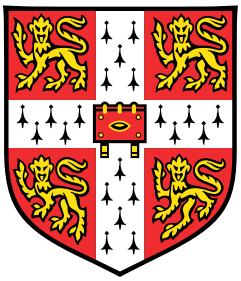
Deep learning for image processing in optical super-resolution microscopy



Charles Nicklas Christensen

Laser Analytics Group in Department of Chemical Engineering and Biotechnology
Computational Biology and Artificial Intelligence Group in Department of Computer Science and Techonology
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Hughes Hall October 2022

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. The thesis contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures. It does not exceed the prescribed word limit for the relevant Degree Committee.

Charles Nicklas Christensen
October 2022

Abstract

Deep learning for image processing in optical super-resolution microscopy

Charles Nicklas Christensen

Optical microscopy is fundamentally governed by a trade-off between image quality, imaging speed and duration. The quality can be considered a function of the signal-to-noise ratio, contrast and image resolution, which are all limited by the amount of light that can be acquired within a set exposure time. Many applications in live-cell imaging have specific requirements for illumination power and exposure time, thus necessitating a compromise with quality. In recent years, this fundamental limitation in optical microscopy has been shifted with the aid of deep learning methods. In this thesis, I propose methods that improve robustness to noise in image processing while making greater use of the available signal in the data. Applications include denoising for improved electron tomography when using cryogenic electron microscopy; image segmentation facilitating quantitative analysis of dynamics in endoplasmic reticulum (ERnet); and versatile reconstruction of super-resolved images from raw data acquired with structured illumination microscopy (ML-SIM). The deep learning methods that are presented are compared to classical image processing alternatives and tested on real experimental data acquired by collaborators in different departments of the university.

The overall finding of the thesis is that deep learning techniques offer a highly effective approach to many problems in bioimaging. With ERnet, it is possible to obtain a segmentation method that is reliable, fast and functional across different experiments without the need for retraining guided by further manual annotations. As for ML-SIM, I show that the reconstruction of structured illumination microscopy data can be treated as the inverse problem of a forward modelling process. This relies on an approximative image formation model that takes uncertainties and noise into account. By training a deep neural network to invert the forward modelled SIM data, a highly generalised reconstruction model can be obtained, which can handle SIM data from multiple microscopes while providing a high reconstruction quality.

The thesis is concluded with a reflective section on where the field is headed and which future applications may be enabled by the advancement of deep learning techniques.

This thesis is dedicated to my beloved parents, brother and sister.

Prøv at hør', Hvor fuglene synger,
Som en sød symfoni – med toner indeni,
Mens at livet lige så stille, Gynger forbi

(Listen to how the birds sing,
Like a sweet symphony – with tones within,
As life so slowly sways by)

— Kim Larsen

Acknowledgements

First of all, I would like to thank my supervisor, Prof. Clemens Kaminski, for consistent support and guidance throughout my PhD project. I am grateful for the patience and open-mindedness he showed me in the beginning of the project with regards to letting me follow the paths that drew my curiosity and interests. Coming from a background of mathematical physics and modelling, the transition to biomedical imaging in a largely experimental group has not been straight-forward, but Clemens certainly has made it easier by welcoming new directions and focus areas for the group. I have also appreciated Clemens' aptitude for ideation and enthusiasm; I have often come out of meetings feeling a renewed sense of motivation. One of the most important things I have learned from him is to think of research projects in terms of narratives, which I find makes everything more meaningful and purposeful as it forces one to think of the bigger picture and ensures that the general direction is steered towards something that will be useful for the community.

I would also like to thank my second supervisor, Prof. Pietro Lió, who has been very forthcoming to meeting to discuss my project whenever I have needed to. Pietro has provided me with a lot of inputs and suggestions that have been helpful in the development of the project.

Several of my group members in Laser Analytics Group have my gratitude: Dr Edward N. Ward, for continued collaboration on our method *ML-SIM* and his never-failing expertise on SIM; Dr Meng Lu, for his many ideas for applications of deep learning to image analysis, which has led to our collaboration on processing and analysing images of the endoplasmic reticulum using neural networks; Lucia Wunderlich, for supplying lots of experimental data and her enthusiasm for trying various image processing methods on her data; Max Barysevich, for many conversations and discussions about deep learning ranging from research papers to project findings; Marius Brockhoff, for valuable input, suggestions and encouragement. I also need to mention some of the previous group members that have been helpful towards my project including Dr Lisa Hecker, who I have collaborated with on SIM reconstruction for an instrument in the group, Dr Miranda Robbins, with whom I have collaborated with on calcium imaging and Dr Stanislaw Makarchuk who has provided useful input for my academic writing.

Outside of the group, I would like to thank: Jana M. Weber from the Sustainable Reaction Engineering group of CEB, for collaboration on applying graph theory to the analysis of

endoplasmic reticulum images; Dr Dari Kimanius from Scheres Lab at the MRC Laboratory of Molecular Biology (LMB) for discussions about deep learning and cryogenic electron microscopy; my PhD advisor Dr Jerome Boulanger, also from LMB, who I have only had a few conversations with but all very insightful. Other collaborators not mentioned here are mentioned in this thesis in the relevant sections.

Finally, my deepest thanks go to my parents, little sister and little brother who always bring joy to my life and offer moral support when I have been in need of it not least during this period of my life where I have been away from home to work on the PhD project. This thesis is dedicated to them and the rest of the Christensen family.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

In this first chapter, I will present an overview of the thesis' subject matter and underline its significance and motivation. Furthermore, a brief description of the thesis' outline is provided to help guide the reader.

1.1 Motivation

Optical fluorescence microscopy is fundamentally limited by the number of photons that can be received from fluorophores. There is a trade-off at play between temporal resolution, image quality (in terms of spatial resolution and signal-to-noise ratio), and achievable imaging duration. For live-cell imaging, which is one of the unique use cases of optical microscopy, a minimum imaging speed is often necessary to capture a certain biological process. This sets a requirement on the exposure time having to be sufficiently short, thus also reducing the number of photons received if everything else stays constant. To compensate for this, one might use higher excitation power to increase the fluorescent intensity, but this will come at the expense of shortening the duration for which useful data can be acquired of the sample. A higher imaging speed and excitation power will also increase the rate of photobleaching, which causes the fluorescence intensity to weaken over time such that the signal-to-noise ratio decreases. In addition to this, the effect of photo-damage and phototoxicity is also increased, which may alter the physiological behaviour of the sample or even distort the structure of the sample.

Machine learning (ML) has emerged as a new field for optical microscopy, which enables many new use cases from automation, high throughput analysis to image restoration. This thesis explores whether the traditional boundaries of imaging can be pushed further by leveraging the versatility and robustness to noise that ML methods enable. The relationship between the aforementioned imaging conditions and the interface explored in this work is illustrated on Figure 1.1. These trade-offs are described in more detail in [183].

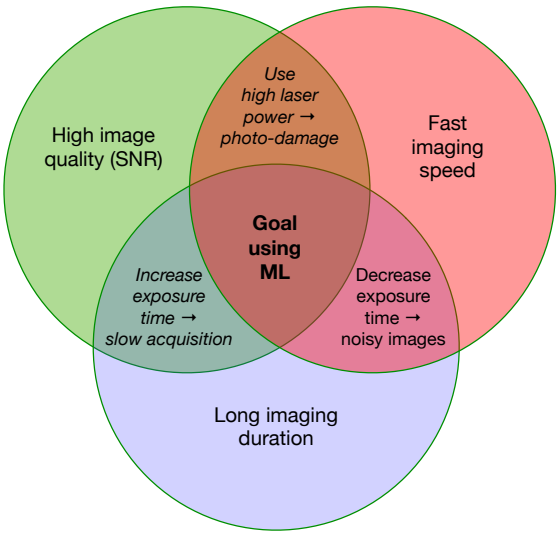


Figure 1.1: Trade-off between imaging speed, image quality and image duration, which is characteristic of optical microscopy.

Applications of low-light imaging capability. The regime depicted in the centre of Figure 1.1 is of particular interest because many applications in bioimaging can benefit directly from the capability of high-quality, fast imaging with a limited photon budget. One use case is the imaging of highly dynamic systems. An example is the imaging of the endoplasmic reticulum to investigate the peristaltic flow of luminal proteins, which has been achieved via structured illumination microscopy (SIM) at a 40 Hz frame rate [78]. Another example is [141], in which the beating heart of a zebrafish is imaged at 100 Hz using light-sheet fluorescence microscopy (LSFM). On the other extreme is long-term imaging, where illumination must be kept minimal to reduce photodamage. Long-term imaging is important for studying developmental biology. LSFM has previously been used to investigate organ morphogenesis in drosophila over a period of 20 hours [175]. Finally, it may also be that a three-dimensional dataset of a living sample is desired. Even if the sample system may not be very dynamic, volumetric imaging may well require a fast acquisition rate, since several planes have to be captured to construct a volume of the sample (a z-stack). Fast volumetric imaging has for instance enabled a detailed study of mitosis, the separation of chromosomes into two new nuclei, by resolving the detailed three-dimensional structure of two chromosomes as they split at 1 volume per second with each volume consisting of 200 planes [163].

These current use cases can be addressed with SIM and LSFM. However, to push the boundaries of spatio-temporal resolution further, new techniques are required. Currently, it is not possible to image dynamic systems using super-resolution SIM at high quality and a rate of hundreds of frames per second, or to image a developing organism for days instead of hours, or acquire tens of volumetric images per second of moving three-dimensional structures with

LSFM. In this thesis, I propose methods for super-resolution reconstruction, image denoising and segmentation that perform well under the challenging conditions of low signal-to-noise ratio and high speed.

1.2 Research questions

This thesis overall deals with methods development. In addition to the pursuit of new and improved methods, I also quantitatively investigate the performance limits of existing and proposed methods and consider multiple biological applications. The focus of the thesis is summarised in the following questions:

- How much can the exposure times potentially be reduced for wide-field and SIM imaging if denoising methods are employed?
- Which advantages do ML enable for typical quantitative analysis workflows in bioimaging?
- How can the generalisability and versatility offered by deep learning benefit image processing tasks such as image segmentation and super-resolution reconstruction for SIM?
- Which training strategies are most fitting for computer vision models in the context of bioimaging?

1.3 Aims and objectives

This PhD project is inspired by the challenges and limitations currently plaguing the field of optical fluorescence microscopy, particularly in live-cell imaging applications. As image quality, temporal resolution, and achievable imaging duration are in constant trade-off, this research primarily aims to innovate and introduce robust machine learning methods that can effectively manage very low signal-to-noise ratios, thereby facilitating low-light imaging.

A central objective of this work is the design, implementation, and training of models that are not only well-performing in a low-light imaging context, but also versatile in their applications.

This research targets the exploration of diverse domains within microscopy image analysis, namely denoising, super-resolution, segmentation, and reconstruction, capitalising on the power of deep learning.

One of the concrete goals has been the creation of a versatile segmentation model designed to facilitate quantitative analysis of the endoplasmic reticulum under a wide array of imaging conditions. An additional aspiration lies in leveraging artificial neural networks for the task of reconstructing SIM images.

Summarily, this project is not confined to the traditional limitations of microscopy image analysis. Instead, it aims to explore novel approaches and robust solutions, ultimately contributing to significant advancements in the field image processing and image analysis in microscopy.

1.4 Outline

The thesis is structured into separate areas of applications of deep learning within scientific imaging that have been studied during the PhD project. The primary scientific contributions are in fluorescence imaging, but the scope is extended to other areas in the case of image restoration, Chapter 3, i.e.: (a) benchmarking for denoising performance assessment using the PatchCamelyon dataset based on histology images, (b) denoising for scanning electron microscopy, and (c) processing of astronomy images for quantitative photometry.

Chapter 2

Background

In this chapter I will cover the theoretical basics of the topics that are essential to the thesis. These topics first and foremost include:

- The physics of optical microscopy
  - The diffraction limit and fluorescence emission
- Super-resolution imaging
  - Structured illumination microscopy
- Machine learning and deep learning
  - Supervised, self-supervised and transfer learning
  - Convolutional neural networks
  - Attention mechanism and transformer network
  - Classification and regression loss functions

2.1 Optical microscopy

The purpose of this section is to introduce some concepts that are essential to imaging theory and will be referred to throughout the thesis in the context of microscopy. The goal of imaging theory is to analyse the mapping process of light travelling through a system onto a 2D plane called the image plane. In a simple imaging application we can consider light travelling from a two-dimensional plane, the object plane, until it is mapped onto the image plane. Structured illumination microscopy is such an application, and to understand its utility the origins of diffraction-limited resolution in fluorescence microscopy will briefly be described.

Diffraction of light is a fundamental phenomenon that occurs when light propagates through an aperture of finite size such as in a pinhole camera or a slit. The first study of the effects of diffraction dates back to Francesco Maria Grimaldi in 1660, who coined the term diffraction based on a Latin word meaning "to break into pieces" [20]. The phenomenon is qualitatively described by the Huygens-Fresnel principle, which states that the wavefront of a propagating wave can be considered a collection of point sources of spherical wavelets [51]. The analytical understanding of the phenomenon can be found in Maxwell's equations [140] from which Ernst Abbe derived his influential theory of image formation [138]. An important result in his pioneering paper from 1873 is that due to diffraction the resolution of a microscope is fundamentally limited by the relationship

$$d = \frac{\lambda}{2n\sin\alpha}, \tag{2.1}$$

where d is the minimum distance between two point sources that can be resolved, λ is the wavelength of the illumination, and $n\sin\alpha = NA$ is the numerical aperture of the microscope's objective lens that depends on the refractive index and incidence angle. It follows from Abbe's limit that oblique or off-axis illumination leads to enhanced optical resolution as well as increasing the refractive index of the medium between the object and the objective.

Following Abbe's proposal of the resolution limit, other criteria were suggested towards the concept of the two-point resolution. Examples are the Rayleigh criterion, Sparrow criterion, criteria based on Fourier theory and the Nyquist theorem [135].

2.1.1 Fourier optics

In general terms, light can be represented as a field $E(\mathbf{r},t)$ that moves in time, t, and space, $\mathbf{r} = (x,y,z)$. The field $E(\mathbf{r},t)$ is related to its Fourier transformed representation $\tilde{E}(\mathbf{k},\nu)$ through the Fourier transform and the inverse Fourier transform

$$E(\mathbf{r},t) = \iiiint \tilde{E}(\mathbf{k},\nu) e^{i2\pi(\mathbf{k}\cdot\mathbf{r}-\nu t)}\,\mathrm{d}k_x\,\mathrm{d}k_y\,\mathrm{d}k_z\,\mathrm{d}\nu \qquad \equiv iFT(\tilde{E}(\mathbf{k},\nu)), \tag{2.2}$$

$$\tilde{E}(\mathbf{k},\nu) = \iiiint E(\mathbf{r},t) e^{-i2\pi(\mathbf{k}\cdot\mathbf{r}-\nu t)}\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z\,\mathrm{d}t \qquad \equiv FT(E(\mathbf{r},t)), \tag{2.3}$$

where the Fourier conjugate variable of the position vector, $\mathbf{k} = (k_x,k_y,k_z)$, is the wavevector that is related to the angular wavevector by $\kappa = 2\pi\mathbf{k}$.

With Fourier theory, the concept of resolution can be treated in a more intuitive way. The point spread function (PSF) is defined as the response function of an imaging system to a point source. It can be thought of as the blurring kernel that represents the effects of diffraction.

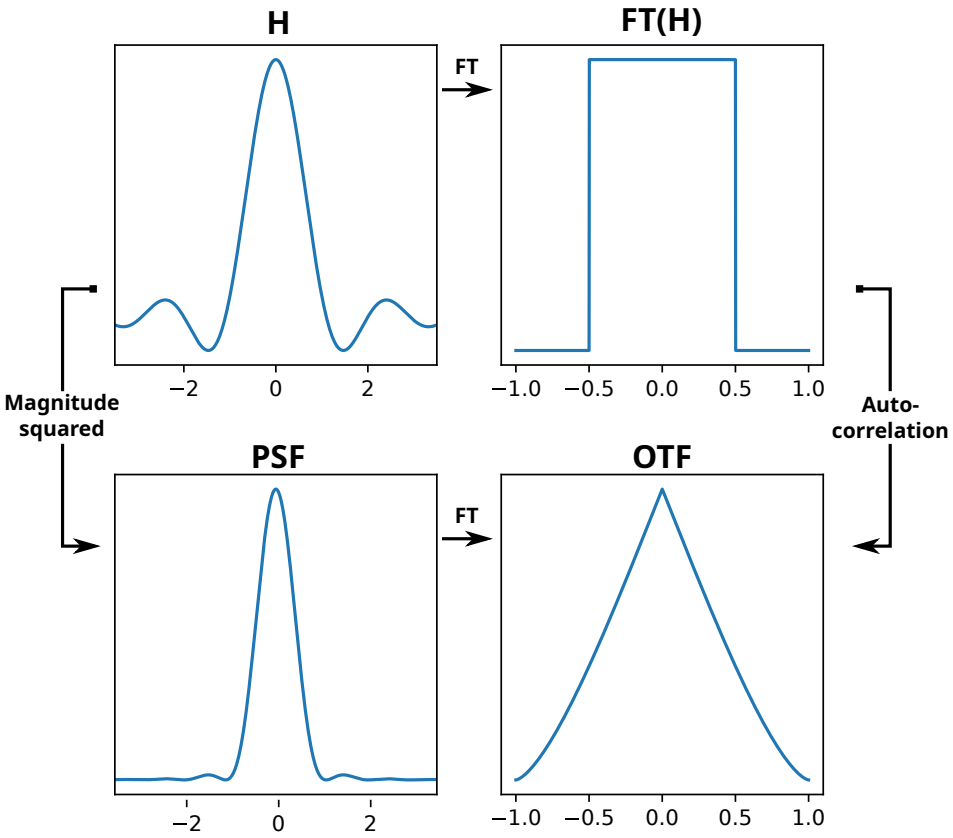


Figure 2.1: Incoherent imaging functions for light passing through a circular aperture.

Physically, the reason for the blurring is due to the loss of information when light travels through an aperture in an imaging system. Light from a fluorescent point source is a spherical wave, but due to the finite size of the aperture only the light that travel with an angle corresponding to the location and size of the aperture will be accepted – the rest of the light is discarded. In the example of the aperture being circular, the field produced at the image plane, say H, whose intensity corresponds to the PSF, sometimes referred to as the amplitude PSF as opposed to the intensity PSF [98], is proportional to the cylindrical Bessel function of order 1 [140], which is visualised on Figure 2.1. The PSF is the squared magnitude of the field H, and H is also related to the so-called coherent transfer function through the Fourier transform as defined in Equation (2.2). The coherent transfer function is a scaled version of the piecewise pupil function that represents the circular aperture as shown on Figure 2.1 and it effectively truncates the information transferred through the aperture. This brings us to the important quantity called

the optical transfer function (OTF), which is the Fourier transform of the PSF

$$\mathrm{OTF}(\mathbf{k}) = \mathrm{PSF}(\mathbf{r}). \tag{2.4}$$

The OTF is also the autocorrelation of the coherent transfer function, and it provides a convenient way to consider resolution because information is only obtained for regions in which the OTF is non-zero up to a cut-off frequency that is consistent with the Abbe resolution limit. The area of frequencies that is supported by the OTF is referred to as the passband of the microscope. However, the frequency support is not uniform as the truncation posed by the pupil function results in a gradual decrease towards the edges of the OTF as is clear from the example of the circular aperture, Figure 2.1. If we assume that the noise in the system is relatively uniform, then it follows that information corresponding to higher spatial frequencies will have progressively lower signal-to-noise ratio (SNR) and a worsening contrast. This points to the relevance of denoising and robustness to noise for super-resolution imaging, which is discussed in more detail in Chapter 3, Chapter 4 and Chapter 5.

2.2 Super-resolution optical microscopy

Optical super-resolution microscopy techniques have emerged over the last three decades and have now become an essential part of the toolbox for biomedical imaging. Optical super-resolution goes beyond image upsampling, which in the computer vision literature also is referred to as super-resolution, see Section 2.4.3. To avoid confusing the two, a distinction will be made between optical and image super-resolution.

The primary techniques in super-resolution optical microscopy are single molecule localisation microscopy (SMLM) [75, 144], stimulated emission depletion microscopy (STED) [9] and structured illumination microscopy [187, 72, 59].

The field of super-resolution optical microscopy was recognised by the Nobel Prize in 2014 given to the inventors of SMLM and STED [75, 144, 9]. Both STED and SMLM can achieve very high resolution, resolving structures smaller than 50 nm, but being fundamentally very different to SIM they lack both the speed and photon efficiency that allow application to live-cell imaging of dynamic samples. However, they have other important merits that I will briefly describe in the following.

2.2.1 Stimulated emission depletion microscopy

STED microscopy relies on the principle that fluorophores can be selectively deactivated, thus reducing the size of the point spread function of the system. The way this works is by

illuminating the sample with a secondary ring-shaped beam, sometimes colloquially referred to as a doughnut-shape, that quenches fluorescence everywhere except at the very focal centre where the depletion beam intensity is zero. The focal centre is instead illuminated with a standard excitation beam that could also be used for confocal fluorescence microscopy [140]. The depletion beam causes stimulated emission to occur from the area around the focal centre, which means the fluorescence signal that instead originates from spontaneous emission in the focal centre is significantly sharpened. For this imaging scheme to image an entire sample, the acquisition must be done by scanning the sample similarly to confocal microscopy imaging. With STED being a laser-scanning technique, the imaging speed is much slower than for wide-field imaging techniques such as SIM.

The resolving power of STED is in theory unrestricted because the achieved resolution, as given by a modified version of Abbe's limit [135], is inversely proportional to $\sqrt{1+I_{\text{STED}}/I_{\text{sat}}}$, in which I_{STED} is the maximum intensity of the superimposed STED beam and I_{sat} is the saturation intensity that represents the required intensity to switch off a fraction of $1/e$, or one half in some publications [74], of the molecules in the depletion zone. The proportionality factor means that the resolution can be increased simply by increasing the STED power, and the ratio of $I_{\text{STED}}/I_{\text{sat}}$ can be several orders of magnitude in practice [140]. This mechanism for achieving super-resolution with STED leads to high laser intensities that often result in damage to biological structures and require special strategies to avoid photobleaching and phototoxicity [200].

As a diffraction-unlimited technique, STED has been used to image cellular structures down to 20 nm [56]. Extensions of STED to multi-colour imaging exist with the most versatile implementations relying on super-continuum lasers for the excitation beam and a second fiber laser for the STED beam [200]. An example of such an implementation is demonstrated in [217], where four-colour live-cell imaging is achieved, albeit with 15 second acquisition time per frame and a small field-of-view. Faster implementations achieving video-rate STED imaging have also been demonstrated [211] at 28 frames per second, but again with a small field-of-view and in this case a low dynamic range with reduced bit depth.

2.2.2 Single molecule localisation

The continued development of photoactivatable molecules has made an avenue of super-resolution methods possible that do not require structured illumination or laser scanning. The underlying principle for this family of methods is to configure the fluorescence from the sample such that only a small subset of the fluorophores emit for every image acquired, thus allowing a final image to be reconstructed from the sequential measurement of these individual fluorescent molecules. Due to the scarcity of the fluorescence in every image, it is improbable for two

fluorophores in proximity to emit at the same time, which means that a detected fluorescent spot uniquely corresponds to the location of a single point source. The resolution limit as described in Equation (2.1) describes the minimum distance, d, where two emitters can be distinguished. However, with the knowledge that a detected spot originates from a single emitter, the point spread function of the imaging system can be taken into account, and the fluorescent spot can be deconvolved to provide a sub-diffraction limit estimate of the location. Assuming that the detection is free from noise from electronic noise and only affected by shot-noise, the localisation precision is given by [140] $d_{\text{loc}} = d/\sqrt{N}$, where N is the number of fluorescence photons detected from the emitting molecule. This relationship means that SMLM is diffraction-unlimited with a resolution that can be improved by acquiring more data. As a result, SMLM offers the highest resolution among the super-resolution techniques with many studies reporting lateral resolutions of 10 to 30 nm [95]. Using the method MINFLUX [61] a lateral resolution of 1-3 nm can even be achieved.

Several methods have been proposed for SMLM including photoactivated localisation microscopy (PALM) [9], stochastic optical reconstruction microscopy (STORM) [178], direct STORM (dSTORM) [71], DNA-based point accumulation for imaging in nanoscale topography (DNA-PAINT) [184] and the aforementioned MINFLUX. The methods overall differ in how the fluorescent photoswitching is implemented, the labelling process of the target protein with a fluorescent dye and the means of determining the localisation of the detected fluorescent spots. For all methods, acquired images are comprised of thousands of stacked frames that are collected sequentially. The reconstruction of a final image, or a 2D or 3D point cloud, is typically done using a Gaussian fit [95], while deep learning has also been proposed as an approach to reconstruction with e.g. Deep-STORM [150].

2.2.3 Structured illumination microscopy

SIM enables optical super-resolution by encoding structural details corresponding to high spatial frequencies of the sample into signals in the lower frequency domain. By unmixing the low-frequency data, information can then be recovered that would otherwise be lost with conventional wide-field imaging. The diffraction limit is described by the optical transfer function (OTF), which represents the transmittable bandwidth of spatial frequency through an imaging system. It is by shifting high spatial frequencies into the accessible passband that super-resolution by SIM is obtained. The OTF is the Fourier transform of the point spread function (PSF), which is the blur kernel in direct space. Conventional wide-field SIM uses sinusoidal illumination patterns formed by the interference of two beams [59], which leads to lateral resolution doubling. To similarly improve the axial resolution, three beams can be used as studied in [60, 131]. The 3D extension of SIM is not explored in this thesis and only

2D-SIM implementations will be considered. The sinusoidal pattern formed by the interfering beams in the conventional SIM implementation appears as stripes in the recorded image. This type of illumination pattern will be referred to as fringe illumination in this thesis. The illumination patterns have an orientation and a phase shift, which are commonly varied over three values to ensure symmetric frequency support, thus leading to a stack of nine frames with different patterns. In mathematical terms, SIM reconstruction solves the inverse problem of this excitation and blurring operation, thereby determining the fluorescent signal that represents the sample.

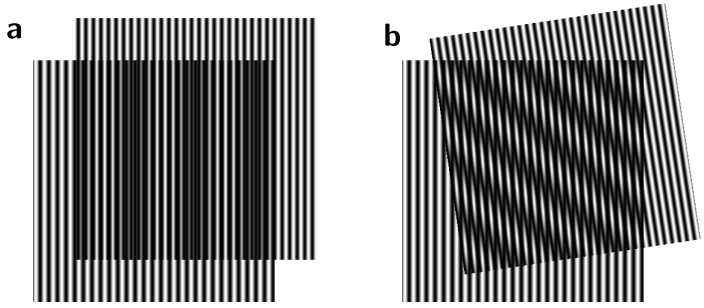


Figure 2.2: The Moiré pattern formed by superposition of two high frequency fringe patterns. The resulting interference pattern has lower spatial frequency depending on whether the superimposed patterns are parallel (a) or non-parallel (b).

The sinusoidal illumination pattern is typically generated by letting two beams interfere causing the Moiré pattern as shown in Figure 2.2. The resulting interference pattern used for illumination consists of fringes with variable spatial frequency k_0 and phase ϕ,

$$I_{\theta,\phi}(x,y) = I_0\left[1 - \frac{m}{2}\cos\left(2\pi(k_x x + k_y y) + \phi\right)\right], \tag{2.5}$$

where k_x, $k_y = k_0\cos\theta$, $k_0\sin\theta$ for a pattern orientation θ relative the horizontal axis, ϕ defines the phase of the pattern (i.e. the lateral shift in the direction of k_0) and m is the modulation depth, which defines the relative strength of the super-resolution information contained in the raw images. The fluorescent response of the sample can then be modelled by the multiplication of the sample structure, $S(x,y)$, i.e. input image, at time t and the illumination pattern intensity $I_{\theta,\phi}(x,y)$. The final image, $D_{\theta,\phi}(x,y)$, is formed after blurring by the PSF, $H(x,y)$, and addition of white Gaussian noise, $N(x,y)$. The use of $H(x,y)$ for the PSF in this context does not refer to the amplitude PSF but the intensity PSF, which differs from Section 2.1.1 and Figure 2.1. Mathematically, we can state the image formation model as

$$D_{\theta,\phi}(x,y) = \left[S(x,y)I_{\theta,\phi}\right] \otimes H(x,y) + N(x,y), \tag{2.6}$$

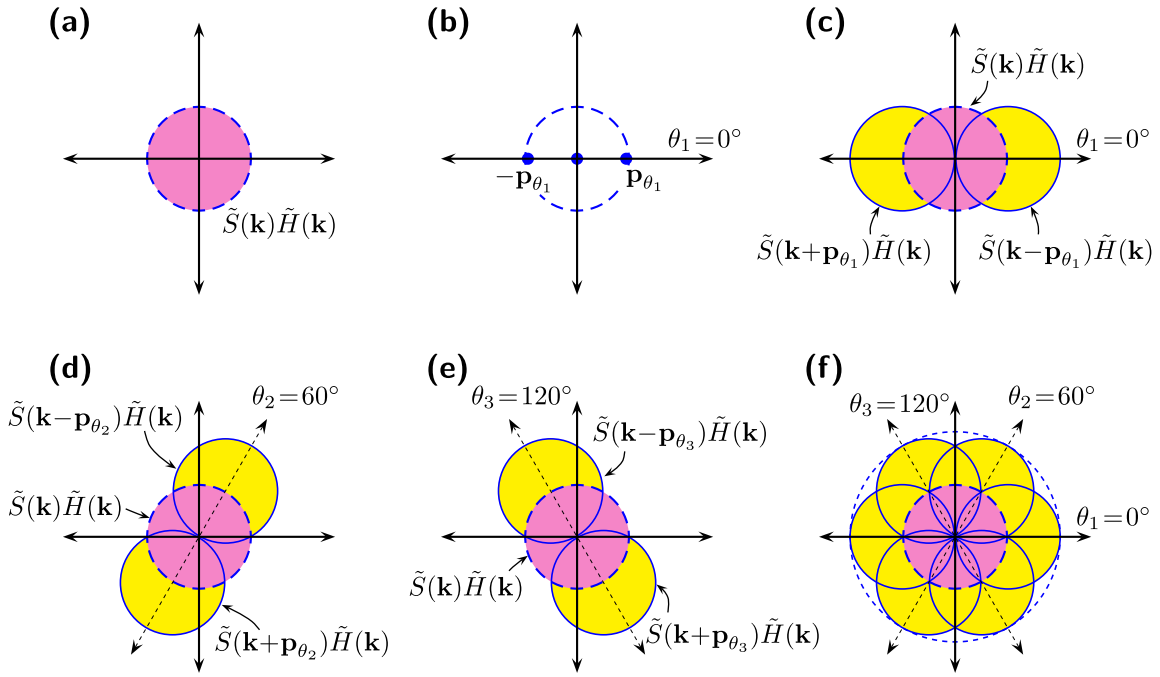


Figure 2.3: Geometric depiction of the frequency mixing principle that underlies SIM. **(a)** The frequency support, i.e. passband of the imaging system as given by the OTF, in a standard wide-field microscope. **(b)** The frequency support associated with the illumination with a sinusoidal pattern with a particular orientation, $\theta_1 = 0$, corresponding to vertical fringes. **(c,d,e,f)** The frequency content of a sample illuminated by the fringe pattern is a linear combination of the information in three circular regions. This means that the previously lost information is now mixed into the passband of the OTF, and it can be computationally recovered with enough data. Figure credit: [103].

where $\otimes$ is the convolution operation. Although relatively standard, the SIM formalism followed from here on is that of [103]. By using the convolution theorem the final image in frequency space can be written

$$\begin{aligned}\tilde{D}_{\theta,\phi}(\mathbf{k}) &= \left[\tilde{I}_{\theta,\phi}(\mathbf{k}) \otimes \tilde{S}(\mathbf{k})\right] \cdot \tilde{H}(\mathbf{k}) + \tilde{N}(\mathbf{k}) \\ &= \frac{I_o}{2}\left[\tilde{S}(\mathbf{k}) - \frac{m}{2}\tilde{S}(\mathbf{k}-\mathbf{p}_\theta)e^{-i\phi}\right. \\ &\qquad \left. -\frac{m}{2}\tilde{S}(\mathbf{k}+\mathbf{p}_\theta)e^{i\phi}\right] \cdot \tilde{H}(\mathbf{k}) + \tilde{N}(\mathbf{k}),\end{aligned} \tag{2.7}$$

where $\tilde{H}(\mathbf{k})$ is the OTF. In Section 2.2.3 the Fourier transform of Equation (B.2) has been used, which is a sum of Dirac delta functions. Since this sum of Dirac delta functions is convolved with the sample distribution, $\tilde{S}(\mathbf{k})$, the frequency content of the $\tilde{S}(\mathbf{k})$ is sifted out as the positions corresponding to the delta functions due to the sifting property [17]. The

Fourier transform of the final image, $\tilde{D}_{\theta,\phi}(\mathbf{k})$, is then seen to be a linear combination of this frequency content within three circular regions of the sample $\tilde{S}(\mathbf{k})$; one region centred at the origin, and two regions centred at $-\mathbf{p}_\theta$ in reciprocal space – see Figure 2.3 for an illustration. This is repeated for three different illumination phases, ϕ, providing the SIM images $D_{\theta,\phi_1}(\mathbf{r})$, $D_{\theta,\phi_2}(\mathbf{r})$ and $D_{\theta,\phi_3}(\mathbf{r})$ of the sample. The phases used are generally $\phi_1 = 0°$, $\phi_2 = 120°$ and $\phi_3 = 240°$. From Section 2.2.3, we then have the following system of equations

$$\begin{bmatrix} \tilde{D}_{\theta,\phi_1}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_2}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_3}(\mathbf{k}) \end{bmatrix} = \frac{I_o}{2}\mathbf{M} \begin{bmatrix} \tilde{S}(\mathbf{k})\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}-\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}+\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \end{bmatrix} + \begin{bmatrix} \tilde{N}_{\theta,\phi_1}(\mathbf{k}) \\ \tilde{N}_{\theta,\phi_2}(\mathbf{k}) \\ \tilde{N}_{\theta,\phi_3}(\mathbf{k}) \end{bmatrix}$$

$$\text{where } \mathbf{M} = \begin{bmatrix} 1 & -\frac{m}{2}e^{-i\phi_1} & -\frac{m}{2}e^{+i\phi_1} \\ 1 & -\frac{m}{2}e^{-i\phi_2} & -\frac{m}{2}e^{+i\phi_2} \\ 1 & -\frac{m}{2}e^{-i\phi_3} & -\frac{m}{2}e^{+i\phi_3} \end{bmatrix} \tag{2.8}$$

The factor of $I_o/2$ in Section 2.2.3 and Equation (2.8) is simply a scaling factor affecting the final image, which can be neglected as the recorded absolute values tend to be of no interest. By isolating the vector of the sifted sample distribution and factoring in the noise, we have the equation

$$\begin{matrix} \text{noisy} \\ \text{estimate} \\ \text{of} \end{matrix} \begin{bmatrix} \tilde{S}(\mathbf{k})\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}-\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \\ \tilde{S}(\mathbf{k}+\mathbf{p}_\theta)\tilde{H}(\mathbf{k}) \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} \tilde{D}_{\theta,\phi_1}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_2}(\mathbf{k}) \\ \tilde{D}_{\theta,\phi_3}(\mathbf{k}) \end{bmatrix} \tag{2.9}$$

The noisy approximations of $\tilde{S}(\mathbf{k})$, $\tilde{S}(\mathbf{k}-\mathbf{p}_\theta)$ and $\tilde{S}(\mathbf{k}+\mathbf{p}_\theta)$ can then be processed with a Wiener filter to increase the accuracy of the estimates. Finally, the centres of frequency components $\tilde{S}(\mathbf{k}-\mathbf{p}_\theta)$ and $\tilde{S}(\mathbf{k}+\mathbf{p}_\theta)$ can be shifted to their correct locations, $+\mathbf{p}_\theta$ and $-\mathbf{p}_\theta$ respectively, in frequency space. As a result, the unmixing of the frequency content is achieved, and the information that was inaccessible prior to the mixing can be recovered. The procedure can be repeated for a set of angular orientations θ of the illuminating fringe pattern in order to support the entire area of surrounding the OTF passband. As mentioned above, three orientations is the conventional choice, and it is adequate to provide frequency support for a circular region close to double the radius of the initial passband, which enables a doubling of resolution following reconstruction compared to the case of using the same optical system for regular wide-field imaging. The reconstruction process of these nine acquired SIM images is studied in Chapter 5.

2.2.4 Alternative approaches to structured illumination microscopy

The conventional implementation of SIM follows that outlined in Section 2.2.3, which is based on [59], but several other SIM techniques have been proposed and studied in the literature. This includes multi-spot SIM [206], speckle SIM [145, 3] and non-linear SIM [73]. Multi-spot and speckle SIM require many more frames per super-resolved reconstruction relative to linear SIM with fringe illumination, which is probably one of the main reasons they have not gained as much popularity in the field. Nevertheless, they present some unique advantages and speckle SIM is described and studied further in Section 5.2. Non-linear SIM generally rely on non-linearities in the fluorophores, such as pushing the fluorophores to their saturation limit, where maximum emission is reached [73], or use photo-switchable dyes to ensure that only fluorophores activated by an excitation pattern responds to subsequent sample illumination [169]. These non-linear modalities enable higher spatial resolution than classical SIM, but are less widely applicable due to the fluorophore requirements and will not be further considered in this thesis.

2.3 Machine learning and deep learning

Historically, in the scientific literature, machine learning has referred to statistical methods that rely on fitting procedures [11, 82].

Generally, the parameters of a mathematical model are fitted to data by minimisation of an objective function describing the discrepancy between data and prediction from the model. The discrepancy can be calculated as a set of differences, called residuals. A common choice for the objective function is a sum of squared residuals, hence the name *least squares* for this approach in the literature. An important early solution to the least squares problem is Ordinary Least Squares method, which assumes that the data follows a linear polynomial model that provides a closed form analytical solution [66].

Other similar methods include ridge regression and Least Angle Regression (LARS) [66]. Other similar methods to standard linear regression include Ridge regression and Least Angle Regression (LARS) [66]. Ridge regression is particularly known for its regularisation property, helping to prevent overfitting by shrinking the coefficients of the model. Similarly, LARS is a method for variable selection and regularisation. Although linear regression-based methods in their simplest forms might struggle to capture non-linear relationships directly, they can be extended or adapted to handle non-linearity. For instance, polynomial regression, a form of linear regression, can be used to approximate non-linear functions, although the complexity and risk of overfitting may increase with the degree of the polynomial. Other types of models have been proposed to capture more complex behaviour. Examples of such models are logistic

regression [34], support vector machines [13, 32], random forests [19] and artificial neural networks [137, 174].

Beyond model construction, the field of machine learning has also developed into different overall categories. These can largely be split into: supervised, unsupervised, self-supervised and reinforcement learning as depicted on Figure 2.4. Supervised learning is the most prevalent and impactful approach as of today, in which a labelled training dataset is available to learn from with pairwise corresponding inputs and outputs. Unsupervised learning deals with problems for which there is only an input signal with nothing to map it to, which can either be due to difficulty in obtaining ground truth data or simply that the problem does not have a concrete output variable. However, the underlying structure and patterns in the data may contain valuable insight, and this can for instance be explored with clustering methods such as K-means clustering. Self-supervised learning is a machine learning approach where the training data is automatically labelled, typically using part of the data input itself. The machine learning algorithm is given a task, often to predict or reconstruct part of the input, which can be treated as a form of supervision. Because of this, it is not necessary to provide explicit labels for the training data, hence the term self-supervised. An example of this is an autoencoder [77, 76, 53], in which the input is mapped to itself, but because of dimensionality reduction in the network architecture, an optimal encoding, i.e. compression, format of the data must be learned to enable a faithful decoding. As such, self-supervised learning shares the advantage of both supervised and unsupervised learning. In a similar vein, there is also semi-supervised learning [154, 25], which is comparable to supervised learning in its objective, but where the training dataset only has a small amount of labelled data and a large amount of unlabelled data. This calls for the use of some unsupervised techniques to extract anything useful from the unlabelled subset. Finally, there is reinforcement learning which is comparable to unsupervised learning in the way that it deals with optimisation without any labelled data to learn from. Instead, in reinforcement learning, the optimisation is based on a well-defined reward function, i.e. a performance metric, in the presence of a set of rules or world mechanics. An example is to optimally play a video game with the score defining the reward [143], or for a robot to learn to walk with the reward representing the amount of seconds it has kept its balance [62]. Reinforcement learning has proven an interesting direction in the field and may pave the way for artificial general intelligence [112], but has yet to truly find its place in the bioimaging community.

Given the relevance to this thesis, I will cover the topic of neural networks in more detail. The methods of this thesis primarily fall in the category of supervised learning with a few exceptions surrounding self-supervised and unsupervised learning. The other mentioned areas while important for perspective and historical reasons will not be covered further.

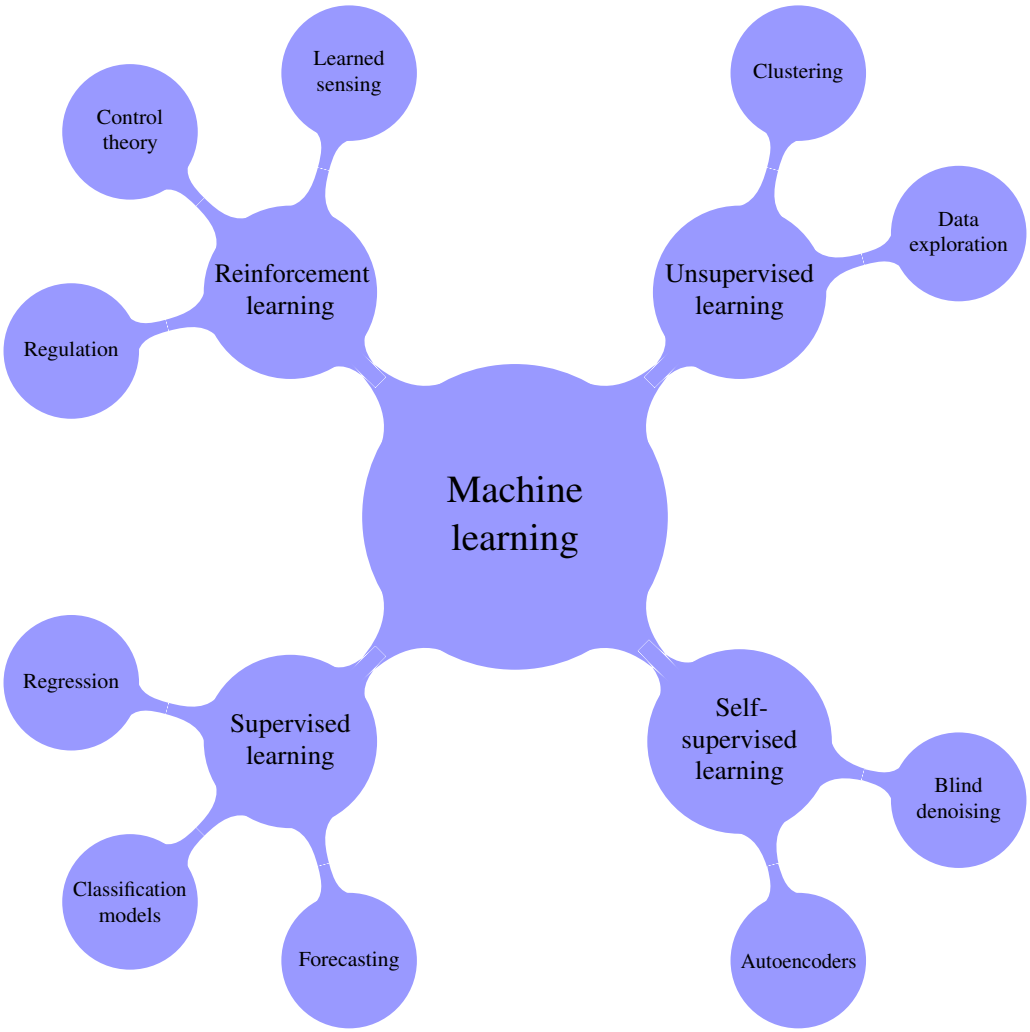


Figure 2.4: Overall categories in the machine learning field. The mentioned examples focus on bioimaging applications.

2.3.1 Artificial neural networks

Artificial neural networks are a family of computational models that are inspired by the animal brain with many smaller units working in parallel [176]. Weights between units encode long-term information and updating the weights is the process in which the neural network learns new information.

A common architecture for most neural networks is the feed-forward network, characterised by a sequence of layers, each performing specific operations. When these operations consist of a weighted sum of inputs, the model is referred to as a multi-layer perceptron (MLP). MLPs can have a single layer – a design of historical importance and often colloquially referred to as vanilla neural networks [66], albeit not effective for complex tasks.

In an MLP, each unit in a layer, called a node or neuron, is connected to other nodes via edges or connections. Each of these connections is associated with a coefficient or weight. The architecture of the network is formed by the nodes and their weighted connections. The weighted sum at each node is then transformed by a non-linear activation function. This non-linearity introduced by the activation function is crucial, enabling the network to model complex, non-linear relationships in the data, beyond the capability of linear models.

At the ends of the network are special layers termed as the input and output layers. These layers serve to translate the data to and from a latent or feature space – the domain where most of the computation occurs. Nodes and layers situated within this latent space, not directly interacting with the input or output data, are called hidden nodes and hidden layers, respectively.

For an MLP, the total input, $x_j^{(n)}$, to a hidden node j in layer n is a weighted sum of the outputs, $y_i^{(n-1)}$, from the previous layer that are connected to j. This can be expressed as:

$$x_j^{(n)} = \sum_i y^{(n-1)} i w^{(n)} j, i, \tag{2.10}$$

where $w_{j,i}^{(n)}$ are the weights associated with the connections from node i in layer $(n-1)$ to node j in layer n. Each hidden node then generates a real-valued output, $y_j^{(n)}$, using a non-linear activation function, such as the sigmoid function:

$$y_j^{(n)} = \frac{1}{1+e^{-x_j^{(n)}}}. \tag{2.11}$$

In this way, the MLP is able to successively transform the input data, layer by layer, through the latent space, ultimately producing the output. See Figure 2.5 for a graphical representation of an MLP.

For the network to do anything useful, the weights have to be learned. In other words, these weights are trainable, and they are adjusted towards their optimal values in a numerical optimisation method that updates their values after every evaluation of an input during the process called model training. The updating scheme can be as simple as gradient descent, which is a first-order method that adjusts the weights by taking a step in the opposite direction of the gradient of the error with respect to the weights. This direction can be considered the steepest descent in the landscape of error. However, while the conceptual simplicity of gradient descent is appealing, it is not necessarily very robust. One key drawback is its propensity to become stuck in local minima. The more sophisticated variant Stochastic Gradient Descent (SGD) [171, 14] addresses this and is a popular optimisation method for model training. SGD

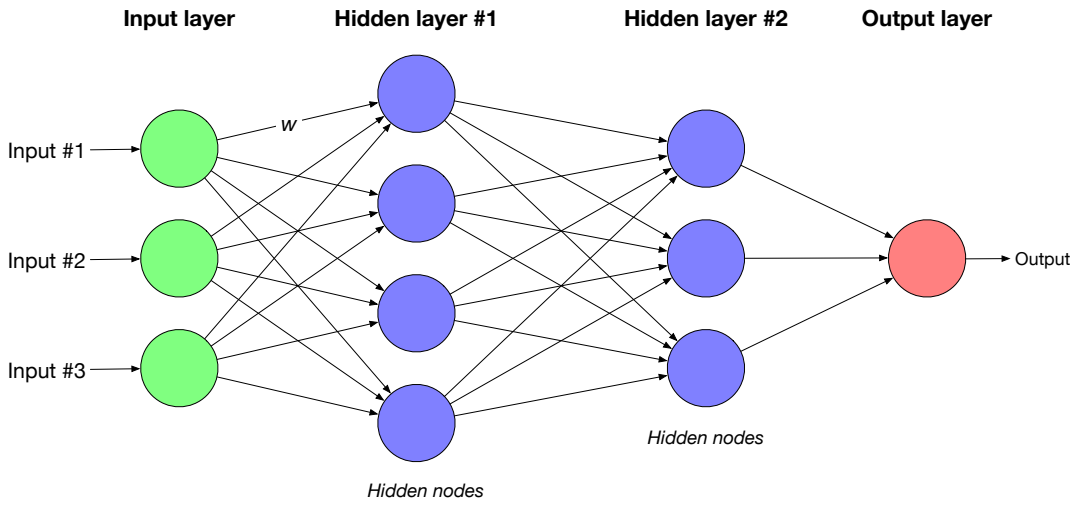


Figure 2.5: A multi-layer perceptron. The input layer is connected to the first hidden layer, which is connected to the second hidden layer, which is connected to the output layer. The connections between the layers have weights w.

introduces an element of randomness in the descent, picking a subset of the dataset to compute the gradient. This not only allows for more efficient computations, especially with large datasets, but also helps prevent the algorithm from settling into local minima. The size of the step taken, i.e. step size, is referred to as learning rate in the machine learning literature. Despite the significant improvements brought by SGD, the optimal learning rate remains a challenge. Addressing this, the Adaptive Moment Estimation (ADAM) method [97] has been proposed. By dynamically adjusting the learning rate for each weight based on the estimates of the first and second moments of the gradients, ADAM provides an efficient and more robust alternative for deep learning model optimisation.

For any of the aforementioned optimisation methods, a calculation of the gradients is necessary. Gradients are determined with an algorithm called backpropagation, which uses the chain rule on an error value, or loss, calculated as the deviation between the network's output and a desired target.

Backpropagation

Determining the gradients of weights with respect to the loss is essential to optimising the weights. The standard algorithm for doing so is backpropagation, which is a numerical method that repeatedly uses the chain rule. The method was first derived in [210] and was later popularised in the context of neural networks by Rumelhart *et al*. [176]. Backpropagation is now an essential part of automatic differentiation systems built into deep learning frameworks like Pytorch and Tensorflow. I will cover the method in the following as it gives a better

understanding of neural networks. Let E be the loss function defined as a sum of squared errors computed by comparing the output from the network's final layer N, the output layer, and the desired output d

$$E = \frac{1}{2}\sum_s \sum_j \left(y_{j,s}^{(N)} - d_{j,s}\right)^2, \tag{2.12}$$

where s is an index over training samples, i.e. input-output pairs, j is an index over the output layer nodes and y is as defined in Equation (2.11). To find the gradients of the weights based on Equation (2.12), we consider the forward pass of a single training sample. By finding the weight gradients of the final layer's output nodes first, the backward pass is meant to propagate these derivatives back from the final layer to the first one. As such, the first step is to compute the partial derivatives in the final layer $\partial E/\partial y$ for each of the output nodes. Note that for simplicity, we will drop the sample index s and consider the derivation from the perspective of a single sample, say, s'. That is, the loss function in the following can be considered as $E|_{s=s'}$ and now differentiating Equation (2.11) for the sample, $s = s'$, yields

$$\frac{\partial E}{\partial y_j^{(N)}} = y_j^{(N)} - d_j. \tag{2.13}$$

Applying the chain rule, we can compute $\partial E/\partial x_j^{(N)}$

$$\frac{\partial E}{\partial x_j^{(N)}} = \frac{\partial E}{\partial y_j^{(N)}} \frac{\partial y_j^{(N)}}{\partial x_j^{(N)}} \tag{2.14}$$

Assuming the activation function that produces y_j is the sigmoid function of Equation (2.11), we can evaluate $\partial y_j/\partial x_j$ and substitute

$$\frac{\partial E}{\partial x_j^{(N)}} = \frac{\partial E}{\partial y_j^{(N)}} \cdot y_j^{(N)} \left(1 - y_j^{(N)}\right). \tag{2.15}$$

This equation makes it possible to determine how a change in the total input $x_j^{(N)}$ will affect the error by substituting in Equation (2.13). The total input is itself a simple linear combination of the outputs from the previous layer according to Equation (2.10), which makes it easy to "backpropagate" the gradients. The gradient of the weights can also be found by using the chain

rule

$$\frac{\partial E}{\partial w_{j,i}^{(N)}} = \frac{\partial E}{\partial x_j^{(N)}} \cdot \frac{\partial x_j^{(N)}}{\partial w_{j,i}^{(N)}} \tag{2.16}$$

$$= \frac{\partial E}{\partial x_j^{(N)}} \cdot y_i^{(N)}. \tag{2.17}$$

The gradient of a hidden node in the penultimate layer, $y_j^{(N-1)}$, is found by considering all its connections to nodes in the output layer via the linear combinations $x_j^{(N)}$

$$\frac{\partial E}{\partial y_i^{(N-1)}} = \sum_j \frac{\partial E}{\partial x_j^{(N)}} \frac{\partial x_j^{(N)}}{\partial y_i^{(N-1)}} \tag{2.18}$$

$$= \sum_j \frac{\partial E}{\partial x_j^{(N)}} \cdot w_{i,j}^{(N)}, \tag{2.19}$$

where the partial derivative of Equation (2.10) has been substituted. By further substituting Equation (2.13) and Equation (2.15) the gradient of a node in a previous layer can now be fully determined by the values in the subsequent layer, which makes it possible to complete the backward pass by reiterating the formula.

With a complete backward pass, the weights in a layer n can be updated with the gradient descent

$$\Delta w^{(n)} = -\lambda \frac{\partial E}{\partial w^{(n)}}. \tag{2.20}$$

Convolutional neural networks

A key advancement in the field of image recognition was the development of Convolutional Neural Networks (CNNs) [104]. Originating from feed-forward neural networks, CNNs are uniquely designed to preserve spatial information, making them especially suited to tasks involving grid-like data structures, such as images.

Unlike the architecture of a traditional multi-layer perceptron, where each neuron connects to every neuron in the preceding layer, CNNs apply filters that convolve with patches of neighbouring pixels. This method enables the network to capture local dependencies within the data and leverage these patterns across the entire input space. The number of filters, often described as the depth of the layer, is predetermined and not learned from the data, with typical depths being a power of 2 (32, 64, 128, etc.) due to hardware optimization benefits.

A significant enhancement in the design of CNN architectures came with the introduction of skip connections. These are links between non-sequential layers that allow the network to carry gradients through multiple layers without substantial degradation, overcoming the vanishing gradient problem, a major obstacle in training deep networks. This architectural feature forms the basis of Residual Neural Networks (ResNets) [70], contributing to their superior performance in various applications.

The seminal work by Krizhevsky, Sutskever, and Hinton, in which they proposed AlexNet, a deep CNN, played a pivotal role in bringing CNNs to the forefront of machine learning research [101]. This model with its eight layers outperformed other architectures at the 2012 ImageNet Large Scale Visual Recognition Challenge. AlexNet's success sparked a wave of research and development in deep CNNs, highlighting their powerful capacity for image recognition and other machine learning tasks.

In recent years, CNNs have become a cornerstone of machine learning, with applications extending beyond visual data to include domains such as natural language processing. This is attributed to the networks' ability to learn hierarchical representations from complex data, opening the door for many applications.

However, despite their successes, CNNs are inherently limited in their ability to capture long-range dependencies, particularly in sequential data like text, or non-local image features. This limitation has led to the rise of a new class of models known as Transformers.

Transformers

A transformer is a deep learning model that uses the concept of self-attention to weigh the significance of each part of the input data. The mechanism of self-attention was first proposed for natural language processing in [198] and has later been adopted for vision tasks in the vision transformer [42].

Intuitively, transformers can be thought of as the generalisation of neural networks where connections are not fixed but can be adapted via the self-attention mechanism. This implies that transformers can essentially generalise to a fully connected neural network should it be required for the task, but generally will find more efficient ways of connectivity similar to a CNN for vision tasks.

Self-attention is based on the attention mechanism which mimics cognitive attention. Attention is computed using a Query-Key-Value (QKV) model. For each part of the input, e.g. each word in a sentence or each patch in an image, three vectors are created: a Query vector, a Key vector and a Value vector. The vectors are formed by using three respective projection matrices on the input each of which consists of coefficients obtained through training. The attention score is then given by the dot product of the Query and Key vectors, and the

result is scaled by the Value vector [115]. For self-attention, the attention calculation follows the same approach but using $Q = K = V$ [198]. The vision transformer is a neural network architecture that uses the self-attention mechanism on images. More details on the application of the self-attention mechanism to SIM reconstruction is provided in 5.3.

2.4 Computer vision

Computer vision is a scientific field with the purpose of developing models for extracting useful information from images and videos to gain a high-level understanding of visual content. The inspiration for such models comes in large part from the human visual system, and indeed it is also a goal in the field to gain an understanding of how humans process visual information [191]. A common motivation for the study of computer vision is the automation of tasks that humans are able to perform, which holds especially true for tasks such as image classification, object detection and video tracking. In many cases however, computer vision offers unique solutions that can solve problems in a way that surpasses human capability.

This section seeks to define terms and concepts from the computer vision literature that are used throughout the thesis.

2.4.1 Image analysis with deep learning

Deep learning methods have become state-of-the-art in virtually all low-level computer vision tasks. In the context of microscopy the range of applications include:

- **Restoration and enhancement.** – improving degraded images by e.g. denoising (removing noise), super-resolution (increasing resolution), inpainting (filling out blanks), artefact removal and deblurring (making images less blurry) [225, 224, 108, 40, 109, 208].
- **Segmentation** – partitioning an image into respective parts each corresponding to a different type of object [173, 156, 31].
- **Reconstruction** – taking raw microscopy data and combining it into meaningful or super-resolution images. In the literature there are several cases of deep learning-based reconstruction methods for stochastic optical reconstruction microscopy (STORM) [150, 157, 16] as well as Fourier ptychography [90, 223]
- **Sample classification** – determining what is seen in the image, i.e. which classes from a set of possibilities are present in the image [69, 100].

2.4.2 Image restoration

Image restoration encompasses multiple low-level computer vision tasks which share the objective of improving image data by accounting for various means of degradation. Commonly, in particular in popular culture, the term ”image enhancement” is used to refer to the same thing as restoration, but a distinction is often made because *enhancement* tends to have a connotation of subjective or perceived image quality, which can involve subjective adjustments e.g. colour balance or introduction of convincingly looking artificial features. Restoration on the other hand focuses on the accuracy of the recovered image data.

2.4.3 Image super-resolution

The task of image super-resolution (SR) in the computer vision literature, as opposed to that of microscopy, refers to the problem of upsampling an image. A common distinction is between single image and multi image SR, which typically requires different approaches depending on the importance of frame-to-frame correlations in the data.

The problem of SR is an ill-posed because uniqueness in the solution cannot be guaranteed. When the task is addressed with a supervised machine learning model, the input-target pairs imply a 1:1 mapping between inputs and targets. However, in reality the mapping is multiple-valued. A low-resolution image can be explained by many high-resolution images because information on the exact positions and orientations of image features is lost in the low-resolution representation. The loss function used to train a neural network to learn the SR mapping function determines how the multitude of explanations are sampled. When using a mean squared error loss function, the trained network outputs an average of all plausible explanations, which results in spatial blurriness for the prediction [105].

2.4.4 Deconvolution

Deconvolution is a type of image restoration, where the objective is to inverse the effect of a convolution operation. In optics and microscopy, the operation that generally is aimed to be inverted is the diffraction caused by convolution with the point spread function (PSF). Deconvolution may also attempt to account for various distortions and aberrations by considering an effective point spread function of an imaging system. For instance, the Hubble Space Telescope had severe spherical aberrations in its first three years of operation resulting from flawed mirrors prior to its first servicing mission, and deconvolution methods were used to make the acquired images useful despite the design flaw [213]. In this thesis, a distinction is made between deconvolution and super-resolution reconstruction, although there is clear

overlap in their restorative purposes. Deconvolution deals more with correcting distortions and deblurring by using sharpening filters, whereas a super-resolution method is a restoration task more focused at utilising priors or multiple realisations of the same image to recover information that is otherwise lost.

2.4.5 Denoising

To formalise the problem of denoising, let us first consider a basic image formation model for a system with noise. The generation of an image, D, can be written as $D = S + N$, where S is the underlying signal and N is the added noise. A usual assumption in microscopy [102] is that images are drawn from a joint distribution

$$Pr(S \cap N) = Pr(S)Pr(N|S), \tag{2.21}$$

where $Pr(N|S)$ is the conditional probability of N given S, i.e. the noise that enters the final image depends on the signal corresponding to e.g. bright and dim regions having different noise statistics.

The probability distribution for the signal, $Pr(S)$, is an arbitrary distribution that is assumed to satisfy

$$Pr(s_i|s_j) \neq p(s_i), \tag{2.22}$$

for two pixels s_i and s_j of S that are local to each other. This means that the pixels of S are not statistically independent, from which it follows that a denoising method may utilise spatially correlated information to recover signal from noise. This stands in contrast to the noise that is assumed to have a conditional probability distribution given by

$$Pr(N|S) = \prod_i Pr(n_i|s_i). \tag{2.23}$$

From this it follows that given a signal S with pixels s_i, the pixels of the noise n_i are conditionally independent.

Depending on the noise sources in the system, the noise may have a zero-mean distribution

$$\mathbb{E}[n_i] = 0, \tag{2.24}$$

and therefore the pixels, d_i, of the final image $D = S + N$ have the expectation value

$$\mathbb{E}[d_i] = s_i. \tag{2.25}$$

This means that averaged over time the final image is identical to the signal. Hence, images acquired with longer exposure times will be cleaner, i.e. higher SNR, if we ignore the effects of photobleaching, sample motion etc.

Noise sources in microscopy. In microscopy, the primary sources of noise are additive Gaussian noise and Poisson noise. The Gaussian noise, which is generally zero-mean and not signal-dependent, predominantly arises from the electronics of the imaging system, such as the image sensor [108]. In contrast, Poisson noise is signal-dependent, stemming from the inherent quantum nature of light [93]. Its absolute impact grows with the strength of the signal, which makes it challenging to mitigate. However, the relative effect of Poisson noise, or noise-to-signal ratio, decreases as the signal increases. This characteristic of Poisson noise facilitates one of the simplest ways to diminish noise: increasing the exposure time. Longer exposure times allow for the integration over a larger number of photons, thereby averaging out the random fluctuations and reducing the relative impact of Poisson noise.

