Thèse n° 8295

Weakly Supervised Deep Learning Methods for Biomicroscopy

Présentée le 24 mars 2021

Faculté des sciences et techniques de l'ingénieur
Laboratoire de l'IDIAP
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Adrian SHAJKOFCI

Acceptée sur proposition du jury

Prof. D. N. A. Van De Ville, président du jury
Prof. F. Fleuret, Dr M. S. D. Liebling, directeurs de thèse
Prof. C. Wählby, rapporteuse
Prof. B. Obara, rapporteur
Prof. A. Oates, rapporteur

École
polytechnique
fédérale
de Lausanne

2021

Abstract

Optical microscopy, an invaluable tool in biology and medicine to observe and quantify cellular function, organ development, or disease mechanisms, requires constant trade-offs between spatial, temporal, and spectral resolution, invasiveness, acquisition time, and post-processing effort. Deep learning technologies have enabled multiple applications that are transforming our day-to-day routines, including the way we approach microscopy. Yet despite the ever-increasing computational power, it is often the lack of labeled training data that is the limiting factor for wide adoption in this domain. Annotating data is often a lengthy and expensive task, since it involves tedious work, generally by skilled experts.

In this thesis, I explored “weakly supervised” learning methods targeted at a variety of applications to enhance microscopy images and extract physical information from a single image. The specificity of these “weakly supervised” methods is the fact that they use very little prior information about the image in order to keep the effort to annotate training data as low as possible. Specifically, I reduced the dimensionality of the learning problem by targeting the experiment towards estimating the parameters of a spatially-variant Point Spread Function (PSF) model using a Convolutional Neural Network (CNN), which does not require instrument- or object-specific calibration. Using such a model permitted to simulate realistically accurate training data that could be generalized, once the model was trained, to real microscopy images. I extensively benchmarked different network architectures, training datasets and simulation modalities towards the optimal PSF prediction performance and robustness to image degradation.

Starting from the estimated PSF model parameters, I developed a variety of applications, such as a semi-blind spatially-variant deconvolution method for image deblurring and enhancement, a robust and fast microscopy auto-focus, a method for the estimation of the object surface from a single 2D image, and a method for the estimation of the object velocity in a fluid, all of them with minimal need for a priori knowledge about the optical setup.

Résumé

La microscopie optique, un précieux outil en biologie et en médecine pour observer et quantifier les fonctions cellulaires, le développement des organes ou les mécanismes des maladies, requiert de nombreux compromis en terme de résolution spatiale, temporelle et spectrale, ainsi qu'en terme d'invasivité, de temps d'acquisition et de difficulté à traiter le signal.

Les technologies d'apprentissage profond ("Deep Learning") ont depuis peu permis de nombreuses applications qui transforment nos routines journalières, y compris la façon dont on se sert de la microscopie. Cependant, malgré une puissance de calcul toujours plus performante, le facteur limitant leur adoption dans ce domaine est souvent le manque de données d'entraînement annotées. En effet, l'annotation des données est souvent une tâche longue et onéreuses, car il s'agit de travail fastidieux qui nécessite des experts confirmés.

Dans cette thèse, j'ai exploré des méthodes d'apprentissage "faiblement supervisées" qui touchent une quantité de différentes applications afin d'améliorer les images en microscopies, et d'en extraire des informations physiques. La particularité de ces méthodes "faiblement supervisées" est qu'elles ne nécessitent que peu d'information préalable sur l'image afin de réduire considérablement l'effort d'annotation. Spécifiquement, j'ai réduit la dimensionnalité du problème d'apprentissage en reciblant le problème vers l'estimation de paramètres d'une fonction d'étalement du point (PSF) qui varie en dans l'espace en utilisant un réseau de neurones convolutionnel (CNN) qui ne nécessie aucune calibration spécifique à l'objet ou à l'instrument. Utiliser un tel modèle permet de simuler de manière réaliste des données d'entraînement qui peuvent ensuite être généralisées, une fois le modèle entraîné, à de vraies images de microscopie. J'ai comparé en détail différentes architectures de réseau convolutionnel, différents jeux de données d'entraînement et plusieurs modalités de simulation afin d'obtenir une prédiction de la PSF la plus exacte possible et la plus robuste aux dégradations du signal.

A partir de ces paramètres du modèle de la PSF, j'ai développé de nombreuses applications, comme une méthode de déconvolution "semi-aveugle" et variante dans l'espace afin de déflouter et améliorer la qualité de l'image, un logiciel d'auto-focus robuste et rapide pour la microscopie, une méthode d'estimation de la surface tridimensionnelle d'un objet à partir d'une image plane, et une méthode d'estimation de la vitesse et de la trajectoire d'objets dans un liquide, tout cela avec uniquement une quantité minime d'informations préalables au sujet de l'objet et du système optique.

Contents

Introduction and Methods Part I

1 Introduction

1.1 Biomicroscopy

Optical microscopy is a powerful tool to comprehend biological systems, enabling researchers and physicians to acquire qualitative and quantitative data about cellular function, organ development, or diseases. However, light traveling through any imaging system undergoes diffraction, which leads to image blur [1]. This represents an intrinsic limit and the determining factor for the resolution of an optical instrument, and thus limits visual access to details. Indeed, the optical system only collects a fraction of the light emitted by any one point on the object, and cannot focus the light into a perfect image. Instead, the light spreads into a three-dimensional diffraction pattern. Image formation can be modeled as the convolution of the original object with a PSF, which is the image of an infinitely small point source in the object space and sums up the optical aberrations [2]. For thin, yet not flat, samples, the PSF remains shift-invariant within small areas of the 2D image, but the three-dimensional depth of the imaged object produces a local blur.

Using a PSF corresponding to the blur in a deconvolution algorithm can be used to both restore details in the image [3] and estimate its depth, which usually requires careful camera calibration [4], acquisition of focal depth stacks ([5], [6]), or coherent imaging, such as digital holographic microscopy [7], to numerically refocus the image. PSF estimation can be achieved by many techniques [8], but most of them are either dependent on a tedious calibration step, such as the experimental measurement of the PSF, or are sensitive to noise or image variability [9]. Blind Deconvolution (BD) techniques are deconvolution methods aimed at improving the image without prior knowledge of the PSF, the object or other optical parameters.

1.2 Deep Learning for microscopy

Machine learning techniques have improved, in a considerable number of fields, our ability to classify images [10], detect objects [11], describe content [12], and estimate image quality [13] (see Fig. 1.1). CNNs, in particular, are built for learning new optimal representations of

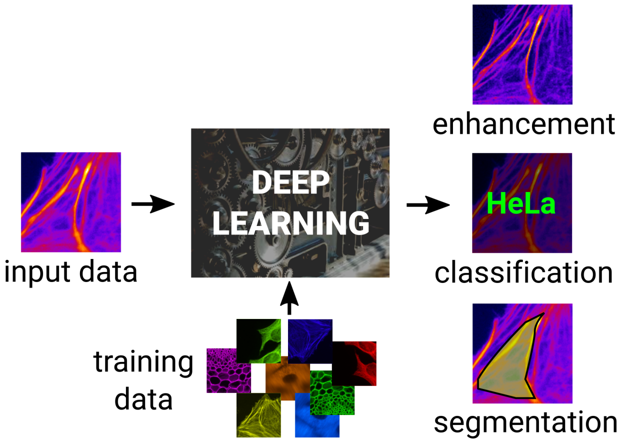


Figure 1.1: Deep learning methods overview.

image data and perform self-regulating feature extraction [14] and appear well adapted to determining the degradation kernel directly from the image texture. Our approach is similar to that of [15] but with PSF models that are tailored to the specificities of microscopy, a concept that we initially introduced in [16] and that has since been used by other groups such as [17]. In particular, we considered a more generic physical model that can accommodate large-support PSFs. CNNs were also used in a end-to-end manner to enhance details in biological images by performing supervised interpolation [18]–[20] or to emulate confocal stacks of sparse 3D structures from widefield images [21].

1.3 The data issue

The following section was published in “Free annotated data for deep learning in microscopy? A hitchhiker’s guide”, Photoniques 104, EDP Sciences, 2020.

Optical microscopy, despite being an invaluable tool in biology and medicine to observe and quantify cellular function, organ development, or disease mechanisms, requires constant trade-offs between spatial, temporal, and spectral resolution, invasiveness, acquisition time, and post-processing effort. As for other imaging fields, learning-based techniques are having a major impact in microscopy, where their potential to improve resolution, reduce invasiveness, or increase the speed of microscopy acquisitions has recently been demonstrated [22]. These techniques, in particular the ones that involve deep neural networks (DNNs), have benefited from the shift of intensive computing tasks to graphics processing units (GPUs) that has taken place over the last decade. Yet despite the ever-increasing computational power, it is often the lack of labeled training data that is the limiting factor for wide adoption. Annotating data is often a lengthy and expensive task, since it involves tedious work, generally by skilled experts. Annotation can be especially challenging in the case of three-dimensional images, common in microscopy, despite the development tools dedicated to this task [23], [24]. While the acquisition and annotation of volumetric data is common in some medical settings and for certain modalities because of their wide use for healthcare applications (e.g. magnetic resonance imaging (MRI), annotated by trained radiologists), bio-microscopy applications often lack similarly large and high-quality annotated volumetric datasets.

For many applications in biology, the time burden, cost, or physical feasibility of acquiring and annotating datasets for deep learning models *de novo* is simply out of the question. Can this requirement for annotated data be relaxed? Is it possible to borrow the knowledge gathered from datasets in other application fields, such as e-commerce or computer gaming, and leverage it for bio-microscopy? Specifically, could annotated datasets be generated from realistic synthetic models of tridimensional objects? Or could more abstract prior knowledge about the data be used to enhance the resolution? Here, we aim to provide an overview of some solutions that have emerged to tackle the problem of gathering sufficient annotated data to train learning-based methods in bio-microscopy. We have grouped the approaches in four broad categories: developing manual annotation strategies, learning from annotated images from other domains, building annotations from simulations, and using self-annotated data. This quest for annotated data is summarized in Fig. 1.2.

1.3.1 Learning from large training datasets

Image segmentation is a common computer vision task in which each pixel of an image is assigned a label corresponding to the object it belongs to. In microscopy, it is used, for example, to delineate, identify, and count cells. Learning-based segmentation methods require a training data set composed of images together with masks that correspond to the objects and their coordinates in the images. This annotation is usually performed manually and can be tedious. Other fields have integrated the task of annotating images into security forms on the web (to exclude robots from accessing content) or into entertaining game puzzles, such as to entice the public to provide quality annotations. Sullivan et al. have proposed to annotate microscopy data by a similar approach, as part of a multiplayer computer game named Eve Online [25]. Using the publicly available data set from the Cell Atlas of the Human Protein Atlas (HPA), they obtained, over one year, nearly 33 million classifications of subcellular localization patterns of immunostained proteins in 20 different organelles and cellular structures. The results were successfully used as a training dataset for a segmentation DNN.

1.3.2 Learning from other domains

Training a machine learning algorithm by re-using computer vision data sets that were originally intended for other tasks is at the core of transfer learning. This approach can overcome the lack of annotated data in one field (such as microscopy) if annotated data exists in a different field. A wide variety of annotated image datasets are available, such as ImageNet [10], MS-COCO [26] or Places [27], which contain foremost scenes depicting everyday objects and situations. In addition to allowing access to a large number of examples, learning from natural-scene images can help avoid learning on images that contain unwanted aberrations,

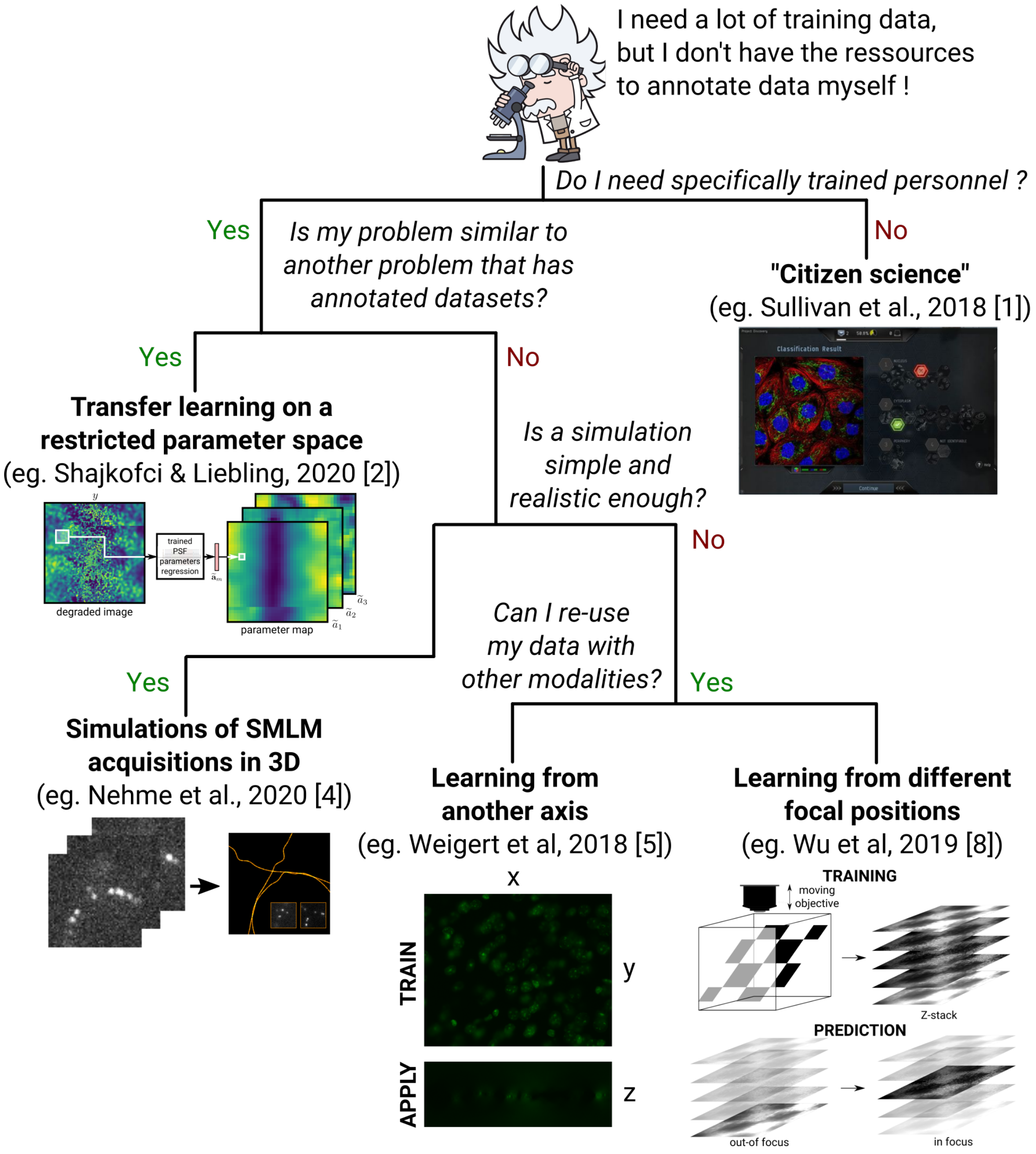


Figure 1.2: The quest for less painful annotation: overview of questions and methods. The illustrations were made by AS and adapted from the original papers.

such as blur and low-light noise commonly found in microscopy images. It becomes essential when developing methods that aim at removing aberrations, for example when using a DNN pipeline for deconvolution or denoising of microscopy images, since the ground truth of natural-scene photographs is more readily accessible that fine structures that challenge the resolution limits of the most powerful microscopes [28].

In most cases of transfer learning, pre-trained models are used as feature extractors [29], as the input of an unsupervised classifier such as support vector machines (SVM), or are fine-tuned with job-specific training data. However, there are some applications where transfer learning can be used without fine-tuning, specifically, when the feature space can be mapped identically in both domains. For image deconvolution and depth estimation, a description of the PSF is a pre-requisite and accurately determining PSF parameters is therefore essential. Recently, we formulated the problem of retrieving the physical PSF parameters of an optical microscope as a regression task for which we trained a DNN. Interestingly, when learning from data that consist of textured images, even if they are different from the end-use application (and possibly unrelated to microscopy), the trained model remains just as accurate as when it is trained on microscopy images [30]. The ability to train on generic data also helps prevent over-fitting the trained model to a specialized and narrow data set, which would make the model less suitable to be used in other situations. It also suggests the possibility of generalizing the trained network to other data types, provided that the new data and the data used for training share a common feature space.

1.3.3 Learning from simulated data sets

Generating annotated data from simulations is another effective approach to produce large and reliable training data sets. In some computer vision applications, such as autonomous car driving, data from simulated computer graphics 3D environments have been effectively used to train segmentation methods. Examples of such datasets include the Flying chairs dataset, or the GTA5 dataset derived from a computer game. Generally, the accuracy of DNN trained only on simulated data is poor, due to the extreme complexity of natural scenes that can hardly be reproduced with simple simulations.

In microscopy, image complexity remains fairly low in some modalities such as single-molecule localization microscopy (SMLM), where features consist of dots. The image processing task, for which DNN have been used, consists in converting images of random subsets of activated fluorophores, obtained over many consecutive diffraction-limited frames, into a high-precision point cloud. Data simulation of a realistic diffraction-limited ground truth is achievable, for example, by filtering the expected (punctate) objects by the optical PSF and take into account a realistic noise model [31]. Such approaches have allowed recent DNN methods such as DeepStorm3D [32] to recover densely overlapping PSFs of many emitting molecules over a large axial range and output a list of their 3D positions. The corresponding training data set were created by simulating a large number of images using randomly generated 3D patterns and the phase mask that governs the PSFs modelled on the physical implementation of the microscope. In order for the simulations to work as a sole training data, one possible way is to restrict the parameter space (i.e. the output of the network) by closely modeling the data generation model to the optics or choose a subset of plausible physical parameters ([30], [32]).

1.3.4 Learning from the input image itself

DNNs are also used for image enhancement such as deblurring (going from a blurry image to a sharper image), denoising (going from a noisy to a clean image), and for super-resolution (going from a low-resolution image to a higher-resolution image). These networks are traditionally trained on pairs of clean and distorted images. In microscopy, where the raw images often already reach the physical limits due to diffraction, a higher-resolution ground-truth image is simply not accessible. To cope with this problem, it is sometimes possible to use approaches that exploit particular features of the data, such as its isotropy or the availability of complementary imaging modalities.

Weigert et al. [18] proposed a pipeline aimed at restoring images using semi-synthetic training data. Specifically, it restores the axial resolution of volumetric images lost due to the axial elongation of the optical PSF and the low axial sampling rate. By assuming that similar features are to be expected regardless of sample orientation, the method leverages the fact that these features can be much better resolved in lateral views than in depth, hence the training to improve depth-resolution is done based on the latter images. Nevertheless, for many other applications, access to higher quality images or synthetic data is not available. To overcome this limitation, Krull et al. [33] developed a self-supervised training method for denoising based on the assumption that the noise is independent from pixel to pixel and that the true intensity of a pixel can be predicted from the local image context since it is not locally independent. The method involves a noise model whose probability distribution is learned from the training data and a network is trained to discriminate the underlying image from the noise. As a training set, a very small dataset of noisy images of the same type can be used. This method is virtually equivalent to training a DNN for every specific noise distribution.

A similar training scheme was also used in [20], [21], where the authors trained a generative adversarial network (GAN)-based DNN to transform an acquired low-resolution image into a high-resolution image using matched pairs of experimentally acquired images after registration and alignment. In [21], these pairs came from images of the same object using a confocal microscope and a super-resolved stimulated emission depletion (STED) microscope. In [20], the authors used a similar DNN to generate images that look as if they had been taken from another focal plane by training from images acquired at different heights in the sample. For both applications, the authors caution that the network must be (re)-trained for each specific image modality or experimental setup, as the methods do not produce ideal results otherwise. The application of such methods therefore remains somewhat limited to cases where the type of images and microscope settings are known beforehand and where a high number of similar images can be acquired, which could be particularly relevant for time-lapse imaging, high-resolution 3D stacks, or imaging of histological samples prepared under controlled and standardized conditions.

1.4 Roadmap

Deep learning technologies have enabled multiple applications that are transforming our day-to-day routines, including the way we approach microscopy. While limitations such as network capacity (can networks learn to predict from the wide variety of data types common in microscopy?), generalization (can a network trained on one type of data be used to handle other types of data?), and overfitting (is the network limited to predicting only what it has already seen?) are some pressing issues that the field is facing, the lack of good quality training data is likely the single most important aspect that affects accuracy and effectiveness of tasks such as enhancement, classification or segmentation. The most promising methods to overcome the scarcity of training data appear to leverage prior knowledge of the physical objects and image formation process [18], [32], [30], or of the noise distribution [33].

Even if we are still a long way from a blind pipeline that will enhance, classify or segment biological data without tedious annotation work, good knowledge of the problem and assumptions about the data allow scientist to already reap the benefits of deep learning tools by crafting adapted training sets without having to produce or wait for the availability of large annotated sets. In this thesis, we are looking for promising methods to circumvent this problem and trade for accuracy the knowledge of the optical path, as defined by the PSF. We call these methods “weakly supervised” or “semi-blind”, because they take advantage of information about the data (with a specific calibration step or a modeling step), but they do not require explicitly an experimental ground truth.

1.4.1 Problem statement

We aim to solve several problems. First, given only an observed degraded image $y(\boldsymbol{s})$, we want to estimate the PSF model $\widetilde{h}_{\boldsymbol{a}(\boldsymbol{s})}(\boldsymbol{r})$ closest to the effective PSF of the imaging system $h(\boldsymbol{s},\boldsymbol{r})$ for any point $\boldsymbol{s}$ *without* requiring additional information on the microscope or any further calibration images acquired with that microscope. Specifically, we want to infer the model parameters $\boldsymbol{a}(\boldsymbol{s})$, first locally, then globally. Then, we want to infer the local depth $z(\boldsymbol{r})$ along the axis of the object at any position $\boldsymbol{r}$ in the plane perpendicular to the optical axis, thereby allowing us to build $x_{3\mathrm{D}}(\boldsymbol{r}, z(\boldsymbol{r}))$. Next, given the local PSF parameters $\boldsymbol{a}(\boldsymbol{s})$ and the blurred image $y(\boldsymbol{s})$, we want to recover an estimate of the non-degraded image $x(\boldsymbol{r})$. Then, we want to integrate and calibrate $a_0(\boldsymbol{s})$ (the focus parameter) in an autofocus software. Finally, we want to find a transformation of the PSF model that could encode the tridimensional flow of particles in a fluid.

