Machine Learning for Detection and Localization of Motion-Blurred Nano-Emitters in Single-Particle Tracking

A simulation study

by

Kevin Jansen, BSc

A thesis presented for the degree of

Master of Science

Supervised by:

Erik Maris, MSc

Dr. Florian Meirer

Dr. Freddy Rabouw

August 2022

Student number:

5682118

Utrecht University

INORGANIC CHEMISTRY & CATALYSIS

Layman’s summary

Catalysis is a field of great importance in current society as it allows for the conversion of molecules to useful products such as transport fuels and fertilizing agents. Therefore, without catalysis, the world population could not have grown from two to seven billion people in the last century. These catalysts are often designed as three-dimensional intertwining pore structures, in which molecules diffuse to be converted. However, sometimes molecules get stuck or regions in the catalyst are inaccessible because of blocked pores, hampering the efficiency of the catalyst. Single-particle tracking is a powerful technique in which molecules can be recorded whilst diffusing inside the pores using microscopy. Thereafter, all positions of the particles in each frame of the recording are determined, and subsequently their paths are reconstructed by linking their positions. Eventually, these paths could be used to recreate the accessible pore structure of catalysts. However, when molecules move fast under the microscope, their signal is spread out in the recordings, making them difficult to detect because of the presence of noise. A solution would be to create an algorithm that can detect the molecules from the noise, but this remains a difficult problem because of the many dimensions attaining to it. Therefore, this algorithm could be approximated instead using convolution neural networks, which are known to be powerful in approximating functions. Convolutional neural networks are computational processing systems that somewhat mimic the human brain in image recognition by detecting features. However, they first need to be “trained” to know which features it needs to look for to e.g. classify molecules from noise. The training is performed by giving them images with a label containing a ground truth, e.g. there is a particle located in this image or there is not. Therefore, simulations are necessary as only then the ground truths are truly known. Additionally, the simulations should reflect real examples of motion-blurred emitters as otherwise the performance of the neural networks will not reflect the performance when they are used in practice. Therefore, in this thesis, we tuned simulation software to resemble a non-simulated dataset, and subsequently simulated a dataset to train the neural networks to be able to detect molecules and predict their average coordinates in the recordings. We can only estimate the average coordinates as during the acquisition of a single frame in the recording the molecules moved, and therefore single x and y-coordinates are not defined. Subsequently, the performance of the neural networks were compared to other methods published in literature. As a result, the detection performance turned out to work ~500% more optimal than the method published in literature. However, the precision in average coordinate

estimation worked ~40% less optimal than software from the literature, i.e. phasor localization. Therefore, a combination of our detection convolutional neural network with the phasor method for estimating average coordinates would be a perfect combination to map pore structures of catalysts.

Abstract

Understanding diffusion of reactants in heterogeneous catalysts is vital for the optimization of their catalytic performance, as limited mass transfer can limit catalytic activity. A powerful method to investigate diffusion is single-particle tracking (SPT), in which often fluorescent nano-sized objects (emitters) e.g. quantum dots or reactant molecules are recorded whilst diffusing by means of widefield fluorescent microscopy. Subsequently, the paths (trajectories) of the emitters are reconstructed and can be used to reveal local heterogeneities in a catalyst, recreate pore structures and obtain diffusion constants for single emitters. However, when emitters move fast under the microscope, their recorded emission signal is spread out and more difficult to detect, referred to as motion-blur. Nevertheless, trajectories can be made by estimating the coordinates (localization) of the average positions of the motion-blurred emitters. However, creating an algorithm for a high dimensional problem like this could be difficult, and therefore easier to approximate using convolutional neural networks (CNNs). In this thesis, we investigated the use of CNNs to detect and localize the average positions of the motion-blurred emitters that diffuse in two dimensions using simulations. As a result, CNNs could potentially create trajectories of motion-blurred particles with higher precision than currently possible, and therefore the pore structure of a catalyst could be mapped with higher precision. Therefore, we used simulation software to create a dataset containing motion-blurred emitters of which the process consisted of four distinct steps: simulating diffusion paths, convolving point spread functions, adding background noise and lastly applying a camera noise model. We tuned the parameters in these four steps so that the resulting simulated dataset resembled a non-simulated SPT dataset, obtained from previous research. Consequently, the emitters' diffusion constants in the simulated dataset ranged from 4e−12 to 8e−13 $m^2\ s^{-1}$, and the emitters' intensity ranged from 0 to 312 photons/emitter/frame. Moreover, we recorded dark images to obtain model parameters that allowed us to recreate the camera noise profile observed in the experiment. Subsequently, we validated the camera model to work optimally at intensities of >8 photons/pixel/frame. Furthermore, we matched the background noise to the non-simulated dataset resulting in 10 photons/pixel/frame, which ensured the simulated dataset to be in a range where the camera model works optimally. Subsequently, we simulated frames containing motion-blurred emitters and frames containing only noise, of which we used 50.000 of both to train a classification CNN for detecting emitters. Thereafter, we used ~500.000 frames containing motion-blurred emitters to train a regression CNN for localization. Hereafter, we

simulated four validation sets of 125.000 frames containing motion-blurred emitters, in which we used combinations of diffusion constants 8e−13 and 4e−12 $m^2 s^{-1}$, and emitter intensities of 135 and 312 photons/emitter/frame to test the detection and localization CNN in different regimes. Subsequently, we compared the detection and localization CNN with published software, where the classification CNN performed increasingly better than benchmarking software when the validation sets got more complex, i.e. the validation sets with lower signal-to-noise ratios. Consequently, the classification CNN managed to obtain ~500% more correct detections in the validation set with the lowest signal-to-noise ratio. However, the localization CNN did not outperform benchmarking localization software, as it performed with a localization precision of ~150 nm for the validation set with the lowest signal-to-noise ratio, whereas the benchmarking phasor localization method had a localization precision of ~105 nm. In conclusion, a combination of our detection CNN with phasor localization results in the most detected emitters with the highest localization precision.

List of Abbreviations

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
SPT	Single-particle tracking
CNN	Convolutional neural network
PSF	Point spread function
SNR	Signal-to-noise ratio
ROI	Region of interest image
CRLB	Cramer-Rao lower bound
EMCCD	Electron multiplying charge coupled device
LSE	Least square estimation
MLE	Maximum likelihood estimation
ND	Numerical density
TP	True positives
FP	False positives

Mathematical Symbols

D	Diffusion constant
k	Number of dimensions
t	Time
n_{ie}	Number of input electrons for electron multiplying unit
n_{oe}	Number of output electrons of electron multiplying unit
n_{ic}	Number of image counts (pixel values in resulting image)
c	Spurious charge parameter
c_b	Camera bias
g	EM gain parameter
r	Readout noise parameter
i	Light intensity parameter
q	Quantum efficiency parameter
σ	Localization precision
$°C$	Degrees Celsius
d	Optical density
I_0	Incident intensity
I_T	Transmitted intensity

Acknowledgements

I could not have wished for a better master thesis opportunity than the one that I had been offered. The combination of a thesis including my field, chemistry, with my passion, programming, is not something common. I deeply want to thank Erik for the effort that he spent in supervising me, which must have been a lot of work. Thank you, Erik! I have become well experienced in using Matlab, which is a skill I am profoundly proud of. And I was able to obtain some experience in machine learning, which was something I truly wanted to learn. You have also taught me a lot about the single-particle tracking world, which to me seems like a technique that could be very important in the future, especially in combination with machine learning. So perhaps I can extend my career in the direction of single particle-tracking. Next to the serious business, we also had a lot of laughs together which I only could have hoped for to have with my supervisor. Even though the past year was a lot of hard work, I will cherish this as the most fun year I had during my education. Furthermore, I would like to thank Florian for always being positive whenever I discussed some of my results in our weekly meetings, which definitely encouraged me during my thesis. You always thought along with the problems I had and came up with potential solutions and new ideas that I could try out. My master thesis experience would not have been the same without having you as my supervisor. Thereafter, I would like to thank Freddy for the occasional feedback whenever I gave a presentation at your group, it was always very useful! Being able to visit your group was especially helpful whenever I had a more mathematical/physical problem such as simulating camera noise. Next, I want to thank everyone that was part of our weekly pore space meetings. Having a group like this to discuss our problems and results definitely helps to accelerate our research. Lastly, having other students around made my thesis a lot more fun, so Bas, Martijn, Renan, Sonja, Floor, and many more of you, thank you! I will remember all the fun times we had, at the parties and e.g. the fun events. All in all, I want to thank everyone for the amazing time I had during my master thesis.

Table of Contents

1. Introduction

Catalysis plays an important role in current society: it allowed for the rapid expansion of the world population during the last century by exploiting the catalysed Haber-Bosch reaction process, and we would not have access to transportation fuels as a catalyst is required to break down the larger petroleum molecules into smaller useful products[1–4]. These two examples utilize a heterogenous catalyst, where usually a liquid or gas phase reactant diffuses to the surface of the catalytic phase to undergo a chemical reaction process[5,6]. One key component of catalysis is the relation of increasing reaction rate with increasing surface area, as the number of active sites increases[5]. Consequently, catalysts are frequently prepared with large intertwining pore structures containing channels that can be as small as a few nanometres in diameter[7]. The investigation of reactant molecules diffusing in these pore networks is crucial, as the catalytic activity can be limited due to limited mass transfer[8]. A powerful method for investigating the pore network of catalysts is single-particle tracking (SPT), which can be used to reveal local heterogeneities in catalysts, recreate the accessible pore network and obtain the diffusion constant of individual molecules[8–12].