In situations where multiple independent images of a static sample are available, a pixel averaging scheme can effectively counter both types of noise. Because Gaussian noise is zero-mean, its contributions can cancel out when averaged across images. For Poisson noise, the random fluctuations diminish relative to the signal through this averaging process. For single image scenarios, a smoothing filter can be employed to reduce noise. This filter works by averaging pixels in close proximity, under the assumption that the true signal varies smoothly, while the noise does not.

However, as we will see in Section 3.4, even in the absence of clean references to learn from, machine learning approaches can provide superior solutions to this problem, transcending the capabilities of these simpler denoising methods.

2.4.6 Segmentation

Image segmentation is the problem of partitioning an image into segments, or regions, that share certain characteristics. The process of segmenting an image can also be considered a classification task, in which each pixel is assigned a label. The goal of segmentation is to transform images into a semantic representation that can be used for quantitative analysis such as locating objects or boundaries, counting certain entities or performing morphological measurements.

The output of a segmentation method is referred to as a segmentation map, which comprises the different segments that collectively cover the entire image. A segment typically represents a type of object or an instance of an object. The meaning of an object naturally depends on the context, but in a bioimaging context it could be cell nuclei or aggregates of a certain biomarker.

More generally, a segment can be any set of pixels that share particular attributes like intensity, colour, shape or texture.

Instance-based segmentation, as opposed to semantic segmentation, is the more advanced problem of distinguishing multiple occurrences of the same type of object by assigning them different sub-labels. As an example, a segment could represent a certain type of cell but would also have an index allowing it to be isolated during subsequent analysis from other occurrences of the same type of cell. In this thesis, I will only consider semantic segmentation.

Many segmentation methods exist ranging from simple thresholding based methods, as described below, to machine learning methods, e.g. the ImageJ plugin WEKA that uses simple models like random forests [123], and to more complex solutions relying on deep neural networks. With a push from scientific areas for which segmentation is essential, the performance of state-of-the-art methods has improved rapidly over the last decade, particularly owing to advances in biomedical science, e.g. [173], and autonomous driving, e.g. [4].

Thresholding. For gray-scale images that display high contrast between the classes for which segmentation is desired, e.g. bright shapes on a dark background, an intensity-based threshold value can be introduced to segment the classes apart. This is the simplest method of image segmentation and is called the thresholding method. The value chosen as the threshold needs careful selection, which can lead to the necessity of fine-tuning on a frame-by-frame basis. Various methods for automatic selection of the threshold value have been proposed, such as Otsu's method [155] which considers the intensity histogram of the image to determine a threshold value that maximises the variance between pixels that belong to different classes, i.e. the inter-class variance.

The thresholding method is prone to corruption from noise because it does not discriminate between individual pixels and groups of pixels. Thus, if the noise floor has a large variance, pixels in the tail of the noise distribution are likely to get assigned the incorrect class. As such, the thresholding method is generally only useful if the data has a high SNR, which is the case when it is used in Section 4.2. Although the thresholding method could be used with a denoising method, an image segmentation method that considers shape and morphology is a better choice for low SNR data.

Similarity to blob detection. Blob detection is a segmentation task that aims to detect regions of an image with similar brightness or colour. An example is the detection of bright spots on a dark background, such as the diffraction-limited blobs that are produced by single emitters in SMLM.

Blobs can be detected by thresholding the magnitude of second-order derivatives of pixel intensity values across an image. Second-order derivatives are calculated with the Laplacian operator. To increase robustness to noise, a Gaussian smoothing filter, g, is generally applied first, leading to a combined operator called the Laplacian of the Gaussian (LoG) [133, 116]. The LoG can be useful for representing features at different scales by adjusting the smoothing parameter, a concept referred to as scale-space representation. One such scale-space representation is a Gaussian scale pyramid. The LoG can be computed by using a convolution operation and the convolution theorem

$$\nabla^2(f * g) = f * \nabla^2 g, \tag{2.26}$$

meaning that the LoG of the image can be obtained via a single convolution with a pre-computed kernel, $\nabla^2 g$. This operation can also be useful for generating scale-space representations.

An alternative, more computationally efficient method is the Difference of Gaussians (DoG), where the Laplacian of the smoothed image is approximated by convolving the image with two Gaussian kernels of different standard deviations and then calculating the difference between them. The DoG operation is computationally cheaper than LoG as it is separable, i.e. the operation can be decomposed into two one-dimensional convolutions, and is often preferred for constructing Gaussian scale pyramids, cf. the SIFT algorithm [124].

A widely available implementation for blob detection utilizing both LoG and DoG operators is included in the Python library *scikit-image* and is used in several sections of this thesis.

2.4.7 Image quality assessment

The two most common metrics to quantify the likeness between an image and a reference image are peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [205]. When the reference image is the ground truth, i.e. an ideal representation of the image without obvious degradations, these relative quantities provide a measure of the quality of the given image.

The PSNR is on a decibel scale and ranges from 0 to infinity. It is defined as

$$\mathrm{PSNR}(x,y) = 10\log_{10}\left(\frac{1}{\mathrm{MSE}}\right), \quad \mathrm{MSE} = \frac{1}{N}\sum_{i=0}^{N-1}\left[x(i) - y(i)\right], \tag{2.27}$$

for two images x and y each consisting of N pixels that are assumed to be represented by floating point numbers ranging from 0 to 1.

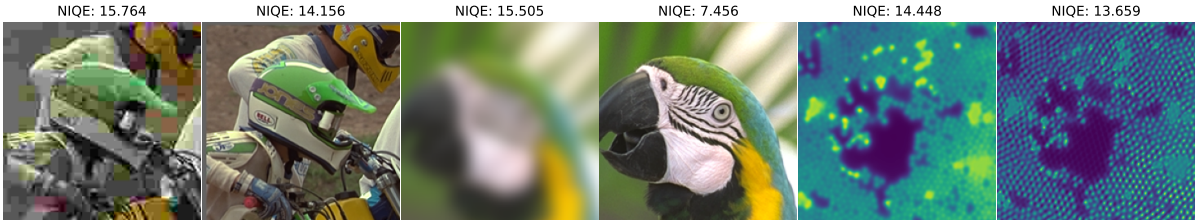


Figure 2.6: NIQE scores compared across three examples of inferior image quality ranging from pixelation and colour noise (left), blurring by convolution with a kernel (centre) and wide-field projection of a SIM stack (right).

SSIM takes on values from 0 to 1 and is defined as

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{2.28}$$

where x and y are two images of equal size, parameters $\mu_{x|y}$ are the means of the images, parameters $\sigma_{x|y}$ are the variances of the images and σ_{xy} is the covariance of the two images. The constants c_1 and c_2 are included to stabilise the division with weak denominator – by default they are set to small values.

To assess image quality in an absolute respect, also known as blind image quality assessment, the Natural Image Quality Evaluator (NIQE) score [142] has been proposed. A similar no-reference metric is proposed by [129]. In [204] a combination of the two measures is used to define a perceptual quality index, which when optimised for in an image reconstruction method can lead to distorted but more realistic looking output. This is described further in Section 3.2. Other ways to define a perceptual quality index is explored in [12].

The most widely used no-reference metric is NIQE, applied at various points in this thesis. It's worth noting that in NIQE, a lower score signifies superior image quality. An example of NIQE scores on three pairs of images is illustrated in Figure 2.6. Alternatively, the NanoJ-SQUIRREL (Super-resolution QUantitative Image Rating and Reporting of Error Locations) method [35] can be utilised for image quality assessment in bioimaging. This method calculates an error map, emphasising areas of poor agreement between a super-resolution image and its diffraction-limited equivalent. Generally, SQUIRREL is not sensitive to artificial sharpening, a typical artefact associated with certain super-resolution techniques, because the super-resolution image is converted into its diffraction-limited equivalent, which is then compared to the original image.

Chapter 3

Image Restoration

In this section we consider the problem of recovering information in degraded images. In the literature denoising is considered one of several tasks within image restoration, and it is particularly important for image processing in optical microscopy because the limited photon budgets tend to give rise to a high degree of degradation from multiple noise sources. Two predominant noise sources are Gaussian noise and Poisson noise. Gaussian noise stems from the electronic read-out noise from the imaging equipment, whereas Poisson noise derives from the discrete quantum mechanical nature of photon emission from fluorophores.

First an account of existing work in the field of denoising is given and a look at applications in optical microscopy. After this we consider the performance of supervised denoising methods on a partially synthesised image set to quantify to which extent the signal-to-noise ratio can be improved using deep learning. This is quantified in terms of exposure time assuming everything else is kept constant, thus asking the question of how much exposure times can be reduced while still retaining the same level of image quality.

After this, we turn to unsupervised, semi-supervised and self-supervised denoising methods. Demonstrations of the application of these methods include evaluations on sample images received from respective collaborators working in optical microscopy, cryogenic electron microscopy and astronomy.

3.1 Literature review

As described in 2.4.2, the area of image restoration covers different low-level computer vision problems including denoising and super-resolution (SR). Each section will go more in depth with the respective literature that applies to topic of the section. Therefore, in this section I will rather lay out some fundamental advancements and directions in the restoration field that has benefited both disciplines.

Early work focused on local methods, in which a pixel's value is determined as a function of neighbouring pixels. For super-resolution such methods would be considered interpolation schemes [94], whereas for denoising the classical approach is filtering methods and smoothing kernels [117].

The natural improvement over local methods is to consider image data elsewhere in the image to gain better insight into the noise distribution or characteristics of the displayed objects' shape and texture. Popular methods include non-local means [23] and derivatives for SR [166]. Wavelet theory is an alternative approach to utilise information beyond the immediate neighbourhood of a pixel by making use of the discrete wavelet transform [158]. Additional information can also be extracted from spatially or temporally correlated images of an object. This has given rise to video super-resolution (VSR) [91], multi-image super-resolution [111] and multi-frame denoising techniques [226].

A great advancement in the field has followed from the advent of machine learning techniques especially using supervised training strategies. Fully-connected multi-layer perceptron networks have been shown to be able to match the performance of classical methods [164], while convolutional neural networks consisting of only a few layers were able to claim state-of-the-art performance several years ago [40].

The depth of neural networks has been steadily increasing while their performance for restoration tasks have increased. Residual neural networks [70] enabled very deep CNN architectures and methods employing these architectures were able to significantly surpass the previous arts [96, 114, 105].

Most recently, transformer networks have started overtaking previous state-of-the-art CNN methods. Vision transformers are able to encompass non-local information beyond the receptive fields of deep CNNs. This is possible because transformers do not rely on kernels of a specific size, but rather use the attention mechanism as described in 2.3.1 to consider global information during training. This has enabled methods such as SwinIR [113] to claim state-of-the-art for super-resolution and denoising.

In recent years there has also been an increasing focus on methods that are not fully supervised, such as Noise2noise [108], described later in this chapter, and [128].

In the bioimaging literature, some early influential works include deconvolution methods [49, 39] and non-local denoisers [37, 15]. Machine learning has also seen a rapid adoption with methods such as content-aware restoration (CARE) [207] for denoising and single image super-resolution [202].

3.2 Image super-resolution

A large body of research exists in the field of single image super-resolution. Traditional computational approaches are primarily based on interpolation schemes such as bilinear, bicubic or Lanczos [43] interpolation. Bicubic interpolation is often used as an efficient approximation of Lanczos resampling, for instance when resizing images in popular software programs ranging from Microsoft Word and Adobe Photoshop to ImageJ. Bicubic upscaling can thus be considered the de facto standard in image rescaling, and it will be used as a baseline in the following.

In recent years the literature of SR has split into two directions: one dealing with achieving the best possible reconstruction errors and another focused on producing the most perceptually pleasing and convincing (referred to as having low perceptual loss) output to a human observer. The reason those two directions are not reconcilable is that the reconstruction errors typically defined by the mean squared error tend to give optimal solutions that do not contain high frequency content but rather appear somewhat blurred or washed out when compared to an original. For the recovered image to contain high frequency features it is necessary to artificially generate features that are not at all in the low-resolution input. This is in general achieved using Generative Adversarial Networks (GANs) [54] with an approach pioneered in [105]. The state-of-the-art methods include SRGAN [105], SRFeat [160], ESRGAN [204] and EnhanceNet [179] – all of which are based on a GAN. The state-of-the-art methods for achieving the best

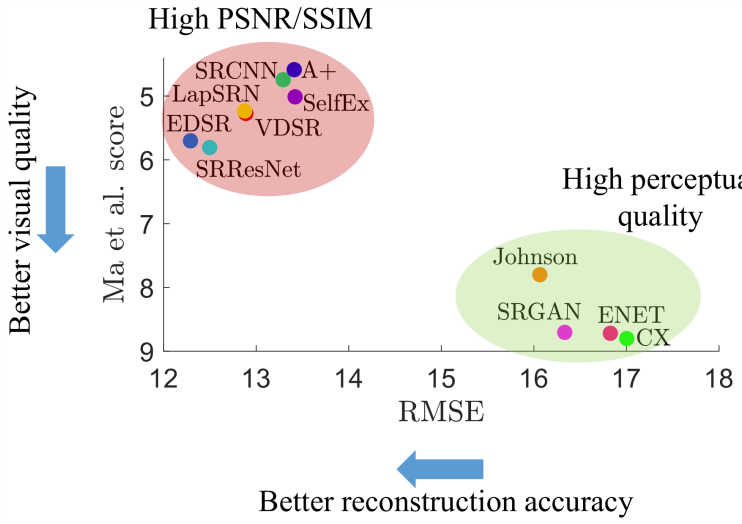


Figure 3.1: Trade-off between reconstruction error and perceptual loss for state-of-the-art methods [12]. Note that models from publications in 2018 are not included.

reconstruction error include SRResNet [105], EDSR [114] and the recent EPSR [197] and RCAN [227].

The trade-off between reconstruction error and perceptual loss of various state-of-the-art methods is summarised in Figure 3.1. A comparison of two current state-of-the-art CNN models in the respective camps, RCAN and ESRGAN, are shown on Figure 3.2. The texture clearly looks more realistic and of higher fidelity in the output of the ESRGAN, while the performance score – peak signal-to-noise ratio (PSNR) in units of dB (introduced formally in Section 3.3) – is higher for RCAN. This is because several of those strands of hair in the ESRGAN are simply made up, which should be evident when comparing closely to the high-resolution (HR) ground truth image.

If the purpose of a super-resolved image is to be used for analysis in quantitative research, then it does not seem appropriate to distort the image data, i.e. generate high frequency features, to make it look more realistic, because in the end the user likely prefers to be confident about what the image shows rather than having an artificially realistic image. Therefore, this PhD project has so far focused on methods that obtain minimal reconstruction errors; the methods in the red ellipsis of Figure 3.1.

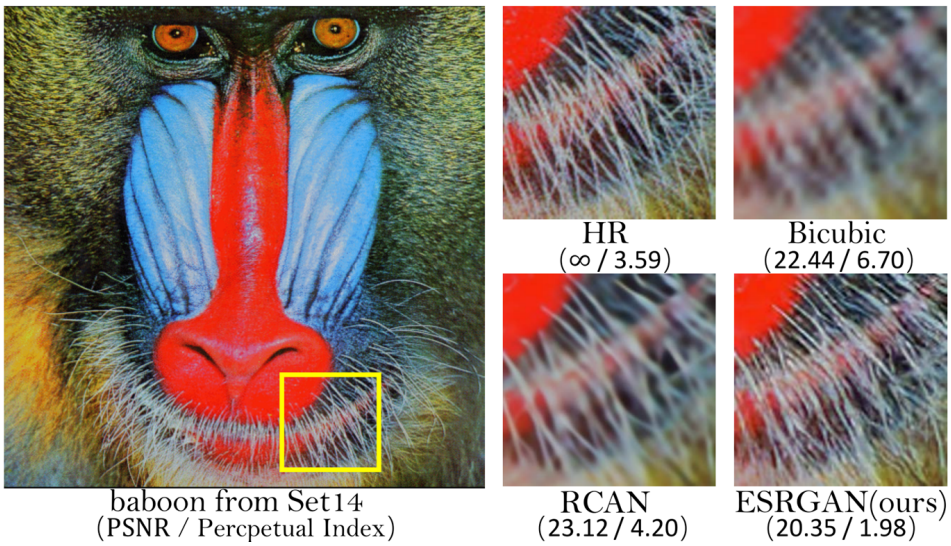


Figure 3.2: Example output of ESRGAN, which uses a Generative Adversarial Network (GAN) architecture to distort the input image to approximate the high-frequency textures. Image credit [204].

3.2.1 Datasets

Models have been tested with different popular benchmarking datasets such as ImageNet [177], DIV2K (DIVerse 2K resolution high quality images) [2] and BSD (Berkeley Segmentation

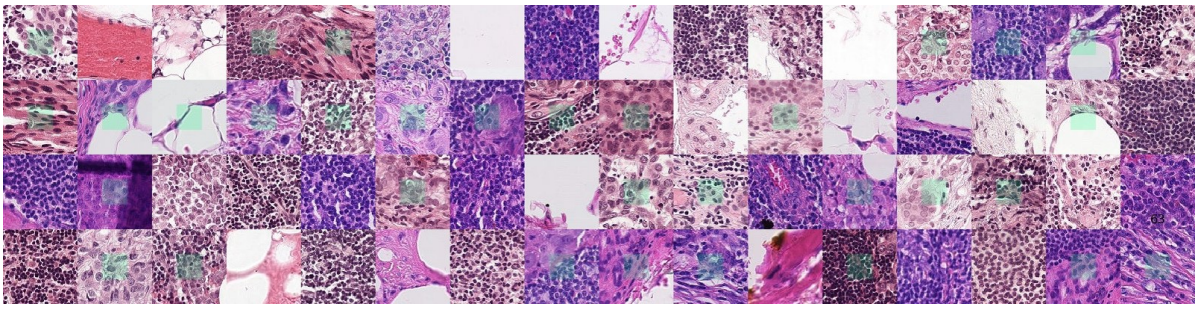


Figure 3.3: The PatchCamelyon (PCam) benchmarking dataset available on GitHub [199].

Dataset) [134]. But in the interest of training and evaluating models on relevant data, microscopy image data acquired from members of the host group (using structured illumination microscopy and light sheet fluorescence microscopy) has also been considered and will be discussed in Section 3.3. However, due to the necessity of a very large quantity of diverse training samples when training deep models, the data from the host group is not currently enough. To ensure that trained models generalise better, a large bio-image dataset called PatchCamelyon (PCam) [199] is used for the moment. PCam consists of bright-field microscopy images of lymph nodes from histology. A random sample of images from the dataset can be seen on Figure 3.3.

3.2.2 Single image super-resolution

Four different models of the ones previously mentioned have mainly been tested: SRResNet, SRGAN, EDSR and RCAN. The method RCAN is a recently proposed model that is found to have very good reconstruction performance but also is quite computationally demanding. Each of the four models are trained on 32000 images from the PCam dataset for at least 30 epochs (training iterations of the entire dataset) using ADAM [97] as an optimiser with a learning rate of 1e-5. The trained models are then evaluated on 50 independent test images also from PCam. The average performance scores measured by the two metrics PSNR and structural similarity index (SSIM) [205], both formally introduced in Section 3.3, can be seen in Table 3.1.

PCam	bicubic	SRResNet	SRGAN	EDSR	RCAN	HR
PSNR	17.38	18.59	18.60	18.76	**19.43**	∞
SSIM	0.433	0.616	0.614	0.610	**0.657**	1

Table 3.1: Comparison of different methods for 16x image upscaling and the original high-resolution, HR, on the benchmark dataset PCam. The methods are bicubic interpolation, SRResNet [105], SRGAN [105], EDSR [114] and RCAN [227]. RCAN is observed to have the best performance on the test set in terms of the metrics PSNR [dB] and SSIM.

Two examples of an input image recovered by bicubic upscaling and compared to a prediction from the trained RCAN model can be seen in Figure 3.4.

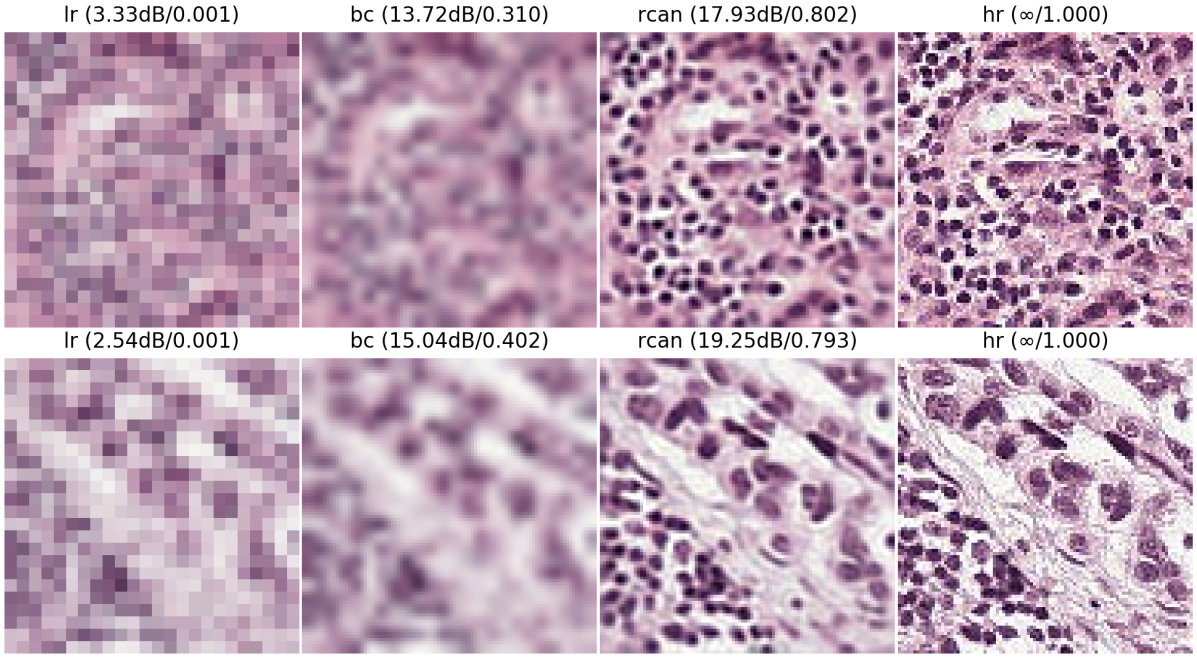


Figure 3.4: Two different test images that are both 4x super-resolved (in each dimension, so 16x in terms of pixel count) from a 24×24-pixel input image to a 96×96-pixel output. From left to right: input low-resolution image (shown here upscaled with simple repetition), image upscaled by bicubic interpolation, the model prediction and the unseen high-resolution image.

A direct comparison of an individual output of the four tested models is shown in Figure 3.5. The SRGAN is found to perform very similarly to SRResNet, which is because they are based on the same model, but have different loss functions. The very similar performance indicates that the loss function of SRGAN has not been configured properly in the test to allow enough distortion for the GAN to really shine. The EDSR model is an improvement of SRResNet and as expected we do see somewhat better performance in Table 3.1. The RCAN model performs significantly better than the other methods, which is impressive given that it was the model trained for the least number of epochs, namely 30 vs 100 for the others (a limit of 10 hours of computation on the ComputerLab GPU cluster was set, and the training of RCAN did not advance further in that time window).

3.3 Supervised denoising

This section reports on methods developed in the first year of the PhD project. The work was presented orally at Focus on Microscopy (FOM) in 2019. The methods are trained in a

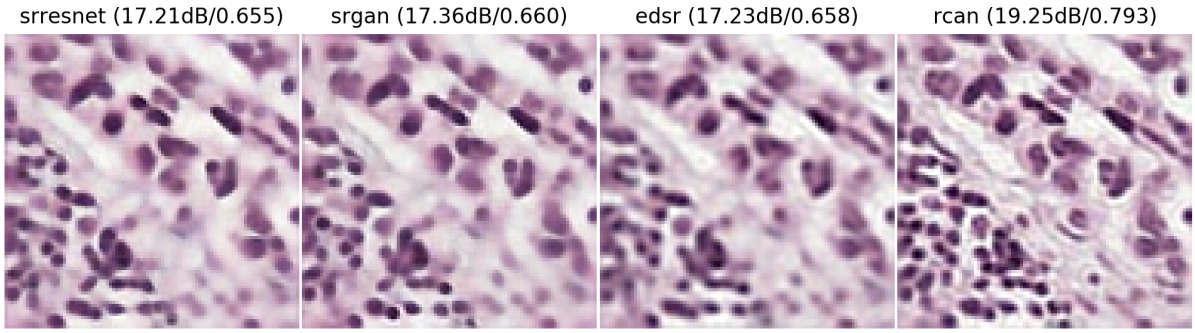


Figure 3.5: Comparison of predictions from state-of-the-art learning-based SR methods for 4x upscaling in each dimension. For the original high-resolution image see Figure 3.4.

supervised fashion for denoising of microscopy images to improve low-light imaging capability. As outlined in Section 1.1, numerous use cases require operating with a highly limited photon budget. One question that this section seeks to illuminate is to what extent state-of-the-art denoising methods can improve low-light imaging performance and advance current use cases.

3.3.1 Related work

The literature on image denoising has traditionally been based on local averaging approaches, such as the application of a Gaussian smoothing filter [23, 117]. Other local filter methods include least mean squares filter [68], anisotropic filters [162] and in the frequency domain; Wiener filters [216] and wavelet thresholding methods [41].

Local methods are computationally light, but have obvious limitations. First, the averaging often involved in local methods introduces blur, which is a degradation by itself, rendering features to be less defined. Secondly, they do not perform well for high noise levels, since the correlations between neighbouring pixels deteriorate [186].

Non-local filters solve some of these problems by using self-similarity of natural images beyond neighbouring pixels [186]. The first method to propose this is the non-local means method [23], in which patches are restored by weighted averaging of all other patches in an image. Since then a number of improvements have been proposed such as invariance to patches that are rotated or mirrored with respect to each other [57], and improved computational efficiency, automated parameter tuning and extension to 3D image stacks [33]. Although the non-local filters are better at high noise levels, they will typically lead to artefacts like over-smoothing [186].

Another category of denoising methods that are distinct to the ones previously mentioned is learning-based methods. The first learning-based methods to become a trend in denoising were sparse dictionary learning methods that attempt to find sparse representations of the

input data in the form of linear combinations. The methods perform denoising by expressing an image patch in the denoised image as a linear combination of other patches in a trained redundant dictionary consisting of many patches obtained from an image dataset [186]. An example of this type of method is the K-SVD method that uses K-clustering with singular value decomposition [186, 45].

More recently, supervised learning has taken over with the emergence of deep learning, and several end-to-end convolutional neural networks (CNNs) have been proposed for denoising. These will be discussed briefly in the following section.

3.3.2 Denoising based on deep learning

A pioneering deep CNN for image analysis is the U-Net [173]. The model was originally intended for segmentation, but it has seen use for restoration tasks such as inpainting [109] and in particular denoising [108, 208].

The central idea of U-Net is that an input image is taken through convolution layers at different resolutions, see Figure 3.6, while employing so-called skip connections at every resolution. After the image is passed through three convolution layers, a pooling operation is used to lower the resolution and this sequence of convolution layers and pooling repeats until a certain number of levels is reached. The pooling operation can be defined in different ways, but typically max-pooling is used with a window width of 2, meaning that each 2×2-pixel segment is reduced to the maximum value of the pixels in the patch. The lower resolution at deeper levels greatly saves computational load, and consequently the number of filters in convolution layers can be increased without causing the training time to increase significantly.

The skip connections in U-Net that are seen as the horizontal lines in the diagram of Figure 3.6 pass intermediate results to subsequent layers. These connections prove to be crucial for a robust convergence during training by avoiding the vanishing gradient problem. This has been further investigated in [132], in which another deep CNN was proposed that has also been used for denoising [108], and it was shown that the skip connections lead to restoration performance gains.

The original U-Net architecture has five levels in this way, but it can be customised for better or worse, such as the more light architecture in [108] with only four levels and fewer convolution filters in the convolution layers. This lighter architecture is referred to as UNet-N2N (N2N meaning Noise2noise after [108]) for the remainder of the chapter. For the numerical experiments presented later, one of the models will be this one. Another variant that will be considered is a customised, heavier version of U-Net with six levels of resolution and more than double the number of filters at the lowest level compared to the original architecture. This model will be referred to as UNet-60M, since it has about 60 million trainable parameters,

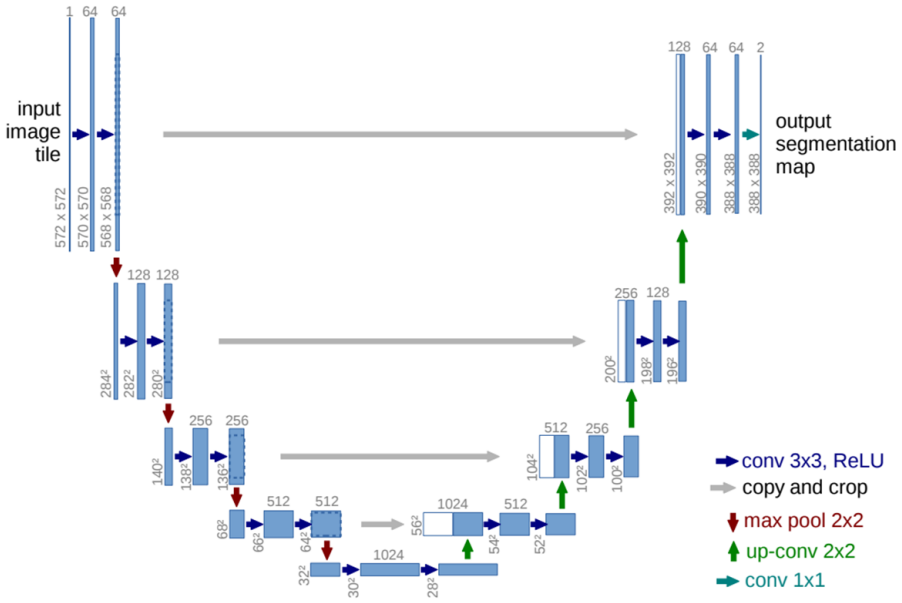


Figure 3.6: Convolutional neural network based on the U-Net architecture [173].

whereas UNet-N2N and the original architecture have approximately 1 million and 13 million parameters, respectively.

3.3.3 Leveraging super-resolution architectures

While the U-Net architecture provides efficiency and robustness, one might wonder how much the restored output suffers by having the majority of computations done on downsampled versions of the input image. The architectures employed in super-resolution neural networks tend to be different. As argued in Section 3.2.2, the field of single image super-resolution (SR) research has seen more activity than that of denoising. The state-of-the-art methods perform very well when trained on microscopy images as indicated by Figure 3.4 and Figure 3.5.

These results have motivated an experimentation in this PhD project of customising the super-resolution models to perform denoising rather than upsampling.

The state-of-the-art SR architectures generally do not have downsampling between layers [40, 105, 114, 227], however they alleviate training by following the structure of residual networks as first introduced with ResNet [70] with image classification in mind and later repurposed for restoration with SRResNet [105]. Residual networks use the previously mentioned skip connections but more rigorously by having the shortcuts after every few stacked convolution layers, which then constitutes the residual building block that can be repeated many times. The residual networks allow for training of very deep networks, which was demonstrated in [70] with an appropriately termed ”aggressively deep model” consisting of 1202 layers

that was trained with no optimisation difficulty, although such networks have a large risk of suffering from overfitting thus needing careful regularisation.

The design idea of residual networks was taken one step further in Enhanced Deep Residual Networks (EDSR) [114] by proposing a modified residual building block called ResBlock, which was found to be superior to the previously proposed and more directly adapted ResNet model called SRResNet [105]. A diagram showing the EDSR network can be seen on Figure 3.7, which includes a block that has simply been coined ”Flexible” since it varies with the different purposes it has successfully been customised for during the research of this PhD project.

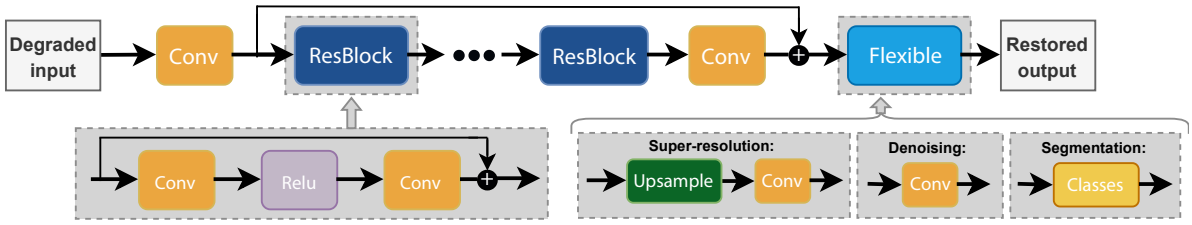


Figure 3.7: EDSR model

Yet another improvement in this class of network architectures was made with Residual Channel Attention Network (RCAN) [227], which augments the ResBlock with two more convolution layers and a global pooling operation. These extra layers are combined with the layers in the standard ResBlock by a skip connection that does not combine old and new results by addition but by multiplication, which the author argues allows the model to learn to adaptively rescale channel-wise features by considering the interdependencies among channels. RCAN also adds more regular skip connections in a design that is coined Residual in Residual (RIR), where a preset number of ResBlocks are considered a group, and a long skip connection is made from the beginning of the group to the end, whereas the skip connection of each ResBlock only passes by a few convolution layers. The author believes this allows abundant low-frequency information to be easily passed on, thus enabling the main network to focus on learning to restore high-frequency information. The benefit of not only having short skip connections but multiple long skip connections is supported by an earlier study [132].

The *Flexible* block of Figure 3.7 is set to be a single convolution layer for denoising tasks, since the image resolution of input and output is equal. Thus, the only thing required by this final convolution layer to produce the model output is to take the numerous channels of the ResBlocks feature maps and reduce them to the desired number of output colour channels, i.e. 1 for grayscale or 3 for RGB, which is achieved by configuring the convolution layer to have the same number of input channels as the feature maps while only applying 1 or 3 trainable filters. For super-resolution the upsampling is ideally done with a fractional convolution layer

(also known as a transposed convolution layer) that has a stride, e.g. a stride of 2 in a fractional convolution layer will give a double resolution. Alternatively a so-called pixel shuffle operation can be used, which is another way to perform sub-pixel convolution with fractional strides. The cheaper way to do upsampling without any extra trainable parameters is to interpolate image, such as bicubic interpolation, before feeding the result to the final convolution layer.

3.3.4 Supervised training dataset via variable exposure time

To assess how much these methods potentially can improve low-light imaging operation, a simple investigation into how noise depends on exposure time was first conducted. As a metric to quantify degradation, the structural similarity index can be used. A wide-field fluorescent microscope was used to acquire several hundreds of images of different parts of a fixed sample of actin with both short and long exposure times, from 5 ms to 200 ms.

This dataset consists of pairwise low-quality and high-quality images that could be used as noisy inputs and clean targets to train a model in a supervised manner. However, the dataset is not large nor diverse enough for training a deep model such that it becomes generalised – overfitting is very likely to happen unless thousands of diverse training samples are available.

For this reason, the PCam dataset, as introduced in Section 3.2.1, is used for training. The aim is to use the original dataset to obtain statistics for the noise that can be used to degrade the PCam dataset to estimate improvements in acquisition rate and to obtain a generalised model that can even denoise the original dataset.

In order to use the PCam dataset for training, it is desirable to have a good approximation of the noise sources governing the original data. A noise model that includes Gaussian and Poisson noise, which is introduced more formally in Section 4.2.1, is tuned to match the experimentally acquired data as closely as possible. On Figure 3.8, an example image in the acquired dataset in its low-quality and high-quality versions are shown with an approximately matching synthetically noised version.

3.3.5 Quantifying potential gains in acquisition speed

An example of a section of the sample in the acquired dataset with different exposure times is shown in Figure 3.9. Using the image with exposure time of 200 ms as a reference, it is seen as expected that the SSIM approaches zero as the exposure time becomes shorter.

The tendency of the SSIM as a function of the exposure time when averaged over all images in the original dataset can be seen to the left in Figure 3.10. As a simple model of how SSIM varies, a two-term exponential function $\mathrm{SSIM}_{\mathrm{fit}}(\tau) = a \cdot \exp(b\tau) + c \cdot \exp(d\tau)$, where τ is exposure time, is used. By fitting the parameters a and c to the data, the relationship between

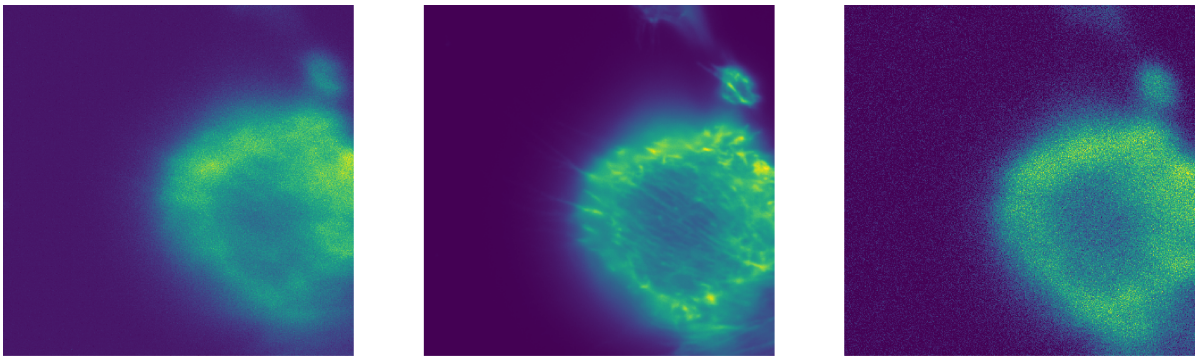


Figure 3.8: Approximating degradation with synthetic noise model. Experimentally degraded 5 ms exposure time image (left), high-quality 200 ms exposure time image (centre) and synthetically degraded 5 ms exposure time image with matching SSIM (right).

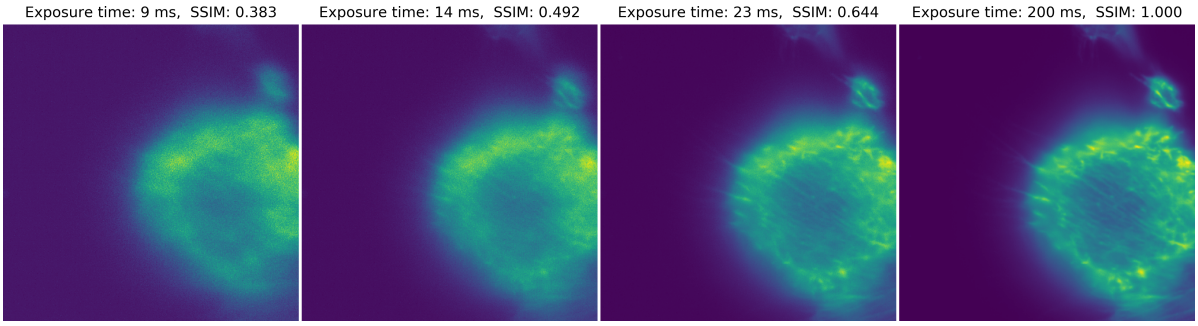


Figure 3.9: Image quality as a function of exposure time.

SSIM and exposure time can be modelled analytically. The fit is also plotted on the figure and is seen to closely follow the trend.

When degrading the high-quality images with the noise model described above with different levels of noise, a similar tendency appears as can be seen on the right of Figure 3.10. An analytical description using a fit of a two-term exponential function is obtained again, i.e. $\mathrm{SSIM}_{\mathrm{fit}}(\eta) = a \cdot \exp(b\eta) + c \cdot \exp(d\eta)$. The two functions for SSIM can now be set equal to correlate exposure time and noise level, i.e. $\mathrm{SSIM}_{\mathrm{fit}}(\tau) = \mathrm{SSIM}_{\mathrm{fit}}(\eta) \Rightarrow \tau(\eta)$, such that equivalent exposure times for a given noise level can be estimated.

3.3.6 Implementation, performance and results

Being able to approximate the effect of noise under low-light conditions, the PCam dataset is used for training with synthetically degraded inputs corresponding to 5 ms exposure. The models implemented are the modified super-resolution models EDSR and RCAN as well as

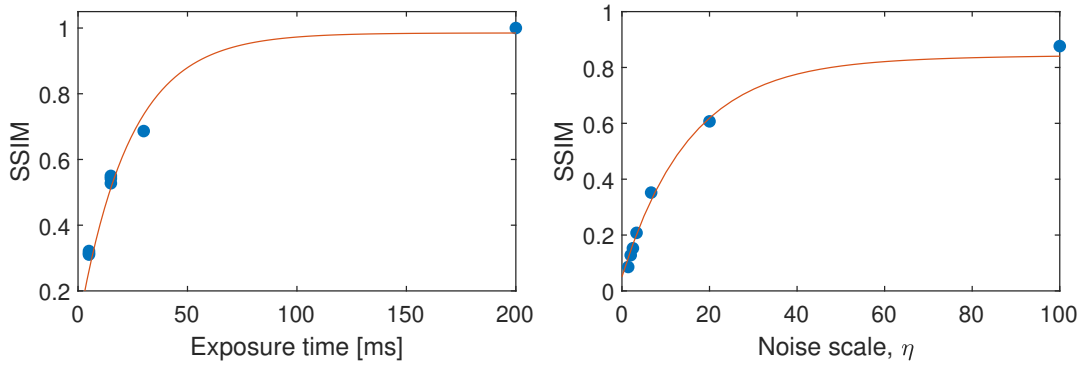


Figure 3.10: Correlating exposure time and noise level via structural similarity index. A two-term exponential function is fitted to each series to provide an analytical description of the tendency.

three variants of U-Net: the original architecture, a lighter architecture referred to as UNet-N2N and a heavier customised architecture referred to UNet-60M.

The number of trainable parameters and the memory usage of each model is depicted on Figure 3.11. The SR models have a relatively low number of parameters of about 1 million, whereas the original and heavy U-Net models have many times more. This is clearly reflected by the memory usage for storing the parameters. However, the memory usage when feeding an input forward through the network, as well as backpropagating the resulting error, is actually lower for the U-Net models, because most computations for these models are done on downsampled versions of the input.

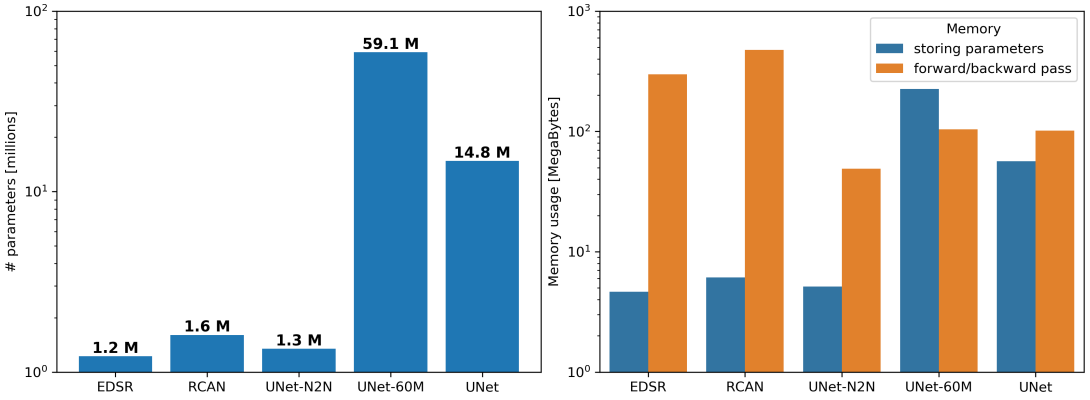


Figure 3.11: Number of trainable parameters in models (left) and their respective memory consumption (right).

Training is done on 30000 training samples from the PCam dataset using the 5 ms synthetically noised inputs. The learning rate is initially set to 10^{-4}, and is halved for every 5 epochs. After every epoch the peak signal-to-noise ratio (PSNR) is evaluated on a separate test set of 100 images.

The convergence of the PSNR during 30 epochs of training can be seen on Figure 3.12. Both SR models are seen to perform well. EDSR tends to match the more parameter-heavy U-Net models. The much heavier U-Net model does not add much in performance. Although it is more computationally expensive, RCAN performs significantly better at around 0.5 dB higher PSNR compared to the U-Net models.

In terms of training time it is clear that the U-Net models save time by exploiting down-sampling. Training for 30 epochs on a GPU cluster the U-Net models take 1000, 1500 and 2400 seconds for UNet-N2N, UNet and UNet-60M, respectively. On the other hand EDSR and RCAN take 4000 and 7000 seconds, respectively. RCAN thus takes the most time, but it also performs best while having a relatively low number of parameters.

An example of a restoration output comparing RCAN and U-Net are shown on Figure 3.13 with a smoothed version as reference. Both model outputs are clearly great improvements from the input resembling the target, i.e. the unseen ground truth, closely. As expected from the test results during training, the RCAN model performs better – again with a PSNR at about 0.5 dB higher than for U-Net. Looking more closely at the two model outputs reveals significant differences in the details of certain features – note green disks in figure. Some features are not recovered at all by the U-Net model, while others appear more washed out.

Based on this example, the image quality of the input image can be converted into its estimated equivalent exposure time of 2 ms, whereas the restored output from the RCAN model corresponds to 33 ms exposure time. This equals a 15 times higher frame rate if one was aiming

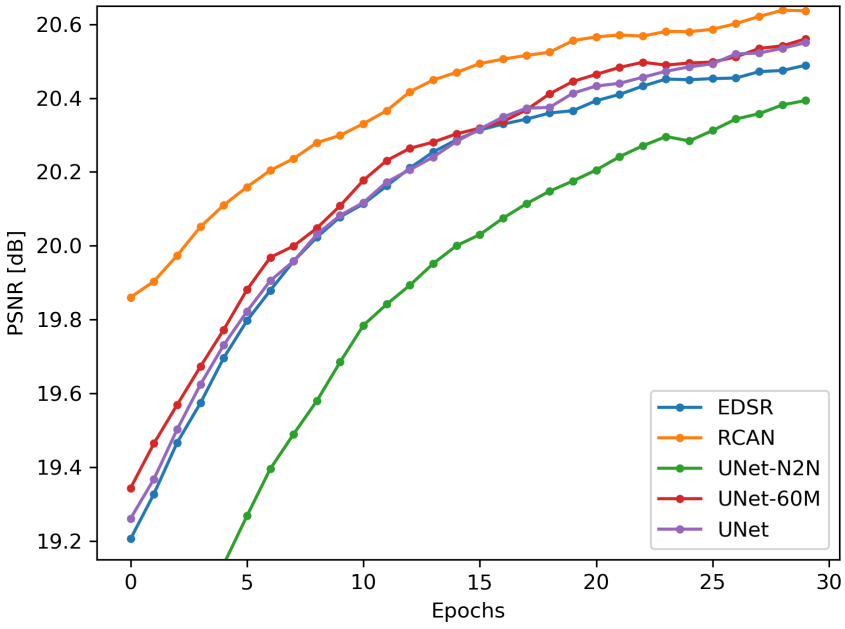


Figure 3.12: Performance of models on test set during training.

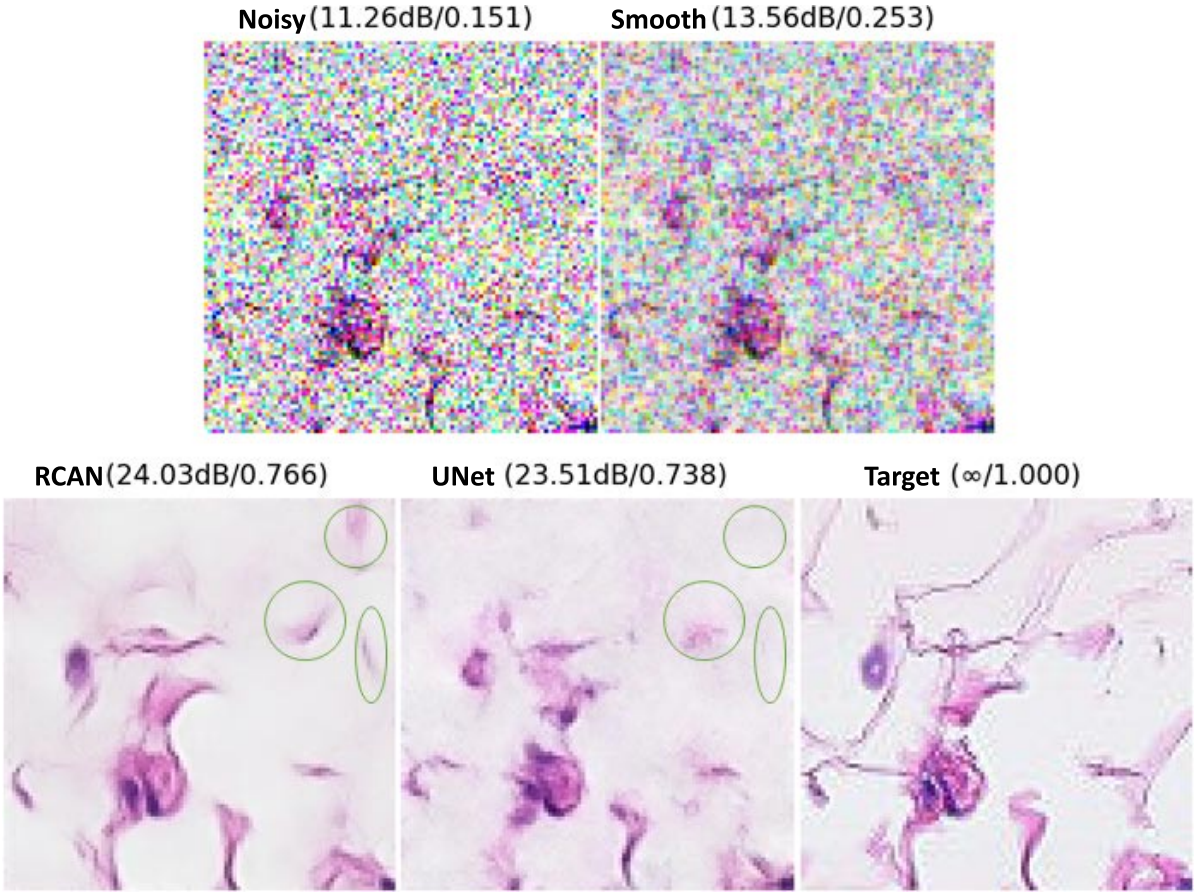


Figure 3.13: Example outputs from models. The output of RCAN has a better PSNR score, and considering the features within the green disks, it is evident that the U-Net model has not managed to resolve the same details as the RCAN model.

for the quality at ≈30 ms exposure time but operating at 2 ms exposure time. A similar figure of 15 times improvement is obtained when evaluating on an ensemble.

3.3.7 Usefulness for quantitative analysis

Going beyond looking at the quality scores, another means of validation could be to consider the benefits for the purpose of performing quantitative analysis. A simple task to quantify an image of a biological sample could be to count the occurrence of some objects in a frame. The PCam dataset consists of a large part of images of cell nuclei. Nuclei are easy to count with a blob detection algorithm. A standard difference of Gaussians blob detection algorithm was used to count nuclei at different exposure times.

For a random sample at 200 ms, 178 nuclei were detected in this way. As shown on Figure 3.14, the number of detected nuclei decreases as the exposure time becomes shorter,

since the degradation due to noise starts to corrupt the shape of nuclei. For this random sample, the count is 127 nuclei at 7 ms exposure time, meaning that almost 30 % of the nuclei now fail to be detected. This misclassification introduces a substantial number of false negatives which would have implications on the outcome of any downstream analysis.

When evaluated on an ensemble, it is found that this misclassification rate is about 21 % for the given noise level. While only the overall misclassification rate is considered here, other metrics such as precision and recall could also provide valuable insights into the balance between false positives and negatives, potentially giving a more complete understanding of detection performance.

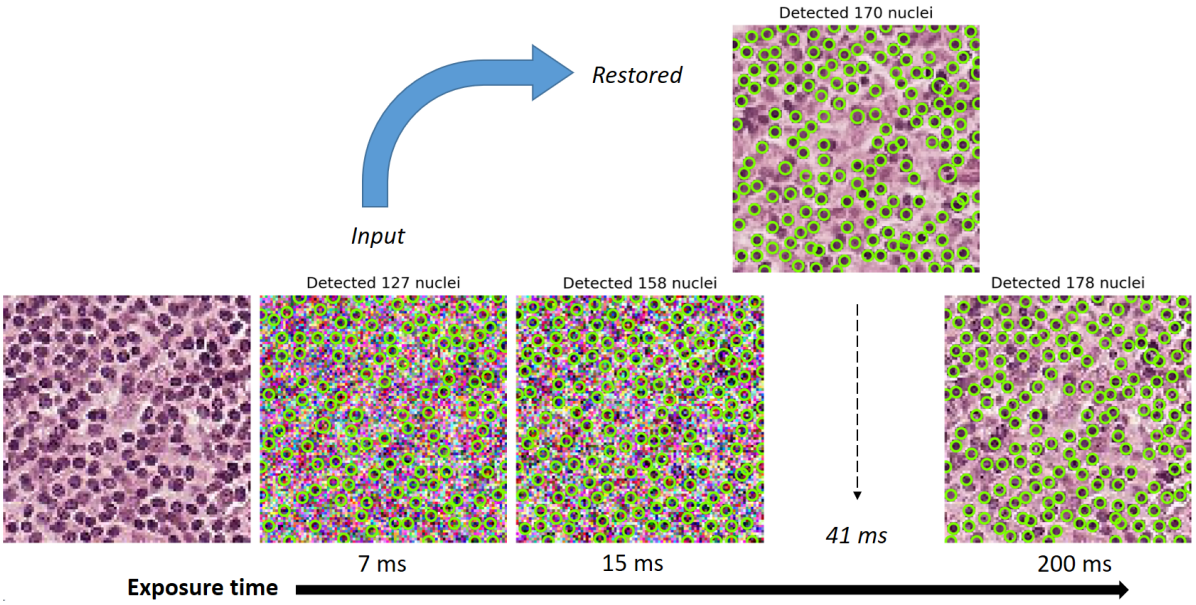


Figure 3.14: Relationship between exposure time and nuclei detection. The decrease in detected nuclei with shorter exposure times illustrates the impact of noise on the misclassification rate.

However, the restored image of this example yields a count of 170, indicating that the misclassification rate drops to just 4 %. The estimated equivalent exposure time of the restored image is 41 ms up from 7 ms. Hence, the improvement of the misclassification rate is found to be as significant as that of the frame rate, indicating that the restoration not only improves quality scores but also has a more tangible practical benefit.

3.3.8 Generalisation from synthetic to real-world data

The results reported thus far in this section suggest significant improvements in low-light imaging when using deep neural networks. However, these outcomes underscore the necessity of a well-curated training dataset to attain this level of performance. An important question then

arises concerning the applicability of these methods: Can a model, trained on a synthetic noise source that is applied to a clean benchmarking dataset, perform well on completely distinct, real-world datasets?

While this scenario carries elements of transfer learning, explored further in Section 5.1, it is more accurately characterised as a test of the model's ability to generalise from synthetic training data to real-world testing data. This process, sometimes referred to as out-of-sample testing in certain contexts, is highly valuable when acquiring a representative real-world training dataset is challenging. It allows us to assess the robustness of the model when handling different types of data, providing important insights into its performance beyond the initial training domain. An example of this can be seen in Figure 3.15, which reveals a reasonable restoration of the original image when compared to a high-quality version with a 200 ms exposure time.

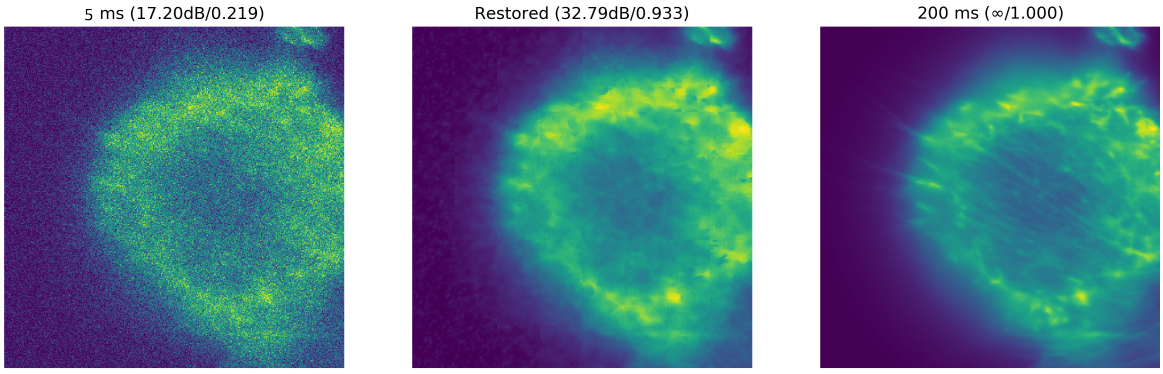


Figure 3.15: Testing generalisation from synthetic to real-world data by applying the trained model to the original image dataset. The training dataset consisted of entirely different sample types augmented with synthetic noise.

The restored image quality has an equivalent exposure time of 65 ms, while the input was 5 ms, almost achieving the frame rate improvement found for the PCam dataset of approximately 15 times. However, upon closer inspection, it is evident that while some features have been resolved, the characteristic stripes of actin are not fully recovered. This is likely because the synthetic training dataset was primarily composed of nuclei images, which has led the model to be proficient at recovering spots but not necessarily stripes. This difference in performance underscores the challenge of generalising from synthetic to real-world data, which is a key focus of Section 5.1.

3.4 Self-supervised denoising

Self-supervised denoising is a useful strategy to training a denoising model when clean targets are unknown or hard to acquire. I will cover a few techniques in this section that have been investigated for use in different areas of scientific imaging during my project:

- Cryogenic electron microscopy in collaboration with Helen Foster from MRC Laboratory of Molecular Biology, Cambridge
- Astronomy images for detecting transits in exoplanet study in collaboration with Peter P. Pedersen from Institute of Astronomy, University of Cambridge
- Fluorescence microscopy:
  - Calcium imaging in collaboration with Miranda Robbins from Department of Zoology, University of Cambridge
  - Kymograph imaging in collaboration with Lucia Wunderlich from my own group Laser Analytics Group in the Department of Chemical Engineering and Biotechnology

The primary methods studied in this section include a traditional method included as a baseline, ND-SAFIR [15] by Jerome Boulanger, and as for deep learning techniques I am focusing on variations of the method Noise2noise (N2N) [108], while also pointing to related methods.

3.4.1 Noise2Noise and variants

To avoid the problem of obtaining clean targets in the context of training a deep learning model for denoising in a supervised manner, one pioneering proposal in the literature is Noise2Noise (N2N) [108]. The method takes noisy images as input like any denoising method, but the same noisy images are also used as targets during training. Despite this, the method is demonstrated to achieve results that are similar to a method that has been trained with clean targets. However, to avoid having the network simply learn an identity mapping of the input, the input and target must be independent with respect to noise but have the identical underlying signal, e.g. a static sample imaged more than once.

To understand why this works let us first consider how traditional supervised training can be formulated as a problem, while keeping the assumptions and notation of Section 2.4.5 in mind. For concreteness, it is assumed that we are interested in training a fully convolutional neural network (CNN) to perform an image-to-image denoising mapping, but the principles of N2N

will hold for a fully connected network (multi-layer perceptron) or e.g. a vision transformer as used in Section 5.3. A CNN has a limited scope of "awareness" when processing an image, X; a prediction for a pixel $\hat{s}_i$ can only be influenced by neighbouring pixels within a certain distance corresponding to a square patch centred at the pixel location of $\hat{s}_i$. The size of this patch is referred to as the receptive field of the CNN [52]. A CNN, from the perspective of single pixel prediction, $\hat{s}_i$, can thus be viewed as a function that takes an image patch around the location of s_i, call it $X_{\Omega(i)}$, in addition to a set of trainable weights, W, and then returns the prediction

$$f(X_{\Omega(i)}, W) = \hat{s}_i. \tag{3.1}$$

The objective in traditional supervised learning is to minimise the empirical risk function, i.e. the total error, which is based on a set of N training samples $\{(X^n, S^n) | n \in [1, \ldots, N]\}$, where each noisy input X^n is an image consisting of M pixels with a corresponding clean ground truth target S^n. Viewing the CNN as a single-pixel mapping function, Equation (3.1), the training dataset can similarly be regarded as $(X^n_{\Omega(i)}, s^n_i)$, where $X^n_{\Omega(i)}$ is a patch from the input image X^n corresponding to the location of the target pixel s^n_i in the ground truth image S^n. The optimal weights of the CNN are then found by minimising the empirical risk function

$$\arg\min_W \sum_{n=1}^{N} \sum_{i=1}^{M} \left(f(X^n_{\Omega(i)}, W) - s^n_i \right)^2. \tag{3.2}$$

The premise of the N2N method is the observation that the solution to Equation (3.2) remains unchanged if the target s^n_i is replaced with random numbers whose expectation value is the same as the target. As seen in Equation (2.25), the expectation of a noisy pixel in a recorded image has the expectation value of the signal, $E[x_i] = s_i$, provided the noise is zero-mean. This means that the targets s^n_i can be replaced by the pixel value at the same location in an independent realisation of the input image, x'^n_i, and the weights W resulting from training the CNN will be identical.

From supervised to self-supervised learning. Methods have been proposed in the literature to alleviate the requirement of independent realisations of the same image that N2N has. In Noise2Void (N2V) [102], a blind-spot is introduced in input images to mask the central part of input images. As described more thoroughly in the following, the pixel value corresponding to the mask is used the new target for training a model. An alternative method Noise2Self [8] proposes to partition the input image in the direction of a multi-dimensional axis, e.g. time, to provide target data from the input sample itself.

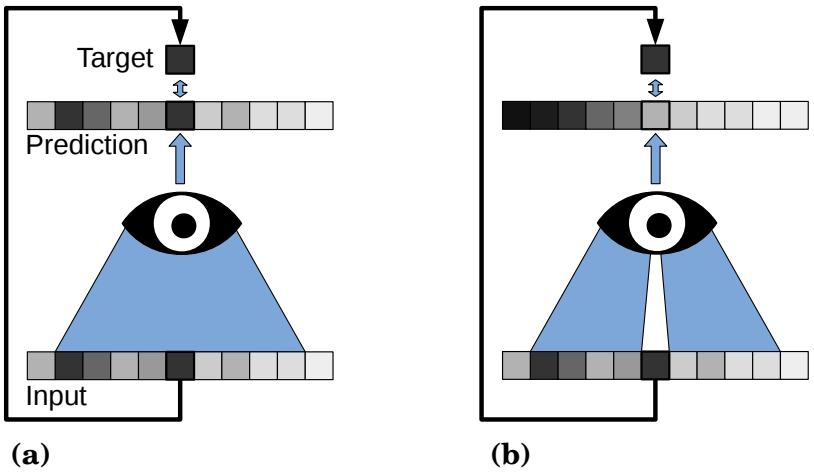


Figure 3.16: The method Noise2Void (N2V) builds upon the idea of Noise2Noise by removing the need for a secondary realisation of the noisy input image. (a) In a conventional CNN, a patch is processed to arrive at a prediction for a single pixel. (b) N2V introduces a blind-spot in the patch corresponding to the location of the target pixel. This forces the network to learn to reconstruct the signal from the neighbouring pixels.

