is placed on other image denoising algorithms which are not surfaced in Table 2 (but commonly utilized in optical microscopy), as follows:

A. WIENER FILTERING

The Wiener filter remains one of the most popular denoising algorithms employed in the image processing workspace today, having been developed and proposed over 7 decades ago by [12]. Mathematically, the Wiener filter may be represented (in the frequency domain and assuming standard white Gaussian noise having variance $\sigma^2$) by the following equation (from [92]):

$$W = \frac{R(\omega) \cdot S_{pp}(\omega)}{|R(\omega)|^2 S_{pp}(\omega) + \sigma_n^2} \quad (47)$$

where $R(\omega)$ is the FT of the convolved input signal and $S_{pp}(\omega)$ is the power spectrum of the input projection [92].

Nonetheless, this indicates that the input projection $S_{pp}$ is unknown, necessitating a viable estimate of $S_{pp}$ to be used instead. Here, [92] proposes the use of the relation $S_{ss}(\omega) = S_{pp}(\omega) \cdot |R(\omega)|^2$, allowing (47) to be simplified into the following:

$$W = \frac{S_{ss}(\omega)}{S_{ss}(\omega) + \sigma_n^2} \cdot \frac{1}{R} \quad (48)$$

which represents 2 filters (denoising and inverse filtering) operating in tandem in the Fourier domain [92].

Wiener filtering is often employed when noise (high frequency signals) is present in the image and seeks to silence/attenuate these frequencies based on their signal-noise ratio (SNR), while simultaneously implementing image deconvolution [92]. A single image subjected to motion-blurring (linear motion: 12, angle of camera
shake: 11°) and Gaussian noise (filter size: 4, \(\sigma = 0.75\)) in MATLAB (© 1984-2019 The MathWorks, Inc.), followed by Wiener deconvolution is portrayed in Fig. 20 below:

**B. MEDIAN FILTERING**

The median filter is well-adapted to removing salt-and-pepper/granular noise in images, which often results from the amplification of dark current/background caused by using high gain settings to compensate for poor S/N ratios. Median filtering involves replacing the RGB values of a pixel with the median RGB values of its neighboring pixels, following the ranking of their intensities [93]. Fundamentally based on the intensity distribution of the neighboring pixels (rather than a single pixel), median filtering is resilient to statistical outliers, while also being relatively easy to implement with a reduced probability of blurring (which is dependent on the size of the filter kernel used) [93]. An image denoised using median filtering is shown in Fig. 21:

**C. FFT FILTERING**

FFT filtering for denoising is another approach often utilized when the noise present in the image is relatively well-distributed across the image and can be reduced through the use of a low pass filter to exclude the high frequency components corresponding to the noise [93]. In addition, FFT is also commonly employed for removing a periodic signal (contribution to an artifact) in the image [94]. The latter implementation of the FFT denoising filter MATLAB code (as discussed in [94]) on a single noisy image and its output is described by Fig. 22:

**IV. ARTIFICIAL INTELLIGENCE (AI) AND ITS ROLE IN OPTICAL NANOSCOPY**

Artificial Intelligence (AI) represents a relatively new domain to have its concepts infused into the realm of...
optical microscopy. Although AI is an encompassing concept which incorporates numerous machine learning (ML) and deep learning (DL) algorithms which may be further sub-classed into supervised, semi-supervised, unsupervised and reinforcement learning algorithms [95] and [96], a particular class of DL algorithms known as convolutional neural networks (CNNs) has been successfully deployed in computational imaging and object detection applications. Some examples of research studies and applications utilizing CNNs in this respect include [97] and [98].

In the employment of deep neural networks (DNNs) for computer vision and image processing, several factors play an essential role in the configuration and fine-tuning of such a network. These include the loss function [99], the number of epochs [100], the convolution kernel used ([101] and [102]), backpropagation, etc. Further details pertaining to each of these aspects and their roles in DNN-training for image analysis are discussed in [103], with some additional contributors (as a complement to [103]) being described in Table 3:

As DNNs represent an emerging field with an ever-increasing utility in image analytics, numerous current studies are focused on developing DNN models which seek to extend the current functionality of deconvolution algorithms for image processing. Notably, a similar trend has been observed in optical microscopical imaging, with some prominent research efforts in this respect being exemplified in [117], [118] and [119]. This is in addition to other recent in silico (albeit non-DNN) approaches (such as [120] and [121]) to achieve super-resolution nanoscopy. It would be prudent at this juncture to emphasize the potential relevance of AI-based approaches to breaching the Abbe diffraction limit (as opposed to the traditional deconvolution algorithms as discussed in Section 1 previously) since the PSF is often a variable function across the image volume, hence DNNs trained to recognize these variations would be better adapted in resolving an image. In this context, we evaluate 3 recent research efforts in the present review, with particular emphasis on their employed DNN architectures, as well as the efficacy of their proposed models in extrapolating the bounds of optical microscopy into the nanoscopy domain. Further details on each of these approaches are provided in the following sub-sections.

A. ANNA-PALM [122]

An eminent study in smart nanoscopy refers to ANNA-PALM [122] which (as its name implies) refers to the utilization of AI-based approaches to achieve super-resolution (SR) fluorescence microscopical imaging akin to that of the empirical technique known as PALM [123]. According to [122], the developers of ANNA-PALM utilized a special conditional GAN (cGAN) named A-net which was based on the pix2pix architecture and comprised of 3 ANNs as follows (from [122]):

| Table 3. Hyperparameters often associated with DNN training (adapted from [112]). |
|-----------------------------------------------|
| DNN Aspect                  | Description                                                                 |
| Optimization functions       | Optimization functions are commonly employed during the training of an ANN to minimize the computed error between the predicted values and the target values. There exist numerous examples of popular optimization functions, which may be classed into first-order, high-order, and heuristic derivative-free optimizers [104]. First-order optimizers utilize the gradient of the loss function, a popular example of which is Stochastic Gradient Descent (SGD) [104], while high-order optimizers involve the Hessian (second-order derivative) matrix $H$ which, though being more challenging caused by estimating its inverse ($H^{-1}$), provides quicker convergence through analysis of the surface curvature [104]. Heuristic derivative-free optimizers (such as coordinate descent [104]) are employed when the derivative of the objective function $f$ and its limits are inaccessible due to being difficult to compute or subject to noise [105]. Gradient descent (GD) is a mathematical process used to discover the minimum turning point in a plot of a parabolic function. GD operates by iteratively updating the input variables in the opposite direction to $f$ during back-propagation, thereby minimizing the cost function. The step size in each iteration is determined by the learning rate $\eta$ which (in turn) influences the number of iterations required to reach the desired global optimum for $f$ [106]. Mathematically, this algorithm may be represented as follows (from [104]):

$$\frac{dE}{d\theta} = \frac{1}{n} \sum_{i=1}^{n} \left( y_i - f_{\theta}(x_i) \right) x_i$$

(49)

$$\theta_{t+1} = \theta_t + \frac{1}{n} \sum_{i=1}^{n} \left( y_i - f_{\theta}(x_i) \right) x_i$$

(50)

given $\frac{dE}{d\theta} = \sum_{i=1}^{n} \left( y_i - f_{\theta}(x_i) \right)^2$ and $f_{\theta}(x) = \sum_{i=1}^{n} \theta_i x_i$ and where $E(\theta)$ is the loss function, $f_{\theta}(x)$ is the original function, $\theta$ is the optimization parameter, $n$ and $D$ are the number of training samples and input features respectively, $x$ is the independent variable and $y$ is the target output [104]. Nonetheless, up-scaling of the GD algorithm is computationally expensive and time-consuming, popularizing the use of a variant known as stochastic GD (SGD) [104]. Here, a single data point is randomly-sampled from the set at each iteration, with an alternative approach (termed mini-batch GD) sampling a subset of data points (instead of individual points) for gradient computation and update [104]. Another popular GD algorithm, called the Nesterov Accelerated Gradient (NAG), incorporates the use of momentum as a means to accelerate the process of gradient descent initially but slow it down as the error function approaches its minimum [104].

Other optimization algorithms (such as AdaGrad and AdaDelta) are also sometimes used – these are able to adapt (hence the prefix ‘Ada’) the learning rate $\eta$ based on the previous gradients of the input parameter [104]. AdaGrad however has a couple of limitations – (i) the initial value of $\eta$ has to be manually determined and (ii) the learning rate diminishes with each iteration of the algorithm due to an increasing gradient of descent, the latter being effectively resolved through imposing a constraint on the number of past learning rates to be considered in AdaDelta [104]. Another popular alternative optimizer, termed Adaptive Moment Estimation (Adam) computes and stores the adaptive learning rates for each parameter (including averages of the past squared gradients $\nu$ and past gradients $\mu$) [104]. Through this approach, Adam is able to outperform other optimizers in its convergence rate (being comparable to SGD(No) in terms of speed) while being able to adapt the learning rate range for individual layers, rather than manual selection (as in SGD) [104]. A mathematical representation of the Adam optimizer (based on the input parameter $a$, $a = 0.001$, exponential decay rates $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) is provided as follows (adapted from [106]):

$$A = \frac{1}{\sqrt{t+\epsilon}} \left( \frac{1}{\beta_1 (t+\epsilon)} \frac{\partial E(\theta)}{\partial \theta} \right)$$

$$\mu_{t+1} = \beta_1 \mu_t + (1 - \beta_1) \frac{\partial E(\theta)}{\partial \theta}$$

$$\nu_{t+1} = \beta_2 

\text{where } \beta_1, \beta_2 \text{ are the exponential decay rates, } \epsilon \text{ is a small constant to prevent division by zero, } \frac{\partial E(\theta)}{\partial \theta} \text{ is the gradient of the loss function, } \mu_t \text{ is the mean of past gradients, and } \nu_t \text{ is the variance of past gradients.}$$
TABLE 3. (Continued.) Hyperparameters often associated with DNN training (adapted from [112]).

\[ m_i = m_i - \Delta m_i + \left( 1 - \beta_i \right) \Delta m_i \]
\[ v_i = v_i - \Delta v_i + \left( 1 - \beta_i \right) \Delta v_i \]
\[ \theta_i = \theta_i - \Delta \theta_i \]

where \( m_i \) and \( v_i \) are estimates of the moments of the mean and uncentered variance of the gradients respectively [106], and (54) is the final equation for updating the input parameter \( \theta \) [106].