The workflow of SPT is summarized in Figure 1, in which first a sample is illuminated with a parallel beam of monochromatic light, and consequently nano-sized objects e.g. quantum dots or single molecules (emitters) emit light of a different colour, which is subsequently recorded utilizing a wide field fluorescent microscope[11]. The emission signal is diffracted in the optical microscope before it reaches the camera, and since the emitters can be considered point sources, the observed signal has a certain spatial intensity distribution known as the point spread function (PSF)[13,14]. Eventually, a movie is acquired consisting of sequential frames with the diffracted signal of various emitters visible at different locations in each frame. Thereafter, a detection step is necessary where the emitters are ‘’cut out’’ of the frames resulting in several smaller images solely containing the emission signal of the emitters, called region of interest images (ROIs). Subsequently, a set of ROIs of individual emitters are isolated by performing a nearest neighbour search over the subsequent frames (Figure 1a). Subsequently, the coordinates in the ROIs are estimated called localization, for which various methods have been developed. Thereafter, the positions are linked to reconstruct the emitters’ path, referred to as a trajectory (Figure 1b). Normally, emitters diffuse slowly, i.e. near stationary, during the frame acquisition of the movie, resulting in a near-perfect PSF and therefore optimal localizations (Figure 1b). However, when observing emitters that move fast during frame

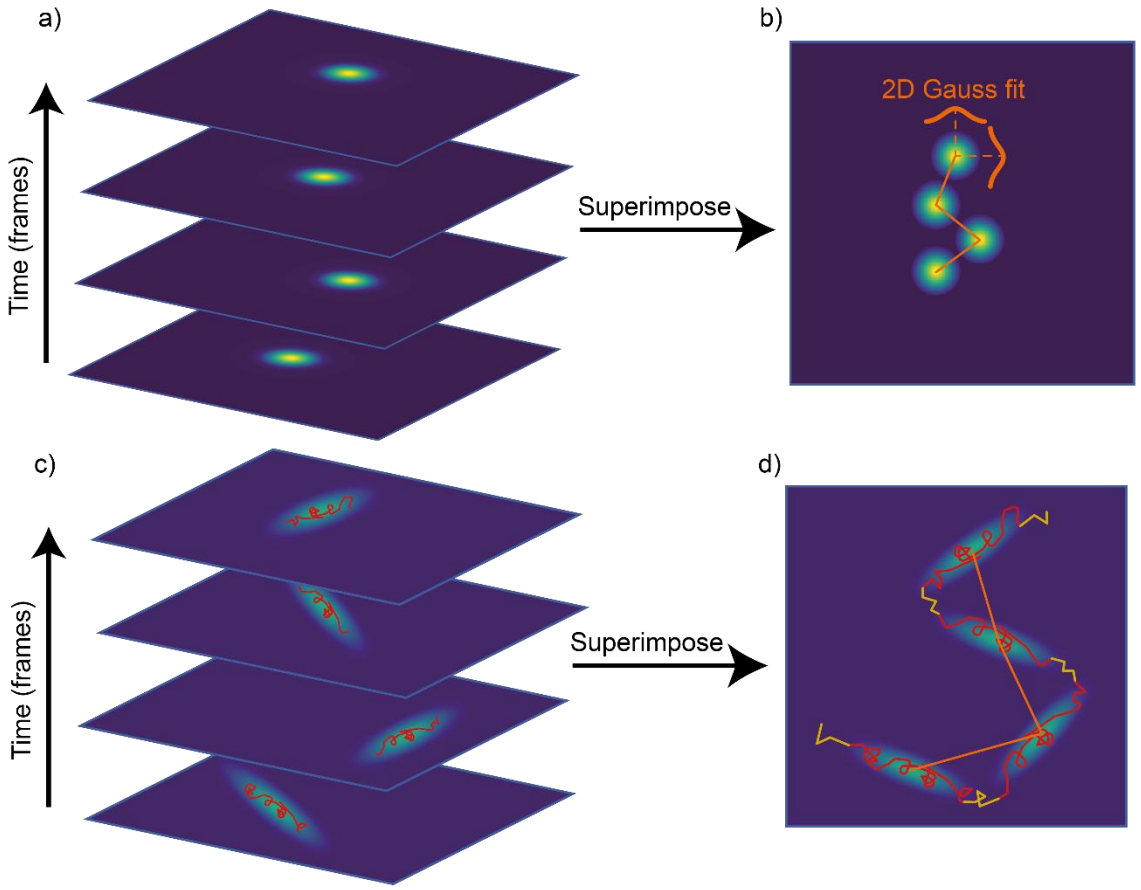


Figure 1: *a) Four isolated ROIs of a movie containing a single diffusing emitter. b) The emission spots can be well fitted with a 2D-Gaussian localize with nanometre precision*[9,11,16,17]*. By superimposing the frames a trajectory can be reconstructed by connecting the coordinates of the emitter. c) Four isolated ROIs containing an emitter that moved during frame acquisition. The red line is the diffusion path during frame acquisition which is overlaid with the motion-blurred signal observed in the movie. d) By superimposing the frames of a motion-blurred emitter the trajectory can be reconstructed by linking the average locations of the emitter in each emission spot (orange line)*[10,15]*. The yellow lines are parts of the diffusion path not imaged because there is dead time between frame acquisitions.*

acquisition of the movie, their emission signal is spread out, referred to as motion-blur (Figure 1c)[10,15]. Since the emitters move during frame acquisition, single coordinates are not defined. Nevertheless, the average position of the emitters can be localized and a trajectory reconstructed by linking the average coordinates in each frame of the emitters (Figure 1d)[10,15]. Therefore, from now on, when referring to localization of motion-blurred emitters, we refer to localizing their average positions.[10,13–15]

Various localization methods have been developed for the localization of emitters, e.g. fitting with a two dimensional (2D) Gaussian, estimating the centroid position, and phasor fitting. The centre of the PSF can be well fitted with a 2D-Gaussian, and as a result the coordinates can be localized with nanometre precision, i.e. down to ~2 nm[9,11,16,17]. However, in the case of motion-blur, the 2D-Gaussian model fits the emission signal less optimally,

resulting in localization errors. Furthermore, the centroid estimator localizes emitters by calculating the weighted centroid position, which could be an interesting method for the localization of motion-blurred emitters as it is independent of motion-blur, but has the downside of a known bias towards the middle of the ROIs because of noise[18]. Moreover, phasor fitting calculates the Fourier coefficients of emission signals for localization, but has the downside of being dependent of ROI size[19].

There is limited information published in the literature about localizing motion-blurred emitters. Regardless, Deschout et al. reported the localization precision for moving emitters using a centroid and 2D-Gaussian fitting localization method[15]. They concluded that centroid localization and Gaussian fitting show a similar localization precision for emitters that move axially during frame acquisition, i.e., in the *z*-direction, but the Gaussian localization precision breaks down for lateral movement, i.e., in the *xy*-direction (which also includes three-dimensional (3D) motion)[15]. They reason that the deterioration of the Gaussian localization precision is because of the deformation of the PSF during lateral movement, resulting in a less optimal fit[15]. Lastly, they tested localization with an ellipsoidal version of a 2D Gaussian, but this only resulted in a slight increase of localization precision[15]. Moreover, Vestergaard et al. reported the precision of centroid localization and 2D-Gaussian fitting as a function of emission intensity and diffusion constant solely for lateral movement. They report that high emission intensities result in higher localization precisions, but the localization precision of both methods deteriorates rapidly with increasing diffusion constants, regardless of emission intensity[12]. Additionally, a localization method named phasor fitting was published by Martens et al., but has not yet been tested on motion-blurred emitters[19]. For near-stationary emitters, a theoretical limit of localization precision is known according to the Fisher's information limit called the Cramer-Rao lower bound (CRLB). However, stochastic motion is not easily implemented in this information limit and therefore a theoretical limit is not known for precisions of methods used for localizing motion-blurred emitters[15]. In conclusion, the localization precisions of the available methods are not optimal for localizing motion-blurred emitters, have not been tested yet, or are possibly already at their limit.

To localize motion-blurred emitters with higher precision than published in the literature, an algorithm should be developped attaining for all dimensions of the problem: the random shape of the emission signal because of random diffusion, the distorted emission signal because of background and camera noise, the extremely low signal-to-noise ratio (SNR), the change of the PSF shape and intensity when imaging emitters out of focus etc. However, the

latter dimension of the problem is not relevant for this thesis as movement is kept in 2D. Developping a complicated algorithm like this can be difficult. Therefore, the algorithm could be approximated by using convolutional neurals networks (CNN).

CNNs are widely known as function approximaters. They are computational processing systems used for image recognition that are heavily inspired on the neurons in a brain, and work by detecting features in the image, i.e edges, corners, squares etc.[20,21]. However, the CNNs first need to be trained to know which features correspond to which output classes, with output classes e.g. coordinates of emitters. Therefore, simulations need to be performed as only then ground truths are truly known. Subsequently, a trained CNN can be used as a function e.g. by inputting an image containing an emitter to retrieve its coordinates. As CNNs are able to approximate *any* function, they have potential to approximate the highly dimensional algorithm to localize motion-blurred emitters, and therefore potentially performing with higher localization precisions than currently possible. Consequently, pore networks of catalysts could be mapped with higher precision. Furthermore, CNNs could be developed to detect emitters from noise in an SPT movie before the localization step, as this can be difficult because of low SNRs, where random noise can resemble the emission signal of emitters. As a result, a function that is able to detect more emitters than regular algorithms result in longer trajectories, which can subsequently be used to estimate the diffusion constant of single emitters with higher precision[9].[20,21]

The use of CNNs to detect and localize motion-blurred emitters has not yet been reported in literature. Nevertheless, CNNs have already proven useful in similar studies where emission signals of non-motion blurred emitters were used to retrieve certain information. The emission signals contain this information because the shape of the PSF depends on variables such as the emitters' axial position[22–25], emission wavelength of the emitter (colour)[26,27], the orientation of an emitter[28] etc. Consequently, CNNs can be trained to derive this information from the emission signals. For example, Kim et al. published a CNN trained for axial localization of emitters located between 400 nm above and below the focus of the microscope, which performed with a localization precision of ~1.5x the CRLB for localizing the *z*-coordinate of emitters located between -400 and +100 nm out of focus, while almost equal to the CRLB when localizing *z* for emitters located between +100 and +400 above focus[29]. Additionally, their CNN was trained for classifying between two emission wavelengths, which performed with >90% accuracy as long as the intensities of the emission signals were >~3000 photons[29]. Moreover, Zhang et al. published a CNN (smNet) for axial localization which

showed similar performance as the axial localization precision Kim et al. published, while also predicting the orientation of emitters and wavefront distortion[30]. Furthermore, Boyd et al. published a CNN (DeepLoco) used for the detection and localization of multiple emitters present in a single frame attached to a larger cellular structure, with the ability to analyse a 20.000-frame movie in a single second[31]. Moreover, Zelger et al. published a CNN for the localization of the 3D coordinates of emitters and compared it to 2D-Gaussian fitting, which showed similar localization precisions across each dimension retaining close to the CRLB[32]. However, their CNN localized 2 to 4.5x faster than 2D-Gaussian localization[32]. The results of the use of CNNs in these studies is a promising indication that CNNs can be effective to detect and localize motion-blurred emitters with a high precision.

In this thesis, we investigated the application of CNNs for both the detection and localization of motion-blurred emitters that have limited movement in the *xy*-plane (2D diffusion). The research questions can therefore be described as follows:

1. Can convolutional neural networks perform better in terms of localization precision compared to the current published localization methods?
2. Can convolutional neural networks perform better in terms of detecting emitters under low signal-to-noise ratio circumstances compared to the current published methods?