I have found the approach of N2V to be useful, and it is the method that I have primarily studied applications of in the context of self-supervised denoising. In N2V, the concept of using noisy targets is taken further by deriving both input and the target from a single noisy training image, X^n. For a patch in this input image, $X^n_{\Omega(i)}$, the target pixel is now taken to simply be the centre pixel, x^n_i, from the image itself around which the patch is located, cf. $\hat{x}^n_i$ in Equation (3.2). Ordinarily, this would lead to the trivial solution of the identity mapping upon training the CNN, namely the mapping function $f(X^n_{\Omega(i)})$ would simply disregard the patch with weights of zero and output x^n_i directly. This is circumvented in N2V by introducing a blind-spot in the patch at the location of the target, see Figure 3.16 for an illustration. This ensures that the model learns a non-trivial mapping as a function of the other pixels in the patch. As for the N2N target x'^n_i from a different realisation of the input imaging, the target x^n_i maintains the property that the learned weights of the CNN will be identical to those obtained had a clean target s^n_i been used. This can be seen from the assumption, Equation (2.23), that the noise components are draws from the conditional distribution $Pr(n_i|s_i)$, and thus independent given the same underlying signal, s_i, with the shared expectation value $\mathbb{E}[n_i] = s_i$.

Hence, N2N and N2V work due to the same properties of noise. However, the presence of the blind-spot in N2V reduces the amount of information available in a patch during inference, and therefore lower accuracy can be expected compared to the supervised approach of N2N. The advantage is that the method works on a single image basis, and the absence of a single pixel is a manageable issue if the pixel size, in terms of spatial resolution, is significantly smaller than features of interest.

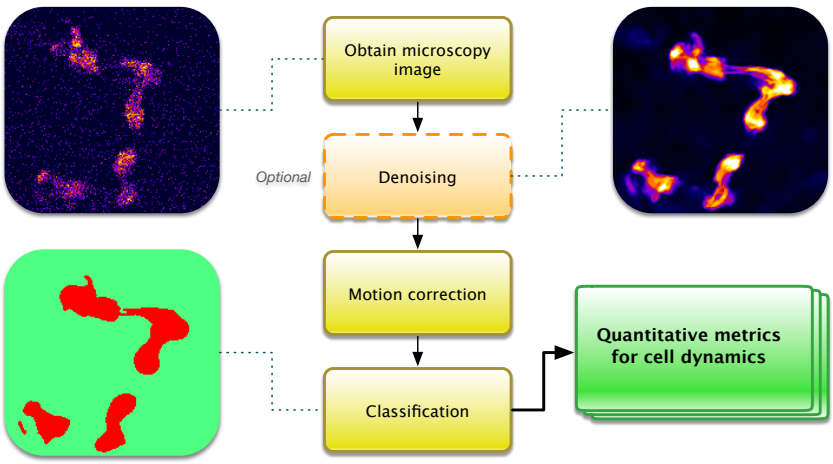


Figure 3.17: The steps of a common pipeline for calcium imaging analysis can be subdivided into three areas before quantitative analysis is performed. Denoising is an optional step that can help to improve signal-to-noise and enhance features. Motion correction may be necessary in cases of drift or movement. Classification can select regions of interest for which quantitative analysis is performed.

3.4.2 Wide-field fluorescence imaging

The first application of a self-supervised denoising method to be considered is for fluorescence microscopy, which was also attempted in Section 3.3 with a conventional supervised approach. A first example is related to calcium imaging, and follows work done in collaboration with former group member Dr Miranda Robbins, some of which is published in co-authored review paper [172]. After considering the application to calcium imaging, a second example of applying the denoising method is presented using imaging data of neurons in Xenopus laevis.

Application to calcium imaging. The ability to image calcium ion dynamics in cells has long been of interest, particularly in the neurosciences, where it can be used as a marker for neuronal excitability. Calcium imaging is an inherently noisy method as imaging of the samples often suffer from low SNR, drift and cell movement, particularly for living organisms. This poses a problem for quantitative analysis with a typical processing pipeline, see Figure 3.17, which may include image denoising, motion correction, classification for cell identification, and quantification of calcium signals [172].

As the later stages of this pipeline depend on image data with an adequately high SNR, the denoising step, although indicated as optional in the pipeline, can be crucial. If denoising is applied effectively, the accuracy of the subsequent quantitative processing tasks are likely to be improved. In the following subsections, we shall see examples of improvement of various

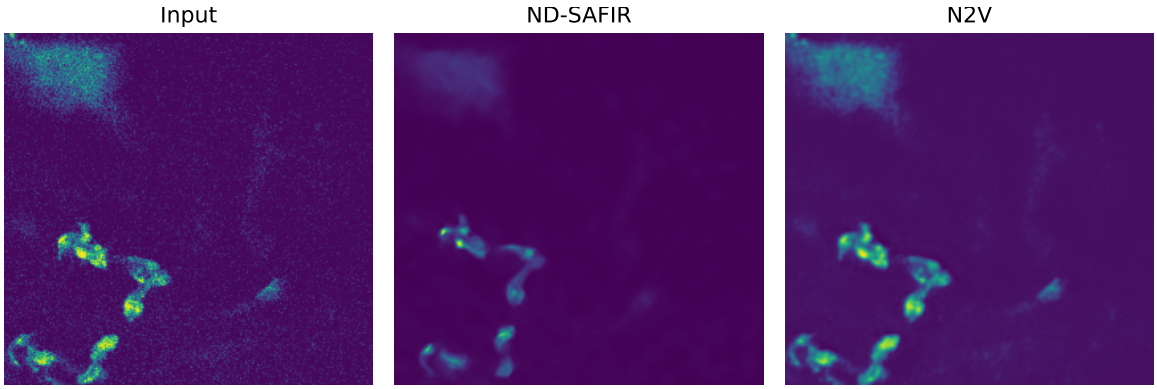


Figure 3.18: Wide-field microscopy image sample of neurons with GCaMP as a marker for calcium ion. The image is denoised with a traditional local denoising method, ND-SAFIR, and a deep learning-based denoiser trained with the Noise2Void (N2V) training strategy.

quantitative analysis steps. However, for this first example we will only consider the qualitative effect of using a denoiser built with the Noise2Void (N2V) training strategy.

Images were acquired with a wide-field microscope and samples were labelled with GCaMP, which is a genetically encoded calcium indicator. A sequence of about 50 frames was acquired over time per field of view. A single time stack is used to train a U-Net CNN model over 20 epochs based on self-supervised learning according to N2V.

An example of an acquired raw image is shown on the left of Figure 3.18. The image is denoised with both ND-SAFIR [15], representing a traditional denoising method for comparison, and a N2V model that has been built based on this dataset. Qualitatively, the output of the ND-SAFIR method appears a bit washed out with some loss of structural information. The output of the N2V denoiser is by comparison more consistent with the raw image, while parts of the structure seem more clearly resolved. The high-frequency noise in the background of the raw image is effectively removed with both denoising methods.

Application to kymograph analysis. Another example of applying the self-supervised denoiser to fluorescence microscopy is the imaging of neurons in frogs, more concretely retinal ganglion cells of Xenopus laevis. A collection of images was acquired by group member Lucia Wunderlich using a wide-field microscope. The axonal segments of the neurons are of special interest, and they can be analysed by representing the image data in the form of kymographs, which are graphical representation of spatial position over time. Kymographs require video data and the specification of a motion axis, which is done in post-processing following the acquisition of wide-field image sequences. However, for a kymograph to provide useful quantitative data, it must first be segmented. A software solution for this used in the

group is KymoButler [84], which is able to distinguish multiple tracks from each other and output the desired quantities, such as measured velocity of a particle. The SNR of the acquired wide-field image sequences tends to be low due to limited photon budgets. This propagates into the kymographs and can cause issues in the segmentation stage when using e.g. KymoButler. A denoising stage used in preprocessing with respect to the kymograph conversion could be used to reduce the effects of the high noise level and possibly lead to more accurate quantitative estimates.

This has been attempted with a batch of wide-field images using both ND-SAFIR and a CNN network trained according to the N2V principle. An example is shown on Figure 3.19. ND-SAFIR is seen to remove a significant proportion of the high-frequency noise as is evident by the more coarse-grained appearance of the remaining noise. Its inability to remove the background more completely is likely due to the lack of tweaking its denoising parameters; it is used here as a blind denoising method with its default parameters. While denoising with ND-SAFIR ideally should reduce the noise further, a reduction of the high-frequency information of the signal is already showing causing a loss of spatial resolution, and it is possible that further denoising by parameter tweaking might result in lower yet spatial resolution. However, the overall denoised image is still arguably an improvement over the raw image, and it is found to improve the results of KymoButler as shown on Figure 3.20.

The N2V-based model on the other hand is able to remove the background much more thoroughly. Apart from the vignetting, the background appears uniform and the structural information of the sample is more clearly seen. By inspecting some smaller, fluorescent blobs in close proximity to each other, they are more distinguishable in the N2V output versus the raw image. In some regions, these blobs appear to overlap in the raw input but are separated in the N2V output. Hence, the fluorescent signal overall seems to have more discernible high-frequency information. In the kymograph on Figure 3.20, the N2V output is also seen to be more clean, which causes KymoButler to detect multiple additional tracks compared to the case of the kymograph based on the raw data.

An identified track on Figure 3.20 has a differently assigned colour if it is recognised as separate to the other tracks. The additional tracks shown on the kymographs where ND-SAFIR and N2V have been applied in preprocessing indicate that the overall processing pipeline has become more robust to noise in the input. However, with the increased sensitivity to information at low SNR, there is also an increased risk of spurious tracks. Some tracks in both the kymographs that have been preprocessed with ND-SAFIR and N2V are likely to be spurious, but those associated with N2V appears more consistent with the input data, thus pointing to the utility of N2V in this application.

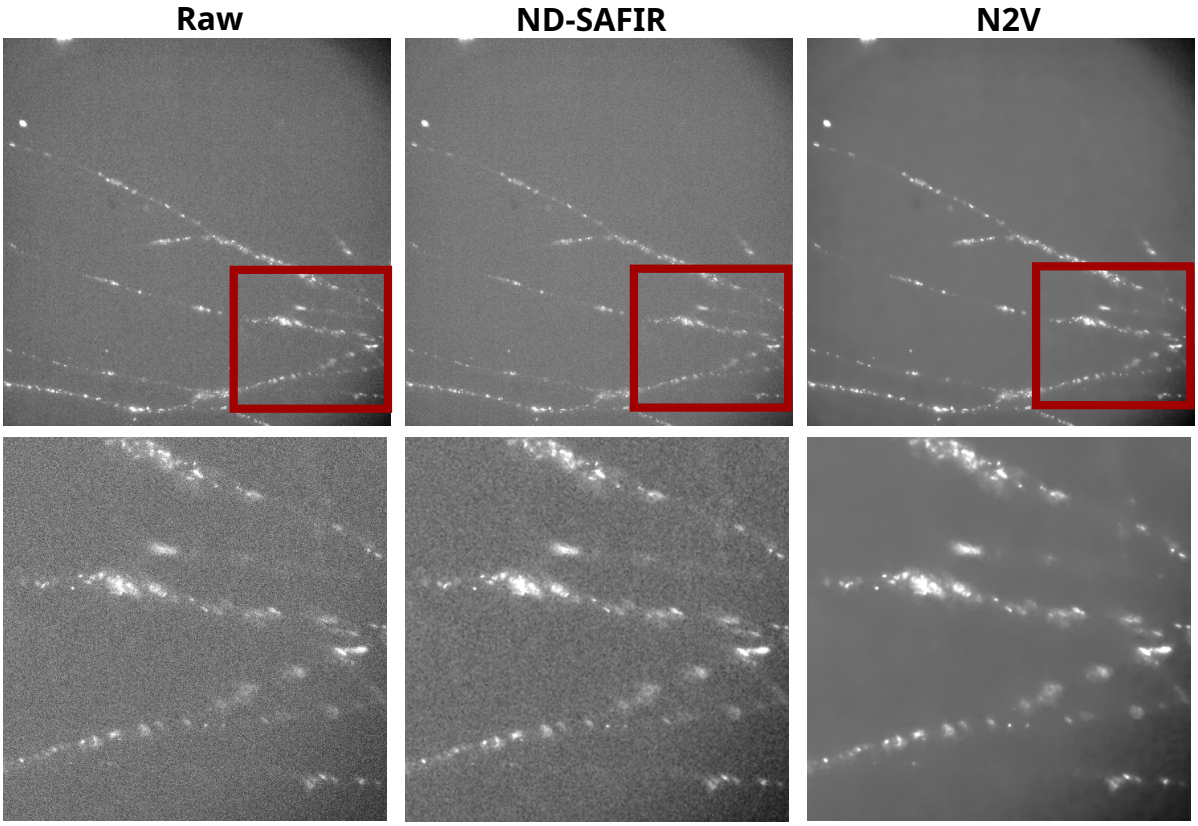


Figure 3.19: Denoising of wide-field fluorescence microscopy data with two different denoising methods: ND-SAFIR [15] and a model trained with the Noise2Void approach [102].

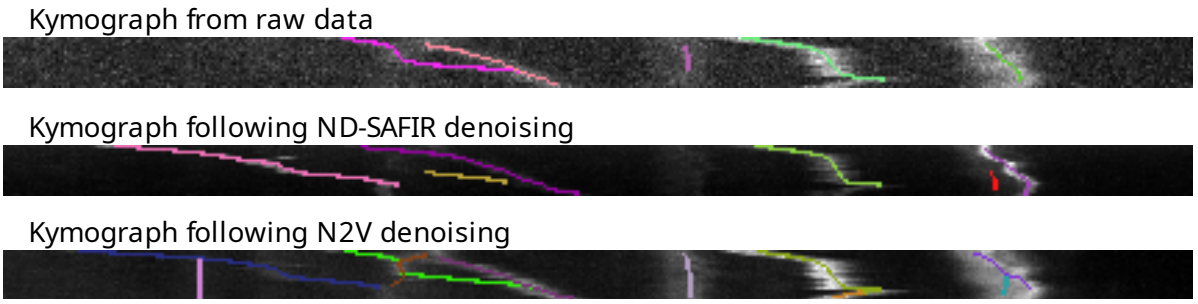


Figure 3.20: Three kymographs overlaid with tracks identified by the KymoButler software [84]. The kymographs are derived from the same source image, and for the two bottom kymographs, preprocessing is applied to the source image as shown in Figure 3.19 with ND-SAFIR and N2V denoising, respectively.

3.4.3 Cryogenic electron microscopy

We will now turn our attention to an application outside of fluorescence microscopy, namely in the related field of cryogenic electron microscopy. The results shown in this section stem from

a collaboration with Dr Helen Foster who has performed the data acquisition and tomographic reconstruction. I have attempted denoising the images prior to reconstruction to investigate whether the reconstruction quality could be improved. The experimental conditions and details of the reconstruction pipeline are described comprehensively in Dr Foster's PhD thesis [50].

In electron tomography, a collection of images are acquired by tilting the specimen through a range as close to 180° as possible. This provides a stack of images of the sample that is to be further processed. It is at this stage that the stack is denoised with a model trained with the self-supervised N2V approach. After the denoising, multiple other steps of processing normally follow, such as gain correction, motion correction and alignment of frames. The projected image intensities can then be reconstructed into a tomogram. The tomogram can be further post-processed with a deconvolution filter to improve the contrast.

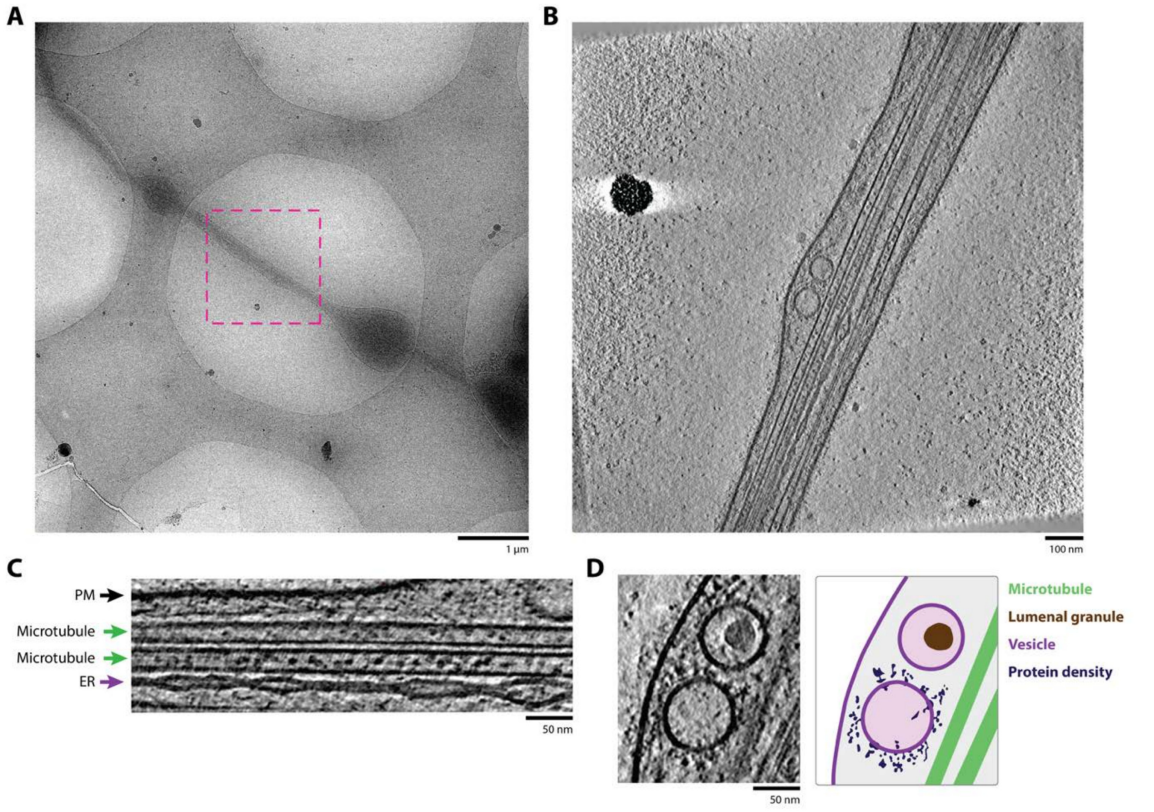


Figure 3.21: Tomogram of microtubules and vesicles in axons. The large field-of-view in (A) shows the larger system with the box of dashed lines indicated the region used for tomographic reconstruction as shown with the z-slice in (B). The imaged filaments and membrane structures are indicated in (C); PM stands for plasma membrane and ER is the endoplasmic reticulum. In (D), a cropped region of electron density in and around vesicles is shown. Figure credit [50].

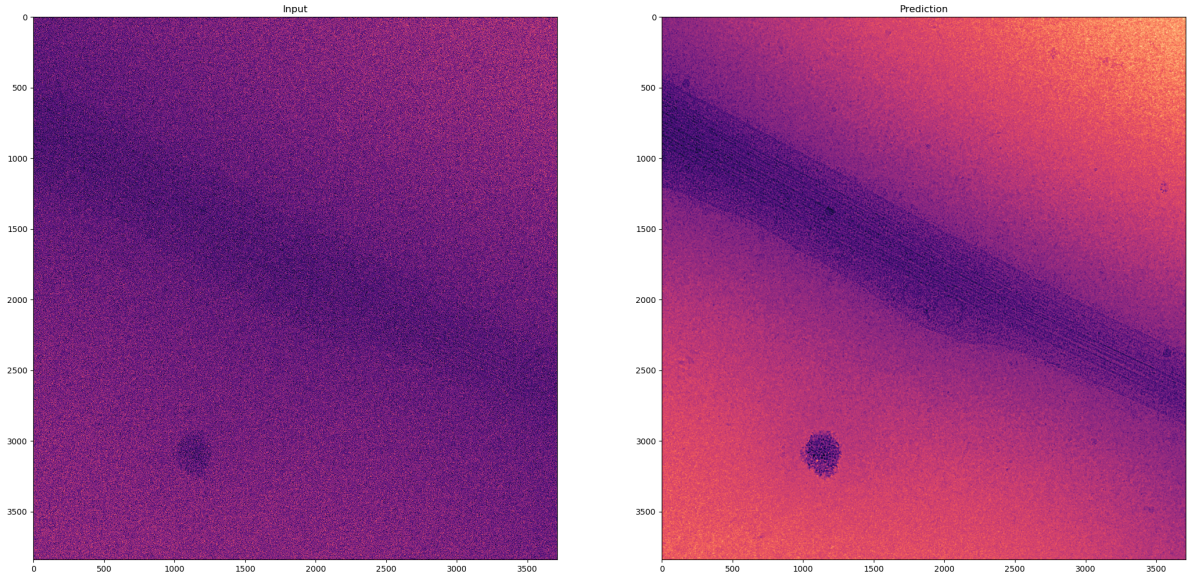


Figure 3.22: A model trained according to the Noise2Void principle applied to cryogenic electron microscopy data of filaments in mouse neurons. The left side shows the input image that is an arrow of 10 frames at the same tilt angle, and the right side is the denoised version of that image.

In the experiment described in this subsection, the tilt angle varied from -60° to $+60^\circ$ with an increment of 2°. At each tilt angle, a total of 10 images were acquired. In total, this provided a stack of 600 images to be denoised. The imaged sample consisted of dorsal root ganglion neurons from a mouse. The images of Figure 3.21 show the sample at a tilt angle of 0 in a large field-of-view and a smaller cropped region of a z-slice from the resulting reconstructed tomogram.

The provided dataset also made the approach of N2N, cf. Section 3.4.1, possible as the 10 acquisitions of the sample at every tilt angle mean that realisation of the sample with independent noise but identical signal are available. Thus, for training a CNN, the training target in a training pair could as well be taken to be one of 9 frames. Alternatively, the 10 frames could be partitioned into two stacks of 5 frames and then averaged separately into a final set of 2 frames with higher SNR. However, in the interest of simplicity, and keeping results more comparable to the other applications studied in this section, the easier option of averaging across the entire 10 frames and then using the self-supervised N2V approach on a single image basis was chosen. The arithmetic mean was used as the average.

The average image based on the 10 frames at a tilt angle of 0 is shown on the left side of Figure 3.22. Clearly, high-frequency information originating from the sample is hard to discern given the high noise level even though 10 frames are averaged. Despite the low quality of the images at the various tilt levels, it is possible to reconstruct the tomogram and apply

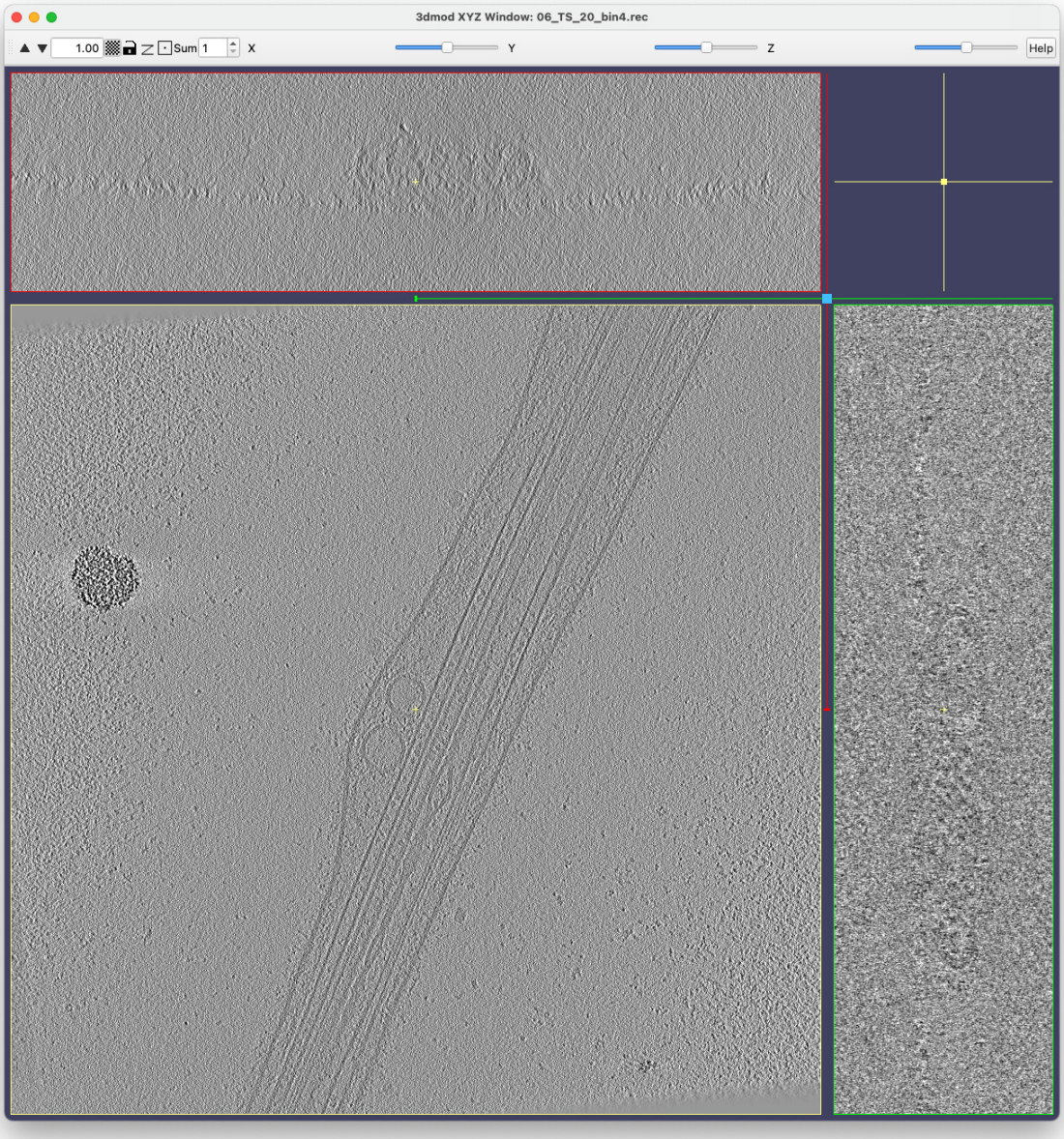


Figure 3.23: Reconstruction a tomogram from cryogenic electron microscopy data based on the raw data provides poor contrast.

deconvolution, which recovers some high-frequency information of the sample as we will see in the following.

When training a CNN model based on the self-supervised N2V approach, a training pair is extracted from a single image. Every non-overlapping patch of the input image of a size similar to the receptive field of the CNN provides a useful training sample. Given the size of the input

images, 3,600x3,600 pixels, a model can be fully trained on training pairs extracted from just a single image. However, to increase the size of the training dataset, the training pairs extracted from images at different tilt angles are used in combination. The single model is then trained and is able to denoise images at all the tilt angles. Upon applying the trained model to the image averaged over the 10 frames at tilt angle 0, the network output is as seen on the right side of Figure 3.22. Without considering the benefits of the tomographic reconstruction yet, the SNR in the output image is evidently significantly improved. High-frequency information of the filaments and vesicles has been recovered, while the background noise has been considerably reduced.

To assess the quality of the denoised image compared to the raw image, the NIQE scores, see Section 2.4.7, are calculated. The raw input image is found to have a NIQE score of 24.9, and the image denoised with N2V has a score of 10.0. Considering the examples of Figure 2.6, for which ground truth images were available in the two cases of synthetic degradation, the difference in score here is significant and would indicate an improvement in the image at least on par with the reconstruction of a raw SIM to a super-resolved image. It is noted, however, that the image dimensions are larger in this case, and it is unclear how NIQE scales with image resolution. The used NIQE model is also built on a collection of macrographs, i.e. images taken of objects at the scale visible to the human eye, which differ significantly in appearance to the EM images. Yet, the difference in NIQE score of the images on Figure 3.22 compared with the SIM images before and after reconstruction on Figure 2.6 is an indication that the N2V-based model denoising model could make an important difference in the tomographic reconstruction pipeline.

For evaluating the tomographic reconstruction quality when using the raw images versus the denoised images in the reconstruction pipeline, a slice in the centre of the 3-dimensional tomogram is used for comparison. I used the software *3dmod* to inspect the structure of the tomogram along each axis. The slice from the tomogram based on raw data is shown in Figure 3.23, whereas the slice from the tomogram based on denoised data is shown on Figure 3.24.

A first observation is that despite the very low SNR seen in an individual image at a specific tilt angle on the left side of Figure 3.22, the tomogram of the same non-denoised image data contains a cleaner signal with more apparent high-frequency information. This may be due to the deconvolution operation in the reconstruction pipeline. However, the contrast of the microtubules and vesicles is still poor, and the height and width maps on the top and bottom of Figure 3.23 hardly have any discernible signal. The z-slice of the tomogram based on the denoised data appears with more clear contrast and more well-defined boundaries of the filaments. The irregular shape of the ER tubules is easy to see in the centre of the image close

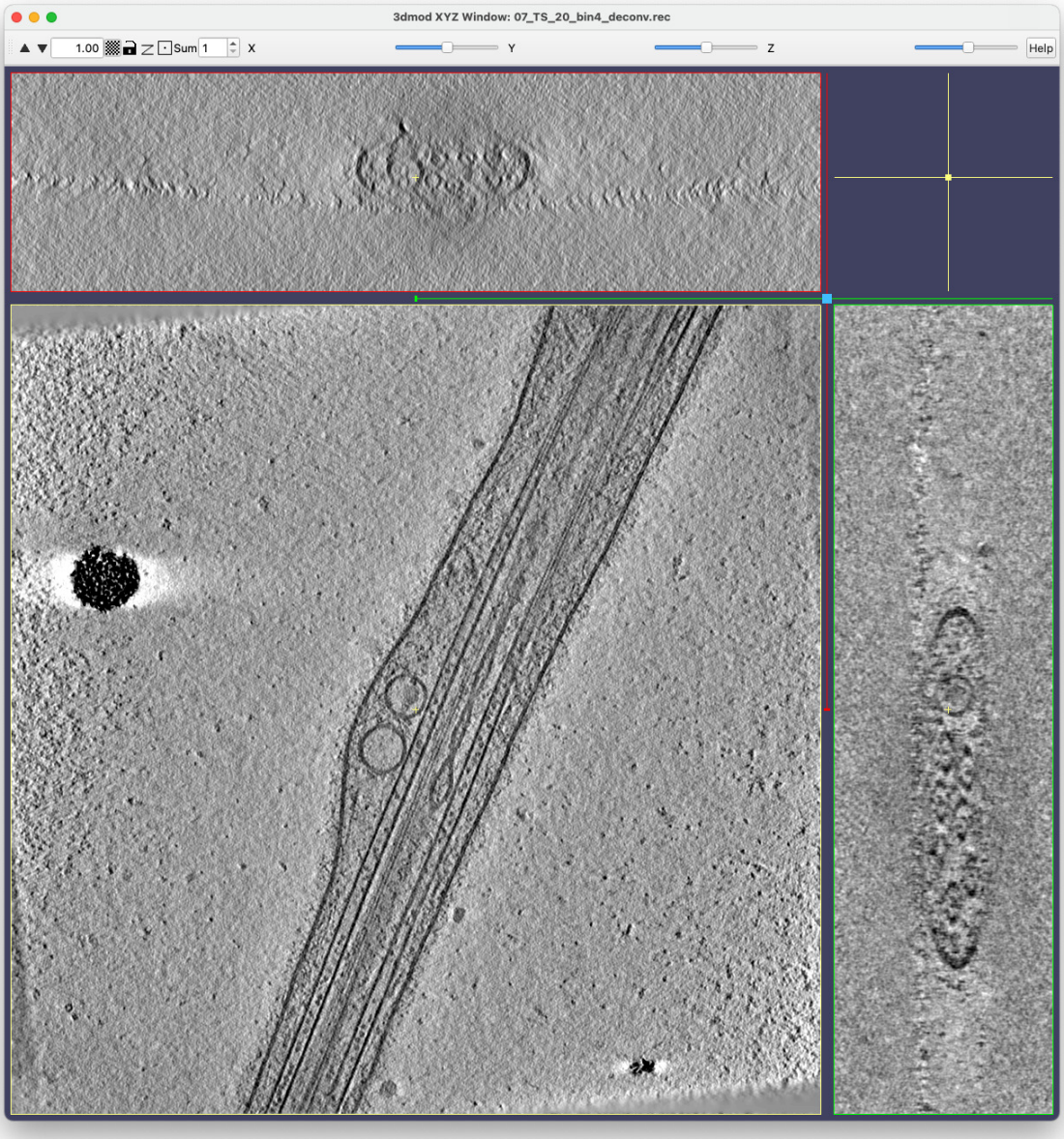


Figure 3.24: Reconstruction a tomogram from cryogenic electron microscopy data with the self-supervised denoising model applied in preprocessing.

to the vesicles, but the structures seem to fade towards the bottom part of the image, which, however, still appears present in the tomogram based on the raw data. This might indicate that the denoising method causes the loss of some structural information. In terms of the height and width maps, the effect of the denoising is clearly beneficial as the 3-dimensional size of the

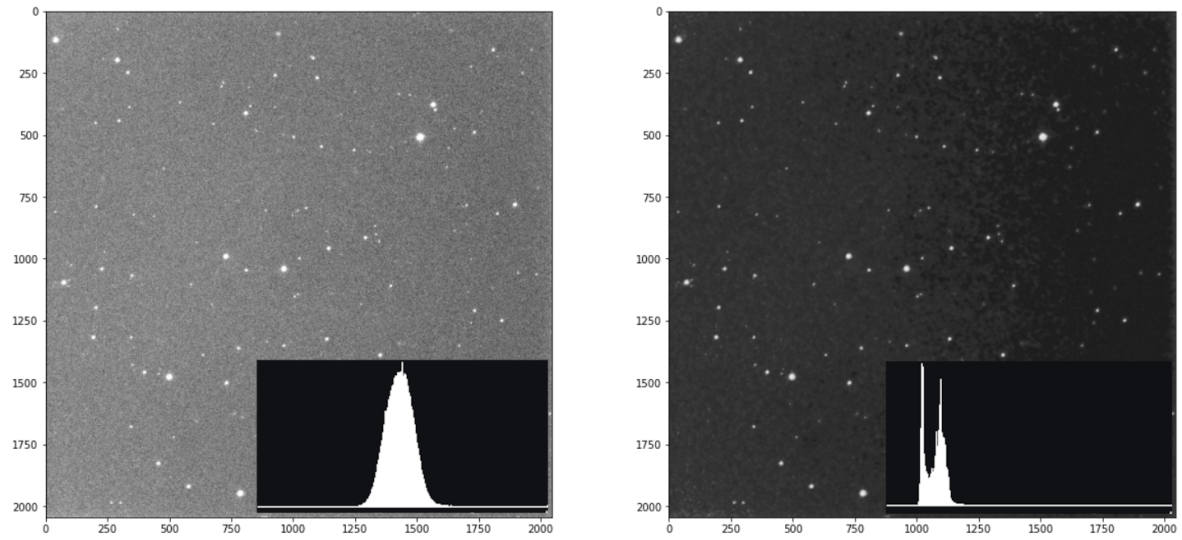


Figure 3.25: Noise2Void method applied to astronomy data for denoising of images with stacked frames.

filaments can now be estimated in contrast to the z-slice from the tomogram based on the raw data.

3.4.4 Image processing for astronomy

Astronomy is another area of scientific imaging where images with low signal-to-noise ratios are pervasive and pose a challenge to quantitative analysis. Hence, denoising methods have the potential to increase robustness of quantitative analysis pipelines similarly to what was found in the preceding subsections. The noise sources in astronomy primarily include shot noise, dark noise, and read-out noise [201]. Dark noise arises from thermally excited electrons of the detector, often called dark current, and is usually negated by cooling the detector. Shot noise is Poisson distributed as a function of the signal, whereas dark noise is Poisson distributed as a function of dark current. The read-out noise is additive and Gaussian distributed. Both shot noise and read-out noise are zero-mean noise sources. However, although the effect of dark noise can be corrected for and be made zero-mean [65], it is by default not zero-mean thereby posing a potential problem for a denoiser trained with the N2N and N2V strategy. In this subsection, I will review the applicability of the machine learning denoiser trained with the self-supervised N2V approach.

In the study of exoplanets, an important measurement is the detection of planetary transits. A transit is the occurrence of an exoplanet passing between a star, often one that it is revolving

around, and an observer. The frequency and duration of the transits in relation to characteristics of the star can be used to gain insight about the planet. An important technique in the detection of transits is photometry [58], which is the measurement of the flux or light intensity of stars. A robust pipeline for photometry has multiple preprocessing steps to counter e.g. dark noise and fixed-pattern noise with dark frame subtraction and flat-field correction [147]. After this initial preprocessing stage, background removal is performed, and stars are then localised in the images for the final extraction of temporal flux statistics [18]. The localisation can be performed manually according to a selection of stars of interest, but we will consider a case of high-throughput analysis, where it is assumed that automatic localisation is required. Automated localisation of stars can be achieved with connected component analysis following accurate background removal or segmentation e.g. by thresholding as described in Section 2.4.6. Alternatively, blob detection similar to the nuclei counting example of Figure 3.14 could be performed directly. In the following, the efficacy of a trained denoiser will be tested by applying it as an additional preprocessing step following flat-field correction but prior to localisation. The localisation was attempted with both thresholding-based segmentation and blob detection based on a difference of Gaussians approach.

The data used for this test was provided by PhD student Peter P. Pedersen at Cambridge Exoplanet Research Centre of Institute of Astronomy, University of Cambridge. Image data was acquired at the group's Chilean observatory. The telescope that was used has an active cooling system, which effectively renders dark noise negligible. The image data was corrected as described above before being provided and used for this test of the self-supervised denoiser. An example of an input image used for training the denoiser is shown on the left side of Figure 3.25. The intensity spectrum has been scaled to fit the entire 8-bit range with clipping of the lower and upper 2 % of the intensity values. Despite clipping, the large variability of the pixel values cause the scaled image to have a background baseline that is approximately in the centre of the range. This non-zero noise floor is also indicated by the histogram in the lower right of the image. Training was applied to extracted patches from a collection of 275 images with the target in a training pair being the centre of each patch similar to the approach in the preceeding subsections.

The denoised image of the previous example using the trained denoiser model is shown on the right side of Figure 3.25. The noise floor is seen to be significantly reduced with a much higher contrast between background and the highlights. Interestingly, a shift in the background level emerges towards the right side of the image, which is not perceptible in the input image. This is also evident in the histogram in the lower right corner. The noise that is still visible in the background has a significantly lower spatial frequency, reminiscent of the application of a smoothing kernel, yet the spatial resolution of the signal from the stars appear to be preserved.

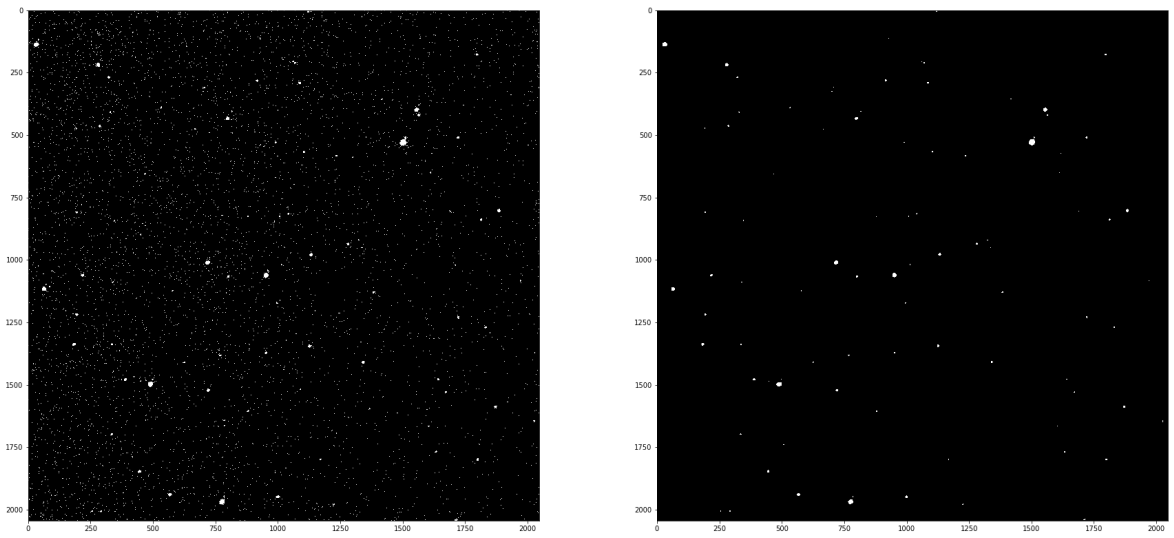


Figure 3.26: Segmentation of image by simply thresholding with and without denoising applied in preprocessing.

Despite the background shift potentially being an artefact, the overall difference in image quality between the input and denoised image is visibly high. The NIQE scores for the input and denoised images are 31.07 and 18.59, respectively, which indicates a similar improvement to the denoising of the EM images in the previous subsection.

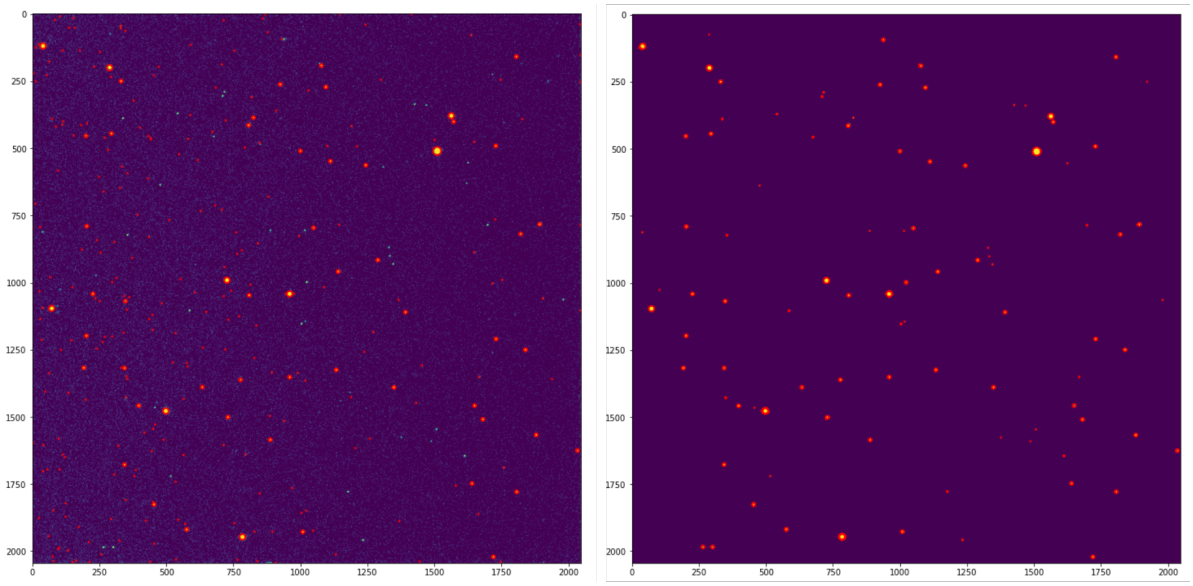


Figure 3.27: Blob detection of image using a difference of Gaussians approach with and without denoising applied in preprocessing.

Segmentation by thresholding is sensitive to distributions of noise that ranges across the entire intensity spectrum. Thus, the segmentation of the noisy input images may be expected to pick up a lot of the noise as the foreground, which ideally should only contain the stars. The input image on Figure 3.25 is segmented with a threshold value tuned to capture the majority of the stars for the foreground. The resulting segmentation map is shown on the left side of Figure 3.26, and the corresponding segmentation map produced by segmenting the denoised image in the same way is shown on the right side. The segmentation of the noisy input image is seen to produce a high number of small foreground patches that derive from the background noise. By performing connected component analysis, a further threshold could be defined in terms of the pixel area of each patch, but without further processing the segmentation map is not likely to be of any use. In contrast, the segmentation map based on the denoised image appears much cleaner. Although a ground truth is not available, the foreground patches seem to consistently correspond to those objects that consistently have a high intensity across the stack of 275 images indicating that the foreground exclusively consists of stars. It is noted, however, that the fine-tuning of the threshold value is done manually and therefore is slightly arbitrary. Arguably, from comparing the denoised image on Figure 3.25 and its segmentation map on Figure 3.26, the threshold value is perhaps too conservative in that it likely ignores some of the signal that may correspond to stars. However, for both the noisy and denoised image, the threshold values have been tuned to capture the most obvious objects, and the test result indicate that the segmentation is considerably more consistent when performing denoising in advance.

A similar result is found when performing blob detection with the difference of Gaussians (DoG) approach as described in Section 2.4.6. When filtering the second derivatives obtained with the DoG approach, a threshold is chosen for the standard deviations of the Gaussian kernels, which correspond to filtering the size of the detected blobs. This enables the method to be less sensitive to the noisy patches with high intensity that may be filtered as foreground in the previous segmentation attempt of the images that were not denoised first. In Figure 3.27, the detected blobs, using the implementation mentioned in Section 2.4.6, for the noisy and denoised image are overlaid with the respective images. The detected blobs for the noisy image appear to include most of the obvious objects that resemble stars. Although fewer than for the segmentation map, there remains a high number of false negatives. The blobs detected for the denoised image agree well with the corresponding segmentation map, and overall appears consistent with what could be expected of an ideal result. While the blob detection of the denoised image is more robust to the noise, a high sensitivity to the signal seems to be preserved as is indicated by the presence of several blobs exclusively on the right side of Figure 3.27.

In summary, the self-supervised denoiser trained after the N2V principle is seen to improve the accuracy of localising the stars whether a segmentation or blob detection approach is taken. Motivated by this test, it seems probable that denoising could be beneficial to the analysis pipeline described in the beginning of this subsection by enabling automation and higher robustness to noise. Although the noise sources in astronomy differ from those governing fluorescence microscopy imaging, the denoiser was found to offer improvements similar to the example of cryo-EM imaging in Section 3.4.3. In particular, dark noise is expected to play a role for astronomy images, which could have caused issues with the N2V training strategy given that the noise source is not zero-mean. While images with dark noise can be made zero-mean as noted in [108] using transformations and corrections such as those proposed in e.g. [65], it was not found to be necessary for the viability of the denoiser on the image data considered here. The likely cause for this is that the dark noise is negligible in the image data due to the cooling system of the telescope.

Chapter 4

Segmentation of image data of the endoplasmic reticulum

In this chapter, the previously introduced super-resolution neural network architectures are modified to perform image segmentation and applied to images of the endoplasmic reticulum (ER). The ER is known to be a highly dynamic environment [7] with processes such as the peristaltic flow of luminal proteins [78, 148] and fluctuations of its shape [212].

With the advent of super-resolution microscopy, the structure of the ER is well-known. The image of the ER in Figure 4.1 shows the major structural domains of the ER, including the nuclear envelope, sheets and peripheral tubules [212]. Mutations in ER-shaping proteins can lead to morphological defects, and many of these proteins have been linked to the pathology of human diseases [212]. One example is reticulon that structurally shapes the ER tubules in the peripheral domain and has been found to be involved with Alzheimer's disease [220].

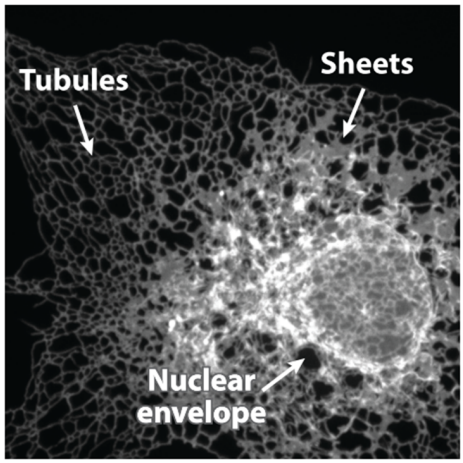


Figure 4.1: Layout of the domains in the endoplasmic reticulum. Image credit [212].

In this chapter the peripheral tubules will be considered with the aim of segmentation. Being able to accurately distinguish between ER tubules and background, enables detailed analysis of the shape of the ER and its dynamics.

4.1 Building a neural network segmentation model

Choosing the first part of our segmentation model to have an architecture built for restoration ensures that it is capable of handling images with low signal-to-noise ratio as it can learn to perform denoising in these early layers of its network. A neural network model intended for image restoration will by default perform regression in order to output pixel value predictions in the same colour space as the input image. This is achieved during model training by minimising an appropriate loss function, typically the mean squared error defined over the dataset as

$$L_{\mathrm{MSE}}(\Theta;D) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{\mathrm{WH}}\sum_{x=1}^{W}\sum_{y=1}^{H}\left[F\left(\Theta;I_i^L\right)_{x,y} - I^H_{i;\,x,y}\right]^2\right), \tag{4.1}$$

where Θ represents the trainable parameters of the network referred to as $F(\cdot)$, while D is the training dataset of size N written as $\left\{I_i^L, I_i^H\right\}_{i=1}^N$ consisting of low-quality input images and high-quality target images with pixel size $H \times W$.

Rather than having the model perform restoration via regression followed by thresholding by intensity values to produce binary segmentation maps, the model is directly optimised to output segmentation maps by modifying it to perform classification. A common choice of loss function for classification models is the cross-entropy loss given by

$$L_{\mathrm{CE}}(\Theta;D) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{\mathrm{WH}}\sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{k=1}^{K} -f^H_{i;\,x,y}(k)\log\left[\frac{\exp\left(F\left(\Theta;I_i^L\right)_{x,y;k}\right)}{\sum_{j=1}^{K}\exp\left(F\left(\Theta;I_i^L\right)_{x,y;j}\right)}\right]\right), \tag{4.2}$$

where k and j are iterators over a total of K unique classes, and $f^H_{i;\,x,y}(k)$ is a function equal to 1 if the target class for the pixel at (x,y) of the i^{th} image is k, and otherwise it is equal to 0. With the model set up for classification, the network $F(\cdot)$ now returns scores for each class for a given input image, from which class probabilities are estimated by applying the softmax function, i.e. the normalised exponential inside the log function. The use of the cross-entropy loss over the mean squared error loss during training greatly improves the performance of models with softmax outputs, since the mean squared error tend to lead to saturation and slow learning (I. Goodfellow, DL book, 2016), which is why the approach of directly optimising the

model to output segmentation maps is preferred. Note that $K = 2$ for the purposes of the binary segmentation used in this work, in which case the innermost summation in $L_{\mathrm{CE}}(\Theta; D)$ over the variable k reduces to the addition of two simple terms known as the binary cross-entropy loss.

4.2 Simulation-supervised segmentation model

The first approach that will be considered for the segmentation of ER images relies on a simulated image formation model for training data generation. The source data used in the image formation model is a collection of experimental images, in Section 4.2.2 through Section 4.2.3, except for in Section 4.2.4 where a fully synthetic data generation approach for ER images is described.

4.2.1 Training data

For images with high signal-to-noise ratio it is easy to perform a binary segmentation of the endoplasmic reticulum simply by pixel intensity thresholding. The difficulty arises when the image data is so degraded that the tubular structure of the ER is no longer intact, such that a thresholding approach would yield disconnected segmented networks. Ideally, the segmentation model should learn how to reconstruct the network structure of the ER, thus filling out blanks between parts that are likely to be connected.

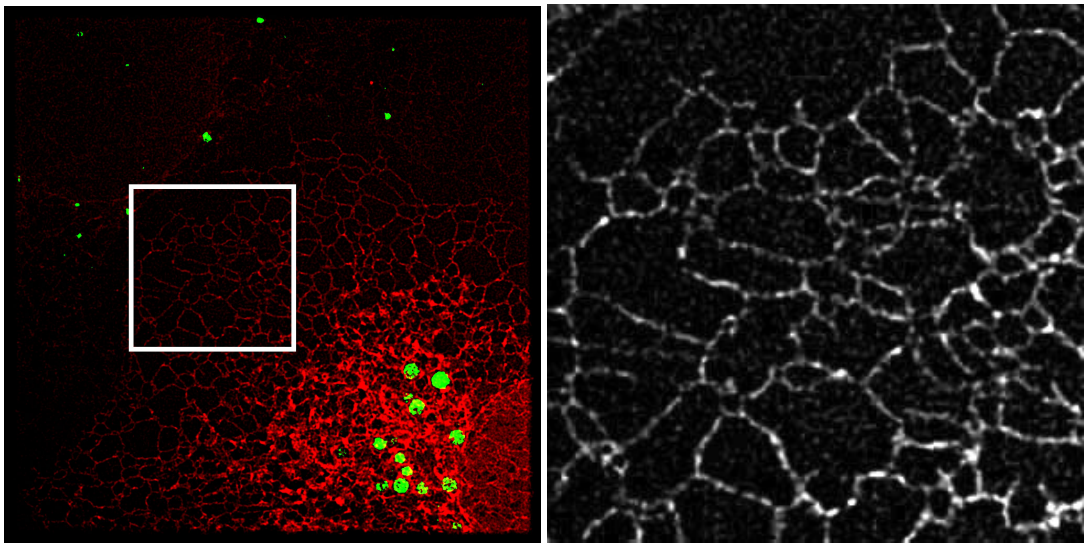


Figure 4.2: Example of a low-quality experimental image that is not useful for training, but can be used for testing the trained model.

To train a network for this problem, a desirable dataset would have matching pairs of images of low-quality and high-quality that correspond in space and time, such that a ground truth image with simple thresholding could form the target data, whereas the low-quality data would represent the actual data acquired in experiments. The important part is the high-quality images as the low-quality ones can be generated similarly to the approach described in Section 3.3.4,

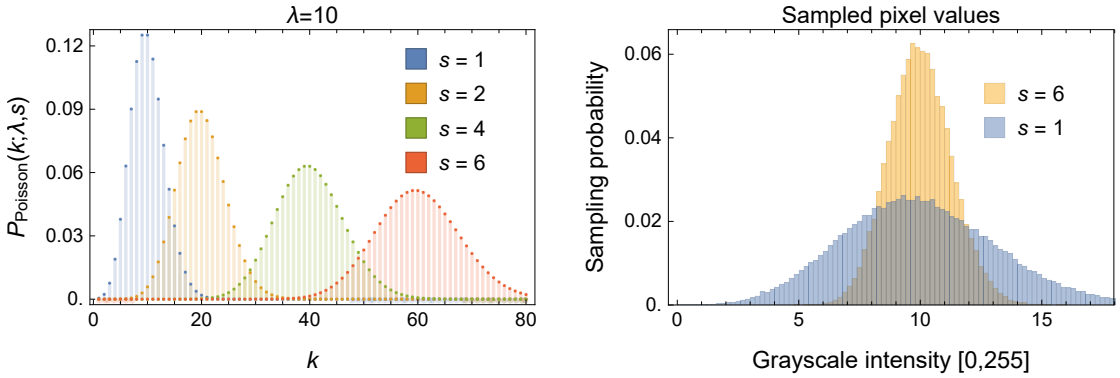


Figure 4.3: Comparisons of different values of the parameter of s in $P_{\text{Poisson}}(k;\lambda,s)$, Equation (4.4).

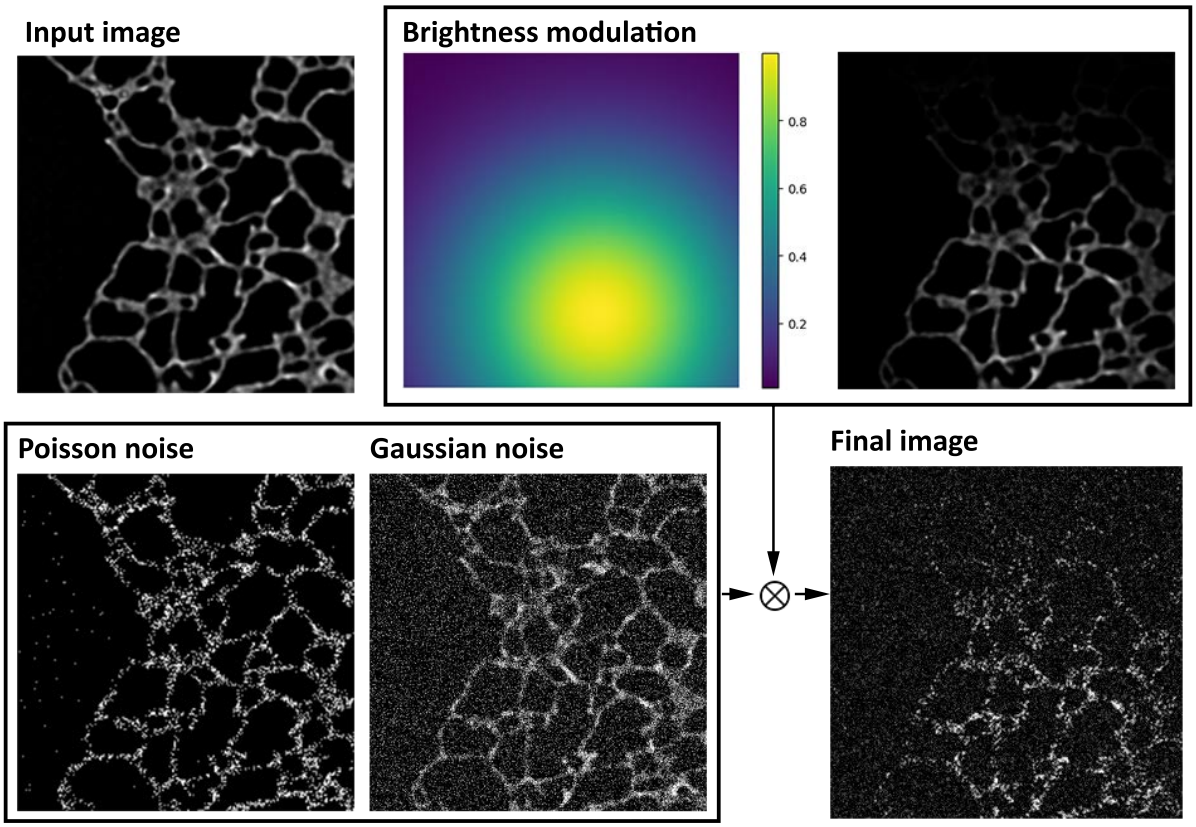


Figure 4.4: Synthetic image generation process simulating the degradation and uneven illumination in an optical imaging system.

where synthetic degradation is introduced. One limitation with this approach is that it may not be possible to even acquire images of particularly high quality, since biological samples of interest in live-cell imaging tend to be dynamic and fragile. A remedy could be to fix the samples in place, thus making them static and devoid of dynamics, or alternatively to use simulated samples as ground truths. In the following, we will consider the case where a small set of experimentally acquired images have adequate quality for providing ground truths derived from intensity threshold.

The ER images that have been available for training, provided by group member Meng Lu, are of live-cell endoplasmic reticulum samples. The quality varies significantly from image to image depending on the used capture settings and the amount of photobleaching. The provided dataset also does not have multiple realisations with the same field-of-view that could have provided the paired low-quality and high-quality training data. This therefore necessitates the aforementioned approach using synthetic degradation to facilitate the supervised learning. Since some provided images are of low-quality to begin with, those images can be used for testing purposes, while the higher quality images can be used for training. To generate a supervised dataset, we can use the high-quality images as ground truth and the synthetically degraded images as inputs.

Although the degradation is synthetic the aim will still be for the trained network to be able to segment the original low-quality damage, which involves learning to reconstruct the network structure of the ER under high noise levels. For this to work well on the real test set, the synthetic degradation should be as realistic as possible by including the noise sources that are known to affect experimentally acquired images. Considering an example of a low-quality experimental image shown on Figure 4.2, it is clear that noise is very prevalent, so much that the tubules of the ER appear to become disjointed at some points. In addition to this the fluorescence intensity is also not consistent over the image, presumably due to photobleaching, causing the signal-to-noise ratio to vary significantly. This behaviour is present in several of the experimental images, and it will be approximated as a radially decreasing brightness. This decreasing brightness is assumed to originate from a central point and follow a two-dimensional Gaussian distribution. Both Gaussian and Poisson noise are expected to be present in the experimental data [85], and thus both noise sources are included in the simulated noise model. The characteristic of each distribution is hard to identify from the images alone, and thus an assumption has to be made about the parameters of the distributions. To make all the assumptions slightly less arbitrary, the degradation parameters are not set to any single set of values but rather take on random values from broad ranges for every generated synthetic image. The brightness modulated image is given by the function

$$I'[x,y] = I[x,y] \exp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)\right), \tag{4.3}$$

where $I(x,y)$ is the original image as a function of pixel row and column (x,y) and the randomly generated origin (x_0,y_0) that is somewhere within the bounds of the image, and σ_x and σ_y are the standard deviations of the kernel.

The Gaussian and Poisson noise is generated by sampling the following probability distributions with respective random variables x and k

$$P_{\text{Gaussian}}(x;\sigma,\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad P_{\text{Poisson}}(k;\lambda,s) = \frac{(\lambda s)^k \exp(-\lambda s)}{k!}, \tag{4.4}$$

where μ is the mean of the Gaussian noise, assumed to be zero, σ is the standard deviation for the Gaussian noise, while λ is the expectation value of the Poisson distribution set to the pixel values of the non-degraded image and s is a parameter controlling the amount of Poisson noise. Given samples from the probability distributions, say x_{Gaussian} and x_{Poisson}, respectively, the resulting noisy images are generated by

$$I_{\text{Gaussian}}[x,y] = I[x,y] + x_{\text{Gaussian}}(\sigma), \quad I_{\text{Poisson}}[x,y] = \frac{x_{\text{Poisson}}(I[x,y],s)}{s}. \tag{4.5}$$

It may not be clear at first how the parameter s affects the resulting Poisson noise given how it enters both Equation (4.4) and Equation (4.5). In Equation (4.4) the s parameter appears as a scaling factor of λ, thus changing the effective expectation value, while the division by s in Equation (4.5) brings the expectation value of the noise back to the value of λ. However, although the final expectation value is unaffected by s, the variance of the samples are affected. This behaviour is shown on Figure 4.3, where it is clear that higher values of s lead to less uncertainty in the final sampling distribution, in spite of the probability density function on the left side becoming more broad.