1.4.2 Method overview

For each of the problems, we summarize the following main steps:

1. Shift-invariant PSF parameter estimation given an image patch (see Chapter 2)
   (a) Select a parametric degradation model for $h_{\boldsymbol{a}}(\boldsymbol{r})$ allowing the generation of PSF/parameters pairs.
   (b) Gather a training library of microscopy images, degrade each image via a spatially-*invariant* convolution with its corresponding PSF, corrupt it with synthetic noise.
   (c) Train a CNN that takes a degraded image patch as input and returns the corresponding degradation model parameters, via regression.
   (d) Given a full microscopy image as input, locally extract a patch, then regress the PSF parameters using the steps above.
   (e) Repeat in all regions of the image.
   (f) Combine the estimated PSF parameters to generate the map $\boldsymbol{a}(\boldsymbol{s})$ of the local PSF model parameters.
2. Application 1: estimate depth from focus using PSF engineering (see Chapter 3))
   (a) Given an image acquired with a PSF-engineered optical system $y(\boldsymbol{s})$ [34]), deduce the depth map $z(\boldsymbol{r})$ from the local parameters $\boldsymbol{a}(\boldsymbol{s})$.
   (b) Generate $x_{3\mathrm{D}}(\boldsymbol{r}, z(\boldsymbol{r}))$.
3. Application 2: spatially-variant blind deconvolution (see Chapter 4)
   (a) Given an input image and the map of estimated local PSF parameters, generate local PSFs $\widetilde{h}_{\boldsymbol{a}(\boldsymbol{s})}(\boldsymbol{r})$.
   (b) Use the generated PSFs in a Total Variation regularized Richardson-Lucy (TV-RL) deconvolution algorithm to recover an estimate for $x(\boldsymbol{r})$.
4. Application 3: auto-focus using the focus parameter of the PSF model (see Chapter 5).
   (a) Given an image acquired $y(\boldsymbol{s})$, deduce the focus map $z(\boldsymbol{r})$ from the local parameters $\boldsymbol{a}(s_0)$.
   (b) Build a μManager plugin to iterate through the stack and refine the position z where $\boldsymbol{a}(s_0)$ is minimal.
5. Application 4: flow estimations of particles in a fluid (see Chapter 6).
   (a) Build a more advanced PSF model with parameters $\boldsymbol{a}$ that could encode three-dimensional speeds and starting positions.
   (b) With a single image acquired with a long exposure time, deduce the speed and starting position of moving particles in a fluid.

In the following chapters, we provide details on each of these steps, illustrated in Fig. 1.3.

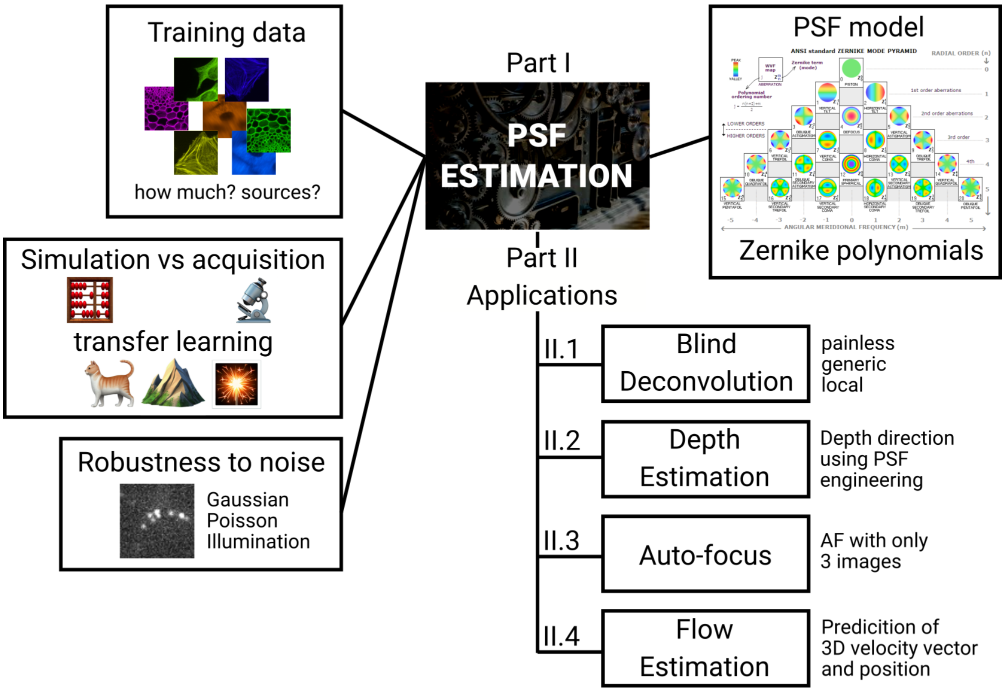


Figure 1.3: Outline of the thesis. Part I is focused on the PSF estimation process, while the Part II focuses on different applications.

2 Deep neural networks for PSF estimation

This chapter was published in A. Shajkofci and M. Liebling, “Spatially-Variant CNN-Based Point Spread Function Estimation for Blind Deconvolution and Depth Estimation in Optical Microscopy,” *IEEE Transactions on Image Processing*, vol. 29, pp. 5848–5861, 2020.

A.S. designed the model and the framework, conceived and carried out the experiments, A.S wrote the manuscript with support from M.L, M.L supervised the project.

2.1 Introduction

Researchers and physicians intensively use optical microscopes to observe and quantify cellular function, organ development, or disease mechanisms. Despite the availability of many volumetric imaging methods (in particular, optical sectioning methods), single-shot wide-field microscopy remains an important tool to image small and relatively shallow objects. However, non-flat areas, which are out of focus, lead to unsharp regions in the image, making localization and visual interpretation difficult. Image formation in a microscope can be modeled by light diffraction, which causes sharp point-like objects to appear blurry [1]. Because the optical system only collects a fraction of the light emanating from a point on the object, it cannot focus the light into a perfect point and, instead, spreads the light into a three-dimensional diffraction pattern described by the Point Spread Function (PSF). As the image is formed by superposing the contribution of all points in the object, knowledge of the local diffraction pattern, which sums up the optical system and its aberrations, can be used to estimate a sharper image [2].

For thin, yet not flat samples, image formation can be modeled as a superposition of 2D PSFs. These are shaped both by the optical system and the three-dimensional depth of the object. Knowledge of the local PSF could therefore both be used to recover the image and estimate its depth, which usually requires careful camera calibration and ad-hoc focus estimation [4], acquisition of focal depth stacks ([5], [6]), or coherent imaging, such as digital holographic microscopy [7], to numerically refocus the image. Using an adequate PSF, i.e. one that corresponds to the blur, in a deconvolution algorithm can restore details in the image [3]. PSF estimation can be achieved by many techniques [8], but most of them are either dependent on a tedious calibration step, such as the experimental measurement of the PSF, or are sensitive to noise or image variability. Blind Deconvolution (BD) techniques are methods able to recover features in the image without prior knowledge of the PSF.

Here, we aim at estimating the local PSF only from the acquired image and use it to reverse the local degradation due to the optical system. Furthermore, we aim at estimating the depth of any location on the surface of a thin object with respect to the focal plane. We rely on a model-based approach that retrieves the PSF given a degraded image patch via a machine learning approach.

Machine learning technologies have improved our ability to classify images [10], detect objects [11], describe content [12], and estimate image quality [13]. Convolutional Neural Networks (CNNs), in particular, have the ability to learn correlations between an image input and a defined outcome and appear well adapted to determining the degradation kernel directly from the image texture. A similar reasoning led to recent results by Zhu et al. [35], Sun et al. [15], Gong et al. [36] and Nah et al. [37], where the direction and amplitude of motion blur was determined by a CNN classifier from images blurred with a Gaussian kernel. Our approach is similar to that of [15] but with PSF models that are tailored to the specificities of microscopy, a concept that we initially introduced in [16] and that has since been used by other groups such as [17]. In particular, we considered a more generic physical model that can accommodate large-support PSFs. CNNs were also used in a end-to-end manner to enhance details in biological images by performing supervised interpolation [18]–[20] or to emulate confocal stacks of sparse 3D structures from widefield images [21].

In this paper, we propose a method to:

1. Find the spatially-variant PSF of the degraded image of a thin, non-flat object directly from the image texture without any instrument-specific calibration step. The PSF determination technique is derived from the one we proposed in [16], which recovers local Zernike moments of the PSF. We focus here on improving the degradation model and quantitatively assess the robustness of the method.
2. Deconvolve the image in a blind and spatially-variant manner, using a regularized Richardson-Lucy algorithm with an overlap-add approach.
3. Extract the depth of a three-dimensional surface from a single two-dimensional image using combinations of Zernike moments.

This technique allows us to enhance the acquired image and recover the three-dimensional structure of a two-dimensional manifold in a 3D space using a single 2D image as an input.

This paper is organized as follows. In Section 2.2, we present the method, comprising the image formation model, the degradation model, the data set generation process, the different neural networks to be trained, the PSF mapping, the deconvolution algorithm, and the depth from focus algorithm. Then, in Section 2.3, we characterize the regression performance of the

CNN for different modalities, as well as the gain in resolution from the deconvolution, and the precision of depth detection. We then discuss our findings in Section 2.4 and conclude in Section 2.5.

2.2 Methods

2.2.1 Object and image formation model

We consider a two-dimensional manifold in 3D-space with local intensity $x(\boldsymbol{r})$, e.g. an infinitely thin sample suspended in a gel, which can be parameterized by the lateral coordinates $\boldsymbol{r} = (r_1, r_2)$ and axial coordinate $z(\boldsymbol{r})$. We express the resulting three-dimensional object as:

$$x_{3D}(\boldsymbol{r}, r_3) = x(\boldsymbol{r})\delta(r_3 - z(\boldsymbol{r})), \tag{2.1}$$

where $(\boldsymbol{r}, r_3) = (r_1, r_2, r_3)$ are coordinates in 3D object space. We further consider an optical imaging system with camera coordinates $\boldsymbol{s} = (s_1, s_2)$ and axial position s_3, characterized by a spatially-varying point spread function $h_{\mathrm{3D}}(\boldsymbol{s}, s_3, \boldsymbol{r}, r_3)$ (see Fig. 2.1). For a fixed axial camera position s_3, the measured intensity by a pixel at position $\boldsymbol{s}$ is given by the convolution (Fig. 2.1 (a)) [38]:

$$y_{\mathrm{3D}}(\boldsymbol{s}, s_3) = \iiint x_{\mathrm{3D}}(\boldsymbol{r}, r_3) h_{\mathrm{3D}}(\boldsymbol{s}, s_3, \boldsymbol{r}, r_3) \mathrm{d}\boldsymbol{r} \mathrm{d}r_3, \tag{2.2}$$

where we assumed, to simplify the notation, that the magnification is 1. When the microscope is focused at the origin ($s_3 = 0$) we define the 2D image $y(\boldsymbol{s}) = y_{\mathrm{3D}}(\boldsymbol{s}, s_3)\big|_{s_3=0}$, which can be obtained via the expression:

$$\begin{aligned} y(\boldsymbol{s}) &= \iiint x(\boldsymbol{r})\delta(r_3 - z(\boldsymbol{r})) h_{\mathrm{3D}}(\boldsymbol{s}, s_3 = 0, \boldsymbol{r}, r_3) \mathrm{d}\boldsymbol{r} \mathrm{d}r_3 && (2.3) \\ &= \iint x(\boldsymbol{r}) h_{\mathrm{3D}}(\boldsymbol{s}, s_3 = 0, \boldsymbol{r}, r_3 = z(\boldsymbol{r})) \mathrm{d}\boldsymbol{r} && (2.4) \\ &= \iint x(\boldsymbol{r}) h(\boldsymbol{s}, \boldsymbol{r}) \mathrm{d}\boldsymbol{r}, && (2.5) \end{aligned}$$

where $h(\boldsymbol{s}, \boldsymbol{r}) = h_{\mathrm{3D}}(\boldsymbol{s}, s_3 = 0, \boldsymbol{r}, r_3 = z(\boldsymbol{r}))$ is a 2D point spread function that incorporates both the local (3D) variations of the optical system and the variable depth of the thin sample. We further assume that $h(\boldsymbol{s}, \boldsymbol{r})$ can be approximated by a parametric function $\widetilde{h}_{\boldsymbol{a}(\boldsymbol{s})}(\boldsymbol{r})$, where the N parameters $\boldsymbol{a}(\boldsymbol{s}) = (a_1(\boldsymbol{s}) \quad a_2(\boldsymbol{s}) \quad \cdots \quad a_N(\boldsymbol{s}))$ can vary for every 2D location $\boldsymbol{s}$ of the image (Fig. 2.1 (b)).

2.2.2 Parametric degradation models

There are many methods for estimating the PSF of a degraded image. Such methods can be categorized into two classes: direct PSF estimation or parametric modeling. In works such as those by Grossmann et al. [4], the PSFs are estimated directly from the image (e.g. using edge detection and a regression model [39], a Maximum a Posteriori (MAP) prediction [40]), or via a camera calibration using images of a defined and known pattern [41] [42]. Levin et al. [43]

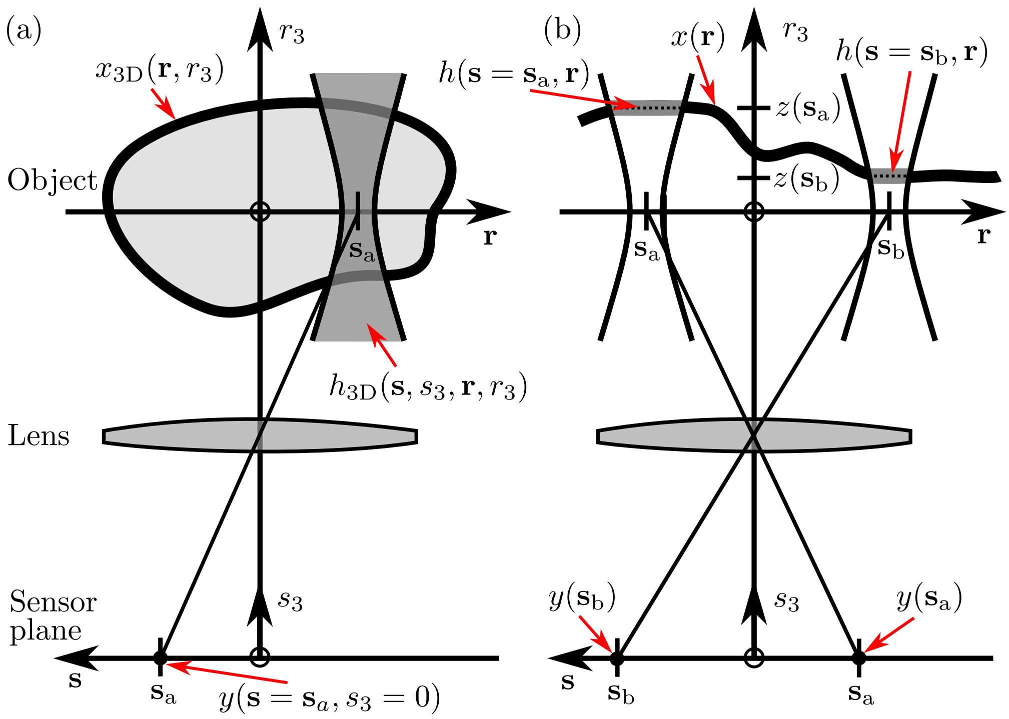

(a)
r_3
$x_{3\mathrm{D}}(\mathbf{r}, r_3)$
Object
$\mathbf{s}_\mathrm{a}$
$\mathbf{r}$
$h_{3\mathrm{D}}(\mathbf{s}, s_3, \mathbf{r}, r_3)$
Lens
Sensor plane
s_3
$\mathbf{s}$
$\mathbf{s}_\mathrm{a}$
$y(\mathbf{s} = \mathbf{s}_a, s_3 = 0)$
(b)
$h(\mathbf{s} = \mathbf{s}_\mathrm{a}, \mathbf{r})$
$x(\mathbf{r})$
r_3
$h(\mathbf{s} = \mathbf{s}_\mathrm{b}, \mathbf{r})$
$z(\mathbf{s}_\mathrm{a})$
$z(\mathbf{s}_\mathrm{b})$
$\mathbf{s}_\mathrm{a}$
$\mathbf{s}_\mathrm{b}$
$\mathbf{r}$
$y(\mathbf{s}_\mathrm{b})$
s_3
$y(\mathbf{s}_\mathrm{a})$
$\mathbf{s}$
$\mathbf{s}_\mathrm{b}$
$\mathbf{s}_\mathrm{a}$

Figure 2.1: Object and image formation model. (a) General thick 3D object case, (b) Approximate manifold case.

showed that a MAP approach to recover the blur kernel is well constrained, but that the MAP global optimum for the recovered image is a blurred image because the strong constraints do not always generalize to unexpected or noisy types of data [44], which are common in microscopy images. Full pixel-wise PSFs can also be estimated using dictionary learning [45], or CNNs [46]. However, this latter kind of estimation is not well constrained and can generate over-fitting artifacts.

In contrast, parametric modeling of the PSF allows to reduce the dimensionality of the optimization problem and to attach a physical meaning to the parameters, such as the relative distance from the focal point, or optical aberrations such as astigmatism. There are many mathematical models to represent the PSF of a microscope. They can take into account both physical characteristics of the objectives (for example numerical aperture, correction types, etc.) and of the experimental conditions (focal distance and immersion medium) [47]. In many cases, the model parameters correspond to physical design conditions, such as optical distances, aperture diameters, or foci. A simple PSF model can be obtained from the Fraunhofer diffraction theory to calculate the diffraction of a circular aperture [48]. The *Gibson & Lanni* model accounts for the immersion medium, the cover-slip, the sample layers, different medium numerical apertures, and the properties of the objective [1]. Despite their theoretical relevance, in practice, numerical values for these parameters may not be available, as detailed information about all experimental and design conditions may be lacking. Even if the parameters are accessible, they may be missing if they were not recorded along with the image.

Since we aim to recover the PSF from the image itself, with minimal knowledge of the imaging conditions, we focused on models specified by only a small number of parameters or whose complexity can be adjusted progressively by considering approximations with a subset of the complete set of parameters. Specifically, we considered PSF models based on Zernike polynomials, which are used to describe the wavefront function of lenses such as the eye [49], as well as anisotropic Gaussian models.

Zernike polynomial decomposition of the pupil function

Optical abnormalities, such as de-focus, astigmatism, or spherical aberrations, can be modeled with a superposition of Zernike polynomials $Z_n(\boldsymbol{\xi})$ in the expansion of the microscope objective's pupil function $W_{\boldsymbol{a}(\boldsymbol{s})}$ [50]:

$$W_{\boldsymbol{a}(\boldsymbol{s})}(\boldsymbol{\xi}) = \sum_{n=1}^{N} Z_n(\boldsymbol{\xi}, a_n(\boldsymbol{s})), \tag{2.6}$$

where $\boldsymbol{\xi}$ denotes the two-vector of spatial coordinates in the pupil plane perpendicular to the optical axis, N the maximal order of considered aberrations, and $a_n(\boldsymbol{s})$ the parameter corresponding to each Zernike term Z_n. In our experiments, Z_1 describes the de-focus term, Z_2 describes the power of the astigmatism (cylinder), and Z_3 describes the astigmatism angle (axis). The pupil function can ultimately be converted into a PSF [51]:

$$\widetilde{h}_{\boldsymbol{a}(\boldsymbol{s})}(\boldsymbol{r}) \propto \left| \mathscr{F}\left\{ W_{\boldsymbol{a}(\boldsymbol{s})}\left(\frac{\boldsymbol{r}}{\lambda}\right)\right\}\right|^2, \tag{2.7}$$

with $\mathscr{F}$ the Fourier transform and λ the wavelength of the light.

Anisotropic Gaussian model

For many applications, Gaussian distributions are sufficiently accurate approximations of the diffraction-limited PSF of wide field microscopes [52]. We extend the model by allowing for anisotropy, which we require to describe astigmatic aberrations, or if the spatial resolution in one lateral direction is different from that in the other. The detection method is in that case similar to the one described in [15]. The PSF is then defined by the anisotropic zero-centered normal probability density function:

$$\widetilde{h}_{\boldsymbol{a}(\boldsymbol{s})}(\boldsymbol{r}) = \frac{1}{\sqrt{2\pi a_1(\boldsymbol{s})}} \exp\left(-\frac{1}{2}\frac{{r_1}^2}{a_1(\boldsymbol{s})}\right) \cdot \frac{1}{\sqrt{2\pi a_2(\boldsymbol{s})}} \exp\left(-\frac{1}{2}\frac{{r_2}^2}{a_2(\boldsymbol{s})}\right), \tag{2.8}$$

where a_1 and a_2 are the variances of the Gaussian in the x and y axes, respectively.

2.2.3 Problem statement

We aim to solve several problems. First, given only an observed degraded image $y(\boldsymbol{s})$, we want to estimate the PSF model $\widetilde{h}_{\boldsymbol{a}(\boldsymbol{s})}(\boldsymbol{r})$ closest to the effective PSF of the imaging system $h(\boldsymbol{s},\boldsymbol{r})$ for any point $\boldsymbol{s}$ *without* requiring additional information on the microscope or any further calibration images acquired with that microscope. Specifically, we want to infer the model parameters $\boldsymbol{a}(\boldsymbol{s})$, first locally, then globally. Next, given the local PSF parameters $\boldsymbol{a}(\boldsymbol{s})$ and the blurred image $y(\boldsymbol{s})$, we want to recover an estimate of the non-degraded image $x(\boldsymbol{r})$. Finally, we want to infer the local depth $z(\boldsymbol{r})$ along the axis of the object at any position $\boldsymbol{r}$ in the plane perpendicular to the optical axis thereby allowing us to build $x_{3\mathrm{D}}(\boldsymbol{r}, z(\boldsymbol{r}))$.