Nonetheless, a recent study [107] has noted current adaptive optimizers (including Adam) to exhibit poor generalization as compared to SGD and are even unable to converge when \( \eta \) is extreme and unstable. In this regard, [107] has proposed 2 other alternative adaptive optimizers (AdaBound and AMSBound) which the authors claim are immune to extreme learning rates while being able to smoothly converge to SGD (or momentum) when the time step is increased. The interested reader is thus encouraged to refer to [107] for further details.

**Normalization**

Normalization procedures are often employed in training an ANN where the input values may have different ranges. Through normalization, the output of a neuron has a distribution with zero mean and unit variance [108]. Normalization may be carried out via either (i) rescaling each value to a specific range (e.g., from 0 to 1) or (ii) by an approach known as batch normalization (BN), which increases training efficiency through smoothing the network optimization landscape [109]. BN has also been characterized to alleviate the vanishing/exploding gradient issue, while posing as a regularizer [110]. In a single perceptron (the building block of a NN), the BN algorithm may be expressed mathematically as follows (from [110]):

\[ y = \frac{y}{\sqrt{x}} + \beta \]
\[ h = \frac{y}{\sqrt{x}} \]

where \( y \) is the neuron output, \( g(\cdot) \) the activation function, \( \beta \) and \( \delta \) are hidden values before and after BN, \( w \) and \( x \) are the weight vector and input to the NN respectively, \( \mu \) and \( \sigma \) are the estimated mean and standard deviation of \( x \) respectively within an independent batch of samples, \( y \) and \( \beta \) are the scale and shift parameters respectively [110]. (54) is the final normalization equation.

Nonetheless, it would be essential to note that BN may also be prone to issues such as intra-batch dependence of sample loss and a discrepancy between the batch and network normalization parameters when using the batch for computing the mean and variance of the dataset [111]. In this regard, other approaches (such as weight normalization or layer normalization) as discussed in [111] may be utilized. In addition, the interested reader is also encouraged to explore [112] and [113] for several other normalization algorithms (such as spectral normalization, switchable normalization, etc.).

**Activation Function**

Activation functions are often employed as 'mathematical gates' regulating the outflow from an input node, either by (i) switching the neuron output on or off (depending on a defined threshold being met) or (ii) mapping the input signals (via a suitable transformation function) into output values required for proper functioning of the ANN [114]. Activation functions may be grouped into 3 main classes—(i) binary step (which switches on/off the output based on the defined threshold being met), (ii) linear and (iii) non-linear (the only type allowing backpropagation and multiple layers to be stacked to form a DNN) [114].

Currently, there exist numerous activation functions, of which 7 commonly utilized non-linear variants (as detailed in [114]) have been consolidated and outlined as follows (all graphs plotted using Desmos [115]):

| Activation Function Type | Description |
|--------------------------|-------------|
| Sigmoid / Log-sigmoid 1/x | Allows a smooth transition of output values in the range [0, 1] but is prone to vanishing gradients (when the input value is very high or low). |

**TABLE 3. (Continued.) Hyperparameters often associated with DNN training (adapted from [112]).**

1. **TanH / Hyperbolic Tangent**
   \[ f(x) = \tanh(x) \]
   Is zero-centered and similar to the sigmoid function but is also prone to the same setbacks faced by the sigmoid activation function.

2. **Rectified Linear Unit (ReLU)**
   \[ f(x) = \max(0, x) \]
   Is computationally efficient and non-linear but is unable to perform backpropagation when inputs are negative or close to zero (the dying ReLU problem).

3. **Leaky ReLU**
   \[ f(x) = \max(0, 0.1x) \]
   Has the advantages of ReLU (without the dying ReLU problem), but predictions are inconsistent when the input is negative.

4. **Parametric ReLU**
   \[ f(x) = \max(ax, 0) \]
   Allows the negative input values to be handled effectively while permitting backpropagation with all the benefits of ReLU. However, it is inconsistent for variable inputs.

5. **Softmax**
   \[ \sigma(j) = \frac{e^{x_j}}{e^{x_{k}} + e^{-x_{k}}} \]
   for \( j = 1, \ldots, k \)
   Is able to handle multi-classified variables (having several probable outputs), thereby being particularly useful for the final output layer of an ANN/DNN.

6. **Swish**
   \[ f(x) = \frac{x}{1 + e^{-ax}} \]
   Developed by researchers at Google, reportedly outperforming ReLU with similar computational efficiency. Further details of Swish are discussed in [114].

* Authors' Note: We propose that this function may be further generalized as follows:

   \[ f(x) = \frac{x}{1 + e^{-a\theta}} \]
   for some trainable parameter \( \beta \).
iii) A cGAN adversarial network $D$ (with 5 convolutional layers) which produces the loss and decides if the input image is obtained empirically or artificially-generated by $G$.

Training of the A-net was conducted using randomly undersampled (sparse) PALM images (having resolutions of $256m \times 256m$, with $m \in \mathbb{Z}$) as an input, and their associated dense PALM variants (for the desired output) [122]. The $G$ network also contains a computational selector (switch) allowing for different structures (e.g. nuclear pores or microtubules) to be resolved [122], while the error map may be defined by the following equation:

$$E_Q(A, W) = (1 - MS_{SSIM}(Q(A), W)) \times (Q(A) + W)$$  \hspace{1cm} (57)

where $E_Q$ refers to the error generated by the network $Q$, $A$ is the reconstructed SR image and $W$ is the widefield input image [122].

The DNN architecture of the A-net (further details of which are elaborated in [122]) is illustrated in Fig. 23:

According to the authors of [122], the convolutional layers in the A-net are coupled with batch normalization procedures, while the dropout layers (dropout rate = 50%) are only utilized during training but disabled during image inference [122]. The activation functions used are ReLUs $f(x) \rightarrow \text{sup}(x,0)$ or ‘leaky’ ReLUs $f(x) \rightarrow \text{sup}(x,0) + \text{inf}(\varepsilon x,0)$, where $\varepsilon = 0.2$, except the final layer of $G$ (which uses a tanh function) and the last layer of $Q$ (which uses a sigmoid function) [122]. The cGAN network in (iii) utilized least squares loss functions (being empirically superior to the log loss functions often employed in GANs [122]), which are defined mathematically as follows:

$$L_D(G, D) = E_{(c, x) \sim p_{\text{data}} (c, x)} [D(c, x) - 1]^2$$
$$+ E_{c \sim p_{\text{data}} (c)} [D(c, G(c, z)) - 1]^2$$  \hspace{1cm} (58)

$$L_{cGAN}(G, D) = E_{c \sim p_{\text{data}} (c), z \sim p_z} [1 - D(c, G(c, z))]^2$$  \hspace{1cm} (59)

Visually, a comparison of the results derived from ANNA-PALM with empirical PALM suggests a high degree of correlation between these 2 image sets (as indicated in Fig. 24), implicating the evident capability of the A-net in performing in silico nanoscopy. Nonetheless, a potential limitation of this approach refers to its mapping of learnt features to unknown images (as surfaced by [125]), thereby obscuring the detection of otherwise present anomalies, which might occur at a relatively low percentage in endogenous tissue.

B. DEEP-STORM [126]

Deep-STORM represents a then-novel approach in providing a rapid, precise way to obtain SR images from randomly-emitting fluorophores without the need to acquire any additional data inputs (e.g. the convolution PSF of an optical system, etc) [126]. As its name implies, Deep-STORM utilizes a DNN coupled with widefield epifluorescent microscopy to directly generate SR images without localizing individual emitters, unlike traditional empirical approaches (such as STORM or PALM) [126]. The CNN-based architecture of Deep-STORM comprises of an encoder (convolution) and a decoder (deconvolution) segment, the former composed of three $3 \times 3$ convolutional layers (with increasing depth) alternated with $2 \times 2$ max pooling layers, while the latter consists of $2 \times 2$ upsampling layers alternated with $3 \times 3$ convolutional layers (with decreasing depth) [126]. Batch normalization and ReLU were also incorporated within the CNN architecture (with 1.3 million nodes for training), appended with a $1 \times 1$ depth-reducing convolutional filter (having a linear activation function) for the final SR image generation [126]. Visually, the architecture of Deep-STORM may be described by Fig. 25 (from [126]):

Deep-STORM was trained using 10k pairs of simulated data points, first generated as twenty $64 \times 64$ pixel frames using the ThunderSTORM plugin [127] within ImageJ [128] and [129]. This was subsequently processed by extracting $500$ random $26 \times 26$ pixel ROIs from each image, which are then up-sampled by 8-fold, forming $208 \times 208$ pixel regions [126]. The training was done over 100 epochs (having

---

**FIGURE 23.** The A-net architecture developed by [122] and employed in ANNA-PALM [122]. A The generator network $G$; B The “low-resolution estimator” network $Q$; C The cGAN network $D$ [122]. All figures are drawn by the authors of the current review and are based on the original A-net architecture as described in [122].