The sequence of degradation steps and their respective effects is shown on Figure 4.4. The shown input image is a randomly selected sample from the high-quality samples. The mean and variance of the Gaussian kernel for the brightness modulation is randomly generated. The s parameter for the Poisson noise and the variance, σ, for the Gaussian noise, are both randomly generated. The noise images are simply added and then modulated via multiplication with the Gaussian kernel Equation (4.3).

An example of an image pair, where the degraded image is output from the degradation model, is shown on Figure 4.5 with the corresponding binary ground truth segmentation image. The ground truth is obtained by thresholding of pixel intensity of the non-degraded image that

is already of high-quality compared to the low-quality example of Figure 4.2, which explains why thresholding works reasonably well.

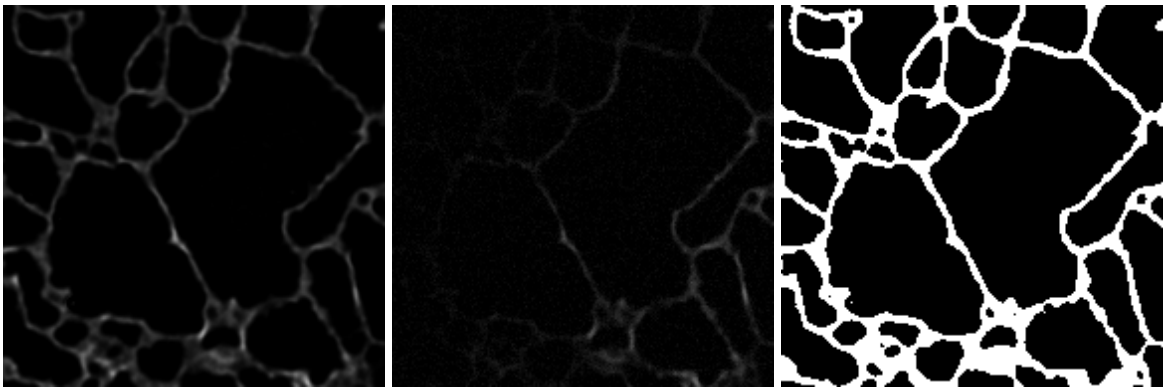


Figure 4.5: Original image and image pair in training dataset. (Left) original relatively high-quality image, (centre) synthetically degraded input image, (right) binarised segmentation image based on thresholding of the original image used as ground truth for training.

4.2.2 End-to-end CNN segmentation model

The training dataset is generated as described in the previous section from 11 relatively high quality 512×512-pixel ER images. A 100 randomly located 192×192-pixel subimages are drawn from each of these source images, where each subimage sample is degraded with a randomly generated set of degradation parameters. In the end a total of 1100 training pairs are available for training. To further enrich the dataset a few data augmentation transformations are applied before feeding a sample into the model, namely any combination of a rotation by 90 degrees, horizontal flip and vertical flip, in total 8 different possible transformations that will make the training dataset slightly more capable.

A separate high-quality source image is reserved for use as a test set, thus providing 100 subimage samples for testing. Finally, a low-quality image akin to that of Figure 4.2 is also randomly sampled but not synthetically degraded for a proper validation of the model's functioning.

The neural network architecture that was found to perform the best is a customised version of the super-resolution model RCAN that also was found to work very well when customised to do denoising in Section 3.3. For the model to be able to do segmentation rather than super-resolution, the final block of the diagram in Figure 3.7 (also note that the ResBlock is slightly different for RCAN as described in Section 3.3) must be a convolution layer that only outputs two channels, one channel for the probability of each of the segmentation classes, i.e. ER or background, being in a given pixel. For simplicity the convolution layer is set to use a kernel of

size 1×1, which ensures the output image has the same dimensions as the input image without applying any padding. The convolution operation is then only over the same pixel across all the feature maps (the existing filter channels that are input to the convolution layer).

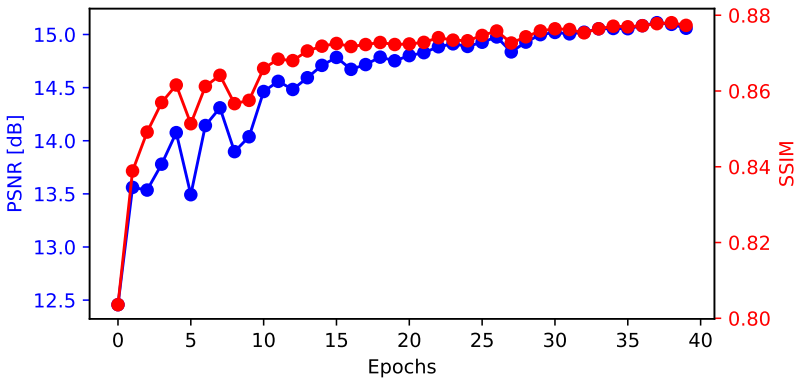


Figure 4.6: Convergence of peak signal-to-noise ratio and structural similarity index during training when model is evaluated on a test set.

The model is trained for 40 epochs, with each epoch iterating through all samples in the training dataset with a batch size of 20. The learning rate was initialised to 10^{-4} and halved every 10 epochs. For every epoch, the model was evaluated on the test set and the typical performance metrics were calculated, namely the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM). A convergence plot of these metrics is shown on Figure 4.6. As is generally the case the metrics are seen to be highly correlated given the coinciding local extrema. After about 30 epochs the improvements in performance start to be marginal. Within the first 10 epochs there seem to be significant fluctuations, which could indicate that the initial learning rate is too high, meaning the step size in each update may cause overshooting of the local minima. But since the learning rate is set to decrease every 10 epochs, the convergence eventually becomes more stable.

4.2.3 Results

The first validation step is to test the trained network on a synthetically degraded image that is separate from the images in the training dataset. This will test whether the trained network has learned to restore and segment images based on the same type of synthetic degradation, and not only have memorised the specific structures in the ER images of the training dataset.

It can be useful to consider outputs based on the test set during training to see how the model learns. Test results at an early stage after just 3 epochs are shown on Figure 4.7. It is clear that the model has not yet learned how to deal with the degradation given how those regions of the

ER that are still relatively clear to the human eye just turn up as background in the model output. Furthermore, the region in the left part of the image that is well-illuminated, near the centre of the Gaussian kernel responsible for the brightness modulation, is not resolved properly in the output. The tubules are very broad and do not appear to have the fine structure observed in the ground truth. This is indicative of neighbouring weights not yet being sufficiently distinctive due to the low number of updates.

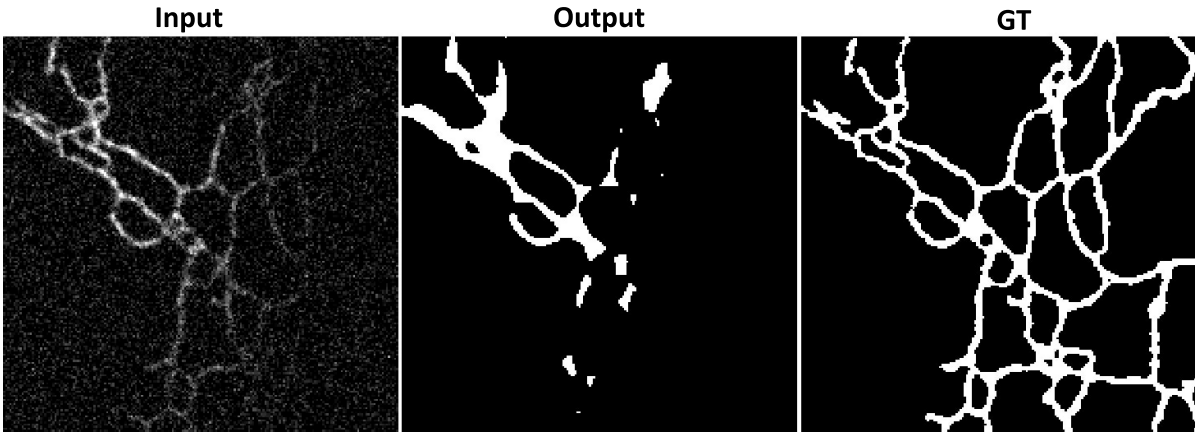


Figure 4.7: Premature test results after training for only 3 epochs.

After 20 epochs the training has converged much better according to Figure 4.6, and indeed the test outputs are found to resemble the ground truths far better, see Figure 4.8. Even in the presence of degradation, the model is able to produce a segmentation map that is almost identical to the ground truth showing nearly the same structural resolution and only a few blanks in the top left corner due to the effects of modulating the brightness.

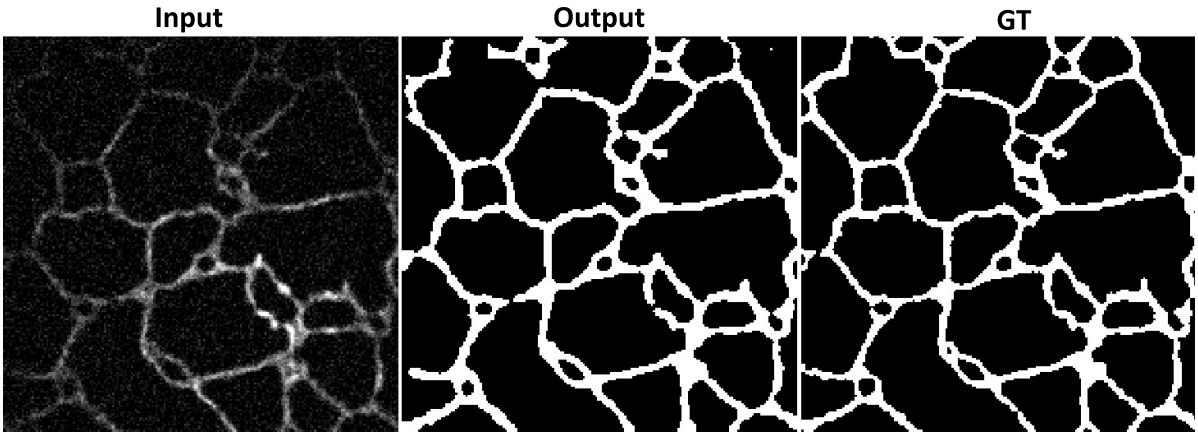


Figure 4.8: Test results after 22 epochs. In spite of the degradation the output closely resembles the ground truth.

Due to the permanent loss of information from the degradation, there are limits to how well the ground truth can be recovered. After 40 epochs where the training is expected to have converged essentially as well as it can for the chosen model, test results still occasionally turn up with significant blanks. One such example is shown in Figure 4.9. It is clear from this example that improvements could be made to the model. Even though the information in the dim region on the right may be unrecoverable, the small isolated patches that do occur in the model output are likely to do more harm than good for any further analysis. Since this is undesirable, the model should ideally be modified to reject isolated blobs. This could potentially be remedied relatively easily by customising the loss function used in training to penalise the presence of disconnected pixel clusters.

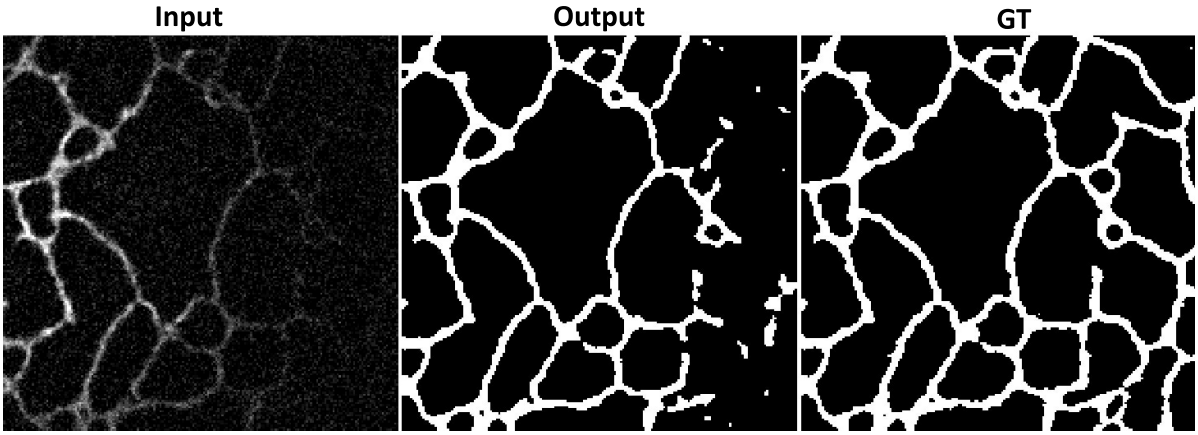


Figure 4.9: Test results after 40 epochs. Although the model is able to recover the ground truth in most of the image, the right side is so degraded that little can be done.

The final validation that will determine whether the model is useful in practice is to test on images that are experimentally degraded. This will test whether the network has generalised so well that it is able to recognise the typical structure of ER and reconstruct it even when the type of degraded input images have never been seen before by the network. These test images are fed directly into the model without any preprocessing in the form of the synthetic degradation. An example of an experimentally degraded input and the corresponding output can be seen in the top row of Figure 4.10. The network structure of the ER is clear in the segmentation with only relatively few patches that are disconnected from the network that should ideally either have been rejected or connected via inpainting (content-aware restoration whereby blanks in the image are filled out) in the most realistic way. For comparison a manually fine-tuned thresholded segmentation image is also generated as well as the segmentation result from a built-in Fiji plugin called WEKA [123] – see bottom two rows in the first column of Figure 4.10.

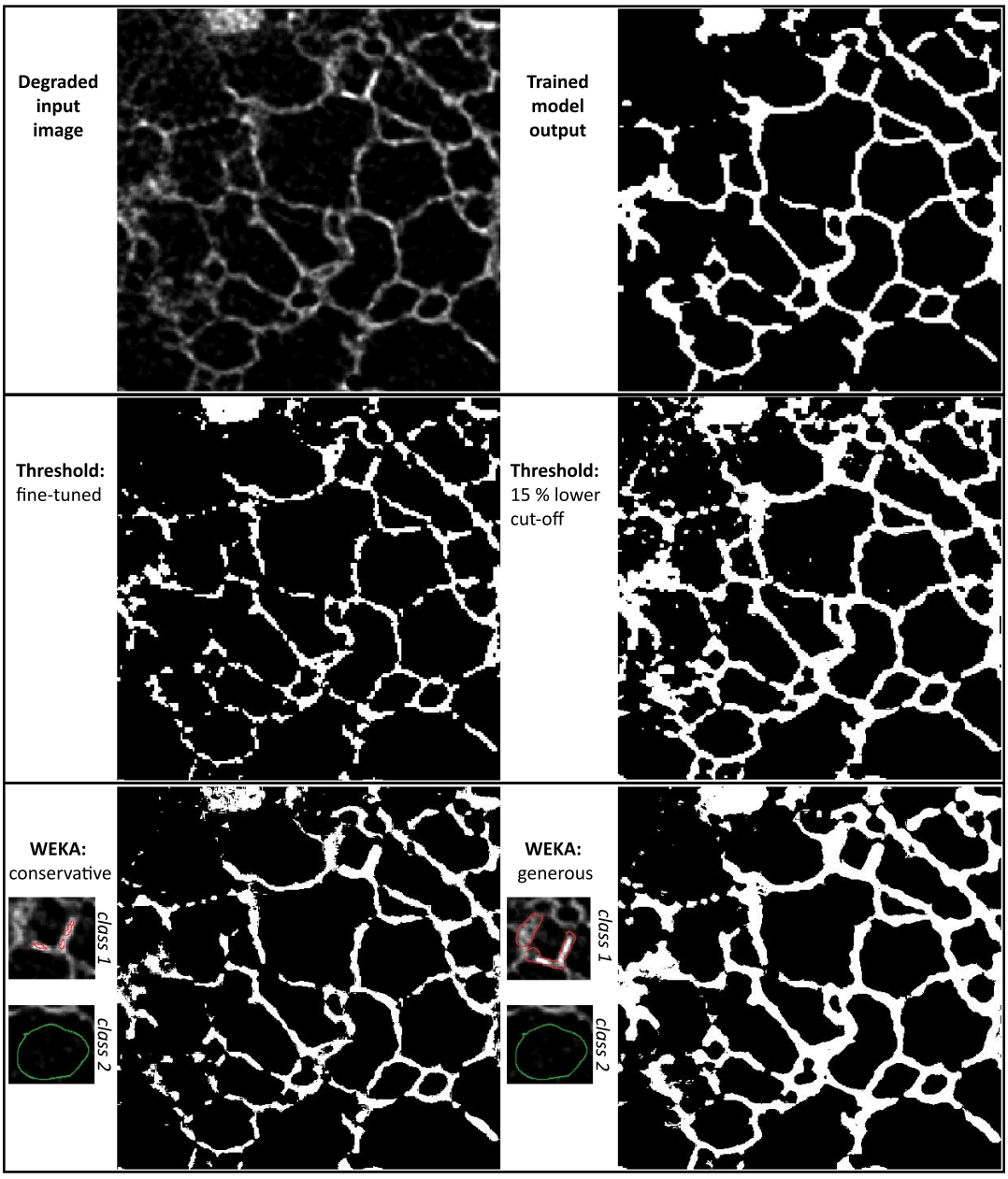


Figure 4.10: Comparison of segmentation maps from different methods made from a test set of experimentally degraded images. The ”trained model output” refers to the neural network model that has been implemented and trained on a synthetically degraded training dataset. In spite of this the model is seen to work well even when the synthetic degradation is no longer present. The other methods are more simple: thresholding by pixel value (grayscale intensity) and a plugin in Fiji called WEKA that uses random forests. Both of these other methods have to be manually tweaked for each image that is to be segmented, while the neural network is more versatile and works directly after having been trained on the separate synthetically degraded dataset.

The segmentation maps from these alternative methods are seen to have many disconnected tubules. To force some of these tubules to connect in the segmentation map, test outputs were also made with the methods configured to be less conservative – see the second column of the bottom two rows in Figure 4.10. For the thresholding approach this means that the cut-off value, for the pixel intensity, for what constitutes background is simply lowered. A 15 % lower cut-off value clearly produces a better connected network structure, but at the price of more isolated patches occurring on the left side of the image in addition to some resolution loss given that all the tubules become significantly broader.

The WEKA plugin uses basic machine learning, by default random forests, to perform the segmentation. The method requires the specification of areas in the input image that correspond to the respective classes, ER and background, after which the classifier can be trained and then run on the entire image. Similarly to the thresholding approach, it is possible to control how conservative the method is towards classifying something as ER. By specifying more dim areas of the ER as part of the example class data that is used for training the classifier, the trained model will become more ”generous” with respect to classifying something as the ER. The areas of the input image used for this example class data can be seen in the miniature images of Figure 4.10. The less conservative output is again seen to have a more well-connected network structure, but with significantly worse resolution due to the tubules being much broader to the point where the shapes look slightly distorted. However, the presence of isolated patches has become worse, presumably because the selected areas in the example class data are larger, thus allowing the model to filter out regions that are significantly smaller than those selections.

For the example in Figure 4.10, the neural network model achieves both an accurate resolution and very few patches, thus not being affected by the trade-off between resolution and number of patches found for the thresholding approach. It is also worth noting that the neural network has not been configured in any way to deal with the experimental test images, whereas the two other methods were either fine-tuned to achieve the best results or in fact trained on the very image itself, which arguably defeats the purpose of the segmentation tool. As such it is clear that the neural network model is far superior in terms of versatility. For the other examples in the experimental test set the same applies: the output from the neural network model appears clean and consistent, and qualitatively better than the alternative methods, but no experimental ground truth images have been available yet, so a quantitative comparison with performance scores has not been possible.

There is room for improvement, both with respect to the training data but also possibly by modifying the model with a more suitable loss function that penalises disconnected patches, but overall the results are promising given how consistent and versatile the model turned out even with the relatively sparse training data available.

4.2.4 Fully synthetic data generation

The approach to segmentation of ER images described so far has been based on a partially synthetic dataset, where the inputs are simulated based on target images that are taken to be segmentation maps of relatively clean experimentally acquired images. We will now consider how the targets as well can be synthesised using a simple model for simulating ER networks. This has multiple advantages: firstly, the ground truths can be made ideal, i.e. no imperfections, and secondly, the size of the dataset and the frequency of less common degradations, such as the apparent disconnections of tubules seen in Section 4.2.3, can be computationally controlled. This is important as the thresholded segmentation maps of the clean ER images previously used as targets, see Section 4.2.1, are only approximations of the real ground truths. The imperfections of the approximation may inhibit the model from learning an accurate representation of the ER network. In light of the positive results of using the self-supervised Noise2Noise and Noise2Void training strategies described in Section 3.4, it is likely that the noise manifesting into the target segmentation maps, using thresholding as in Section 4.2.1, averages out during training similarly to zero-mean noise source in a model trained with the Noise2Noise principle. However, given noisy data, it is difficult to address the issue of visually appearing disconnected tubules described in Section 4.2.3 because the target data when prepared either with thresholding or WEKA also will contain the same disconnected tubules, thus hindering the model to properly learn to compensate for it. By using simulated ground truth images, the problem of acquiring clean data is avoided, and it is possible to generate a diverse dataset by using the image formation model introduced in Section 4.2.1 that reflects the apparent disconnects of tubules in the input images, while providing a fully intact ER network as the target.

Given a dataset of the ER generated in this fashion, a neural network can be trained to reconstruct the shape of the ER from the degraded images. However, the more degraded the images are, the more ill-posed the reconstruction problem is and the uniqueness of a solution diminishes. For highly ill-posed problems, a standard approach to training a model, e.g. training a CNN with supervised learning using a MSE loss function, tends to lead to conservative inference as described in Section 3.2. In contrast, as also noted in Section 3.2, a generative adversarial network (GAN) is more at liberty to distort the output, thereby being able to compensate for entirely missing information such as the disconnected tubules.

The purpose of this section is two-fold: (a) to demonstrate how ER can be modelled in a simple way, and (b) to qualitatively explore the potential of GANs for ER image segmentation.

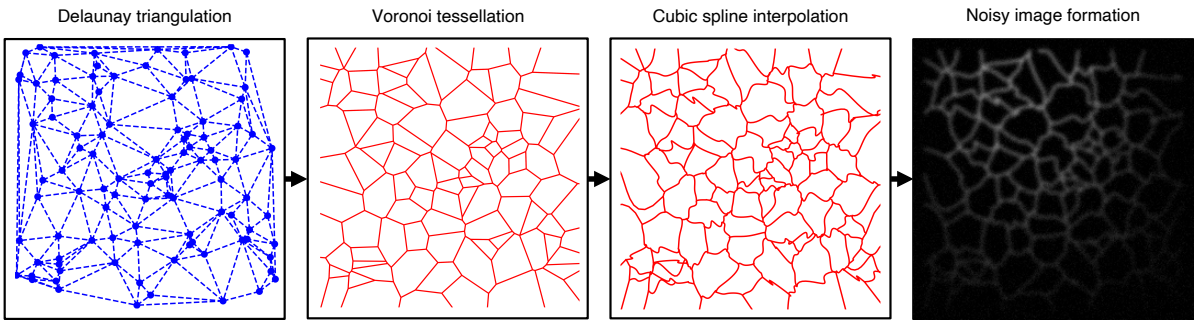


Figure 4.11: Fully synthetic data generation pipeline based on randomly generated uniformly distributed points.

Modelling the endoplasmic reticulum

The model proposed for synthesising the ground truths follows a simple algorithm. First, a set of uniformly distributed coordinates are generated. Delaunay triangulation is performed on the coordinates providing a set of triangles. The circumcentre of each triangle becomes a vertex in a Voronoi diagram. The diagram is formed by connecting vertices between adjacent triangles. This procedure is called Voronoi tessellation. Each edge in the undirected graph corresponding to the Voronoi diagram is perturbed in the following way: the midpoint of an edge is moved in a random direction by a random distance that is smaller than the length of the edge. The displaced midpoint and the two ends of the edge are then used to produce a smooth curve using cubic spline interpolation of the three points. Finally, the network of vertices connected with interpolated curves is processed with the image formation model specified in Section 4.2.1 producing the final synthetic input image. The target image is the corresponding image before noise and brightness modulation are applied. The steps of this algorithm are shown on Figure 4.11.

By generating thousands of training pairs akin to this example, it is possible to train a model to reconstruct the shape of the ER given the degraded inputs. As mentioned in the beginning of this section, the application of GANs to this ill-posed segmentation problem could prove suitable although with the potential pitfalls posed by introducing distortions as addressed in Section 3.2. A GAN model with the previously described modified RCAN architecture used for both generator and discriminator, the two components of a GAN [54], has been trained using 1000 generated training samples. An example output is shown on Figure 4.12, where two regions are highlighted across three versions of the image: turquoise and red rectangles on output from a lightly trained GAN model, output from a well-trained GAN model and the original ground truth image, respectively. The example demonstrates that the GAN model ends up learning to perform a mostly faithful segmentation even when there is only a very dim, or

missing, signal, but as seen in the red rectangle the ability to reconstruct the tubules from scarce information can also lead to the creation of spurious connections.

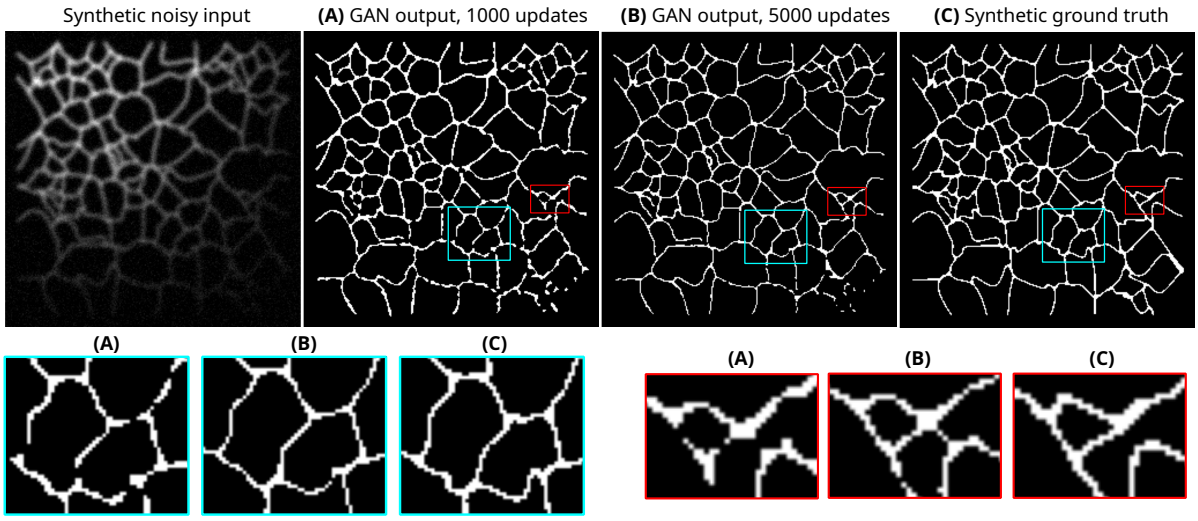


Figure 4.12: Example output from a generative adversarial network (GAN) model using the fully synthetic training dataset. The restoration performance appears impressive, but the risk of introducing false features is high as indicated by the cropped regions with red border.

Discussion

To force a trained model to focus specifically on the issue of tubules that appear disconnected, a randomised truncation of tubules can be added to the process depicted on Figure 4.11. An example of a training pair using this sequence of steps for data generation is shown on Figure 4.13 with the disconnected tubules emphasised on the cropped regions.

Introducing these "explicit" disconnects between tubules in the training data could potentially improve segmentation performance further, but has only been preliminarily explored in the PhD project. Early results indicate that when using GANs, the risk of spurious reconstructed tubules manifesting in the output image is significant.

However, if an application would require conservativeness in inference over the ability to produce more complete reconstruction output, then the same synthetic image data could instead be used with a regular segmentation model as seen in Section 4.2. This is expected to lead to models for which the presence of spurious connections, i.e. artefacts, in output would be kept minimal, while still learning to be adaptive and robust to missing data or data with low signal-to-noise ratios. Another direction, also only preliminarily explored in the PhD project, is to use a custom loss function that penalises disconnected tubules and favours those that are intact. One such loss function that has been tested is a topological loss function [22] that uses

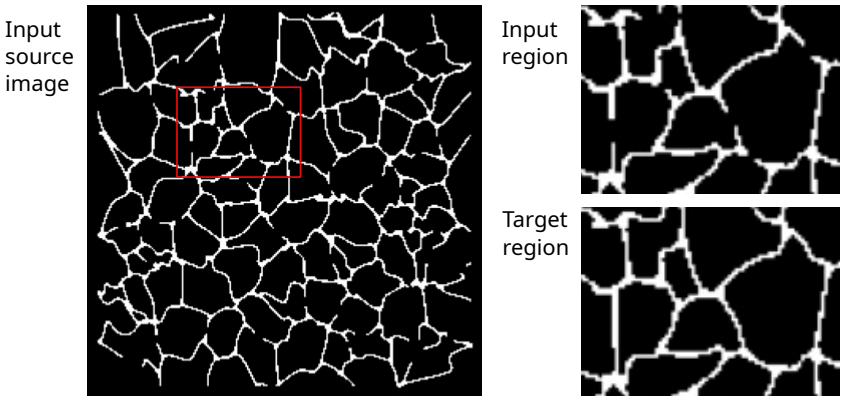


Figure 4.13: The data generation and image formation algorithm can easily be modified to produce input images with non-smooth disconnects. This poses a more difficult segmentation problem for the neural network as there is a complete lack of information, thus requiring an educated guess by the model that can be achieved with an adversarial loss function.

persistent homology to optimise for topology-aware output. Initial results are not promising, but it is possible that this avenue could prove fruitful if explored further.

4.3 Supervised segmentation model

In this section, I will describe work on ER segmentation that follows a more standard supervised approach, which was used to analyse experimental data featured in a publication that I co-authored [127]. The simulation-supervised and fully synthetic dataset generation described in the preceding sections represent an alternative approach to the one taken here. Rather than modelling the noise sources and image formation, input data is simply raw wide-field images that have been experimentally acquired. The target data is prepared by manually annotating images using an image editing software. To speed up the annotation process, the manual labour can rather be performed as corrections to an initial segmentation map produced by thresholding or segmentation with the WEKA plugin described earlier in this chapter.

4.3.1 Residual neural network for segmentation

The segmentation of the tubular networks of the endoplasmic reticulum is again carried out using a convolutional neural network (CNN). The network architecture of choice is a deep residual network inspired by EDSR and RCAN [114, 227]. These models are among a class of residual learning networks such as the ones used in Chapter 3 for image restoration.

A proposed network that performs this pixel-wise classification to output segmentation maps is shown on Figure 4.14. The architecture with a sequence of blocks surrounded by convolutional layers and a long skip connection is the same for both EDSR and RCAN. The definition of the block is different between the two, and for simplicity the EDSR block is shown, although both blocks were used in testing. In general, the RCAN based block was preferred. The main difference from the EDSR and RCAN architectures to that of Figure 4.14 is the replacement of the super-resolution block, an upsampling module, with a module that decodes all the feature channels from previous convolutional layers into class scores. This is done using feature pooling, in which a convolutional layer with kernel size 1×1 reduces the number of feature channels to the number of unique classes in the segmentation map.

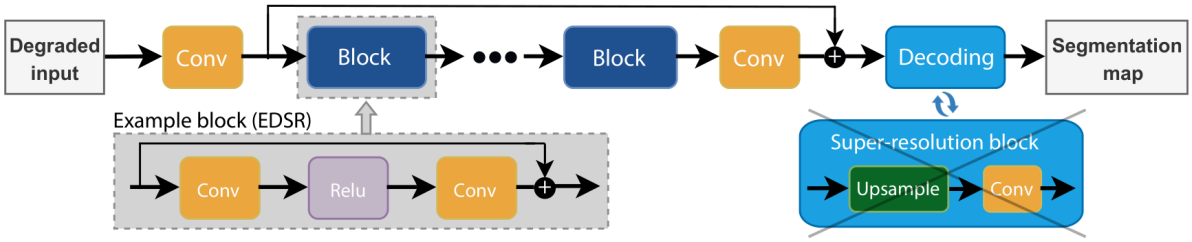


Figure 4.14: Architecture of the residual CNN used for segmentation. The overall structure follows that of EDSR and RCAN, except for the replacement of the super-resolution block with a decoder module that reduces the number of feature channels to the number of unique classes in the segmentation map using a convolutional layer with a corresponding number of output channels and a kernel size of 1×1. This operation is sometimes referred to as feature pooling.

Given appropriate training pairs this network can learn to map low signal-to-noise ratio images into clean segmentation maps. Since the network is capable of restoration, it is not necessary to explicitly denoise images prior to inputting them to the model. However, to alleviate the complexity of training the network, raw images were first denoised using the denoising method ND-SAFIR [15], which includes a noise parameter estimation of Poisson-Gaussian noise that is typical for optical microscopy.

As for preparing the training data, a very crude segmentation could be performed using grayscale pixel intensity thresholding after applying this denoising method to raw images. A few of these segmentation maps were then manually cleaned and finalised by drawing in a raster graphics editor. These partially hand-drawn segmentation maps then served as targets, i.e. ground truths, in the supervised training of the segmentation network.

To make it feasible for the network to learn to segment the ER images from the relatively small training dataset, multiple ways of data augmentation were used. Firstly, each segmentation training pair was randomly cropped many times, which shifts the structures from frame to frame and brings the image size down to a manageable size for a graphics card (256×256 pixels).

This provided a few hundred subimages of different regions of the segmentation examples. Those subimages were then randomly flipped (horizontally and vertically) and rotated (by 90 or 180 degrees) to obtain more training data. Other ways to augment data could have been to randomly change brightness or synthetically add noise to images, but this was not found to be necessary.

Training was done in batches of five images (each being 256×256 pixels) with a learning rate of 0.001 using the Adam optimisation method for a total of 40 epochs. The learning rate was halved after every ten epochs. The network was configured to have four blocks, of the RCAB, which amounts to 40 convolution layers each having 64 filters with a 3×3 kernel size, constituting a total of about 1.3 million trainable parameters. The trained network outputs binary segmentation maps that can then easily be skeletonised by a standard thinning algorithm [107].

The implementation has been made with the machine learning library Pytorch. The code as well as training and test data for segmenting endoplasmic reticulum images is freely available at https://github.com/charlesnchr/ERNet.

4.4 Segmentation of sequential images

In an extension of the work described in Section 4.3, improvements have been made to utilise the spatio-temporal information in video data of the ER. In addition to this, a more extensive analysis pipeline has been introduced to quantify ER dynamics and morphology over time. This subsequent work is reported in the pre-print article [126] co-authored with group member Meng Lu, who has acquired experimental data and assisted with data analysis, and Jana M. Weber who has contributed with methods from the field of graph theory. This section largely follows this pre-print and the work that led to it, but I only include the parts that are relevant to the thesis. From the time of writing the paragraphs below, further work has been carried out in relation to preparing the pre-print manuscript, which means the results described here predates those in the publication. However, as the overall methodology has not changed, I have kept this section largely unchanged. One significant difference is the model architecture, which in the case of the published segmentation method is based on a transformer network, while a CNN similar to that in the previous sections but modified for spatio-temporal data is used below. In Section 5.3, I will describe this most recent transformer-based model in more detail, albeit in the context of structured illumination microscopy.

4.4.1 Processing pipeline

Using structured illumination microscopy (SIM), a stack of image frames for every time point is acquired, each with a distinct illumination pattern. These frames are reconstructed with a SIM reconstruction method for which there are two methods that we have used, ML-SIM [30] and FairSIM [146]. The reconstructed super-resolution images form a time-lapse sequence of images that are processed with ERnet to obtain segmentation maps. ERnet is applied using a temporal window that includes multiple adjacent frames to improve the segmentation performance by utilising similarities of the temporally correlated sequential image data. This approach is more robust and provides higher segmentation quality than processing the batch of sequential frames one-by-one. This is rationalised by the fact that any single frame is more prone to random noise and imaging artefacts, whereas a set of sequential frames collectively contain more information about the background and structure of the sample, thus for example having a higher signal-to-noise ratio. The performance benefits of this scheme are presented in Section 4.4.3. The resulting segmentation maps are skeletonised using a standard algorithm [107], whereafter nodes and edges of the skeleton are identified, such that a graph representation of the network can be determined. The graphs are non-spatial in the sense that the physical locations of the nodes are disregarded, hence specifically representing the network aspect of the ER, e.g. connectivity dynamics and branching. Finally, various metrics such as average node degree and assortativity, which are standard in graph theory, are extracted from the graph representation of the ER, constituting a unique new way of quantifying the ER. A simplified overview of the pipeline is shown on Figure 4.15, while a more complete version is described in Section A.1.

4.4.2 Architecture of spatio-temporal extension of ERnet

As noted in the beginning of this subsection, the spatio-temporal extension of the ERnet model exists with two different architectures; a CNN that directly evolved from Section 4.3.1 and a more recently developed transformer network as described in the co-authored pre-print publication [126]. Below we will consider the former of the two versions, and instead consider the transformer model for a different application inSection 5.3.

The CNN version of the spatio-temporal extension of ERnet has a residual network architecture. However, preceding the residual blocks of the network is a single convolutional layer that is applied to each of the individual temporal frames in the video sequence according to the temporal window. The temporal frames are then concatenated in feature space to form a feature map where each channel represents a different time point. In addition to the residual network structure, ERnet utilises a channel attention mechanism based on Residual Channel Attention

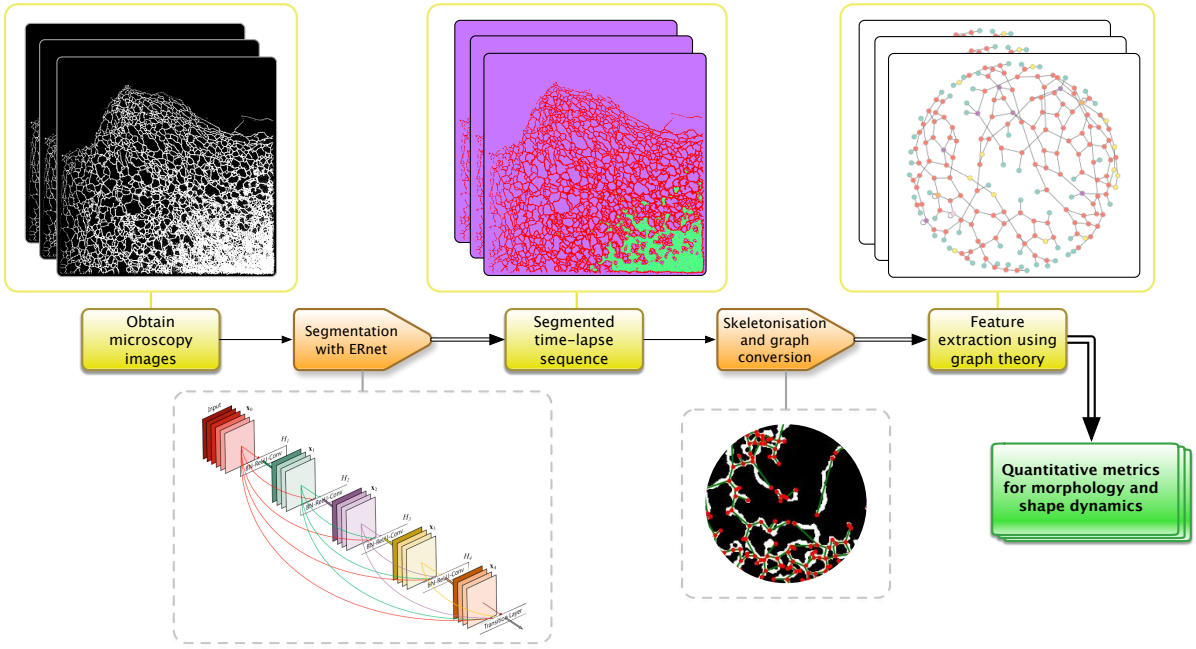


Figure 4.15: The processing pipeline takes in a sequence of fluorescence microscopy images, e.g. wide-field or reconstructed structured illumination images. The sequence is a time stack acquired with a fast imaging speed. The images can be multi-colour, but only the colour related to the endoplasmic reticulum is processed onwards. A moving window consisting of adjacent frames from the sequence of images is then input to a deep residual neural network, ERnet, in order to exploit the temporally correlated information in the time sequence. The network performs automatic and robust segmentation of the ER tubules, sheets and background, i.e. a multi-class segmentation problem. The segmentation map of the ER tubules is then binarised so that a standard skeletonisation algorithm can be run. Finally, the skeletons are converted to graph representations for further analysis.

Network (RCAN) [227]. Here the standard additive skip connections are complemented by another type of connection that combines tensors before and after a block using element-wise multiplication with learned weights Figure 4.16. This allows the model to learn to adaptively rescale channel-wise features by considering the interdependencies among channels. This is beneficial in the context of the sequential image data that is processed by ERnet, since each channel is chosen to represent a frame of the sample at a different point in time, which leads to frames that are highly spatially correlated depending on the frame rate.

4.4.3 Training and benchmark of ERnet

Multiple ERnet models were trained on a manually annotated training dataset. As a baseline method, the popular U-Net is used [173]. The models vary in the number of residual groups (RG) and the size of the temporal window (N). In the following, three ERnet models will be

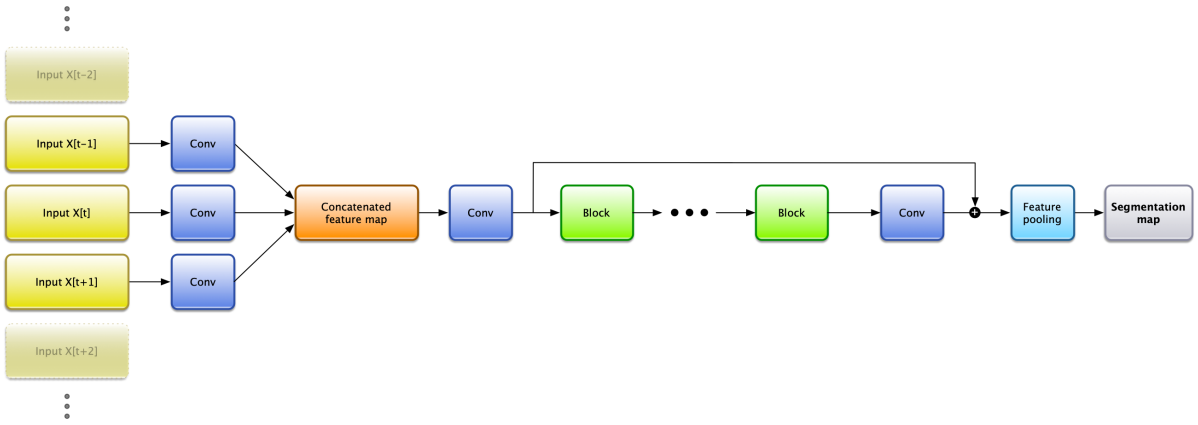


Figure 4.16: The ERnet model uses a deep residual neural network structure architecture based on the RCAN model. The final layer is modified with a dense layer to output integer classes. For improved performance using the temporal information the inputs are concatenations of adjacent frames after passing through a single convolutional layer.

compared: a small model with RG = 3 and N = 1, a large model with RG = 5 and N = 1 and a complete model with both RG = 5 and N = 3. As for U-Net, two models with respectively N = 1 and N = 3 are tested.

The performance on a separate validation dataset is evaluated during training to ensure all models have converged, cf. Figure 4.17.

After training the model performance is evaluated on a larger test set to obtain accurate averages of the segmentation performance using intersection over union (IoU) as a metric with. The results are depicted on Figure 4.18.

4.4.4 ERnet graphical user interface

During the PhD project, I have developed a graphical desktop application that can run Python implementations of neural network models using the deep learning library Pytorch. The primary motivation for developing this software has been to run reconstruction of structured illumination microscopy images with the method ML-SIM covered in Chapter 5. The program has grown to include segmentation models by the use of a plugin system, such that its functionality can be switched from running ML-SIM models to ERnet models. The original program is described further in Section B.1.1, and the plugin-based version, *Mambio* (**M**ulti-**a**nalysis **m**achine learning-enabled **b**atched **i**nput-**o**utput), is shown on Figure 4.19. The software is open-source and available at https://github.com/charlesnchr/ERnet-v2.

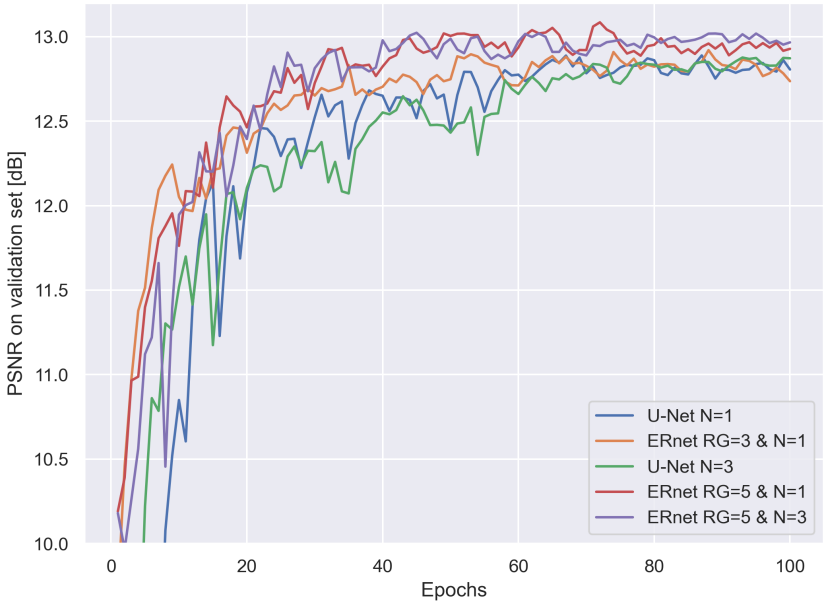


Figure 4.17: Convergence plot for models trained on a supervised, manually annotated, dataset of segmented ER images. Performance is measured in peak signal-to-noise ratio (PSNR). Performance differences in the well-converged regime after about 80 epochs indicate that ERnet with 5 residual groups and a stencil of 3 frames is the best, which is consistent with the more rigorous test results shown on Figure 4.18.

4.4.5 Quantitative analysis of dynamic ER structures

Here I will report quantitative results obtained using the methodology described in the preceding sections for a set of experiments in which the effect of different drugs on ER morphology is investigated.

Video recordings were captured using SIM to study the highly dynamic tubule and sheet regions of ER. The pipeline of Figure 4.15 was then used to analyse the data. When used on this data, ERnet provides quantitative information about the movement and structural changes of tubules in ER, which has previously been reported to be associated with disease phenotypes [126]. ERnet was first tested using SIM images of COS-7 cells by quantifying these intracellular changes. Figure 4.20 shows a single frame of the ER (gray) from a set of sequential images. The resulting segmentation map includes the entire ER structure, which is differentiated from the cytosol background. The ER structure is further classified into tubular (cyan) and sheet domains (yellow). The tubular ER is then skeletonised from the segmentation map, which

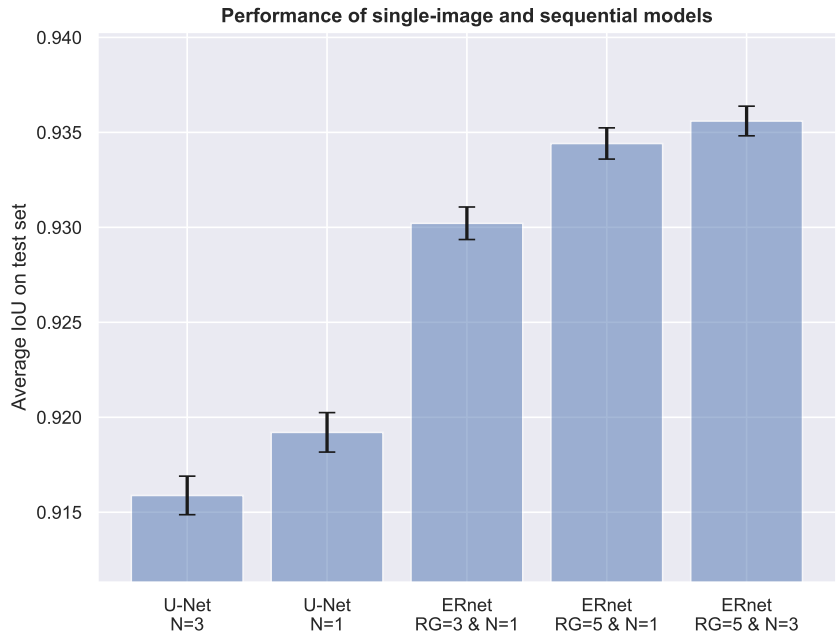


Figure 4.18: Test scores on a test set separate from the training dataset for the different segmentation models. Performance measured in intersection over union (IoU). Error bars indicate the standard error of the mean IoU across 2000 test images. ERnet with 5 residual groups and a stencil of 3 frames is found to be the best model by a significant margin, validating the idea of using the architecture.

provides nodes (tubule junctions, shown in red) and edges (tubules, green) that are subsequently used for graph analysis.

The network connectivity described by the graph representation can be visualised with the Python package *graph-tool* [161], which produces a connectivity graph as shown on Figure 4.21. The connectivity graph highlights that the ER network primarily consists of three-way junctions (red nodes in Figure 4.21) and tubular growth tips (green nodes in Figure 4.21).

The integrity of the ER is assessed by considering each disconnected ER region a fragment. As the ER is constantly reshaping, the total number of fragments fluctuates during each acquisition as shown on Figure 4.22c. However, despite these continual structural changes, ERnet reveals that in typical healthy cells, a single large fragment comprises the majority of all edges and nodes over the entire imaging duration. As quantitative parameters, the node and edge ratios are defined as the number of nodes or edges in the largest fragment divided by the total number of nodes or edges, respectively, see Figure 4.22b. By definition these values range from close to 0 (fully fragmented ER) to 1 (fully connected). Additionally, ERnet quantified the degrees of the ER nodes, i.e. the number of edges (tubules) that connect to each node (junction).

Figure 4.19: ERnet plugin for Mambio, an Electron-based desktop application capable of running deep learning models, adds support for performing segmentation with ERnet and the subsequent analysis steps described in Section A.1. Different models can be loaded and images can easily be batch processed.

As shown in Figure 4.22a, three-way junctions are the most abundant and account for 78 % of all junction types in this example. Despite the prevailing picture of ER morphology as a tubular network of interconnected three-way junctions, ERnet also identified nodes connected with more than three edges (tubules), i.e. multi-way junctions. The presence of multi-way junctions indicates that ER tubules may have a more complex, heterogeneous organisation than previously thought.

The assortativity and clustering coefficients as shown in Figures 4.22d and 4.22e, which describe connectivity patterns of nodes, were calculated based on the above metrics. The assortativity coefficient measures the tendency of nodes to connect with others of the same degree [152], while the clustering coefficient reflects the tendency of nodes to cluster together. Assortativity coefficients range from -1 (fully heterogeneous connectivity, i.e. nodes only connect with those of different degrees) to 1 (fully homogeneous connectivity, i.e. nodes only

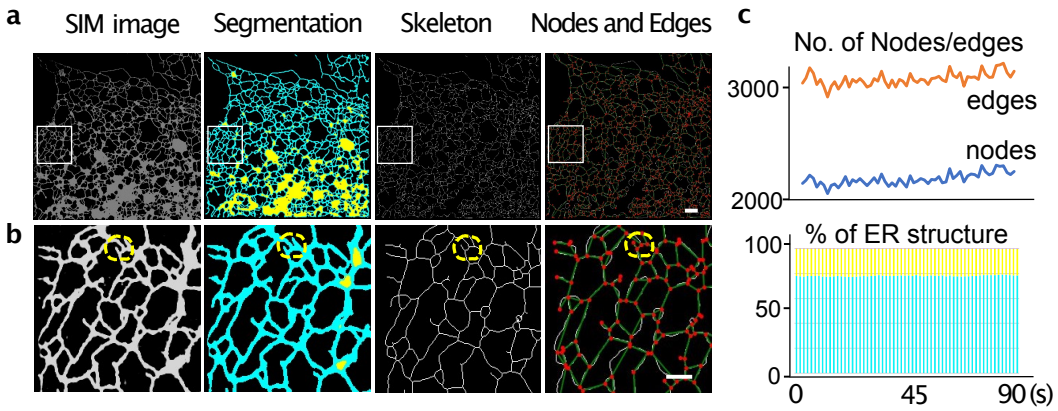


Figure 4.20: Segmentation, skeletonisation and graph conversion of sequential SIM images of the ER. **(a)** Full field-of-view images. From left to right: (1) SIM image, (2) segmentation of image into ER tubules (cyan) and sheet region (yellow), (3) skeletonisation of the tubular domain, and (4) identification of nodes (red spots) and edges (green lines) based on the skeleton structure. Scale bar: 5 um. **(b)** Zoomed-in regions of the above panel. The yellow dashed circles indicate nodes that are closely positioned but can still be identified by ERnet. Scale bar: 2 um. **(c)** Quantitative analysis of the ER shown in (a). Top panel: quantification of edges and nodes of the ER tubules of the sequential frames over a period of 90 s. Bottom panel: percentage of the ER tubules (cyan) and sheet (yellow) over the same period.

connect with those of the same degree). Clustering coefficients describe another aspect of a node's connectivity: they measure if the neighbouring nodes of a given node tend to connect to each other, i.e. to cluster. Similarly, for clustering coefficients, 1 describes a perfectly clustered network, while 0 signifies no clustering. Figure 4.22d shows the ER as a slightly assortative network, which suggests a tendency of nodes to connect with nodes of the same degree. Additionally, the low clustering coefficients of Figure 4.22e implies a lack of aggregation of nodes and edges in the overall ER of this cell.

To further investigate the structural dynamics of the ER, the lifetime of multi-way junctions was tracked along with their transitions from multi-way to three-way junctions. Figures 4.23a and 4.23b show the rapid transitions between three-way (yellow arrows) and multi-way junctions (blue arrows) driven by ER tubule reshaping. The formation of four or five-way junctions needs simultaneous connections of more than three tubules at the same junction, which occurs with a lower probability than the formation of a three-way junction that only requires the connection of three tubules. Additionally, any movement of a tubule away from its multi-way junction can lead to the collapse of this junction and the generation of at least two three-way junctions. Therefore, as shown in Figure 4.22f, the average lifetime of a multi-way junction is much shorter, i.e. less than a third (10.1 s vs 30.8 s) of that of three-way junction.

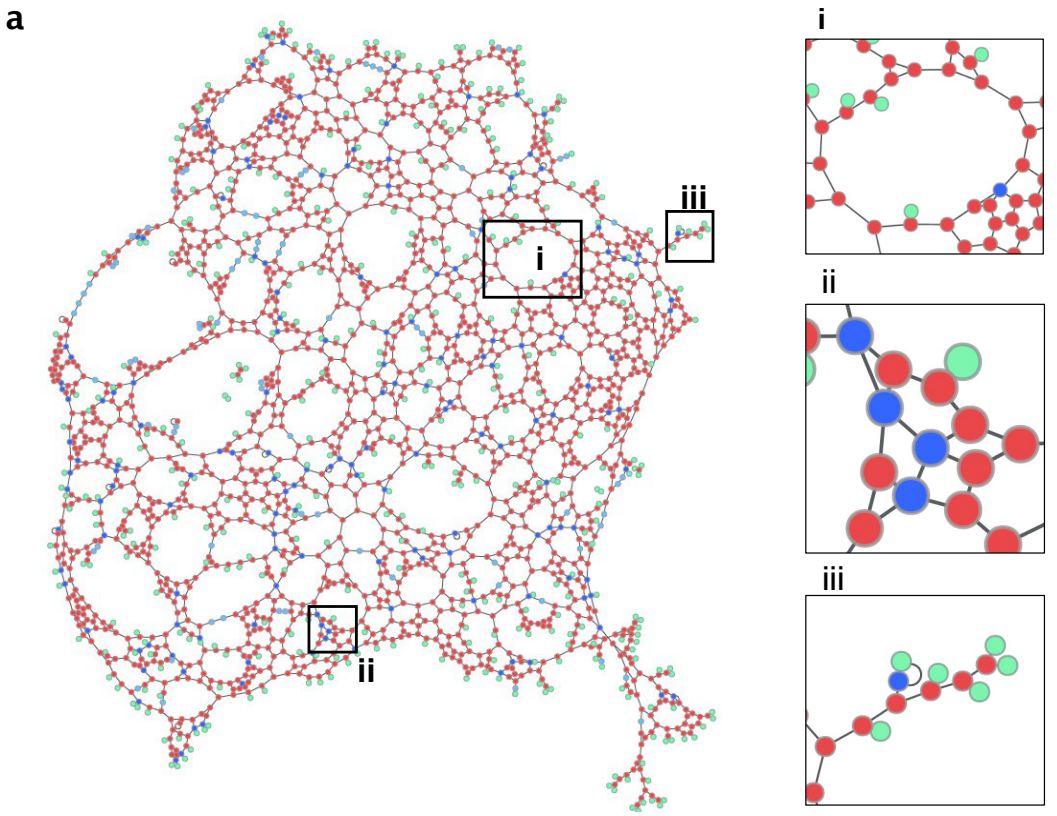


Figure 4.21: The topology of an ER tubular network is represented by a connectivity graph. i: a polygonal structure organized by three-way junctions (red spots) and tubules (gray lines), ii: a representative region of multi-way junctions (dark blue spots), iii: a representative region of ER tubular growth tips (green spots).

4.4.6 Identification of phenotypic characteristics with ERnet

As mentioned in Chapter 4, ER morphological defects have been linked to a variety of human diseases [212]. These defects may be caused by mutations in genes, encoding proteins responsible for reshaping ER, or by metabolic perturbations in the cell. However, the exact phenotypic ER behaviour under these conditions has not yet been thoroughly characterised. Using ERnet, the ER morphological defects in stress models mimicking the ER phenotypes in two neurodegenerative diseases, namely Hereditary Spastic Paraplegias (HSPs) and Niemann-Pick disease type C (NPC), can be analysed. ERnet was used to examine the ER morphology defects in individual cells of different models by measuring two topological features. The selected features are the degree of ER tubule fragmentation, quantified as the node ratio defined in Section 4.4.5, and the heterogeneity in the tubular connections as given by the assortativity also defined in Section 4.4.5. Compared with control cells, it was found that Atlastin (ATL) knock-out (KO) leads to a collapse of the ER network integrity. Such ER fragmentation was

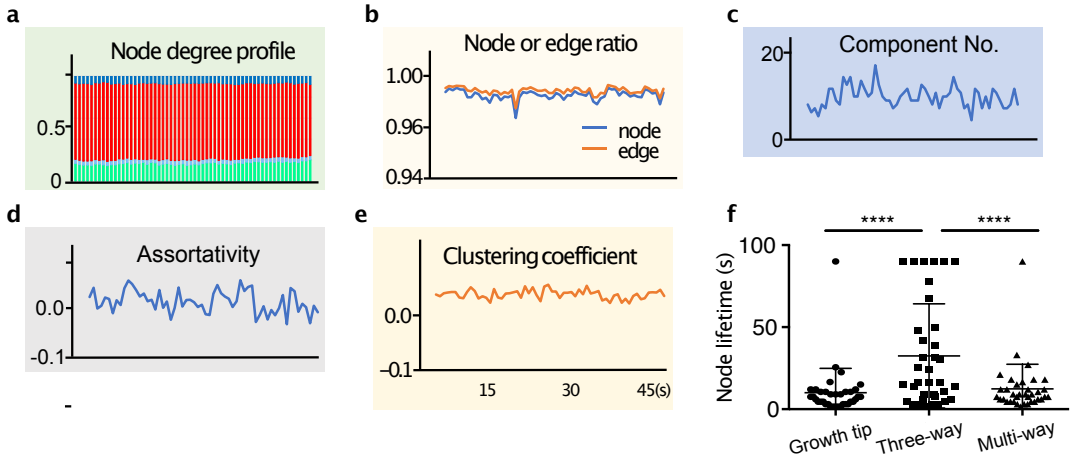


Figure 4.22: Quantitative analysis of the cell shown in (a) over a time window of 45 s. **(a)** Quantification of the nodes of various degrees over time, showing a dominance of third-degree nodes (three-way junctions). Same colour scheme as in Figure 4.21. **(b)** Changes in the node and edge ratio over time. **(c)** Number of components (ER fragments) over time. **(d-e)** Changes in assortativity and clustering coefficients over time. **(f)** Quantification of the lifetime of junctions (nodes) with various degrees. $****$: $P < 0.0001$, Tukey's one-way ANOVA with $n \geq 20$ events per condition from three independent experiments.