2.2.4 Method overview

For each of the problems, we summarize the following main steps:

1. Shift-invariant PSF parameter estimation given an image patch (see Section 2.2.5)
   (a) Select a parametric degradation model for $h_{\boldsymbol{a}}(\boldsymbol{r})$ allowing the generation of PSF/parameters pairs.
   (b) Gather a training library of microscopy images, degrade each image via a spatially-*invariant* convolution with its corresponding PSF, corrupt it with synthetic noise.
   (c) Train a CNN that takes a degraded image patch as input and returns the corresponding degradation model parameters, via regression.
2. Local PSF estimation given a full degraded microscopy image (see Section 2.2.6)
   (a) Given a full microscopy image as input, locally extract a patch, then regress the PSF parameters using the steps above.
   (b) Repeat in all regions of the image.
   (c) Combine the estimated PSF parameters to generate the map $\boldsymbol{a}(\boldsymbol{s})$ of the local PSF model parameters.

In the following subsections, we provide details on each of these steps.

2.2.5 PSF parameter estimation in image patches (shift-invariant image formation model)

Data set generation for CNN training

Given an image patch as input, we wish to estimate the degradation model parameters corresponding to the spatially-invariant PSF that degraded the patch. Since neural networks are trained by adjusting their internal weights using backpropagation of the derivative of a loss function between the ground truth of the training data and the output of the network [53], we establish a training set T (Fig. 2.2). For that purpose, we gather input images $x^{(k)}$. Since reliable public image data sets have limited size, we augment the size of the training sets by rotating each image by an angle of ±90°, so we have K images, $x^{(k)}$, $k = 0, ..., K-1$.. We induce a synthetic degradation by convolving the images with K generated PSFs $h_{\boldsymbol{a}^{(k)}}$ of model parameters $\boldsymbol{a}^{(k)}$ drawn from a normalized and scaled uniform distribution:

$$\psi^{(k)}(\boldsymbol{s}) = (h_{\boldsymbol{a}^{(k)}} * x^{(k)})(\boldsymbol{s}). \tag{2.9}$$

We also consider the two predominant sources of noise in digital image acquisition: the stochastic nature of the quantum effects of the photoconversion process and the intrinsic thermal and electronic fluctuations in the CCD camera [54]. The first source of noise comes from physical constraints such as a low-power light source or short exposure time, while the second is signal-independent. This motivates the noise model as a mixed Gaussian-Poisson noise process. Therefore, we define noise with the two following components:

- A random variable $n_p(\boldsymbol{s}) \sim \mathscr{P}(\lambda = \psi^{(k)}(\boldsymbol{s}))$ following a Poisson distribution of probability $P\{n_p(\boldsymbol{s}) = i\} = e^{-\lambda}\lambda^i / i!$.
- A random variable $b(\boldsymbol{s}) \sim \mathscr{N}(0, \sigma^2)$ following a Gaussian distribution with zero-mean and variance σ^2.

The image noise model for data set generation is then:

$$\psi^{(k)}_{\text{noisy}}(\boldsymbol{s}) = \beta n_p(\boldsymbol{s}) + b(\boldsymbol{s}), \tag{2.10}$$

with β a number between 0 and 1 reflecting the quantum efficiency of the CCD [55]. Images that did not comply with a minimal variance and white pixel ratio were tagged as invalid, i.e. $a_0^{(k)} = 1$ (see Section 2.2.5).

We cropped the images $\psi^{(k)}_{\text{noisy}}(\boldsymbol{s})$ to a size $K_\psi \times L_\psi$ by randomly selecting the position of a region of interest of that size. We then paired these image patches with their respective PSF parameter vector $\boldsymbol{a}^{(k)}$ in order to form the training set $T = \{(\psi^{(k)}_{\text{noisy}}(\boldsymbol{s}), \boldsymbol{a}^{(k)})\}_{k=0}^{K-1}$.

CNN training modalities

We considered several neural networks (whose architectures we further describe below) and trained them to learn the PSF model parameters described in Section 2.2.2. The task of the network is to estimate, only from the k^{th} input image patch $\Psi^{(k)}(\boldsymbol{s})$, the parameters $\widetilde{\boldsymbol{a}}^{(k)}(\boldsymbol{s})$ that have been used by the PSF model to degrade that input image. Since there are cases where the PSF estimation is not possible, e.g. where the sample lacks texture, such as in uniformly black or grey areas, we added a boolean parameter $a_0^{(k)}$ (whose values can be either 0 or 1), which indicates the legitimacy of the sample. The total number of estimated parameters is then $N+1$. We aim at minimizing the distance between the output of the network $\widetilde{\boldsymbol{a}}^{(k)}$ and the

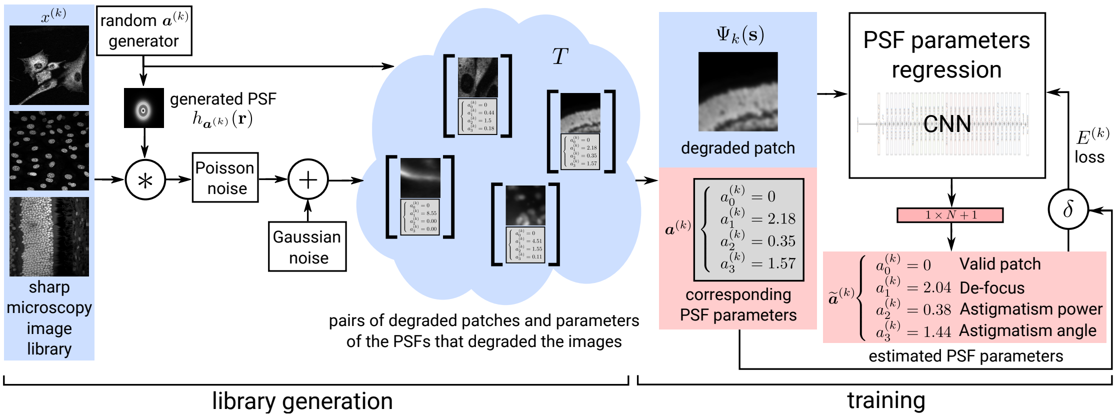

$x^{(k)}$
sharp microscopy image library
random $\boldsymbol{a}^{(k)}$ generator
generated PSF $h_{\boldsymbol{a}^{(k)}}(\mathbf{r})$
$*$
Poisson noise
$+$
Gaussian noise
T
pairs of degraded patches and parameters of the PSFs that degraded the images
$\Psi_k(\mathbf{s})$
degraded patch
$\boldsymbol{a}^{(k)}$
$a_0^{(k)} = 0$
$a_1^{(k)} = 2.18$
$a_2^{(k)} = 0.35$
$a_3^{(k)} = 1.57$
corresponding PSF parameters
PSF parameters regression
CNN
$1 \times N + 1$
$E^{(k)}$ loss
δ
$\tilde{\boldsymbol{a}}^{(k)}$
$a_0^{(k)} = 0$ Valid patch
$a_1^{(k)} = 2.04$ De-focus
$a_2^{(k)} = 0.38$ Astigmatism power
$a_3^{(k)} = 1.44$ Astigmatism angle
estimated PSF parameters
library generation
training

Figure 2.2: Data set gathering (left) and CNN training (right) pipelines. From a large library of sharp microscopy images, small patches are created, blurred with a PSF generated from random parameters, and degraded with a Poisson-Gaussian noise mixture (see Section 2.2.5). The resulting patches and the parameters are stored in the training set T, that is used for training the CNN. Using backpropagation of the loss function, the CNN output is trained towards the prediction of the PSF model parameters (Section 2.2.5).

ground-truth PSF parameters $\boldsymbol{a}^{(k)}$. Therefore, in the training phase, we updated the weights of the CNN using the modified Euclidean loss function:

$$E^{(k)} = \gamma \left(a_0^{(k)} - \tilde{a}_0^{(k)} \right)^2 + \frac{1 - a_0^{(k)}}{2N} \sum_{n=1}^{N} (a_n^{(k)} - \tilde{a}_n^{(k)})^2, \tag{2.11}$$

with γ a hyperparameter regulating the importance of the validity parameter, that we set to 1 in our further experiments.

We choose to rely on networks that showed good performance in the ImageNet competition, which is a benchmark in object classification on hundreds of categories [56], [57]. Donahue et al. [58] showed that deep convolutional representations can be applied to a variety of tasks and detection of visual features, which drove our selection for estimating optical aberrations. Hendrycks et al. [59] extensively discussed whether the networks were robust to changes in input illumination, noise, and blur. While residual networks that use skip-connections such as ResNet [60] appear to be more robust to input noise than primitive feedforward networks such as AlexNet [10], their performance appears to be surpassed by newer multibranch models such as ResNeXt [61] or Densenet [62]. In the context of our specific task, we compared the performance (see Section 2.3.2) and the robustness to degradation (see Section 2.3.3) of several of the above architectures.

After training, the networks can regress the spatially-invariant PSF parameters $\tilde{\boldsymbol{a}}^{(k)}(\boldsymbol{s})$ from a single input image patch $\Psi^{(k)}(\boldsymbol{s})$.

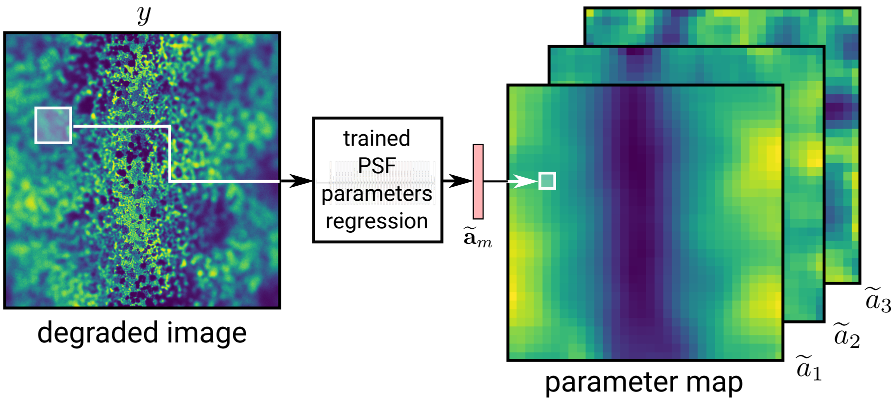


Figure 2.3: Spatially-variant PSF parameter mapping using a sliding window over the acquired image, as input of a convolutional neural network.

2.2.6 Spatially-variant PSF parameter mapping

Given a trained CNN that is able to recover the degradation parameters from a single image patch, we now turn to the problem of locally estimating the parameters of the different PSFs that degraded a larger input image. To achieve this, we use an overlapping sliding window over the input image $x(\boldsymbol{r})$ with stride t that is fed into the locally invariant regression CNN trained in Section 2.2.5 (see Fig. 2.3). We store the resulting parameters in the map $\widetilde{\boldsymbol{A}} = (\widetilde{\boldsymbol{a}}^{(0)} \quad \widetilde{\boldsymbol{a}}^{(1)} \quad \dots \quad \widetilde{\boldsymbol{a}}^{(M)})$, where $\widetilde{\boldsymbol{a}}^{(m)}$ is the output of the neural network for patch m and M is the total number of patches. Using the PSF model, we generate from $\widetilde{\boldsymbol{A}}$ a spatially-variant map of local PSF kernels defined as $\widetilde{H} = (\widetilde{h}^{(0)}(r) \quad \widetilde{h}^{(1)}(r) \quad \cdots \quad \widetilde{h}^{(M)}(r))$.

The overlapping window over the input image yields a map of

$$\left(\lfloor (K_x - K_\psi)/t \rfloor + 1\right) \times \left(\lfloor (L_x - L_\psi)/t \rfloor + 1\right) \tag{2.12}$$

kernels, with K_x, L_x, and K_ψ, L_ψ being the width and height of the input image and the window step size, respectively. For example, a 1024 × 1024 pixel input image using 128 × 128 pixel patches and $t = 64$ yields $M = 13 \times 13$ spatially-dependent PSF kernels. We fill every patch with a validity parameter $a_0 = 1$ (i.e. invalid) with the content of the inverse Euclidean distance-weighted average of the four-connected nearest neighbors using K-Nearest Neighbor regression [63] in order to avoid boxing artifacts during the later deconvolution process.

2.3 Experiments

In this section we report our efforts to characterize the performance of our method as well as its dependency to several hyper-parameters, such as the choice of the PSF model, the neural network architecture, the training set size, or its content. We furthermore tested the regression performance of the PSF parameter regression and its robustness to Signal-to-Noise Ratio (SNR) degradation and the absence of texturing. Finally, we also assessed the quality of deconvolved images and the accuracy of the estimated depths.

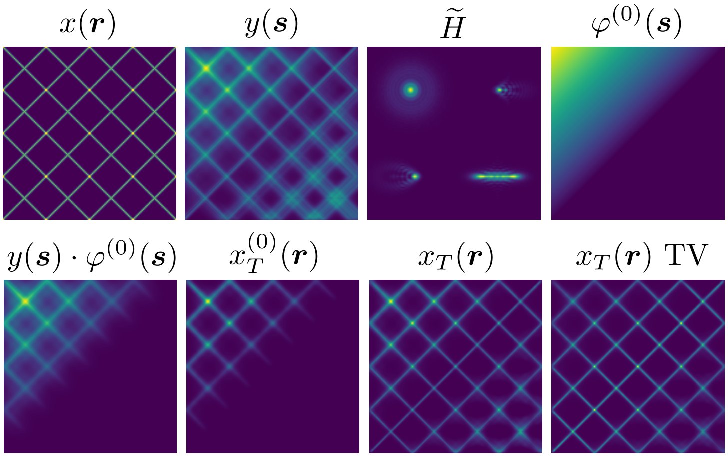


Figure 2.4: Synthetic experiment involving variables and results of Eq. (4.2). Starting from a ground truth image ($x(\boldsymbol{r})$), a local map of PSF (smooth interpolation between the 4 shown PSFs) and local weight combination $\boldsymbol{\varphi}(\boldsymbol{s})$, we generated a blurred image y. The deconvolution method (similar to [64] and [65]) starts from a map of locally estimated PSFs to deconvolve and recombine into a single image $x_T(\boldsymbol{r})$ and $x_T(\boldsymbol{r})$ TV.

2.3.1 Infrastructure

Our PSF parameter estimation method depends on three main variables: the content and size of training data sets, the PSF parametric model, and the neural network architecture. We briefly describe the different options below.

Training, validation and test data sets for CNN regression performance

We gathered images from four different data sources:

1. **[micr]** microscopy images collected from [66], [67] and [68],
2. **[nat]** common images from the MIT Places365 data set [27] that gathers natural and man-made photographs,
3. **[poi]** synthetic images of points on a black background (Fig. 2.5),
4. **[syn]** synthetic images of cells (Fig. 2.5).

The rationale for using natural and synthetic images is that these data sources are much more abundant than microscopy images, often sharper and royalty free, making it possible to quickly assemble a large dataset. We combined these data to generate six different data sets (Table 2.1) and prepared the library as described in Section 2.2.5. We randomly selected two times 10,000 images to form a validation set and a test dataset that the networks never use during the training process. The validation dataset is used for selecting the best learning rate and early

Table 2.1: Name and size of the different training, validation and test data sets as input of the CNN.

Data set	K_{train}	K_{valid}	K_{test}
syn	440,000	10,000	10,000
poi	330,000	10,000	10,000
micr	2,700,000	10,000	10,000
micrsm	270,000	10,000	10,000
nat	2,700,000	10,000	10,000
micr-syn-poi	3,470,000	10,000	10,000

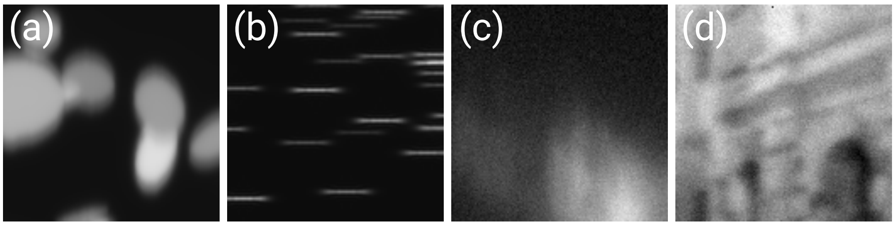


Figure 2.5: Examples of degraded input patches from different data sources: (a) synthetic cells [syn], (b) synthetic points [poi], (c) microscopy images [micr], (d) natural images [nat]. The images have been degraded by the Zernike-3 PSF model (see Table 2.2) and noise as described in Section 2.2.5

.

stopping epoch for every training, while the testing set is used for performance assessment. We added synthetic black images to every data set to avoid misdetection of non-textured parts of the image and explicitly set $a_0^{(k)} = 1$ for these samples.

PSF models and parameters

We considered two different PSF model types: Zernike polynomials (Section 2.2.2) with $N = 1, 2$, or 3 parameters, and Gaussian (Section 2.2.2), with either $N = 1$ or 2 parameters, as described in Table 2.2.

Table 2.2: PSF models selected for data set generation (Section 2.2.5), with the number and name of free parameters.

PSF model	N	Parameters
Zernike-1 (Z-1)	1	focus
Zernike-2 (Z-2)	2	cylinder, axis
Zernike-3 (Z-3)	3	focus, cylinder, axis
Gaussian-1 (G-1)	1	width
Gaussian-2 (G-2)	2	width x axis, width y axis

CNN architectures and training modalities

We compared two residual neural networks architectures trained from scratch: 34-layer ResNet [60] and 50-layer ResNeXt [61]. They were adapted for accepting normalized gray-scale input image patches of size $W_\psi \times H_\psi = 128 \times 128$ pixels. Additionally, we fine-tuned, using our training dataset, the same network already trained on the ImageNet data set (available on the PyTorch website). For the latter model, we re-scaled the input images to the network input size using bilinear interpolation. We trained the models for 20 epochs with PyTorch 1.0 using the Adam optimizer [69] and a learning rate between 0.001 and 0.01 defined by the validation set performance.

2.3.2 Characterization of the CNN regression performance

We analyzed the performance of our system for regressing the PSF parameters. The metrics we used to assess the performance of the network is the goodness-of-fit of the parameter estimation compared to the ground truth. We quantified it in terms of the squared Pearson correlation coefficient R^2 averaged over all PSF parameters:

$$R^2 = \frac{1}{N} \sum_{n=0}^{N} \left(1 - \frac{\sum_{k=0}^{K_{\text{test}}-1} (a_n^{(k)} - \tilde{a}_n^{(k)})^2}{\sum_{k=0}^{K_{\text{test}}-1} (a_n^{(k)} - \bar{a}_n)^2} \right), \tag{2.13}$$

with K_{test} the number of samples in the test set. We calculated the correlation coefficient only for samples that contained texture in the ground-truth (i.e. when $a_0^{(k)} = 0$) and discarded the others.

Characterization of CNN regression performance when training and test data set types are the same

We started by assessing the performance of the CNNs when the test set is made of the same image type as the training set. Table 2.3 summarizes the performance of the regression of test data for every combination of training data sets (Table 2.1), PSF models (Table 2.2) and CNN architectures.

Variables: data set type, PSF model type, network architectures.
Fixed: the data set type is the same for training and testing.
Evaluation criterion: R^2 between the degradation parameters used to generate the test image and the parameters recovered by the CNN.

In most cases, the correlation coefficient is superior to 80%, which indicates a very good degree of overall correlation. The worst cases are with models trained for Zernike-3, that yield $0.61 < R^2 < 0.96$. We notice a few differences in the regression performances between Gaussian and Zernike models. Indeed, images blurred with a Gaussian model tend to be better

Table 2.3: Results of regression analyses (in terms of R^2) for $N = 10,000$ test images in data sets shown in Table 2.1 using the same data type for training, validation (model selection) and test.

				ResNet-34			
		[syn]	[poi]	[micr]	[micrsm]	[nat]	[micr-syn-poi]
PSF models	Z-1	0.99	0.99	0.98	0.73	0.98	0.68
	Z-2	0.67	0.99	0.81	0.80	0.80	0.79
	Z-3	0.95	0.98	0.78	0.58	0.84	0.88
	G-1	0.99	0.99	0.98	0.92	0.92	0.99
	G-2	0.99	0.99	0.99	0.97	0.91	0.99

				ResNet-34-pretrained			
		[syn]	[poi]	[micr]	[micrsm]	[nat]	[micr-syn-poi]
PSF models	Z-1	0.99	0.99	0.99	0.89	0.99	0.81
	Z-2	0.93	0.97	0.92	0.81	0.94	0.90
	Z-3	0.97	0.89	0.95	0.77	0.80	0.89
	G-1	0.94	0.99	0.98	0.99	0.94	0.99
	G-2	0.99	0.99	0.99	0.98	0.95	0.99

				ResNext-50			
		[syn]	[poi]	[micr]	[micrsm]	[nat]	[micr-syn-poi]
PSF models	Z-1	0.98	0.99	0.97	0.69	0.97	0.65
	Z-2	0.69	0.98	0.85	0.74	0.90	0.85
	Z-3	0.64	0.94	0.90	0.72	0.61	0.85
	G-1	0.99	0.99	0.97	0.80	0.91	0.98
	G-2	0.99	0.99	0.99	0.92	0.97	0.98

recognized by the neural network, with $R^2 > 0.90$, than images blurred with a Zernike model that fluctuates around $0.60 < R^2 < 1.00$. When looking at the performance of a smaller [micr] training data set compared to the full [micr] data set, we notice that the performance of the smaller data set is always worse or equal, no matter which CNN model or PSF model used. Finally, we observe that the overall performance of ResNext-50 is lower than the performance of both ResNets.

Characterization of CNN regression performance when the type of training and test data set differ

We assessed the robustness of our regression method when the system is tested on image types other than those it has been trained for.

Variables: data set types for both training and test sets.
Fixed: the network architecture (ResNet-34), the PSF model (Gaussian-2). The model is

Table 2.4: Evaluation (in terms of R^2) for $N = 10,000$ images input into a Resnet-34-pretrained network trained for regression of the Gaussian-2 PSF model parameters with different training set / test sets pairs. The learning rate and epoch were selected using results from a test set of $N = 10,000$ separate images.