**FIGURE 24.** A comparison of microtubule images obtained using Widefield, PALM and ANNA-PALM. Notice the agreement between images using ANNA-PALM with both widefield (a) and sparse PALM (b) inputs, as well as with the ground truth/dense PALM output (c). Adapted from [122].
a batch size of 16 frames) with an Adam optimizer (Gaussian kernel, $\sigma = 1$, learning rate $\epsilon = 0.001$) [126]. The loss function used for training was based on regression analysis, utilizing the squared Euclidean distance between the 2D Gaussian-convolved network’s predicted points and its corresponding ground truth, while having an $l_1$ penalizer [126]. A mathematical representation of the loss function is indicated as follows (adapted from [126]):

$$\ell (x, \hat{x}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{x}_i \otimes h_k - x_i \otimes h_k \right\|_2^2 + \left\| \hat{x}_i \right\|_1 \quad (60)$$

where $\hat{x}_i$ refers to the predicted values, $x_i$ refers to the ground truth, $h_k$ is the Gaussian kernel, $N$ is the size of the training dataset, and $\otimes$ refers to the convolution operator (adapted from [126]).

Verification of the network accuracy was performed using both empirically-obtained datasets and simulated data points (as (i) samples of quantum dots imaged under different laser powers and composites and (ii) obtained from [130] respectively). In all instances, Deep-STORM exhibited significantly-reduced run-times while maintaining the overall integrity of the SR image. A visual representation comparing the output from (and runtimes of) Deep-STORM with CEL0 [131] is shown in Fig. 26:

From the results gleaned from [126], the authors have emphasized the key strengths of Deep-STORM as being able to generate SR videos of fluorescently-labelled structures (without much intervention from the user), although they have also acknowledged that it did not provide localization-specific information of the resolved molecules, despite using this information for creating the SR image [126].

### C. IsoNet-1 and IsoNet-2 [132]

In 2017, a couple of DNN architectures (named IsoNet-1 and IsoNet-2) were proposed by [132] for restoring isotropic resolution in the axial plane (induced primarily by the severe anisotropy exhibited by the oblique PSF in the z-axis). Both IsoNet-1 and IsoNet-2 are based on a CNN architecture, with the main difference between the two being that IsoNet-1 utilizes a traditional CNN kernel-downsampling approach (with a ReLU activation function) while IsoNet-2 is based off a U-Net architecture with skip connections [124]. In the development of IsoNet, the authors of [132] considered 2 primary factors – (i) the process of formation of a stereographic volume in fluorescence microscopy and (ii) the anisotropic PSF volume, leading them to propose a DNN model based on sparsity-induced super-resolution coupled with image deconvolution. Nonetheless, [132] has also highlighted the absence of ground truth data for training their DNN, leading them to utilize the same dataset for training (a concept described as self super-resolution). In this regard, [132] has emphasized a key difference between conventional methods of image restoration and their proposed approach – the former utilizes generalized iterative signal-decoding procedures (without any information on the sample) while the latter seeks to understand the formation of the observed blurred image from its actual uncorrupted impulse response and potentially achieve the resolution attainable in the xy image plane for the z-axis, from the equation $y = S_\sigma (\hat{h} * x)$ (where $\hat{h}$ is the 3d rotated PSF and $S_\sigma$ is the downsampling function in the z-axis for a defined factor $\sigma$). This, according to [132], was done through inverse mapping from assembled lateral image ROIs, with the authors of [132] choosing to deconvolve the orthogonally-rotated PSF (in the xy-plane) by the average PSF volume (assuming a spherically-bounded PSF), which they referred to as $h_{iso}$. This method has a clear advantage of minimizing the PSNR loss as opposed to directly deconvolving the observed impulse response with a standardized
function, although one may surmise that it is founded on the assumption of a uniform PSF being distributed across the observed volume, which (in most instances) would be untrue. The results obtained by [132] seem promising, with relatively high PSNR values reported for both IsoNet-1 and IsoNet-2 – PSNR_{IsoNet-1}: 32.47, PSNR_{IsoNet-2}: 35.61 for simulated nuclei samples with a gaussian split PSF, as compared to traditional RL deconvolution (PSNR: 27.48), SRCNN (PSNR: 25.89) or the original blurred input image (PSNR: 25.84). To further verify the efficacy of their approach, [132] applied a 3D watershed algorithm on cellular boundaries, with a reported SEG of 0.913 (for IsoNet-2) as opposed to 0.742 (for the blurred input image) and 0.923 (assuming an isotropic PSF). With real data, equally promising results are depicted in [132], leading the developers of IsoNet to propose how utilizing their approach would aid in eliminating phototoxic cell damage. However, the authors of [132] acknowledged (as well) a key limitation of their approach – that a sampling rate greater than the Shannon limit is required for effective axial super-resolution.

D. DEEP-Z [133]

As a putative successor to IsoNet-1 and IsoNet-2 [132], Deep-Z was conceived by [133] as a way to generate 3D optical microscopical stacks from a 2D optical image plane. Deep-Z uses a conditional GAN (cGAN) trained with a Z-stack of 2D microscopical stacks from a 2D optical image plane. Deep-Z was conceived by [133] as a way to generate 3D optical images from 2D optical image planes. The loss functions employed were either (i) outputs from the generator G and the discriminator D or (ii) the target labels Z, with each block expressing the following mathematical transformation:

\[ z_{i+1} = LReLU \left( \text{CONV}_k \left( \text{LReLU} \left( \text{CONV}_l \left(z_i\right)\right)\right) \right) \]

where \( z_i \) refers to the input value, \( z_{i+1} \) refers to the output value (for some level \( i \)), LReLU (leaky ReLU) has a slope of 0.01, \( i_1 \) and \( i_2 \) (as subscripts of the CONV operator) refer to the number of convolutional layers [where \( i_m = 3(2^n+n+2) \) for \( m = 1 \) or 2, \( n \in \mathbb{Z}, 1 \leq n \leq 6 \)] [133].

Following the discriminator, mean pooling is used to downscale the parameter set size to 3072, appended with fully connected (FC) layers (size: 3072 \times 3072) employing leaky ReLU functions, and a single FC layer (size: 3072 \times 1) using a sigmoid activation function [133]. The convolutional kernels used are \( 3 \times 3 \) matrices with a 1-pixel stride in each dimension, except for the second CONV in (63) which has a 2-pixel stride in each dimension (for a resolution reduction factor of 2). Xavier initialization was used for the weights, with bias set to 0.1. The output (a discriminator score) lies in the range [0, 1] with ‘0’ being false and ‘1’ being true. A detailed visual representation of the Deep-Z architecture is provided in [133] and demonstrated in Fig. 27 below:

Deep-Z was trained in 2 phases – input images to the generator \( G \) were 256 \times 256 \times 2 (the second channel corresponding to the DPM) while those to \( D \) were either (i) outputs from the generator (256 \times 256) or (ii) the target \( z^{(i)} \). The loss functions employed for the generator \( L_G \) and the discriminator \( L_D \) are as follows (from [133]):

\[ L_G = \frac{1}{2N} \sum_{i=1}^{N} \left[ D\left(G\left(x^{(i)}\right)\right) - 1 \right]^2 + \alpha \frac{1}{2N} \sum_{i=1}^{N} \text{MAE}\left(x^{(i)}, z^{(i)}\right) \]

\[ L_D = \frac{1}{2N} \sum_{i=1}^{N} \left[ D\left(G\left(x^{(i)}\right)\right) \right]^2 + \frac{1}{2N} \sum_{i=1}^{N} \left[ D\left(z^{(i)}\right) - 1 \right]^2 \]

where \( N \) is the batch size, \( G(x^{(i)}) \) refers to the generator output for \( x^{(i)}, z^{(i)} \) is the target label, MAE refers to mean absolute error, \( \alpha \) (the regularization parameter for the GAN and MAE loss in \( L_G \)) = 0.02 [133].
An Adam optimizer was utilized during the training (learning rate, $\varepsilon_{LP} = 3 \times 10^{-5}$, $\varepsilon_{LG} = 10^{-3}$), with validation performed every 50 iterations and the optimal network chosen based on minimizing MAE loss. The DPM values were rescaled by a factor of 0.1, due to the refocusing distance spanning from $-10\mu m$ to $10\mu m$. On testing, the results obtained from Deep-Z bear a significantly close resemblance to the ground truth (GT) image with the difference between these 2 sets of images exhibiting a much higher structural similarity index (SSIM) and lower root mean square error (RMSE) as compared to the input image, implicating the high predictive accuracy of Deep-Z as an in silico refocusing method. Sample images indicating this correlation (obtained from [133]) are described in the following figure (Fig. 28):

According to [133], Deep-Z may be putatively extended in future applications as an initial module for neuronal imaging (reducing the need to acquire multiple image stacks), while also potentially incorporating additional acquired information at different optical sections. The advantages of Deep-Z (from our perspective) are clear and represent a well-planned step in achieving computational nanoscopy – axial resolution limitations (which are more severe than lateral resolution limits for widefield microscopy) coupled with haze (generated from out-of-focus image planes) place a severe limitation and obstacle in breaching the Abbe diffraction limit (as defined in (2) previously). Nonetheless, we also believe that Deep-Z exhibits further growth potential, especially where image corruption by noise is evident (particularly in the presence of low fluorescence signals).

V. DEEP LEARNING FOR IMAGE DENOISING

The use of DL algorithms has long found a niche for solving an ill-posed problem in image denoising, yielding comparatively better results than traditional denoising algorithms (such as Wiener or median filtering as discussed previously). Through developments in the use of ANNs for denoising (as expounded in [134] and [135]), other studies (such as [136]) have proceeded to explore using DNNs for image denoising with promising results – [136] illustrated the use of a 4 hidden-layer CNN (each layer comprising 24 nodes) coupled with a $5 \times 5$ convolution kernel to denoise images subjected to Gaussian noise with an unknown variance, producing results which surpassed traditional non-blind denoising algorithms then (CN1/CN2 PSNR = 24.12/24.25 vs BLS-GSM PSNR = 23.78, FoE PSNR = 23.02) and at much higher rates ($\sim$42 times quicker than FoE and between 68% – 133% of BLS-GSM). Elemental in this context is that [136] highlighted the need for (i) an extremely small learning rate for the final layer of the CNN (0.001) as opposed to a larger learning rate for all other layers ($r = 0.1$) and (ii) the utilization of a gradient learning algorithm to tune multiple parameters based on the input images. Similarly, [137] proposed a smart denoising algorithm termed content-aware image restoration (CARE), which seeks to denoise fluorescence microscopical images while simultaneously achieving axial super-resolution with 20-fold faster image acquisition and at 60-fold lower light intensities, although [137] also indicated CARE as being constrained by its image-specific training dataset. In this regard, it would be prudent to further evaluate the utility of other DNN models for image denoising, of which 3 such architectures are described in greater detail in the following sections. For a more elaborate discussion on the various denoising approaches, the interested reader is also encouraged to refer to [138], which seeks to provide a holistic overview of the image denoising algorithms currently available (the following 3 algorithms not being discussed in this survey).