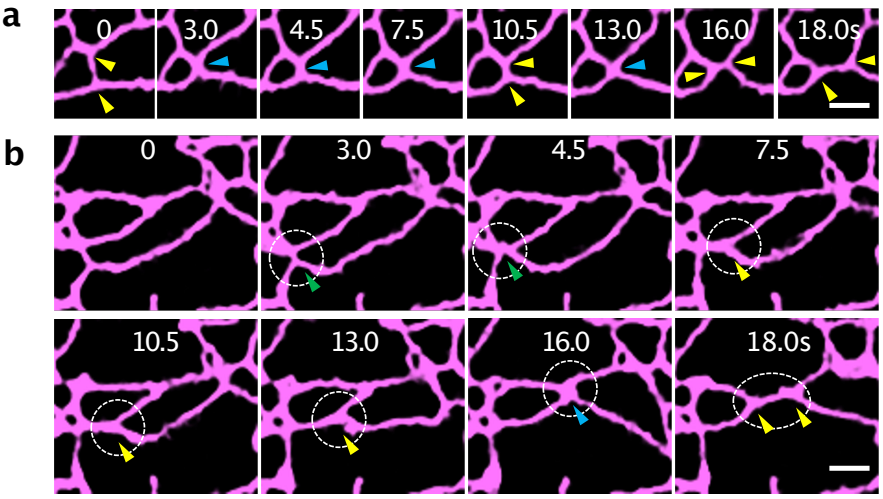


Figure 4.23: Examples of transitions between three-way (yellow arrows) and multi-way junctions (yellow arrows: three-way, blue arrows: four-way, green arrows: five-way) junctions. Scale bar: 1 µm.

clearly observed in ATL KO cells by an increased number of fragments and 20-fold reduction of the node ratio (99 % in control vs. 5.4 % in ATL KO), see Figure 4.24. ERnet also highlighted that the lack of ATL significantly altered the connectivity in ER tubular network as seen by a reduced proportion of three-way junctions among all the nodes (26 % vs. 78 % in control) and by the heterogeneous connectivity (assortativity of -25). These measurements provided

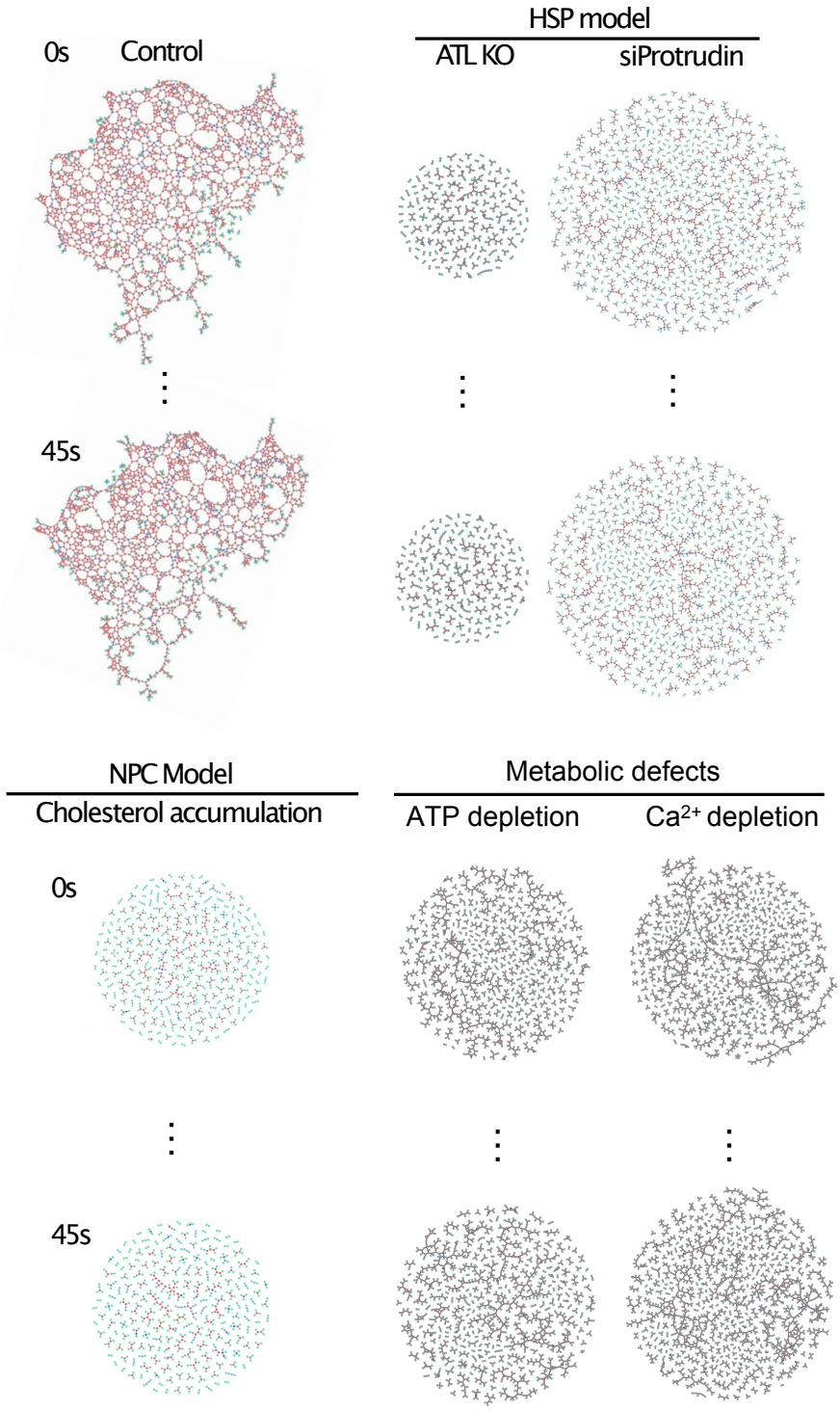


Figure 4.24: Connectivity graphs of ER structures in models mimicking phenotypes of HSPs and NPC and metabolic stress induced by calcium and ATP depletion. Nodes of different degrees are labelled with different colours: green (degree 1), light blue (degree 2), red (degree 3), dark blue (degree > 3).

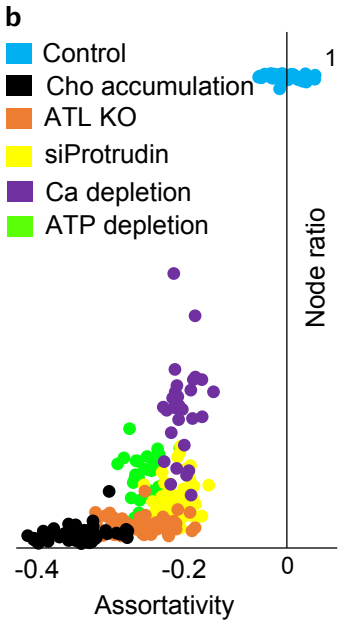


Figure 4.25: Topological features of the ER tubular network in above conditions were quantitatively analysed by ERnet. The effects on ER structures from different treatments can be visualised and compared by plotting the distribution of tubule fragmentation (node ratio, y-axis) and assortativity coefficient (x-axis). The analysis of ER phenotype for e.g. ATL KO cells reveals a severe fragmentation and altered connectivity in the distribution plot.

quantitative rather than descriptive evidence of ATL's role in ER tubular network formation, which was previously reported to be crucial for the fusion of ER membranes and subsequently form continuous networks [228]. With these quantitative analyses, one can compare morphological defects caused by different treatments. In another model of HSPs, depletion of protrudin resulted similarly in ER tubular network fragmentation (305 fragments) and in heterogeneous connectivity although to a lesser extent than ATL. The similar phenotypes observed in both genetic models suggest the connectivity defect in the ER may be a general cause of HSPs.

Induced cholesterol accumulation in lysosomes was also investigated. The drug U18666A is known to block the movement of cholesterol out of lysosomes. By administration of U18666A a blockage of the cholesterol transfer from lysosomes can be triggered [99]. The accumulation of cholesterol in lysosomes leads to lysosome deposition in perinuclear regions, thus affecting the ER structure and distribution [127]. ERnet revealed that the ER of U18666A-treated cells features a disassortative network (-0.34) and a low node ratio (3.4 %) suggests a highly fragmented structure, see Figures 4.24 and 4.25, which highlights that lysosomal defects can strongly affect the ER.

Finally, the performance of ERnet was tested for data of cells undergoing ER collapse due to metabolic manipulations that significantly affect the overall homeostasis inside the cell. The

sequential SIM images showed that the ER largely loses its dynamic reshaping capabilities upon the administration of the blocker SKF96365 that inhibits store-operated calcium entry [139]. On Figures 4.24 and 4.25, the corresponding ER appears fragmented and has a disassortative network. The inhibitor NaN_3 can deplete adenosine triphosphate (ATP) [136], which supports all the energy consuming processes inside the cell including ER tubule elongation, retraction and membrane fusion. Therefore, ATP depletion by NaN_3 was expected to significantly inhibit the structural dynamics of the ER. Analysis with ERnet confirms a high level of fragmentation in the ER tubular network from the lack of ATP, see Figures 4.24 and 4.25. However, the ER defects associated with this phenotype have not been found to be as severe as those caused by the depletion of ER reshaping proteins given that the node ratio of ER in ATP depleted cells is nearly 4-fold of that in ATL KO cells (0.19 vs 0.05).

In conclusion, the results of this section point to the advantages of ERnet as a tool for quantitative analysis. The method has sufficient sensitivity to detect subtle ER morphology changes while retaining the ability to work across multiple samples and acquisitions without the need of retraining, which is a capability that other segmentation methods suitable for ER segmentation do not possess, see Section 4.2.3 where the method WEKA is explored. These strengths of ERnet have enabled a detailed analysis of network connectivity facilitating the investigation of ER-related disease phenotypes.

Chapter 5

Reconstruction for SIM

The study of SIM reconstruction has been a key focus in my PhD project. In this chapter I will report on the most important results I have obtained in researching methods for SIM reconstruction. Some of the results are published and some are in review at the moment of writing. I will indicate this where relevant.

5.1 Universal reconstruction of structured illumination microscopy images

The content of this section overall follows that of my publication "ML-SIM: universal reconstruction of structured illumination microscopy images using transfer learning" [29, 30]. The presentation is adapted for this thesis and additional results are reported.

As described in Section 2.2.3, structured illumination microscopy (SIM) has become an important technique for optical super-resolution imaging because it allows a doubling of image resolution at speeds compatible with live-cell imaging. However, the reconstruction of SIM images is often slow, prone to artefacts, and requires multiple parameter adjustments to reflect different hardware or experimental conditions. Here, I introduce a versatile reconstruction method, ML-SIM, which makes use of transfer learning to obtain a parameter-free model that generalises beyond the task of reconstructing data recorded by a specific imaging system for a specific sample type. I demonstrate the generality of the model and the high-quality of the obtained reconstructions by application of ML-SIM on raw data obtained for multiple sample types acquired on distinct SIM microscopes. ML-SIM is an end-to-end deep residual neural network that is trained in an auxiliary domain consisting of simulated images but is transferable to the target task of reconstructing experimental SIM images. By generating the training data to reflect challenging imaging conditions encountered in real systems, ML-SIM becomes robust to

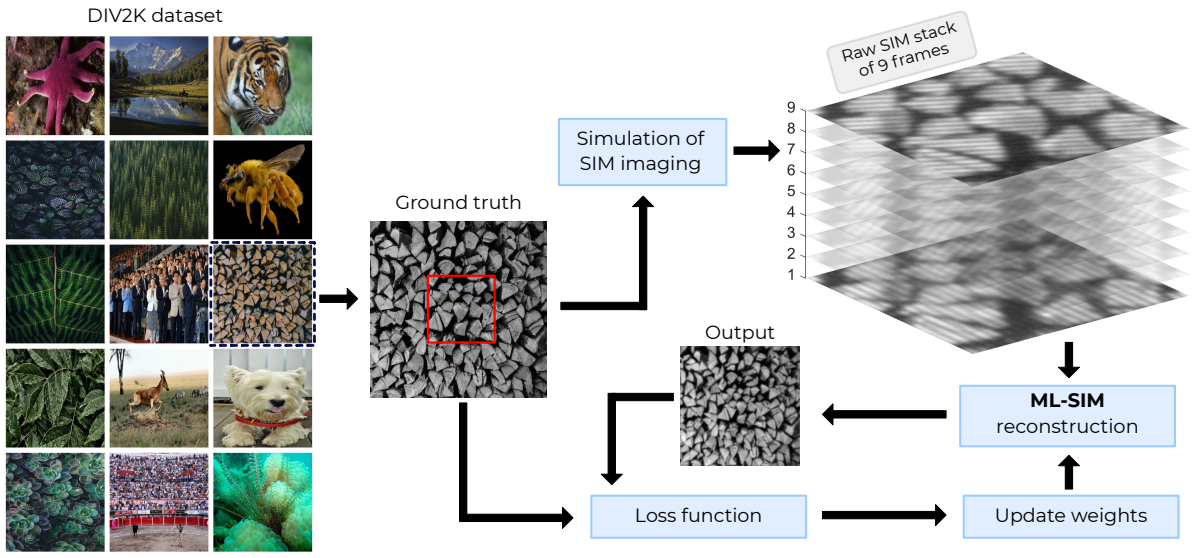


Figure 5.1: Data processing pipeline for ML-SIM. Training data for the model is generated by simulating the imaging process of SIM on high-quality photographs using a model adapted from the open-source library OpenSIM. The simulation can be further optimised to reflect the properties of the experimental system for which the reconstruction method is desired, for example to match the pixel size of the detector or numerical aperture of the detection optics. The outputs of the simulation are image stacks of the same size as those acquired by the microscope (here 9 frames).

noise and irregularities in the illumination patterns of the raw SIM input frames. Since ML-SIM does not require the acquisition of experimental training data, the method can be efficiently adapted to any specific experimental SIM implementation. I compare the reconstruction quality enabled by ML-SIM with current state-of-the-art SIM reconstruction methods and demonstrate advantages in terms of generality and robustness to noise for both simulated and experimental inputs, thus making ML-SIM a useful alternative to traditional methods for challenging imaging conditions. Additionally, reconstruction of a SIM stack is accomplished in less than 200 ms on a modern graphics processing unit, enabling future applications for real-time imaging. Source code and software for the method are available at http://ML-SIM.github.io.

5.1.1 Introduction

Structured illumination microscopy (SIM) is an optical super-resolution imaging technique that was proposed more than a decade ago [187, 72, 59, 60, 182], and continues to stand as a powerful alternative to techniques such as Single Molecule Localization Microscopy (SMLM) [144, 9] and Stimulated Emission Depletion (STED) microscopy [75]. The principle of SIM is that by illuminating a fluorescent sample with patterned illumination, interference patterns are generated that contain information about the fine details of the sample structure that are

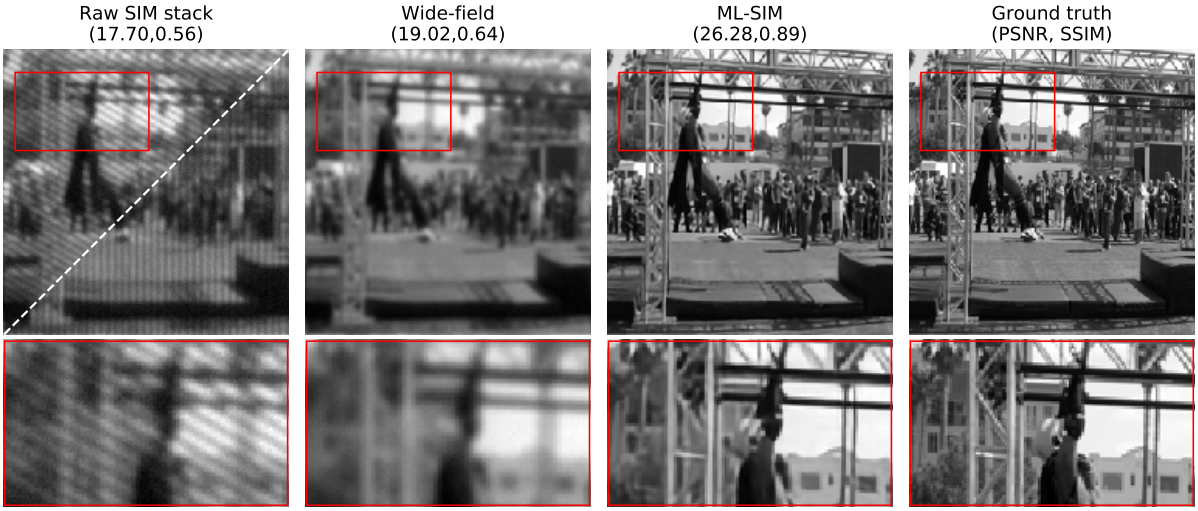


Figure 5.2: Generation of training datasets for ML-SIM. Column 1: Sample from test partition of dataset (ground truth) transformed to a raw data stack of 9 frames via simulation of the SIM imaging process. Two different orientations are shown for the excitation patterns. Column 2: Wide-field image, obtained as the mean of the 9 raw frames. Column 3: Super-resolved image obtained through reconstruction with ML-SIM. Column 4: Ground truth. The image quality metrics shown in brackets are the peak signal-to-noise ratio and the structural similarity index [205], respectively.

unobservable in diffraction-limited imaging. In the simplest case of a sinusoidal illumination pattern with a spatial frequency of k_0, the images acquired are a superposition of three copies of the sample's frequency spectrum, shifted by $+k_0$, 0, and $-k_0$. The super-resolution image can be reconstructed by isolating the three superimposed spectra and shifting them into their correct location in frequency space. The resulting spectrum is then transformed back into real space, leading to an image that is doubled in resolution. Isolating the three frequency spectra is mathematically analogous to solving three simultaneous equations. This requires the acquisition of three raw images, with the phase of the SIM patterns shifted with respect to one another along the direction of k_0. Ideally, these phase shifts are in increments of $2\pi/3$ to ensure that the averaged illumination, i.e. the sum of all patterns, yields a homogeneous illumination field. Finally, to obtain isotropic resolution enhancement in all directions, this process is repeated twice, each time with the patterns rotated by $2\pi/3$, to yield a total of 9 images (i.e. 3 phase shifts for each of the 3 pattern orientations).

While SIM can be extended to resolve features down to the 50-60 nm range [110, 169], it does not offer the highest resolution of the available super-resolution methods. However, the inherent speed of SIM makes it uniquely suited for live-cell imaging [193, 218]. SIM also requires relatively low illumination intensities, and therefore reduces phototoxicity and photobleaching compared to other methods. Many of the drawbacks of SIM relate to the reconstruction process, which can be time-consuming and prone to artefacts. In all but optimal

imaging conditions, deviations from the expected imaging model or the incorrect estimation of experimental parameters (pixel size, wavelength, optical transfer function, image filters, phase step size etc.) introduce artefacts, degrading the final image quality [5]. This becomes especially prominent for images with low signal-to-noise ratios, where traditional methods will mistakenly reconstruct noise as signal, leading to artefacts that can be hard to distinguish from real features in the sample. At worst, the reconstruction process fails completely. These issues can introduce an element of subjectivity into the reconstruction process, leading to a temptation to adjust reconstruction parameters until the 'expected' result is obtained. In addition, traditional reconstruction methods are computationally demanding. The processing time for a single reconstruction in popular implementations such as FairSIM [146], a plugin for ImageJ/Fiji, and OpenSIM running in MATLAB [103], can reach tens of seconds even on high-end machines, making real-time processing during SIM image acquisition infeasible. Finally, traditional methods cannot easily reconstruct images from SIM data that is underdetermined, e.g. inputs with fewer than 9 frames and / or recordings with uneven phase steps between frames. These drawbacks limit the applicability of SIM when imaging highly dynamic processes [194]. Examples include the peristaltic movement of the endoplasmic reticulum [78] or the process of cell division [163], which require low light level imaging at high speed to reduce the effects of phototoxicity and photobleaching.

In this work, I propose a versatile reconstruction method, ML-SIM, that addresses these issues with transfer learning. Transfer learning is a branch of machine learning that aims to exploit the knowledge obtained in an auxiliary domain to facilitate solving a specific task in the target domain [159]. While there are several methods to achieve this, a modern approach is to train a deep neural network to solve a similar task on a large dataset in the auxiliary domain, after which the network can be fine-tuned by slight changes to network architecture and retrained on a much smaller set of examples from the target domain [125]. ML-SIM uses an end-to-end deep residual neural network that is trained in an auxiliary domain consisting of simulated images using a high degree of randomisation. The training in the auxiliary domain is, in our case, sufficient for the network to generalise to a wide range of practically encountered conditions. This means that further fine-tuning of the model by training on real-world datasets, e.g. obtained from actual SIM experiments, is mostly not necessary. However, further fine-tuning and retraining is possible and supported by ML-SIM, thus offering maximal flexibility of the method to work for any experimental SIM implementation. Importantly, no output images from traditional reconstruction methods are required for training, thereby avoiding having the network undesirably learn to reproduce the reconstruction artefacts that affect traditional methods. In a recent study [86], the problem of performing SIM reconstruction with a neural network, using U-Net [173], was attempted in exactly this manner of using traditional

reconstructed outputs as targets for training, thus simply approximating the current methods and prohibiting the network from becoming superior. Furthermore, the proposed deep residual network of ML-SIM is found to be significantly more capable of SIM reconstruction than the simpler U-Net — see Section 5.1.4. The training data in the auxiliary domain is generated by a simulation of the SIM imaging process, cf. Figure 5.1. It is due to these training pairs of synthetic inputs and ideal, high-resolution targets (ground truths) for supervised learning that ML-SIM avoids exposing the model to the traditional reconstruction artefacts. Although the training data used is simulated and unrelated to real microscopic samples, I find that the method indeed generalises beyond the auxiliary domain, and I demonstrate successful application to experimental data obtained from two distinct SIM microscopes. This greatly empowers the method in the context of generalised reconstruction for super-resolution SIM imaging, since models can be customised to SIM setups of any configuration by changing simulation parameters used in the generation of the training data.

5.1.2 Methods

Convolutional neural networks

Artificial neural networks consist of a sequence of layers that each performs a simple operation, typically a weighted sum followed by a non-linear activation function, where every weight corresponds to a neuron in the layer. The weights are trainable, meaning that they are updated after every evaluation of an input performed during training. The updating scheme can be as simple as gradient descent with gradients determined via backpropagation of a loss calculated as the deviation between the network's output and a known target. A convolutional layer is no different but utilises spatial information by only applying filters to patches of neighbouring pixels. The number of learned filters in one layer is a parameter but is typically a power of 2, such as 32, 64 or 128. The network links past layers to present layers by skip connections to avoid the vanishing gradient problem. This type of architecture is known as a residual neural network [70].

Motivated by the results summarised in Figure 5.8, and with the certainty that the entire input stack is utilised for the output reconstruction, the RCAN architecture was chosen for ML-SIM. The depth of the network was chosen to be around 100 convolutional layers (10 residual groups with 3 residual blocks). The network was then trained for 200 epochs with a learning rate of 10^{-4}, which was halved after every 20 epochs, using the Adam optimiser [97]. The models were implemented with Pytorch and trained using an Nvidia Tesla K80 GPU for approximately a day per model. Models have been trained on the DIV2K dataset [2], which consists of 1000 high-resolution images of a large variety of objects, patterns and environments.

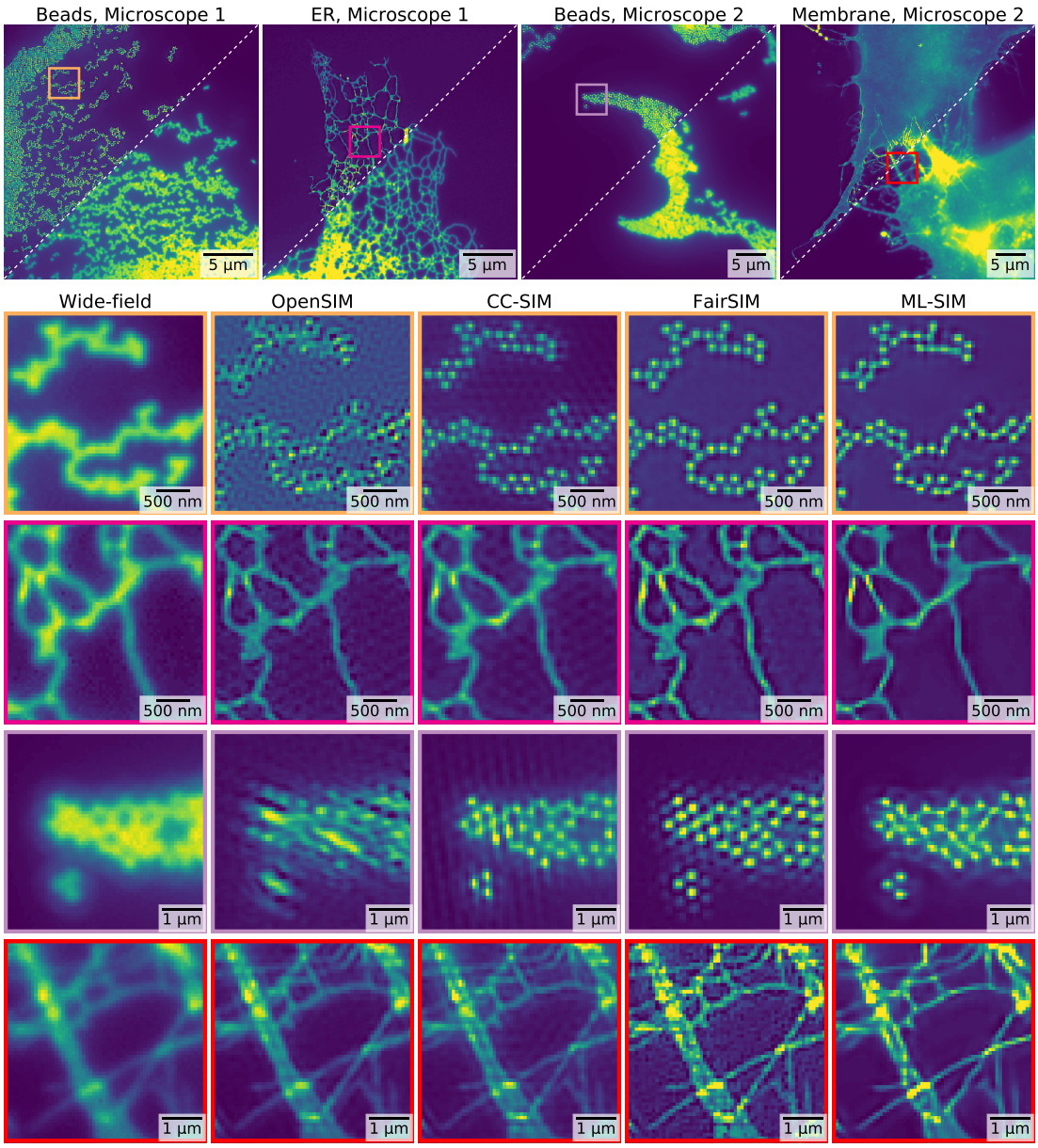


Figure 5.3: Reconstruction of SIM images from four different samples imaged on two different experimental SIM set-ups. Microscope 1 uses a spatial light modulator for stripe pattern generation [48], while microscope 2 uses interferometric pattern generation. Both instruments were used to image a sample consisting of fluorescent beads and biological samples featuring the endoplasmic reticulum (ER) and a cell membrane, respectively. (Top) Full field-of-view images where each upper left half shows the reconstruction output from ML-SIM and each lower right half shows the wide-field version taken as the mean of the raw SIM stack. (Bottom) Cropped regions of reconstruction outputs from OpenSIM [103], CC-SIM [215], FairSIM [146] and ML-SIM. Panels in rows 2 to 5 correspond to regions indicated by coloured boxes in the full-frame images.

A small set of randomly selected images were reserved for validation testing during training, but otherwise the entire dataset was used for training.

Generating simulated data

Based on the image formation model of Equation (2.6), 9 images were generated for each target image, corresponding to three phase shifts for each of three pattern orientations – see Figure B.3 for a depiction. The PSF is modelled using a Bessel function of order 1, which as described in Section 2.1.1 is the theoretical solution for light propagation through a circular aperture. The function used is given by

$$h(r,\theta) = \left(\frac{J_1(sr)}{sr}\right)^2, \tag{5.1}$$

where (r,θ) are the polar coordinates in the sample plane and s is a scaling factor used to adjust the PSF/OTF width. The value of s used is in the range of 0.6-0.9, and physically it is proportional to both the wavenumber and the numerical aperture [140].

In addition to the Gaussian noise, $N(x,y)$, in (2.6), added pixel-by-pixel, a random error is added to the parameters for the stripe patterns, k_0, θ and ϕ, to approximate the inherent uncertainty in an experimental setup for illumination pattern generation. The importance of including these types of errors is described in Section B.1.7. The use of ML-SIM with different configurations of phase shifts and pattern orientations is covered in Section B.1.6. The option of using Poisson noise rather than Gaussian noise is explored in Section B.1.5.

The images generated from Equation (2.6) are used as inputs in a supervised learning approach. The targets used to calculate loss for optimising the neural network are the clean grayscale source images used as $S(x,y)$. These targets could as well be blurred with a PSF corresponding to the best theoretically achievable resolution of standard SIM, but as explored in Section B.1.9 this is not found to be beneficial, and instead the unmodified source images are used as targets and referred to as ground truths. The values for m and k_0 that are used for data generation are given in Section B.1.7.

5.1.3 Structured illumination microscopy methodology

Microscopy

For the experimental data described in Section 5.1.4, two custom-built SIM microscopes were used. For the imaging of the endoplasmic reticulum (ER), a SIM instrument based on a phase-only spatial light modulator was used. The microscope objective used was a 60X 1.2 NA water immersion lens and fluorescence was imaged with a sCMOS camera. Cells were labelled with VAPA-GFP and excited by 488 nm laser light. For the imaging of the cell membrane, a novel SIM setup based on interferometry for the pattern generation was used [120]. In this system,

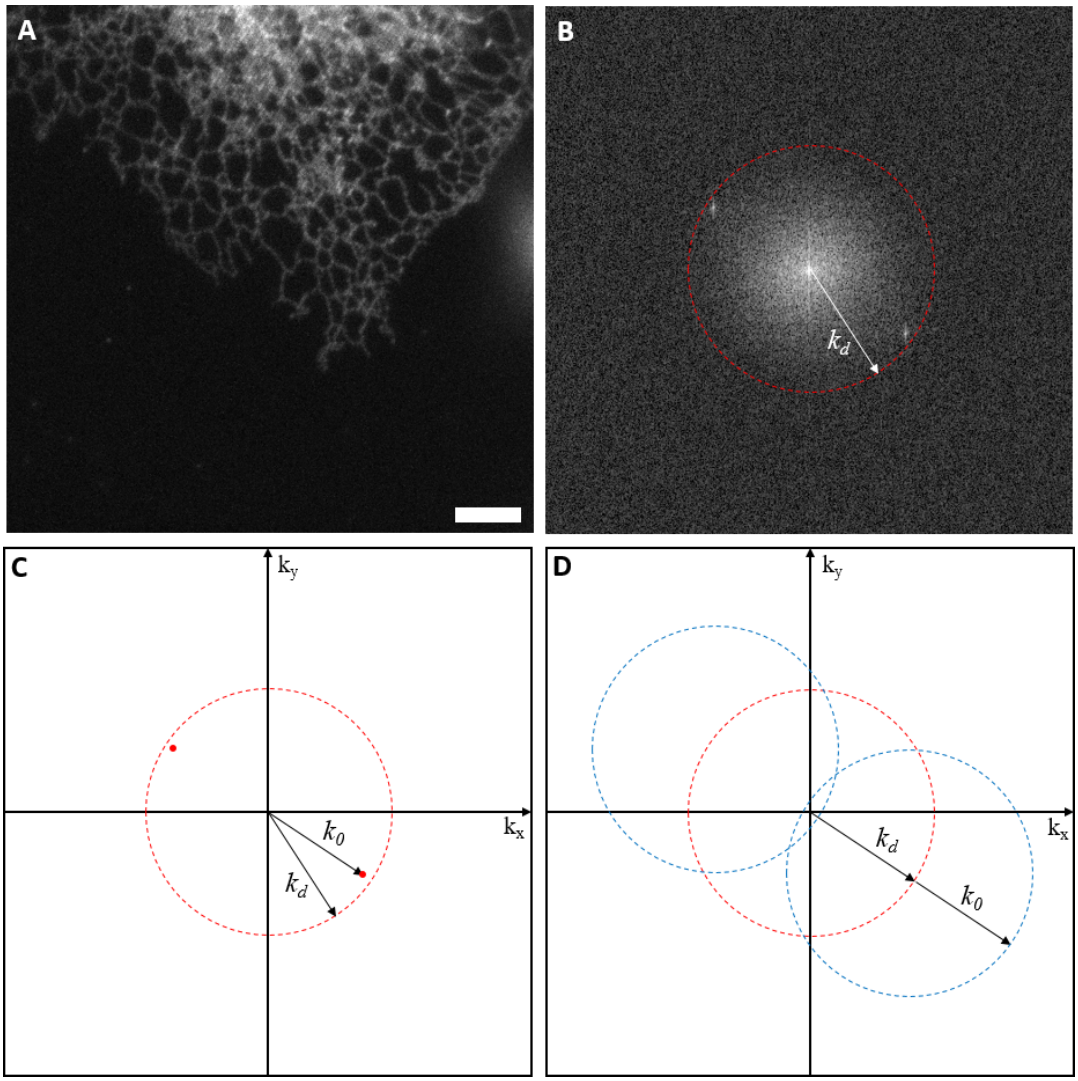


Figure 5.4: SIM methodology visualised in frequency space. (A) Raw image captured during SIM. Scale bar is 5 μm. (B) 2D Fourier transform of A. The resolution limit can be visualised as a cutoff frequency k_d beyond which no spatial frequency information from the sample is collected. The frequency components of the striped illumination pattern are visible as bright peaks close to the cutoff frequency. (C) The frequency components of the excitation pattern, k_0, are chosen to be as close to the diffraction limit as possible, to maximise resolution increase. The interference of the patterned illumination with the sample pattern means the observed region of frequency space now contains frequency components from outside the supported region, shifted by $\pm k_0$. (D) By shifting the phase of the pattern, the regions of frequency space can be isolated and moved to the correct location in frequency space. The maximum spatial frequency recovered is now $k_d + k_0$.

the angle and phase shifts are achieved by rotating a scanning mirror, the repeatability of which introduces uncertainty into the phase shifting. The microscope objective used was a 60X 1.2 NA water immersion lens, and fluorescence was imaged with an sCMOS camera. The cell membrane was stained with a CAAX-Venus label and excited with 491 nm laser light. On both

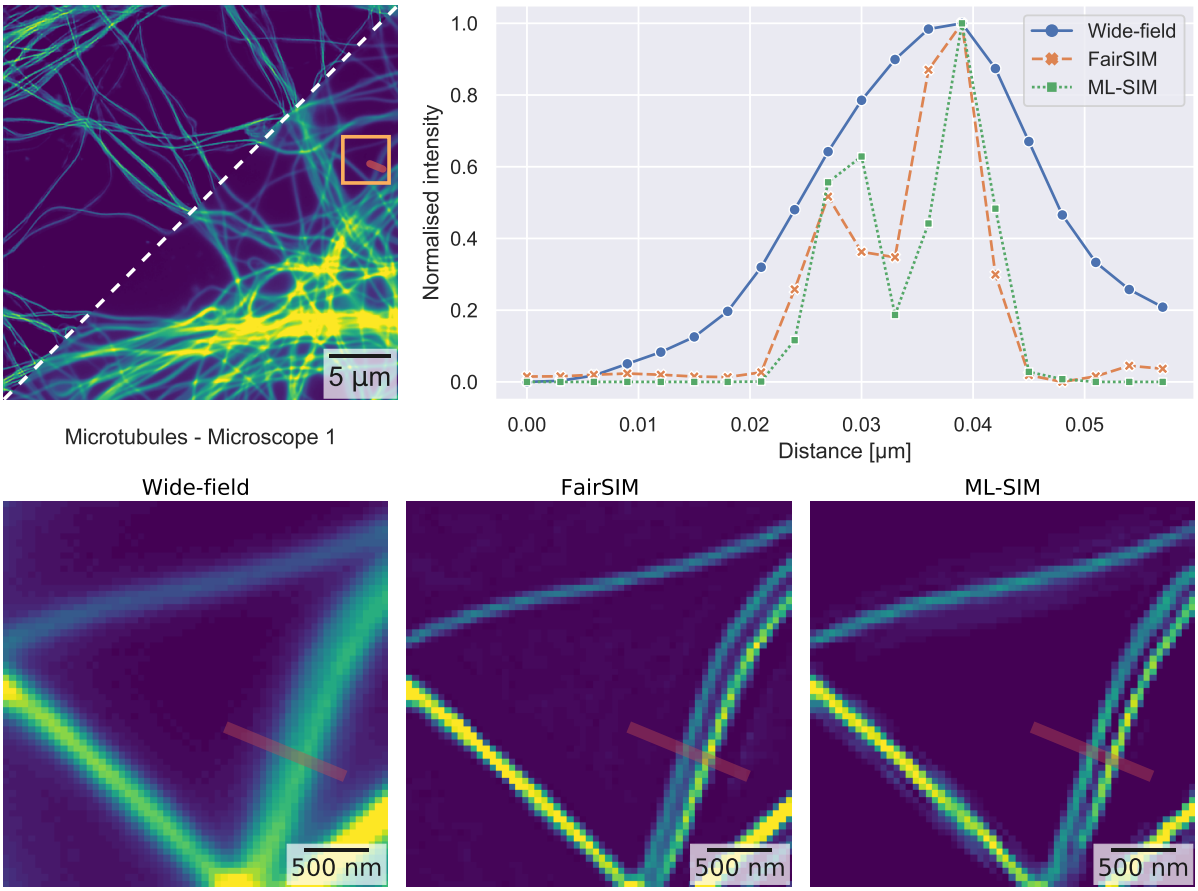


Figure 5.5: Reconstruction of a SIM image of tubulin structures. The reconstruction output of ML-SIM is compared with a wide-field projected image and FairSIM. (Top) Full field-of-view of reconstructed image and line profiles across two parallel microtubules at the position indicated by the red line. While the microtubules are not resolved in wide-field mode, both ML-SIM and FairSIM enable them to be clearly distinguished. (Bottom) Cropped regions of the reconstruction outputs corresponding to the area enclosed by the yellow rectangle.

systems, 200 nm beads labelled with Rhodamine B were excited by 561 nm laser light. For both images, the traditional reconstruction methods that have been tested managed to reconstruct the raw SIM stacks, although with varying success for the interferometric SIM setup due to the irregularity of the phase stepping.

5.1.4 Results

Using machine learning to train a reconstruction model

ML-SIM is built on an artificial neural network. Its purpose is to process a sequence of raw SIM frames (i.e. a stack of nine images representing the number of raw images acquired in a SIM experiment), into a single super-resolved image. To achieve this, a supervised learning

approach is used to train the network, where pairs of inputs and desired outputs (ground truths) are presented during training to learn a mapping function. These training data could be acquired on a real SIM system using a diverse collection of samples and experimental conditions. However, the targets corresponding to the required inputs are more difficult to obtain. At least two approaches seem possible: (A) using outputs from traditional reconstruction methods as targets [86]; and (B) using images from other super-resolution microscopy techniques that are able to achieve higher resolution than SIM (e.g. SMLM or STED). Option (A) would prohibit ML-SIM from producing reconstructions that surpass the quality of traditional methods and would be prone to reproducing the artefacts mentioned in Section 5.1.1. Option (B) requires a capability to perform correlative imaging of the same sample, which may be difficult to achieve since training requires hundreds or even thousands of distinct data pairs [67]. In addition, both approaches require the preparation of many unique samples to build a training set diverse enough for the model to generalise well. Hence, these options were not pursued in this work, and instead I approached the problem by starting with ground truth images, and simulating inputs by mimicking the SIM process in silico, allowing for very diverse training sets to be built. I used the image set DIV2K [2], which consists of 1000 high-resolution images of a large variety of objects, patterns and environments. To generate the SIM data, images from the image set were resized to a standard resolution of 512×512 pixels and transformed to grayscale. Raw SIM images were then calculated using a SIM model adapted from the OpenSIM package [103]. The model and underlying parameters are described in Section 5.1.2. The simulated raw SIM stacks were used as input to the neural network and the output compared to the known ground truth in order to calculate a loss to update the network weights. Figure 5.1 shows an overview of the training process with an example of a simulated SIM input. The architecture of the neural network is further described in Section B.1.3.

Application of the trained model

For a start, I tested that the network had learned to reconstruct simulated SIM stacks. Prior to training, a separate partition of DIV2K was selected for testing. A sample from this test partition is shown in Figure 5.2. The stripe pattern for two of the nine frames of the input SIM stack is shown in the leftmost panel. The stripe patterns cancel out when all 9 frames are summed together (second column), and this corresponds to the case of even illumination in a wide-field microscope. Compared to the wide-field image, the reconstruction from ML-SIM is seen to have a much improved resolution with a peak signal-to-noise ratio (PSNR) value more than 7 dB higher, as well as a significantly higher structural similarity index (SSIM). Beyond these metrics, several features of the image can be seen to be resolved after reconstruction that

were not visible beforehand, such as the vertical posts seen to the right side of the cropped region.

It should also be noted that the reconstruction has not amplified the noise by introducing any evident artefacts, even though the input image featured a significant amount of Gaussian noise in addition to randomisation of the stripe frequency and phase – see Section 5.1.2 for definitions of those parameters. As further described in Section 5.1.2, the neural network underlying ML-SIM is different to those of generative networks, which means that the model is more strongly penalised during training for introducing image content that is not in the real image. I argue that, even though this results in slightly more blurred output images than would be achievable with a generative network[105, 204], the absence of artificial features is preferable in scientific imaging applications. This trade-off is referred to as minimising the mathematical reconstruction error (e.g. root-mean-square deviation) rather than optimising the perceptual quality [12, 197].

While ML-SIM is able to reconstruct simulated SIM stack inputs, it is of course only valuable if it also works on real SIM data, acquired experimentally. The ML-SIM model was trained on input data from simulations, using data bearing little resemblance to real-world biological SIM data. Any success for real-world SIM reconstructions, therefore, requires the model to have generalised the SIM process in such a way that it becomes independent of image content and sample type. This requires a realistic simulation of the SIM imaging process to generate training data that is sufficiently diverse, and reflects measurement imperfections as encountered in practical SIM imaging. The former was avoided through the use of a diverse training dataset, and the latter through the use of the well-known imaging response function (Section 5.1.2, Equation (2.6)), and introduction of uncertainty in the stripe patterns. To test ML-SIM on experimental data, SIM images of different samples were acquired with two different SIM setups [221]. The resulting reconstructed outputs are shown in Figure 5.3, in which they are compared to outputs of traditional reconstruction methods: OpenSIM [103], a cross-correlation (CC-SIM) phase retrieval approach [215, 24], and FairSIM [146]. The images are grayscale images of signal intensity mapped to the Viridis colour table. ML-SIM is seen to obtain resolution on par with the other methods but produces less noisy background and fewer artefacts. The bottom two rows of images of beads and cell membranes were acquired with phase steps deviating from the ideal $2\pi/3$. This reflects a difficulty with the interferometric SIM setup (see Section 5.1.2) to achieve equidistant, and precisely defined, phase steps for each illumination pattern angle. This means that the reconstruction algorithm must handle inconsistent phase changes, a factor only the cross-correlation method was capable of handling. However, although CC-SIM has improved resolution, artefacts are apparent, seen as vertical

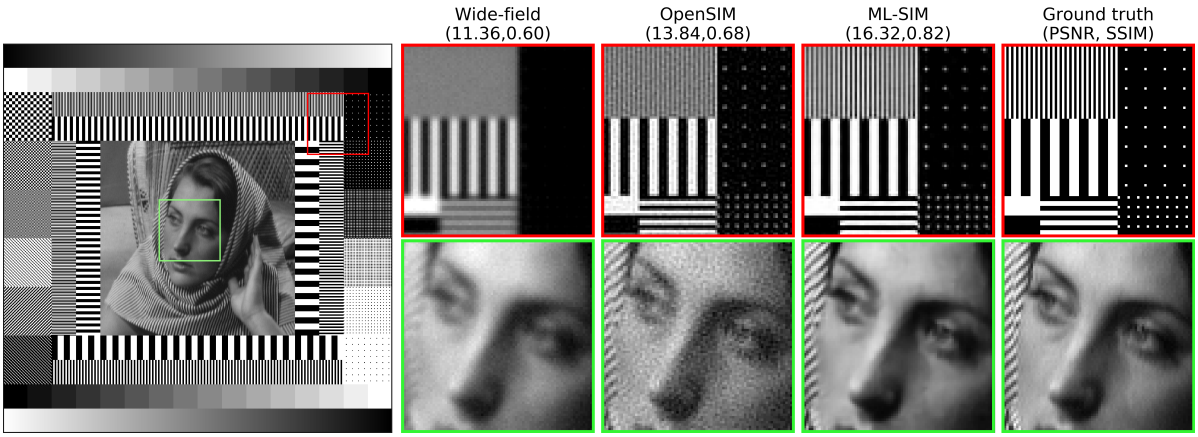


Figure 5.6: Reconstructions of a test target with OpenSIM and ML-SIM and comparison to the ground truth. OpenSIM was found to be the best performing traditional method on this test sample, both in terms of PSNR and SSIM with the other methods achieving PSNR scores of 12.56 dB (CC-SIM) and 12.88 dB (FairSIM).

lines and ringing in the images. ML-SIM, on the other hand, reconstructed with fewer artefacts and strongly improved background rejection.

To further demonstrate the super-resolution performance of ML-SIM, a sample of 30 nm microtubules labelled with Alexa-647 was imaged on microscope 1. The reconstruction outputs and line profiles across neighbouring microtubules for both ML-SIM and FairSIM are shown on Figure 5.5. The displayed cropped region contains two parallel microtubules which are separated by a gap of size below the diffraction limit and thus not resolved in the wide-field image. In the outputs from ML-SIM and FairSIM, the gap is clearly visible. The distance between the peaks in the line profiles for ML-SIM and FairSIM is $\simeq$ 150 nm, which is close to the theoretically achievable resolution with standard SIM [59]. Analysis of the resulting OTFs after reconstruction is also provided in Section B.1.8.

The application of ML-SIM to TIRF-SIM image data using a sample image from the official FairSIM test image repository is described in Section B.1.10.

Performance assessment

I performed a quantitative comparison of ML-SIM with traditional reconstruction methods on reconstructions of simulated raw SIM stacks generated from two image datasets; a subset of 10 DIV2K images, unseen during training, and 24 images from a dataset referred to as Kodak 24, commonly used for image restoration benchmarking [2, 108]. Parameters for OpenSIM, CC-SIM and FairSIM were all systematically adjusted to produce the highest achievable output quality possible. Consequently, each method required completely different parameter configurations than those used for reconstructions of the experimental data shown in Figure 5.3.

For ML-SIM however, there were no tunable parameters. The optical transfer function (OTF) is estimated within each method even though the function is known for the simulated images – this is the same premise as for the reconstruction of the experimental samples in Figure 5.3, for which the OTFs were unknown. Each method applies an identical Wiener filter to the final reconstruction output, whereas the output of ML-SIM is untouched. The performance scores of all methods averaged over the entire image sets are listed in Table B.1 with scores for wide-field as a reference in terms of peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). For both metrics, ML-SIM has the highest scores with a PSNR that is ~2 dB higher than that of OpenSIM. CC-SIM and FairSIM lag behind, but both methods still succeed in improving the input beyond the baseline wide-field reference. The performance gap between OpenSIM and the other traditional methods is likely due to a better estimation of the OTF, because OpenSIM assumes an OTF that is similar to the one used when simulating the SIM data.

A more challenging test image than those based on DIV2K and Kodak 24 images is shown in Figure 5.6. This simulated test image is reconstructed with the same three traditional methods. OpenSIM is found to achieve the best reconstruction quality of the three with a PSNR score of 13.84 dB versus 12.56 dB for CC-SIM and 12.88 dB for FairSIM. The same image reconstructed with ML-SIM results in a PSNR score of 16.32 dB – again about 2 dB higher than that of OpenSIM. Two cropped regions comparing OpenSIM and ML-SIM are shown in Figure 5.6. The area in the upper right corner of the test image is particularly challenging to recover due to the single-pixel point patterns and the densely spaced vertical lines. While the points vanish in the wide-field image, these are recovered both by OpenSIM and ML-SIM. The resolution of the point sources are slightly superior in the ML-SIM reconstruction, and ML-SIM manages to recover the high-frequency information in the top line pattern very well. Overall it is also seen that the reconstruction from ML-SIM contains much less noise, which is especially evident in the zoomed region of the face. This suggests that ML-SIM is less prone to amplify noise present in the input image. I tested this further by gradually adding more Gaussian image noise to the input image, and again comparing the reconstructions from the various methods. The results of this test are shown in Figure 5.7, where it is clearly seen that ML-SIM performs best at high noise levels. As more noise is added the gap in performance is seen to increase between ML-SIM and the other models indicating that the neural network has learned to perform denoising as part of the reconstruction process. This is supported by the cropped regions on the right side of the figure, which show a higher level of detail in the image when compared to the input, wide-field and OpenSIM images. OpenSIM was found to perform consistently well in this noise test, whereas FairSIM and CC-SIM struggled to reconstruct at all for higher noise levels. This is not surprising, since added noise may cause the parameter

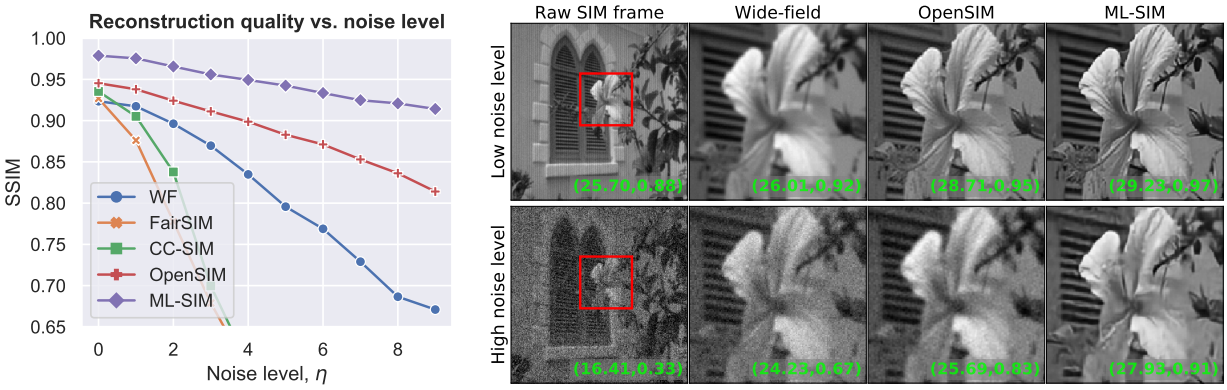


Figure 5.7: (Left) Reconstruction quality as measured by the structural similarity index, SSIM, as a function of the amount of noise added to an input image. Gaussian noise is added to every frame of the raw SIM stack. Noise is normally distributed with a standard deviation $\eta \cdot \sigma$, where σ is the standard deviation of the input image. (Right) Images at low ($\eta = 0$) and high noise levels ($\eta = 9$) reconstructed with OpenSIM and ML-SIM, respectively. PSNR and SSIM scores using the ground truth as reference are shown in the lower right-hand corner of every image.

estimation to converge to incorrect optima, which can heavily corrupt the reconstruction outputs. As a result, the reconstruction outputs from FairSIM and CC-SIM were of poorer quality than the wide-field reference at higher noise levels.

Several architectures were tested as part of this research to select the one most suitable for ML-SIM. U-Net [173] is a popular, versatile and easily trained network, but its performance was found to fall short of state-of-the-art single image super-resolution networks such as EDSR [114] and RCAN [227]. These super-resolution networks have been customised to be able to handle input stacks of up to 9 frames and output a single frame with no upsampling, i.e. the upsampling modules of those networks have been omitted – see Figure B.2 for a depiction. In addition to testing different network architectures the number of frames of the input raw SIM stack, up to a total of 9, was also varied. In the left-hand side of Figure 5.8 the convergence of test scores on a validation set during training are shown for the various architectures and input configurations considered. It is found that SIM reconstruction with subsets containing only 3 or 6 frames still performed significantly better than if the network learns to perform a simpler deconvolution operation by just training on a wide-field input. This confirms that the network learns to extract information from all 9 frames in the full stack versus a subset of it or the mean of its frames. Only using a subset of 3 frames does however cause a substantial reconstruction quality loss compared to using 6 frames, which is not surprising since the corresponding analytical reconstruction problem becomes underdetermined for fewer than 4 frames [194]. The RCAN model performed better than EDSR with a consistently higher PSNR score when trained on all 9 frames, while performing similarly to EDSR when trained with 3 fewer frames. Based on these results RCAN was chosen as the default architecture for ML-SIM. The fact that

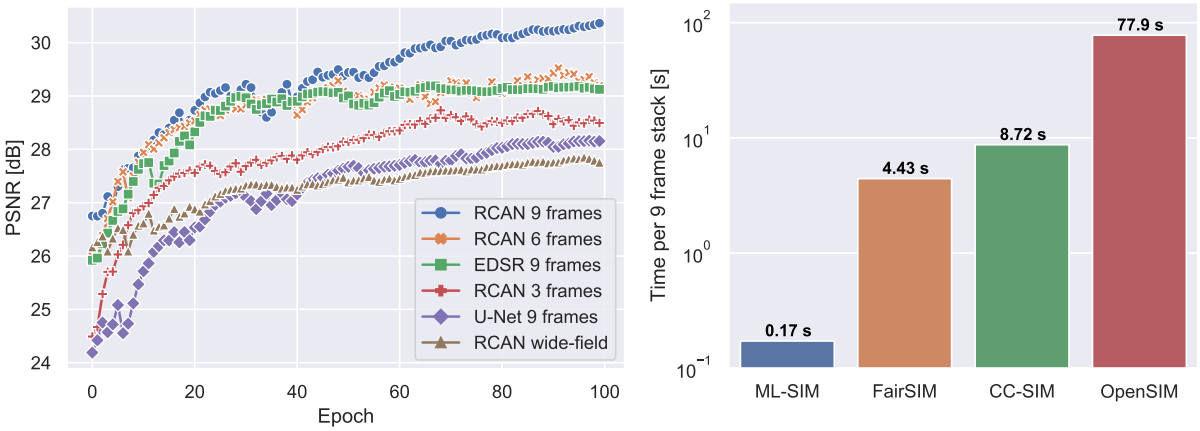


Figure 5.8: (Left) Validation test set scores during training for different network architectures and input dimensions. The two state-of-the-art single image super-resolution architectures, RCAN and EDSR, have been modified to perform SIM reconstruction. The number of frames of the raw SIM stack, up to a total of 9, is also varied to confirm that the network learns to extract information from all 9 frames in the full stack. (Right) Computation time for reconstruction of a single raw SIM stack of 9 frames. The shown run times are averages of 24 consecutive reconstructions with sample standard deviations of 0.0034, 0.13, 0.51 and 3.7 seconds for ML-SIM, FairSIM, CC-SIM and OpenSIM, respectively.

reconstruction with fewer than 9 frames is possible could be exploited for compressed, faster SIM imaging as done in [86] using a U-Net model, although this inevitably comes at a loss of quality.

Regarding the computation time for each reconstruction method, I measured the average time for reconstructing the raw SIM stacks based on the Kodak 24 image dataset one by one. The timing for each method is then the mean of 24 time samples with an associated standard deviation. The timings are shown on the right-hand side of Figure 5.8. Computations were performed on a computer running Windows 10 with an Intel i5 6500 CPU, 16 GB DDR4 RAM and an Nvidia GTX 1080 Ti GPU. ML-SIM finishes a reconstruction in less than 200 ms, which is more than an order of magnitude faster than the other methods. Substantial speed-ups are to be expected when using neural networks due to the computations being greatly parallelisable, thus making it easy to use GPU acceleration – this was similarly found in [150], where a neural network was used for reconstruction of stochastic optical reconstruction microscopy images. The traditional methods for SIM reconstruction are more difficult to parallelise, partly because the numerical optimisation algorithms needed for parameter estimation tend to be iterative and sequential. This therefore provides a computational advantage of ML-SIM. At ~170 ms per reconstructed image from a SIM stack of 9 frames, the reconstructed image rate is about 6 per second, corresponding to an imaging system that captures 54 frames per second, which could provide fluent, real-time, super-resolution feedback to the user during image acquisition.

Web app, desktop app and source code

The source code for training ML-SIM and applying the model for reconstruction is available in a public repository on GitHub, https://github.com/charlesnchr/ML-SIM, and figshare [30]. This repository includes source code for generating the training data by simulating the SIM imaging process with parameters that can be easily adapted to reflect specific SIM setups (e.g. by changing stripe orientations, number of frames, etc.). The repository also holds source code for a desktop program with pre-built installers for Windows, macOS and Linux. The program makes it easy to use ML-SIM and perform batch processing via a graphical user interface. During installation required dependencies such as Python, Pytorch and pre-trained ML-SIM models are automatically fetched. If the pre-trained models perform suboptimally, it is easy to train a new model that is more specific to a given SIM setup and set the program to use this custom model for reconstruction. The program includes a plugin for μManager [44] that enables a real-time live-view of ML-SIM reconstructed output during acquisition in many imaging systems thanks to the wide support of camera drivers in μManager. See Section B.1.1 for more details. Furthermore, I have created a web app accessible via http://ML-SIM.github.io with a browser-based online implementation of ML-SIM that is ready for quick testing using a pre-trained model and does not require installation of any software.

Fully synthetic data generation

The ML-SIM approach that has been described in the preceding sections rely on a diverse underlying image dataset for synthesising rich training data using the SIM image formation model Equation (2.6). However, besides rationalising the requirement of using a diverse dataset with the need for the model to generalise, no empirical evidence has been provided to show why fully synthesised image data may be problematic. Fully synthesised image data in this context refers to procedurally generated images produced by an algorithm rather than using a set of photographs, e.g. the DIV2K dataset, as input to the image formation model. In this section, I will briefly consider how fully synthetic image data qualitatively can give rise to poorer reconstruction quality for a trained model, and an example of how fully synthetic image data can be beneficial is also given.

To simulate natural images, I use the scale-invariant “dead leaves"" model proposed in [106]. The model is based on a set of disks with random radii from a $1/r^3$ distribution that are randomly added to an image, overlapping and occluding other disks. Naturally occurring objects in photographs follow such a cubic power law, which is supported by findings in [106] where empirical statistics of natural images are reported to have an excellent agreement with

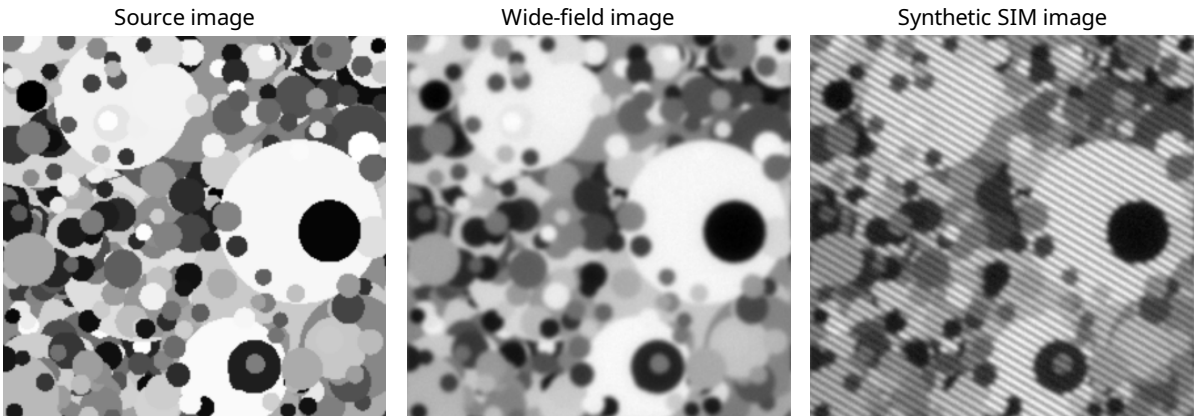


Figure 5.9: Fully synthetic image generated by following the dead leaves model [106]. (Left) The original dead leaves image. (Centre) Wide-field projection of image stack when using the SIM image formation model Equation (2.6). (Right) A SIM frame from the image stack produced by the image formation model.

dead leaves images. The colour of a disk is assigned according to a uniform random distribution. An example of an image based on the dead leaves model can be seen in Figure 5.9.

One advantage of a fully synthetic image generation model is that an unlimited number of training pairs can be produced. This makes it easy to build a large training dataset, which enables the training of a neural network by following the same approach as for the ML-SIM models in the preceding sections. With a dataset of 10,000 training images and a network with the RCAN architecture, a model is obtained that is highly capable of reconstructing dead leaves images. However, when attempting to reconstruct one of the more complex SIM images based on the DIV2K dataset, it becomes clear that the simple shapes of the disks have caused a bias in how the reconstruction recovers structural information. An example of this is shown on Figure 5.10, where the reconstruction output appears to have a spot-like texture in the regions with more detail. This result may not be a surprise given that the model has only ever seen disks in the training data, although of varying size, and therefore will have a propensity towards introducing shapes resembling disks. Qualitatively, this indicates that the lack of detailed, high-frequency information in the training data as well as diverse geometries can cause the model to not generalise and not perform very well on test data.

Although more realistic and complex models for image generation could potentially be made, it is likely a steep task in itself to design an algorithm that can approximate the richness of natural images. Recently, generative models based on a mechanism called stable diffusion, such as DALL-E 2 [168], have proven able to output remarkably diverse and complex images. It is possible that such networks could provide purely generated training data that would be

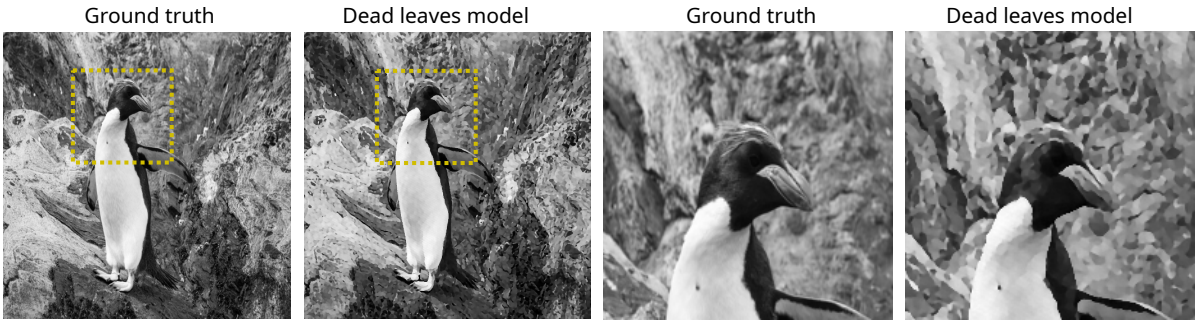


Figure 5.10: Reconstruction of a SIM image based on the DIV2K dataset, which has image content that is significantly more complex than the dead leaves images. The original image is shown on the left with the reconstruction output next to it. On the right half of the figure are two cropped regions that highlight differences in the high-frequency regions of both foreground and background.

useful for training a SIM reconstruction model, but before such models become more common, using natural images as input to a SIM image formation model is probably the best choice.