	Test data sets					
Training data sets	[syn]	[poi]	[micr]	[micrsm]	[nat]	[micr-syn-poi]
[syn]	1	0.93	0.76	0.83	0.81	0.85
[poi]	0	1	0	0	0	0
[micr]	0.97	0.97	0.99	0.99	0.99	0.99
[micrsm]	0.97	0.95	0.95	0.98	0.89	0.97
[nat]	0.99	0.80	0.96	0.93	0.96	0.96
[micr-syn-poi]	1	0.99	0.99	0.99	0.94	0.99

already trained and selected using an independent validation dataset.
Evaluation criterion: R^2 between the degradation parameters used to generate the test input and the parameters recovered by the CNN.

Table 2.4 gathers the regression performance obtained using a Gaussian-2 PSF model and the ResNet-34 network, with training and testing data sets of different types. The regression is robust to different train and test data set types ($R^2 < 0.90$) except when the CNN is trained with [poi] and, to a lesser extent, with [syn]. Surprisingly, networks trained on natural ([nat]) images perform as well as networks trained on microscopy images.

2.3.3 Robustness of PSF regression against input degradation

Degradations on the input images are unavoidable in biological environments. Indeed, microscopes are often used for a variety of sample types and preparations and are calibrated by different people. Settings such as illumination brightness, exposure time, and contrast frequently change or are operator-dependent. Furthermore, as described in Section 2.2.5, low light and electronics induce noise in the acquired image. Since we aim at training a regression network that is not specific to any defined acquisition condition, we characterized the robustness of the neural network to extrinsic modifications of the image quality.

The list of handled degradations is the following:

- global illumination level,
- non-uniform illumination (e.g. caused by poorly adjusted Köhler illumination),
- zero-mean Gaussian noise,
- signal-dependent Poisson noise,

- mixed Gaussian-Poisson noise.

Variables: degradation strength, degradation type, three different CNNs, two different PSF models (G-1, Z-1).
Fixed: the CNNs are already trained with [micr-syn-poi] and selected using an independent test dataset. The test data set is common to all modalities.
Evaluation criterion: R^2 between the degradation parameters that were used to generate the test image and the parameters recovered by the CNN.

Fig. 2.6 summarizes the performance of the CNN networks as a function of degradation strength. All networks are robust to partial or full brightness changes in the input image. The addition of Gaussian noise to the input results in a slow and linear decay in performance, whereas the application of Poisson noise to the data decrease the performance much faster as the noise strength increases. Using CNNs trained for regression of Zernike polynomial PSF model parameters, the regression performance is decaying linearly as a function of the amount of noise we apply in the input picture. For Gaussian models, the parameter estimation usually breaks with less noise than with the Zernike polynomial model. Without any degradation, networks regressing Zernike polynomials are less accurate than networks for Gaussian PSF models, but they appear to be more robust when the input is noisy. Indeed, with a very strong (strength of 0.5 in Fig. 2.6) Gaussian and Poisson noise, Zernike-1 CNNs dropped from $R^2 = 0.95$ to $R^2 = 0.85$, as opposed to the Gaussian-1 CNNs, which dropped from $R^2 = 0.99$ to less than $R^2 = 0.60$. Surprisingly, contrary to the findings in a recent benchmark [59], we found that CNNs based on ResNeXt performed worse than their ResNet counterparts. Finally, we trained new models without adding synthetic noise to the training dataset (see Eq. (2.10)). Performance of these networks was the same as their counterparts for the illumination degradations, but dropped to $R^2 = 0$ when the test images contained even only moderate Poisson and Gaussian noise.

2.4 Discussion

Hereafter, we discuss the results of the experiments described in Section 2.3.

2.4.1 Characterization of the CNN regression performance

Many regression accuracies are above $R^2 = 0.90$ in Table 2.3, which shows that our neural networks can accurately regress PSF model parameters (in particular, when images are textured). The recovered parameters can be used to generate synthetic PSFs that are similar to the ones that degraded the image.

Our network is most accurate when applied to images of the same type as the ones used for training and the performance scales with the size of the training set. Therefore, the more data

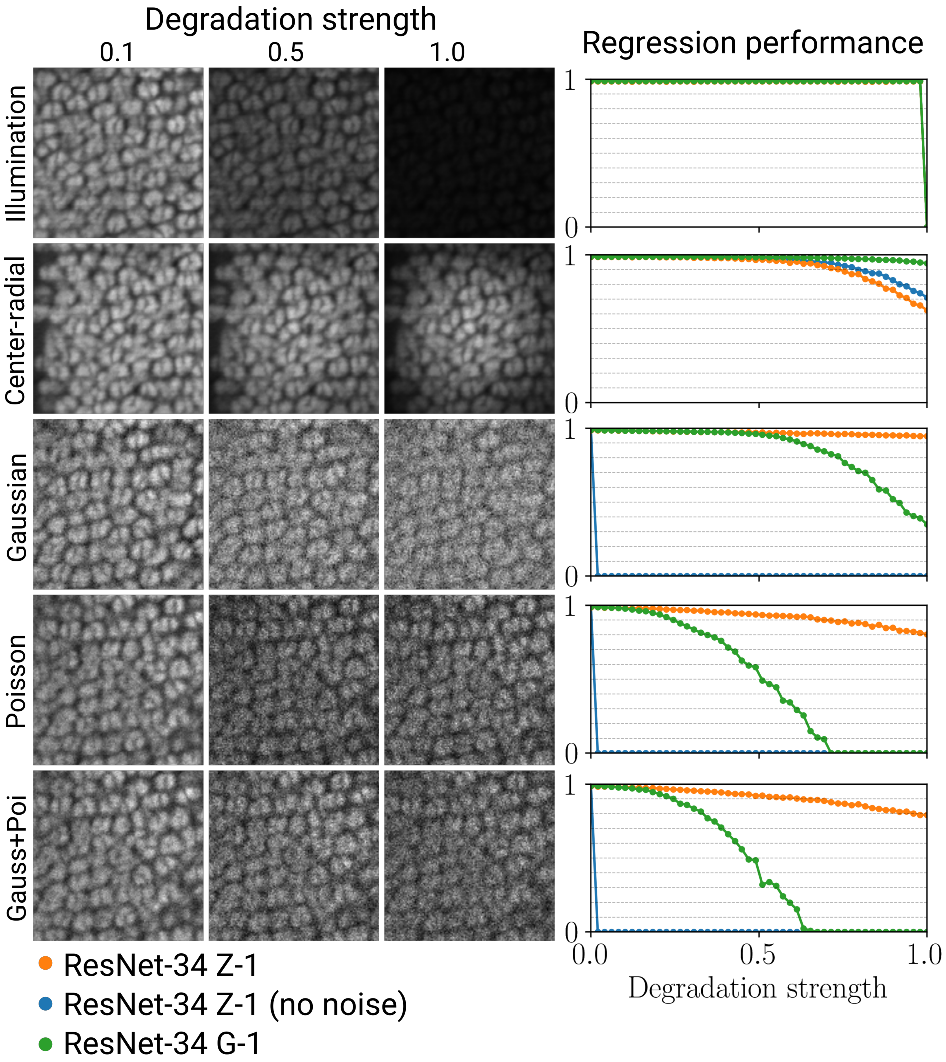


Figure 2.6: Regression performance of CNNs architectures trained on [micr] and evaluated on the [micr] test set with various types of degradation and variable strength. The regression performance is shown in terms of R^2 using $N = 10,000$ images. We trained as well a CNN without adding noise in the training set (blue). All three networks exhibit the same performance for the illumination degradation (top right).

we gather, the more precise and robust the predictions are. However, adding synthetic training data to augment a natural images data set does not increase the efficacy of PSF estimations. The network fails to predict the PSF parameters if the network is trained with a very narrow type of data and more variety increases generalization potential. Nevertheless, when learning from a data set of images containing texture, even if different from the test set type, the model remains as accurate as when training the data set using microscopy images. This suggests that one could avoid the need to gather costly microscopy ground-truth data, or that it might be possible to learn from images of other microscope types (e.g. confocal microscopes) and use the trained models with wide field microscopy images. Furthermore, the high correlation score ($R^2 > 0.8$) obtained for [micr] test images using networks trained with [nat] suggests that the networks did not undergo overfitting and were able to generalize on other data types.

In comparison to ResNet-34, ResNeXt-50 requires a larger number of images to be accurate since the regression accuracy drops drastically (from $R^2 = 0.97$ to $R^2 = 0.79$) using a smaller data set. It is consistent with the general idea that the amount of training data must scale with the depth of the network to be able to generalize well.

Networks trained for Gaussian PSFs estimate parameters with a better accuracy than networks trained to find Zernike polynomials parameters. This could be explained by the fact that although the Zernike polynomial parameters are independent when describing the pupil function, they can compensate each other when forming the PSF (which is obtained by a non-linear operation on the pupil function, see Eq. (2.7)).

Using an NVIDIA GeForce GTX 1080 GPU, the estimation of the PSF parameters of a 1024×1024 px image with 64 PSFs takes around 5.9 ± 0.1 ms, which is in the same scale as the usual camera exposure time. This suggests that real-time applications in a microscope could be feasible.

While transfer learning (i.e. networks trained with ImageNet for image detection prior to training) does not help to improve the final accuracy of the regression task because the network was already able to learn from the original data set in a reasonable amount of time, training networks by starting from pre-trained models tends to speed up convergence during learning.

2.4.2 Robustness analysis against input degradation

We noticed that all our models are invariant to changes in illumination, certainly due to inherent normalization steps in the CNN architecture, and are overall robust to small to medium amounts of noise. However, when the signal-to-noise ratio strongly decreases, the correlation coefficient tends to decrease as well.

Additionally, we observed that, in comparison to ResNet, CNNs based on ResNeXt perform generally worse when noise is applied. This result contradicts observations reported in a recent benchmark [59], where ResNext is more robust than ResNet to Gaussian noise. However, this publication scores the network's accuracy for a classification task into image type categories, which is a different application than our regression of aberrations and might explain the discrepancy.

We found the synthetic degradations we added in the training set (Eq. (2.10)) to be a necessary step to achieve robustness to noise. Indeed, the performance dropped when this step was omitted.

2.5 Conclusion

In this work, we have shown that CNNs, in particular residual networks, can be used to extract local blur characteristics from microscopy images in the form of parameters of a PSF model with only minimal knowledge about the optical setup. Our system is robust to signal perturbations and does not need to be trained specifically on images of the target imaging system. This flexibility allows the user to perform, without taking measurements beyond the images of interest, a wide range of tasks in microscopy image processing. We will investigate in the following chapters possible applications in biomicroscopy that can be inferred from the estimation PSF parameters with minimal *a priori* knowledge of the optical setup.

Applications Part II

3 Depth estimation

This chapter was published in A. Shajkofci and M. Liebling, “Spatially-Variant CNN-Based Point Spread Function Estimation for Blind Deconvolution and Depth Estimation in Optical Microscopy,” *IEEE Transactions on Image Processing*, vol. 29, pp. 5848–5861, 2020.

A.S.conceived and carried out the experiments, A.S wrote the manuscript with support from M.L, M.L supervised the project.

3.1 Methods

The spatially-variant PSF parameter mappings obtained in Section 2.2.6 yield local parameters, such as the blur, that are a function of the distance of the object to the focal plane. However, due to the symmetry of the PSF in depth, this function is ambiguous about the sign of the distance map $z(\boldsymbol{s})$ (above or below the focal plane). That is why we now aim at estimating the depth map $z(\boldsymbol{s})$ for every lateral pixel $\boldsymbol{s}$ of the 2D manifold in 3D space using our trained neural network and one single image as input. To achieve this, we use a local combination of Zernike polynomial coefficients $\boldsymbol{a}(\boldsymbol{s})$. The de-focus coefficient $a_1(\boldsymbol{s})$ is linked to the distance of the object to the focal plane, but there is no information about whether the object is in the front or behind the focal point. To address this limitation, we took inspiration from several methods to retrieve the relative position of a particle by encoding it in the shape of its PSF (either via use of astigmatic lenses ([34], [70]) or by use of a deformable mirror to generate more precise and complex PSF shapes [71]).

We used two cylindrical lenses of focal length −400 µm and 400 µm, separated by 3.4 cm thereby giving a combined focal length of $f = 6000\,\text{mm}$ and placed them in the infinity space of a microscope to generate an imaging system with an astigmatic PSF (Fig. 3.1). We used the networks trained in Section 2.2.5 using 2D-Zernike models to infer the depth map $z(\boldsymbol{s})$ from the 2D image $y(\boldsymbol{s})$ of the tridimensional surface $x_{3\text{D}}(\boldsymbol{r}, z(\boldsymbol{r}))$. We defined a distance metric by multiplying the output focus parameter and the normalized and zero-centered astigmatism direction:

$$z(\boldsymbol{s}) = a_1(\boldsymbol{s})\left(\frac{2a_3(\boldsymbol{s})}{\pi} - 1\right), \tag{3.1}$$

with $a_1(\boldsymbol{s})$ the spatially local de-focus Zernike coefficient, and $a_3(\boldsymbol{s}) \in (0, \pi)$ the spatially local Zernike coefficient encoding the direction of astigmatism.

3.2 Results

The PSF parameter describing the local blur that we obtained from the image in Section 2.2.6 lacks information about the relative direction of the object from the focal plane. To infer the local axial distance map $z(\boldsymbol{s})$ in the sample, we applied the method described in Section 4 using the astigmatism created by a cylindrical lens. As an imaging sample, we used a grid of 200 µm × 200 µm squares which we laser-printed on a transparent plastic foil. We placed the grid towards the focal plane and tilted it by 3°, 6°, or 10°, so that the in-focus position was in the middle of the field of view (Fig. 3.1 (a)). With a field of view of 655 µm × 655 µm, such a rotation yielded depth ranges of 48.2 µm, 94.7 µm, or 159.9 µm, respectively.

We were able to retrieve the local Zernike coefficient parameters of focus (a_1), cylinder (a_2), and axis (a_3). We inferred the depth map $z(\boldsymbol{s})$ using Eq. (3.1) for $N = 30$ acquired images in total (see Fig. 3.1 (e)).

Variables: input images of surfaces with a varying tilt angle from the focal plane.
Fixed: the network architecture (ResNet-34-pretrained for regression of Zernike parameters with astigmatism). CNNs are already trained, the test data set is fixed.
Evaluation criterion: ℓ_1-error between the actual axial position of the surface and the position extracted from the image.

Using a CNN ResNet-34 trained with [micr], we obtained a correlation coefficient (R^2) between the average slope and a line fit of more than 0.96. From this line fit we calibrated the system to spatial units so that we could build a depth map. The method was accurate with an absolute average ℓ_1-error of 1.81 ± 1.39 µm in depth (corresponding to a $1.61 \pm 1.23\%$ of the relative depth boundaries), obtained by comparing the error between the known position of the object in depth and the calibrated distance. Results in Table 3.1 reveal that the relative error increases as the maximum depth of the object increases. Depth estimation is thus more precise around the focal plane.

3.3 Discussion

We showed that the focus parameter of the PSF models is a function of the distance between the sample and the focal plane, and that the sign of the axial distance could be recovered

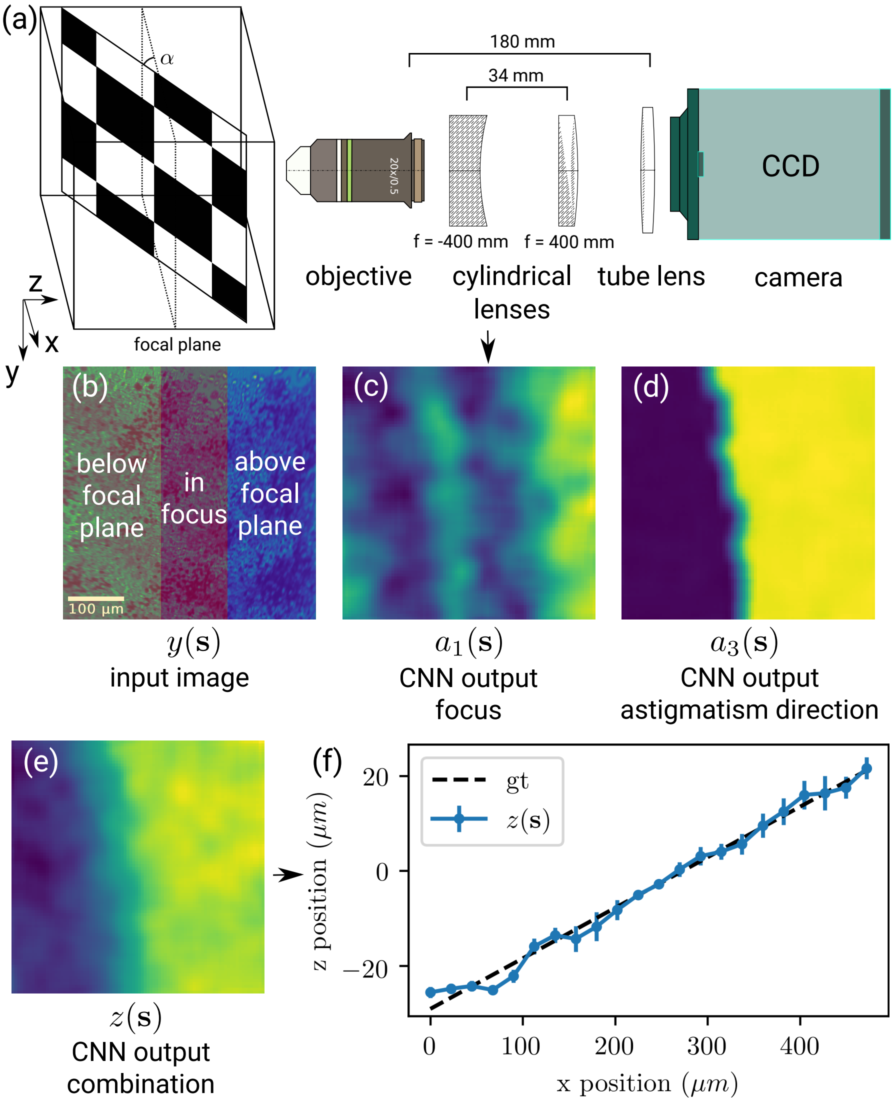


Figure 3.1: Depth estimation of a plane using controlled astigmatic aberrations by use of cylindrical lenses. (a) Optical system with using cylindrical lenses in the infinite tube space to induce astigmatism. (b) Image of a grid taken from the camera, with highlighted parts of the surface above and below the focal plane. (c),(d) Output of the CNN using (b) as input. The resolution of the map is $M = 31 \times 31$ different PSFs. (e) Weighted combination of (c) and (d) to form the depth map. (f) Projection of the depth of the surface over the y axis and comparison to the ground truth data.

Table 3.1: Analysis of the precision of the depth recovery of a plane using controlled astigmatic aberrations by use of cylindrical lenses. The test images are acquisitions of $N = 30$ printed grids tilted 3°, 6° and 10°. The network is a ResNet-34 trained on data set [micr].

Tilt angle	R^2	Absolute error	Relative error
10°	0.967	$3.50 \pm 2.62\,\mu\text{m}$	$2.18 \pm 1.64\%$
6°	0.988	$1.31 \pm 1.15\,\mu\text{m}$	$1.38 \pm 1.21\%$
3°	0.989	$0.61 \pm 0.41\,\mu\text{m}$	$1.26 \pm 0.85\%$

from higher Zernike coefficients when using engineered PSFs. Furthermore, we found that the depth function at any point in the image could be obtained from an affine function that combines two Zernike coefficients.

We recovered the relative depth of both sides of the printed grid in real microscopy acquisitions, which extends the idea of PSF engineering introduced in [70] for point-like structures, to work for fully textured images. Using a textured plane image, we have been able to recover the depth to up to 160 μm. However, this accuracy decreases when the imaged objects lack texture. Using other shapes of engineered PSFs (e.g. quadripoles [71]) could potentially lead to improved depth accuracy.

Since use of an engineered PSF degrades the image, simultaneous depth retrieval and high-resolution might best be carried out by splitting the acquisition line (e.g. with a beam splitter) to record the image on one side and a PSF engineered image on the other.

Depth retrieval is also possible using the parameters of a Gaussian-2 PSF model, but the precision is improved using the PSF model based on Zernike polynomials.

4 Blind spatially-variant deconvolution

This chapter was adapted from A. Shajkofci and M. Liebling, “Semi-blind spatially-variant deconvolution in optical microscopy with local point spread function estimation by use of convolutional neural networks,” in *IEEE ICIP*, 2018, pp. 3818–3822 and A. Shajkofci and M. Liebling, “Spatially-Variant CNN-Based Point Spread Function Estimation for Blind Deconvolution and Depth Estimation in Optical Microscopy,” *IEEE Transactions on Image Processing*, vol. 29, pp. 5848–5861, 2020.

A.S. designed the model and the framework, conceived and carried out the experiments, A.S wrote the manuscript with support from M.L, M.L supervised the project.

4.1 Introduction

Optical microscopy is a powerful tool to comprehend biological systems, enabling researchers and physicians to acquire qualitative and quantitative data about cellular function, organ development, or diseases. However, light traveling through any imaging system undergoes diffraction, which leads to image blur [1]. This represents an intrinsic limit and the determining factor for the resolution of an optical instrument, and thus limits visual access to details. Indeed, the optical system only collects a fraction of the light emitted by any one point on the object, and cannot focus the light into a perfect image. Instead, the light spreads into a three-dimensional diffraction pattern. Image formation can be modeled as the convolution of the original object with a PSF, which sums up the optical aberrations [2]. For thin, yet not flat, samples, the PSF remains shift-invariant within small areas of the 2D image, but the three-dimensional depth of the imaged object produces a local blur. Using a PSF corresponding to the blur in a deconvolution algorithm can be used to restore details in the image [3].