A. MULTI-LEVEL WAVELET-CNN (MWCNN) [139]

Multi-level Wavelet-CNN (MWCNN) is based on an integration of 2D Haar wavelets with a traditional CNN architecture, as a means of optimizing the size of the receptive field while potentially reducing computational complexity [139]. Here, the MW packet transform (as the skeletal framework of MWCNN) is alternated with a convolutional block, the former involving the use of 2D DWT for splitting of the sub-band images $x_i$ into $f^i$ images for a $n$-level MW

![FIGURE 27. The CNN architecture as utilized in Deep-Z [133]. All figures are drawn by the authors of the current review and are based on the original Deep-Z architecture as described in [133].](image)

![FIGURE 28. Digitally refocused C. elegans fluoromicrographs. Notice the improved SSIM and RMSE of the $Z = -5\mu m$ and $Z = 4\mu m$ layers elucidated through Deep-Z (when compared with the GT images at these positions), as opposed to the original input image (panel a). Figure adapted from [133].](image)
The authors of [139] utilized the ADAM optimizer for training the MWCNN with the following loss function:

\[ L(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} \| F(y_i, \Theta) - x_i \|^2_F \]  

where \( \{y_i, x_i\}_{i=1}^{N} \) is the training set having \( x_i \) as the \( i \)th ground truth image and \( y_i \) as the corresponding input image, \( \Theta \) and \( F(y_i, \Theta) \) refer to the network parameters and the output of the MWCNN model respectively [139].

According to [139], application of the MWCNN algorithm reported significantly high PSNR (dB) / SSIM results for almost all assayed datasets when compared with other algorithms (e.g. 33.17/0.9357 for MWCNN vs 32.49/0.9244 for IRCNN on the Urban100 image having noise level \( \sigma = 15 \), or 32.23/0.8999 for MWCNN vs 31.85/0.8942 for DnCNN on the BSD100 image with a scale factor, \( S \) of 2) [139]. This clearly demonstrated the efficacy of MWCNN in image denoising, single image SR (SISR) and removal of JPEG image artifacts. Nonetheless, [139] also noted that the image denoising was only performed on grayscale (and not color) images, while only the luminance (\( Y \) channel) (in a YCbCr model) was processed with SISR. The measured computational efficiency of MWCNN however was comparable to other in-class algorithms, exhibiting a run-time of 0.3575s for denoising a 1024 × 1024-pixel image, as compared to 15.77s for a similar image denoised by RED30, although DnCNN and TNRD depicted run-times of 0.168s and 0.116s respectively [139]. A similar trend was observed for SISR (0.3167s for MWCNN vs 25.23s for DRRN or 0.1411s for LapSRN) and JPG artifacts removal (0.2931s for MWCNN vs 14.69s for MemNet or 0.095s for TNRD), all of which were performed on 1024 × 1024-pixel images [139]. In all these instances, non-bulk image processing pointed to the potential usage of MWCNN for optimal post-processed image quality, although batch processing may favor other comparable alternatives.

In addition to the above, [139] had also compared the use of different wavelets [traditional Haar, Daubechies-2 (DB2) as well as an amalgamation of Haar in the contracting arm and DB2 in the expanding arm] embedded within their proposed MWCNN framework, coupled with additional frameworks such as WaveResNet and deep convolutional framelets (DCF). The findings from these experiments indicate (i) the significant impact of mismatched pixel information caused by binning of incongruent pixels (termed ‘the gridding effect’ [139]) on image restoration, (ii) that a sum (rather than a concatenation) operator should be deployed in the U-Net architecture of the MWCNN and (iii) the preferable use of Haar wavelets throughout the U-Net architecture (as compared to DB2 or Haar/DB2) for optimal image quality (denoted by the PSNR) and efficiency (run-time) of the MWCNN model [139].

**FIGURE 29.** The U-Net architecture used as a skeletal framework for development of the MWCNN algorithm (as discussed in [139]). The yellow-filled box implements the convolution kernel followed by a batch normalization (BN) and ReLU activation for 4 times in all layers, except for the penultimate layer, where only convolution is applied. Figure adapted from [139].
the convolutional layer and the sub-sampling layer equations (the former incorporating a 3D convolution function nested within a traditional ReLU activation) denoted as follows:

\[ q_j = \max \left( 0, b_j + \sum_{i} k_{ij} \otimes p_i \right) \quad (67) \]

\[ q_i^{r,c} = \max_{0 \leq m, n < s} \left\{ p_i^{r,s+m,c,s+n} \right\} \quad (68) \]

where \( b_j \) is the bias term, \( k_{ij} \) is the convolution kernel (\( \otimes \) represents the 2D convolution operator), \( q_i^{r,c} \) is the \( (r,c) \) pixel in the \( i \)th output subsampled feature map, \( p_i \) is the \( i \)th input feature map (having a size of \( s \times s \)) (adapted from [140]).

A visual description of the DNN architecture is portrayed in Fig. 30 below:

Subsequently, a softmax activation function coupled with SGD having an update rule \( z_{l+1} = 0.9z_l - 0.0005bw_l - \beta \cdot \left( \frac{\delta L}{\delta w_l} \right) \) where \( w_{l+1} = w_l + z_{l+1} \) was used to refine the weights at each node, allowing the DNN to distinguish noise from signal impulses with the proposed method demonstrating significant improvement in image denoising – for e.g., in the House (256 × 256) image, the proposed method exhibited a PSNR\(_{\sigma=10}\) = 37.53 and a PSNR\(_{\sigma=70}\) = 28.96, as compared to NLFMT (PSNR\(_{\sigma=10}\) = 34.87 and PSNR\(_{\sigma=70}\) = 25.62), although [140] acknowledged that NLFMT did not generate artifacts. Another assayed method (BM3D) was also unable to surpass the performance achieved by the proposed method (BM3D: PSNR\(_{\sigma=10}\) = 36.71 and PSNR\(_{\sigma=70}\) = 27.91), although BM3D reportedly executed \( \sim 4 \) times more rapidly than the proposed method for the Lena and Barbara images (\( t_{\text{Lena}} = 6.54s \) and \( t_{\text{Barbara}} = 6.12s \) for BM3D, as compared to \( t_{\text{Lena}} = 28.79s \) and \( t_{\text{Barbara}} = 26.49s \) for the proposed algorithm) respectively. Nonetheless, the authors of [140] highlighted the inability of the presently-proposed algorithm for denoising ultrasound and color images, which they intend to incorporate in future versions of their method.

C. MP-DCNN [141]

Developed by the authors of [141], MP-DCNN represents a means of achieving image denoising without edge-induced artifacts in a substantially noisy environment through the employment of a joint loss function in a CNN [141]. Here, MP-DCNN utilizes a 3 × 3 convolution kernel in order to maximize the dimensions of the receptive field to \( (2f+1)^2 \), where \( f \) refers to the CNN depth [141]. This serves to optimize the chances of recovering pixels which have a significantly low S/N ratio. Visually, the architecture of the MP-DCNN network may be described by Fig. 31 below:

The MP-DCNN architecture consists of 10 convolution layers (and an output layer), the first 5 of which utilize the leaky ReLU activation function and have feature map outputs numbering \( (32, 32, 64, 64, 128) \) respectively, while the next 4 convolutional layers use a ReLU activation function, with feature maps numbering \( (64, 64, 32, 32) \) respectively [141]. The final convolution layer uses a 1 × 1 convolution kernel to generate \( c \) feature maps, with an added residual unit being employed to speed up the process while improving the overall network performance [141].

The authors of [141] have utilized a couple of loss functions within their developed MP-DCNN network architecture (namely MSE and perceptual loss) combined into a joint loss function (i.e. \( L_{\text{joint}} = L_{\text{MSE}} + \lambda L_{\text{SegNet}} \)) with each of these component losses (\( L_{\text{MSE}} \) and \( L_{\text{SegNet}} \)) being defined in (69) and (70) as follows (from [141]):

\[ L_{\text{MSE}} = \frac{1}{h} \sum_{i=1}^{n} \left( \frac{1}{w \times h} \sum_{j=1}^{w} \sum_{k=1}^{h} \left\| f_i(j, k) - X_i(j, k) \right\|^2 \right) \quad (69) \]

\[ L_{\text{SegNet}} = \frac{1}{w_i h_i} \left\| C_i(Y - R(Y) - C_i(X)) \right\|^2 \quad (70) \]

where \( f \) is the denoised output image, \( X \) is the input image, \( n \) is the sample size, \( w \) and \( h \) are the dimensions (width and height) of the sample image respectively [141].