To answer the research questions, we simulated a dataset containing images of motion-blurred emitters that we subsequently used to train and validate CNNs. To ensure reliable results for the performance of the CNNs, we tuned the simulation software so that the resulting dataset resembled a non-simulated 2D SPT dataset obtained from previous research, frequently referred to in this thesis as the "non-simulated dataset"[33]. Subsequently, we tuned the SNR, diffusion constants and background noise to the non-simulated dataset. Additionally, we recorded dark images and noise with constant light intensities emitted on the camera to obtain parameters for a camera noise model. Thereafter, we validated the camera model. Eventually, we simulated a dataset containing motion-blurred emitters and a dataset containing images of only noise. Afterwards, we used the simulated datasets to train a classification (detection) CNN and a regression (localization) CNN, to be able to detect and localize motion-blurred emitters, respectively. Subsequently, we simulated four dataset each containing motion-blurred particles with a different combination of emission intensity and diffusion constant, to be able to test the performance of the CNNs in different situations. Thereafter, we compared the performance of the detection and localization CNN with published software.

The structure of this thesis is as follows. First, we introduce a theoretical background for SPT, simulations of motion-blurred emitters, and localization methods in Chapter 2. Second, Chapter 3 describes the methods used to obtain various parameters to tune our simulations with a non-simulated 2D diffusion experiment experiment[33]. This includes an experiment performed for the acquisition of dark and white light images, to be used to model camera noise. Third, in Chapter 4 we discuss the results about the validity of the simulations and the performance of the detection and localization CNNs compared to other methods published in the literature. Furthermore, a conclusion and outlook of this thesis is given in Chapters 5 and 6, respectively.

2. Theoretical Framework

In the following Chapter we discuss necessary theory for understanding this thesis. First, we cover a more in-depth explanation of fluorescent microscopy in Section 2.1. Subsequently, we discuss multiple topics that are needed to understand the backbone of the simulations in Chapters 2.2-2.4. That is, an in depth discussion of the PSF in Section 2.2, the diffusion in question for our motion-blurred emitters, i.e. self-diffusion, in Section 2.3, and lastly the origin of background and camera noise disrupting the fluorescent microscopy movies in Section 2.4. Additionally, a stochastic model is discussed for simulating camera noise in Section 2.4. Lastly, localization methods are discussed, with CNNs covered in Section 2.5, and published benchmarking localization methods in Section 2.6.

2.1 Fluorescence Microscopy

Contrast is needed to perceive details of objects contained in images. A method to create contrast is the use of fluorescence. For this, the use of fluorescent emitters are required, which are molecules or quantum dots that absorb light in a specific wavelength range and subsequently release light lower in energy. A well described example that compares the use of emitters in fluorescent microscopy is given in the book by Ulrich Kubitscheck[34]: a firefly in daylight is difficult to observe, but in the evening when a firefly emits light with a darker background, it becomes easier to spot due to an increase of contrast. Similarly, contrast is made when an emitter is imaged and stands out against a black background in fluorescent microscopy. However, there is one major difference between the example of a firefly and emitters in the microscopy technique: an emitter has to be excited before it emits light (of a lower energy). Therefore, the microscope has to be built in a way that the emitted light can be efficiently separated from the excitation light, resulting in a remaining contrast when the emission signal hits the camera. Consequently, a fluorescent microscope comprises multiple different parts to allow for the high contrast to remain.[13,34]

Figure 2 describes the components of a fluorescent microscope, starting with an

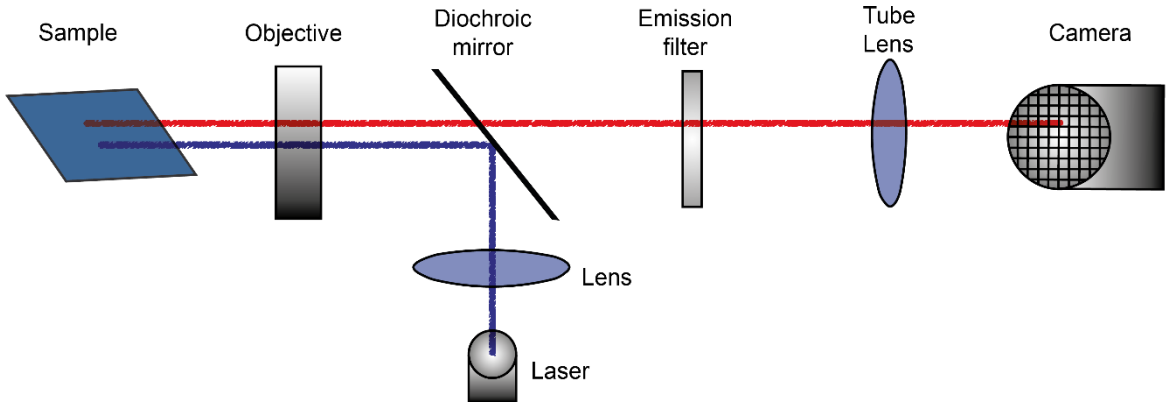


Figure 2: *Schematic representation of a fluorescence microscopy setup. A laser is focused via a dichroic mirror on the back focal plane of the microscope, resulting in a parallel beam illuminating a wide range of the sample throughout its entire depth. Consequently, emitters in the sample emit light in all directions, of which the light captured by the objective lens reaches the dichroic mirror. Subsequently, the dichroic transmits the longer wavelength emission light, after which the light is filtered in the emission filter, and is lastly focussed on an camera using a tube lens to produce an image/movie.*[13,34]

excitation source. The excitation source used in SPT is often a laser since it results in a fluorescence emission signal that can be detected with high sensitivity[9,11]. Subsequently, the laser is focused via a dichroic mirror on the back focal plane of the objective of the microscope, resulting in a parallel beam of monochromatic light hitting a wide region of the sample throughout its entire depth (widefield microscopy). Consequently, multiple emitters in the sample are excited towards higher states and release light in all directions of lower energies when relaxed back to lower states. The light collected by the objective, which consists of both emission and initial excitation light, is subsequently emitted back on the dichroic mirror. A dichroic mirror has the property of reflecting light of shorter wavelengths, i.e. in the range of excitation wavelengths, but transmitting light of longer wavelengths, i.e. in the range of emission wavelengths. Consequently, the emission light is transmitted through the mirror whilst the shorter wavelength excitation light is reflected. Thereafter, the emission light reaches the emission filter, where solely a desired band of wavelengths is transmitted as a filter, whilst light with wavelengths outside the lower and upper limit of the filter are reflected back to the source. Lastly, the light is focussed using a tube lens on the camera, i.e. for SPT often a highly sensitive camera such an electron multiplication charge-coupled device (EMCCD) camera, where the light is processed towards an image/movie.[13,34]

A limitation of using fluorophores in fluorescence microscopy is photobleaching, where emitters are chemically altered and as a consequence losing their fluorescent property, which in SPT hinders tracking of emitters for longer periods of time[11]. Alternatively, quantum dots

are commonly used in SPT[11], which have the fluorescent property and are more photostable[9,35].

2.2 Diffraction and the Point Spread Function

Diffraction is often demonstrated with the single-slit experiment, where a coherent light wave, e.g. a laser, is emitted on a slit or pinhole (Figure 3a). Subsequently, the planar wavefront

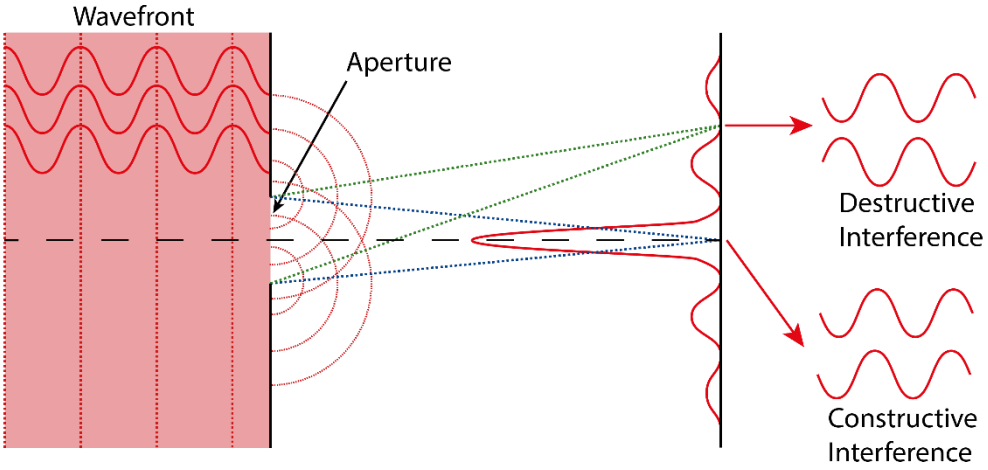


Figure 3: *A planar wavefront encounters a slit, where according to Huygens' principle new cylindrical wave fronts originate at each position in the slit. When wavelets travel equal distances, they arrive in phase and constructive interference causes a observed intensity. In contrast, wavelets arriving out of phase result in destructive interference and dimmer spots are observed. (Image based Fig. 12 of Vangindertael et al. [36])*

converts to a cylindrical wave front as it encounters the slit, as if the slit itself behaves like a point source. In fact, according to Huygens' principle, every infinitesimal point located in the slit behaves as a new point source causing interference between the individual originating wavefronts. Consequently, a characteristic diffraction pattern can be observed, where the centre of the pattern results in the highest intensity as all wavelets arrive in phase, resulting solely in constructive interference. When wavelets travel different distances, it can result in destructive interference and a dimmer spot is observed.[34,36]

The single slit experiment can be extended to 2D, where a wave front encounters a pinhole (Figure 4). Subsequently, destructive and constructive interference result in an airy

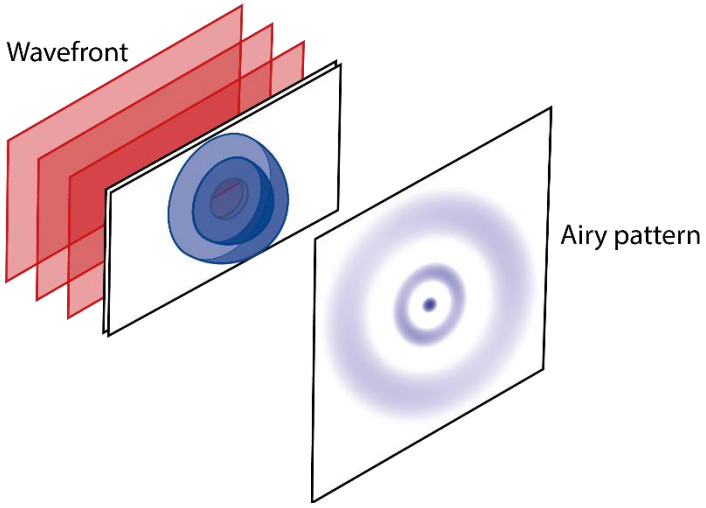


Figure 4: *A wavefront encounters a pinhole where according to Huygens' principle new spherical wavelets originate at every position in the pinhole. Consequently, destructive and constructive interference results in the airy pattern, consisting of a bright spot referred to as the airy disk surrounded by alternating dark and bright concentric rings.*

pattern, with the highest intensity observed in the middle resulting solely from constructive interference referred to as the airy disk, which is surrounded by alternating dark and bright concentric rings.