Deconvolution techniques can be categorized into three classes: (1) Non-blind methods, (2) entirely blind methods, and (3) parametric semi-blind algorithms. Non-blind methods require knowledge of the PSF [45]. One of the main difficulties in practice is to determine the original PSF that characterizes the actual optical system without discovering it empirically by acquiring a 3D image of a fluorescent bead, which is a tedious and time-consuming calibration step. The two latter classes fall into blind deconvolution (BD) techniques, which improve the image without prior knowledge of the PSF, the object or other optical parameters. Entirely blind algorithms, such as [64] are based the optimization of a penalty function or a maximum a posteriori (MAP) estimation of the latent image or kernel [43]. However, these methods typically use strong constraints such as sharpness along the object edges and do not always generalize to unexpected or noisy types of data [44], which are common in microscopy images. Also, many BD techniques are computationally expensive, especially for larger convolution kernels, and assume spatially invariant PSFs. Finally, parametric or semi-blind algorithms are blind methods that are constrained by knowledge about the transfer function distribution, such as a diffraction model or a prior on the shape of the PSF ([5], [72]). Parametric models allow reducing the complexity of the optimization problem, increasing the overall robustness, and avoiding issues such as over-fitting. However, it remains hard to estimate the parameters from experimental data. We will focus on this third class of deconvolution methods, by addressing the central problem of how to best infer the parameters without measuring any of them experimentally.

Machine learning recently improved the ability to classify images [10], detect objects, or describe content [12]. Convolutional Neural Networks (CNNs) [73], in particular, are built for learning new optimal representations of image data and perform self-regulating feature extraction [14]. Because of their ability to learn correlations between high- and low-resolution training samples, CNNs appear well adapted to our problem of determining the blur kernel. A similar reasoning led to recent results in [15] and [37], where the direction and amplitude of motion blur was determined by a CNN classifier from images blurred with a Gaussian kernel.

Here we present a spatially-variant BD technique aimed at microscopy of thin, yet non-flat objects. Our method combines local determination of the PSF and spatially-variant deconvolution using a regularized Richardson-Lucy (RL) algorithm [65]. To find the PSF in a computationally tractable way, we train a CNN to perform regression of model parameters on synthetically blurred image patches. The novel aspects of our approach are:

1. Our method does not require the experimental measurement of a PSF, only synthetic training data is necessary.

2. Compared to non-parametric BD, the problem complexity remains low and therefore is more easily amenable to optimization.

3. Parameters with a physical meaning are inferred from the image itself.

4. The algorithm is computationally efficient, resulting in a near real-time kernel regression and mapping at the expense of a much longer, yet straightforward, training phase.

4.2 Methods

Given the degraded image and a local map of PSF parameters, we restore the input using Total Variation regularized Richardson-Lucy (TV-RL) deconvolution. Richardson-Lucy (RL) is an iterative maximum-likelihood approach and assumes that the noise follows a Poisson distribution [74], which is well adapted for microscopy. The method is subject to noise amplification, which can, however, be counterbalanced by a regularization term that penalizes the l_1 norm of the gradient of the signal ([75], [65]). Here, we assume that the PSF is spatially invariant in small parts of the image. Spatially-variant convolution techniques have been extensively reviewed by Denis et al. [76]. Hirsch et al. [64] have shown that the local invariance assumption can be improved by filtering the input with every local PSF and then reconstructing the image using interpolation. We extend this method by its inclusion in the TV-RL algorithm. Rather than interpolating deconvolved images, the overlap-add filtering method, as described in [64], [77], interpolates the PSF for each point in the image space. The idea for such method is: (i) to cover the image with overlapping patches using smooth interpolation, (ii) to deconvolve each patch with a different PSF, (iii) to add the patches to obtain a single large image. The equivalent for convolution can be written as:

$$x(\boldsymbol{r}) = \sum_{m=0}^{M} (\widetilde{h}^{(m)} * (\varphi^{(m)} \odot y))(\boldsymbol{r}), \tag{4.1}$$

where $\varphi^{(m)}(\boldsymbol{s})$ is the masking operator of the mth patch. We illustrated the masking and deconvolution steps in Fig. 2.4. Since the RL algorithm tends to exacerbate edges and small variations such as noise, we use Total Variation (TV) regularization to obtain a smooth solution while preserving the borders [65]. The image at each RL iteration becomes:

$$x_{i+1}(\boldsymbol{r}) \quad = \quad \sum_{m=0}^{M} \left[\frac{(\widetilde{h}^{(m)} * (\varphi^{(m)} \cdot y))(\boldsymbol{r})}{(\widetilde{h}^{(m)} * x_i^{(m)})(\boldsymbol{r})} * \widetilde{h}^{(m)}(-\cdot) \right] \cdot \frac{x_i^{(m)}(\boldsymbol{r})}{1 - \lambda_{TV} \operatorname{div}\left(\frac{\nabla x_i^{(m)}(\boldsymbol{r})}{|\nabla x_i^{(m)}(\boldsymbol{r})|} \right)}, \tag{4.2}$$

with $y(\boldsymbol{s})$ the blurry image, $x_i(\boldsymbol{r})$ the deconvolved image at iteration $i = 1, ..., I$, $x_i^{(m)}(\boldsymbol{r})$ the $K_y \times L_y$ deconvolved patch at iteration i, M the number of patches (and different PSFs) in one image x, $\widetilde{h}^{(m)}$ the $K_h \times L_h$ PSF for patch m and λ_{TV} the TV regularization factor. $\nabla x_i^j(r)$ is the finite difference operator, which approximates the spatial gradient.

4.3 Results

We next wanted to verify that the parameters recovered by the CNN were producing PSFs that are sufficiently accurate to be usable to enhance the details in the image, despite not being specifically measured. To this end, we devised a deconvolution experiment to compare images deconvolved by our method with those obtained by other blind deconvolution techniques. As test input, we used 256 × 256 pixels image patches from the [micr] data set (see Section 2.2.5). Using Eq. (2.10), we degraded each quadrant of the input image with a specific, randomly-generated 127 × 127 pixels PSFs using parameters $\boldsymbol{a}(\boldsymbol{s})$ drawn from a uniform random distribution allowing us to systematically explore the parameter space. We subsequently inferred a PSFs map via the CNN from the blurry image as described in Section 2.2.6. Finally, we deconvolved the image by applying the method described in Section 3. In the experiment, we set $\varphi^{(m)}$ as a bilinear interpolating function and, similarly to [65], $\lambda_{TV} = 0.1$. Since using FFT-based calculations implies that the PSF is circulant, we took into account field extension to prevent spatial aliasing. We fixed the number I of RL iterations to 20.

We assessed the reconstruction quality by computing the SNR and Structural Similarity (SSIM) [78]. We compared the deconvolution results to spatially-invariant blind deconvolution techniques [79], [80] and [81], and the spatially-variant method from [82]. In the latter cases, we used the estimated PSF in the TV-RL algorithm with the same number of iterations and λ_{TV}. Since the estimation of a full PSF by these methods would take more than 20 minutes per sample, we constrained the support of the PSF to 31×31 pixels. We computed the scores by taking the difference between the “ground truth" SNR and SSIM of images deconvolved using the PSFs actually used to degrade the images, and the deconvolution results using PSFs regressed with the CNN or other BD techniques. Theses values are therefore reported as ΔSNR and ΔSSIM.

Finally, in order to recover details lost due to the aberrations of actual microscope objectives (such as out-of-focus blur and astigmatism), we acquired different fixed samples (HeLa cells actin (Alexa Fluor 635) and HeLa cells anti-α-catenin (Alexa Fluor 488) with a $10\times/0.3$ air objective, *Convallaria majalis* bulb autofluorescence with a $20\times/0.7$ air objective) both in focus and slightly out of focus. Then, starting from a 256×256 patch of the out-of-focus picture only, we sought to retrieve a sharper picture containing the details of the in-focus picture. We compared qualitatively the in-focus image, the out-of-focus image, our method with four PSFs detected with a 128×128 stride, [79], [82], and the imaged obtained via a “sharpen” high-pass filter.

Variables: network architecture, two different PSF models (Z-1 and G-1) for the degradation and detection parts.
Fixed: CNNs are already trained, the test data set [micr] is fixed.
Evaluation criterion: difference of SNR and SSIM between the ground truth image and the deconvolved image.

Results in Table 4.1 indicate an average improvement of both SNR (1.88 dB) and SSIM (0.09) of our spatially variant BD. In comparison to spatially-invariant BD and other spatially-variant BD techniques improves the image by 1.55 dB SNR and 0.08 SSIM. Deconvolution results are equivalent when the degradation and detection models are mismatched. The qualitative results shown in Fig. 4.1 highlight the stability of our method, which improves the degraded image to a detail level similar or better than the one of the in-focus image. Using the algorithm of Kotera et al. [79], the blurry features are well recovered, but the images have less detail. Furthermore, this algorithm converges to an aberrated image (Fig. 4.1 (b)) when the image

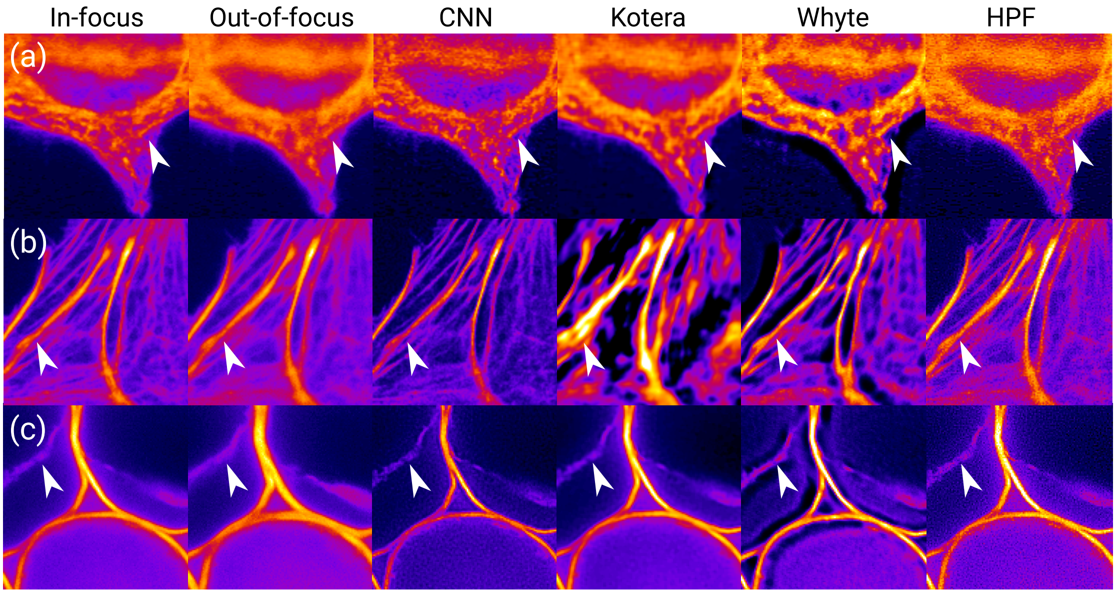


Figure 4.1: We acquired different fixed samples ((a) HeLa cells actin (Alexa Fluor 635) and (b) HeLa cells anti-α-catenin (Alexa Fluor 488) with a 10× /0.3 air objective, (c) *Convallaria majalis* bulb autofluorescence with a 20× /0.7 air objective) both in-focus and slightly out-of-focus. Then, starting from a out-of-focus 256 × 256 patch only, we seek to retrieve a sharper picture containing the details of the in-focus picture. We compared qualitative results of (from left to right), the in-focus image, the out-of-focus image, the proposed method (CNN) with four PSFs detected with a 128 × 128 stride, [79], [82], and a "sharpen" high-pass filter. Arrow edges indicate regions with features of particular interest.

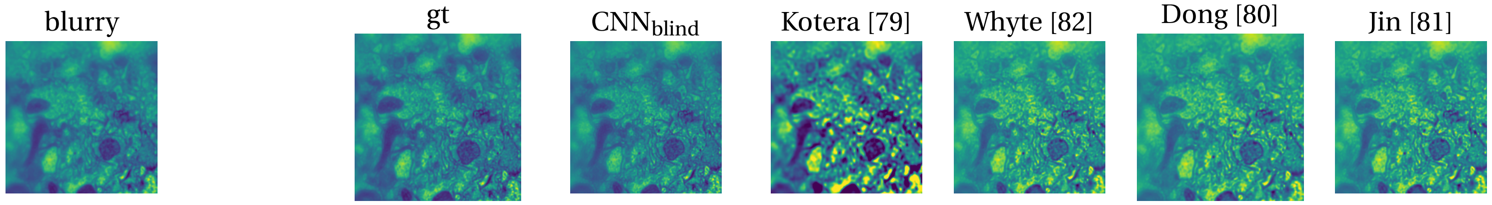
contains long filaments. The method from Whyte et al. [82] enhances the contrast of the blurry image, however, it creates hallucinations near edges, which were not part of the original image. Finally, the high-pass filter, as expected, enhances both high-frequency features and noise.

4.4 Discussion

Using the spatially-variant PSF map inferred from the PSF output, we have been able to reconstruct details in a degraded image without any prior information on the image content or the optical system. Given that our method does not require adjusting parameters or experimentally measuring a PSF (which is labor intensive), it leads to results faster than non-blind deconvolution methods.

We noticed that, because we use a constrained PSF model, our deconvolution method does not suffer from drawbacks sometimes associated to other deconvolution and super-resolution methods. For example, techniques based on MAP optimization sometimes converge to exotic forms of PSFs that are not consistent with the physics of optics, causing image deformation or loss of features [83]. We could illustrate this by the example in Fig.4.1 (b) for the method from Kotera et al. [79], which diverges when directed filaments are shown to the algorithm

Table 4.1: Image deconvolution benchmark with 350 input images (256 × 256) from the test set [micr] for each experiment. We computed the difference in SNR and SSIM between the ground truth image and the deconvolved image. We compared the following methods: TV-RL with four known PSFs (gt), TV-RL with four PSFs estimated by the proposed CNN ($\mathrm{CNN}_{\mathrm{blind}}$), Kotera et al., 2013 [79], Whyte et al., 2014 [82], Dong et al. [80] and Jin et al. [81]. The CNN is a ResNet-34 trained on data set [micr].



Degradation	Detection	$\Delta\mathrm{SNR}_{\mathrm{gt}}$	$\Delta\mathrm{SNR}_{\mathrm{CNN\text{-}blind}}$	$\Delta\mathrm{SNR}_{[79]}$	$\Delta\mathrm{SNR}_{[82]}$	$\Delta\mathrm{SNR}_{[80]}$	$\Delta\mathrm{SNR}_{[81]}$
Z-1	Z-1	2.14 ± 0.71	**1.65** ± 0.62	−0.81 ± 0.71	0.12 ± 0.31	0.61 ± 0.58	1.34 ± 0.62
Z-1	G-1	2.14 ± 0.71	**1.72** ± 0.71	−0.61 ± 0.67	0.02 ± 0.31	0.83 ± 0.59	1.48 ± 0.67
G-1	Z-1	2.24 ± 0.73	**2.02** ± 1.04	−1.20 ± 0.97	−0.01 ± 0.55	0.89 ± 0.74	1.78 ± 0.76
G-1	G-1	2.24 ± 0.73	**2.11** ± 0.72	−1.25 ± 1.03	−0.31 ± 0.87	0.68 ± 0.9	1.58 ± 0.70
		$\Delta\mathrm{SSIM}_{\mathrm{gt}}$	$\Delta\mathrm{SSIM}_{\mathrm{CNN\text{-}blind}}$	$\Delta\mathrm{SSIM}_{[79]}$	$\Delta\mathrm{SSIM}_{[82]}$	$\Delta\mathrm{SSIM}_{[80]}$	$\Delta\mathrm{SSIM}_{[81]}$
Z-1	Z-1	0.10 ± 0.03	**0.10** ± 0.04	0.05 ± 0.04	0.02 ± 0.02	0.09 ± 0.04	0.09 ± 0.04
Z-1	G-1	0.10 ± 0.03	**0.09** ± 0.03	0.03 ± 0.05	0.02 ± 0.01	0.08 ± 0.04	0.08 ± 0.03
G-1	Z-1	0.08 ± 0.03	0.08 ± 0.06	0.01 ± 0.05	0.01 ± 0.03	0.07 ± 0.04	**0.08** ± 0.03
G-1	G-1	0.08 ± 0.03	0.08 ± 0.04	0.01 ± 0.06	0.03 ± 0.03	0.07 ± 0.04	0.08 ± 0.04

and creates artifacts.

Similarly, the use of denoising CNNs for image enhancement can lead to phantom details that could falsify underlying biological features, or discard high-frequency features that are mistaken for noise [84]. When using constrained PSF models such as the ones we use, deconvolution algorithms such as RL will still produce a reasonable image even if the predicted PSF is not exactly matching the PSF corresponding to the blur. This is likely due to the inherent constraint of a model with a small number of parameters that enforces the shape of the PSF. The outcome is a higher average SNR and SSIM of the reconstruction using our method compared to other BD algorithms. Another advantage of using constrained PSF models is that they can model PSFs with very large supports (sizes). Classical BD only allows for a smaller support, as using more pixels creates higher complexity. Nevertheless, our models are currently unlikely suitable for some types of degradation, where BD methods were successfully applied, such as for compensating for motion blur with rotation [85].

Deconvolution results are equivalently efficient both in terms of SNR and SSIM when the degradation and detection models are not similar (e.g. degradation using a Gaussian PSF and estimation of the PSF using a Zernike model). This particular point is relevant since it confirms the robustness of the image enhancement process when there is a mismatch between the degradation PSF that we want to model (i.e. the optical system PSF) and the model itself.

Finally, we observed that methods [82], [80] and [81], due to their multiscale optimization approach, were considerably slower than the one we propose, taking up to 4 minutes to deblur a 256 × 256 image, whereas our method takes less than 3 seconds using the same machine to both estimate the PSF and perform TV-RL. The difference in run times can be explained by our GPU implementation, but as well because of the inherent nature of traditional optimization algorithms that alternate kernel and image estimation, which limits the parallelizability of the calculations.

5 DeepFocus: a Few-shot Microscope Slide Auto-Focus

This chapter was adapted from A. Shajkofci and M. Liebling, "DeepFocus: A Few-Shot Microscope Slide Auto-Focus Using a Sample Invariant CNN-Based Sharpness Function," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2020, pp. 164–168.

A.S. designed the model and the framework, conceived and carried out the experiments, wrote and published the software. Parts of the graphical interface of the micro-manager plugin was written by Cevahir Köprülü, an EPFL student. A.S wrote the manuscript with support from M.L, M.L supervised the project.

5.1 Introduction

Modern microscopy techniques rely on many components that are remotely controllable. This allows implementing control loops that limit the need for human-supervised operation. Auto-focusing systems, in particular, are used extensively in the acquisition of timelapses in developmental or cellular biology or to automatically image slides in a slide scanner. In the former application, imaged specimens tend to drift from the focal plane over time because of specimen growth, flow of the medium, or motion caused by temperature changes. In the latter case, variability in the mounting of the slides requires per-slide adjustment.

Autofocus (AF) systems seek to determine the optimal shift by which to adjust the axial position to maximize image sharpness. AF solutions can be hardware-based (e.g. laser-based sensing of the sample drift [87] or phase detection by an auxiliary sensor [88]) or image-based, which does not require any modification of the optical path of the microscope as a focus score is retrieved from the image itself [89].

We can classify image-based AF algorithms into two categories. The first comprises AF methods that use iterative minimization of a one-dimensional objective function, the focus score, to move the object to the point at which it is sharpest. Because the output of the function is not predictable and depends on the sample, the AF has to acquire tens to hundreds of images at different axial positions in order to converge to a non-local optimum [89]. A high number of

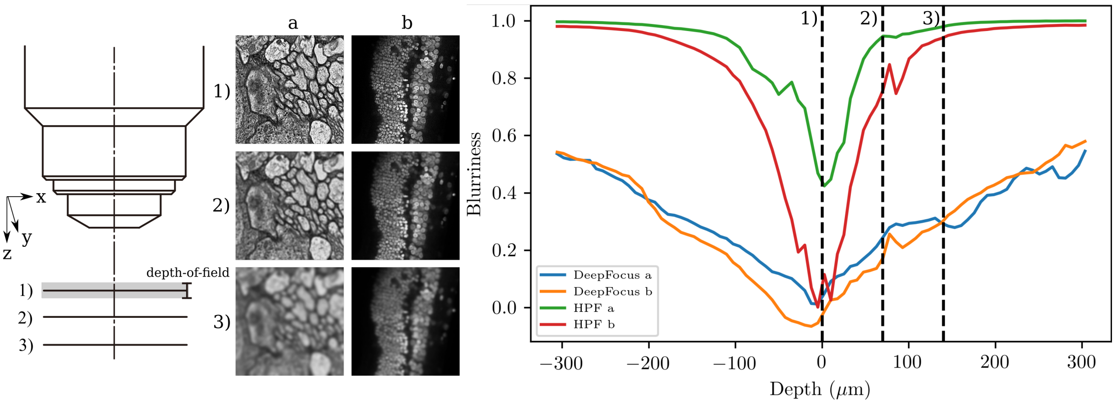


Figure 5.1: The object may be outside of the DOF and appear blurry. Here we quantified the blur $b(z)$ using DeepFocus and an high-pass filter (HPF) for two different images. Using HPF, $b(z)$ changes shape and slope when different objects are presented under the microscope, and there is a lack of depth information for $|z| > 150\,\mu\text{m}$. Using DeepFocus, the slopes for both images are similar in shape and retain information about depth in the whole $[-300; 300]\,\mu\text{m}$ region.

image acquisitions can be damaging for the sample, especially in fluorescence microscopy [90].

Additionally, existing objective functions only give a meaningful result in the neighborhood of the focal plane, and lose information (i.e. the gradient of the curve is zero) farther away from the focal plane. Furthermore, depending on the software implementation and the imaging modality, the acquisition of hundreds of images can take up to several minutes. The second category comprises single shot AF techniques (that need only one or a few images). Thanks to end-to-end CNNs, they take an image as input and directly deduce the optimal shift to be in focus ([91]–[93]). The drawback of these direct methods is that a long and computationally-intensive CNN training with a microscope objective-specific training data set, must be repeated whenever the optical system changes. Furthermore, these methods are not directly available in open microscope control software, such as μManager [94].