The results of implementation of MP-DCNN proved promising, with the former depicting relatively superior results when compared to other image denoising algorithms such as WNNM, TNRD and BM3D, amongst others (PSNR values for the BSD68 dataset when utilizing MP-DCNN\(_{\sigma=15}\) = 32.05 \( \triangleright \) BM3D\(_{\sigma=15}\) = 31.07 or TNRD\(_{\sigma=15}\) = 31.42, MP-DCNN\(_{\sigma=50}\) = 26.68 \( \triangleright \) WNNM\(_{\sigma=50}\) = 25.87 or DnCNN\(_{\sigma=50}\) = 26.23, etc) [141]. In addition, structural similarity (SSIM) comparisons between MP-DCNN with other image denoising algorithms reveal a similar trend – MP-DCNN clearly outperforms even its closest assayed predecessor (DnCNN) with more significant improvement margins for noisier datasets – SSIM values for the BSD68 dataset when utilizing MP-DCNN\(_{\sigma=15}\) = 0.8829 \( \triangleright \) DnCNN\(_{\sigma=15}\) = 0.8826, while MP-DCNN\(_{\sigma=50}\) = 0.7104 \( \triangleright \) DnCNN\(_{\sigma=50}\) = 0.7076, with a similar stellar performance reported for MP-DCNN when using other test images [141]. In addition, the developers of MP-DCNN have also found their proposed algorithm to be much more efficient than most other tested algorithms – the run-time of MP-DCNN for a 1024 × 1024 image was 10.9 ± 0.18 s, as compared to 12.1 ± 0.2s for DnCNN [141]. All these
results reported by [141] seemingly suggest the superiority of MP-DCNN over other tested algorithms (such as DnCNN, etc), especially in denoising highly noisy images, although the authors of [141] also acknowledged potential areas for development of their proposed framework (i.e. MP-DCNN) by addressing non-Gaussian noise models, denoising large image datasets, as well as color images and videos [141]. In our perspective, one may also consider evaluating how different activation functions (such as leaky ReLU or sigmoid) may also be utilized in the MP-DCNN framework (particularly in its 6th–9th convolutional layers) rather than the traditional ReLU presently expounded in [141].

VI. COMMERCIAL APPLICATIONS OF AI-BASED DEHAZING / DEBLURRING ALGORITHMS

Although traditional deconvolution algorithms (e.g. Nearest Neighbors, Richardson-Lucy deconvolution, etc) have long been harnessed in the commercial sector within applications such as Huygens Professional (Scientific Volume Imaging B.V.) [23], Imaris 9.3 (© Oxford Instruments 2019) [142] and AutoQuant X3 (© Media Cybernetics, Inc.) [24] amongst others, the recent emphasis on the infusion of AI in optical nanoscopy has spurred some organizations to develop imaging applications utilizing such intelligent nanoscopy techniques, which are the subject of this review and discussed in the subsequent sub-sections. Hence, although the commercial implementation of conventional deconvolution algorithms is not being discussed here, the interested reader is encouraged to explore the widespread uses of these algorithms by visiting these manufacturer websites directly.

A. ADAPTIVE DECONVOLUTION AND LIGHTNING

Adaptive deconvolution (AD) represents a novel approach in computational super-resolution and image deconvolution, through utilizing a decision mask to automatically extract deconvolution parameters for individual voxels within the imaging volume [143]. This eliminates the need for the user to provide a PSF/blurring kernel manually, either by computation or determined empirically through supplying images of sub-diffraction-sized beads (as is utilized in NBD algorithms). AD is integrated as an optional deconvolution method within LIGHTNING (© Leica Microsystems), combining the speed and efficiency derived from parallel GPU processing with the decision mask (used in AD) to provide a fully automated, voxel-specific super-resolved image [143] with near-real-time efficiency, thereby seeking to extend 4 of the 6 vertices of the imaging octahedron exemplified in Fig. 32:

A comparison between AD and traditional/classical deconvolution is depicted in Fig. 33 (adapted from [143]):

![FIGURE 33. A Traditional deconvolution; B LIGHTNING workflows. Notice the additional decision mask present in LIGHTNING, which is absent in classical deconvolution processes. Figures adapted with permission from [143], © Leica Microsystems, 2018.](image)

The initial pre-processing step of AD involves determining the background ($B_g$) where the signal-noise ratio (SNR) is computed by approximating the grayscale values $g(x, y)$ for each pixel based on its neighborhood and a smoothing kernel $f_b$ as follows:

$$SNR = S[g(x, y), f_b(x, y)] / N \quad (71)$$

where $N = N[g(x, y)]$ and $B_g = b(x, y) \propto \max[b_{\text{global}}(\text{SNR})] \quad [143]$.

A general assumption here is that the signal is minimally impacted upon by noise, agreeing with well-defined image processing algorithms [143]. In the next stage a decision mask is created from individual voxels using the SNR and $B_g$ values obtained for each voxel and an image quality map generated for the entire image as shown in Fig. 34 (from [143]):

The image quality is then used to compute the adaptation coefficient (regularization parameter) in an inverse fashion for use in deconvolution in the following step [143].

The third (and final) stage in AD involves actual deconvolution of the image using the regularization parameters obtained from the decision mask previously and the MAP (maximum a posteriori)-based deconvolution algorithm modified according to the imaging method used (e.g. STED, multiphoton, etc) and fine-tuned to the optics used by Leica Microsystems [143]. The iteration count for the MAP-based
deconvolution algorithm is also determined automatically, with the iterations being ceased when there is no significant differences between the key features of the images at the $(n + 1)^{th}$ iteration as compared to the $n^{th}$ iteration. As such, this automation uniquely characterizes LIGHTNING from traditional deconvolution, with the differences between the outputs of these procedures distinctly visualized in Fig. 35.

For the accustomed user, LIGHTNING also incorporates non-AD (i.e. traditional) deconvolution algorithms, which utilize a globally determined set of parameters for image deconvolution as compared to AD (which computes local voxel-based parameters) [143]. Nonetheless, [143] has demonstrated LIGHTNING to be able to achieve a resolution of up to 120nm (accurate as of Sep 2018, with improved algorithms being able to further enhance the effective resolution of the image) as shown in Fig. 36 below:

**B. COMPUTATIONAL CLEARING AND THE THUNDER IMAGER BY LEICA MICROSYSTEMS GMBH**

An alternative contrast-enhancement methodology also developed by Leica Microsystems, computational clearing (CC) is utilized in Leica’s THUNDER Imagers to remove haze (originating from out-of-focus optical planes) while preserving focused features of interest [144] in the image plane. In this respect, CC is particularly adapted to dehazing widefield microscopical images (particularly widefield epifluorescence (WE) microscopy), where the absence of a pinhole allows light rays emerging from different optical sections to interfere with the image plane as described in Fig. 37:

Here, [144] describes a widefield epifluorescent image $I(r)$ to be estimated mathematically as follows:

$$I(r) \approx psf_{\text{if}}(r) \ast f(r) + psf_{\text{of}}(r) \ast f(r)$$

where $f(r)$ is the fluorophore intensity distribution and $psf_{\text{of}}(r)$ and $psf_{\text{if}}(r)$ refers to the PSF of the in-focus and out-of-focus light rays [144].

However (and as specifically mentioned in [144]), the PSF of the out-of-focus blur has a wider lateral spread and
Full Width at Half Maximum (FWHM) than the PSF of the in-focus signal, allowing scale discriminating operations (such as wavelet transforms) to be effectively utilized in this context. This formed the basis for an iterative algorithm to be constructed, which may be expressed mathematically as follows (from [144]):

\[
I_{\text{out}} = \text{argmin}_{I_{\text{out}}} \left[ \| I - \hat{I}_{\text{out}} \|^2 \right]
\]  

(73)

where the structural length scale (i.e. the lateral (full-width) distance at \(\sim 13.5\%\) of the maximum (peak) intensity) of the predicted out-of-focus blur \(I_{\text{out}}\) (referred to as \(S [I_{\text{out}}] > r_0\) (the LAS X-defined “feature” scale) [144].

In this respect, CC is used to reveal (rather than generate) a deblurred image, allowing the original relative intensities of the key features in the image to be preserved [144]. A CC-processed image is thus depicted in Fig. 38 as follows:

Due to their intensity conservation of the individual features, CC-processed images may be further utilized for quantitative analysis (e.g. in colocalization microscopy) although proper calibration protocols and care has to be exercised when attempting such procedures. Further details on quantifying CC-images are discussed in [144], which cites relatively close correlations between the signal intensity distributions of the fluorescent sample and its background both before and after application of CC (also termed Instant CC/ICC), with a Kolgomorov-Smirnov distance of 0.05 ± 0.02 [144]. This clearly depicts the ability of CC to allow direct fluorescent quantitation without the need for implementation of additional background removal algorithms [144].

An alternative mode of CC in THUNDER, termed Large Volume CC (LVCC) which combines Adaptive Deconvolution (AD) with CC, has also been noted in [144] to allow imaging up to a depth of 140 to 150 \(\mu\)m, as compared to traditional widefield imaging experiments (which can only resolve up to a depth of 50\(\mu\)m). Such deep field imaging allows LVCC to be used with thicker samples (negating the need for microtomy in some instances) although it should be emphasized that the maximum imaging depth depends very much on the refractive index of the sample and its light-scattering ability [144]. Further details on LVCC imaging are discussed in [144], with a figure extracted from [144] (indicating the utility of LVCC in neuronal imaging) being depicted in Fig. 39:

Finally, a third implementation of CC in THUNDER named Small Volume CC (or SVCC) is similar to LVCC, but which is intended for imaging thin (\(\sim 80\mu\)m) samples. Here, [144] has indicated how SVCC may be used to improve the resolution of an acquired image, by utilizing a single 40nm-diameter bead and achieving resolution improvements of up to \(\sim 2\) times laterally (FWHM\(_{\text{Raw}}/\text{SVCC} = 1.961\)) and \(>2.5\) times axially (FWHM\(_{\text{Raw}}/\text{SVCC} = 2.5641\)) [144]. Nonetheless, [144] highlights that these improvements in resolution were measured using the size of the spot corresponding to the image bead, instead of the actual distance between 2 structures separated by these distances (which was deemed by [144] to be not empirically possible).