Let us now consider the case where a point source, e.g. an emitter, is imaged in focus of the objective lens utilizing a widefield fluorescence microscope (Figure 2). The point source emits a spherical wave of which a cone is captured by the objective lens, which transforms the light in a parallel wavefront. Thereafter, according to Huygens' principle, each point in the back vocal plane is the origin of new spherical wavelets. Consequently, when the wavelets are projected on the camera using a tube lens, the wavelets interfere exactly at the image plane, resulting in the same airy pattern as depicted in Figure 4. The airy pattern originated when imaging with a microscope is called the PSF, as it describes the microscopes' response to a point source emitter.[9,34,36] However, the observed emission signal when imaging a point source emitter depends on the axial position of the emitter, i.e. the PSF depends on axial position (Figure 5a). Therefore, the airy pattern only arises when a point source is imaged which lies exactly in the focus of the microscope (Figure 5c). In SPT, it is the centre of the airy disk which can be fitted optimally with a 2D Gaussian for localization[37]. However, emitters with axial positions above or below the focus result in different emission signals (Figure 5b), which can be fit with a 2D Gaussian less optimally.[34,36,37]

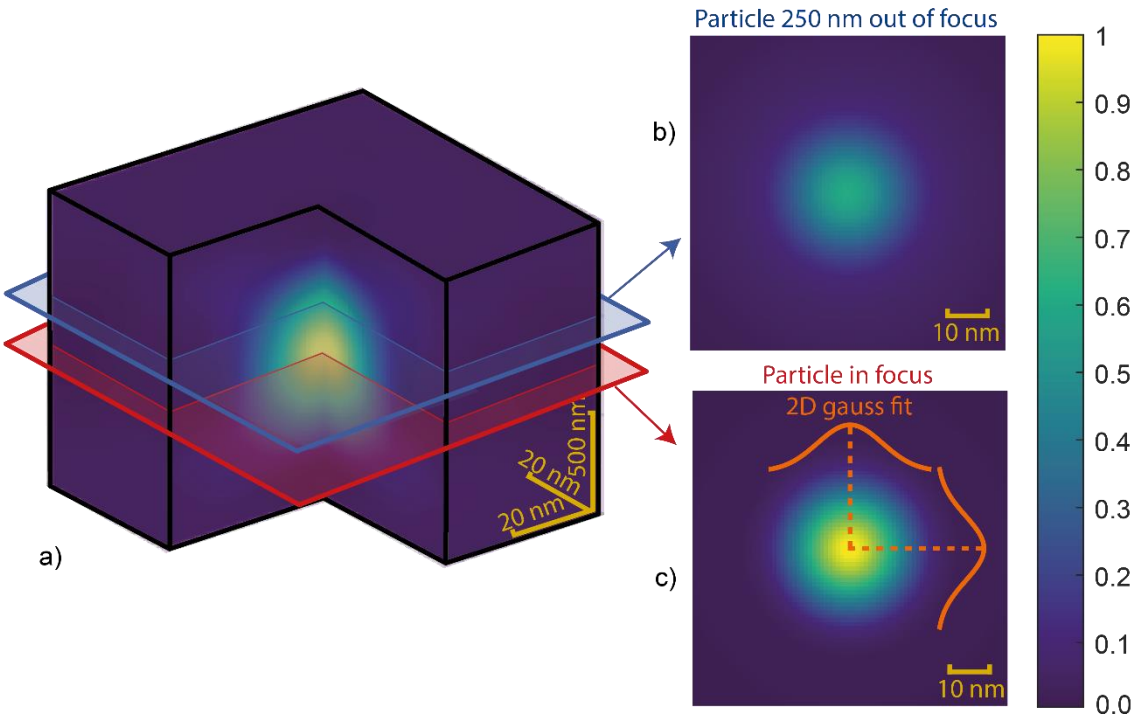


Figure 5: *a) When imaging a point source, the observed signal differs with axial position of the emitter. Therefore, the PSF can be described as a three-dimensional function, where each cross Section describes an imaged point source at different z-positions. b) An imaged emitter located 250 nm out of focus of the objective lens. c) An emitter located exactly in focus of the objective lens results in an airy pattern, which can be fit optimally with a 2D-Gaussian for localization in SPT.*

2.3 Self-Diffusion

Emitters diffusing in a sample undergo rapid random movements because of the interaction with the surrounding media, usually being water or solvent molecules, referred to as Brownian motion[38]. In the case of this thesis, we assume no external forces acting on the emitters, i.e. there is no chemical potential gradient for the emitters to diffuse along. This type of diffusion is referred to as self-diffusion, which can be observed for reactants diffusing in porous catalysts[39]. Consequently, the mean distance travelled by emitters self-diffusing equals zero, as the probabilities of diffusing in any direction are equal. Nevertheless, the emitter still travels around its original position. This ‘’wandering’’ is called a random walk of which Kärger et al. describes a perfect example in 1D (one-dimensional): consider an emitter that is located at an axis at position $x = 0$ and time $t = 0$[40]. Subsequently, for every timestep Δt the emitter has a probability of ½ to move with Δx and an equal probability to move the same distance in the opposite direction, i.e. $-\Delta x$. Consequently, a histogram containing locations of the emitter at different positions results in a normal distribution, which widens when t is increased.

Consequently, the amount of 'wandering' can be described with the standard deviation of the position histograms, or equally by calculating the root mean square displacement as

$$\langle r^2(t) \rangle = 2kDt \tag{1}$$

with D the diffusion constant and k the number of dimensions.[40,41]

Figure 6.a and Figure 6.b depict a self-diffusing emitter in 2D for a relatively short and

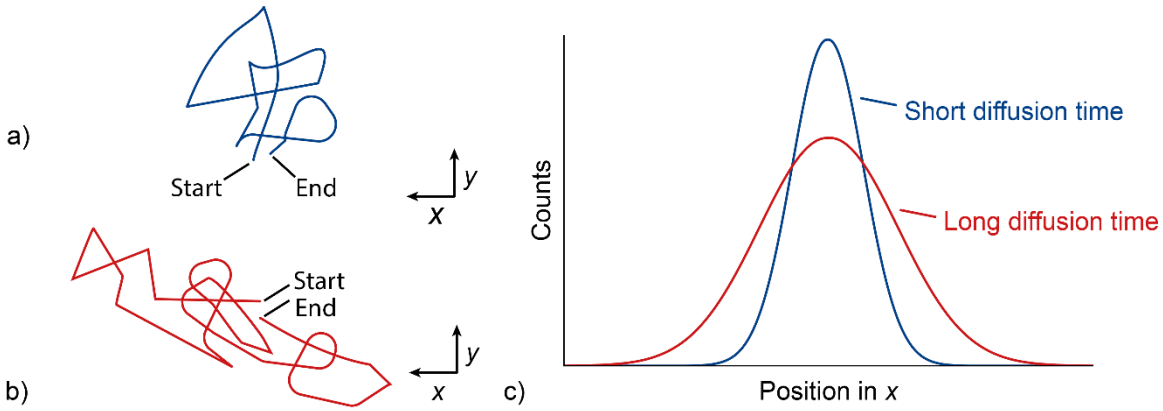


Figure 6: *a) The diffusion path of a self-diffusing emitter in 2D with a relatively short diffusion time. The squiggly line is the result of the instantaneous random movements because of Brownian motion. b) The diffusion path of a self-diffusing emitter in 2D with a relatively long diffusion time. c) Histograms of the positions with respect to* x *for the emitters in a) and b). The histograms are normally distributed and widen for longer diffusion times.*

long diffusion time, respectively. Considering the x position, the emitter in Figure 6.b has wandered a larger region than the emitter in Figure 6.a, which is reflected by a wider histogram in .c. Notably, the mean distance travelled in x for both cases is ≈ 0, as the movements in x are random.

2.4 Noise and Localization Error

Additional photons landing on the camera which do not originate from the desired emitters that we try to image are considered background noise. There are two sources that contribute to background noise: scattered photons and auto-fluorescence, of which auto-fluorescence is the process of natural emission of light by e.g. biological compounds and glass[42,43]. Consequently, every pixel of the resulting image/movie turns out brighter as the additional photons land on the detector evenly across its surface.[44,45]

Multiple processes contribute to camera noise, as an EMCCD camera consists of several components, with an overview of the components and the processes depicted in Figure 7. Frame acquisition starts with opening of the camera shutter, allowing photons originating from the