However, fully synthetic images could have other uses such as testing or tweaking an already trained ML-SIM model. With a particular spatial distribution and disks of smaller, fixed radius, one could attempt to approximate images of beads. Some examples are shown on Figure 5.11 with the layout of beads in two different spatial distributions. By generating a set of such images and retraining already trained models with relatively few updates, I have qualitatively observed that ML-SIM models can be fine-tuned towards reconstructing images of beads with slightly higher fidelity. Spot-like shapes resembling blobs of diffracted point sources are typical for fluorescence microscopy images but probably less so in natural images of macroscopic objects such as in the DIV2K image dataset. Therefore, there could be a benefit to having a subset of a training dataset consist of fully synthetic images with point sources. This is something that could be explored more thoroughly, but based on a few attempts, I have not found the slight reconstruction improvements to point-like shapes worthwhile and in the interest of maximising the versatility of ML-SIM, I have favoured diverse image datasets without any proportion of fully synthesised imagery.

5.1.5 Discussion

I have demonstrated and validated a SIM reconstruction method, ML-SIM, which takes advantage of transfer learning by training a model in an auxiliary domain consisting of simulated images and generalises to the target task of reconstructing experimental SIM images with no fine-tuning or retraining necessary. The training data was generated by simulating raw SIM image data from images obtained from common image repositories which served as ground truths.

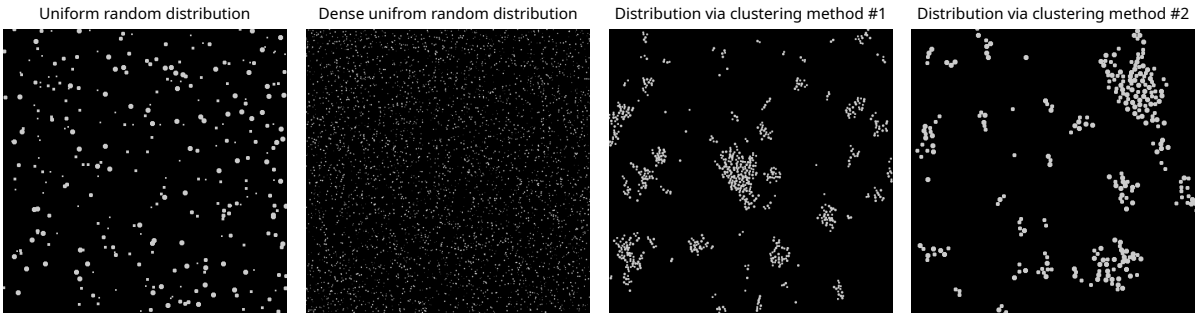


Figure 5.11: Synthetic images of beads with certain densities and spatial distributions could be useful in testing and possibly fine-tuning ML-SIM models. Two examples on the left show beads with a spatial distribution according to uniform random distribution. On the right are two examples where beads have a higher probability to co-locate creating clusters that mimic e.g. the experimentally acquired images of beads on Figure 5.3.

ML-SIM successfully reconstructed artificial test targets that were of a completely different nature than the diverse images used to generate the training datasets. More importantly, it successfully reconstructed real data obtained by two distinct experimental SIM implementations. I compared the performance of ML-SIM to widely used reconstruction methods: OpenSIM [103], FairSIM [146], and CC-SIM [215]. In all cases, reconstruction outputs from ML-SIM contained less noise and fewer artefacts, while achieving similar resolution improvements. Through randomisation of phase shifts in the simulated training data, it was also possible to successfully reconstruct images that could not be processed successfully with two of the traditional reconstruction methods. ML-SIM shows robustness to inconsistent variations in the SIM imaging parameters and deviations from equidistant phase shifts. Similarly, ML-SIM reconstructed images that were strongly degraded by noise even when other methods failed.

A central advantage of the transfer learning approach of the ML-SIM method is that the simulated data that constitute the auxiliary domain can be made arbitrarily diverse by randomisation of optical parameters, enabling the model to become highly generalised for the target task. Furthermore, the simulation can also be optimised to a specific system by changing relevant optical parameters. In principle, this makes the method applicable to any SIM setup irrespective of its configuration. For instance, a SIM setup with another illumination pattern configuration, e.g. 5 orientations and 5 phase shifts (5×5 stacks), is trivial to support with ML-SIM by changing just two parameters in the pipeline. General, pre-trained models for SIM microscopes with configurations for 3×3, 3×5 and 5×5 stacks are provided at http://ML-SIM.github.io along with source code and software to use them.

A future direction could be to fine-tune the training data by incorporating a more sophisticated image formation model. This image formation model might also take certain optical

aberrations into account. Currently, out-of-focus light from above and below the focal plane is not simulated in the training data. As with the other reconstruction methods, this can result in artefacts in regions of the sample with dense out-of-focus structures. Given that the spatial frequency information required to remove this background is available in SIM, it is possible that an updated ML-SIM network could be constructed to incorporate an efficient means for background rejection [167, 151].

5.2 Speckle SIM

So far in this chapter, I have treated SIM as a technique that relies on illumination with fringe illumination, i.e. a sinusoidal illumination pattern. However, SIM does not necessarily need this to work as pointed out in Section 2.2.4. There is previous work on using spot illumination [206] and speckle patterns [145, 3]. In this section, I will consider the most generic case of illumination, where nothing is controlled and the pattern is purely random, i.e. speckle pattern. This calls for a different approach to reconstructing data because the analytical Fourier formalism is of little use when the support in frequency space is similarly poorly defined. This generic implementation of SIM is referred to in the literature as *speckle SIM* [140] and reconstruction requires solving an optimisation problem that is less well-posed than for regular SIM. I have implemented the Blind-SIM reconstruction algorithm of [145], which is the original paper that proposed a method called Blind-SIM. Furthermore, I have also studied the problem of blind SIM in the context of ML-SIM as well, and I have trained models with training data that has been synthesised with a speckle pattern image formation model. In this section, I will assess the potential for an ML-SIM based method to solve this more difficult reconstruction problem by comparing it to the regular approach in the literature.

5.2.1 Speckle illumination patterns

Speckle patterns arise from interference of coherent light that has been put out of phase due to reflection from a surface or scattering from propagation through a diffuser. The knowledge of the microscopic structure of the surface from which light is reflected, or the composition of the diffuser, cannot be known. Therefore, it is necessary to address speckle patterns in statistical terms. Speckle patterns are often modelled with a random walk approach. This is because the speckle effect results from the interference of many waves with the same frequency but with different phases and amplitudes. The waves interfere both constructively and destructively, thus forming a resultant wave whose intensity varies randomly. If we model each wave as a vector, then it follows that the magnitude of the resulting vector can be anything from zero to the sum of magnitudes of all the individual vectors. Hence, the final speckle can be viewed as a product of a 2-dimensional random walk. The statistics and physics of the random interferences that cause speckle patterns have long been understood [55]. For polarised light, the probability density function of the intensities across a speckle pattern is given by

$$P(I) = \frac{1}{\langle I \rangle} \exp\left(\frac{-I}{\langle I \rangle}\right), \tag{5.2}$$

where $\langle I \rangle$ is the average intensity.

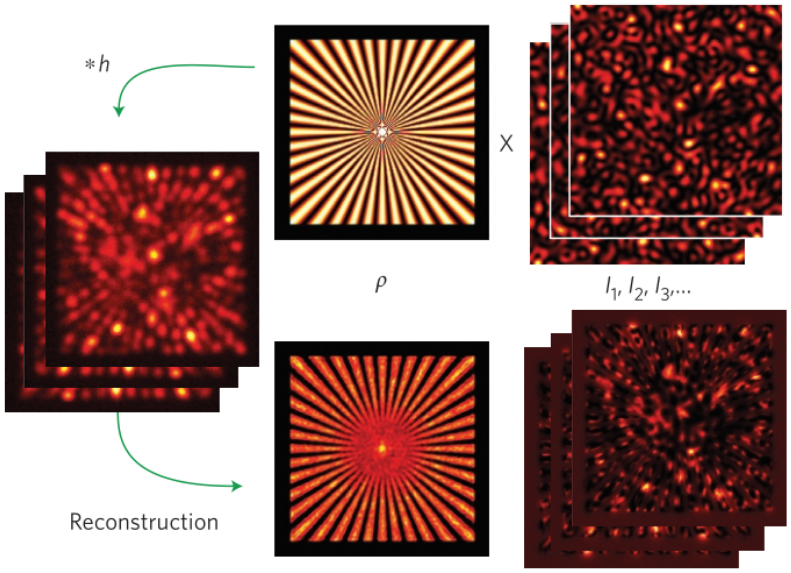


Figure 5.12: Reconstruction scheme with the Blind-SIM algorithm. Given a set of speckle patterns that illuminate a sample, the algorithm performs the joint optimisation problem of determining both the illumination patterns and the sample fluorescence density.

Due to its complexity, or random nature, one may think it is difficult to estimate the intensity distribution given a measurement of a sample illuminated by a speckle pattern. However, this is the very approach taken by Mudry *et al*. [145], in which a stack comprising N acquisitions with different speckle illumination patterns is used to estimate the fluorescence signal in addition to the N speckle pattern. This approach is depicted on Figure 5.12.

5.2.2 A numerical optimisation method for speckle SIM

As for fringe SIM, Equation (2.6), the image formation model is assumed to be given by the convolution of the PSF h and the fluorescence density, $I\rho$, where I is the spatially varying illumination pattern and ρ is the fluorophore density, i.e.

$$M = (I\rho) \otimes h, \tag{5.3}$$

where $\otimes$ is the convolution operator. For the blind SIM problem, we consider a sample that is successively illuminated by L distinct illumination patterns $I_{l=1,\dots,L}$. It is assumed that there is no motion in the time window of the acquisition of these L images, and therefore the fluorophore

density can be treated as constant. The individual images can thus be written

$$M_{l=1,..,L} = (I_l \rho) \otimes h. \tag{5.4}$$

Based on these L images, the goal is to reconstruct both the fluorophore density and the L incident light patterns. Collectively, this corresponds to solving for $L+1$ unknowns given L equations, which means the problem is underdetermined and thus ill-posed. To turn the reconstruction problem into a well-posed optimisation problem, the constraint that the sum of all the incident intensities must be homogeneous over the sample plane is introduced. In other words, this means that the sample is uniformly illuminated when accumulating all of the L light patterns. The constraint can be written as

$$\sum_{l=1}^{L} I_l \approx L I_0, \tag{5.5}$$

where I_0 is the baseline intensity that is constant over the sample plane. Using this constraint the number of unknowns can be reduced by expressing the last illumination pattern as a function of the other patterns

$$I_L = L I_0 - \sum_{l=1}^{L-1} I_l. \tag{5.6}$$

As a result the $L-$th equation from Equation (5.4) can be rewritten as

$$M_L = \left[\left(L I_0 - \sum_{l=1}^{L-1} I_l \right) \rho \right] \otimes h, \tag{5.7}$$

which is now fully determined from the L other unknowns. The fluorophore density in addition to the $L-1$ first light patterns are jointly estimated using an iterative numerical optimisation method that minimises the objective function

$$F(\rho, I_{l=1,\ldots,L-1}) = \sum_{l=1}^{L-1} \|M_l - (I_l \rho) \otimes h\|^2 + \|M_L - \left[\left(L I_0 - \sum_{L-1}^{l=1} I_l \right) \rho \right] \otimes h\|^2, \tag{5.8}$$

where $\|.\|$ is the elementwise, pixel-by-pixel, Euclidean norm.

To solve Equation (5.8) using an iterative optimisation algorithm, the gradients of the objective function are required. Mudry *et al*. derive the gradients for Equation (5.8) in Appendix 3 of [145], which enables a method like gradient descent, variants of which are also used for backpropagation Equation (2.20).

Mudry *et al*. further show in the Supplementary Information of [145] that the system of equations representing Equation (5.8) can be rewritten by making use of the fact that the fluorophore density ρ and intensities I_l are real and positive, which can be imposed by substituting the auxiliary variables $I_l = i_l^2$ and $\rho = \xi^2$. This enables the use of the conjugate gradient method as this numerical method requires a system of equations whose matrix is positive-definite [165]. The conjugate gradient method increases the stability of the numerical optimisation, especially in the presence of noise.

Implementation of the Blind-SIM algorithm. I have attempted to obtain the data and source code used in the original Blind-SIM publication [145]. The code is not open-sourced and I have not managed to get in touch with the authors. Instead, I have made my own implementation of the Blind-SIM algorithm. The gradients used in the implementation are found in the supplementary information of [145]. In the implementation, I have tested both a gradient descent method and the author's suggested conjugate gradient method as the numerical iterative solver backbone. I have found significant improvements in performance and stability using the conjugate gradient method, which has therefore been used to obtain the results in this section.

The code for the implementation has been written in Python, which admittedly is not the best choice for a numerical method or scientific computing as a whole, but performance has not been essential for the experiments of this section. The implementation is validated using a generated test target described in Section 5.2.6 to provide a ground truth. The error between the iterative output and the ground truth is observed to converge towards zero as a function of iteration. This is shown in the convergence plot on Figure 5.13. The error decreases multiple orders of magnitude, indicating that the numerical scheme works as intended. The run time for the total of 300 iterations shown on the plot was 50 minutes on a mid-tier workstation with an Intel 10th generation CPU. Despite not being an optimised implementation, this performance does not differ significantly from [145] as the Supplementary Information reports that each iteration took between 2.5s to 95s depending on the number of speckle patterns used and the number of pixels in each stacked frame.

5.2.3 Using ML-SIM to solve the blind SIM problem

ML-SIM was originally developed to be used for fringe SIM as described in Section 5.1, but apart from differences in dimensionality, the deep neural network is not explicitly configured to process fringe illuminated input stacks. Instead, that customisation is purely reflected in the training data, and thus a different training dataset based on another type of illumination pattern

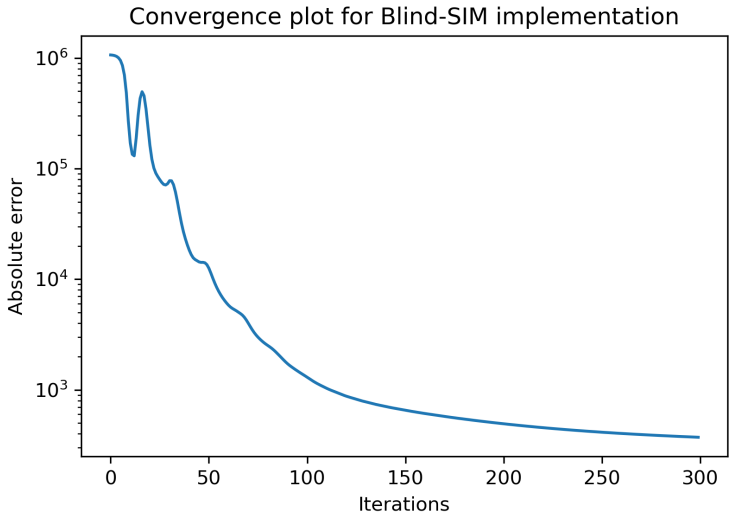


Figure 5.13: Convergence plot for the implementation of Blind-SIM using the gradient descent method for optimisation.

should work as well, so long as the problem overall is not ill-posed, e.g. a stack with only a single frame illuminated by a fringe pattern would not enable super-resolution by SIM.

In the following subsections, I detail attempts made to treat the problem of reconstructing SIM data illuminated with speckle patterns using a model architecture that is overall identical to the one in Section 5.1. The baseline method will be my own implementation of the Blind-SIM algorithm as there are, to the best of my knowledge, no publicly available software that can handle speckle SIM reconstruction, which stands in contrast to the multiple choices of methods to compare with in Section 5.1 for fringe SIM reconstruction.

Overall, this investigation into using ML-SIM for speckle SIM should be viewed as early results that may motivate a more full study. As such, a physical experimental realisation is lacking, indeed even real experimental data, and applications to biomedical imaging are yet to be developed. Even so, in the light of the successful reconstruction of experimental samples by ML-SIM seen in Section 5.1, the results of this section seem promising.

5.2.4 Data generation using speckle pattern illumination

In attempting to address the blind SIM reconstruction problem with ML-SIM, a crucial point is whether the image formation model with speckle illumination can be approximated accurately.

As a simple approximation, the intensity distribution is modelled by generating a large number of randomly distributed small disks, i.e. random positions are assigned to each disk. The disks are allowed to overlap, in which case the occurrence of interference must be

treated. Instead of simulating random interference, with a range of constructive and destructive outcomes, the intensities of the disks are treated purely as additive as a simplification. This corresponds to total constructive interference as if the incident wavefronts were in perfect phase. At first sight, it may seem crude to make such a simplification given the very cause of speckle patterns is the phase mismatch of wavefronts. However, total destructive interference is implicitly modelled in this approach as it simply would manifest in the complete lack of intensity, i.e. dark spots, in the image when wavefronts cancel each other out, which indeed occurs due to the non-uniform random distribution of the generated disks. The inaccuracy of this approach may rather be in the probability distribution of the positions of the disks and the low granularity of the intensities given that partial interference is not modelled.

The intensity distribution of the simulated speckle pattern was assessed by generating speckles with 10,000 and 20,000 disks with random positions that follow a uniform random distribution. The intensity of each disk is set to unitless value of 0.1, which means that the overlap of e.g. two disks result in an intensity value of 0.2 based on the approximation described above. The radii of the disks vary randomly in a small interval of $2-5$ pixels and they are contained in a 512×512-pixel image. The histogram of the recorded intensities versus the expected theoretical probability density distribution Equation (5.2) is shown in Figure 5.14. Without simulating destructive interferences, it is clear that the overall trend between the simulated speckle intensity distribution and the theoretical expectation is similar for the number of disks generated, i.e. 10,000 and 20,000. In the limit of a high number of disks, this approximation breaks down because every pixel will be occupied by disks, thus causing a non-zero intensity baseline without dark spots but instead spots of lower relative intensity. In this limit, the histogram approaches a normal distribution centred around the mean intensity. Despite the fact that the exponential decay rate as a function of intensity differs between the histogram and theoretical distribution, and the granularity of the simulated intensity distribution is low, the overall agreement has been found to be adequate to facilitate the experiments of this section.

An example of a speckle generated with this approximate procedure is shown on the left side of Figure 5.15. As a fundamental assumption of the Blind-SIM algorithm is that the illumination intensity must approach a uniform distribution over time, it is important that the generated speckle patterns have this property. To be more precise, the accumulated intensity of subsequent speckle patterns, i.e. when adding the intensities linearly, must average out any local extrema. Since the positions of the disks are generated from a uniform random distribution and the intensities are added linearly, it is clear that an accumulated signal will also become more uniform as a function of the number of patterns added. By measuring the signal accumulated over 30 subsequent speckle patterns, we see on the right side of Figure 5.15 that the uniformity

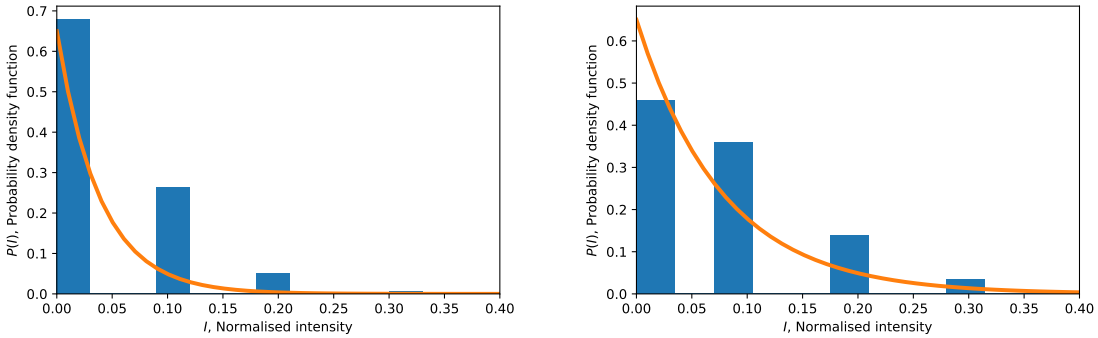


Figure 5.14: Histogram of the intensities in the speckle pattern formed by the collection of generated disks (bars) and the expected distribution from theory (curve). The disks have a random radius of 2-5 pixels and are contained in a 512×512-pixel image. (Left) Speckle based on 10,000 disks. (Right) Speckle based on 20,000 disks.

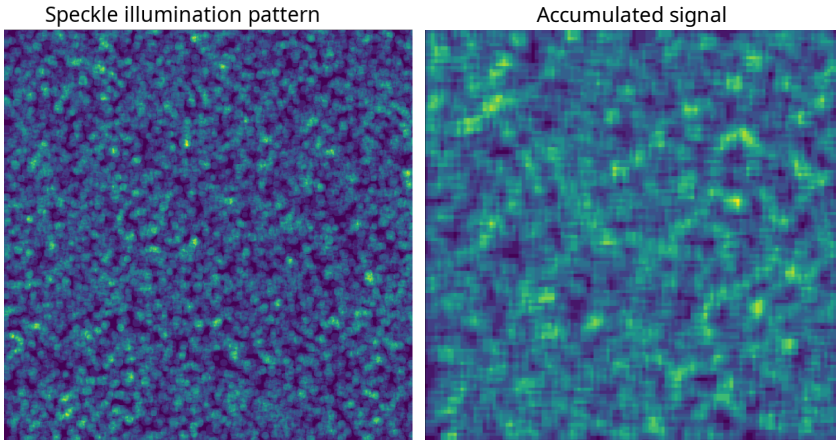


Figure 5.15: Example of simulated speckle pattern (left) and the gradual trend towards a uniform field as more distinct speckle patterns are accumulated (right).

is increased and the variance decreased compared to the intensity distribution of individual speckle patterns on the left side.

Based on the speckle pattern image formation model, synthetic image samples can be generated. A synthetic acquisition made with the simulated speckle pattern used as illumination is shown in Figure 5.16. The image consists of 100 stacked frames with respective speckle patterns, and the first and last frame of the stack is shown. The wide-field image is calculated using the same PSF but uniform illumination. The ground truth image is the original source image used for determining both the wide-field and speckle images.

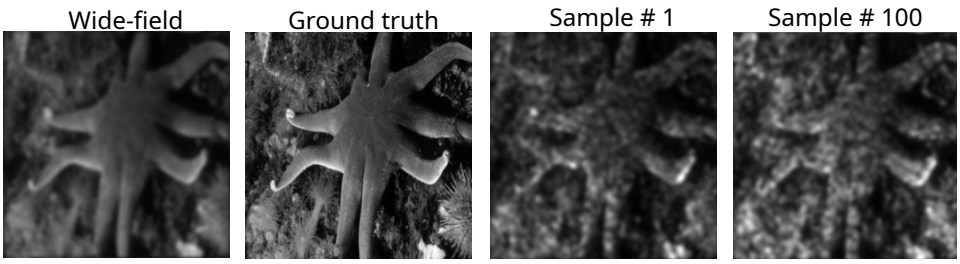


Figure 5.16: Image sample generated from the image formation model that uses speckle patterns for illumination. The ground truth image is the source image that is used to generate the image sample, which consists of 100 frames each illuminated by a distinct speckle pattern. The wide-field image is produced using the same PSF but with a uniform illumination field.

5.2.5 ML-SIM model training for speckle SIM

An important assumption of the numerical optimisation approach to speckle SIM that is taken in the Blind-SIM method is that the complete stack of speckle illuminated frame acquisitions approximately form a uniform illumination field when averaged. This of course becomes increasingly true the more speckle frames are averaged as the individual variability from frame to frame cancels out. To test the importance of this assumption in the context of an ML-SIM model trained on speckle patterns, training datasets with a varying number of speckle illumination patterns for each acquired stack of a sample were generated. The dataset with the highest number of speckles used for illumination is 100, which is of the same order of magnitude as the count of 160 frames per stack used in [145]. The other datasets use gradually fewer speckles for illumination, namely 75, 50, 25 and 10. As such, the dataset with the least number of speckle patterns used for illumination still has more frames per stack than the default fringe SIM configuration of 3×3 frames that is primarily the basis of results in Section 5.1. However, the assumption of the approximative homogeneity upon integrating all illumination patterns is far from satisfied in the case of only 10 speckle patterns, and therefore it would already before considering performance results seem futile to bring the count lower than 10.

Five ML-SIM models were trained each using one of the above-mentioned datasets. The same training hyperparameters, e.g. learning rate, number of epochs and batch size, and the same overall model architecture based on the RCAN network were used, with the only difference being in the input layer that needs to match the dimensionality of the input stacks. The SSIM performance scores on a validation set during training are recorded and the results of the five models can be seen on Figure 5.17. As expected, the models trained with datasets based on a lower count of speckle patterns, especially 10 and 25, perform significantly worse. On the other end of the spectrum, the relative performance difference between models using

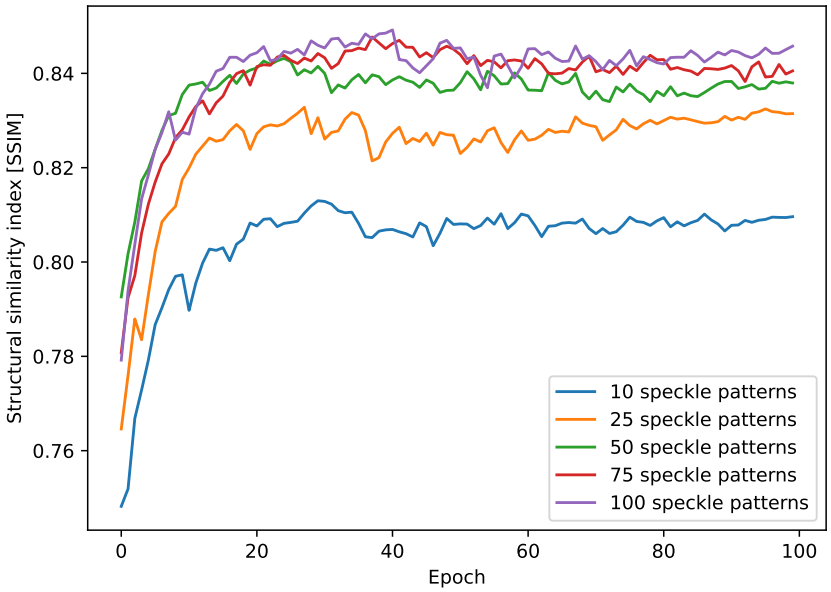


Figure 5.17: ML-SIM models trained with different numbers of speckle patterns compared in terms of the SSIM score on a validation test set during training. A low number of speckle patterns results in significantly inferior performance. However, the model also starts to show diminishing gains when using more than 50 speckle patterns.

more than 50 speckle illumination patterns seem to diminish – for instance, there is almost negligible difference between the model using 75 speckle patterns versus the one that uses 100.

5.2.6 Performance comparison

To compare the performance of the Blind-SIM algorithm with a speckle-based ML-SIM model, I have followed the example of Mudry *et al*. of using a star-like 2-dimensional test target whose fluorescence density is given by

$$\rho(r,\theta) \propto [1+\cos(40\theta)], \tag{5.9}$$

where (r,θ) are the polar coordinates of a point in the sample plane. The advantage of this target is that its radial structure entails that information gradually has a higher spatial frequency closer to the centre. Due to diffraction, and limited image resolution in general, there will always be a cut-off radius under which the radial features are not distinguishable. The test target produced by Equation (5.9) is shown on Figure 5.18 with a wide-field version obtained from blurring with the PSF, which is modelled according to Equation (5.1) that is similar to

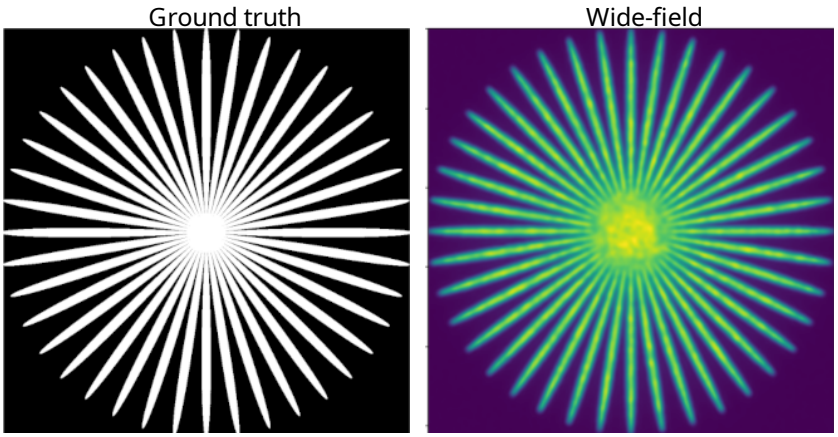


Figure 5.18: Test target used for evaluating the performance of the implementation of the Blind-SIM algorithm and the ML-SIM model trained on synthetic data generated with speckle pattern illumination.

the theoretical PSF when imaging with a circular pupil [140]. In [145], the PSF used towards their reported simulation results is also similar to Equation (5.1) with the exception of another scaling factor that depends on the numerical aperture. Since the numerical aperture is assumed to be a constant this difference is not important.

The implementation of the Blind-SIM algorithm is compared with the ML-SIM model trained on 100 speckle illumination patterns with respect to reconstructing the star-shaped test target. The result is shown in Figure 5.19. The line profiles reveal that the Blind-SIM algorithm introduces a significant amount of artefacts given the highly asymmetric profile despite the input being completely symmetric. As for the ML-SIM model, the output image appears both sharper and cleaner while the line profile is more consistent. The full width half maximum of the central peak, see the dashed line in Figure 5.19, is measured to be 10 % narrower than for the wide-field equivalent image. A distinction needs to be made here regarding the wide-field image as it is not a projection of the input stack, as seen in Section 5.1, because such a projection with the speckle illumination patterns would result in a non-uniform bias compared to the ground truth; rather the wide-field image is calculated using the same PSF but with a uniform illumination field.

ML-SIM with speckle illumination versus fringe illumination

The initial test shown on Figure 5.19 indicates that ML-SIM is proficient in dealing with reconstruction of SIM images with speckle patterns. This begs the question whether an ML-SIM model trained for speckle pattern illumination can compete with one trained for fringe pattern illumination as studied in Section 5.1 with respect to reconstruction accuracy. It seems

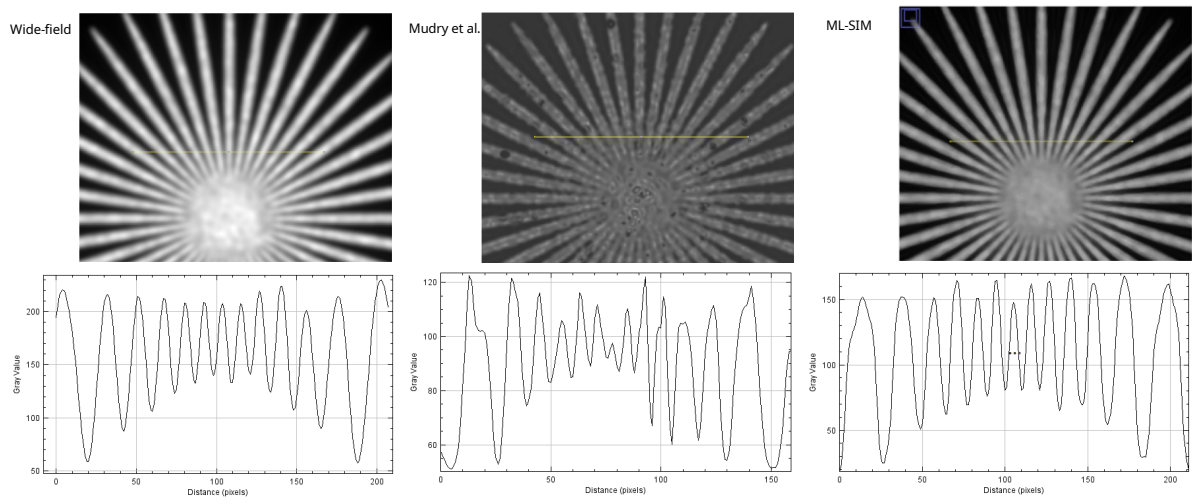


Figure 5.19: Comparison of reconstructed output from my implementation of the Blind-SIM algorithm proposed by Mudry *et al*. [145] with ML-SIM trained on synthetic training data generated with simulated speckle illumination patterns.

obvious that fringe patterns are going to be more photon efficient; after all, the standard configuration of SIM with 3×3 patterns is well optimised in terms of how the frequency support is expanded to an area with twice the radius while keeping the overlap of the frequency content between the acquired frames small. The redundancy of frequency information when using speckle patterns is going to be greater on average since as much as 100 stacked frames are needed for an image. However, it seems interesting to compare the reconstruction performance between the two approaches when the input stacks have a similar image quality in their wide-field projections.

To make a test of two ML-SIM models that are trained for either kind of illumination, two datasets are created, one with fringe SIM data and another with speckle SIM data. The two datasets are generated from the same PSF, which would by default make the input images for fringe illumination much cleaner given the non-uniformity inherent to speckle illumination. Therefore, to match the quality of the input data for the two datasets, the fringe illumination patterns are additionally degraded by introducing a large margin of error for the phase and frequency, such that the likelihood of the fringes in the sum of the 3×3 SIM frames cancelling out is low. This results in wide-field projections that are corrupted by these errors in the fringe pattern, and thus show a similar non-uniformity to the speckle pattern illumination. With the increased difficulty of the reconstruction task for SIM data with fringe illumination, the two approaches can be more fairly compared. In practice, this might correspond to the choice between a standard SIM instrument that is unable to produce consistent fringe patterns, e.g. due to uneven phase steps and optical aberrations in the system such as for the interferometric

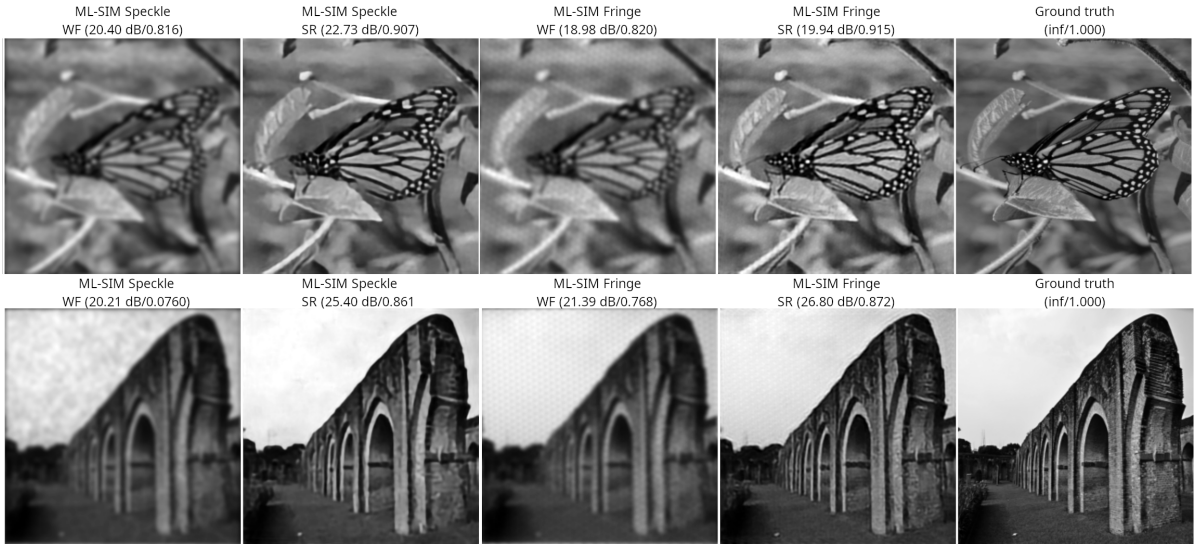


Figure 5.20: Two variants of the ML-SIM model, one based on SIM with speckle pattern illumination and the other with fringe illumination as presented in Section 5.1. The variants are compared here in terms of inputs that are similarly degraded from their respective image formation models. The wide-field (WF) version in each case represents the wide-field projection, i.e. the average intensity projection of the input stack.

Microscope 2 in Section 5.1, and an instrument that has a more simple setup relying on speckle patterns formed by illumination through a diffuser.

The datasets of speckle illumination patterns and fringe illumination patterns now have matching input image qualities and are used to train two models, ML-SIM Speckle and ML-SIM Fringe, respectively. The two models share an identical architecture, except for the input layer, and they are trained with an equal number of training updates and training samples. A subset of each dataset reserved for testing is used to evaluate the performance of each model, and two example outputs from the respective test subsets are shown on Figure 5.20. While the average input image quality across the two datasets is similar, there are differences on a basis of individual samples. In the first of the two examples on Figure 5.20, the wide-field projection (WF) for ML-SIM Speckle has a higher PSNR than for ML-SIM Fringe, while the opposite is true in the second example. For the two examples, the reconstruction outputs of the two models show a similar improvement over the wide-field projections, but the scores for the ML-SIM Speckle model are slightly higher with outputs showing fewer artefacts. This is also the case when averaged over the entire respective test sets. The artefacts that appear in the outputs of ML-SIM Fringe are due to the large margin of error introduced in the fringe pattern generation, which the model is unable to compensate for completely.

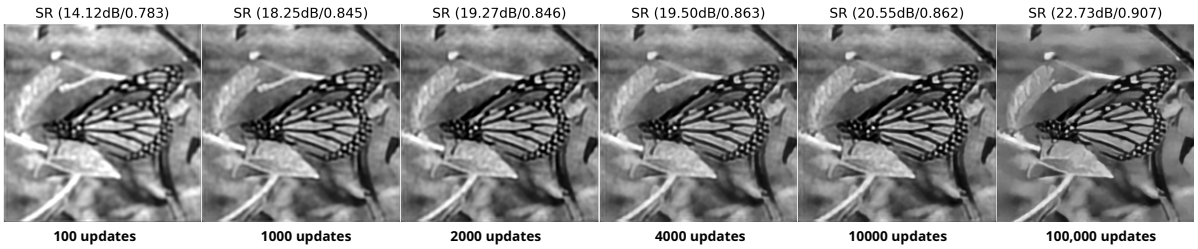


Figure 5.21: The spot-like artefacts that derive from the non-uniformity of the speckle illumination patterns gradually disappear from the outputs of ML-SIM models depending on the number of updates of the network weights that are performed during training.

The slight performance edge of the speckle-based ML-SIM model is likely due to the larger input stack. Although the wide-field projections have a similar quality, these projections are calculated as the pixelwise arithmetic mean across the input stacks, but as seen in Section 3.4.1 the arithmetic mean is not necessarily the optimal way to even out noise in a stack of frames with identical signal. Therefore, a wide-field projection may not be a very good measure of the quality of an entire stack in terms of the collective structural information it contains. However, the test of the two models still provide qualitative insight into the viability of using ML-SIM for SIM data acquired with speckle pattern illumination. The results indicate that an experimental realisation facilitating ML-SIM for reconstruction could function in a comparable fashion to the vanilla ML-SIM method of Section 5.1.

The non-uniformity in intensity that comes with speckle pattern illumination appears to not have manifested into the reconstruction output from the ML-SIM model trained with speckle patterns when considering the examples of Figure 5.19 and Figure 5.20. There seem to be no indications of spot-like artefacts as is the case for the reconstruction outputs from the Blind-SIM algorithm. However, the reconstruction outputs from ML-SIM models that have been trained less clearly exhibit this type of artefact. The spot artefacts progressively get less pronounced as more updates to the network's weights are performed. This is shown on Figure 5.21.

5.2.7 Discussion

An implementation I have made of the Blind-SIM algorithm proposed in [145] has been described and evaluated. The data used for testing is generated with an approximate image formation model that mimics the random nature of speckle patterns. The performance of the implemented Blind-SIM algorithm has been compared to an ML-SIM based model that is trained with synthetic training data that has been generated with the same image formation model.

The ML-SIM model with speckle patterns shows potential. The performance in terms of run time and output accuracy for reconstructing a test target also used in [145] points to the neural network based approach having large advantage for the highly numerical optimisation-dependent problem of ”blind-SIM”. For a mid-tier workstation the run time for a single reconstruction of the Blind-SIM algorithm was 50 minutes using 300 iterations in the iterative solver written in Python. An inference with ML-SIM running on the same machine, although using a graphics processing unit, evaluates in less than a second. In terms of keeping artefacts in reconstruction output to a minimum, ML-SIM appears to also have an edge. When comparing reconstruction performance for ML-SIM models trained for SIM with fringe illumination and speckle illumination, respectively, for input that is similarly degraded, it is clear that fringe patterns are not a prerequisite for a well-working ML-SIM model. In fact, early results indicate that fringe patterns may not even be an advantage over speckle patterns provided adequately large images of stacked frames, 50+ frames, illuminated with distinct speckle patterns are used as input.

The experiments of this section although promising are still early results. A realisation with an experimental setup would be an important next step for a thorough validation of the method. I have attempted to obtain experimental speckle pattern from the research group that published the original Blind-SIM paper but was unsuccessful. An experimental implementation for Blind-SIM could likely be relatively easy to build since the speckle patterns could be generated by propagating the illuminating beam through a diffuser as it is demonstrated in [145], yet this has been out of the scope for my PhD project. If the ML-SIM based method would work with an experimental setup, the overall system could be interesting for various applications in biology as well as contributing towards providing an alternative to the standard SIM instruments using fringe illumination.

5.3 Spatio-temporal Vision Transformer for Super-resolution Microscopy

The content of this section is based on my pre-print publication "VSR-SIM: Spatio-temporal Vision Transformer for Super-resolution Microscopy" [28].

In Section 5.1, it was established that reconstruction of SIM data generally is prone to artefacts. This becomes especially problematic when imaging highly dynamic samples because previous methods rely on the assumption that samples are static. Here, I propose a new transformer-based reconstruction method, VSR-SIM, that uses shifted 3-dimensional window multi-head attention in addition to channel attention mechanism to tackle the problem of video super-resolution (VSR) in SIM. The attention mechanisms are found to capture motion in sequences without the need for common motion estimation techniques such as optical flow. I take an approach to training the network that relies solely on simulated data using videos of natural scenery with a model for SIM image formation. I demonstrate a unique use case enabled by VSR-SIM referred to as rolling SIM imaging, which increases temporal resolution in SIM by a factor of 9. This method can be applied to any SIM setup enabling precise recordings of dynamic processes in biomedical research with unprecedented granularity.

5.3.1 Introduction

Optical microscopy is limited by the diffraction of light occurring in the optics of imaging systems. For visible light, the diffraction limit, also known as the Abbe resolution limit [138], is around 200 nm. Structured illumination microscopy (SIM) is an optical microscopy technique that can achieve a two-fold spatial resolution improvement, thus enabling sub-diffraction limit imaging – a regime important for biomedical imaging [83]. Furthermore, SIM is live-cell compatible as it can be performed at relatively low excitation power. A significant challenge in applying SIM, however, is the reconstruction of the acquired data into super-resolved images. The reconstruction problem in SIM is an inverse problem similar to deconvolution [192] but makes use of shifted high frequency information. The frequency-shifted signals are obtained by illuminating the sample with a temporal sequence of illumination patterns, generally sinusoidal fringes with varying orientations and phase shifts, and an image is captured for each respective pattern.

The collection of SIM images corresponding to the sequence of illumination patterns, typically a stack of 9 frames, is then used to reconstruct a super-resolved image. Since the photon efficiency of SIM makes the technique live-cell compatible, it is possible to image highly dynamic phenomena [78, 163], yet the reconstruction methods that are most widely

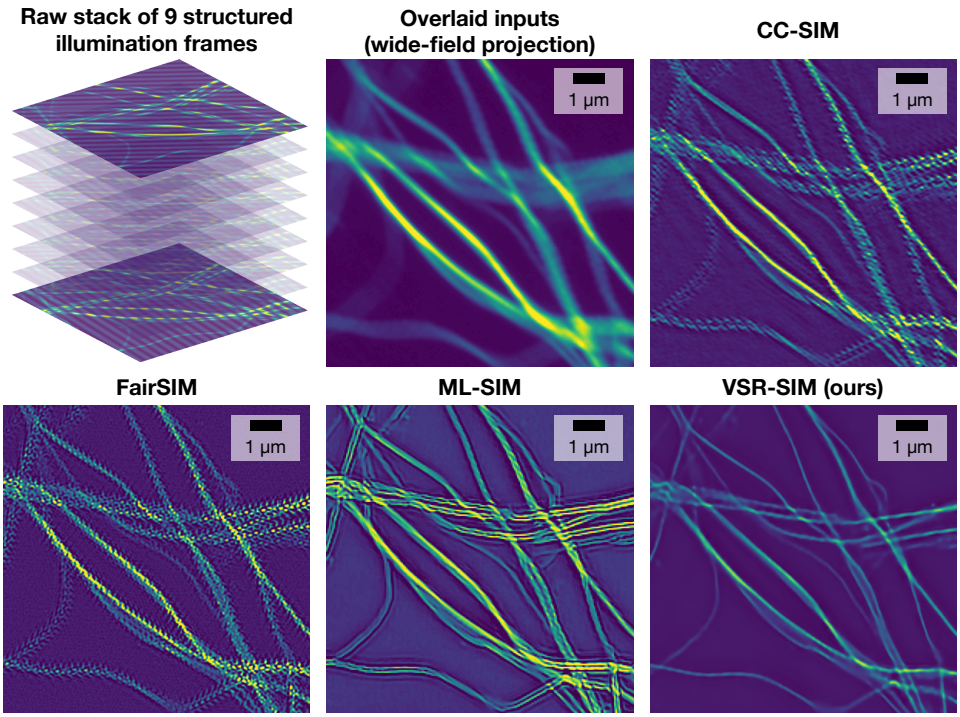


Figure 5.22: Structured illumination microscopy image sequences of dynamic samples give rise to motion artefacts for previous reconstruction methods such as cross-correlation SIM (CC-SIM) [214], FairSIM [146] and ML-SIM (Section 5.1). The input image stack is an experimental sample of microtubules.

used do not utilise the temporal dimension of the acquired data [146, 214, 103, 80], because the standard semi-analytical Fourier formalism assumes a static sample. Hence, motion of the sample between acquired frames manifests as motion blur and reconstruction artefacts – see Figure 5.22.

Deep learning offers an effective way to achieve motion compensation for video super-resolution (VSR). Recent studies demonstrate reconstruction of SIM images using neural networks [29, 86, 118], offering advantages such as improved speed and robustness to noise,

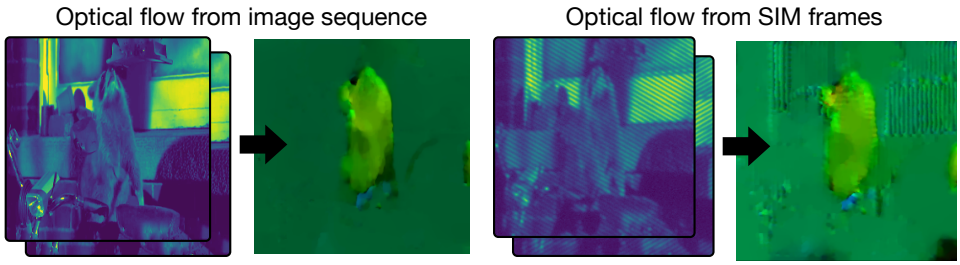


Figure 5.23: Optical flow computed from SIM frames leads to artefacts, thus making it less useful in video processing of SIM data.

but none of these reconstruction methods make use of the temporal dimension of the live-cell data. To obtain a spatio-temporal reconstruction method for SIM, I identify the following two problems to overcome: (a) ground truth data for supervised learning will inherit motion blur if the targets are obtained from traditional reconstruction methods; (b) regular motion estimation methods do not work accurately on SIM data. Machine learning implementations for SIM reconstruction generally use as ground truth data a collection of carefully performed reconstructions from traditional methods, which relies on an analytical framework that assumes static samples, thus causing motion artefacts to manifest in the training data. As for (b), a common way to incorporate high-level reasoning about motion and occlusion in a model is bidirectional optical flow. However, such algorithms are not directly suited for SIM imaging, because the illumination patterns in the raw data prevent accurate calculation of motion – the varying patterns tend to be confused with motion of the subject as illustrated on Figure 5.23.

Here, a method is proposed to address these two problems by building upon recent advances in using neural networks for SIM reconstruction and video super-resolution. I generalise the approach to supervised learning proposed in the previously described method ML-SIM (Section 5.1), in which SIM image formation is modelled to obtain synthetic training data. Instead of simulating SIM image data using static images, I use video sequences instead, which facilitates the learning of motion compensation. To address (b), we propose a 3D transformer network architecture that solely relies on attention mechanism rather than optical flow to handle subject motion. The contributions are three-fold:

- I demonstrate a new approach to synthesising training data for machine learning models to learn spatio-temporal SIM reconstruction, in which SIM image formation is simulated using video data sequences as inputs. This enables models to be optimised for highly dynamic sequential live-cell SIM data.

- I propose a video super-resolution transformer architecture that uses shifted windows with 3-dimensional patches to capture the spatio-temporal correlations in live-cell SIM data with windowed multi-head attention. I introduce residual connections between transformer blocks with channel attention as an additional attention mechanism.

- I showcase a unique application of this method, rolling SIM imaging, where a moving window of frames is used for reconstruction. The reconstruction method lends itself particularly well to rolling SIM imaging because it can be recast as a video super-resolution problem, where the reconstruction of each SIM stack uses SIM frames from the previous and subsequent SIM stack acquisition. This increases the temporal resolution of SIM imaging by a factor of 9, enabling dynamic processes in biomedical research to be resolved without the motion artefacts that plague previous methods.

5.3.2 Related work

Optical super-resolution microscopy. Several semi-analytical methods have been proposed for SIM reconstruction [59, 6, 215, 214, 103, 146], e.g. FairSIM and OpenSIM. These methods rely on Fourier transformations, Wiener filters and iterative deconvolution, which can induce honeycomb and ringing artefacts, especially when noise and motion blur are significant [38]. Multiple machine learning implementations for SIM reconstruction have been proposed in the past year [86, 118, 29] based on convolutional neural networks that take in SIM stacks and output super-resolved images. Two such examples are U-Net-SIM [86] and ML-SIM (Section 5.1) using U-Net [173] and RCAN [227] backbones, respectively. These methods offer reconstruction with fewer frames, higher processing speed and increased robustness to noise compared to Fourier methods. None of these studies considered fast-moving samples. In [81], however, SIM is applied to image highly dynamic samples using a semi-analytical reconstruction method. This is achieved using rolling SIM imaging, as further explored in Section 5.3.5, with a very short exposure time, such that motion artefacts can be minimised. This can lead to impressive frame rates, but at a significant loss of image quality, i.e. low signal-to-noise ratio from which spatial resolution decreases. This trade-off between temporal and spatial resolution is prevalent in the field because none of the existing reconstruction methods for SIM exploits the spatio-temporal nature of live-cell data. Applications of existing methods may only reduce motion artefacts via this trade-off, whereas the capability to perform motion compensation during reconstruction would handle these artefacts directly while maintaining image quality.

Image and video super-resolution. Methods that use convolutional neural networks as a backbone have long been state-of-the-art for image and video super-resolution (SR). Dong *et al*. pioneered the pursuit of learning-based methods for image SR by achieving superior performance to traditional methods using a CNN with only three layers [40]. A similar network for VSR was proposed by Kappeler *et al*. [91]. With the emergence of residual networks [70], it became possible to build deeper networks. Ledig *et al*. repurposed ResNet for SR with the network SRResNet [105]. An attention mechanism was introduced by Zhang *et al*. [227] with residual channel attention network (RCAN) becoming a new state-of-the-art method. More recently, multi-head attention has been introduced for SR using transformer-based architectures with IPT [26] and SwinIR [113].

For VSR, the spatio-temporal correlations between input frames are essential to model for optimal performance. Most VSR methods use frame alignment enabled by motion estimation and compensation [119]. For motion estimation, a popular approach is using optical flow [79]. A state-of-the-art VSR method that uses optical flow is RBPN [63], which is based

on a recurrent CNN architecture. Recently, the method BurstSR [10] was proposed for SR reconstruction of images taken in quick succession with a handheld camera. The problem is similar in principle to SIM reconstruction, but the method is not directly applicable as it is based on optical flow for alignment. Methods that do not use optical flow tend to rely on 3D convolutional networks [88, 122]. However, Choi *et al*. demonstrated that channel attention as a sole mechanism is a strong baseline for motion compensation in the related problem of video interpolation [27].

Vision transformer. With the advent of Vision Transformer (ViT) [42] transformer networks are beginning to replace CNNs for low-level computer vision tasks. ViT introduced multi-head self-attention (MSA) for image input, which proves to be a very flexible mechanism for vision, but does require a substantial number of trainable parameters compared with equivalently performing CNNs. Liu *et al*. demonstrated that using a hierarchy of shifted window MSA modules, their proposed transformer architecture, Swin, can incorporate the large receptive field of ViT, while having the same efficient inductive bias that CNNs offer [121]. Variations of the Swin transformer have become state-of-the-art in image restoration, SwinIR [113], and video classification [122].

5.3.3 Temporal SIM data generation

Acquiring a real pairwise dataset for supervised learning in the context of super-resolution microscopy is problematic. Experimentally, the ground truths cannot be obtained, which leaves the options of using either the output from traditional reconstruction methods as a target [87, 118] or a different optical super-resolution modality [202]. The former approach prevents the method from generalising and improving beyond traditional methods, and the latter is highly prone to artefacts, while not being live-cell compatible. Therefore, I take the approach of generating a synthetic dataset using a SIM image formation model [30] on a video dataset, which provides ideal ground truths and diverse training data.

Video datasets

BBC video dataset. Inspired by DIV2K [2] for SISR, as used in Section 5.1, I built a large video dataset focusing on diversity and high-resolution footage. Specifically, this dataset was designed to have targets of at least 1024×1024 pixels to make the image formation model more consistent with typical experimental data from SIM systems, thus facilitating model inference performance. Many previous VSR datasets are limited in scope and are intended for video classification [92, 1] or more suitable for testing, e.g. REDS [149], while others only

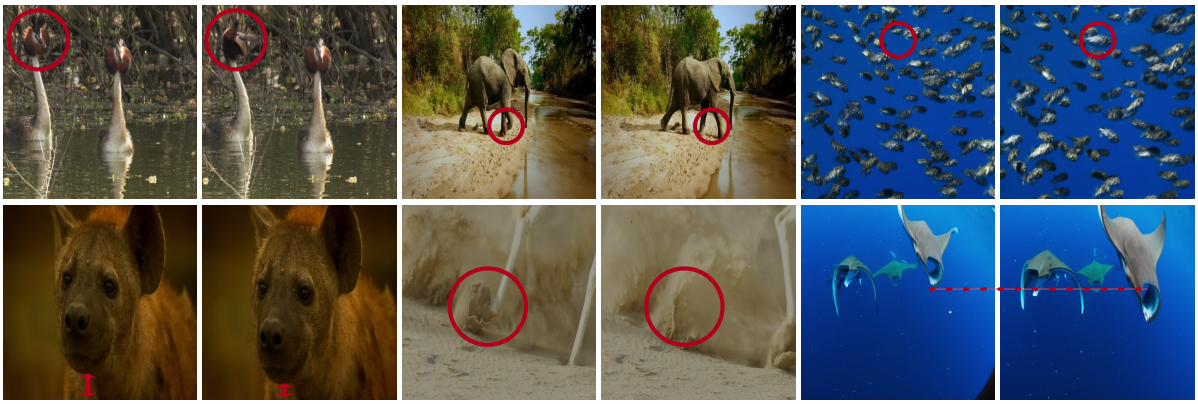


Figure 5.24: First and last image from 6 image sequences from the BBC video dataset.

have a small subset of high-resolution, diverse data, e.g. Vimeo90k [219]. The new dataset consists of 200 hours of high-quality footage from nature documentaries produced by the BBC. Samples are included here with permission from BBC and video data have been obtained under the ERA License. The collection of videos was sampled to generate 100,000 image sequences, each consisting of 9 frames. Immediately consecutive frames were used for the sequences, i.e. no frame skipping was used, and every sequence was 5 seconds ahead of the previous. The original videos are in either FHD, 1920×1080 pixels, or UHD, 3840×2160 pixels, resolutions, but were downsampled with bicubic interpolation to 512×512 pixels for inputs and 1024×1024 pixels for targets, because the models were trained to upscale by a factor of 2 corresponding to standard SIM implementations. The source videos are encoded with H.264, which utilises frame-to-frame compression, but I found compression artefacts to be negligible. Sample sequences can be seen in Figure 5.24. As indicated on Figure 5.24, the sequences tend to have a fixed background while objects are subject to motion.

A subset was reserved for testing, for which I also used DIV2K and REDS. As DIV2K dataset is a single image dataset, the image data generated with the image formation model described in the following paragraph corresponded to imaging static subjects. The REDS dataset feature videos recorded with a handheld camera with a high level of image translation from frame to frame. To make the motion in the REDS video even more extreme, I prepared an extra test set by sampling the videos with frame skipping, such that only every second frame was kept. The combined datasets were used to prepare four test sets to assess reconstruction performance in different motion regimes. The difficulty associated with one of these datasets depend on the level of motion that its samples exhibit. I quantified this using the mean and maximum value of optical flow magnitude averaged over all samples in the respective datasets. See Table 5.1 for further specification.

Test sets	Motion regime			
	Static	Medium	Fast	Extreme
Source	DIV2K	BBC	REDS	REDS
Data type	Image	Video	Video	Video
Frame skip	-	No	No	Yes
Samples #	200	50	10	10
Max flow	0	10.2	27.3	46.2
Median flow	0	1.5	10.4	18.1

Table 5.1: The four test sets that have been prepared for experiments using the source datasets DIV2K [2], a subset of the BBC video dataset, and REDS [149]. The motion is amplified by skipping every other frame for the Extreme test set. Motion is quantified by calculating the maximum and median of the magnitude of optical flow between the first and centre frame in all sequences for a dataset at 512×512-pixel resolution.

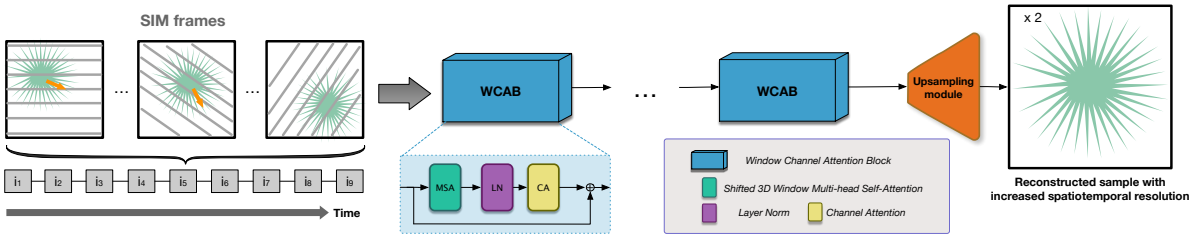


Figure 5.25: Architecture of the proposed windowed channel attention network. Skip connections are added between the attention blocks in a similar fashion to residual networks.

Image formation model. The image formation model used for VSR-SIM largely follows that of ML-SIM, Section 5.1, with the inclusion of time dependency for the sample distribution.

The illumination fringe pattern, $I_{\theta,\phi}(x,y)$, is given by the standard sinusoidal interference pattern Equation (B.2). The fluorescent response of the sample is then modelled by the multiplication of the sample distribution, $S_t(x,y)$, i.e. input image, at time t and the illumination pattern intensity $I_{\theta,\phi}(x,y)$. As before, the final image, $D_{t,\theta,\phi}(x,y)$, is formed after blurring by the PSF, $H(x,y)$, and addition of white Gaussian noise, $N(x,y)$,

$$D_{t,\theta,\phi}(x,y) = \left[S_t(x,y)I_{\theta,\phi}\right] \otimes H(x,y) + N(x,y), \tag{5.10}$$

where $\otimes$ is the convolution operation. The set of sampled images from a sequence in the video dataset corresponds to the time points $t \in [1,9]$. A full SIM stack is comprised of the set $\left\{D_{t,\theta,\phi} \,|\, t \in [1,9]\right\}$, where each value of t is associated with a distinct illumination pattern, i.e. a unique permutation of θ and ϕ. Each consecutive 9 frames then contain a full cycle of illumination patterns. As for ML-SIM, both Gaussian noise and random errors in the pattern

generation are simulated to approximate the inherent uncertainty in an experimental setup and force the model to generalise beyond system-specific parameters. Poisson noise can further be introduced to more realistically approximate the noise sources present in experimental data. For implementation details and specification of optical parameters, see Section B.2.3.

5.3.4 Model architecture

The proposed model is inspired by the vision transformer network [42] in particular its more efficient shifted window variant, Swin [121], with its extension for video classification, Video Swin [122], and adaption to image restoration, SwinIR [113]. Swin introduced the inductive bias to self-attention called shifted window multi-head attention (SW-MSA), which can be compared to the inductive bias inherent to convolutional networks. SwinIR introduced residual blocks to the Swin transformer to help preserve high-frequency information for deep feature extraction. The Video Swin transformer generalised the SW-MSA to three dimensions, such that spatio-temporal data can be included in the local attention for the self-attention calculation. Further to this, the success of the channel attention mechanism in [227] inspires the inclusion of this other inductive bias in addition to 3D local self-attention following the SW-MSA approach.

The inputs to the model have dimension $T \times H \times W \times C$, where T is 9 for SIM reconstruction and C is 1. A shallow feature extraction module in the beginning of the network architecture Figure 5.25 projects the input into a feature map, F_0, of $T \times H \times W \times D$ dimension, where the embedding dimension, D, is a hyperparameter. The feature map is passed through a sequence of residual blocks, denoted Window Channel Attention Block (WCAB)

$$F_i = H_{\text{WCAB}}(F_{i-1}), \quad i = 1,..,n \tag{5.11}$$

Inside each WCAB is a sequence of Swin Transformer Layers (STLs), in which multi-head self-attention is calculated using local attention with shifted window mechanism. Inputs to STL layer are partitioned into $\frac{T}{P} \times \frac{HW}{M^2}$ 3D tokens of $P \times M^2 \times D$ dimension. For a local window feature, $x \in \mathbb{R}^{P \times M^2 \times D}$, query, key and value matrices, $\{Q,K,V\} \in \mathbb{R}^{PM^2 \times D}$, are computed by multiplication with projection matrices following the original formulation of transformers [198]. Attention is then computed as

$$\text{Attention}(Q,K,V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V, \tag{5.12}$$

where $B \in \mathbb{R}^{P^2 \times M^2 \times M^2}$ is a relative positional bias found to lead to significant improvements in [121]. STLs are joined in a way similar to the residual blocks, although the use of SW-MSA is

alternated with a version without shifted windows, W-MSA, ensuring that attention is computed across window boundaries, which would not have been the case without SW-MSA.