C. TOPAZ LABS AI-BASED IMAGE ENHANCEMENT

The Topaz Labs AI Bundle (comprising of Topaz Sharpen AI, Topaz DeNoise AI, Topaz JPEG to Raw AI, Topaz Gigapixel AI, Topaz Mask AI, Topaz Studio 2 and Topaz Adjust AI) (\(\epsilon\) Topaz Labs) [145] represents a recent commercial software innovation in the macrophotography domain which has garnered significant interest due to its ability to produce eye-catching and vibrant images. For comparison purposes, a single image processed using Adjust AI, Sharpen AI and Gigapixel AI is being shown in the following diagram (Fig. 40):

From the above, it can be observed that Adjust AI (\(\epsilon\) Topaz Labs) is particularly adept at dehazing an acquired image, coupled with hue and contrast enhancement of the deblurred image. This may be presumed to be akin to traditional deblurring and deconvolution algorithms, such as Wiener filtering. Similarly, Gigapixel AI (\(\epsilon\) Topaz Labs) seeks to recover lost signal information caused by zooming into an image.
(an aspect of digital magnification) allowing a more highly resolved image to be generated. On a different note, Sharpen AI (ε Topaz Labs) [146] utilizes an ANN trained with millions of images to recall and generate a sharpened image from an out-of-focus image [147]. Each of these approaches represent novel conceptions in the field of AI-mediated image processing, which may (upon further improvements) find some relevance in biomedical and geological research in the near future. In particular, the DNN within Sharpen AI could be subjected to transfer learning for the infusion of new learnt features (potentially from inherent optical aberrations in the Fourier domain) to enhance its capability in facilitating biomedical research.

**VII. CURRENT LIMITATIONS AND POTENTIAL FUTURE ADVANCEMENTS IN COMPUTATIONAL NANOSCOPY**

Despite there having been numerous recent developments in the field of optical nanoscopy, extensive unexplored terrains in this respect exist, being mediated through the utilization of *in silico*-based approaches combined with existing optical microscopical techniques and enhancements. In particular, one such dilemma refers to noise coupled with haze, and while significant progress has been made with the introduction of AI-based concepts and DNN architectures in microscopical imaging, the road towards a total recovery of the spatial information contained in the image remains relatively unexplored. Imperatively, it would be crucial to understand the basis of signal impulses constituting noise – that of ultra-high frequencies unresolvable by the microscope objective lens amalgamating to form a blurred signal. Resolution of the individual components of noise (and thus their discernment) for elimination of ‘true noise’ (arising from aberrations in the optical train) as opposed to *pseudo*-noise (due to sub-resolved structures in the image) would thus be crucial in fueling the drive towards super-resolved diascopic nanoscopy. In this respect, the reader would be encouraged to pursue a true definition of image noise and its causative factors, developing potential filters to discern and selectively eliminate ‘true noise’, while attempting to resolve *pseudo*-noise.

Another aspect worthy of further investigation refers to the resolution of Fourier transform waveforms (when coupled with AI) as a means of minimizing the mapping of learnt features from previous super-resolved images (during the training of the DNN) to novel images during deployment. This would be particularly favorable in light of the fact that DNNs have been found to be computationally superior to ANNs [148] due to the multiple paths to the output layer (mathematically represented by the exponential growth factor ∏ni, where n refers to the number of nodes in the layer i) which consequentially permit higher resolutions of input feature deviations to be realized in output actualization while also accounting for spatiotemporal variations in the PSF during live-cell imaging. An exemplification of this aspect is currently being investigated by our research team as a potential approach to achieving true super-resolution microscopy via *in silico* methods. It would be prudent to mention that this may also be supplemented through existing studies conducted in digital holographic microscopy (DHM) [149] and tomography [150], which are seeking to explore the potential of phase-shifted tomograms to reveal previously-unidentifiable information.

A third possibility for future research in this domain refers to how super-resolution nanoscopic approaches may be integrated into coherent Raman spectroscopy (CRS), as highlighted by [151]. Here, [151] states that current Raman scattering detection capabilities are less sensitive than those of fluorescence, even when utilizing coherent laser sources, posing a major hurdle to be overcome in this respect. Other avenues currently being explored in this domain (with promising future advancements) include super-resolved magnetic resonance imaging (MRI) microscopy [152], nanoscale image resolution via defocused Z-stack acquisition [153], *in-situ* PSF retrieval for 3D single-molecule localization microscopy (SMLM) imaging [154] and 3D super-resolution time-lapse microscopy [155], all of which represent a push towards computational super-resolution in advanced medical imaging processes. Complementarily (& although not being the subject of this review), optical (non-computational) super-resolution microscopy has experienced a leap ahead as well, with the introduction of microspheres for non-fluorescence nanoscopy applications (see [156] for details).

On the andragogical front, a merger of augmented reality (AR) or virtual reality (VR) with nanoscopy would bridge the nanoscopy-immunology gap while being particularly relevant in scientific education (as discussed in [157] and [158]). Here, researchers (both new and experienced) would find such immersive experiences particularly helpful in understanding key processes with the potential to even uncover insights (such as biomolecular interactions) that were previously unfathomed.

Computational nanoscopy would also find a niche (which is in fact being currently explored) in pathological examinations and diagnostics, especially in the detection of cancerous tissue or diseased cells (which would often necessitate human expertise). Some studies expounding research conducted in this field include [159], [160] and [161]. This would potentially also lead to the future (presently unexplored) terrain of *non-fluorescent in vivo* optical nanoscopy and molecular imaging, for the visualization (and potentially nano-manipulations) of molecular motions, metabolic pathways and dynamics of biochemical reactions in real-time and *in situ*.

In the manufacturing sector too, computational nanoscopy also holds promise in nanomanufacturing processes and their corresponding quality analysis procedures (especially for silicon wafers and chips utilized in the semiconductor industry). With rapid advancements made in computing hardware and future CPU fabrication processes reaching single-digit nm dimensions (such as Intel’s Meteor Lake architecture using 7nm chips [162]), it would be sensible to explore the utilization of computational nanoscopy in
this sector and how it may facilitate the execution of such processes.

The following infographic (Fig. 41) summarizes some of these aspects for easier identification:

Nonetheless, most of these algorithms are plagued with numerous limitations arising as a consequence of imperfections in the optical train. A prominent problem surfaced by [163] refers to what [163] deemed as the “hallucination problem”, although we presume that this may not be a significant concern in computational nanoscopy, especially where an extensive dataset (comprising varied samples which addresses the issue being investigated) is utilized, except where highly specific post-acquisition image processing and analysis protocols (such as feature detection and object tracking) are desired, in which instance, an over-fitted DNN (prone to such hallucination issues) may be utilized. In addition, fuzzy logic approaches (such as that employed in [164], [165] & [166]) may also contribute to network hallucination (despite outperforming even the state-of-the-art super-resolution methods as mentioned in [166]), since fuzzy logic approximates the mapping of image patches to highly-resolved images, while this approximation may (at times) result in incorrect image mappings. In this respect, it would also be noteworthy to mention that future approaches in computational super-resolution microscopy should seek to integrate both deconvolution and deep learning principles, by using deep learning to accurately discern the optical PSF and noise of an optical system, which may vary spatiotemporally across the sample, especially in living cells.

The elucidated PSF may then be coupled to a deconvolution method for obtaining the impulse response at individual locations in the sample, with pixel-wise accuracy. Only by doing so, can one achieve true super-resolution computational nanoscopy, which would be immune to network hallucinations and optical aberrations, amidst a variable PSF. This may also be fueled through the advent and recent on-going developments in the field of quantum computing (QC) [167], which utilizes advanced future-cutting edge hardware to drive computational nanoscopy to unprecedented heights.

VIII. CONCLUSION

Although we have sought to holistically evaluate the recent developments in the field of computational nanoscopy in the present review, recent developments spurred by ongoing research (particularly in artificial intelligence and deep neural networks) are resulting in new findings in the field of computational nanoscopy even at the time of authoring this review. Hence, the field of intelligent computational nanoscopy represents an exciting new phase in the future of optical microscopy, as we enter an era merging traditional microscopical imaging modalities with recent advancements in machine learning and computing, such as QC. It is desired (through the readership of this review) that one would be inspired to further explore and potentially contribute to future advancements in this sector, spurring the advancement of optical nanoscopy to greater (and previously unfathomed) heights in scientific research and visualization.

ACKNOWLEDGMENT

The authors would like to thank the respective individuals (whose articles the authors have reviewed) for kindly consenting to them reusing their published images in the present review, so as to provide a holistic coverage of the developments in the field (where possible). In addition, the authors would also like to thank this appreciation to the reviewers of IEEE Access and the respective personnel of Leica Microsystems who have reviewed and suggested some changes to certain sections of the initial manuscript (which we have implemented in the current article).

REFERENCES

[1] J. S. Silfies, S. A. Schwartz, and M. W. Davidson. *The Diffraction Barrier in Optical Microscopy*. Nikon Instruments Inc. Accessed: Nov. 11, 2019. [Online]. Available: https://www.microscopyu.com/techniques/super-resolution/the-diffraction-barrier-in-optical-microscopy
[2] K. D. Hauch and B. D. Ratner, “Microscopy for biomaterials science.” in *Biomaterials Science (An Introduction to Materials in Medicine)*, 3rd ed. B. D. Ratner, A. S. Hoffman, F. J. Schoen and J. E. Lemons, Eds. New York, NY, USA: Academic, 2013, pp. 677–692.
[3] M. W. Davidson. *Numerical Aperture*. Nikon Instruments Inc—MicroscopyU. Accessed: Nov. 17, 2019. [Online]. Available: https://www.microscopyu.com/techniques/super-resolution/nuclear-aperture
[4] K. N. Fish, “Total internal reflection fluorescence (TIRF) microscopy.” *Current Protocols in Cytometry*, vol. 50, no. 1, p. 12, Oct. 2009.
[5] (2011). *Super-Resolution Microscopy Tutorial*. The University of Utah—Advanced Microscopy. Accessed: Nov. 17, 2019. [Online]. Available: https://advanced-microscopy.utah.edu/education/super-res/
[6] *The Nobel Prize in Chemistry 2014*. NobelPrize.org. Accessed: Apr. 24, 2019. [Online]. Available: https://www.nobelprize.org/prizes/chemistry/2014/press-release/
[7] S. W. Hell and J. Wichmann, “Breaking the diffraction resolution limit by stimulated emission: Stimulated-emission-depletion fluorescence microscopy,” Opt. Lett., vol. 19, no. 11, pp. 780–782, 1994.