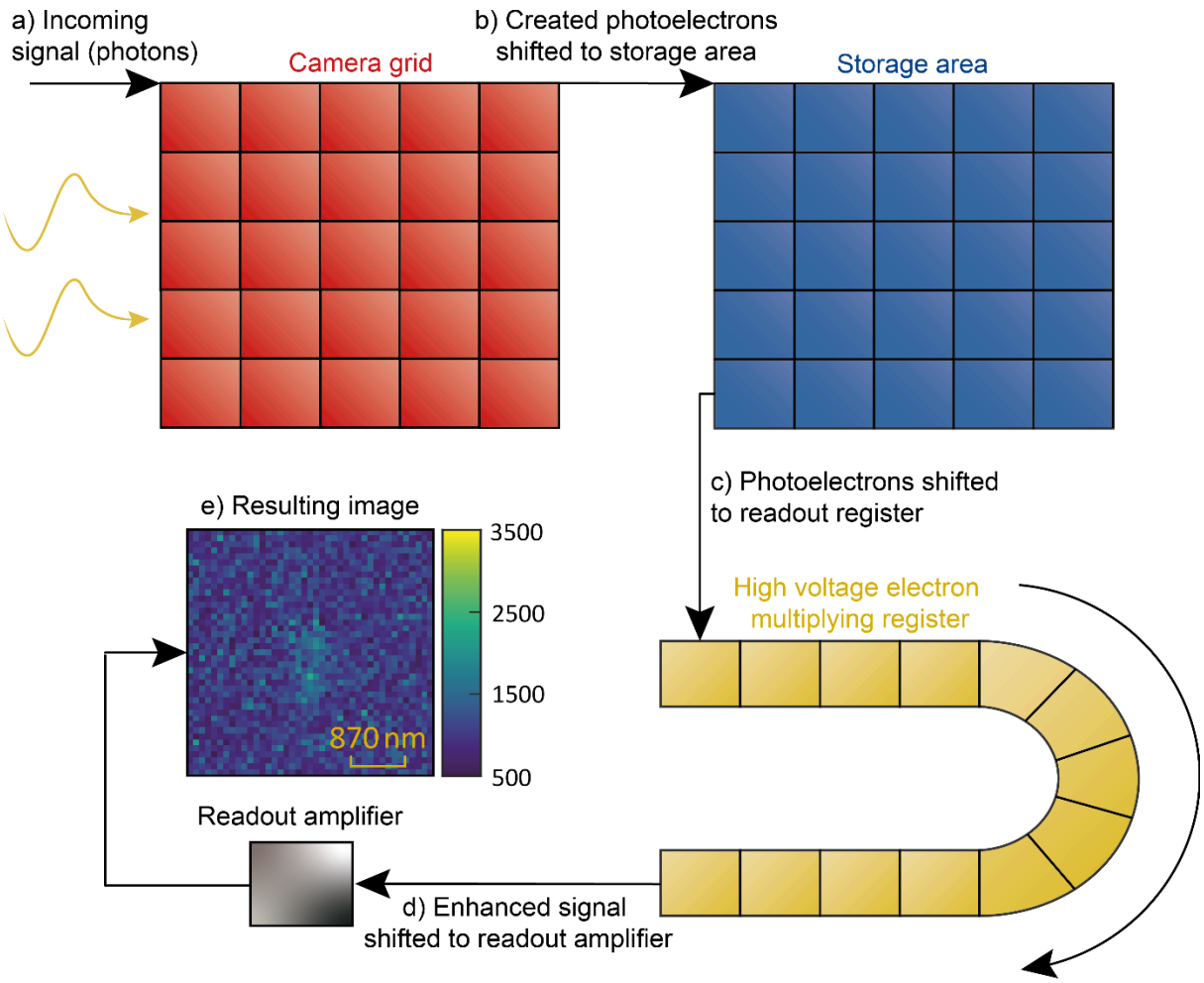


Figure 7: *Sketch of an EMCCD camera. a) Photons originating either from emitters in the sample or background noise land on the camera grid. Thereafter, upon collision of photons with the camera grid photoelectrons might be created depending on the quantum efficiency, and are subsequently stored in each pixel of the grid via storage capacitors. b) After frame acquisition the photoelectrons are shifted towards the storage area, which allows for the subsequent frame to be acquired whilst the storage grid is read out. c) Subsequently, photoelectrons are shifted towards the electron multiplying register where the signal is enhanced by impact ionization. d) Lastly, the electrons are converted to a digital value in the readout amplifier, e) resulting in an image. Adapted from Harpsøe et al.*[45].

sample or from background noise to land on the camera grid (Figure 7a). Subsequently, collision of the photons with the camera grid might cause the creation of a photoelectron, of which the probability is determined by the quantum efficiency of the camera. The quantum efficiency is typically 90% in EMCCD cameras[44]. Thereafter, the photoelectrons are stored in the pixels of the camera grid utilizing storage capacitors. Eventually, when the frametime has elapsed the shutter of the camera closes, and subsequently the photoelectrons are shifted to the storage grid by applying an electric field (Figure 7b). During the shifting of photoelectrons collisions with the lattice can result in the creation of extra electrons (impact ionization), of which the extra created electrons are referred to as spurious charge. As the extra electrons are

processed towards a signal in the final image not originating from the emitters, spurious charge is considered a source of camera noise. Thereafter, once the photoelectrons are fully shifted towards the storage grid, the subsequent frame acquisition can start whilst simultaneously the storage grid is read out. The noise created by spurious charge together with the quantum efficiency can stochastically be described as the Poisson distribution

$$P(n_{ie}; iq + c) \tag{2}$$

where i is the number of photons arriving at a pixel, q the quantum efficiency, c the spurious charge parameter, and n_{ie} number of input electrons for the next component of the camera: the electron multiplying register. Subsequently, row by row the photoelectrons are shifted from the storage grid to the electron multiplying register. In the electron multiplying register the electrons from each pixel are multiplied by deliberately carrying out impact ionization, which is performed by applying high voltages to the field causing acceleration of the electrons and therefore high velocity collisions with the lattice. Consequently, the multiplied photoelectrons enhance the SNR in the final image/movie, which is a crucial concept in SPT due to limited photon statistics, frametime and motion blur[12]. The electron multiplication process can stochastically be described as the gamma distribution

$$\gamma(n_{oe}; n_{ie}, g) \tag{3}$$

with g the electron multiplication gain (EM gain) parameter which determines the magnitude of electron multiplication, and n_{oe} the output electrons from multiplying register. Thereafter, the electrons are converted to pixel values (image counts) in the readout amplifier. Parameter f determines the factor between the conversion of electrons to image counts as $f = n_{oe} / n_{ic}$, with n_{ic} the number of image counts. Additionally, a random error is introduced when the output electrons of the electron multiplying register are processed towards image counts, called the readout noise, which is the most dominating form of noise in an EMCCD camera[45]. Consequently, signal of emitters are only visible in the resulting images when the EM gain enhances the signal more than the readout noise can distort it, emphasizing the importance of using an EMCCD camera. The readout noise is normally distributed as

$$N(fn_{ic}; n_{oe}, r) \tag{4}$$

with r the readout noise parameter. As the readout noise is normally distributed, the errors can be negative. Consequently, to avoid negative image counts, a camera bias c_b is added.[44,45]

The stochastic equations (equations (2)-(4)) describing each part of the camera can be combined to one complete stochastic equation describing the probability distribution of the complete process in the EMCCD camera. For this, a matrix multiplication of equation (2) with (3) is performed of which the result is convoluted with equation (4), resulting in the probability distribution function

$$p(n_{ic}; i, q, c, g, r, f) = ((P(iq + c) \circ G(g)) * N(r))(fn_{ic}) \quad (5)$$

Consequently, equation (5) can be used to model camera noise in our simulations.[44,45]

Figure 8 visualizes the effects of background and camera noise in 1D when an emitter

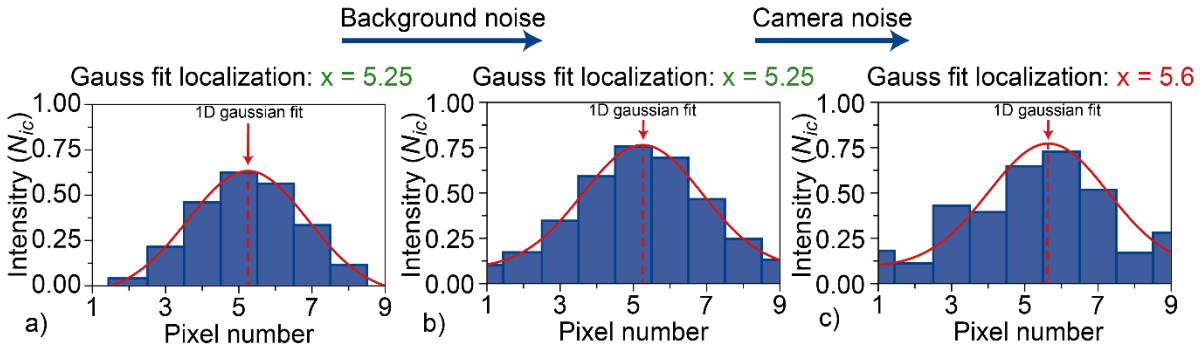


Figure 8: *a) One-dimensional example of the resulting signal of a emitter that is located at x = 5.25. Each bin in the histogram corresponds to a pixel of the one-dimensional image. The emitter location can be obtained by fitting the histogram with a one-dimensional Gaussian distribution. b) Due to background noise, the counts of each bin is raised evenly due to additional photons hitting the camera grid. c) Camera noise disturbs the signal when the photoelectrons are processed into an image. Fitting the final signal with a one-dimensional Gaussian comes with a localization error.*

is imaged with an EMCCD camera located at position x = 5.25. In the unrealistic case of imaging without background or camera noise, a 1D version of the PSF is imaged (Figure 8) (because of diffraction, Section 2.2). However, in the unrealistic case of solely imaging with background noise, additional photons land on the camera distributed evenly and therefore n_{ic} is higher in each pixel (Figure 8b). Realistically, both background and camera noise disrupt the image, which results in n_{ic} being randomly higher or lower in each pixel compared to imaging without noise (Figure 8c). Depending on the SNR, the signal could fade under the camera noise, resulting in an undetectable emitter. However, in case of Figure 8c, the SNR is high enough for the emitter to be localized by fitting with a 1D-Gaussian, but the 1D Gaussian will not fit optimally because of the camera noise resulting in a localization error.

Figure 9 shows a histogram of localization errors for an arbitrary localization method

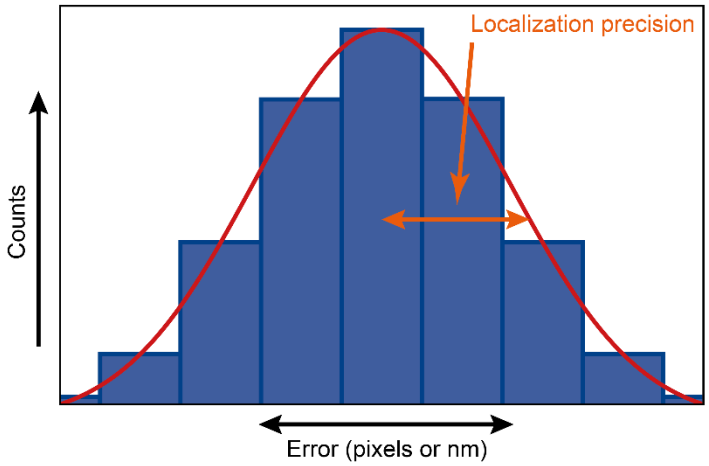


Figure 9: A histogram containing errors of a localization method used when imaging several emitters under the same conditions. Consequently, the localization precision of a method is defined as the standard deviation of the error distribution, expressed in pixels or nanometres.

which was used to localize many images of the same emitter. Consequently, the histogram will be normally distributed. The standard deviation of the distribution is used to define the localization precision of a method, usually expressed in pixels or nanometres. This can be defined mathematically for e.g. the localization precision in *x* as

$$\sigma_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{p,i} - \bar{x}_p)^2} \tag{6}$$

with n the number of estimates, $\bar{x}_p$ the mean true position, and $x_{p,i}$ the localizations. Subsequently, the localization precision expresses the performance of a localization method and can be used to compare methods.[46,47]

2.5 Convolutional Neural Networks

CNNs somewhat mimic the human brain in image recognition by finding features or patterns within an image, with features being e.g., edges, corners, squares etc. Consequently, the observed features contribute to the probability of output classes, which in case of classification CNNs can be binary, e.g., an emitter is detected or there is not, or in case of regression could be scalars, e.g., the *x* and *y* coordinates of the location of an emitter. A CNN consist of several layers performing mathematical operations on the input images, and the images are subsequently shifted through all these layers one by one, called forward propagation.