In this paper, we propose a local, CNN-based focus scoring function that remains nearly invariant when imaging different types of samples or modalities on any given microscope. We developed a correlation-based AF algorithm that takes advantage of the broad shape and unimodal minimum of this function, which helps to speed up convergence and remaining effective even when the imaged object is far from the focal (several times the depth-of-field (DOF), see Fig. 5.1). Since our CNN method does not require a microscope-specific data set for training besides a single stack of an arbitrary object, it is *plug-and-play*.

This paper is organized as follows. In Section 2.2, we present the blurriness scoring function, the calibration process, and the AF algorithm. In Section 2.3, we experimentally verify the scoring function's assumed invariance to a variety of samples and characterize performance with respect to the number of images and in comparison to common AF scoring functions, using both simulated and experimentally acquired data. We discuss our findings and conclude in Section 2.4.

5.2 Problem statement

We consider a specimen, modeled as 2D manifold in 3D space (such as a thin microscopy slide), that we wish to image with a widefield microscope in bright field, fluorescence, or phase contrast. The entire specimen or some regions in the field-of-view (FOV) can be out of focus and outside of the DOF (see Fig. 5.1). We assume the microscope has a motorized stage for adjusting the focus. We aim at finding the optimal axial shift Δz by which to adjust the sample position such that it is in focus. We seek a solution that (i) does not require a manually selected reference image to be matched (such that the method can be used both for maintaining focus in live timelapses but also for imaging collections of fixed samples) (ii) requires a minimal number of images (to limit photodamage) (iii) shall not require imaging calibration specimens (PSF measurement beads, etc.) or large-scale, microscope-specific training.

5.3 Method description

The principle behind our proposed algorithm is to measure a blurriness score $b(z_i)$ for a few (M) images acquired at different focus positions z_i, $i = 1, \dots, M$, resulting in a set of pairs $\{(z_i, b(z_i)) | i = 1, \dots, M\}$ and to determine the necessary focal shift Δz such that $\{(z_i - \Delta z, b(z_i)) | i = 1, \dots, M\}$ matches a microscope objective-specific, sample-invariant, depth-blurriness response curve $b_{\text{calib}}(z)$ using cross-correlation. The curve invariance assumption has been similarly used by the model-based curve fitting approach of [95].

For this approach to work, we need a focus estimation function that is invariant to the sample shape or texture (sample-invariance) but co-variant with the sample's axial position and sufficiently informative beyond the immediate vicinity of the focal plane. To this end, we chose an estimator of the local optical properties of the microscope objective [16]. Briefly, it relies on a trained CNN to regress the parameters of a Zernike polynomial PSF model [50], given a blurry image patch as an input. Here, we use the estimated Zernike coefficient corresponding to focus as a blurriness score, which provides, given an image as input, a local blurriness score $b[x, y, z]$ for the indicated position depth z.

The trained CNN [16] does not require re-training when used on different microscopes or different microscope objectives and produces a curve whose shape (up to an axial scaling) is invariant to the sample (an aspect that we verify experimentally in Section 5.4). In order to determine the axial scaling, which is instrument-dependent, we require a calibration step consisting in the acquisition of a full stack of an arbitrary planar and textured object. This yields a blurriness map $b_{\text{calib}}(z)$ that we center with its minimum at the origin.

We now describe our proposed AF, which follows the structure illustrated in Fig. 5.2 and is summarized in the steps:

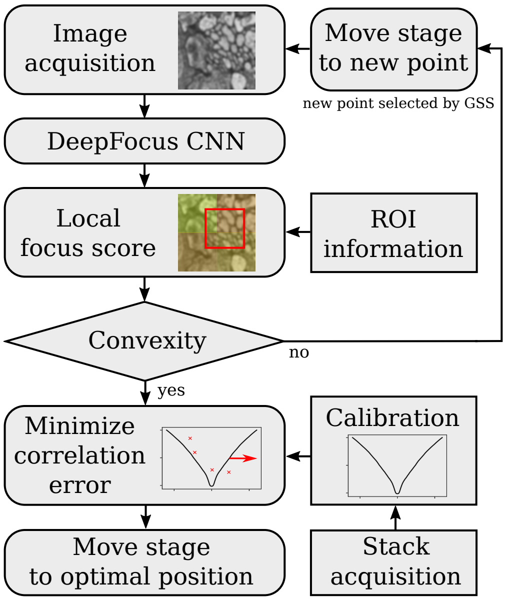


Figure 5.2: Flowchart of the AF algorithm (see Section 5.3).

1. Fit b_{calib} to a Moffat distribution [96] and extract its full width at half maximum (FWHM). Set $M = 3$, let z_1 be the initial focal plane position, and initialize a Golden Section Search (GSS) algorithm with the interval $[z_2, z_3] = [z_1 - 2\,\text{FWHM}, z_1 + 2\,\text{FWHM}]$. Acquire images at $z = z_1, z_2, z_3$ and compute, using the CNN, the blurriness scores $b[z_1], b[z_2], b[z_3]$.
2. Check the convexity of $b[z_i]$, $i = 1, \ldots, M$ by fitting $b[z_i]$ to a quadratic polynomial. If the R^2 of the polynomial fit is higher than the R^2 of a linear fit, go to Step 6. Otherwise go to Step 3.
3. Increment $M \mathrel{+}= 1$. Update the GSS triplet to obtain and move to a new axial position z_M.
4. Acquire an image at the current axial position z_M.
5. Compute, using the CNN, the blurriness score $b[z_M]$ and go to Step 2.
6. Compute using cross-correlation the local optimal shift $\Delta z(x, y)$ minimizing the squared distance:

$$\Delta z(x, y) = \underset{\Delta z}{\arg\min} \sum_{i=1}^{M} \left(b[x, y, z_i] - b_{\text{calib}}(z_i - \Delta z)\right)^2.$$

7. Move the sample by Δz, averaged for the Region of Interest (ROI) in the (x, y) plane.

5.4 Characterization of regression invariance to image diversity

Since our AF algorithm relies on the invariance of $b_{\text{calib}}(z)$ to the type of imaged sample, we investigated whether our proposed CNN indeed satisfied this condition and whether other (existing) focus metrics could be substituted.

We gathered $N_{\text{synth}} = 1000$ images from the evaluation dataset of [16] and blurred them with Gaussian PSFs mimicking a 10×, NA 0.3 objective for $M = 132$ points in the depth range $-120\,\mu\text{m} \leq z \leq 120\,\mu\text{m}$. In addition, we acquired $N_{\text{exp}} = 1000$ stacks of fixed rat brain slices tagged with three fluorescent stains using a widefield transmission light microscope with a

Table 5.1: Using the same experimental conditions as Fig. 5.3, we quantified the scoring function performance in terms of Standard Deviation (SD) in a 100µm range (lower is better) and conditional entropy between the focus score and the distance (lower is better) in the whole 240µm range. Using experimental acquisitions, DeepFocus outdo all other tested functions in terms of SD. Additionally, our method has for both modalities the lower conditional entropy and thus is more informative.

Focus score function	σ_{synth}	H_{synth}	σ_{exp}	H_{exp}
DeepFocus	0.03	**1.59**	**0.03**	**5.17**
HPF	0.06	6.38	0.08	44.31
Tenengrad [97]	0.10	4.17	0.06	21.77
LAPV	0.03	14.29	0.04	38.28
EWC [98]	**0.01**	3.47	0.16	8.21
SML	0.04	7.74	0.06	16.89
WS [99]	0.07	5.57	0.19	11.30

10×, NA 0.3 objective in a depth range of $-60\,\mu\text{m} \leq z \leq 60\,\mu\text{m}$. We then computed $b_{\text{calib}}(z)$ using DeepFocus and other methods, including HPF, LAPV, SML [100], Tenengrad [97], EWC [98], and WS [99], which cover a broad range of focus measures, as reviewed in [89], [101]–[103].

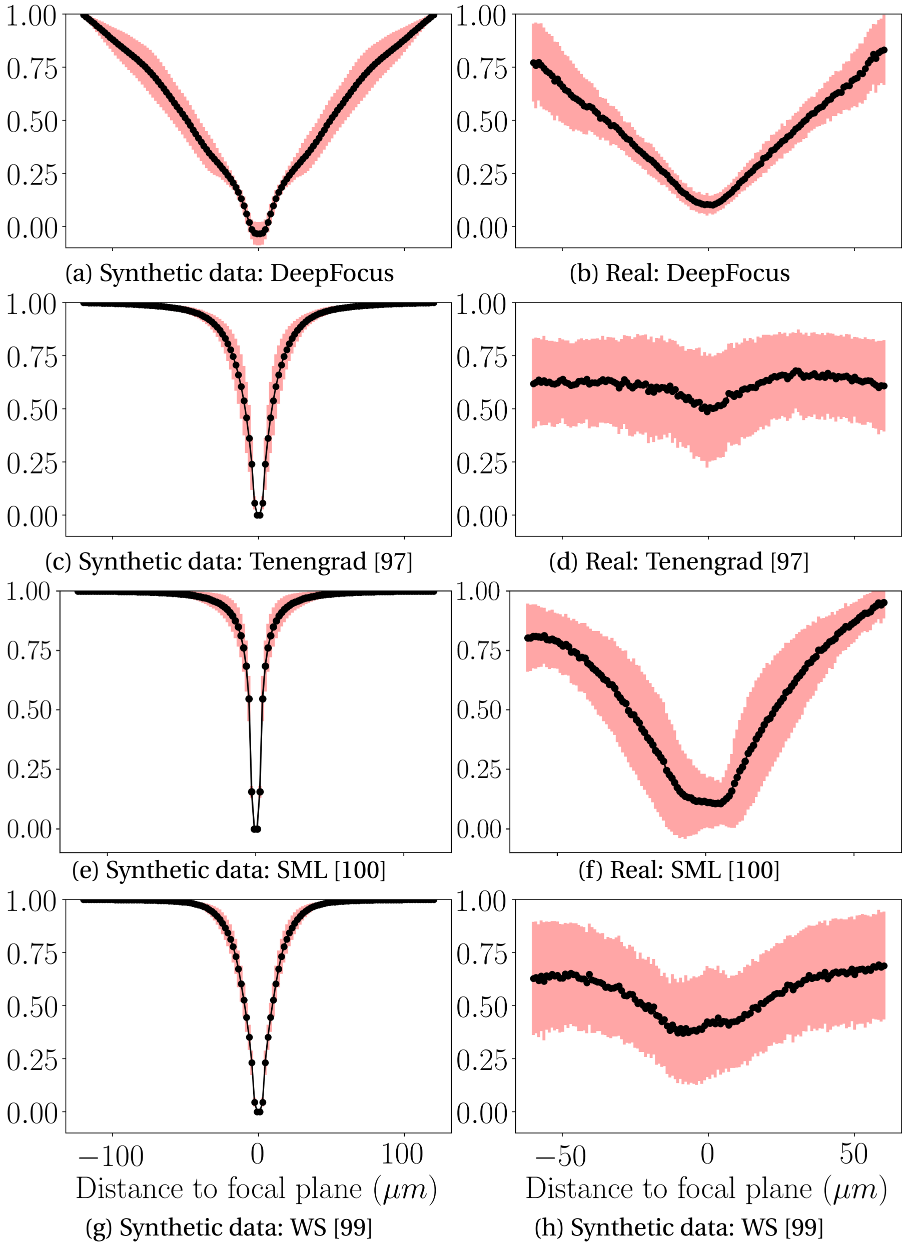
In Table 5.1, we reported the average SD of $b_{\text{calib}}(z)$ over all input images. Using the experimental dataset, our method had an average SD of $\sigma = 0.03$ (normalization scale with 1 and 0 the blurriest and sharpest values, respectively). We noticed, as illustrated in Fig. 5.3 (a) and (b), that DeepFocus' SD increased when $|z|$ increases (i.e when the acquired pictures contain a medium-to-high blur). A low SD implies that $b_{\text{calib}}(z)$ is similar with different types of imaged specimens. Other methods had a SD of $0.04 < \sigma < 0.19$, and hence confirmed the variance of these focus metrics with image diversity.

5.5 Characterization of information measure of the scoring function

We next investigated how robustly our proposed DeepFocus measure can report (de)focus information as the distance from focus is increased up to 10 times the DOF. We observed (Fig. 5.3) that focus metrics other than ours were unable to give any information about z from $b_{\text{calib}}(z)$ whenever $|z|$ is higher than 60 µm, as they reach a value that does no longer vary as the position is increased further. Since the gradient in such plateau regions is small, minimization algorithms could not converge quickly. To quantify these visual observations regarding the uncertainty of recovering z from any given $b_{\text{calib}}(z)$, we computed the conditional entropy:

$$H(\mathbf{B}|\mathbf{Z}) = -\sum_{b_{\text{calib}}\in\mathscr{B},z\in\mathscr{Z}} p(b_{\text{calib}}|z)\log\left(\frac{p(b_{\text{calib}}|z)}{\sum_{z_i} p(b_{\text{calib}}|z_i)}\right),$$

where $\mathbf{B}$ and $\mathbf{Z}$ are random variables representing the calibration blurriness score and the axial distances, $\mathscr{B}$ and $\mathscr{Z}$ their support sets, and $p(b_{\text{calib}}|z)$ the probability of a score b_{calib}, given



(a) Synthetic data: DeepFocus

(b) Real: DeepFocus

(c) Synthetic data: Tenengrad [97]

(d) Real: Tenengrad [97]

(e) Synthetic data: SML [100]

(f) Real: SML [100]

(g) Synthetic data: WS [99]

(h) Synthetic data: WS [99]

Figure 5.3: Comparison of the output of different sharpness scoring functions as a function of z, centered at the origin. We used as input synthetically blurred images (left) and stacks of fluorescent rat brain tissue with a 10×, NA 0.3 objective (right). Using DeepFocus, the SD of $b_{\text{calib}}(z)$ around the focal plane is lower than with the other scoring functions. Additionally, scoring functions other than DeepFocus do not infer depth information ($b_{\text{calib}}(z) = 1$ for all z) when $|z| > 50\mu\text{m}$.

the distance z. A high conditional entropy value implies a high uncertainty of detecting the right z position for a given b_{calib}. The results, compiled in Table 5.1, reveal that DeepFocus had a conditional entropy of $H_{\text{synth}} = 1.59$, a value smaller than that obtained when using any

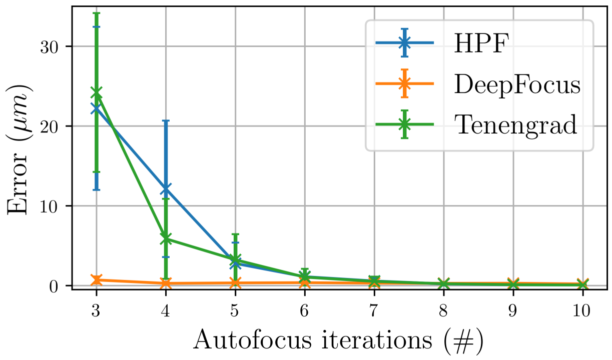


Figure 5.4: Comparison of the AF error using 3 different AF scoring functions for 100 samples. We quantified the distance between the theoretical focus plane position and the AF output as a function of the number of AF iterations which represent additional input images. DeepFocus yields an error of 0.27 ± 0.18 µm with 4 iterations. With 8 iterations or more, the others methods are on par or more accurate than ours.

of the other scoring functions instead. In the case of experimental acquisitions, we observed again an improvement in terms of entropy ($H_{\text{exp}} = 5.17$), where other methods have values in the range $8.21 < H_{\text{exp}} < 44.31$. We further determined the threshold distance after which no distance information can be inferred from the image, i.e when the image is too blurry to make the AF converge. DeepFocus retained depth information for a range of 120 μm with a 10×, NA = 0.3 objective, which is equivalent, using the diffraction-limited DOF formula, to 11 times the DOF (10.7 µm). In comparison, metrics like WS and SML achieved ranges of only 4 and 7 times the DOF, respectively.

5.6 Characterization of the AF error as a function of the number of acquisitions

We finally investigated how accurately DeepFocus could retrieve the focal distance as a function of the number of images acquired. We used 100 blurred images from the generated dataset in Section 5.4 with a known in-focus position and computed its distance to the output position of the AF. We also compared our method to other autofocus scoring functions (for which we used a bounded Brent's method as optimizer). The results are summarized in Fig. 5.4.

We observed that our proposed AF converged rapidly (3 iterations), while the two other focus functions needed more than twice as many images to reach a similar focus accuracy. Using 8 iterations or more, we did not notice a better accuracy with our method compared to Tenengrad or HPF.

In our experiments, we showed that the variance of b_{calib} over multiple images was usually lower using DeepFocus than when using other focus scoring functions, especially near the focal plane. Our explanation would be that the CNNs, already known to be translation-invariant [104], have been trained specifically for the recognition of the PSF parameters

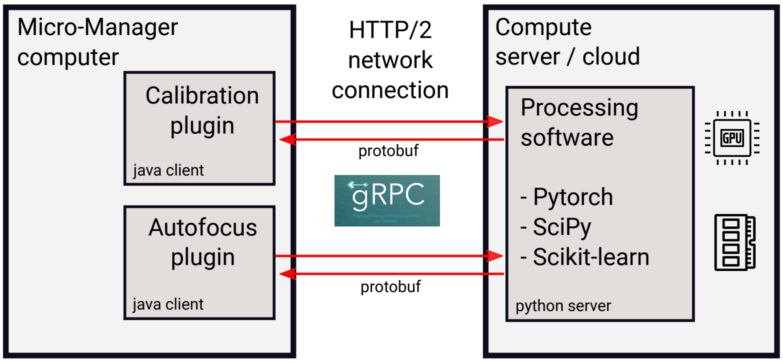


Figure 5.5: Schematical explanation of DeepFocus computational relationship between the μManager plugin and the Python/PyTorch server.

without discrimination on the input image type and position. By contrast, crafted features such as HPFs are computed from content-based calculations and differ from one image to another. When the image is acquired at a large distance from the focal plane, we noticed a loss of spatial features in the acquired image, due to the large FWHM of the PSF that degraded it. However, we have been able to retrieve depth information from the image up to 2.5 times farther away from the focal plane than with other methods. That could be mostly explained by the fact that DeepFocus computes features from a 128×128 px window, while Gradient-based methods use a much smaller window, such as 3×3 or 5×5.

5.7 Implementation and U-Net extension

We implemented the software in Python with a Java interface for μManager using a client-server architecture. Communication between the μManager client and the PyTorch server was achieved using gRPC and a common proto-file for inputs and outputs. More importantly, we developed the algorithm with a newer neural network model, adapted from U-Net [105], but with residual connections between the encoding layers. Using this network, the PSF parameter map could be computed in a single pass without a moving window. This method achieved the same accuracy, but with a better spatial resolution and up to 40 times better speeds.

5.8 Conclusion

In summary, we developed an AF method based on a combination of an CNN scoring function and optimization algorithms that are relying on the invariance of the scoring function. We showed that DeepFocus was robust to changes amongst samples, which enables the retrieval of the optimal axial shift using a correlation-based optimization process that needs as few as 3 images to converge. Our method is currently limited to imaging thin samples and further work will investigate the procedure for thicker objects. We implemented the calibration step and AF algorithm as two plugins for the μManager microscopy acquisition engine [94]. They are available at `https://github.com/idiap/deepfocus`.

6 Flow motion estimation

6.1 Introduction

Life is all about movement. From the microscale to the macroscale, organisms undergo growth, nutrients flow or diffuse in their environment. Quantification of the displacement in time of particles, organelles, or organisms can be done using optical flow [106]. Optical flow is a method aimed at determining the distribution of apparent velocities of any movement in an image. In medicine, and more specifically in cardiac imaging, optical flow proved to be correlated with the flow patterns measured using computational dynamics [107]. In photograhy, deep neural networks (DNNs) recently allowed for the prediction of 3D optical flow in a computationally-efficient way and with a good accuracy [15], [36], [37].

Optical flow is usually computed using two image frames at different time points. In microscopy, the physical scales are orders of magnitude smaller than in photography, especially in the axial direction due to the very small depth of field. For optical flow to be applied successfully, the two reference images must be taken in a short interval of time. High-speed cameras are still uncommon in microscope stations and fast movement happening during the exposure time causes motion blur.

Here we present a method for estimating the movement of out-of-plane particles in a fluid, from a single optical microscopy wide-field image with a long exposure time. We take advantage of the motion blur by estimating the parameters of a spatially-variant PSF for every point in the image. Since the PSF has been modeled to take into account the displacement in both axial and lateral directions, we are able to extract from the input image a three-dimensional vector field of the motion.

This chapter is organized as follows. In Section 6.2, we present the method, comprising the image formation model and the estimation of the displacement vector field. Then, in Section 6.3, we characterize the performance of the method by firstly simulating random movements in the images, then by simulating a flow of particles in a cylindrical pipe. Finally, we acquired microscopy images of movings beads in a fluidic device and compared the results obtained with our method with the physical fluid velocity. We then discuss our findings and conclude in Section 6.4.

6.2 Methods

6.2.1 Problem statement

We consider imaging a slice $i(\boldsymbol{s}, s_3 = z_0) = i(\boldsymbol{s})$ of a specimen ($\boldsymbol{s}$ being its lateral coordinates), where a flow of particles occurs with a tridimensional displacement vector field $\boldsymbol{v}(\boldsymbol{s}) = (v_1(\boldsymbol{s}) \quad v_2(\boldsymbol{s}) \quad v_3(\boldsymbol{s}))$. We consider as well the time step Δt, during which the pixel at position $\boldsymbol{s}$ moves following a $\boldsymbol{v}(\boldsymbol{s})$ displacement vector. From the input image i acquired using a wide-field light microscope with an exposure time Δt, we aim to predict, after adequate scaling, the 3D vector field $\boldsymbol{v}$.