[8] F. C. Zanacchi and A. Diaspro, “Ground state depletion imaging,” in Encyclopedia of Biophysics, G. C. K. Roberts, Ed. Berlin, Germany: Springer, 2013.

[9] U. Endesfelder and M. Heilmann, “Direct stochastic optical reconstruction microscopy (dSTORM),” in Advanced Fluorescence Microscopy: Methods in Molecular Biology (Methods and Protocols), vol. 1251, New York, NY, USA: Humana Press, 2014, pp. 263–276.

[10] J. Siebenmorgen, Y. Novikau, R. Wolleschensky, K. Weisshart, and D. Perrone and P. Favaro, “A clearer picture of total variation blind deconvolution,” IEEE Signal Process. Mag., vol. 27, no. 6, pp. 370–375, Mar. 1998.

[11] R. Prevedel, Y.-G. Yoon, M. Hoffmann, N. Pak, G. Wetzstein, S. Kato, and A. Vaziri, “Simultaneous whole-animal 3D imaging of neuronal activity using light-field microscopy,” Nature Methods, vol. 11, no. 7, pp. 727–730, Jul. 2014.

[12] L. C. Zsitney and A. Diaspro, “Deblurring Images Using the Lucy-Richardson Algorithm,” IEEE Trans. Image Process., vol. 7, no. 5, pp. 1–8, Dec. 1998.

[13] D. Chang and C.-W. Wang, “Total variation blind deconvolution,” IEEE Trans. Image Process., vol. 7, no. 3, pp. 73–80, 2008.

[14] H. Li, C. Guo, D. Kim-Holzapfel, W. Liu, Y. Alshuller, B. Schroeder, M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, “Light field microscopy,” ACM Trans. Graph., vol. 25, no. 3, pp. 797–806, Jul. 2006.

[15] K. Sastry, (Apr. 2002). Iterative Constrained Deconvolution Methods. Piscataway, NJ: IEEE Press. [Online]. Available: https://atol.lylver.org/AstroSurf/PDF/interactive_cons_decon_rev102.pdf

[16] M. H. Loew, “Application of image restoration (Jansson Van-Cittert) to simultaneous whole-animal 3D imaging of neuronal activity using light-field microscopy,” Planetary Sci. Lett., vol. 27, no. 22, pp. 31644–31666, 2009.

[17] K. Srivastava, (Apr. 2002). Iterative Constrained Deconvolution Methods. Piscataway, NJ: IEEE Press. [Online]. Available: https://atol.lylver.org/AstroSurf/PDF/interactive_cons_decon_rev102.pdf

[18] J. Opt. Soc. Amer. B, Opt. Image Sci., vol. 27, no. 2, pp. 295–302, 11 2010.

[19] C. Vonesch, and M. Unser, “DeconvolutionLab2: An open-source software for deconvolution microscopy,” in Proc. ICASSP, IEEE Int. Conf. Acoust., Speech, Signal Process., vol. 6, Apr. 1981, pp. 47–49.

[20] A. Stefanou, J. Page, P. Symvoulidis, G. G. Westmeyer, and T. Lasser, “Artifact-free deconvolution in light field microscopy,” Opt. Expr., vol. 27, no. 22, pp. 31644–31666, 2019.

[21] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, “Light field microscopy,” ACM Trans. Graph., vol. 25, no. 3, pp. 924–934, 2006.

[22] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, “Light field microscopy,” ACM Trans. Graph., vol. 27, no. 5, pp. 1–8, Dec. 2008.

[23] L. C. Zsitney and A. Diaspro, “Deblurring Images Using the Lucy-Richardson Algorithm,” The MathWorks, Inc. Accessed: Dec. 1, 2019. [Online]. Available: https://www.mathworks.com/help/images/deblurring-images-using-the-lucy-richardson-algorithm.html

[24] T. Schrödel, R. Raskar, M. Zimmer, E. S. Boyden, and A. Vaziri, “Simultaneous whole-animal 3D imaging of neuronal activity using light-field microscopy,” Biomed. Optics Express, vol. 10, no. 1, pp. 29–49, 2019.

[25] K. Srivastava, (Apr. 2002). Iterative Constrained Deconvolution Methods. Piscataway, NJ: IEEE Press. [Online]. Available: https://atol.lylver.org/AstroSurf/PDF/interactive_cons_decon_rev102.pdf

[26] D. Chang and C.-W. Wang, “Total variation blind deconvolution,” IEEE Trans. Image Process., vol. 7, no. 6, pp. 73–80, 2008.

[27] H. Li, C. Guo, D. Kim-Holzapfel, W. Liu, Y. Alshuller, B. Schroeder, M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, “Light field microscopy,” ACM Trans. Graph., vol. 25, no. 3, pp. 924–934, 2006.

[28] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, “Light field microscopy,” ACM Trans. Graph., vol. 27, no. 5, pp. 31644–31666, 2019.

[29] R. Prevedel, Y.-G. Yoon, M. Hoffmann, N. Pak, G. Wetzstein, S. Kato, T. Schrödel, R. Raskar, M. Zimmer, E. S. Boyden, and A. Vaziri, “Simultaneous whole-animal 3D imaging of neuronal activity using light-field microscopy,” Nature Methods, vol. 11, no. 7, pp. 727–730, Jul. 2014.

[30] L. C. Zsitney and A. Diaspro, “Deblurring Images Using the Lucy-Richardson Algorithm,” The MathWorks, Inc. Accessed: Dec. 27, 2019. [Online]. Available: https://www.mathworks.com/help/images/deblurring-images-using-the-lucy-richardson-algorithm.html

[31] J. Huang and D. Mumford, “An iterative technique for the rectification of observed distributions,” Astronomical J., vol. 79, p. 745, Jun. 1974.

[32] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Efficient marginal likelihood optimization in blind deconvolution,” in Proc. CVPR, Jun. 2011, pp. 2657–2664.

[33] D. Strong and T. Chan, “Edge-preserving and scale-dependent properties of total variation regularization,” Inverse Problems, vol. 19, no. 6, pp. 5165–5187, Dec. 2003.

[34] J. Huang and D. Mumford, “A statistical interpretation of edge detection,” Ann. Statist., vol. 29, no. 1, pp. 1–48, 2001.

[35] Y. L. You and M. Kaveh, “Anisotropic blind image restoration,” in Proc. 3rd IEEE Int. Conf. Image Process., Sep. 1996, pp. 461–464.

[36] J. Huang and D. Mumford, “Statistics of natural images and models,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 1999, pp. 541–547.

[37] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Sharing parameters in blind deconvolution algorithms,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 12, pp. 2354–2367, Dec. 2011.

[38] D. Strong and T. Chan, “Edge-preserving and scale-dependent properties of total variation regularization,” Inverse Problems, vol. 19, no. 6, pp. 5165–5187, Dec. 2003.

[39] J. Huang and D. Mumford, “A local extremes, runs, strings and multiresolution,” Ann. Statist., vol. 29, no. 1, pp. 1–48, 2001.

[40] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Efficient marginal likelihood optimization in blind deconvolution,” in Proc. CVPR, Jun. 2011, pp. 2657–2664.
[53] N. Wagner, N. Norlin, J. Gierten, G. de Medeiros, B. Balázs, J. Wittbrodt, L. Hufnagel, and R. Prevedel, “Instantaneous isotropic, volumetric imaging of fast biological processes,” *Nature Methods*, vol. 16, no. 6, pp. 497–500, 2019.

[54] E. H. Adelson and J. Y. A. Wang, “Single lens stereo with a plenoptic camera,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 99–106, Feb. 1992.

[55] R. Ng, C. K. Chui, and M. Unser, “Ten good reasons for using spline wavelets,” in *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, Jan. 2006.

[56] R. Rubinstein, P. Peleg, and M. Elad, “Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model,” *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 661–677, Feb. 2013.

[57] M. Broxton, L. Grosenick, S. Yang, N. Cohen, A. Andalman, K. Deisseroth, and M. Levoy, “Wave optics theory and 3-D deconvolution for the light field microscope,” *Opt. Expr.*, vol. 21, no. 21, pp. 25418–25439, 2013.

[58] J. Li, F. Luisier, and T. Blu, “PURE-LET image deconvolution,” *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 92–105, Jan. 2018.

[59] W. Meinelt, J.-C. Olivo-Marin, and E. D. Angelini, “Denoising of microscopy images: A review of the state-of-the-art, and a new sparsity-based method,” *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3842–3856, Aug. 2018.

[60] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Phys. D: Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, Nov. 1992.

[61] C. Louchet and L. Moisan, “Total variation denoising using iterated conditional expectation,” in *Proc. 22nd IEEE Eur. Signal Process. (EUSIPCO)*, Sep. 2014, pp. 1592–1596.

[62] A. Buades, B. Coll, and J. M. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, Jan. 2006.

[63] [Handson-on Image Processing with Python](https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/visual-recognition-overview.html), Packt Publishing Limited, 2018, pp. 127–439.

[64] [Affectiva](https://affectiva.com/), Boston, MA, USA, Tech. Rep. Accessed: Nov. 24, 2020. [Online]. Available: https://affectiva.com/.