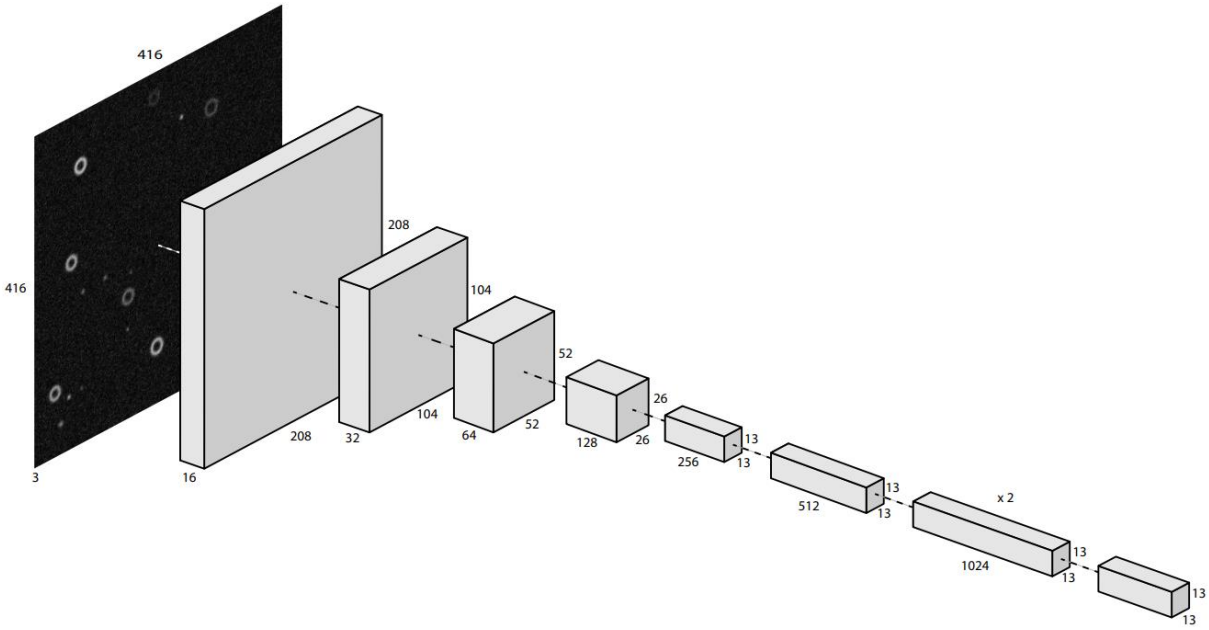


Figure 10: *An example of a CNN architecture. Each block represents a convolutional layer of a CNN. Max pooling layers decrease the blocks in* xy*-dimesions, for down sampling. The depth of the blocks equal the number of filters used for convolution. Eventually, the highest pixel values originating from convolution operations of the original image contribute to the output classes (output classes are not shown in the image). (Adapted from Fränzl et al.*[51])

The design in number of layers and order are referred to as the architecture of the CNN (Figure 10). Convolutional layers are most important in CNNs, which are used to enhance features.[21,48,49]

The mathematical operation of convolution in a CNN is depicted in Figure 11, where a filter is shifted over the original image performing multiplications of the superimposed values

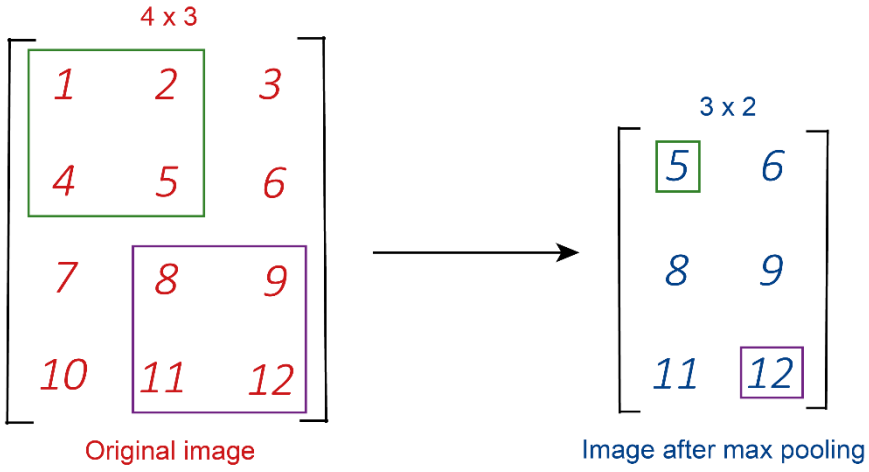


Figure 11: *A convolution is performed by shifting a filter over the original image. Each position where the kernel is overlapped results in a convolution that produces a new single value. The row and column position of the filter in the original image produces the new value in the feature map at the same row and column positions, as depicted with the green and purple squares.*

of the image and filter. When the values of the filter resembles those of the original image when superimposed, a high pixel value is produced in the convoluted image. For example, Figure 12

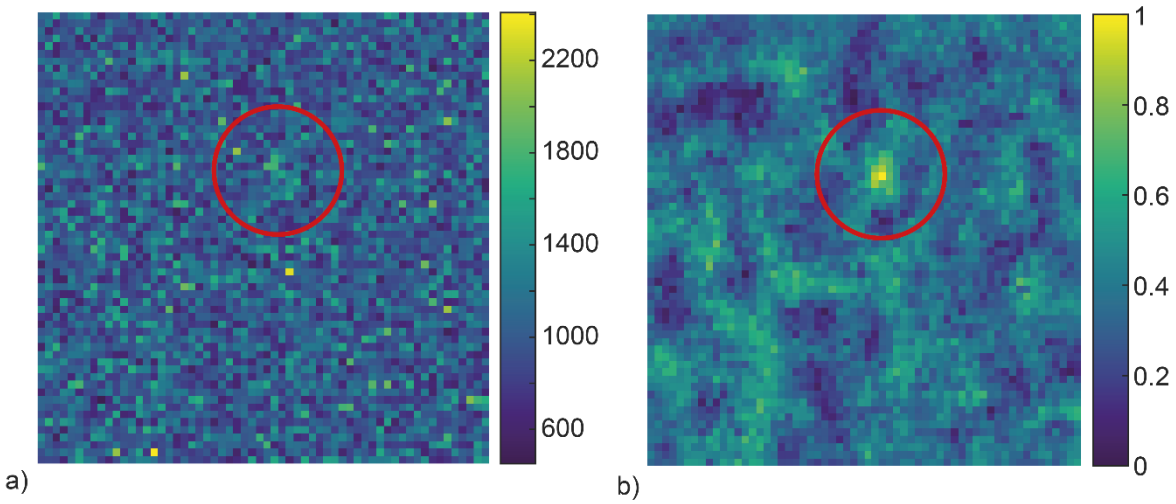


Figure 12: *a) An input image containing an emitter. b) The resulting feature map after convolution of the filter with the original image. High pixel values are produced where the filter resembled the image, i.e. the filter resembled a feature of a motion-blurred emitter and consequently the emitter itself is enhanced after convolution.*

shows a convolution operation of a filter resembling a motion-blurred emitter with an image containing an emitter, and as a consequence a high pixel value is produced in the convoluted image (Figure 13b). The convoluted image is referred to as a feature map, as all high pixel values in the feature map contain information about filter resemblance with features. Subsequently, an activation function e.g. ReLu is applied to the feature maps to introduce non-linearities in the CNN, crucial for the approximation of *any* function (Figure 13). Essentially, the ReLu activation function sets all negative values in the feature map to zero.[21,48,49]

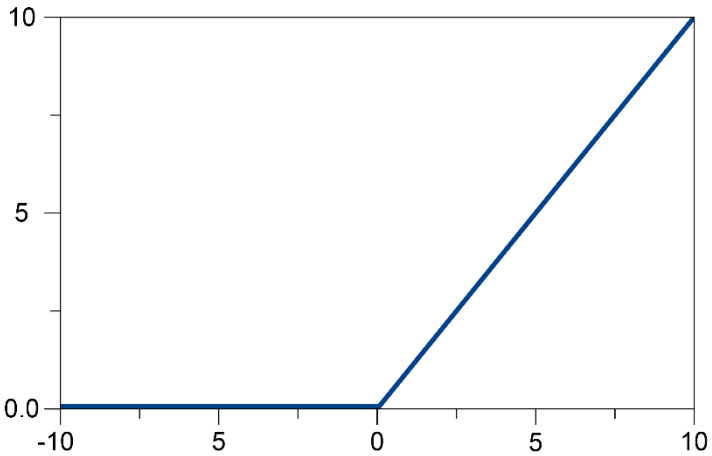


Figure 13: *The ReLu activation function. Each node containing a negative value is set to 0, while every node containing a positive value is kept the same.*

In max-pooling a new smaller image is created by selecting only the highest pixel values in certain regions of the original image, consequently down sampling the images (Figure 14).