6.2.2 Image formation model

We model the acquired image i as the convolution of the object o with a PSF h, integrated over time. We then define that the convolution of a moving object with a PSF is equivalent to the convolution of a fixed object with the projection over time of the PSF, as follows:

$$\begin{aligned} i(\boldsymbol{s}) &= \int_0^{\Delta t} \iiint o(\boldsymbol{r}, t) h_{3\mathrm{D}}(\boldsymbol{s}, s_3 = z_0, \boldsymbol{r}) d\boldsymbol{r} dt \\ &= \int_0^{\Delta t} \iiint o(\boldsymbol{r} + \boldsymbol{v}(\boldsymbol{r}) t, 0) h_{3\mathrm{D}}(\boldsymbol{s}, z_0, \boldsymbol{r}) d\boldsymbol{r} dt \\ &= \iiint o(\boldsymbol{r}, 0) \left(\int_0^{\Delta t} h_{3\mathrm{D}}(\boldsymbol{s}, z_0, \boldsymbol{r} - \boldsymbol{v}(\boldsymbol{r}) t) dt \right) d\boldsymbol{r} \\ &= \iint o(\boldsymbol{r}, 0) h_{\text{time-projected}}(\boldsymbol{s}, z_0, \boldsymbol{r}) d\boldsymbol{r} \\ &= h * o(\boldsymbol{s}), \end{aligned} \tag{6.1}$$

with $\boldsymbol{s}(t)$ the spatial coordinates of a pixel at time t, and $h_{3\mathrm{D}}(t)$ the depth-dependent PSF of the optical system. h is therefore the (2D) spatially-variant optical PSF projected over time and takes into account the movement of the object during the time interval Δt.

We need a displacement estimation function that is invariant to the sample shape or texture but co-variant with the sample's axial position. To that end, we chose an estimator of the local optical properties of the microscope objective that we integrate over time. We chose for $h_{3\mathrm{D}}$ a Zernike polynomial-based PSF model [16], [50].

6.2.3 Estimation of the displacement vector field

Similarly to [30], we want to train a CNN that extracts the displacement vector $\boldsymbol{v}(\boldsymbol{s})$ from the input image $i(\boldsymbol{s})$. We create a training set of $K = 400'000$ images taken from [27] that are blurred by spatially-variable PSFs. To do so, we first define for every k-th image, N non-overlapping 2D masks $m_k^n(\boldsymbol{s})$, with $n = 0, ..., N-1$. Then, we define, for every mask, a PSF $h_{\boldsymbol{v}_k^n}$ generated using the parameters $\boldsymbol{v}_k$ drawn from a uniform distribution ($[-1, 1]$ for the lateral component of $\boldsymbol{v}$ and $[0, 1]$ for the axial component of $\boldsymbol{v}$) and $z_k(\boldsymbol{s})$, which is the axial position where the object o is in focus. We get the final K training images by multiplying the masked input image by the PSFs in the Fourier domain:

$$i_k(\mathbf{s}) = \beta b_p\left(\lambda = \mathscr{F}^{-1}\left[\sum_{n=0}^{N-1} \mathscr{F}(h_{\boldsymbol{v}_k^n})\mathscr{F}(m^n * i_k)\right](\mathbf{s}), \boldsymbol{s}\right) + b_g(\boldsymbol{s}), \tag{6.2}$$

with β a number between 0 and 1 reflecting the camera quantum efficiency, $b_p(\lambda, \boldsymbol{s})$ a random variable following a Poisson distribution, and $b_g(\boldsymbol{s})$ a random variable following a zero-mean half-normal distribution. Since there are cases where the PSF estimation is not possible, e.g. where the sample lacks texture, such as in uniformly black or grey areas, we added a boolean parameter $w_k(\boldsymbol{s})$ (whose values can be either 0 or 1), which indicates the "legitimacy" of the sample (i.e is this image textured enough to yield useful information?).

We trained a U-Net CNN [105] with a ResNet encoder [60] pretrained on ImageNet [10], in order to predict, with the image $i(\boldsymbol{s})$ as input, the map of parameters $(\widetilde{\boldsymbol{v}}(\boldsymbol{s}), \widetilde{z}_k(\boldsymbol{s}), \widetilde{w}_k(\boldsymbol{s}))$ converted using cylindrical coordinates . We assessed in [30] that such network is robust to unwanted image degradations such as Poisson and Gaussian noise. We trained the network for 50 epochs in PyTorch [108] with RAdam [109] optimizing the following loss function:

$$E^{(k)} = \gamma\,(w_k(\boldsymbol{s}) - \widetilde{w}_k(\boldsymbol{s}))^2 + \frac{1 - w_k(\boldsymbol{s})}{U+1}\left[\sum_{u=1}^{U}(|v_u(\boldsymbol{s})| - |\widetilde{v}_u(\boldsymbol{s})|)^2 + (z_k(\boldsymbol{s}) - \widetilde{z}_k(\boldsymbol{s}))^2\right], \tag{6.3}$$

with $U = 3$ components in $\boldsymbol{v}$, γ a hyperparameter regulating the importance of the validity parameter w, that we set to 1 in our further experiments.

6.3 Experiments

We aim at defining the performance of the method using test data generated in the same way as the training data, but with a separate data set of $K_{\text{test}} = 5000$ images cropped at 224×224 pixel, preliminary acquired using a Leica DM 5500, a 10x/0.3 objective, and fixed fluorescent samples (HeLa cells actin (Alexa Fluor 635) and HeLa cells anti-α-catenin (Alexa Fluor 488)) (see Section 4.3). Specifically, we took sharp and immobile images and blurred them with two generated PSFs modeling different three-dimensional flow rates from a uniform distribution. We then used the CNN trained in Section 6.2.3 to predict the flow vector $\widetilde{\boldsymbol{v}}(\boldsymbol{s})$, the axial position $\widetilde{z}_0(\boldsymbol{s})$, and the "validity" parameter $w_k(\boldsymbol{s})$. Since it is a regression problem, our metric was set to be the squared Pearson correlation coefficient R^2 averaged over all dimensions.

We then turned to a more realistic experiment. Indeed, we generated a second synthetic testing dataset by simulating the flow in a cylinder where the camera and the focal plane are perpendicular to the flow direction. Due to the small DOF in microscopy, the effect of the cylinder curvature is negligible. The flow vector map $\boldsymbol{v}(\boldsymbol{s})$ is then similar to Fig. 6.2 (b). We

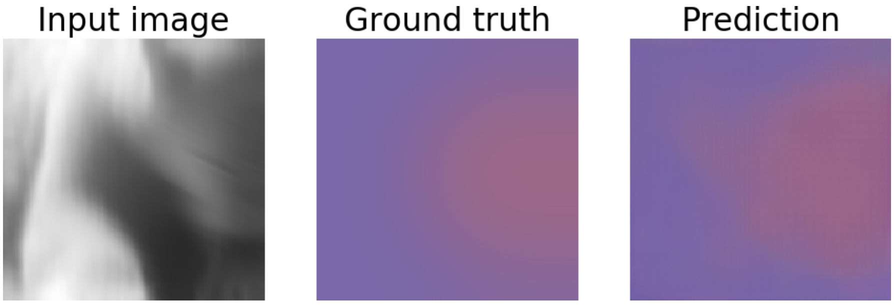


Figure 6.1: Velocity vector estimation from a single motion blurred image from the dataset acquired in Section 4.3 and two PSFs drawn from a random distribution. The network output and the ground truth vectors are represented in the RGB spectrum with $\widetilde{v}_1(\boldsymbol{s})$ in the red channel, $\widetilde{v}_2(\boldsymbol{s})$ in the green channel, and $\widetilde{v}_3(\boldsymbol{s})$ in the blue channel.

neglected as well the effects of the non-slip condition at the walls present in Poiseuille flow.

6.4 Results

Our experiments on simulated data confirm the network's capability to regress a pixel-wise motion vector from a single blurred image. Indeed, when it came to the task of estimating two different motion vector in two zones in an image, the network achieved a regression coefficient of $R^2 = 0.92$ averaged over all pixels of $N = 1000$ images of 224×224 pixel (see Fig. 6.1). Similarly, to retrieve the cylindrical flow profile in the second experiment, the regression score was computed at $R^2 = 0.91$ using the same conditions as before (see Fig. 6.3).

In all our experiments, the axial component was predicted with a systematically greater error than the lateral components. That could be explained by the confusion between an object with larger axial velocity, but started its motion right in focus, compared to an object with a smaller axial velocity, but whose motion happens out-of-focus. Both situations yielded similar-looking PSF since the generation of the PSF from the parameters $\boldsymbol{v}(\boldsymbol{s})$ and $\widetilde{z}_0(\boldsymbol{s})$ is not a perfectly bijective transformation.

During the training process, we noticed that the global accuracy was highly sensitive to the training set data size. Indeed, the network showed signs of over-fitting when it was trained with less than $K = 100'000$ images. In these experiments, we showed that parametric PSF regression could not only predict the depth of a surface (see Chapter 3), but the same method, with the right transformations and loss function, can predict the motion of this object from a single image.

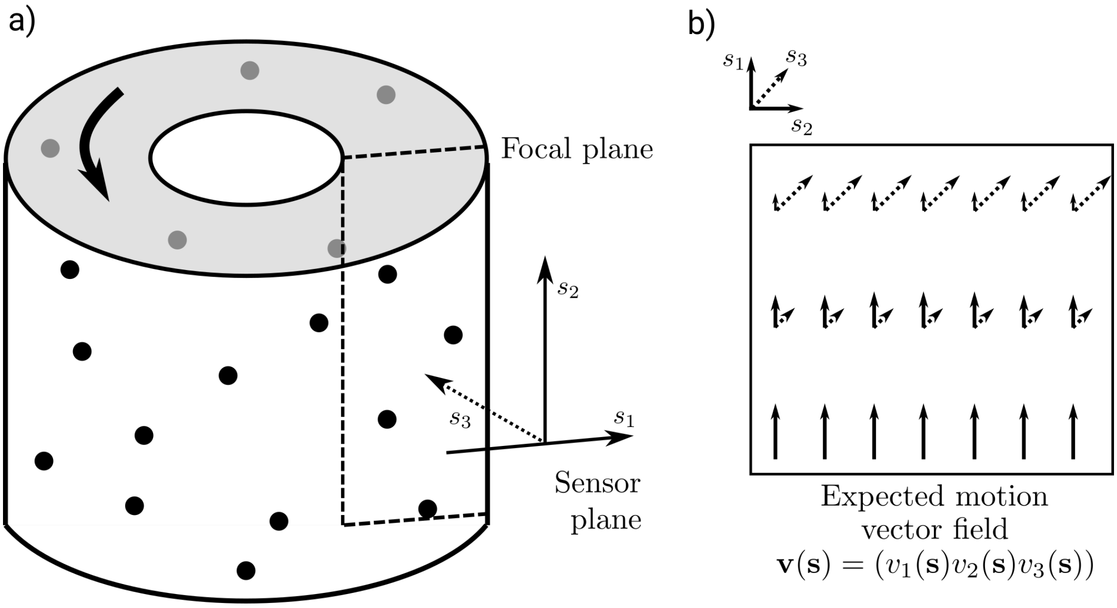

a)
Focal plane
s_2
s_3
s_1
Sensor plane
b)
s_1
s_3
s_2
Expected motion vector field
$\mathbf{v}(\mathbf{s}) = (v_1(\mathbf{s})v_2(\mathbf{s})v_3(\mathbf{s}))$

Figure 6.2: (a) Simulation of a flow in a cylinder. (b) simulation of its expected flow profile as captured by the camera. The flow vector $\boldsymbol{v}(\boldsymbol{s})$ has a greater lateral component in the bottom of the image, and a larger axial component in the top of the image.

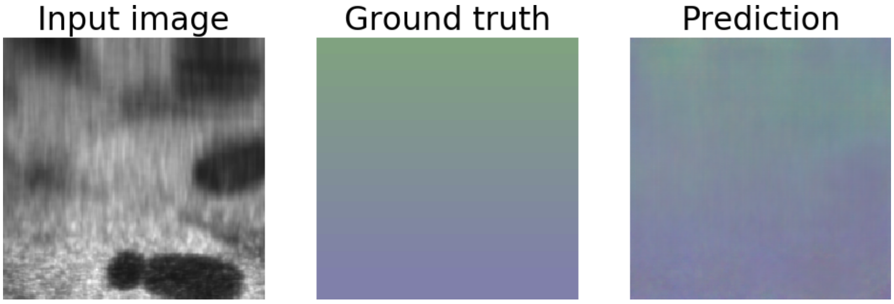


Figure 6.3: Velocity vector estimation from a single motion blurred image from the dataset acquired in Section 4.3 and a gradient of PSF mimicking the conditions of Fig. 6.2. The network output and the ground truth vectors are represented in the RGB spectrum with $\widetilde{v}_1(\boldsymbol{s})$ in the red channel, $\widetilde{v}_2(\boldsymbol{s})$ in the green channel, and $\widetilde{v}_3(\boldsymbol{s})$ in the blue channel.

7 Conclusion and outlooks

In this thesis, we have shown that CNNs, in particular residual networks, can be used to extract local blur characteristics from microscopy images in the form of parameters of a PSF model with only minimal knowledge about the optical setup and more specifically without any kind of PSF measurement, a step deemed specially tedious and time-consuming. Our PSF estimation method is robust to signal perturbations and does not need to be trained specifically on images of the target imaging system. This flexibility allows the user to perform, without taking measurements beyond the images of interest, a wide range of tasks in microscopy image processing, including deblurring, obtaining its tridimensional shape in a single shot, using the focus parameter in a fast converging auto-focus software or retrieve the object velocity in three dimensions.

The studies presented in this thesis gives evidence that algorithms already popular in photography can be applied in the environment of bio-microscopy. However, noise and optical models must be defined appropriately to fit to the specific high-NA, low-light, high noise environment of microscopy. More specifically, we set a framework, using a PSF and noise model based on physical properties of the wide-field microscope, that prevents the algorithms from diverging to hallucinations or to results that could not be physically explained.

The methods we developed have room for improvement by substituting the CNN by more complex DNN architectures. For example, [110], [111] achieved significant improvements by combining transformer networks, variational encoders, attention modules and a generative adversarial network (GAN) loss that would most likely better extract information from some textured parts of the image, while it would discard other parts or noise. However, more complex models require much heavier computing infrastructure in terms of GPU clustering and data management which are tasks that may not be readily available in a standard academic environment.

Deconvolution, depth estimation, autofocus and flow estimation are four direct applications that I discussed in this thesis, but there are plenty of other uses for PSF parameter regression that the scientific community could work on. One example of such new application would be to combine multiple locally out-of-focus images into one all-in-focus image without any kind of calibration procedure. Another example of extension would be to modify the flow estimation pipeline such that the motion vector is recorded over time. From that we could then deduce 3D trajectories in the case of a tracking experiment.

Finally, we showed that the integration of DNN features into classic optimization algorithms (e.g. RL for deconvolution and Newton-like optimizers for the auto-focus) can benefit both from the incredible accuracy and robustness of CNNs, and the dozens of years of mathematical validation of classic algorithms. Combined with the input of multi-modal sensors, actuators and real-time processing to integrate feedback loops based on complex features, this work is a step into the world of intelligent microscopes that combine real-time hardware and software to (semi-)blindly enhance, detect, and track biological samples.

Bibliography

[1] S. F. Gibson and F. Lanni, "Diffraction by a circular aperture as a model for three-dimensional optical microscopy," vol. 6, no. 9, pp. 1357–1367, 1989.

[2] D. Sage, L. Donati, F. Soulez, D. Fortun, G. Schmit, A. Seitz, R. Guiet, C. Vonesch, and M. Unser, "DeconvolutionLab2: An open-source software for deconvolution microscopy," *Methods*, vol. 115, pp. 28–41, 2017.

[3] A. Griffa, N. Garin, and D. Sage, "Comparison of deconvolution software in 3D microscopy: A user point of view—part 1," *G.I.T. Imaging & Microscopy*, vol. 12, no. 1, pp. 43–45, 2010.

[4] P. Grossmann, "Depth from focus," *Pattern Recognit. Lett*, vol. 5, no. 1, pp. 63–69, 1987.

[5] F. Aguet, D. Van De Ville, and M. Unser, "Model-based 2.5-D deconvolution for extended depth of field in brightfield microscopy," *IEEE Trans Image Process*, vol. 17, no. 7, pp. 1144–1153, 2008.

[6] A. Shihavuddin, S. Basu, E. Rexhepaj, F. Delestro, N. Menezes, S. M. Sigoillot, E. Del Nery, F. Selimi, N. Spassky, and A. Genovesio, "Smooth 2D manifold extraction from 3D image stack," *Nat Commun*, vol. 8, p. 15 554, 2017.

[7] E. Cuche, P. Marquet, and C. Depeursinge, "Simultaneous amplitude-contrast and quantitative phase-contrast microscopy by numerical reconstruction of Fresnel off-axis holograms," *Appl. Opt.*, vol. 38, no. 34, p. 6994, 1999.

[8] J.-A. Conchello and J. W. Lichtman, "Optical sectioning microscopy," *Nat Methods*, vol. 2, no. 12, pp. 920–931, 2005.

[9] P. Campisi and K. Egiazarian, *Blind Image Deconvolution: Theory and Applications*. CRC Press, 2017, 474 pp.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep Convolutional Neural Networks," in *NIPS*, Curran Associates, Inc., 2012, pp. 1097–1105.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 39, no. 6, pp. 1137–1149, 2017.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE CVPR*, 2014, pp. 580–587.

[13] S. J. Yang, M. Berndl, D. Michael Ando, M. Barch, A. Narayanaswamy, E. Christiansen, S. Hoyer, C. Roat, J. Hung, C. T. Rueden, A. Shankar, S. Finkbeiner, and P. Nelson, "Assessing microscope image focus quality with deep learning," *BMC Bioinformatics*, vol. 19, no. 1, 2018.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a Convolutional Neural Network for non-uniform motion blur removal," in *IEEE CVPR 2015*, 2015.

[16] A. Shajkofci and M. Liebling, "Semi-blind spatially-variant deconvolution in optical microscopy with local point spread function estimation by use of convolutional neural networks," in *IEEE ICIP*, 2018, pp. 3818–3822.

[17] D. He, D. Cai, J. Zhou, J. Luo, and S.-L. Chen, "Restoration of Out-of-Focus Fluorescence Microscopy Images Using Learning-Based Depth-Variant Deconvolution," *IEEE Photonics Journal*, pp. 1–1, 2020.

[18] M. Weigert, U. Schmidt, T. Boothe, A. Müller, A. Dibrov, A. Jain, B. Wilhelm, D. Schmidt, C. Broaddus, S. Culley, M. Rocha-Martins, F. Segovia-Miranda, C. Norden, R. Henriques, M. Zerial, M. Solimena, J. Rink, P. Tomancak, L. Royer, F. Jug, and E. W. Myers, "Content-aware image restoration: Pushing the limits of fluorescence microscopy," *Nat Methods*, vol. 15, no. 12, pp. 1090–1097, 2018.

[19] Y. Rivenson, Y. Zhang, H. Günaydın, D. Teng, and A. Ozcan, "Phase recovery and holographic image reconstruction using deep learning in neural networks," *Light Sci Appl*, vol. 7, no. 2, p. 17 141, 2018.

[20] Y. Wu, Y. Rivenson, H. Wang, Y. Luo, E. Ben-David, L. A. Bentolila, C. Pritz, and A. Ozcan, "Three-dimensional virtual refocusing of fluorescence microscopy images using deep learning," *Nat Methods*, 2019.

[21] H. Wang, Y. Rivenson, Y. Jin, Z. Wei, R. Gao, H. Günaydın, L. A. Bentolila, C. Kural, and A. Ozcan, "Deep learning enables cross-modality super-resolution in fluorescence microscopy," *Nat Methods*, vol. 16, no. 1, pp. 103–110, 2019.

[22] C. Belthangady and L. A. Royer, "Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction," *Nature Methods*, vol. 16, no. 12, pp. 1215–1225, 12 2019.

[23] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.

[24] K. Grünberg, O. Jimenez-del-Toro, A. Jakab, G. Langs, T. Salas Fernandez, M. Winterstein, M.-A. Weber, and M. Krenn, "Annotating Medical Image Data," in *Cloud-Based Benchmarking of Medical Image Analysis*, Cham: Springer International Publishing, 2017, pp. 45–67.

[25] D. P. Sullivan, C. F. Winsnes, L. Åkesson, M. Hjelmare, M. Wiking, R. Schutten, L. Campbell, H. Leifsson, S. Rhodes, A. Nordgren, K. Smith, B. Revaz, B. Finnbogason, A. Szantner, and E. Lundberg, "Deep learning is combined with massive-scale citizen science to improve large-scale image classification," *Nat Biotechnol*, vol. 36, no. 9, pp. 820–828, 2018.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 740–755.

[27] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 40, no. 6, pp. 1452–1464, 2018.

[28] Yuanhao Gong and I. F. Sbalzarini, "A Natural-Scene Gradient Distribution Prior and its Application in Light-Microscopy Image Processing," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 1, pp. 99–114, 2016.

[29] S. Kim, W. Kim, Y.-K. Noh, and F. C. Park, "Transfer learning for automated optical inspection," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2517–2524.

[30] A. Shajkofci and M. Liebling, "Spatially-Variant CNN-Based Point Spread Function Estimation for Blind Deconvolution and Depth Estimation in Optical Microscopy," *IEEE Transactions on Image Processing*, vol. 29, pp. 5848–5861, 2020.

[31] D. Sage, T.-A. Pham, H. Babcock, T. Lukes, T. Pengo, J. Chao, R. Velmurugan, A. Herbert, A. Agrawal, S. Colabrese, A. Wheeler, A. Archetti, B. Rieger, R. Ober, G. M. Hagen, J.-B. Sibarita, J. Ries, R. Henriques, M. Unser, and S. Holden, "Super-resolution fight club: Assessment of 2D and 3D single-molecule localization microscopy software," *Nat. Methods*, vol. 16, no. 5, pp. 387–395, 2019.

[32] E. Nehme, D. Freedman, R. Gordon, B. Ferdman, L. E. Weiss, O. Alalouf, T. Naor, R. Orange, T. Michaeli, and Y. Shechtman, "DeepSTORM3D: Dense 3D localization microscopy and PSF design by deep learning," *Nature Methods*, pp. 1–7, 2020.

[33] A. Krull, T. Vičar, M. Prakash, M. Lalit, and F. Jug, "Probabilistic Noise2Void: Unsupervised Content-Aware Denoising," *Front. Comput. Sci.*, vol. 2, 2020.

[34] H. P. Kao and A. S. Verkman, "Tracking of single fluorescent particles in three dimensions: Use of cylindrical optics to encode particle position," *Biophys. J.*, vol. 67, no. 3, pp. 1291–1300, 1994.