After the final STL, the m-th layer, in a WCAB, a transposed 3-dimensional convolutional layer is used to project the 3D tokens back into a $T \times H \times W \times D$ feature map, $F_{i,m}$. A channel attention module is then used on $F_{i,m}$ to determine the dependencies between channels following the calculation of the channel attention statistic [227]. The mechanism works by using global adaptive average pooling to reduce the feature map to a vector which after passing through a 2D convolutional layer becomes weights that are multiplied back onto $F_{i,m}$, such that channels are adaptively weighed. A residual is then obtained by adding a skip connection from the beginning of the i-th WCAB to prevent loss of information, i.e. low-frequency information, and the vanishing gradient problem. A fusion layer combines the temporal dimension and the channel dimensions. For the final upsampling module, I use the sub-pixel convolutional filter [188] to expand the image dimensions by aggregating the fused feature maps. The implementation is available on GitHub[1] and further documented in Section B.2.

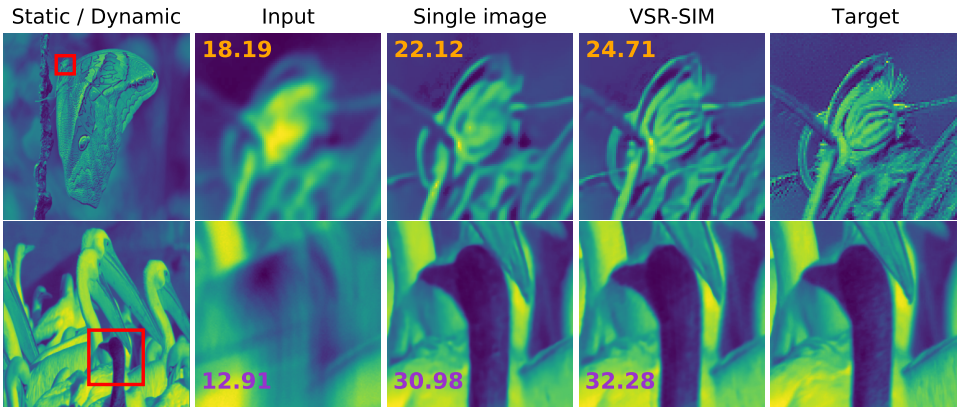


Figure 5.26: For static subjects (top row) the method defaults to standard SIM reconstruction, which has very significant improvements over a deconvolution baseline trained with the same architecture. For dynamic input data (bottom row) the advantage of SIM diminishes depending on the level of motion, but importantly VSR-SIM does not generate motion artefacts in this setting.

5.3.5 Experiments

Implementation details. All models described in the following were trained using the Adam optimiser and a mean squared error loss function with a learning rate of 1e-4 that is halved every 100,000 iterations. A total of 500,000 iterations were completed, equating to 5 epochs of the BBC training dataset. A set of 4 Nvidia A100 GPUs was used with a batch size per GPU of 4. Training samples were randomly cropped to 128×128-pixel inputs and 256×256-pixel targets,

[1] https://github.com/charlesnchr/VSR-SIM

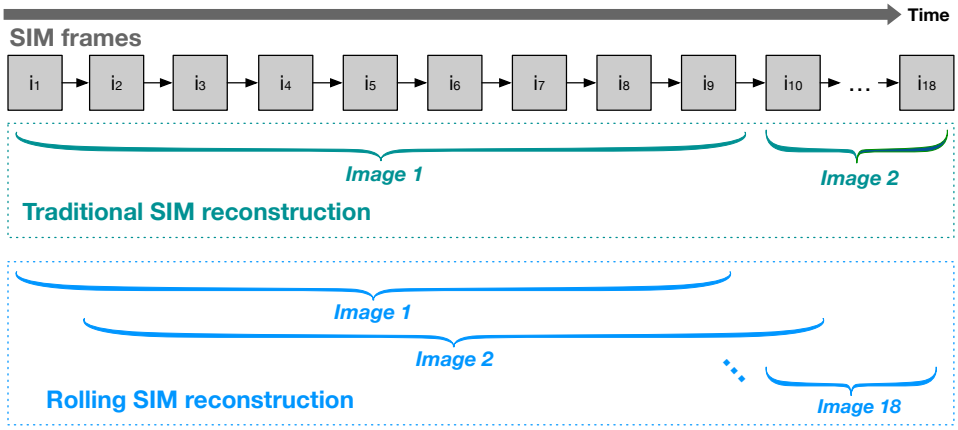


Figure 5.27: Rolling SIM imaging scheme for structured illumination microscopy, which is utilised in the proposed method.

while inference was performed with 512×512-pixel inputs resulting in 1024×1024-pixel outputs. For VSR-SIM, the WCAB number, STL number, window size, embedding size D and attention head number are set to 6, 6, 8, 96 and 6, respectively. The hyperparameters of the other tested architectures follow original implementations and are further specified in Section B.2.5.

Reconstruction method	Test set (PSNR)	
	Static	Medium
Wide-field baseline	22.79	17.31
CC-SIM [214]	27.99	16.98
OpenSIM [103]	28.34	14.04
FairSIM [146]	28.54	15.34
ML-SIM (Section 5.1)	32.30	18.41
VSR-SIM	**34.74**	**30.15**

Table 5.2: Synthetic test sets were evaluated with four existing SIM reconstruction methods and VSR-SIM. The static test set was generated using still images from DIV2K [2] and the dynamic test set was generated using image sequences sampled from the BBC video dataset. At high levels of motion, other SIM reconstruction methods fail, but VSR-SIM can maintain a high reconstruction quality for the dynamic test set.

Comparison with state-of-the-art

SIM reconstruction methods. The Static and Medium test sets, see Table 5.1 for details, were evaluated with VSR-SIM and four existing SIM methods: CC-SIM [214], OpenSIM [103], FairSIM [146] and ML-SIM (Section 5.1). The results are listed in Table 5.2 based on peak signal-to-noise ratio (PSNR). For the Static test set, the difference in reconstruction

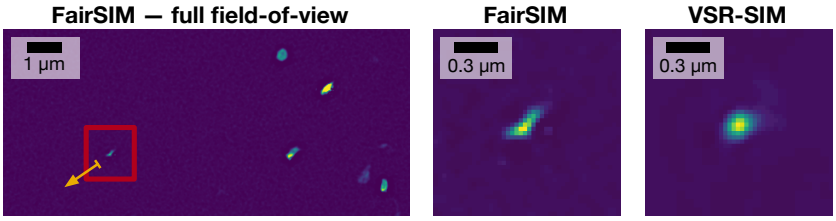


Figure 5.28: Lysosome, a spherical vesicle, moving rapidly in the endoplasmic reticulum. FairSIM unable to handle motion blur reconstructs an elongated shape, while VSR-SIM reconstructs a circular shape consistent with the known shape of the lysosome.

Method	Test set (PSNR)			
	Static	Medium	Fast	Extreme
Bicubic†	26.40	26.35	22.63	21.08
SISR†	31.23	28.08	25.38	22.50
VSR†	31.15	28.15	25.41	**22.98**
VSR-SIM	**34.74**	**30.15**	**26.04**	22.95
RBPN	33.16	29.25	25.29	21.48
Wide-field	26.24	22.99	19.32	18.77

Table 5.3: Test of VSR-SIM method in different motion regimes compared with baseline models trained and evaluated using input without structured illumination. †: methods based on input without structured illumination patterns. The SISR and VSR baselines use the same architecture as VSR-SIM. The sub-diffraction limit resolution of SIM is lost when the amount of motion becomes extreme but is still achievable with the Fast test set. RBPN that uses optical flow for motion estimation was not found to perform comparably, suggesting that optical flow is not needed.

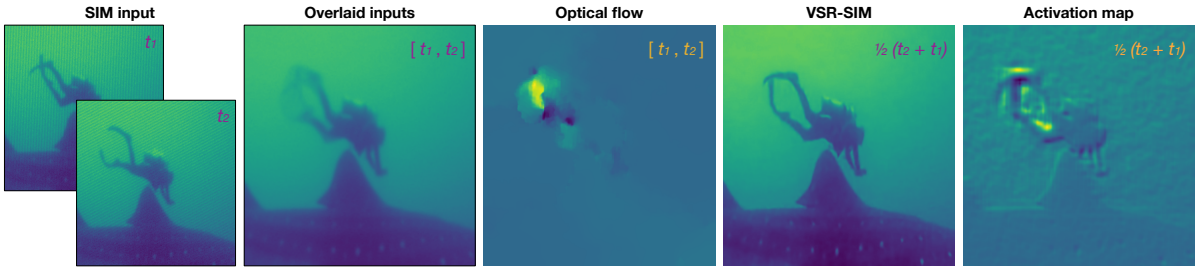


Figure 5.29: Self-attention appears to emphasise the regions, in which motion occurs. The activations from the final attention heads are found to be well correlated with intensity maps of optical flow.

quality is relatively even, but for the Medium test set, most previous methods fail to surpass the diffraction-limited wide-field baseline. This is due to motion artefacts and inaccurate numerical optimisation (e.g. parameter estimation using peak finding in the case of FairSIM) becoming substantial. An example illustrating motion artefacts in reconstruction output for an input sample with significant motion is shown in Figure 5.22.

I tested the spatio-temporal resolution of reconstruction on a real sample by imaging fast-moving lysosomes along the endoplasmic reticulum (ER) in COS-7 cells. I use the SiR-lysosome fluorophore with an excitation wavelength of 652 nm. Given the same raw data, differences are clear in the shape of the lysosome following reconstructing with FairSIM and VSR-SIM, see Figure 5.28. FairSIM produces an elongated shape, suggesting that motion blur is reconstructed into features, which is further supported by the simulated test in Section 5.3.5.

CA	SW-MSA	3D window	Score (PSNR)
✓			29.06
	✓		29.48
	✓	✓	29.10
✓	✓		30.01
✓	✓	✓	**30.15**

Table 5.4: Ablation study on the inclusion of different attention mechanisms. CA is channel attention [227], SW-MSA [121] and 3D window refers to 3D window attention for spatio-temporal data [122]. The scores are based on evaluations on the Medium test set.

Ablation study

No structured illumination patterns. An important baseline for SIM reconstruction is deconvolution. A single image deconvolution method is useful for wide-field imaging to counter the effect of the PSF and noise sources, but it cannot provide optical SR. I trained a model with the same architecture as VSR-SIM using the equivalent dataset without illumination patterns. Examples of output can be seen in Figure 5.26, illustrating the SISR baseline model versus VSR-SIM that takes SIM input. In the first input sample, the subject is static, and the quality difference of the outputs is significant. For more dynamic subjects, the difficulty of the SIM reconstruction problem increases, and the difference to the SISR baseline is smaller. I explored this further by testing models on the four test sets shown in Table 5.1. The four test sets are evaluated with a deconvolution SISR baseline, a deconvolution VSR baseline, a state-of-the-art VSR method RBPN [63] and the VSR-SIM method. The two baseline models are based on the VSR-SIM architecture but trained and tested without structured illumination patterns, while RBPN and VSR-SIM are trained with SIM inputs. Only the centre frame in a sequence corresponding to the target is input to the SISR model, whereas the VSR model works on the full image sequence. The test results in Table 5.3 show that VSR-SIM enables high-quality SIM reconstruction in every motion regime. The quality of the reconstruction outputs is markedly better than for the baselines in all but the most extreme case with frame skipping. Hence, at

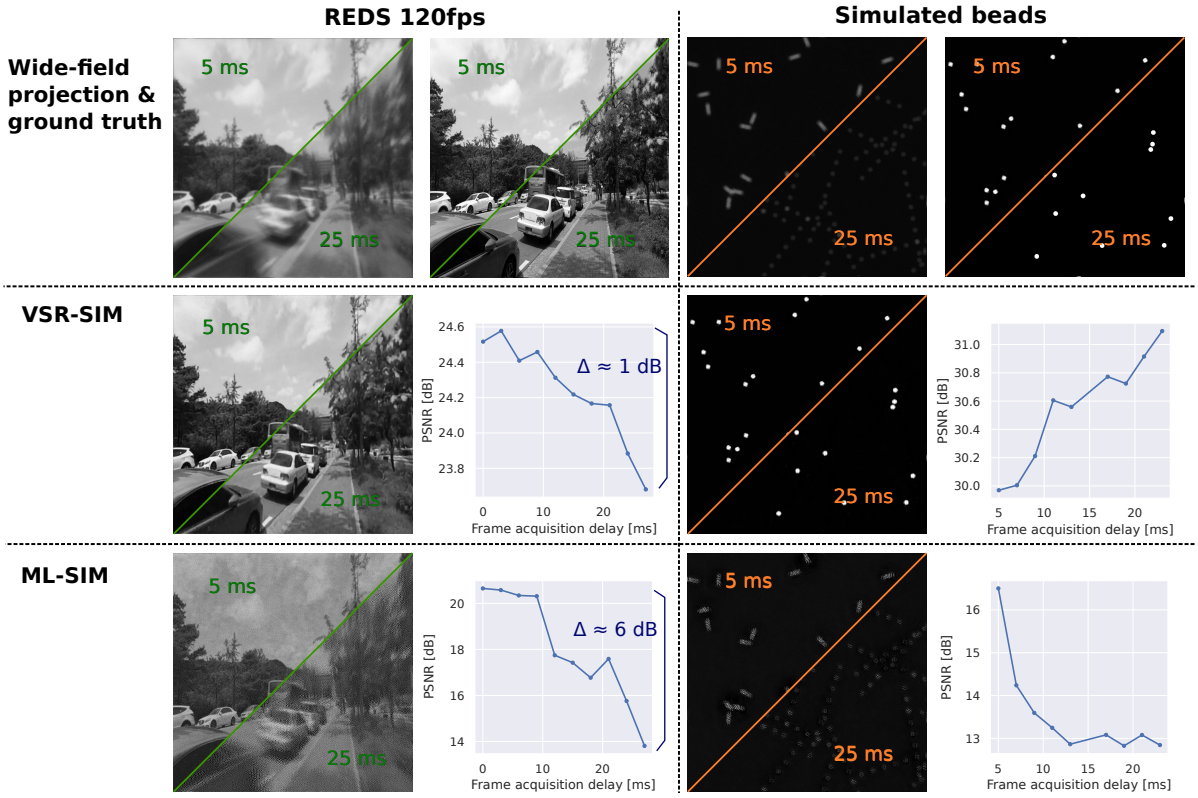


Figure 5.30: Reconstruction performance for VSR-SIM does not collapse for inputs that exhibit significant levels of motion. Given the same inputs sequences, the motion can be controlled via a set delay between frames. This is done with frame skipping for a high frame rate video sequence, REDS 120fps [149], and sequences of simulated beads.

very high levels of motion, the SIM modality does not offer an advantage over conventional imaging. This is consistent with the theoretical findings of Ströhl and Kaminski [194].

Optical flow. As illustrated in Figure 5.23, the calculation of optical flow can be hindered by the presence of an illumination pattern. The quantitative impact of including optical flow is tested by training RBPN, which uses optical flow to input aligned frames into a recurrent network using a mechanism called back-projection. In Table 5.3, it is found that the VSR-SIM model outperforms RBPN in different motion regimes, despite not using optical flow. This indicates that the two attention mechanisms of VSR-SIM are sufficient to attend to regions that exhibit a lot of motion. This is further explored by visualising the activation maps from the final attention heads in the network, see Figure 5.29. Comparing the two frames for t_1 and t_2, it is clear that the motion in this sequence occurs in a very specific region, which is picked up by the optical flow intensity projection as well as the activation map.

Attention mechanisms. The respective importance of multi-head self-attention, 3D window attention and channel attention is investigated by training different variants of the model on the same training dataset and testing them with our Medium test set. The results are summarised in Table 5.4. The most significant mechanism according to these results is the multi-head self-attention, which is implemented similarly to SwinIR [113] when 3D window attention is excluded.

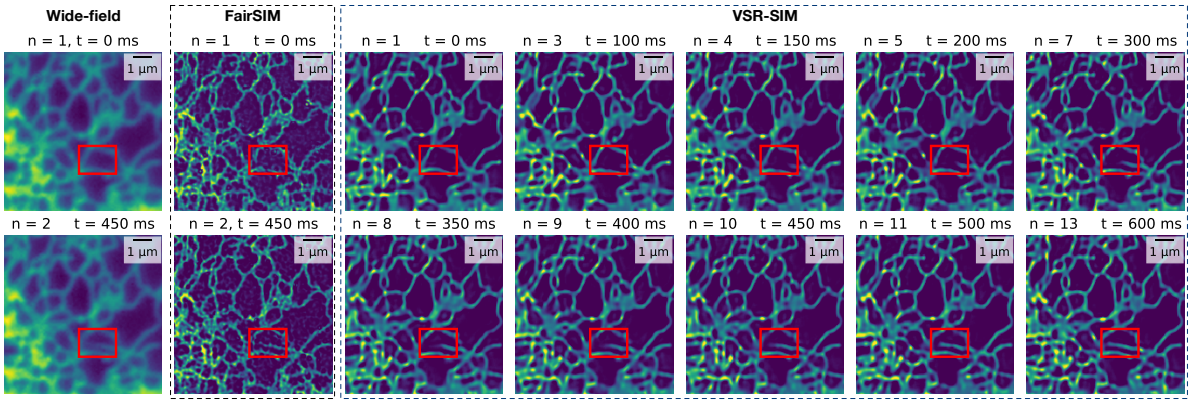


Figure 5.31: The proposed method, VSR-SIM, and the widely used method FairSIM applied to a SIM image sequence of the endoplasmic reticulum. Both methods offer significant improvements over wide-field imaging. The rectangle emphasises a reshaping event of a tubule. Compared with FairSIM, the proposed method achieves 9 times higher temporal resolution by enabling the rolling SIM imaging scheme, see Figure 5.27. The spatial resolution of FairSIM is higher, but also contains more artefacts.

Speed limit of SIM reconstruction

As indicated in Table 5.3, the reconstruction quality of VSR-SIM approaches that of a similarly trained deconvolution method, meaning that the sub-diffraction imaging enabled by SIM becomes increasingly difficult to achieve as the motion increases. Importantly, however, since VSR-SIM is trained on SIM video data spanning multiple motion regimes, the case of extreme motion does not cause the method to collapse and perform significantly worse than the deconvolution baseline. I investigate this ability further by reconstructing inputs that have a variable delay between frames and comparing the results to those of ML-SIM, which has no capability to handle motion. As the input data, samples from a high frame rate video sequence from Reds [149] are used in addition to generated images of moving simulated beads. The results are shown on Figure 5.30. Although the performance decreases as the frame delay increases, the drop is much smaller than for ML-SIM; namely 1 dB versus 6 dB over the range of 0-25 ms frame delay in the case of the video sequence from Reds. In the case of the simulated beads, the performance does not decrease. This indicates that VSR-SIM is able to entirely

ignore the adjacent frames in a SIM stack if the motion is high enough, which presumably becomes easier for the model to do as the spatial separation between the beads increases.

Rolling SIM algorithm

When performing SIM reconstruction with conventional methods, the order of illumination patterns in a stack has to be consistent across stacks. To increase the temporal resolution of SIM, one can use frames that belong to adjacent stacks, thus having a rolling window for which frames are included in the current stack, which reduces the number of frames to be acquired per individual stack. This scheme for SIM imaging is illustrated in Figure 5.27. In the scheme depicted here, the rolling window is shifting by a single frame at a time, therefore increasing the temporal resolution by a factor of 9. To reconstruct SIM frames according to a rolling window, the reconstruction method must be able to handle inputs with varying order of illumination patterns. I address this by shuffling illumination patterns for every training sample that is generated for the training data. The shuffling is without replacement such that a complete cycle is always present in an input. This approach forces the model to learn to handle arbitrary orderings facilitating the use of the rolling SIM scheme. Combined with the motion compensating reconstruction method that can work at motion regimes that traditionally would be unmanageable, imaging at high speed with high granularity becomes possible. This capability lends itself well to applications with fast-moving samples that exhibit intricate movement behaviour. The scheme can similarly be applied for long-term imaging by utilising the higher photon efficiency coming with acquiring only a single frame per reconstructed output.

Improving temporal resolution. To demonstrate the model applied to the rolling SIM scheme, I performed an experiment imaging endoplasmic reticulum in COS-7 cells, labelled with the sec61-mApple and imaged with an excitation wavelength of 561 nm. The FairSIM reconstruction method [146] is used as a baseline as it is widely used in the microscopy community [180]. The endoplasmic reticulum is the largest membrane structure inside the cell and displays drastic reshaping with constant tubule elongation, retraction and junction formation as shown on Figure 5.31. This dynamic reshaping is important to regulate the morphology and function of the ER inside the cell. Compromised reshaping dynamics of the ER are associated with a variety of diseases, including Alzheimer's disease [220], which makes it important to record, measure and understand these dynamics. On Figure 5.31 an occurrence of reshaping can be seen in the area marked by the rectangle over a sequence of 20 frames each acquired with a 50 ms exposure time. Using FairSIM for the reconstruction provides only two super-resolved SIM images, rendering the reshaping event very abrupt and less noticeable.

Using VSR-SIM with the rolling SIM scheme, the raw sequence leads to 19 reconstructed outputs, of which 12 are included, showing a significantly more granular process. FairSIM, however, is seen to recover more high-frequency information in its two outputs indicating that it achieves a higher spatial resolution, although at the expense of more artefacts.

5.3.6 Discussion

I have proposed a new transformer architecture that combines channel attention with multi-head self-attention computed using shifted 3D windows. This architecture is shown to excel at the SIM reconstruction task for dynamic inputs. A demonstration of using the method for a use case in medical research was made with the implementation of rolling SIM imaging, in which a moving window of SIM frames are used for reconstruction providing a temporal resolution that is 9 times higher, while still providing comparable spatial resolution well beyond the diffraction limit. The proposed method can be used for any SIM imaging system as it is purely trained on synthetic data using the proposed image formation model that can be easily adapted to different SIM configurations.

Chapter 6

Discussion

The last decade has seen the emergence of deep learning as a new paradigm for signal processing. The term deep learning does not have a very precise definition, but generally refers to the use of artificial neural networks (ANNs) that are considered deep. *Deep* in this context is a relative denotation as it used to refer to networks with less than 10 layers, e.g. [40], but after the introduction of ResNets [70], deep networks could have hundreds of layers [96] to more than a thousand [114]. Perhaps a better definition, albeit arguably similarly vague, is that deep learning is the pursuit of models that achieve a deep understanding and latent representation of the data they model. Regardless of definition, deep learning can be seen as an advanced branch of machine learning, ML, that is at the frontier of the computational bandwidth and capability of modern information technology. The primary way of advancing the field has been to scale up model complexity and data volume. This is a trend that may not be possible to uphold according to [222], which suggests it outpaces Moore's law [181]. Research in more efficient machine learning using simpler models and less data is a crucial direction for the field if ML methods are to continue improving at the rate they have and not become a technology exclusive to large technology companies and organisations. Regardless of the future scalability, ML has already enabled methods to obtain a high degree of generalisation and become robust to noise and perturbations in input, thus paving the way for automation and improved techniques.

Many advances can be attributed to the push from large technology companies that have explored and in many cases employed deep learning applications to tackle problems that are important to their businesses. Examples include image and video classification for categorising content (e.g. Google Photos), image segmentation for autonomous driving (e.g. Tesla Autopilot), natural language processing for text summarisation (e.g. OpenAI's GPT-3 [21]) and real-time language translation (e.g. Google Translate). There has also been an increasing interest in fundamental research from the industry in areas of reinforcement learning for cognitive reasoning, especially by DeepMind. Examples include the computationally solved unfolding

of misfolded proteins, AlphaFold [89], the complex decision-making in playing games like chess and Go or real-time strategy video games such as Star Craft 2 [189]. A recent example also based on reinforcement learning is the learned magnetic control of tokamak plasmas in fusion reactors [36]. For many of these problems, artificial intelligence (AI) solutions have achieved super-human performance. Facial recognition has been found to be more accurate [195] when using deep neural networks than an average human operator. As for the examples of the cognitive challenge of playing games, AI implementations have been demonstrated to best professionals. The remarkable aspect of these achievements is that they are entirely data driven. No specific set of heuristics or rules have enabled the algorithmic approaches to have the success they have had, nor is it simply due to the raw power of modern computational resources. Brute force methods are simply not feasible for many of these problems. For some of these strategy games, the combinatorics of the game mechanics lead to an exponential increase in computational complexity as a function of turns or units of time. In these cases, rather than building an algorithm that uses sophisticated logical rules, reinforcement learning is used to ”learn by doing”. For instance, AlphaGo [189] uses a system of two models that are playing each other for many iterations until the models obtain the capability of high-level reasoning for how to optimally play the game. There are two schools of thought regarding the fundamental significance of this achievement and similar breakthroughs. On one hand, advocates argue that these advancements are a clear step towards artificial general intelligence (AGI), while a counterargument is that rather than actually being evidence of a deep understanding and, some might say, intuition regarding cognitive problems, the models simply learn to do advanced pattern recognition and provide output that is effectively maximum likelihood projections. This rationale is commonly associated with the Chinese room argument [185] and somewhat touches on the computational-representational understanding of mind (CRUM) hypothesis [196]. However, the topic is of a more philosophical nature as is the case with question of how to define intelligence, and for the purpose of this section, it is only noted that major advances have been made in using AI to empower methods for signal processing, image generation and solving cognitive problems.

The adoption of deep learning in the bioimaging community has been rapid. Many applications address computer vision problems such as automated image classification and improved image processing – some examples are given in Section 2.4.1. In the larger field of biomedical research, many diverse applications of ML have been published ranging from unsupervised learning for clustering analysis of electrophysiological data with spike sorting [46], the aforementioned AlphaFold using reinforcement learning and sequence-based deep learning for prediction of protein-protein interactions [64].

At the time of writing, the largest proportion of applications appear to rely on supervised learning both for the general computer vision literature and for bioimaging specifically. The areas of machine learning besides supervised learning, see Figure 2.4, remain relatively unexplored for microscopy. An exception for bioimaging is self-supervised learning for denoising as described in Section 3.4. However, at the system and acquisition level, there are only a few publications that propose ML approaches such as learned sensing with LED arrays for ptychography [90, 223]. While the notion of a thinking microscope has been proposed conceptually [170], it seems to be an avenue of untapped potential. Given the advances in autonomous driving and using AI for cognitive problems using reinforcement learning, it seems likely that adaptiveness and automation can be achieved through similar technology and transform the way that microscopes are used. It seems plausible that this could lead to automated and accurate optimisation of all acquisition parameters during imaging, thereby enabling both high throughput and long imaging sessions in a robust and versatile way. With that in mind, it is probable that the greatest impact of AI on the field of microscopy is yet to be seen.

It is widely believed that reinforcement learning is the most likely pathway to AGI. If the promise of reinforcement learning holds up, I would expect that it could also generate significant impact on the analysis and signal processing side of microscopy and not only with respect to hardware control and acquisition. In the short term, there is little doubt that current ML approaches to e.g. denoising and SIM reconstruction will see many incremental improvements as neural network architectures become more efficient and computational power scales. Opponents of the idea that signal processing can be much further improved may argue that there is only a certain amount of information available in a given image and state-of-the-art methods are already close to utilising that information to the fullest, thus making any further improvements speculative. In [130], Manton makes the distinction that a signal processing method for SIM can be considered as either adding information or extracting information. This distinction implies that adding information is questionable, essentially guesswork, and if the resultant reconstruction output is not purely based on extracted information, then it is speculative. While it is easy to agree that it is not a good idea to add false features to an image, I do think Manton's distinction is an oversimplification. Indeed, the ability for a method to add information is deeply correlated with the task of extracting information. If a model were trained very specifically on one sample type, it may be capable of extrapolating information from a weak signal by leveraging previously seen data of similar samples. Then it becomes clear that the model will be more robust and accurate with respect to extracting any parameters for e.g. SIM reconstruction. One example is finding the peaks on the Fourier transform of a raw SIM image. If an algorithm to find the peak is only based on thresholding, it would be very prone to offsets due to noise. If on the other hand a model would have learned to

recognise the general shape of the frequency transformed signal taking into account how optical aberrations might perturb the signal, this model may end up with an improved estimate for the peak localisation. Anything the model would do with these more accurately determined parameters would indirectly be due to the fact that the model recognised parallels with previous examples of the same type of sample, and the way the final image would be produced is by adding this information from seemingly nowhere. Therefore, "adding" image features as per Manton's definition does not have to imply that the method is necessarily speculative because the added features could in fact be more statistically accurate estimates. It is however important to acknowledge that in this paradigm there is a conceivable risk of "adding" with the negative connotation, i.e. a method introducing artefacts or "hallucinating" as sometimes used in the literature.

It is also worth noting that the problem of hallucinations can be ameliorated to some extent. Generative models may be optimised in ways that do not increase the likelihood of observed data for instance with an adversarial loss. This means that the model may be free to introduce image features that have a small probability of corresponding to the ground truth driven by a very specific mechanism that enables this "creative freedom". It can lead to images with a high perceptual quality, i.e. visually seeming highly convincing, but the image data is in this case indeed guesswork. In the fully synthetic SIM section, we saw that a model trained with MSE still can end up learning an inaccurate representation of reality if the training data is not realistic, i.e. distortions in the general shape of objects due to the simplistic geometry of the dead leaves model. But I would argue that this rather shows an inability to introduce high-frequency information than it is a case of introducing unseen features.

An important challenge for the future of the deep learning field is scalability. At the current rate, improvements will soon require infeasible amounts of data and model parameters. A way out of this trend will be to focus on higher data efficiency as noted by Andrew Ng [153]. Learning from less data is possible by using transfer learning across modalities and domains [209] and by using more general computational principles such those proposed in zero-shot, single-shot and few-shot approaches e.g. gating modules [190] for deciding if a given sample comes from an unseen class and generative modules to generate feature representations of unseen classes [47]. Another avenue may be to incorporate more analytical modelling, whereby knowledge of nature, laws of physics etc. could be guiding the model learning potentially making huge datasets unnecessary. For this reason, I also believe hybrid approaches in which modelling is used to facilitate simulation-supervised methods, akin to ML-SIM, will be valuable in the future of the field. Hybrid approaches can also overcome some fundamental problems of acquiring ground truth data, while allowing existing data to be more efficiently utilised when coupled to a physical model that represents a fundamental understanding of the world.

Chapter 7

Conclusion

In this thesis, I have proposed and studied a set of image processing methods for computer vision problems related to bioimaging by using deep learning. The common theme of these methods is to make them work in regimes where classical methods are prone to fail. By increasing robustness to noise, the fundamental trade-off of in optical microscopy between image quality, imaging speed and imaging duration can be dealt with more favourably.

Below, I summarise these methods and the respective findings in using them. In addition to the proposed methods outlined below, I have built and tested denoising models using the Noise2void training strategy with CNN architectures. These models have served multiple different use cases, namely as a preprocessing step to kymograph analysis and calcium imaging in optical microscopy; cryogenic electron tomography; and photometry in astronomy. Based on these tests, I have found that quantitative analysis can often benefit from denoising when performed as a preprocessing step, and the self-supervised Noise2void training strategy can provide high denoising performance in the absence of ground truth.

Firstly, a supervised denoising method was implemented, which was trained by generating training data with synthetic noise sources that matches experimental data. A benchmarking histology dataset acquired using bright-field microscopy, PCam [199], was used as ground truth data and the inputs were then synthesised. The performance of the trained model was used to get an indication of the potential gains of applying deep learning to fluorescence microscopy. Despite the difference in modality, a direct application of the trained model to experimental fluorescence microscopy data was attempted. While the cross-modality performance of the denoiser was not at a level where it would be useful for general application, the transfer learning aspect of this training and inference approach showed promise.

A set of segmentation models for processing images of the endoplasmic reticulum (ER) have been proposed. Supervised learning was used for all of them and different training strategies were explored. As a first iteration, a model was trained using training data produced by applying

intensity thresholding to images with high signal-to-noise ratio (SNR). This provided consistent ground truth data which could then be transformed with an image degradation model to give inputs. This degradation process consists of a non-uniform brightness modulation and the introduction of significant levels of noise modelled as both Gaussian and Poisson noise. The resulting method was found to qualitatively outperform the ImageJ plugin Weka when evaluated across multiple acquisitions of the same sample, which indicates that the neural network has a superior ability to generalise than the simpler models in Weka based on random forests by default. To push the model towards cleaner segmentation maps, a supervised segmentation model was built using experimentally acquired training inputs and manually annotated targets. This provided a consistent and generalised model used in the publication [127]. As a further improvement, the segmentation model was made capable of utilising temporal information by using a transformer network with an architecture that supports spatio-temporal connections in the input. This has provided a method that yields higher quality segmentation maps for temporal inputs and the work is published in [126].

Arguably, the most significant scientific contribution of this thesis is the machine learning-based reconstruction method for structured illumination microscopy (SIM), ML-SIM. The method relies on a simulation-supervised training strategy, in which the image formation in SIM is modelled, thus providing ideal ground truths. An important finding is that if the training data is diverse and the simulated optical parameters are similar to those used in a real system, a neural network can be trained purely on synthesised data but still performs well on experimentally acquired data. This allows the method to address any configuration of SIM and to account for any relevant aberration provided that the image formation process can be forward modelled. The method was published in [30] and demonstrated high-quality reconstruction results across two fundamentally different SIM systems with the same trained model. The richness of the source image dataset was also found to be of importance. As seen in this thesis, well-performing models were obtained when trained on the image dataset DIV2K, which features several hundreds diverse, high-resolution photographs of various objects and organisms. However, when fully synthesised image data was used for training by using the simplistic dead leaves model, the reconstruction outputs suffered in quality with clear artefacts emerging.

An important aspect that the first ML-SIM model did not account for was motion. Since SIM is often used for live-cell imaging, a sequence of the commonly used 9 frames for SIM can easily exhibit a significant level of motion. As a remedy, an extended version of ML-SIM has been proposed, which uses ideas from the problem video super-resolution (VSR) in the computer vision literature. This extended method, VSR-SIM, has a vision transformer architecture and is trained by modelling SIM image formation on sampled video data enabling

the learning of spatio-temporal frame alignment. This makes consistent reconstruction output obtainable in a range of motion regimes from completely still to highly dynamic image stacks.

Future avenues of work that would be of interest include self-supervised training strategies and reinforcement learning for segmentation and SIM reconstruction. Given the performance of the self-supervised denoising models as reported in this thesis, this seems to be a promising direction for other problems in image reconstruction for bioimaging. Another important direction to explore is the incorporation of 3D spatial information, (x, y, z), in the proposed methods for denoising, segmentation and SIM reconstruction. The models used for segmentation and SIM reconstruction in this thesis have been shown to be extendable to 3D spatio-temporal data (x, y, t) and clearly benefit from the inclusion of the temporal dimension. The axial dimension may present similar improvements due to the correlated depth information, and enabling 3D SIM reconstruction output with an approach based on ML-SIM and VSR-SIM could provide an important tool for biomedical research.

References

[1] Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S. (2016). Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*.

[2] Agustsson, E. and Timofte, R. (2017). NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. Technical report.

[3] Ayuk, R., Giovannini, H., Jost, A., Mudry, E., Girard, J., Mangeat, T., Sandeau, N., Heintzmann, R., Wicker, K., Belkebir, K., and Others (2013). Structured illumination fluorescence microscopy with distorted excitations using a filtered blind-SIM algorithm. *Optics letters*, 38(22):4723–4726.

[4] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation.

[5] Ball, G., Demmerle, J., Kaufmann, R., Davis, I., Dobbie, I. M., and Schermelleh, L. (2015a). SIMcheck: a Toolbox for Successful Super-resolution Structured Illumination Microscopy. *Scientific Reports*, 5(1):15915.

[6] Ball, G., Demmerle, J., Kaufmann, R., Davis, I., Dobbie, I. M., and Schermelleh, L. (2015b). SIMcheck: a toolbox for successful super-resolution structured illumination microscopy. *Scientific reports*, 5(1):1–12.

[7] Bannykh, S. I. and Balch, W. E. (1997). Membrane dynamics at the endoplasmic reticulum–Golgi interface. *The Journal of cell biology*, 138(1):1–4.

[8] Batson, J. and Royer, L. (2019). Noise2Self: Blind denoising by self-supervision. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:826–835.

[9] Betzig, E. (1995). Proposed method for molecular optical imaging. *Optics Letters*, 20(3):237.

[10] Bhat, G., Danelljan, M., Van Gool, L., and Timofte, R. (2021). Deep Burst Super-Resolution. *arXiv preprint arXiv:2101.10997*.

[11] Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

[12] Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., and Zelnik-Manor, L. (2018). 2018 PIRM Challenge on Perceptual Image Super-resolution. Technical report.

[13] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

[14] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pages 177–186. Springer.

[15] Boulanger, J., Kervrann, C., Bouthemy, P., Elbau, P., Sibarita, J.-B., and Salamero, J. (2009). Patch-based nonlocal functional for denoising fluorescence microscopy image sequences. *IEEE transactions on medical imaging*, 29(2):442–454.

[16] Boyd, N., Jonas, E., Babcock, H. P., and Recht, B. (2018). DeepLoco: Fast 3D Localization Microscopy Using Neural Networks. *bioRxiv preprint bioRxiv:10.1101/267096*.

[17] Bracewell, R. N. and Bracewell, R. N. (1986). *The Fourier transform and its applications*, volume 31999. McGraw-hill New York.

[18] Bradley, L., Sipocz, B., Robitaille, T., Tollerud, E., Deil, C., Vinícius, Z., Barbary, K., Günther, H. M., Bostroem, A., Droettboom, M., et al. (2016). Photutils: Photometry tools. *Astrophysics Source Code Library*, pages ascl–1609.

[19] Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

[20] Brewster, D. and Bache, A. D. (1833). *A Treatise on Optics...: First American Edition, with an Appendix, Containing an Elementary View of the Application of Analysis to Reflexion and Refraction*. Carey, Lea, & Blanchard.

[21] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[22] Brüel-Gabrielsson, R., Nelson, B. J., Dwaraknath, A., Skraba, P., Guibas, L. J., and Carlsson, G. (2019). A Topology Layer for Machine Learning. 108.

[23] Buades, A., Coll, B., and Morel, J.-M. J.-M. (2005). A non-local algorithm for image denoising. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2(0):60–65 vol. 2.

[24] Cao, R., Chen, Y., Liu, W., Zhu, D., Kuang, C., Xu, Y., and Liu, X. (2018). Inverse matrix based phase estimation algorithm for structured illumination microscopy. *Biomedical Optics Express*, 9(10):5037.

[25] Chapelle, O., Scholkopf, B., and Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

[26] Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., and Gao, W. (2020). Pre-Trained Image Processing Transformer.

[27] Choi, M., Kim, H., Han, B., Xu, N., and Lee, K. M. (2020). Channel attention is all you need for video frame interpolation. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pages 10663–10671.

[28] Christensen, C. N., Lu, M., Ward, E. N., Lio, P., and Kaminski, C. F. (2022). Spatio-temporal vision transformer for super-resolution microscopy. *arXiv preprint arXiv:2203.00030*.

[29] Christensen, C. N., Ward, E. N., Lio, P., and Kaminski, C. F. (2020). ML-SIM: A deep neural network for reconstruction of structured illumination microscopy images. *arXiv preprint arXiv:2003.11064*, pages 1–9.

[30] Christensen, C. N., Ward, E. N., Lu, M., Lio, P., and Kaminski, C. F. (2021). ML-SIM: universal reconstruction of structured illumination microscopy images using transfer learning. *Biomedical Optics Express*, 12(5):2720.

[31] Ciresan, D., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851.

[32] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20:273–297.

[33] Coupé, P., Yger, P., Prima, S., Hellier, P., Kervrann, C., and Barillot, C. (2008). An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images. *IEEE transactions on medical imaging*, 27(4):425–441.

[34] Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.

[35] Culley, S., Albrecht, D., Jacobs, C., Pereira, P. M., Leterrier, C., Mercer, J., and Henriques, R. (2018). Quantitative mapping and minimization of super-resolution optical imaging artifacts. *Nature methods*, 15(4):263–266.

[36] Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de Las Casas, D., et al. (2022). Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419.

[37] Deledalle, C.-A., Tupin, F., and Denis, L. (2010). Poisson nl means: Unsupervised non local means for poisson noise. In *2010 IEEE international conference on image processing*, pages 801–804. IEEE.

[38] Demmerle, J., Innocent, C., North, A. J., Ball, G., Müller, M., Miron, E., Matsuda, A., Dobbie, I. M., Markaki, Y., and Schermelleh, L. (2017). Strategic and practical guidelines for successful structured illumination microscopy. *Nature Protocols*, 12(5):988–1010.

[39] Dey, N., Blanc-Féraud, L., Zimmer, C., Kam, Z., Olivo-Marin, J.-C., and Zerubia, J. (2004). A deconvolution method for confocal microscopy with total variation regularization. In *2004 2nd IEEE International Symposium on Biomedical Imaging: Nano to Macro (IEEE Cat No. 04EX821)*, pages 1223–1226. IEEE.

[40] Dong, C., Loy, C. C., He, K., and Tang, X. (2014). Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307.

[41] Donoho, D. L. (1995). De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627.

[42] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

[43] Duchon, C. E. (1979). Lanczos Filtering in One and Two Dimensions. *Journal of Applied Meteorology*, 18(8):1016–1022.

[44] Edelstein, A., Amodaj, N., Hoover, K., Vale, R., and Stuurman, N. (2010). Computer control of microscopes using μmanager. *Current protocols in molecular biology*, 92(1):14–20.

[45] Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745.

[46] Eom, J., Park, I. Y., Kim, S., Jang, H., Park, S., Huh, Y., and Hwang, D. (2021). Deep-learned spike representations and sorting via an ensemble of auto-encoders. *Neural Networks*, 134:131–142.

[47] Felix, R., Reid, I., Carneiro, G., et al. (2018). Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37.

[48] Fiolka, R., Beck, M., and Stemmer, A. (2008). Structured illumination in total internal reflection fluorescence microscopy using a spatial light modulator. *Optics letters*, 33(14):1629–1631.

[49] Fish, D., Brinicombe, A., Pike, E., and Walker, J. (1995). Blind deconvolution by means of the richardson–lucy algorithm. *JOSA A*, 12(1):58–65.

[50] Foster, H. (2021). *Visualisation of dynein complexes in vitro and inside cells*. PhD thesis, University of Cambridge.

[51] Fowles, G. R. (1989). *Introduction to modern optics*. Courier Corporation.

[52] Goodfellow, I. (2016). Generative Adversarial Network. In *NIPS*.

[53] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*.

[54] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

[55] Goodman, J. W. (1976). Some fundamental properties of speckle. *JOSA*, 66(11):1145–1150.

[56] Göttfert, F., Wurm, C. A., Mueller, V., Berning, S., Cordes, V. C., Honigmann, A., and Hell, S. W. (2013). Coaligned dual-channel sted nanoscopy and molecular diffusion analysis at 20 nm resolution. *Biophysical journal*, 105(1):L01–L03.

[57] Grewenig, S., Zimmer, S., and Weickert, J. (2011). Rotationally invariant similarity measures for nonlocal image denoising. *Journal of Visual Communication and Image Representation*, 22(2):117–130.

[58] Grimm, S. L., Demory, B.-O., Gillon, M., Dorn, C., Agol, E., Burdanov, A., Delrez, L., Sestovic, M., Triaud, A. H., Turbet, M., et al. (2018). The nature of the trappist-1 exoplanets. *Astronomy & Astrophysics*, 613:A68.

[59] Gustafsson, M. G. (2000). Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *Journal of Microscopy*, 198(2):82–87.

[60] Gustafsson, M. G. L., Shao, L., Carlton, P. M., Wang, C. J. R., Golubovskaya, I. N., Cande, W. Z., Agard, D. A., and Sedat, J. W. (2008). Three-dimensional resolution doubling in wide-field fluorescence microscopy by structured illumination. *BIOPHYSICAL JOURNAL*, 94(12):4957–4970.

[61] Gwosch, K. C., Pape, J. K., Balzarotti, F., Hoess, P., Ellenberg, J., Ries, J., and Hell, S. W. (2020). Minflux nanoscopy delivers 3d multicolor nanometer resolution in cells. *Nature methods*, 17(2):217–224.

[62] Haarnoja, T., Ha, S., Zhou, A., Tan, J., Tucker, G., and Levine, S. (2018). Learning to walk via deep reinforcement learning. *arXiv preprint arXiv:1812.11103*.

[63] Haris, M., Shakhnarovich, G., and Ukita, N. (2019). Recurrent back-projection network for video super-resolution. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:3892–3901.

[64] Hashemifar, S., Neyshabur, B., Khan, A. A., and Xu, J. (2018). Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics*, 34(17):i802–i810.

[65] Hasinoff, S. W., Sharlet, D., Geiss, R., Adams, A., Barron, J. T., Kainz, F., Chen, J., and Levoy, M. (2016). Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (ToG)*, 35(6):1–12.

[66] Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

[67] Hauser, M., Wojcik, M., Kim, D., Mahmoudi, M., Li, W., and Xu, K. (2017). Correlative Super-Resolution Microscopy: New Dimensions and New Opportunities. *Chemical Reviews*, 117(11):7428–7456.

[68] Haykin, S. S., Widrow, B., and Widrow, B. (2003). *Least-Mean-Square Adaptive Filters*, volume 31. Wiley Online Library.

[69] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

[70] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:770–778.

[71] Heilemann, M., Van De Linde, S., Schüttpelz, M., Kasper, R., Seefeldt, B., Mukherjee, A., Tinnefeld, P., and Sauer, M. (2008). Subdiffraction-resolution fluorescence imaging with conventional fluorescent probes. *Angewandte Chemie International Edition*, 47(33):6172–6176.

[72] Heintzmann, R. and Cremer, C. G. (1999). Laterally modulated excitation microscopy: improvement of resolution by using a diffraction grating. In *Optical Biopsies and Microscopic Techniques III*, volume 3568, pages 185–196. SPIE.

[73] Heintzmann, R., Jovin, T. M., and Cremer, C. (2002). Saturated patterned excitation microscopy—a concept for optical resolution improvement. *JOSA A*, 19(8):1599–1609.

[74] Hell, S. W. (2003). Toward fluorescence nanoscopy. *Nature biotechnology*, 21(11):1347–1355.

[75] Hell, S. W. and Wichmann, J. (1994). Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy. *Optics Letters*, 19(11):780.

[76] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

[77] Hinton, G. E. and Zemel, R. (1993). Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6.

[78] Holcman, D., Parutto, P., Chambers, J. E., Fantham, M., Young, L. J., Marciniak, S. J., Kaminski, C. F., Ron, D., and Avezov, E. (2018). Single particle trajectories reveal active endoplasmic reticulum luminal flow. *Nature cell biology*, 20(10):1118.

[79] Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence*, 17(1-3):185–203.

[80] Huang, X., Fan, J., Li, L., Liu, H., Wu, R., Wu, Y., Wei, L., Mao, H., Lal, A., Xi, P., and Others (2018a). Fast, long-term, super-resolution imaging with Hessian structured illumination microscopy. *Nature biotechnology*, 36(5):451–459.

[81] Huang, X., Fan, J., Li, L., Liu, H., Wu, R., Wu, Y., Wei, L., Mao, H., Lal, A., Xi, P., Tang, L., Zhang, Y., Liu, Y., Tan, S., and Chen, L. (2018b). Fast, long-term, super-resolution imaging with Hessian structured illumination microscopy. *Nature Biotechnology*, 36(5):451–459.

[82] Ij, H. (2018). Statistics versus machine learning. *Nat Methods*, 15(4):233.

[83] Iqbal, K., Liu, F., Gong, C.-X., and Grundke-Iqbal, I. (2010). Tau in Alzheimer disease and related tauopathies. *Current Alzheimer Research*, 7(8):656–664.

[84] Jakobs, M. A., Dimitracopoulos, A., and Franze, K. (2019). Kymobutler, a deep learning software for automated kymograph analysis. *Elife*, 8:e42288.

[85] Jezierska, A., Talbot, H., Chaux, C., Pesquet, J.-C., and Engler, G. (2012). Poisson-Gaussian noise parameter estimation in fluorescence microscopy imaging. In *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1663–1666. IEEE.

[86] Jin, L., Liu, B., Zhao, F., Hahn, S., Dong, B., Song, R., Elston, T. C., Xu, Y., and Hahn, K. M. (2020a). Deep learning enables structured illumination microscopy with low light levels and enhanced speed. *Nature communications*, 11(1):1–7.

[87] Jin, L., Liu, B., Zhao, F., Hahn, S., Dong, B., Song, R., Elston, T. C., Xu, Y., and Hahn, K. M. (2020b). Deep learning enables structured illumination microscopy with low light levels and enhanced speed. *Nature Communications*, 11(1).

[88] Jo, Y., Oh, S. W., Kang, J., and Kim, S. J. (2018). Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3224–3232.

[89] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

[90] Kappeler, A., Ghosh, S., Holloway, J., Cossairt, O., and Katsaggelos, A. (2017). PtychNet: CNN based Fourier ptychography. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1712–1716. IEEE.

[91] Kappeler, A., Yoo, S., Dai, Q., and Katsaggelos, A. K. (2016). Video Super-Resolution With Convolutional Neural Networks. *IEEE Transactions on Computational Imaging*, 2(2):109–122.

[92] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., and Others (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

[93] Kervrann, C. and Trubuil, A. (2004). An adaptive window approach for poisson noise reduction and structure preserving in confocal microscopy. In *2004 2nd IEEE International Symposium on Biomedical Imaging: Nano to Macro (IEEE Cat No. 04EX821)*, pages 788–791. IEEE.

[94] Keys, R. (1981). Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6):1153–1160.

[95] Khater, I. M., Nabi, I. R., and Hamarneh, G. (2020). A review of super-resolution single-molecule localization microscopy cluster analysis and quantification methods. *Patterns*, 1(3):100038.

[96] Kim, J., Lee, J. K., and Lee, K. M. (2015). Accurate Image Super-Resolution Using Very Deep Convolutional Networks. Technical report.

[97] Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. Technical report.

[98] Kino, G. S. and Corle, T. R. (1996). *Confocal scanning optical microscopy and related imaging systems*. Academic Press.

[99] Ko, D. C., Gordon, M. D., Jin, J. Y., and Scott, M. P. (2001). Dynamic movements of organelles containing niemann-pick c1 protein: Npc1 involvement in late endocytic events. *Molecular biology of the cell*, 12(3):601–614.

[100] Kraus, O. Z., Ba, J. L., and Frey, B. J. (2016). Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, 32(12):i52—-i59.

[101] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

[102] Krull, A., Buchholz, T.-O., and Jug, F. (2019). Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2129–2137.

[103] Lal, A., Shan, C., and Xi, P. (2016). Structured illumination microscopy image reconstruction algorithm. *IEEE Journal on Selected Topics in Quantum Electronics*, 22(4):1–15.

[104] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[105] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:105–114.

[106] Lee, A. B., Mumford, D., and Huang, J. (2001). Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. *International Journal of Computer Vision*, 41(1-2):35–59.

[107] Lee, T.-C., Kashyap, R. L., and Chu, C.-N. (1994). Building skeleton models via 3-d medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478.

[108] Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., and Aila, T. (2018). Noise2Noise: Learning Image Restoration without Clean Data.

[109] Lempitsky, V., Vedaldi, A., and Ulyanov, D. (2018). Deep Image Prior. Technical report.

[110] Li, D., Shao, L., Chen, B.-C., Zhang, X., Zhang, M., Moses, B., Milkie, D. E., Beach, J. R., Hammer, J. A., Pasham, M., et al. (2015). Extended-resolution structured illumination imaging of endocytic and cytoskeletal dynamics. *Science*, 349(6251):aab3500.

[111] Li, X., Hu, Y., Gao, X., Tao, D., and Ning, B. (2010). A multi-frame image super-resolution method. *Signal Processing*, 90(2):405–414.

[112] Li, Y. (2017). Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.

[113] Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., and Timofte, R. (2021). SwinIR: Image Restoration Using Swin Transformer.

[114] Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M. (2017). Enhanced Deep Residual Networks for Single Image Super-Resolution. Technical report.

[115] Lin, T., Wang, Y., Liu, X., and Qiu, X. (2021). A Survey of Transformers. *arXiv preprint arXiv:2106.04554*, 1(1).

[116] Lindeberg, T. (1994). Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270.

[117] Lindenbaum, M., Fischer, M., and Bruckstein, A. (1994). On Gabor's contribution to image enhancement. *Pattern Recognition*, 27(1):1–8.

[118] Ling, C., Zhang, C., Wang, M., Meng, F., Du, L., and Yuan, X. (2020). Fast structured illumination microscopy via deep learning. *Photonics Research*, 8(8):1350–1359.

[119] Liu, H., Ruan, Z., Zhao, P., Dong, C., Shang, F., Liu, Y., and Yang, L. (2020). Video Super Resolution Based on Deep Learning: A Comprehensive Survey.

[120] Liu, W., Liu, Q., Zhang, Z., Han, Y., Kuang, C., Xu, L., Yang, H., and Liu, X. (2019). Three-dimensional super-resolution imaging of live whole cells using galvanometer-based structured illumination microscopy. *Optics express*, 27(5):7237–7248.

[121] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021a). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows.

[122] Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., and Hu, H. (2021b). Video Swin Transformer. pages 1–12.

[123] Lormand, C., Zellmer, G. F., Németh, K., Kilgour, G., Mead, S., Palmer, A. S., Sakamoto, N., Yurimoto, H., and Moebis, A. (2018). Weka trainable segmentation plugin in imagej: A semi-automatic tool applied to crystal size distributions of microlites in volcanic rocks. *Microscopy and Microanalysis*, 24(6):667–675.

[124] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[125] Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., and Zhang, G. (2015). Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems*, 80:14–23.

[126] Lu, M., Christensen, C. N., Weber, J. M., Konno, T., Läubli, N. F., Scherer, K. M., Avezov, E., Lio, P., Lapkin, A. A., Kaminski Schierle, G. S., and Kaminski, C. F. (2022). Ernet: a tool for the semantic segmentation and quantitative analysis of endoplasmic reticulum topology for video-rate super-resolution imaging. *bioRxiv*.

[127] Lu, M., Van Tartwijk, F. W., Lin, J. Q., Nijenhuis, W., Parutto, P., Fantham, M., Christensen, C. N., Avezov, E., Holt, C. E., Tunnacliffe, A., Holcman, D., Kapitein, L., Kaminski Schierle, G. S., and Kaminski, C. F. (2020). The structure and global distribution of the endoplasmic reticulum network are actively regulated by lysosomes. *Science Advances*.

[128] Lugmayr, A., Danelljan, M., and Timofte, R. (2019). Unsupervised learning for real-world super-resolution. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3408–3416. IEEE.

[129] Ma, C., Yang, C. Y., Yang, X., and Yang, M. H. (2017). Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding*, 158:1–16.

[130] Manton, J. D. (2022). Answering some questions about structured illumination microscopy. *Philosophical Transactions of the Royal Society A*, 380(2220):20210109.

[131] Manton, J. D., Ströhl, F., Fiolka, R., Kaminski, C. F., and Rees, E. J. (2020). Concepts for structured illumination microscopy with extended axial resolution through mirrored illumination. *Biomedical Optics Express*, 11(4):2098–2108.

[132] Mao, X.-J., Shen, C., and Yang, Y.-B. (2016). Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections. Technical report.

[133] Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217.

[134] Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423.

[135] Masters, B. R. (2020). *Superresolution Optical Microscopy: The Quest for Enhanced Resolution and Contrast*. Springer.

[136] McAbee, D. D. and Weigel, P. (1987). Atp depletion causes a reversible redistribution and inactivation of a subpopulation of galactosyl receptors in isolated rat hepatocytes. *Journal of Biological Chemistry*, 262(5):1942–1945.

[137] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133.

[138] Mccutchen, C. W. (1967). Superresolution in microscopy and the Abbe resolution limit. *JOSA*, 57(10):1190–1192.

[139] Merritt, J. E., Armstrong, W., Benham, C., Hallam, T., Jacob, R., Jaxa-Chamiec, A., Leigh, B., McCarthy, S., Moores, K., and Rink, T. (1990). Sk&f 96365, a novel inhibitor of receptor-mediated calcium entry. *Biochemical Journal*, 271(2):515–522.

[140] Mertz, J. (2019). *Introduction to optical microscopy*. Cambridge University Press.

[141] Mickoleit, M., Schmid, B., Weber, M., Fahrbach, F. O., Hombach, S., Reischauer, S., and Huisken, J. (2014). High-resolution reconstruction of the beating zebrafish heart. *Nature methods*, 11(9):919.

[142] Mittal, A., Soundararajan, R., and Bovik, A. C. (2013). Making a 'completely blind' image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212.

[143] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[144] Moerner, W. E. and Kador, L. (1989). Optical detection and spectroscopy of single molecules in a solid. *Physical Review Letters*, 62(21):2535–2538.

[145] Mudry, E., Belkebir, K., Girard, J., Savatier, J., Moal, E. L., Nicoletti, C., Allain, M., Sentenac, A., Le Moal, E., Nicoletti, C., Allain, M., and Sentenac, A. (2012). Structured illumination microscopy using unknown speckle patterns. 6:312–315.

[146] Müller, M., Mönkemöller, V., Hennig, S., Hübner, W., and Huser, T. (2016). Open-source image reconstruction of super-resolution structured illumination microscopy data in ImageJ. *Nature Communications*, 7:10980.

[147] Murray, C., Delrez, L., Pedersen, P., Queloz, D., Gillon, M., Burdanov, A., Ducrot, E., Garcia, L., Lienhard, F., Demory, B.-O., et al. (2020). Photometry and performance of speculoos-south. *Monthly Notices of the Royal Astronomical Society*, 495(2):2446–2457.

[148] Nadeem, S. and Maraj, E. N. (2014). The mathematical analysis for peristaltic flow of nano fluid in a curved channel with compliant walls. *Applied Nanoscience*, 4(1):85–92.

[149] Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., and Lee, K. M. (2019). NTIRE 2019 Challenge on Video Deblurring and Super-Resolution: Dataset and Study. In *CVPR Workshops*.

[150] Nehme, E., Weiss, L. E., Michaeli, T., and Shechtman, Y. (2018). Deep-STORM: super-resolution single-molecule microscopy by deep learning. *Optica*, 5(4):458.

[151] Neil, M., Juškaitis, R., and Wilson, T. (1998). Real time 3D fluorescence microscopy by two beam interference illumination. *Optics Communications*, 153(1-3):1–4.

[152] Newman, M. E. (2002). Assortative mixing in networks. *Physical review letters*, 89(20):208701.

[153] Ng, A. (2022). Unbiggen ai. *IEEE Spectrum.(https://spectrum.ieee.org/andrew-ng-data-centric-ai)*.

[154] Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine learning*, 39:103–134.

[155] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.

[156] Ounkomol, C., Seshamani, S., Maleckar, M. M., Collman, F., and Johnson, G. R. (2018). Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. *Nature methods*, 15(11):917.

[157] Ouyang, W., Aristov, A., Lelek, M., Hao, X., and Zimmer, C. (2018). Deep learning massively accelerates super-resolution localization microscopy. *Nature Biotechnology*, 36(5):460–468.

[158] Pan, Q., Zhang, L., Dai, G., and Zhang, H. (1999). Two denoising methods by wavelet transform. *IEEE transactions on signal processing*, 47(12):3401–3406.

[159] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

[160] Park, S. J., Son, H., Cho, S., Hong, K. S., and Lee, S. (2018). SRFeat: Single Image Super-Resolution with Feature Discrimination. Technical report.

[161] Peixoto, T. P. (2014). The graph-tool python library. *figshare*.

[162] Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639.

[163] Planchon, T. A., Gao, L., Milkie, D. E., Davidson, M. W., Galbraith, J. A., Galbraith, C. G., and Betzig, E. (2011). Rapid three-dimensional isotropic imaging of living cells using Bessel beam plane illumination. *Nature methods*, 8(5):417.

[164] Plaziac, N. (1999). Image interpolation using neural networks. *IEEE transactions on image processing*, 8(11):1647–1651.

[165] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press.

[166] Protter, M., Elad, M., Takeda, H., and Milanfar, P. (2008). Generalizing the nonlocal-means to super-resolution reconstruction. *IEEE Transactions on image processing*, 18(1):36–51.

[167] Qian, J., Lei, M., Dan, D., Yao, B., Zhou, X., Yang, Y., Yan, S., Min, J., and Yu, X. (2015). Full-color structured illumination optical sectioning microscopy. *Scientific Reports*, 5(1):14513.

[168] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

[169] Rego, E. H., Shao, L., Macklin, J. J., Winoto, L., Johansson, G. A., Kamps-Hughes, N., Davidson, M. W., and Gustafsson, M. G. (2012). Nonlinear structured-illumination microscopy with a photoswitchable protein reveals cellular structures at 50-nm resolution. *Proceedings of the National Academy of Sciences*, 109(3):E135–E143.

[170] Rivenson, Y. and Ozcan, A. (2018). Toward a thinking microscope. *Optics and Photonics News*, 29(7):34–41.

[171] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

[172] Robbins, M., Christensen, C. N., Kaminski, C. F., and Zlatic, M. (2021). Calcium imaging analysis–how far have we come? *F1000Research*, 10.

[173] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv preprint arXiv:1505.04597*.

[174] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[175] Royer, L. A., Lemon, W. C., Chhetri, R. K., Wan, Y., Coleman, M., Myers, E. W., and Keller, P. J. (2016). Adaptive light-sheet microscopy for long-term, high-resolution imaging in living organisms. *Nature biotechnology*, 34(12):1267.

[176] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

[177] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252.

[178] Rust, M. J., Bates, M., and Zhuang, X. (2006). Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (storm). *Nature methods*, 3(10):793–796.

[179] Sajjadi, M. S., Scholkopf, B., and Hirsch, M. (2017). EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis. Technical report.

[180] Samanta, K. and Joseph, J. (2021). An overview of structured illumination microscopy: recent advances and perspectives. *Journal of Optics*.

[181] Schaller, R. R. (1997). Moore's law: past, present and future. *IEEE spectrum*, 34(6):52–59.

[182] Schermelleh, L., Heintzmann, R., and Leonhardt, H. (2010a). A guide to super-resolution fluorescence microscopy. *Journal of Cell Biology*, 190(2):165–175.

[183] Schermelleh, L., Heintzmann, R., and Leonhardt, H. (2010b). A guide to super-resolution fluorescence microscopy. *Journal of Cell Biology*, 190(2):165–175.

[184] Schnitzbauer, J., Strauss, M. T., Schlichthaerle, T., Schueder, F., and Jungmann, R. (2017). Super-resolution microscopy with dna-paint. *Nature protocols*, 12(6):1198–1228.

[185] Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and brain sciences*, 3(3):417–424.