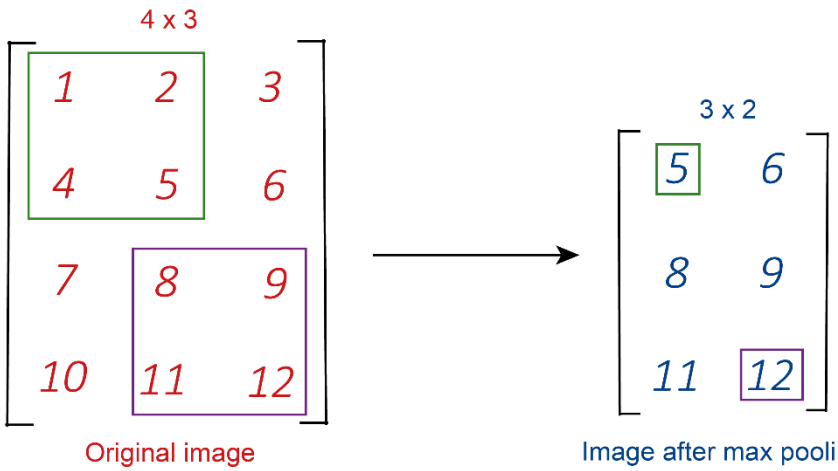


Figure 14: *Max pooling proceeds by sliding a filter over the original image outputting only the highest pixel values, i.e. the filter overlapped as the green square produces a value of five in the max-pooled image. A stride of one is used to down sample the image, ensuring the change in dimension from 4 x 3 to 3 x 2.*

As a result, the high pixels values remain in the newly created images, which are the pixels that contain the filter resemblance information. Furthermore, down sampling of the images can be accelerated by using a higher stride. Stride determines the amount of movement during shifting of the filter, e.g. a stride of one repeatedly shifts the filter by one column or row at a time, and a stride of two shifts the filter two columns or rows at a time. Therefore, a higher stride results in a lower amount of pixels selected for the new image. [21,48,49]

The several layers in a CNN (Figure 10) consist mostly of consecutive convolution into max-pooling layers. The deeper layers in the CNN are necessary to be able to detect more complicated features, e.g. motion-blurred emitters. Let us consider the architecture of the CNN in Figure 10. The input of the CNN is 416x416x3, as the image containing emitters contained three different colour channels (RGB). Subsequently, 16 different filters are used for convolution. However, the convolution of a single filter is performed across the entire depth of the input block, and results in a single 2D feature map. Consequently, 16 filters result in 16 feature maps. Subsequently, the 16 feature maps are down sampled in *xy*-dimensions from 416x416 to 208x208. Thereafter, 32 filters are convolved across the entire depth of the previous block containing the 16 feature maps, resulting in a new block of feature maps of dimensions 208x208x32. Subsequently, the block is max-pooled again etc. Essentially, since a filter convolved over the entire depth of a feature map block, the resulting feature maps enhances more sophisticated features that are combinations of the previous features found. [21,48,49]

After a sequence of convolutional and max-pooling layers, the remaining block of abstract features maps is flattened into a one-dimensional vector. Subsequently, each value in

the vector is connected to the different output classes. The full connectivity between these two layers is called a fully connected layer. [21,48,49]

Neurons in a CNN refer to all connections that have a weighted input from a previous layer to an output for the next layer, e.g. a convolution operation with a certain filter or all the connections in a fully connected layer. In addition, these connections contain a bias. A bias is added to allow for more freedom in function approximation. For example, after convolution the outputs are transformed by the activation function (Figure 12), and by adding a bias afterwards essentially the activation function can be shifted to the left and right. Subsequently, during training images with labels are inputted, i.e. ground truths, and the weights and biases are iteratively changed until all neurons contributing to the correct output classes are activated, i.e. the weights are turned up. For example, the weights of filters resembling motion-blurred emitters are activated, as are the connections with the correct output class in the fully connected layers. Consequently, when inputting an image containing a motion-blurred emitter, the correct output class will be activated.[21,48,49]

The loss function calculates the total error of a CNN. Therefore, the derivative with the loss function with respect to the weights and biases can be used to change the values so that the total loss decreases, called gradient-descent (). For a regression CNN the root mean square error (RMSE)

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{7}$$

can be used as the loss function. with $\hat{y}_i$ e.g. the predicted coordinates of all emitters in the batch (batch explained further in this Section), y_i the ground truth coordinates of all emitters, n the number of emitters in the batch. $\hat{y}_i - y_i$ results in the error of the prediction of each coordinate, and the value is squared to prevent negative values. The weights and biases are changed in direction of the end to the beginning of the architecture of a CNN, called backwards propagation, as neurons further in the network depend on the neurons in the layers before them. [21,48,49]

During training of a CNN, thousands of images have to be inputted to make sure a function is approximated that subsequently predict correctly when new images are inputted. These set of images are called the training set. It is computationally expensive to input the

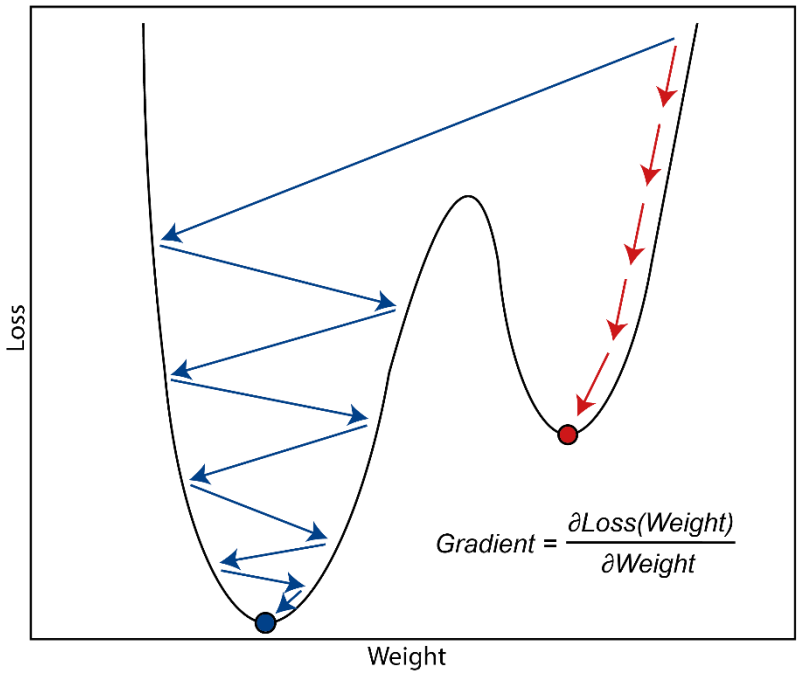


Figure 15 *When the steps in gradient descent are small and kept of the same magnitude, one can easily end up in a local minimum as depicted with the red arrows. Consequently, the CNN will not function optimally. The global minimum can be found when the learning rate starts off with high values and is slowly decreased overtime, as depicted with the blue arrows.*

whole training set and subsequently perform backward propagation repeatedly. Instead, small batches of images are inputted after which backwards propagation takes place, of which the size is called the batch size. The process of forward propagation of the batch and subsequently performing backwards propagation is called an iteration. An epoch is elapsed when all training images are inputted to the CNN once, which happens multiple times during training.[21,48,49]

The loss function can be viewed as a mountain valley, containing peaks and valleys analogue of total error. It is important to end up at the bottom of the global minimum of this mountain valley when performing backwards propagation, as only then the function is approximated optimally resulting in the lowest errors. Therefore, the learning rate is an important parameter during the training of CNNs. The learning rate should be set high to end up in the global minimum, and subsequently low to descent to the bottom of it (blue arrows in Figure 15). Otherwise, one could end up in a local minimum with a non-optimal approximation of the function (red arrows in Figure 15). For that, the parameter ‘’learning rate drop factor’’ determines the amount of epochs needed to be elapsed until the learning rate is changed, where it subsequently is multiplied with the parameter ‘’learning rate drop factor’’ to be lowered. [21,48,49]

CCNs are notorious for overfitting. This means that the weights in the CNN get tuned to the training data such that it does not perform well on new data. Feeding a trained CNN a set of new images to test how well it performs is called validation. The set of images used are subsequently called the validation set. Overfitting is the result of neurons becoming co-dependent on each other, which causes their weights to affect the optimization process of the weights of other neurons. A solution to this is the use of a dropout layer, which randomly selects a set of neurons of which the output is set to zero. Consequently, in each iteration a different selection of neurons determines the output, and backward propagation is performed based on this performance. As a result, the CNN is trained on a random selection of neurons in each iteration, subsequently making it impossible neurons to become co-dependant on each other. [21,48,49]

2.6 Benchmark Localization methods

The weighted centroid is equal to calculating the centre of mass but for images. For example, the centroid location of a 1D imaged emitter is performed by a multiplication of each pixel value with its pixel position, and a subsequent division of the summation of all pixel values. Consequently, the resulting weighted centroid location will lean towards regions in the 1D image containing the highest pixel values. The same can be performed for a 2D image, where the centroid position for x is defined as

$$C_x = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} x_i I_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{n} I_{i,j}} \tag{8}$$

with i the column indices, j the row indices, x_i the value of the column index of the i^{th} pixel, and $I_{i,j}$ the pixel value of the i,j^{th} pixel. Similarly, the centroid position for y, C_y, can be determined analogue to equation (8).

The emission signal can be fit with a 2D-Gaussian model to localize coordinates. Often, least square estimation (LSE) or maximum likelihood estimation (MLE) is used to iteratively fit the model to the emission signal by adapting the model parameters[47]. Additionally, the 2D-Gaussian model can be adapted to an ellipsoidal 2D-Gaussian model, which has the freedom in extending the fit in x and y dimensions independently. Moreover, an adaption can be made to a rotational ellipsoidal 2D-Gaussian, which additionally can be rotated with respect to the x and y axes. Mathematical descriptions of all three models are given in Appendix A1.

3. Methods/Experimental

In the following Chapter we describe the methods used to be able to answer our research questions. First, we discuss the simulation software in Section 3.1. Second, we describe experiments in Section 3.2 performed to obtain the camera model parameters. Third, Section 3.3 describes the methods related to the use of CNNs. Lastly, we discuss benchmarking published localization software in Section 3.4. We prepared a table containing all parameters used in the methods, i.e. used in the simulation software and the benchmarking published detection and localization software (Appendix C).