[35] X. Zhu, S. Cohen, S. Schiller, and P. Milanfar, “Estimating Spatially Varying Defocus Blur From A Single Image,” *IEEE Trans. on Image Process.*, vol. 22, no. 12, pp. 4879–4891, 2013.

[36] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. V. D. Hengel, and Q. Shi, “From Motion Blur to Motion Flow: A Deep Learning Solution for Removing Heterogeneous Motion Blur,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, 2017, pp. 3806–3815.

[37] S. Nah, T. H. Kim, and K. M. Lee, “Deep multi-scale convolutional neural network for dynamic scene deblurring,” in *IEEE CVPR*, 2017.

[38] J. W. Goodman, “Chapter 6-1,” in *Introduction To Fourier Optics*, 2nd Revised edition, Englewood, Colo: W.H.Freeman & Co Ltd, 2005, p. 129.

[39] B. Chalmond, “PSF estimation for image deblurring,” *CVGIP*, vol. 53, no. 4, pp. 364–372, 1991.

[40] N. Joshi, R. Szeliski, and D. J. Kriegman, “PSF estimation using sharp edge prediction,” in *IEEE CVPR*, 2008, pp. 1–8.

[41] A. Reuter, H.-P. Seidel, and I. Ihrke, “BlurTags: Spatially varying PSF estimation with out-of-focus patterns,” in *WSCG*, 2012, p. 9.

[42] J. Brauers, C. Seiler, and T. Aach, “Direct PSF estimation using a random noise target,” in *SPIE Electronic Imaging*, 2010, 75370B.

[43] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, “Understanding blind deconvolution algorithms,” *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 33, no. 12, pp. 2354–2367, 2011.

[44] D. A. Fish, A. M. Brinicombe, E. R. Pike, and J. G. Walker, “Blind deconvolution by means of the Richardson–Lucy algorithm,” *J. Opt. Soc. Am. A*, vol. 12, no. 1, pp. 58–65, 1995.

[45] F. Soulez, “A “Learn 2D, Apply 3D" method for 3D deconvolution microscopy,” in *IEEE ISBI*, 2014, pp. 1075–1078.

[46] J. Herbel, T. Kacprzak, A. Amara, A. Refregier, and A. Lucchi, “Fast point spread function modeling with deep learning,” *J. Cosmol. Astropart. Phys*, vol. 2018, no. 07, pp. 054–054, 2018.

[47] F. Aguet, D. Van De Ville, and M. Unser, “An accurate PSF model with few parameters for axially shift-variant deconvolution,” in *IEEE ISBI*, 2008, pp. 157–160.

[48] G. B. Airy, “On the diffraction of an object-glass with circular aperture,” *Trans. Cambridge Philos. Soc.*, vol. 5, p. 283, 1835.

[49] D. R. Iskander, M. J. Collins, and B. Davis, “Optimal modeling of corneal surfaces with Zernike polynomials,” *IEEE Trans Biomed Eng*, vol. 48, no. 1, pp. 87–95, 2001.

[50] F. von Zernike, “Beugungstheorie des schneidenver-fahrens und seiner verbesserten form, der phasenkontrastmethode,” *Physica*, vol. 1, no. 7, pp. 689–704, 1934.

[51] J. W. Goodman, *Introduction to Fourier Optics*. Roberts and Company Publishers, 2005, 520 pp.

[52] B. Zhang, J. Zerubia, and J.-C. Olivo-Marin, “Gaussian approximations of fluorescence microscope point-spread function models,” *Appl. Opt.*, vol. 46, no. 10, pp. 1819–1829, 2007.

[53] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient backprop,” in *Neural Networks: Tricks of the Trade*, Springer, 2012, pp. 9–48.

[54] F. Luisier, T. Blu, and M. Unser, “Image denoising in mixed poisson–gaussian noise,” *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 696–708, 2011.

[55] E. N. Gilbert and H. O. Pollak, “Amplitude distribution of shot noise,” *Bell System Technical Journal*, vol. 39, no. 2, pp. 333–350, 1960.

[56] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *IEEE CVPR*, 2009, p. 8.

[57] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet large scale visual recognition challenge,” *Int J Comput Vis*, vol. 115, no. 3, pp. 211–252, 2015.

[58] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition,” in *ICML*, 2014, pp. 647–655.

[59] D. Hendrycks and T. G. Dietterich, “Benchmarking neural network robustness to common corruptions and surface variations,” in *ICLR*, 2019.

[60] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE CVPR*, 2016, pp. 770–778.

[61] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, “Aggregated Residual Transformations for Deep Neural Networks,” in *IEEE CVPR*, 2017, pp. 5987–5995.

[62] G. Huang, Z. Liu, L. v d Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *IEEE CVPR*, 2017, pp. 2261–2269.

[63] K. Hechenbichler and K. Schliep, “Weighted k-nearest-neighbor techniques and ordinal classification,” Ludwig-Maximilians-Universität München, Working Paper 399, 2004.

[64] M. Hirsch, S. Sra, B. Scholkopf, and S. Harmeling, “Efficient filter flow for space-variant multiframe blind deconvolution,” in *IEEE CVPR*, 2010, pp. 607–614.

[65] N. Dey, L. Blanc-Féraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia, “Richardson–Lucy algorithm with total variation regularization for 3D confocal microscope deconvolution,” *Microsc. Res. Tech.*, vol. 69, no. 4, pp. 260–266, 2006.

[66] M.-A. Bray, A. N. Fraser, T. P. Hasaka, and A. E. Carpenter, “Workflow and metrics for image quality control in large-scale high-content screens,” *J Biomol Screen*, vol. 17, no. 2, pp. 266–274, 2012.

[67] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter, “Annotated high-throughput microscopy image sets for validation,” *Nature Methods*, vol. 9, no. 7, p. 637, 2012.

[68] E. Williams, J. Moore, S. W. Li, G. Rustici, A. Tarkowska, A. Chessel, S. Leo, B. Antal, R. K. Ferguson, U. Sarkans, A. Brazma, R. E. C. Salas, and J. R. Swedlow, “The image data resource: A bioimage data integration and publication platform,” *Nat Methods*, vol. 14, no. 8, pp. 775–781, 2017.

[69] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *ICLR*, 2015.

[70] B. Huang, W. Wang, M. Bates, and X. Zhuang, “Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy,” *Science*, vol. 319, no. 5864, pp. 810–813, 2008.

[71] A. Aristov, B. Lelandais, E. Rensen, and C. Zimmer, “ZOLA-3D allows flexible 3D localization microscopy over an adjustable axial range,” *Nat Commun*, vol. 9, no. 1, p. 2409, 2018.

[72] R. Morin, S. Bidon, A. Basarab, and D. Kouamé, “Semi-blind deconvolution for resolution enhancement in ultrasound imaging,” in *IEEE ICIP*, 2013, pp. 1413–1417.

[73] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[74] W. H. Richardson, “Bayesian-based iterative method of image restoration,” *J. Opt. Soc. Am.*, vol. 62, no. 1, pp. 55–59, 1972.

[75] T. Chan and Chiu-Kwong Wong, “Total variation blind deconvolution,” *IEEE Trans. on Image Process.*, vol. 7, no. 3, pp. 370–375, 1998.

[76] L. Denis, E. Thiébaut, F. Soulez, J.-M. Becker, and R. Mourya, “Fast approximations of shift-variant blur,” *Int. J. Comput. Vis*, vol. 115, no. 3, pp. 253–278, 2015.

[77] M. Temerinac Ott, O. Ronneberger, R. Nitschke, W. Driever, and H. Burkhardt, “Spatially variant Lucy-Richardson deconvolution for multiview fusion of microscopical 3D images,” in *IEEE ISBI*, 2011, pp. 899–904.

[78] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in *IEEE ACSSC*, vol. 2, 2004, pp. 1398–1402.

[79] J. Kotera, F. Šroubek, and P. Milanfar, “Blind deconvolution using alternating maximum a posteriori estimation with heavy-tailed priors,” in *CAIP*, Springer, 2013, pp. 59–66.

[80] J. Dong, J. Pan, Z. Su, and M.-H. Yang, “Blind Image Deblurring with Outlier Handling,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2497–2505.

[81] M. Jin, S. Roth, and P. Favaro, “Normalized Blind Deconvolution,” presented at the ECCV 2018, vol. 11211, Cham: Springer International Publishing, 2018, pp. 694–711.

[82] O. Whyte, J. Sivic, and A. Zisserman, “Deblurring shaken and partially saturated images,” *Int. J. Comput. Vis*, vol. 110, no. 2, pp. 185–201, 2014.

[83] D. Perrone and P. Favaro, "Total Variation Blind Deconvolution: The Devil Is in the Details," in *IEEE CVPR*, 2014, pp. 2909–2916.

[84] D. Liu, B. Wen, X. Liu, Z. Wang, and T. Huang, "When Image Denoising Meets High-Level Vision Tasks: A Deep Learning Approach," in *IJCAI*, 2018, pp. 842–848.

[85] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, no. 3, p. 1, 2008.

[86] A. Shajkofci and M. Liebling, "DeepFocus: A Few-Shot Microscope Slide Auto-Focus Using a Sample Invariant CNN-Based Sharpness Function," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, 2020, pp. 164–168.

[87] Y. Liron, Y. Paran, N. G. Zatorsky, B. Geiger, and Z. Kam, "Laser autofocusing system for high-resolution cell biological imaging," *J Microsc*, vol. 221, no. 2, pp. 145–151, 2006.

[88] L. Silvestri, M. C. Müllenbroich, I. Costantini, A. P. Di Giovanna, L. Sacconi, and F. S. Pavone, "RAPID: Real-time image-based autofocus for all wide-field optical microscopy systems," Bioengineering, BioRxiv, 2017.

[89] Y. Sun, S. Duthaler, and B. J. Nelson, "Autofocusing in computer microscopy: Selecting the optimal focus algorithm," *Microscopy Research and Technique*, vol. 65, no. 3, pp. 139–149, 2004.

[90] V. Magidson and A. Khodjakov, "Circumventing photodamage in live-cell microscopy," *Methods Cell Biol*, vol. 114, pp. 545–560, 2013.

[91] L. Wei and E. Roberts, "Neural network control of focal position during time-lapse microscopy of cells," *Sci Rep*, vol. 8, no. 1, p. 7313, 2018.

[92] S. Jiang, J. Liao, Z. Bian, K. Guo, Y. Zhang, and G. Zheng, "Transform- and multi-domain deep learning for single-frame rapid autofocusing in whole slide imaging," *Biomed. Opt. Express*, vol. 9, no. 4, p. 1601, 2018.

[93] H. Pinkard, Z. Phillips, A. Babakhani, D. A. Fletcher, and L. Waller, "Deep learning for single-shot autofocus microscopy," *Optica*, vol. 6, no. 6, p. 794, 2019.

[94] A. Edelstein, N. Amodaj, K. Hoover, R. Vale, and N. Stuurman, "Computer control of microscopes using µManager," in *Current Protocols in Molecular Biology*, vol. 92, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2010, pp. 1–17.

[95] S. Yazdanfar, K. B. Kenny, K. Tasimi, A. D. Corwin, E. L. Dixon, and R. J. Filkins, "Simple and robust image-based autofocusing for digital microscopy," *Opt. Express*, vol. 16, no. 12, p. 8670, 2008.

[96] A. F. J. Moffat, "A Theoretical Investigation of Focal Stellar Images in the Photographic Emulsion and Application to Photographic Photometry," *Astronomy and Astrophysics*, vol. 3, p. 455, 1969.

[97] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Proceedings*, 2010, pp. 2432–2439.

[98] Hanghang Tong, Mingjing Li, Hongjiang Zhang, and Changshui Zhang, "Blur detection for digital images using wavelet transform," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 17–20, 2004.

[99] M. Liebling and M. Unser, "Autofocus for digital Fresnel holograms by use of a Fresnelet-sparsity criterion," *J. Opt. Soc. Am. A*, vol. 21, no. 12, pp. 2424–2430, 2004.

[100] S. K. Nayar and Y. Nakagawa, "Shape from focus: An effective approach for rough surfaces," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1990, 218–225 vol.2.

[101] J. H. Price and D. A. Gough, "Comparison of phase-contrast and fluorescence digital autofocus for scanning microscopy," *Cytometry*, vol. 16, no. 4, pp. 283–297, 1994.

[102] J. M. Mateos-Pérez, R. Redondo, R. Nava, J. C. Valdiviezo, G. Cristóbal, B. Escalante-Ramírez, M. J. Ruiz-Serrano, J. Pascau, and M. Desco, "Comparative evaluation of autofocus algorithms for a real-time system for automatic detection of Mycobacterium tuberculosis," *Cytometry*, vol. 81A, no. 3, pp. 213–221, 2012.

[103] U. Ali and M. T. Mahmood, "Analysis of blur measure operators for single image blur segmentation," *Applied Sciences*, vol. 8, no. 5, p. 807, 2018.

[104] Y. LeCun, "Learning invariant feature hierarchies," in *Computer Vision – ECCV 2012. Workshops and Demonstrations*, A. Fusiello, V. Murino, and R. Cucchiara, Eds., 2012, pp. 496–505.

[105] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., vol. 9351, Cham: Springer International Publishing, 2015, pp. 234–241.

[106] T. Corpetti, D. Heitz, G. Arroyo, E. Mémin, and A. Santa-Cruz, "Fluid experimental flow estimation based on an optical-flow scheme," *Exp Fluids*, vol. 40, no. 1, pp. 80–97, 2006.

[107] O. Brina, R. Ouared, O. Bonnefous, F. van Nijnatten, P. Bouillot, P. Bijlenga, K. Schaller, K.-O. Lovblad, T. Grünhagen, D. Ruijters, and V. M. Pereira, "Intra-Aneurysmal Flow Patterns: Illustrative Comparison among Digital Subtraction Angiography, Optical Flow, and Computational Fluid Dynamics," *American Journal of Neuroradiology*, vol. 35, no. 12, pp. 2348–2353, 2014.

[108] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

[109] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the Variance of the Adaptive Learning Rate and Beyond," in *ICLR 2020*, 2020.

[110] Z. Wan, B. Zhang, D. Chen, P. Zhang, D. Chen, J. Liao, and F. Wen, "Bringing Old Photos Back to Life," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, 2020, pp. 2744–2754.

[111] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, “Learning Texture Transformer Network for Image Super-Resolution,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, 2020, pp. 5790–5799.

[112] V. Gardeux, F. P. A. David, A. Shajkofci, P. C. Schwalie, and B. Deplancke, “ASAP: A web-based platform for the analysis and interactive visualization of single-cell RNA-seq data,” *Bioinformatics*, vol. 33, no. 19, pp. 3123–3125, 2017.

[113] N. Hadadi, J. Hafner, A. Shajkofci, A. Zisaki, and V. Hatzimanikatis, “ATLAS of Biochemistry: A Repository of All Possible Biochemical Reactions for Synthetic Biology and Metabolic Engineering Studies,” *ACS Synth. Biol.*, vol. 5, no. 10, pp. 1155–1166, 2016.

Publications

Published

A. Shajkofci and M. Liebling, "Spatially-Variant CNN-Based Point Spread Function Estimation for Blind Deconvolution and Depth Estimation in Optical Microscopy," *IEEE Transactions on Image Processing*, vol. 29, pp. 5848–5861, 2020

A. Shajkofci and M. Liebling, "DeepFocus: A Few-Shot Microscope Slide Auto-Focus Using a Sample Invariant CNN-Based Sharpness Function," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2020, pp. 164–168

Shajkofci A. & Liebling M., "Free annotated data for deep learning in microscopy? A hitchhiker's guide", *Photoniques (EOS) 104*, EDP Sciences, 2020

A. Shajkofci and M. Liebling, "Semi-blind spatially-variant deconvolution in optical microscopy with local point spread function estimation by use of convolutional neural networks," in *IEEE ICIP*, 2018, pp. 3818–3822

V. Gardeux, F. P. A. David, A. Shajkofci, *et al.*, "ASAP: A web-based platform for the analysis and interactive visualization of single-cell RNA-seq data," *Bioinformatics*, vol. 33, no. 19, pp. 3123–3125, Oct. 1, 2017

N. Hadadi, J. Hafner, A. Shajkofci, *et al.*, "ATLAS of Biochemistry: A Repository of All Possible Biochemical Reactions for Synthetic Biology and Metabolic Engineering Studies," *ACS Synth. Biol.*, vol. 5, no. 10, pp. 1155–1166, Oct. 21, 2016

In preparation

Shajkofci A. & Liebling M., "Assessing Flow in Light Microscopy with Deep Learning and Applications in Cardiac Imaging", *IEEE International Symposium on Biomedical Imaging (ISBI)*, 2021

Presentations

Shajkofci A., “DeepFocus: A Few-Shot Microscope Slide Auto-Focus Using a Sample Invariant CNN-Based Sharpness Function”, *IEEE International Symposium on Biomedical Imaging (ISBI)*, 2020

Shajkofci A., “DeepFocus: A Few-Shot Microscope Slide Auto-Focus Using a Sample Invariant CNN-Based Sharpness Function”, *Quantitative BioImaging Conference (QBI)*, 2020

Shajkofci A., “Semi-blind spatially-variant deconvolution and focus estimation in optical microscopy by use of convolutional neural networks”, *Quantitative BioImaging Conference (QBI)*, 2019

Shajkofci A., “Semi-blind spatially-variant deconvolution in optical microscopy using convolutional neural networks”, *Swiss Machine Learning Days*, 2018

Funding

This thesis has been funded by the Swiss National Science Foundation under Grant 206021_164022 and Grant 200020_179217, and Valais/Wallis Ambition.

ADRIAN SHAJKOFCI

Swiss | Champs du Bourg 28 | 1920 Martigny | 079 412 36 98 | adrian.sh@mycable.ch

SCIENTIFIC SKILLS

- Background in maths, applied mathematics, physics, optics, statistics, cellular and molecular biology, genetics, biochemistry, neurosciences, biophysics, electronics.
- Proficiency in analog and digital signal processing for multidimensional data, sampling, filtering, statistical analysis.
- Excellent skills in data handling, dimensionality analysis and reduction, clustering, classification, regression using machine learning strategies.
- Good Skills in optimization, simulation, cost function design, data processing pipeline design, neural networks, deep learning, convolutional neural networks, autoencoders.
- Good skills in optics, microscope design, hardware and software.
- Good skills in embedded electronics, IoT integration.
- Good skills in graph theory, optimal pathway generation, connectivity and optimization.
- Good knowledge of bioinformatics, genetics, systems biology and systems chemistry.
- Basic biological lab and methods knowledge.
- Good planning skills and task management. Ease at taking decisions.
- Languages: native French speaker, English written and spoken, German read.

COMPUTER SCIENCE SKILLS

- Excellent proficiency in C++ / Python / SQL / PHP programming languages with focus on memory management, performance and scalability.
- Good knowledge of C / C# / Matlab / R / Java / HTML / CSS / Javascript programming languages.
- Knowledge of main web development frameworks (Vue.JS, Laravel, Django, React)
- Database design and optimization for MySQL, Oracle, Redis, HDF.
- Neural network architectures using Caffe, PyTorch, Tensorflow and DLib.
- Setup and administration of Linux / Windows Server servers, clusters and virtualized machines (ESXi, XenServer, Docker).
- Source code distribution and management systems.

PROFESSIONAL EXPERIENCE

Doctoral assistant, Computational Bioimaging Group, IDIAP Research Institute 2016-2020

[Martigny, Lausanne, Switzerland]

- Signal processing for digital (microscopy) images. Development of convolutional neural networks for spatial super-resolution (blind deconvolution), temporal super-resolution (blind supervised time-lapse interpolation) and pipeline parameter optimization.
- Development of a robotics image acquisition platform aimed at reproducible research.

Architecture administrator, Laboratory of Computational Systems Biotechnology, EPFL 2016-2018

[Lausanne, Switzerland]

- In silico enzymatic reactions and pathways as a service. Database distribution and server API. Server load scaling. System architecture design and optimization for big data and large-scale computing.

Software engineer, Laboratory of Computational Systems Biotechnology, EPFL 2014-2016

[Lausanne, Switzerland]

- Development of a web-based software suite (C++ / PHP / SQL / JS) for in silico enzymatic pathways visualization, search and comparison. Visualization of results using dynamic graphs & charts.
- Software development (C++/ SQL) of in silico enzymatic reactions and pathways generation.

Software engineer (Master's thesis), Laboratory of Systems Biology and Genetics, EPFL 2016

[Lausanne, Switzerland]

- Development of a web-based software suite (Java / JavaScript / R) for single-cell gene expression analysis. Integration of multiple normalization, filtering, dimensionality reduction and clustering algorithms. Visualisation of the results in 2D and web-based 3D (WebGL).

Software engineer (internship), Saphetor SA, EPFL Innovation Park 2015

[Lausanne, Switzerland]

- Big data for patient-based human genome annotation. Genetic databases integration in C++ / SQL.

Founder and CTO, Coronasense 2020-present

- Hardware and software development of an autonomous temperature sensor following the COVID-19 outbreak.
- Sourcing and manufacturing management.

Founder and head software engineer, Darluok Server 2006-present

- Active development of a real-time game server in C++/SQL. 480'000 users are registered with more than 4000 simultaneous connections.
- Development of PHP/SQL frontend and backend.
- Creation of UNIX clusters and automation of maintenance operations.

Programmer Analyst (internship), C6 Intelligence 2010

[Basingstoke, United Kingdom]

- Name recognition algorithm development (PHP / Visual Basic).

Teaching Assistant in Computer Science (Prof. J. Sam), EPFL, Lausanne, Suisse 2013–2014

Teaching Assistant in Computer Science (Prof. J. Rougemont), EPFL, Lausanne, Suisse 2014-2015

Audio Engineering Society member (AES) 2014-present

Master's degree (MSc) in Bioengineering, EPFL, Lausanne, Suisse 2014-2016

Bachelor's degree (BSc) in Life Sciences and Engineering, EPFL, Lausanne, Suisse 2010-2013

Swiss Federal Scientific Maturity Diploma, Collège de l'Abbaye, St-Maurice, Suisse 2005-2010