[65] [Overview: Visual Recognition in Watson Studio](https://www.ibm.com/docs/en/watson-studio?topic=visaual-recognition-overview), IBM. Accessed: Dec. 12, 2019. [Online]. Available: https://dataplatform.cloud.ibm.com/docs/content/wsi/analyze-data/visual-recognition-overview.html

[66] K. Janocha and W. M. Czarnecki, “On loss functions for deep neural networks in classification,” 2017, arXiv:1702.05659. [Online]. Available: http://arxiv.org/abs/1702.05659

[67] S. Do, K. D. Song, and J. W. Chung, “Basics of deep learning: A radiologist’s guide to understanding published radiology articles on deep learning,” *Korean J Radiol.*, vol. 21, no. 1, pp. 33–41, 2020.

[68] J. Ludwig, *Image Convolution*. Portland State University. Accessed: Dec. 12, 2019. [Online]. Available: http://web.pdx.edu/~jdhul/courses/Archive/geog481w07/Students/LudwigImageConvolution.pdf

[69] S. H. Ahn, *Convolution*. Accessed: Dec. 27, 2019. [Online]. Available: http://www.songho.ca/dsp/convolution/convolution.html

[70] G. Barbastathis, A. Ocsan, and G. Situ, “On the use of deep learning for computational imaging,” *Optica*, vol. 6, no. 8, pp. 921–943, 2019.
AI Networks Generate Super-Resolution from Basic Microscopy, The Scientist Magazine, Accessed: Apr. 24, 2019.

Available: https://www.the-scientist.com/news-opinion/ai-networks-generate-super-resolution-from-basic-microscopy-65219

E. Nehme, L. E. Weiss, T. Michaely, and Y. Shechtman, “Deep-STORM: Super-resolution single-molecule microscopy by deep learning,” Optica, vol. 5, no. 4, pp. 458–464, 2018.

M. Ovesny, P. Křížek, J. Borkovec, Z. Švindrych, and G. M. Hagen, “ThunderSTORM: A comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging,” Bioinformatics, vol. 30, no. 16, pp. 2389–2390, Aug. 2014.

C. T. Rueden, J. Schindelin, M. C. Hiner, B. E. DeZonia, A. E. Walter, E. T. Arena, and K. W. Eliceiri, “ImageJ2: ImageJ for the next generation of scientific image data,” BMC Biol., vol. 18, no. 1, p. 529, Dec. 2017.

J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, “Fiji: An open-source platform for biological-image analysis,” Nature Methods, vol. 9, no. 7, pp. 667–682, Jul. 2012.

D. Sage, H. Kirschn, T. Pengo, N. Sturman, J. Min, S. Manley, and M. Unser, “Quantitative evaluation of software packages for single-molecule localization microscopy,” Nature Methods, vol. 12, no. 8, pp. 717–724, Aug. 2015.

S. Gazagnes, E. Soubies, and L. Blan-Feraud, “High density molecule localization for super-resolution microscopy using CEL0 based sparse approximation,” in Proc. IEEE 14th Int. Symp. Biomed. Imag. (ISBI), Apr. 2017, p. 4.

M. Weigert, L. Royer, F. Jug, and G. Myers, “Isotropic reconstruction of 3D fluorescence microscopy images using convolutional neural networks,” in Medical Image Computing and Computer-Assisted Intervention—MICCAI, M. E. A. Descoteaux, Ed. Springer, 2017, pp. 126–134.

Y. Wu, Y. Rivenson, H. Wang, Y. Luo, E. Ben-David, L. A. Bentolila, C. Priz, and A. Ozcan, “Three-dimensional virtual refocusing of fluorescence microscopy images using deep learning,” Nature Methods, vol. 16, no. 12, pp. 1323–1331, Dec. 2019.

Y. Zhou, R. Chellappa, and B. Jenkins, “A novel approach to image restoration based on a neural network,” in Proc. Int. Conf. Neural Netw., San Diego, CA, USA, Jun. 1997, pp. 269–276.

S. Sağıroğlu and E. Beşdok, “A novel approach for image denoising based on artificial neural networks,” J. Polytech., vol. 15, no. 2, pp. 71–86, 2012.

V. Jain and H. S. Seung, “Natural image denoising with convolutional networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2008, pp. 769–776.

M. Weigert et al., “Content-aware image restoration: Pushing the limits of fluorescence microscopy,” Nature Methods, vol. 15, no. 12, pp. 1090–1097, Dec. 2018.

B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, “Image denoising review: From classical to state-of-the-art approaches,” Inf. Fusion, vol. 55, pp. 220–244, Mar. 2020.

P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, “Multi-level wavelet-CNN for image restoration,” 2018, arXiv:1805.07071 [Online]. Available: http://arxiv.org/abs/1805.07071.

A. Sharoodi, R. Shahdossi, and Z. Rahemi, “Edge-preserving image denoising using a deep convolutional neural network,” Signal Process., vol. 159, pp. 20–32, Jun. 2019.

S. Gai and Z. Bao, “New image denoising algorithm via improved deep convolutional neural network with perceptive loss,” Expert Syst. Appl., vol. 138, Dec. 2019, Art. no. 112815.

I. Arganda-Carreras and J.-Y. Tinevez, “Fiji: An open-source platform for biological-image analysis,” Nature Methods, vol. 12, no. 8, pp. 717–724, Aug. 2015.

J. Reymann, Lightning White Paper—Image Information Extraction by Adaptive Deconvolution, Wetzlar, Germany: Leica Microsystems, 2018.

I. V. Dobrin, Lightning White Paper—Image Information Extraction by Adaptive Deconvolution, Wetzlar, Germany: Leica Microsystems, 2015.

A. G. T. Watson, “STORM data analysis and super-resolution imaging,” in Medical Image Computing and Computer-Assisted Intervention—MICCAI, Cham, Switzerland: Springer, 2015.

S. Akhter, Lightning White Paper—Image Information Extraction by Adaptive Deconvolution, Wetzlar, Germany: Leica Microsystems, 2018.

J. Schindelin and L. Bertrand, Thunder IMagers: How Do They Really Work, Wetzlar, Germany: Leica Microsystems, 2015.

A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, “Image denoising review: From classical to state-of-the-art approaches,” Inf. Fusion, vol. 55, pp. 220–244, Mar. 2020.

P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, “Multi-level wavelet-CNN for image restoration,” 2018, arXiv:1805.07071 [Online]. Available: http://arxiv.org/abs/1805.07071.

A. Sharoodi, R. Shahdossi, and Z. Rahemi, “Edge-preserving image denoising using a deep convolutional neural network,” Signal Process., vol. 159, pp. 20–32, Jun. 2019.

S. Gai and Z. Bao, “New image denoising algorithm via improved deep convolutional neural network with perceptive loss,” Expert Syst. Appl., vol. 138, Dec. 2019, Art. no. 112815.

H. Salyer, Getting Started with Topaz Sharpen AI, Accessed: Dec. 15, 2019. [Online]. Available: https://topazlabs.com/sharpen-ai

B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, “Image denoising review: From classical to state-of-the-art approaches,” Inf. Fusion, vol. 55, pp. 220–244, Mar. 2020.

P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, “Multi-level wavelet-CNN for image restoration,” 2018, arXiv:1805.07071 [Online]. Available: http://arxiv.org/abs/1805.07071.

A. Sharoodi, R. Shahdossi, and Z. Rahemi, “Edge-preserving image denoising using a deep convolutional neural network,” Signal Process., vol. 159, pp. 20–32, Jun. 2019.

S. Gai and Z. Bao, “New image denoising algorithm via improved deep convolutional neural network with perceptive loss,” Expert Syst. Appl., vol. 138, Dec. 2019, Art. no. 112815.

I. Arganda-Carreras and J.-Y. Tinevez, “Fiji: An open-source platform for biological-image analysis,” Nature Methods, vol. 12, no. 8, pp. 717–724, Aug. 2015.

J. Reymann, Lightning White Paper—Image Information Extraction by Adaptive Deconvolution, Wetzlar, Germany: Leica Microsystems, 2018.

I. V. Dobrin, Lightning White Paper—Image Information Extraction by Adaptive Deconvolution, Wetzlar, Germany: Leica Microsystems, 2015.

A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, “Image denoising review: From classical to state-of-the-art approaches,” Inf. Fusion, vol. 55, pp. 220–244, Mar. 2020.

P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, “Multi-level wavelet-CNN for image restoration,” 2018, arXiv:1805.07071 [Online]. Available: http://arxiv.org/abs/1805.07071.

A. Sharoodi, R. Shahdossi, and Z. Rahemi, “Edge-preserving image denoising using a deep convolutional neural network,” Signal Process., vol. 159, pp. 20–32, Jun. 2019.

S. Gai and Z. Bao, “New image denoising algorithm via improved deep convolutional neural network with perceptive loss,” Expert Syst. Appl., vol. 138, Dec. 2019, Art. no. 112815.

H. Salyer, Getting Started with Topaz Sharpen AI, Accessed: Dec. 15, 2019. [Online]. Available: https://topazlabs.com/sharpen-ai

B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, “Image denoising review: From classical to state-of-the-art approaches,” Inf. Fusion, vol. 55, pp. 220–244, Mar. 2020.

P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, “Multi-level wavelet-CNN for image restoration,” 2018, arXiv:1805.07071 [Online]. Available: http://arxiv.org/abs/1805.07071.

A. Sharoodi, R. Shahdossi, and Z. Rahemi, “Edge-preserving image denoising using a deep convolutional neural network,” Signal Process., vol. 159, pp. 20–32, Jun. 2019.

S. Gai and Z. Bao, “New image denoising algorithm via improved deep convolutional neural network with perceptive loss,” Expert Syst. Appl., vol. 138, Dec. 2019, Art. no. 112815.