[186] Shao, L., Member, S., Yan, R., Li, X., and Liu, Y. (2014). From Heuristic Optimization to Dictionary Learning : A Review and Comprehensive Comparison of Image Denoising Algorithms. *IEEE Transactions on Cybernetics*, 44(7):1001–1013.

[187] Sheppard, C. J. R. (1988). Super-resolution in confocal imaging. *Optik*, 80(2):53–54.

[188] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. (2016). Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. *arXiv preprint arXiv:1609.05158*.

[189] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

[190] Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. (2013). Zero-shot learning through cross-modal transfer. *Advances in neural information processing systems*, 26.

[191] Sonka, M., Hlavac, V., and Boyle, R. (2014). *Image processing, analysis, and machine vision*. Cengage Learning.

[192] Ströhl, F. and Kaminski, C. F. (2015). A joint richardson—lucy deconvolution algorithm for the reconstruction of multifocal structured illumination microscopy data. *Methods and Applications in Fluorescence*, 3(1):014002.

[193] Ströhl, F. and Kaminski, C. F. (2016). Frontiers in structured illumination microscopy. *Optica*, 3(6):667.

[194] Ströhl, F. and Kaminski, C. F. (2017). Speed limits of structured illumination microscopy. *Optics Letters*, 42(13):2511.

[195] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708.

[196] Thagard, P. (2005). *Mind: Introduction to cognitive science*. MIT press.

[197] Vasu, S., Madam, N. T., and N, R. A. (2018). Analyzing Perception-Distortion Tradeoff using Enhanced Perceptual Super-resolution Network. Technical report.

[198] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):5999–6009.

[199] Veeling, B. S., Linmans, J., Winkens, J., Cohen, T., and Welling, M. (2018). Rotation equivariant CNNs for digital pathology. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11071 LNCS:210–218.

[200] Vicidomini, G., Bianchini, P., and Diaspro, A. (2018). Sted super-resolved microscopy. *Nature methods*, 15(3):173–182.

[201] Vojtekova, A., Lieu, M., Valtchanov, I., Altieri, B., Old, L., Chen, Q., and Hroch, F. (2021). Learning to denoise astronomical images with u-nets. *Monthly Notices of the Royal Astronomical Society*, 503(3):3204–3215.

[202] Wang, H., Rivenson, Y., Jin, Y., Wei, Z., Gao, R., Günaydin, H., Bentolila, L. A., Kural, C., and Ozcan, A. (2019). Deep learning enables cross-modality super-resolution in fluorescence microscopy. *Nature methods*, 16(1):103–110.

[203] Wang, X., Yu, K., Chan, K. C. K., Dong, C., and Loy, C. C. (2020). BasicSR: Open Source Image and Video Restoration Toolbox. https://github.com/xinntao/BasicSR.

[204] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Loy, C. C., Qiao, Y., and Tang, X. (2018). ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. Technical report.

[205] Wang, Z., Bovik, A. C., and Sheikh, H. R. (2004). Image quality assessment: From error measurement to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600 – 612.

[206] Ward, E. N., Torkelsen, F. H., and Pal, R. (2018). Enhancing multi-spot structured illumination microscopy with fluorescence difference. *Royal Society open science*, 5(3):171336.

[207] Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S., and Others (2018a). Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nature methods*, 15(12):1090–1097.

[208] Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S., Rocha-Martins, M., Segovia-Miranda, F., Norden, C., Henriques, R., Zerial, M., Solimena, M., Rink, J., Tomancak, P., Royer, L., Jug, F., and Myers, E. W. (2018b). Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nature Methods*, 15(12):1090–1097.

[209] Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):1–40.

[210] Werbos, P. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences. *PhD thesis, Committee on Applied Mathematics, Harvard University, Cambridge, MA*.

[211] Westphal, V., Rizzoli, S. O., Lauterbach, M. A., Kamin, D., Jahn, R., and Hell, S. W. (2008). Video-rate far-field optical nanoscopy dissects synaptic vesicle movement. *Science*, 320(5873):246–249.

[212] Westrate, L., Lee, J., Prinz, W., and Voeltz, G. (2015). Form follows function: the importance of endoplasmic reticulum shape. *Annual review of biochemistry*, 84:791–811.

[213] White, R. L. (1992). Restoration of images and spectra from the hubble space telescope. In *Astronomical Data Analysis Software and Systems I*, volume 25, page 176.

[214] Wicker, K. (2013). Non-iterative determination of pattern phase in structured illumination microscopy using auto-correlations in Fourier space. *Optics Express*, 21(21):24692.

[215] Wicker, K., Mandula, O., Best, G., Fiolka, R., and Heintzmann, R. (2013). Phase optimisation for structured illumination microscopy. *Optics Express*, 21(2):2032.

[216] Wiener, N., Wiener, N., Mathematician, C., Wiener, N., Wiener, N., and Mathématicien, C. (1949). *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*, volume 113. MIT press Cambridge, MA.

[217] Winter, F. R., Loidolt, M., Westphal, V., Butkevich, A. N., Gregor, C., Sahl, S. J., and Hell, S. W. (2017). Multicolour nanoscopy of fixed and living cells with a single sted beam and hyperspectral detection. *Scientific reports*, 7(1):1–11.

[218] Winter, P. W. and Shroff, H. (2014). Faster fluorescence microscopy: advances in high speed biological imaging. *Current opinion in chemical biology*, 20:46–53.

[219] Xue, T., Chen, B., Wu, J., Wei, D., and Freeman, W. T. (2019). Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125.

[220] Yang, Y. S. and Strittmatter, S. M. (2007). The reticulons: a family of proteins with diverse functions. *Genome biology*, 8(12):234.

[221] Young, L. J., Ströhl, F., and Kaminski, C. F. (2016). A Guide to Structured Illumination TIRF Microscopy at High Speed with Multiple Colors. *Journal of Visualized Experiments*, (111).

[222] Zhang, D., Mishra, S., Brynjolfsson, E., Etchemendy, J., Ganguli, D., Grosz, B., Lyons, T., Manyika, J., Niebles, J. C., Sellitto, M., et al. (2021). The ai index 2021 annual report. *arXiv preprint arXiv:2103.06312*.

[223] Zhang, J., Xu, T., Shen, Z., Qiao, Y., Zhang, Y., Hang, J. I. Z., Ingfa, T. X. U., Hen, Z. I. Y. I. S., Iao, Y. I. Q., and Hang, Y. I. Z. (2019). Fourier ptychographic microscopy reconstruction with multiscale deep residual network. *Optics Express*, 27(6):8612–8625.

[224] Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155.

[225] Zhang, K., Zuo, W., and Zhang, L. (2018a). FFDNet: Toward a fast and flexible solution for CNN-Based image denoising. *IEEE Transactions on Image Processing*, 27(9).

[226] Zhang, L., Vaddadi, S., Jin, H., and Nayar, S. K. (2009). Multiple view image denoising. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1542–1549. IEEE.

[227] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. (2018b). Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301.

[228] Zhao, X., Alvarado, D., Rainier, S., Lemons, R., Hedera, P., Weber, C. H., Tukel, T., Apak, M., Heiman-Patterson, T., Ming, L., et al. (2001). Mutations in a newly identified gtpase gene cause autosomal dominant hereditary spastic paraplegia. *Nature genetics*, 29(3):326–331.

Appendix A

Supplementary information for ERnet

A.1 Processing pipeline for ERnet

The entire pipeline for the method that relies on ERnet for segmentation and graph processing for quantitative analysis is illustrated on Figure A.1. Super-resolution microscopy images are reconstructed with ML-SIM, denoised with a non-local denoiser, ND-SAFIR, after which the machine learning method presented in this paper, ERnet, is applied to provide segmented images of the endoplasmic reticulum (ER) tubular and sheet structures. The segmented images can then be further analysed by converting the segmentation maps into graph representations providing several graph metrics which along with image statistics derived from the segmentation map gives unique insight into the characteristics of the ER.

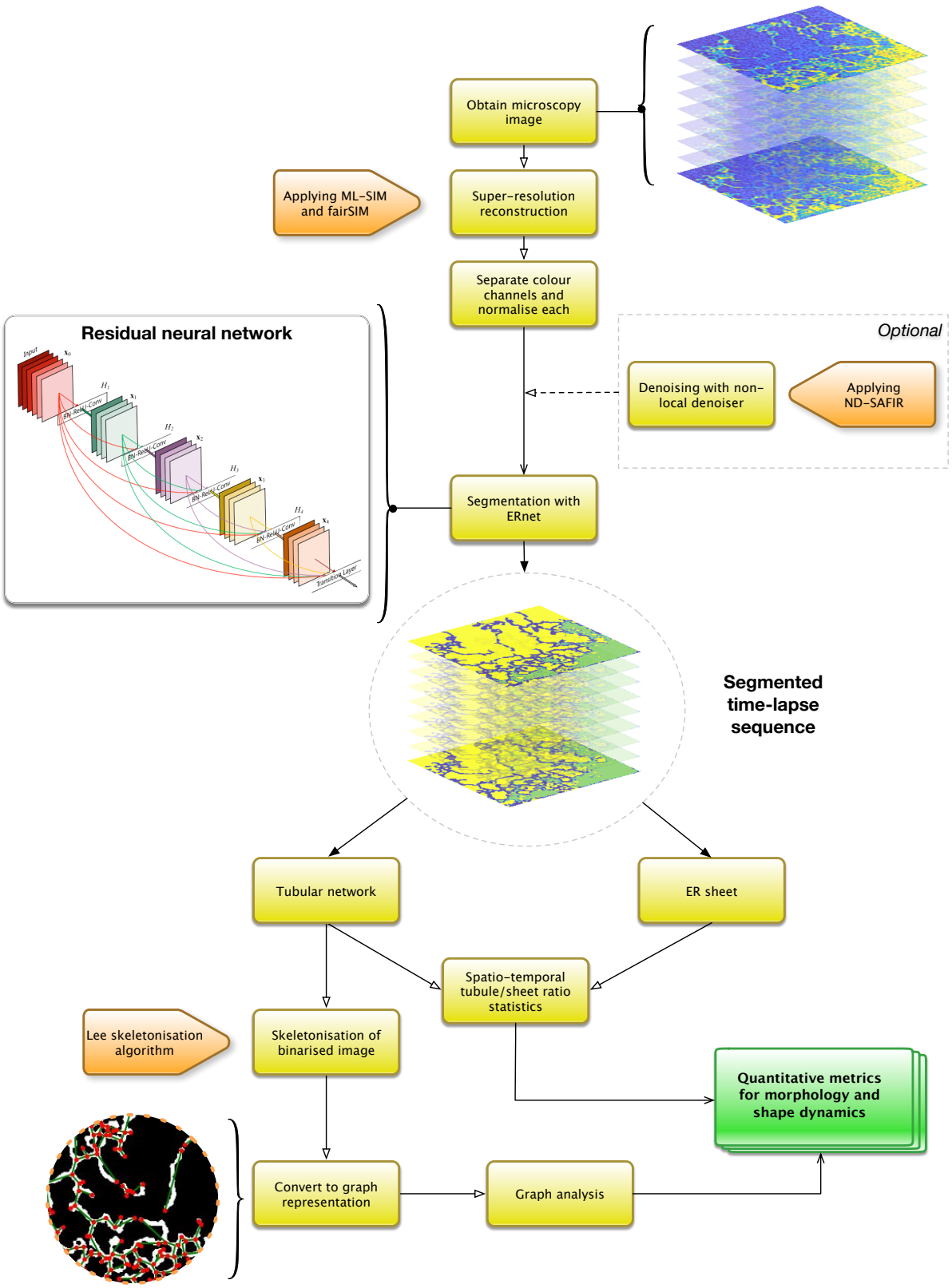


Figure A.1: Pipeline for the processing and quantitative extraction of morphology and shape dynamic metrics.

Appendix B

Supplementary information for ML-SIM & VSR-SIM

B.1 ML-SIM

B.1.1 ML-SIM desktop program

An easy to install and use desktop application for Windows, macOS and Linux has been developed for ML-SIM and is available at GitHub, https://github.com/charlesnchr/ML-SIM, and figshare [30]. The program allows one to batch process a set of directories including subdirectories that contain TIFF stacks, in addition to customising and selecting the model used for reconstruction. The program is based on NodeJS and Python, using Pytorch as the deep learning framework underneath. Required dependencies are downloaded automatically. GPU acceleration is available with CUDA-compatible Nvidia GPUs. The program includes a plugin for µManager that can be activated to enable a real-time live-view of ML-SIM reconstructed output with a frame rate over 5 FPS on a medium-tier PC with a recent GPU. See Figure B.1 for a screenshot.

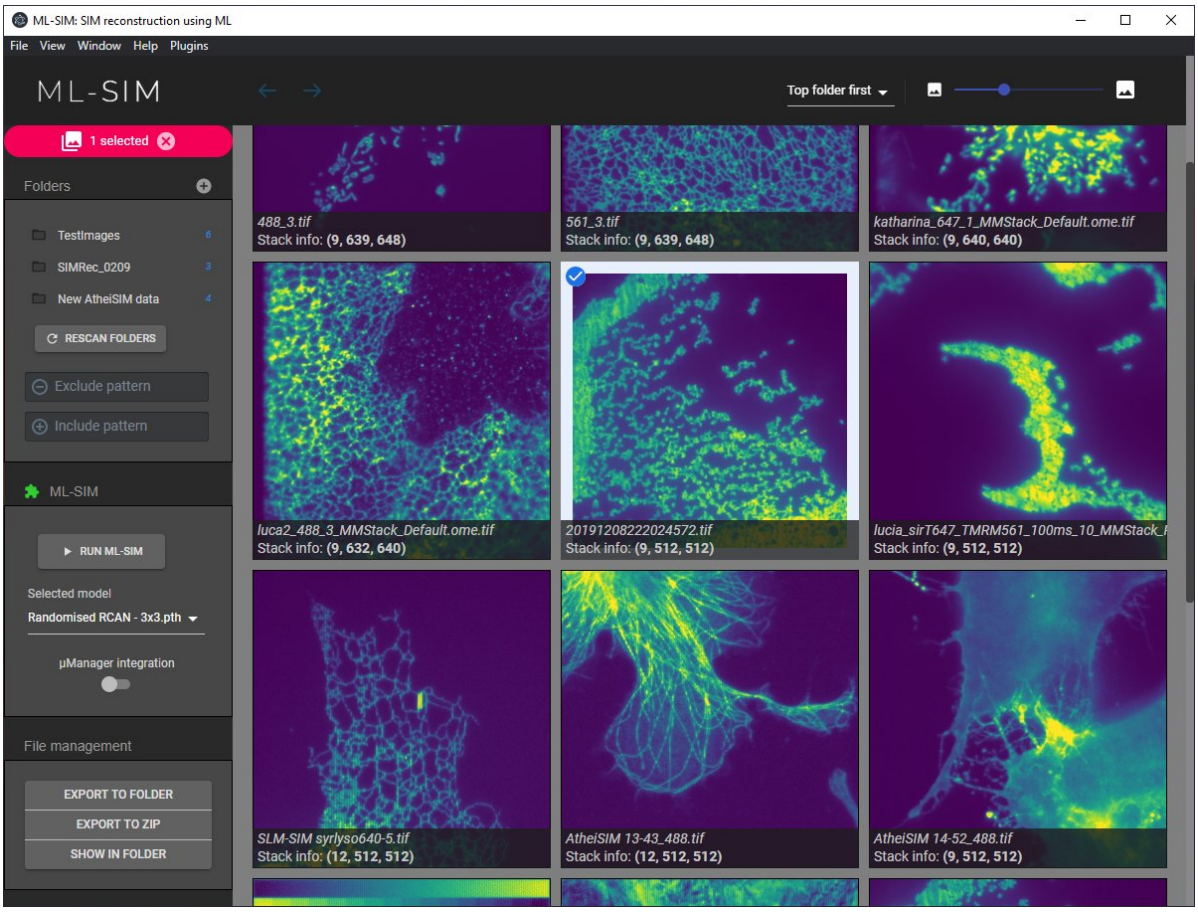


Figure B.1: Interface of ML-SIM desktop program with two open folders. Batch processing is possible by selecting multiple or all images in the view, and the specific ML-SIM model used can be changed from a drop-down menu.

B.1.2 Performance assessment on test image set

Test on two different image sets, DIV2K and Kodak 24. The sets consist of 10 and 24 images, respectively, all of which are distinct from the original images used for the training data.

	DIV2K Test Set		Kodak 24 Set	
	PSNR [dB]	SSIM	PSNR [dB]	SSIM
Wide-field	25.31	0.84	24.05	0.85
CC-SIM	25.37	0.89	24.61	0.86
FairSIM	25.34	0.86	25.32	0.86
OpenSIM	28.46	0.91	27.36	0.92
ML-SIM	**30.30**	**0.95**	**30.22**	**0.96**

Table B.1: Test scores on simulated raw SIM data generated from image sets DIV2K and Kodak 24 for commonly used reconstruction methods and for ML-SIM.

B.1.3 Residual neural network architecture of ML-SIM

The model used in ML-SIM is a deep residual neural network which is largely based on the ResNet architecture and the extensions to single image super-resolution with EDSR and RCAN. A diagram is shown in Figure B.2.

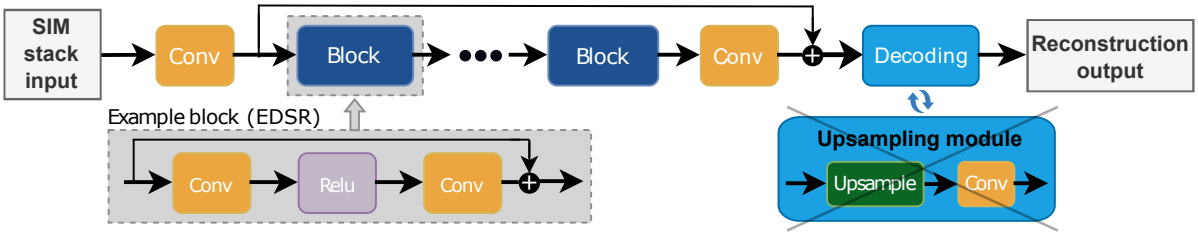


Figure B.2: The architecture of ML-SIM is inspired by state-of-the-art single image super-resolution architectures. Here the architecture of EDSR is shown, but the same structure applies to RCAN only with a more complex block called a channel attention block. ML-SIM has a RCAN architecture without an upsampling module and with a larger input layer that handles 9 frames.

B.1.4 Structured illumination microscopy methodology

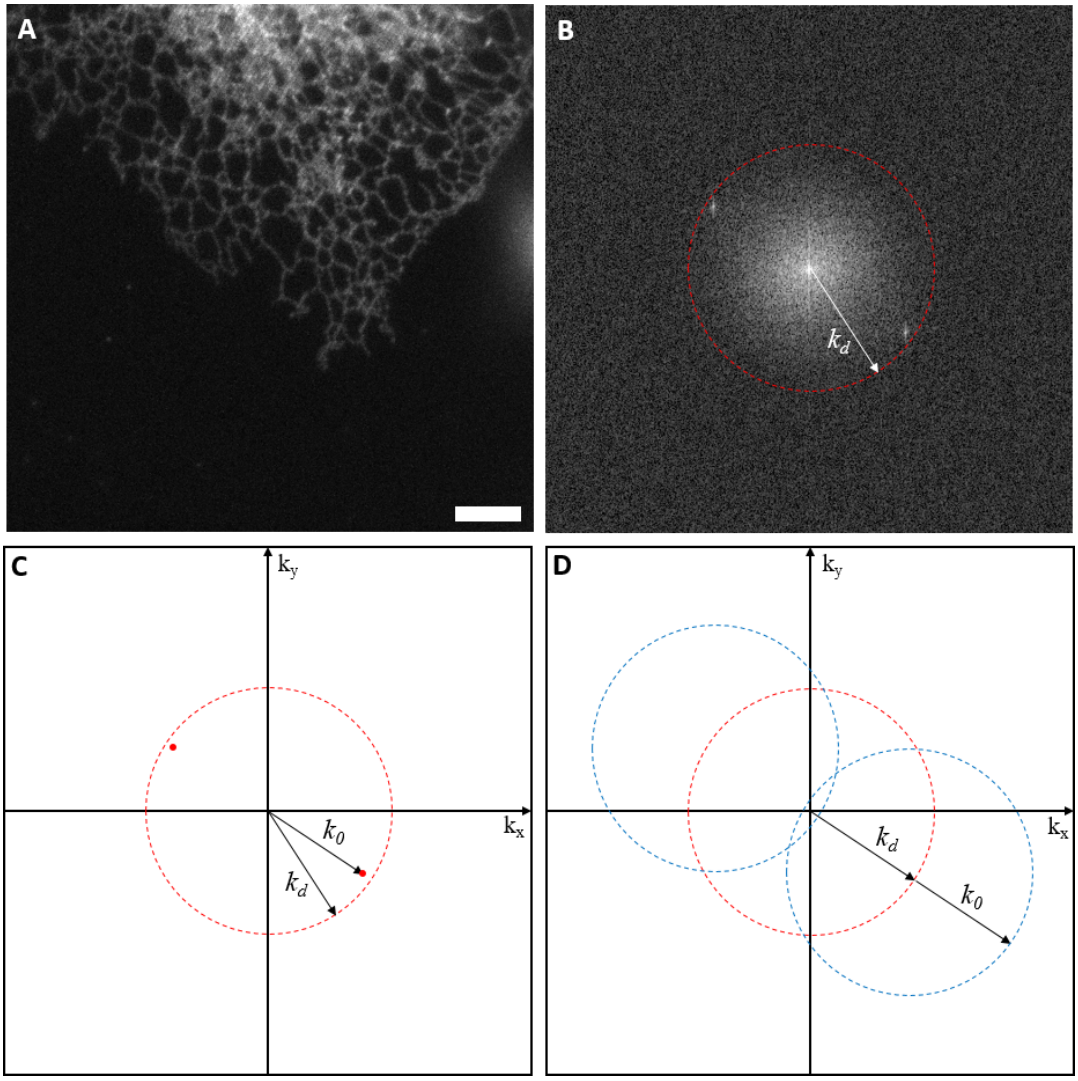


Figure B.3: SIM methodology visualised in frequency space. (A) Raw image captured during SIM. Scale bar is 5 μm. (B) 2D Fourier transform of A. The resolution limit can be visualised as a cutoff frequency k_d beyond which no spatial frequency information from the sample is collected. The frequency components of the striped illumination pattern are visible as bright peaks close to the cutoff frequency. (C) The frequency components of the excitation pattern, k_0, are chosen to be as close to the diffraction limit as possible, to maximise resolution increase. The interference of the patterned illumination with the sample pattern means the observed region of frequency space now contains frequency components from outside the supported region, shifted by $\pm k_0$. (D) By shifting the phase of the pattern, the regions of frequency space can be isolated and moved to the correct location in frequency space. The maximum spatial frequency recovered is now $k_d + k_0$.

B.1.5 Poisson noise for data generation

By default the ML-SIM model uses Gaussian noise source for data generation. The underlying Gaussian distribution is randomised from image to image to make the model more generalised. In microscopy, however, Poisson noise is often the predominant noise source [93]. I tested whether the performance of ML-SIM is significantly affected by the noise model used to generate the test data and performed reconstructions of images corrupted by Poisson noise. The results are shown in Figure B.4 below. I have not found a strong sensitivity on the type of noise source used for data generation and other factors, such as blur caused by the PSF, out-of-focus light and errors in the SIM illumination pattern, (i.e. errors in phase shifts or stripe orientations) were found to have a more significant effect. On the other hand, high levels of synthetic noise used for data generation may be detrimental to the final performance of the model.

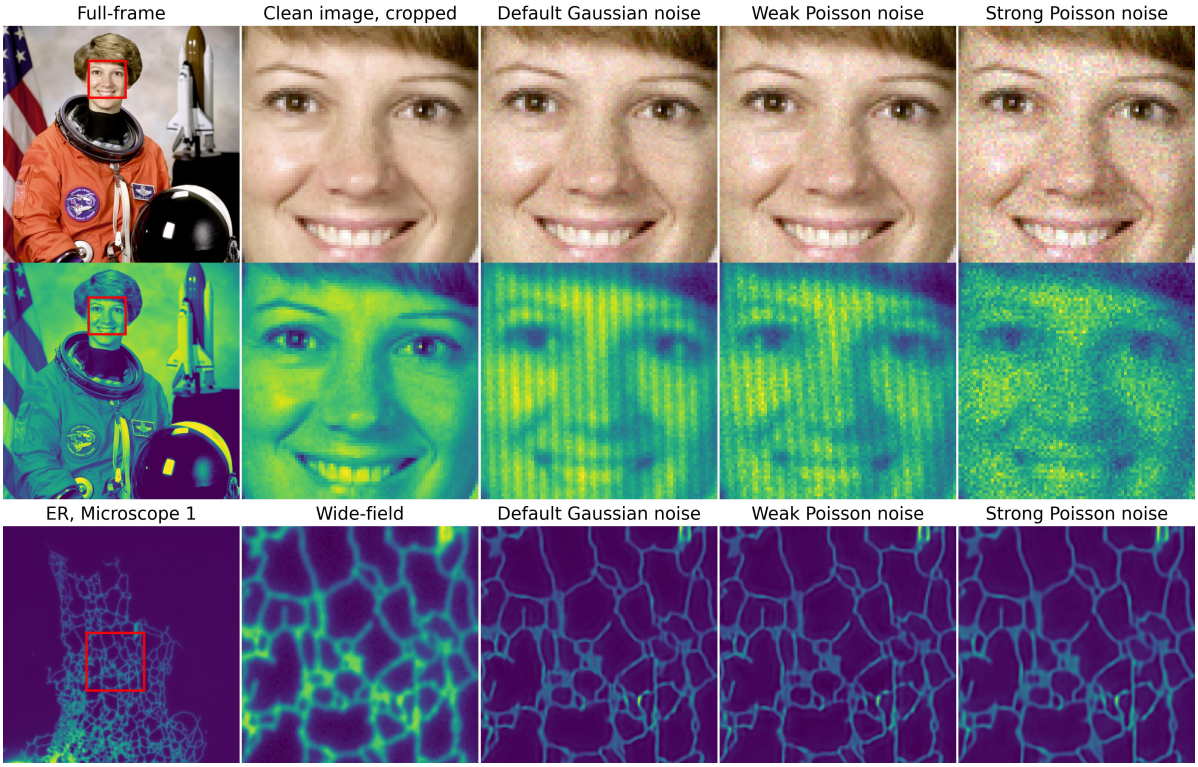


Figure B.4: Model output obtained when using either Gaussian noise and Poisson noise in training data. (Top) Examples of the noise models applied to a clean RGB image. (Center) Training data sample when the same noise distributions are used in the data generation pipeline that simulates SIM image formation. (Bottom) Resulting reconstruction output when models are trained on simulated SIM data using the respective noise distributions.

B.1.6 Influence of SIM stack size

Almost all the reconstruction outputs presented in Section 5.1 are based on a ML-SIM model trained to work on a SIM configuration with illumination patterns consisting of three orientations and three phase shifts, a 3×3 configuration. However, the ML-SIM pipeline fully supports any configuration of SIM, and the usual benefits of using larger SIM stacks also apply here. One benefit is noise robustness and consequently an improved reconstruction quality, but at the risk of photodamage to the sample and lower imaging speed. The improvement in reconstruction quality when used on simulated test images is shown in Figure B.5, where models for 3×3 (default), 3×5 and 5×5 SIM configurations are compared. The mean value of the respective structural similarity index measures is obtained by averaging over a total of 1000 test images that have been reconstructed with each method. Each test image exists in three versions according to the different SIM configurations, but the underlying point spread function, as well as the noise and error characteristics, is similar.

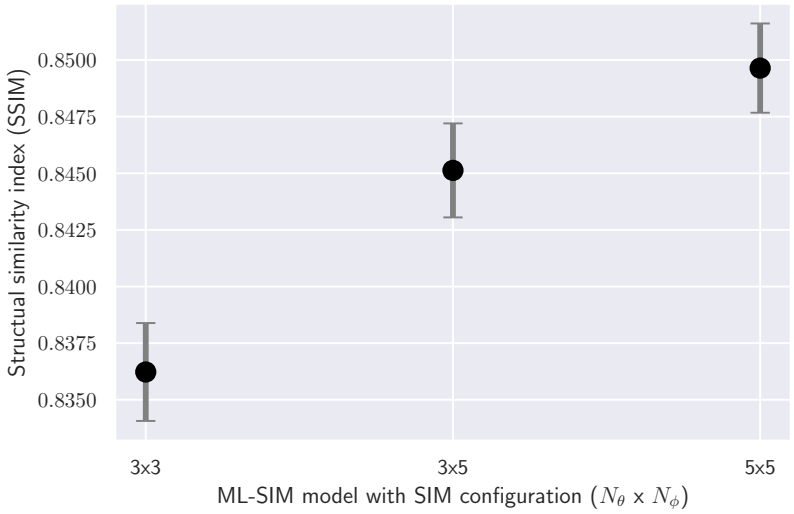


Figure B.5: Average SSIM score for three different ML-SIM models, each with distinct SIM configurations based on the number of illumination stripe orientations, N_θ, and the number of phase shifts, N_ϕ, when tested on 1000 test images with similar noise levels. The error bars indicate the standard error of the mean.

B.1.7 Modulation depth, frequency, phase errors and orientation angles

As described in Section 2.2, the illumination stripe patterns are calculated from their spatial frequency k_0 and a phase ϕ,

$$I_{\theta,\phi}(x,y) = I_0\left[1 - \frac{m}{2}\cos\left(2\pi(k_x \cdot x + k_y \cdot y) + \phi\right)\right], \tag{B.1}$$

where $[k_x, k_y] = [k_0\cos\theta, k_0\sin\theta]$ for a pattern orientation θ and m is the modulation depth. The training data for training ML-SIM is generated with randomised values for k_0 and m by sampling uniformly from the intervals $k_0 \in [0.22, 0.28]$ cycles/px (cycles per pixel) and $m \in [0.65, 0.95]$, respectively. In a standard SIM implementation a number of illumination phase shifts, ϕ, are used at each orientation according to an evenly spaced interval. For a typical configuration of three orientations and three phase shifts (3×3), the phase shift values might therefore be 0, $\frac{1}{3} \times 2\pi$ and $\frac{2}{3} \times 2\pi$.

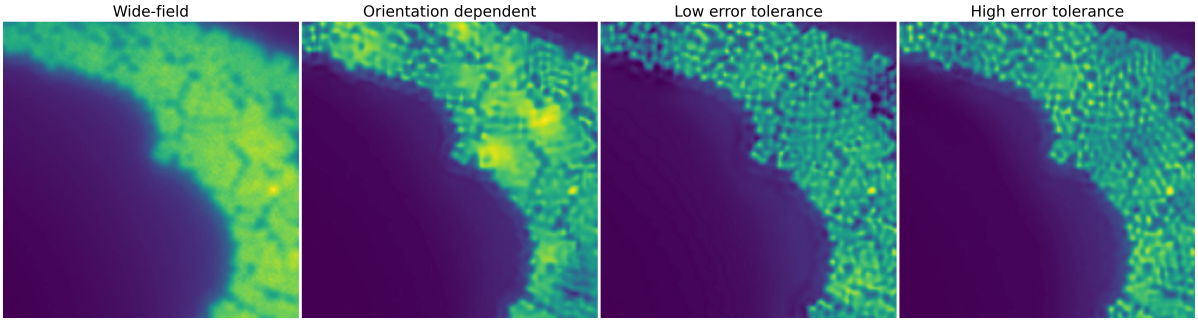


Figure B.6: Performance of ML-SIM models trained with fixed orientation ordering (orientation dependent), low level of phase shift errors (low error tolerance) and high level of phase shift errors (high error tolerance – this is the default ML-SIM model). Example reconstruction outputs of the respective models.

Depending on the nature of the SIM instrumentation that produces the illumination patterns, these phase shifts will be offset by some error, and furthermore, they may not be highly consistent from image stack to image stack. Thus, it is of high importance to include an approximation of phase errors in the training data generation for ML-SIM to obtain a model that is robust to such errors. In the most extreme case, the phase shifts could be completely random with no constraint as to whether the values are too similar or not sufficiently spaced across the 2π period. This is how the default ML-SIM model presented in Section 5.1 has been trained with the aim of improving the generality of the model. This is referred to as a model with high (phase) error tolerance. A corresponding model with everything kept the same but with consistent phase steps, i.e. each phase only deviating from its ideal value by a few percent, is referred to as a model with low error tolerance.

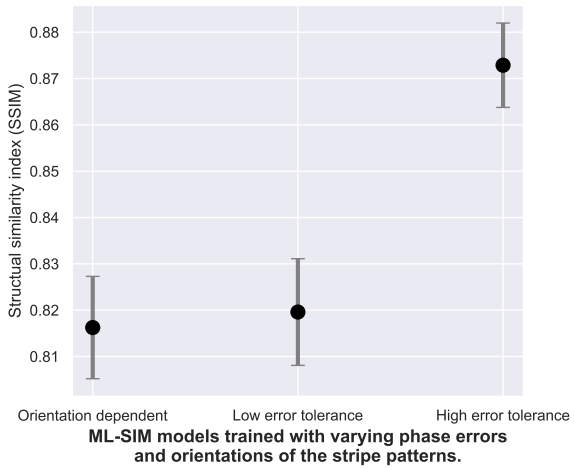


Figure B.7: Mean reconstruction qualities of respective models when averaged over 100 test images that contain a high level of phase shift errors and random orientation orderings. The error bars indicate the standard error of the computed means.

Another parameter that will vary across distinctive, real SIM systems is the order of the illumination stripe orientations. The 9 frames in a 3×3 SIM stack might be ordered according to orientation angles of 0°, 0°, 0°, 120°, 120°, 120°, 240°, 240°, 240° from the first axis, e.g. the x-axis of the image frame. However, there is no standard across different systems, so the order of the frames could equally correspond to orientation angles of 240°, 240°, 240°, 0°, 0°, 0°, 120°, 120°, 120°. In addition to this, there are also offsets and errors in the actual angles. To make ML-SIM able to work well despite uncertainty about the particular ordering, and in the presence of errors and other offsets to the orientations, the simulated SIM images in the training data consist of all the permutations by using randomisation. A model that is not trained with these different permutations becomes orientation dependent – i.e. if the ordering of orientations is fixed in all of the training data samples.

The above-mentioned ML-SIM models have been tested on actual SIM images acquired experimentally and a simulated test dataset of 100 images with a high presence of phase errors and random orientation ordering. An example of reconstruction outputs of a SIM image of beads on Microscope 2, as defined in Section 5.1, in addition to the mean structural similarity index measures across the simulated test image set are shown on Figure B.6. The output from the two models with low and high error tolerance appear similar on experimental data, and only significantly differ when testing on the simulated images that are known to have a high level of phase shift errors. The model that is orientation-dependent appears to lose both resolution and contrast when testing on experimentally acquired SIM images, as indicated in the example on

Figure B.6, but performs at a similar quality as the model with low error tolerance on the test images with high phase shift errors.

B.1.8 Inspection of frequency support

The resolution improvement provided by ML-SIM can also be visualised in frequency space as an extension of the spatial frequency passband (i.e. high spatial frequencies in the Fourier transform of the reconstructions). Figure B.8 shows a comparison of reconstruction techniques in frequency space, and Figure B.9 shows a plot of the normalised intensity in frequency space. The raw data was acquired by imaging microtubules labelled with Alexa-647 on the spatial light modulator based SIM microscope with a 647 nm excitation laser and a 1.2 numerical aperture water immersion objective.

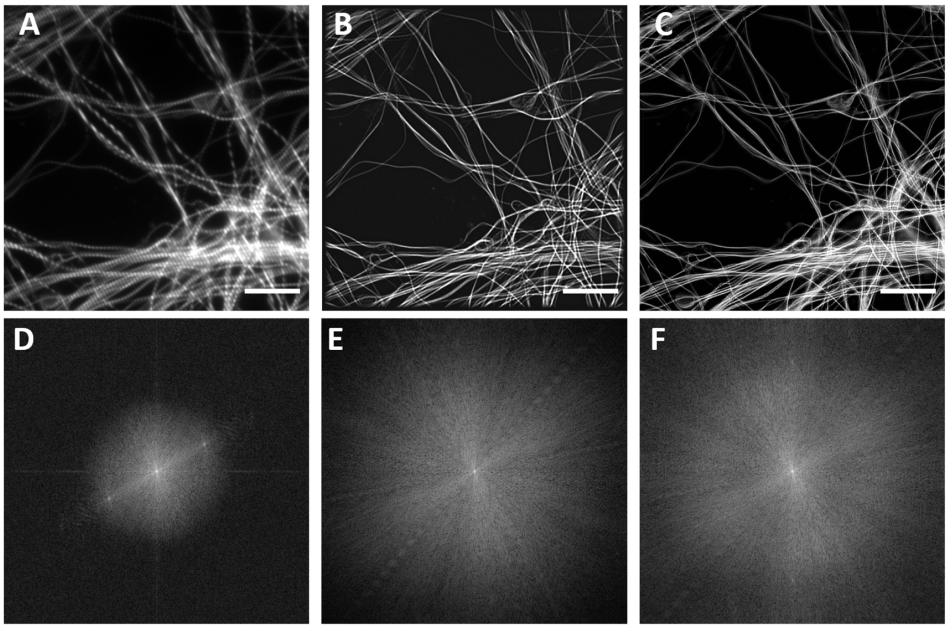


Figure B.8: Fourier Spectrum Analysis (FSA) of SIM reconstruction methods. A: raw striped-pattern SIM frame from microscope. B: FairSIM reconstruction. C: ML-SIM reconstruction. D, E, and F: log Fourier transforms of A, B, and C respectively. Stripe patterns appear on D as peaks in frequency space. The graph depicts the normalised intensity of the log Fourier transform as a function of spatial frequency. Orange line: wide-field; gray line: FairSIM; blue line: ML-SIM. Both ML-SIM and FairSIM have extended the range of frequencies supported, indicating high-resolution information is present in the reconstruction. FSA was performed for a reconstruction of SIM data acquired on microscope 1 of microtubules labelled with Alexa-647.

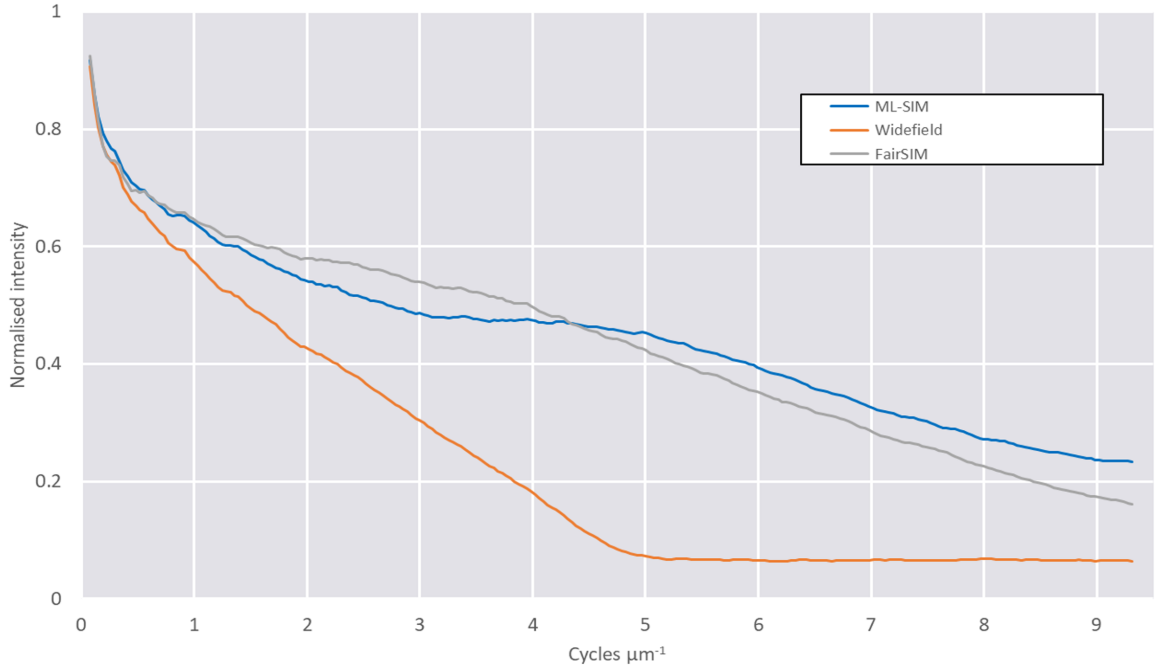


Figure B.9: Normalised intensity of the log Fourier transform as a function of spatial frequency. Orange line: wide-field; gray line: FairSIM; blue line: ML-SIM. Both ML-SIM and FairSIM have extended the range of frequencies supported, indicating high-resolution information is present in the reconstruction. Note that the cut-off frequency for the wide-field is lower than that predicted from the Abbe limit as spherical aberrations inevitably degrade frequency support.

B.1.9 Training ML-SIM with ideal SIM targets

The standard ML-SIM model used throughout the paper is trained with clean and unmodified images as targets in a supervised learning approach. However, the targets could instead have been limited to the resolution corresponding to the theoretical optimum of standard SIM reconstruction, i.e a resolution increase of a factor of 2 over a wide-field image. This is enabled

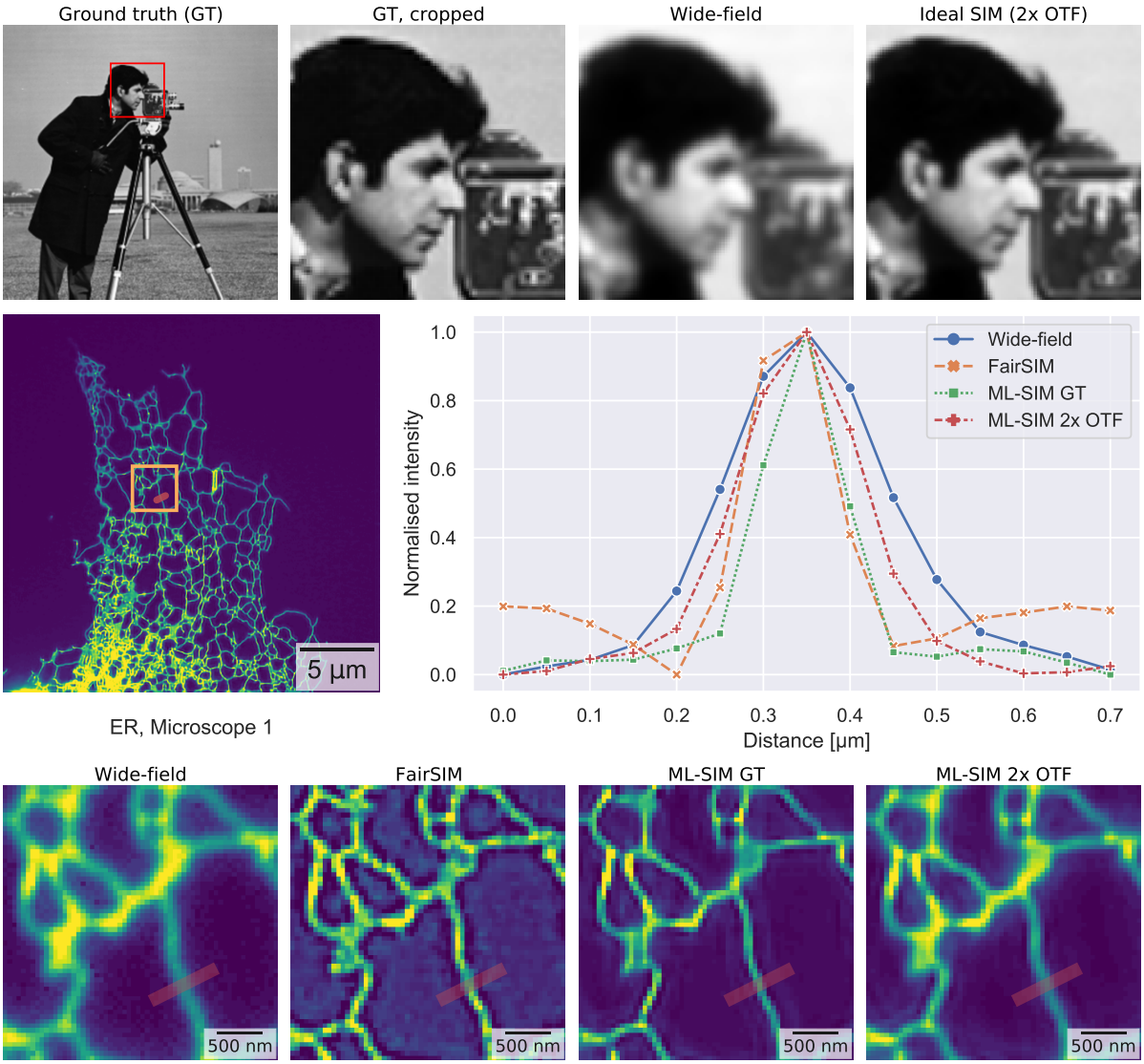


Figure B.10: Two ML-SIM models are compared with FairSIM: one trained with ground truth (GT) images as targets, and another trained with simulated ideal SIM reconstructions as targets. (Top) Sample training image illustrating the two types of targets. (Centre) Full field-of-view ER image and line profiles are used to compare the intensity along the displayed red line for the different reconstruction outputs. (Bottom) Cropped regions are displayed, showing the reconstruction outputs corresponding to the area enclosed by the yellow rectangle.

by gaining the frequency support of a modified optical transfer function (OTF) with twice the radius over the wide-field equivalent OTF. A more conservative model could be obtained in this way at the expense of resolution. This is illustrated on Figure B.10, where an ML-SIM model trained in such a way (ML-SIM 2x OTF) provides reconstruction output of lower resolution than the default ML-SIM model (ML-SIM GT). While other studies on applying deep learning to microscopy have reported on content-aware approaches [202, 207], ML-SIM is trained with its diverse training data to avoid sample-specific models, thus in principle preventing resolutions in reconstructions that exceed the theoretical SIM optimum. Yet, basic features such as simple curves, lines, edges and corners are arguably similar between natural objects across different length scales. Imposing this resolution constraint during training may thus cause the reconstruction quality to suffer as indicated by the corresponding line profile in Figure B.10. The full width at half maximum of the peaks of the tubular profiles is found to be approximately 120 nm for ML-SIM GT and FairSIM, 180 nm for the constrained ML-SIM GT model and 230 nm for the wide-field projection.

B.1.10 Applying ML-SIM to TIRF-SIM data

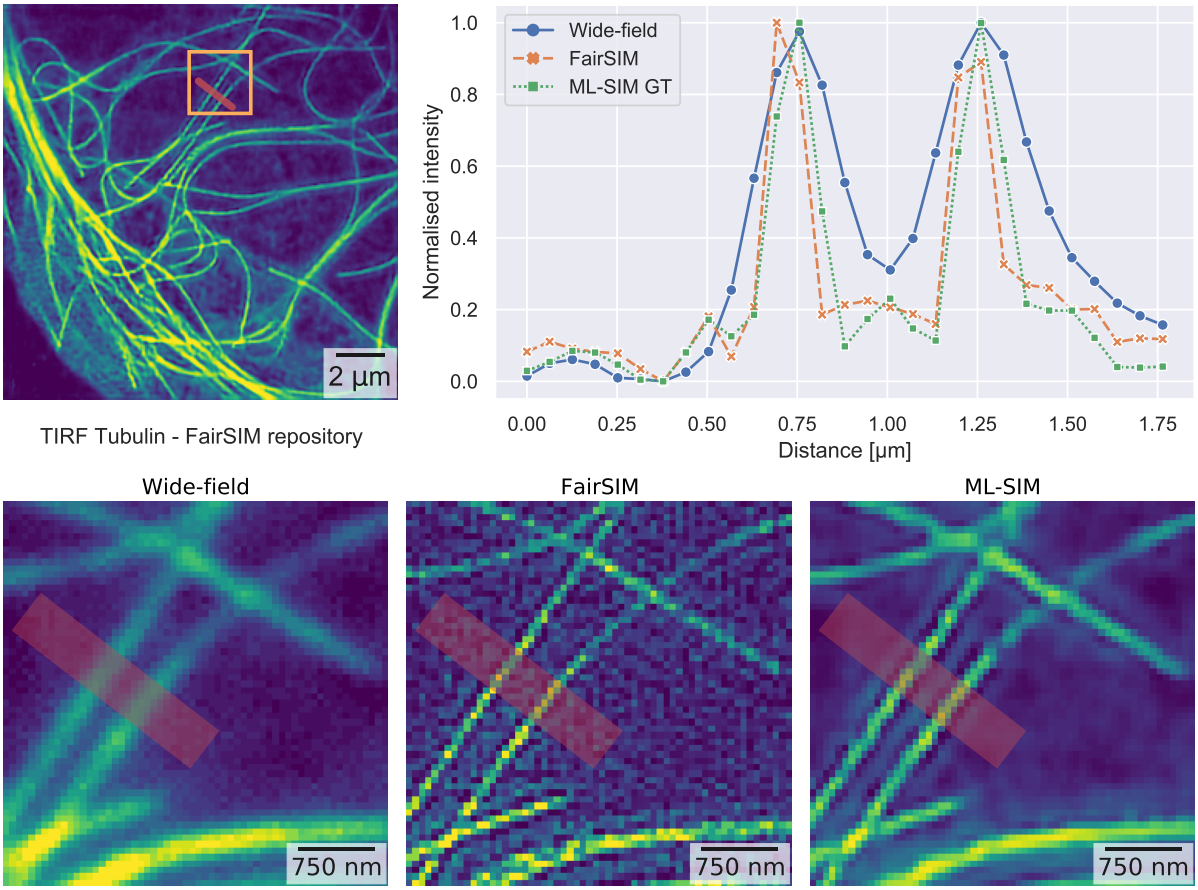


Figure B.11: Resolution improvement when reconstructing a TIRF-SIM image of tubulin from the official FairSIM repository. ML-SIM reconstruction output is compared with a wide-field projected image and FairSIM. (Top) Full field-of-view reconstructed TIRF-SIM image and line profiles comparing the intensity along the displayed red line for the different reconstruction outputs. (Bottom) Cropped regions of the reconstruction outputs corresponding to the area enclosed by the yellow rectangle.

ML-SIM is also tested on a third SIM system which is distinct from Microscopes 1 and 2 described in Section 5.1 in that it uses total internal reflection fluorescence structured illumination microscopy (TIRF-SIM) and produces raw SIM images at a resolution of 256×256 pixels per frame, while ML-SIM is trained for images with 512×512 pixels per frame. Rather than training a separate model specifically for this system, the TIRF-SIM data is reconstructed with the same ML-SIM model used throughout Section 5.1 to further demonstrate its generality. The TIRF-SIM image used here is a test image of tubulin from the open-source FairSIM repository [146]. Reconstruction output and line profiles from the respective methods across the tubulin structures are shown on Figure B.11.

B.2 VSR-SIM

B.2.1 Implementation of VSR-SIM

The source code for the implementation of VSR-SIM has been made available on GitHub at https://github.com/charlesnchr/VSR-SIM. The source code is structured as follows:

- Video dataset sampling. Powershell script:
  `scripts/sample_documentary_videos.ps1`.
- Image formation model. Python code:
  `scripts/im_form_model/SIMulator.py`.
- Data generation script:
  `scripts/datagen_pipeline.py`.
- Model architecture based on Pytorch:
  `basicsr/archs/vsr-sim_arch.py`.
- Training code:
  `basicsr/train.py`.
- Inference code for testing:
  `inference/inference_options.py`.
- RBPN code base based on official implementation:
  `RBPN-PyTorch`

The documentation is in the file `README.md`, which contains snippets of code to perform data generation, training and inference, respectively.

B.2.2 Test sets

To test on image sequences that do not exhibit motion, the DIV2K image dataset [2] is used. The images are stills, and therefore when used with the SIM image formation model, the synthetic SIM stacks correspond to sequences of static objects. For the Moderate test set used in Section 5.3, I use a subset of 50 unique sequences from the 100,000 sequences of the BBC dataset reserved for testing.

I also use 10 videos from the REDS validation dataset [149] to test with samples in which the entire field-of-view is translated from frame to frame in addition to the arbitrary motion of objects as with the Moderate test set. The videos in the REDS dataset are recorded with a

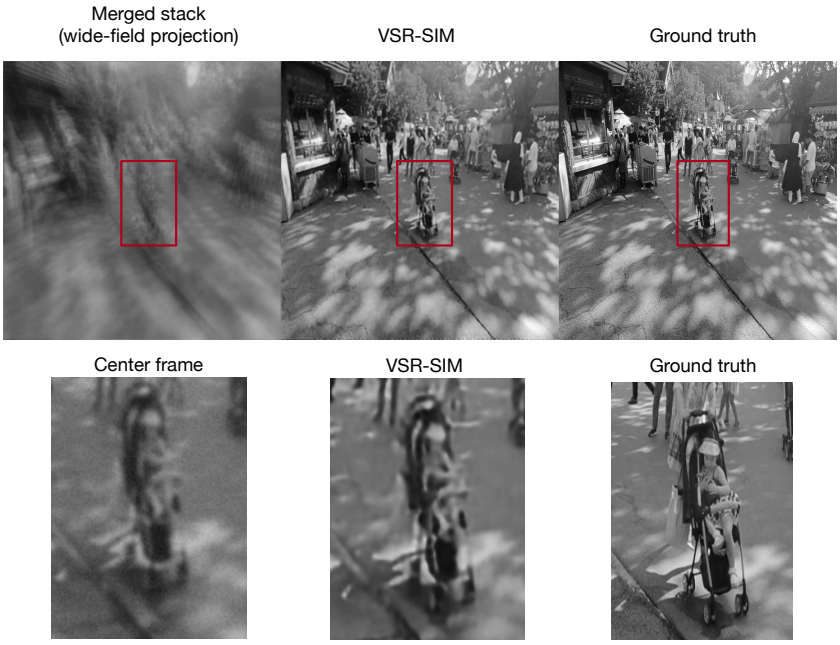


Figure B.12: To exaggerate the effect of motion during SIM reconstruction, the REDS [149] dataset is used with frame skipping. SIM reconstruction is seen to still work and not produce motion artefacts, yet the performance gain over a single image SR baseline becomes small as also suggested in Table 5.3.

handheld device that is subject to significant movement itself. This dataset constitutes a more difficult test set when fed into the image formation model that simulates the SIM imaging process. For the test set referred to as Fast in Section 5.3, I have sampled the 9 first frames in each video and use these to form a single SIM image sequence for each respective video. The Extreme test set is generated from the first 17 frames, where every other frame has been skipped to provide 9 frames with double the temporal spacing from frame to frame. An example output of VSR-SIM when evaluated on a sample from the Extreme test set is shown on Figure B.12.

B.2.3 Parameters for image formation model

The following image formation model and its implementation is inspired by the one used for ML-SIM in Section 5.1.

As described in Section 5.3.3, the illumination stripe patterns are calculated from their spatial frequency k_0 and a phase ϕ,

$$I_{\theta,\phi}(x,y) = I_0 \left[1 - \frac{m}{2}\cos\left(2\pi(k_x x + k_y y) + \phi\right)\right], \tag{B.2}$$

where $[k_x, k_y] = [k_0 \cos\theta, k_0 \sin\theta]$ for a pattern orientation θ and m is the modulation depth. The training data for VSR-SIM is generated with randomised values for k_0 and m by sampling

uniformly from the intervals $k_0 \in [0.22, 0.28]$ cycles/px and $m \in [0.65, 0.95]$, respectively. In a standard SIM implementation, a number of illumination phases, ϕ, are used at each orientation according to an evenly spaced interval. For a typical configuration of three orientations and three phase shifts (3×3), the phase shift values might therefore be 0, $\frac{1}{3} \times 2\pi$ and $\frac{2}{3} \times 2\pi$.

Depending on the SIM instrumentation that produces the illumination patterns, these phase shifts will be offset by some error, and furthermore, they may not be highly consistent from one image stack to another. Thus, it is important to include an approximation of phase errors in the training data generation for VSR-SIM to obtain a model that is robust to these errors. In the most extreme case, the phase shifts could be completely random with no constraint as to whether the values are too similar or not sufficiently spaced across the 2π period. This is how the default VSR-SIM model presented in Section 5.3 has been trained with the aim of improving the generality of the model.

Another parameter that will vary across distinctive, real SIM systems is the order of the illumination stripe orientations. The 9 frames in a 3×3 SIM stack could be ordered according to orientation angles of 0°, 0°, 0°, 120°, 120°, 120°, 240°, 240°, 240° from the first axis, e.g. the x-axis of the image frame. However, there is no standard across different systems, so the order of the frames could equally correspond to orientation angles of 240°, 240°, 240°, 0°, 0°, 0°, 120°, 120°, 120°. In addition to this there are also offsets and errors in the actual angles. To make VSR-SIM capable of reconstructing frames according to the rolling SIM imaging scheme, and in the presence of errors and other offsets to the orientations, the simulated SIM images in the training data consist of all the permutations by using randomisation.

B.2.4 Image and video super-resolution methods

Multiple state-of-the-art architectures were adapted for the SIM reconstruction problem. Using the Medium test set, see Table 5.1, I compare the performance of four architectures with the proposed architecture shown in Figure 5.25. The fully convolutional RCAN [227] is included as a CNN baseline, while the other methods are more recent and use either optical flow or multi-head attention. The results are shown in Table B.2 indicating that the combination of channel attention and multi-head attention is advantageous, which is further explored in the ablation study of Section 5.3.5.

B.2.5 Hyperparameters for tested models

For the comparison with state-of-the-art models in e.g. Table 5.4 the following models were used: RCAN, SwinIR, Video Swin and RBPN. Below I provide hyperparameters that were used for each of them and the reference implementations. Paths for files specifying the parameters

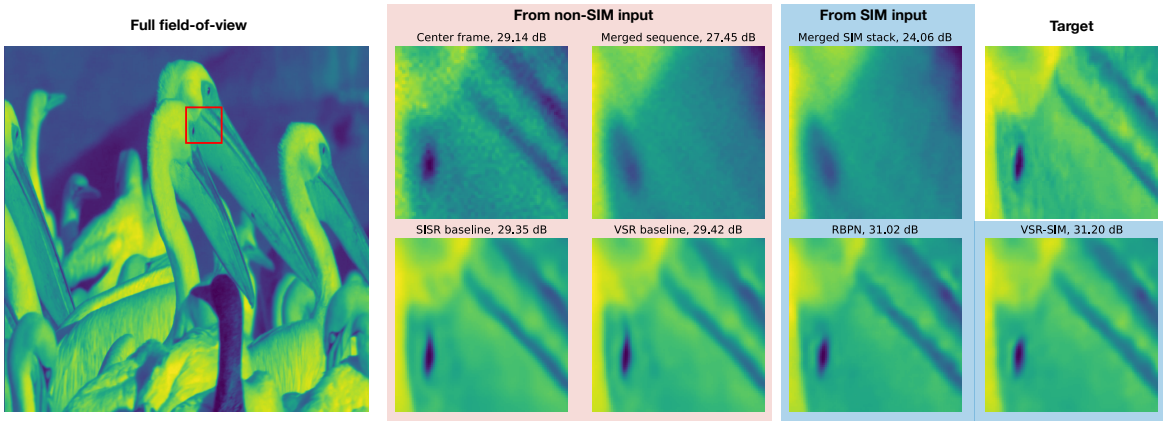


Figure B.13: Single image SR (SISR) and video SR (VSR) baselines compared with SIM stack reconstruction using RBPN (based on optical flow) and VSR-SIM (ours). The displayed metrics are peak signal-to-noise ratio (PSNR). The centre frame corresponds to the time point for which SR is desired, and this is used solely as input for the SISR model without illumination patterns. Overlaying the frames of the image sequence reveals that significant translation occurs. The sequence without illumination patterns is used as input for the VSR model. The stack with patterns is used as input for RBPN and VSR-SIM, and the wide-field projection is the average across this stack.

Adapted architecture	Score (PSNR)
RCAN [227]	29.06
RBPN [63]	29.25
SwinIR [113]	29.48
Video Swin [122]	29.10
VSR-SIM (ours)	**30.15**

Table B.2: Test scores on the Medium test set for various architectures that have been adapted for performing video super-resolution of SIM sequences.

are given for those models that have been integrated in the main codebase. These files have the path prefix `basicsr/options/..` in the supplementary code.

- **RBPN** [63]
  Option file: `RCAN/RCAN.yml`
  Reference implementation: RBPN GitHub.
  https://github.com/alterzero/RBPN-PyTorch
  No. channels in: 9
  No. channels out: 1
  No. of initial feature channels: 256
  No. of deep feature channels: 64

No. of stages: 3
No. of residual blocks: 5

- **RCAN** [227]
  Option file: `RCAN/RCAN.yml`
  Reference implementation: BasicSR [203].
  https://github.com/xinntao/BasicSR
  No. channels in: 9
  No. channels out: 1
  No. of feature channels: 64
  No. of residual groups: 10
  No. of residual blocks: 20
  Squeeze factor: 16
  Residual scale: 1

- **SwinIR** [113]
  Option file: `SwinIR/SwinIR.yml`
  Reference implementation: SwinIR GitHub
  https://github.com/JingyunLiang/SwinIR
  No. channels in: 9
  No. channels out: 1
  Window size: 8
  No. of Swin transformer layers: 6
  Depths of Swin transformer layers: (6, 6, 6, 6, 6, 6)
  Embedding size: 180
  Attention head number: (6, 6, 6, 6, 6, 6)

- **Video Swin** [122]
  Option file: `VideoSwin/VideoSwin.yml`
  Reference implementation: Video Swin GitHub
  https://github.com/SwinTransformer/Video-Swin-Transformer
  No. channels in: 1
  No. channels out: 1
  Patch size: (3,4,4)
  Window size: (2, 7, 7)
  MLP ratio: 4
  No. of Swin transformer layers: 4
  Depths of Swin transformer layers: (2, 2, 6, 2)

Embedding dimension: 96
Attention head number: (3, 6, 12, 24)

- **VSR-SIM**
  Option file: `VSR-SIM/VSR-SIM.yml`
  Implementation: `basicsr/../vsr-sim_arch.py`
  No. channels in: 1
  No. channels out: 1
  Patch size: (3,4,4)
  Window size: (2, 8, 8)
  MLP ratio: 2
  No. of Swin transformer layers: 5
  Depths of Swin transformer layers: (6, 6, 6, 6, 6)
  Embedding dimension: 192
  Attention head number: (8, 8, 8, 8 ,8)