3.1 Simulation Software

Grey scale motion-blurred emitters were simulated in MATLAB using software that consisted of four distinct steps: diffusion path calculation, PSF convolution, background noise addition, and camera noise addition (Figure 16). First, a 2D random walk was calculated by computing random steps in *x* and *y* directions using a probability distribution with standard deviation

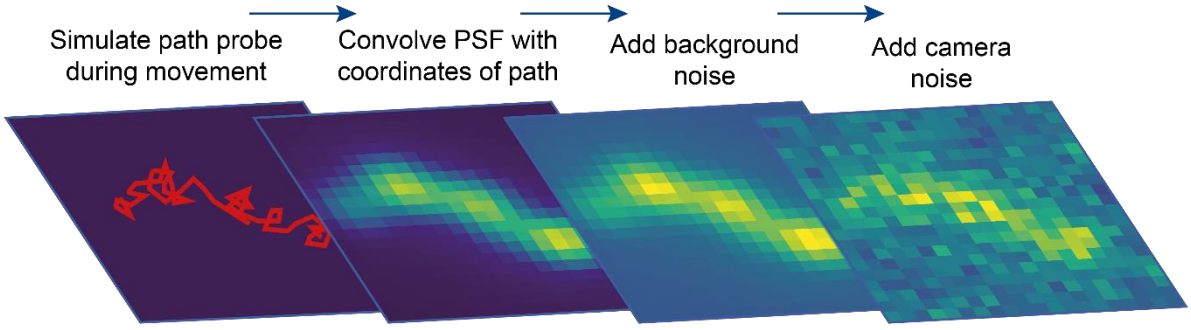


Figure 16: *Sketch of how the simulation of a single frame containing a motion-blurred emitter was performed. a) Random walk of emitter is calculated. b) An equidistant set of points on the diffusion path were convolved with the PSF, by dividing the total amount of emitted photons per frame over the set of points. c) Background noise was added to the image by adding a value to every pixel in the image. d) Camera noise is added by a using a camera model from Hirsch et al.[44].*

$\sqrt{2Dt}$[50]. The diffusion constants used were the average diffusion constants observed in the non-simulated dataset[33]. The timestep used depended on the amount of oversampling of the diffusion path, which was chosen to be 30, and the total frametime, equalized to the non-simulated dataset was 0.03 s[33]. Therefore, each 30 subsequent positions of the random walk were calculated with a timestep of 0.001 s. A camera grid was simulated for the emitters to diffuse in with dimensions of 60×60 pixels, with the pixel size equalized to the non-simulated dataset of 86.67 nm[33]. Afterwards, a PSF was calculated utilizing vectorial diffraction theory based on Backer et al. (parameters App.C), where subsequently the PSF of emitters in focus

was used as we simulated 2D diffusion (Figure 5c)[28]. Thereafter, to convolve the PSFs with sub-pixel precision and match the dimension of the pixel grid, the PSFs were shifted and resized to be subsequently convolved with a number of photons located on the coordinates of each position in the random walk to simulate motion-blur (App. D). For this, trajectories were reconstructed in the non-simulated dataset to isolate stationary emitters[33]. Subsequently, the maximum pixel of each ROI containing a stationary emitter was used to create an intensity histogram. Afterwards, ROIs containing stationary ($D = 0\ m^2\ s^{-1}$) emitters were simulated and their average maximum pixel values were tuned to the intensity histogram by varying the amount of photons for PSF convolution. Thereafter, background noise was simulated by adding a constant value to each pixel (the background noise parameter b_n). For this, a region only containing noise in the non-simulated dataset was isolated. Subsequently, noise was simulated (with a working camera model, Section 3.2) and was tuned to the non-simulated dataset by optimizing the background value parameter using MLE. Lastly, a camera model was applied to the simulations of motion-blurred emitters tuned to the non-simulated dataset (Section 3.2).

3.2 Camera Model Parameter Acquisition

A stochastic camera model (Section 2.4) was used to model camera noise, of which the parameters were obtained by following the methods described in Hirsch et al.[44]. Three experiments are described in Hirsch et al.[44] utilizing the camera to be simulated. Therefore, the experiments were performed on the same camera as the non-simulated dataset, i.e. an iXon Ultra 888 camera. In each experiment 2000 frames of dimensions 512×512 were recorded with frametime 0.03 s, readout time 0.02675 s, CCD temperature of −70 °C and frame transfer turned on (faster frame acquisition rates). First, a recording was made with a closed shutter and EM gain of 300, resulting in dark images. Second, the setup in Figure 17 was used to record a movie with an EM gain of 300 where a set of reflective filters were employed between the camera grid and a lamp emitting a constant light intensity. Subsequently, this was repeated whilst interchanging the set of filters to have total numerical densities (ND) ranging from 4.5 to 7.5, with increments of 0.5 resulting in eight movies. For an ND filter with optical density d the optical power transmitted through the filter is determined by $I_T = I_0 10^{-d}$ with I_0 the incident intensity and I_T the transmitted intensity. Third, the previous experiment was repeated with the EM gain turned off using a set of filters with a total ND ranging from 2.0 to 5.0, with increments of 0.5, resulting in seven movies.

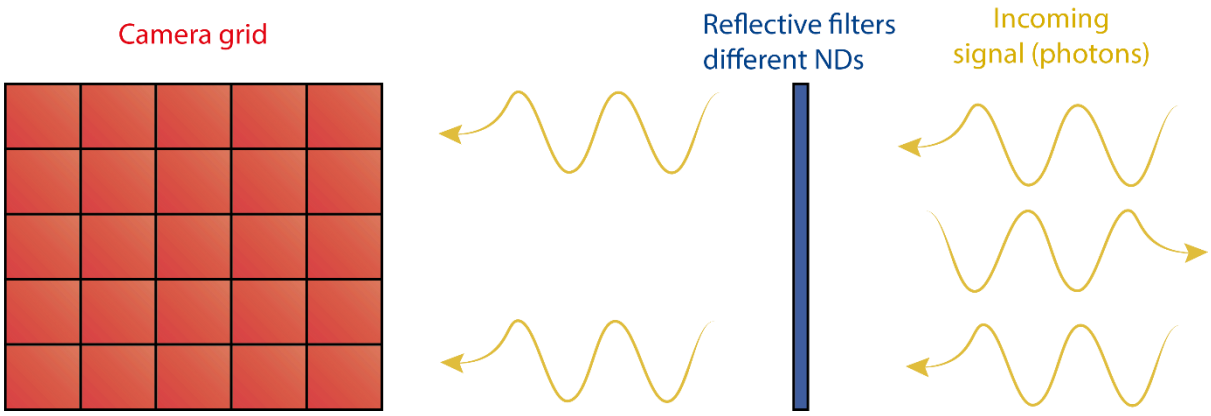


Figure 17: *Experimental setup for finding the f (ADU conversion) and g (EM gain) parameters for the camera model. A video is taken by shining light with a constant intensity on the EMCCD camera grid, which is performed multiple times for filters with a different numerical density blocking a different amount of light.*

3.3 Convolutional Neural Networks

A modified version of the CNN designed by Fränzl et al. was used as architecture for the detection and localization CNNs, i.e. the layer sequences remained the same but the *xy*-dimensions of the input images and convolution/max-pooling blocks were adjusted to be in line with the *xy*-dimensions of our simulated training and validation sets[51]. Additionally, a dropout layer was added to prevent overfitting. Furthermore, the output layers of the detection and localization CNNs differed as a classification and regression layer are needed as outputs, respectively. Lastly, the detection CNN contained two output neurons, i.e. binary output, and the localization CNN contained a single output neuron to estimate the *x* coordinate. Subsequently, input images could be transposed to determine the *y*-coordinate. The architecture of both CNNs are summarized in Table 1.

Table 1: *Architecture of the detection and localization CNN based on Fränzl et al.*[51].

Layer Type	*Num. Filters*	*Size/Stride*	*Output*
Input (60×60)	-	-	-
Dropout (50%)	-	-	-
Conv.	64	6 × 6/1	60 × 60 × 64
Maxpool	-	2 × 2/2	30 × 30 × 64
Conv.	128	3 × 3/1	30 × 30 × 128
Maxpool	-	2 × 2/2	15 × 15 × 128

Conv.	256	2 × 2/1	15 × 15 × 256
Maxpool	-	2 × 2/2	8 × 8 × 256
Conv.	512	2 × 2/1	8 × 8 × 512
Maxpool	-	2 × 2/1	8 × 8 × 512
Conv.	1024	3 × 3/1	8 × 8 × 1024
Conv.	1024	3 × 3/1	8 × 8 × 1024
Conv.	1024	1 × 1/1	8 × 8 × 1024
Fully connected	-	-	1 × 1 × 2
Classification/regression	-	-	-

During training the Adam optimizer was used for both CNNs which is an optimized algorithm to perform efficient gradient descent[52]. Moreover, a learning rate test was performed to determine the learning rate, ending learning rate, learning rate drop period and the learning rate factor[53]. A standard batch size of 64 was used and the number of epochs was calculated such that the CNNs could be trained in a manner of days. The training settings of the CNNs are summarized in Table 2.

Table 2: Training settings used for training the detection and localization CNN.

Training Option	*Detection CNN*	*Localization CNN*
Optimizer	Adam	Adam
Initial learn Rate	0.23e −3	0.32e −2
Learn rate drop factor	0.7	0.7
Learn rate drop Period	1	1
Number of Epochs	34	30
Mini batch size	64	64

3.4 Training and Validation Set Simulations

For the detection CNN a training set of 200.000 frames of dimensions 60×60 were simulated with a 1:1 ratio of frames containing a motion-blurred emitter and frames containing pure noise. For the localization CNN a training set was simulated of 900.000 frames of equal dimensions.

The diffusion constant for each emitter in the training sets ranged from 8e−13 to 4e−12 $m^2\ s^{-1}$ and the intensity from 0 to 312 photons/pixel/frame, which were randomly determined.

Four validation sets were simulated to validate the performance of the CNNs in different regions. All validation sets contained 125.000 frames with dimensions 60×60 containing a motion-blurred emitter where various combinations of diffusion constant and emitter intensity were made. Additionally, each validation set contained 125.000 frames of simulated noise of the same dimensions. The validation sets are numbered from 1 to 4, summarized in Table 1.

Table 1: *The four simulated validation sets each containing motion-blurred emitters with a different combination of diffusion constant and intensity.*

Validation set	*Diffusion constant ($m^2\ s^{-1}$)*	*Intensity (photons)*
1	8e−13	135
2	4e−12	135
3	8e−13	312
4	4e−12	312

The average position of the motion-blurred emitters in the frames were determined by calculating the average *x* coordinate of the 30 positions of the random walk, analogue to this the average *y* coordinate was determined. Consequently, these average positions were used as ground truth labels for training the localization CNN. Additionally, the simulations in both the training and validation sets were performed such that the average positions of the emitter could only be positioned in a 30×30 box located in the centre of the 60×60 frames, ensuring the emission signal to be always fully visible in the frames.

3.5 Detection of Motion-Blurred Emitters

The performance of the detection CNN was tested by comparing it to published benchmark detection software, i.e. DoM v1.2.2 by Katrukha et al. (App. A.2)[54]. First, DoM was utilized to detect emitters in the four validation sets (parameters of DoM in App. C). The threshold in DoM was set such that few false positives (false detections, FP) arose. For fair comparison with the CNN, FPs were only counted when DoM returned the detection of an emitter in a noise image. Additionally, a maximum of one FP per noise image was counted. Furthermore, localizations within a 7×7 box around the ground truth were considered true positives (correct detections, TP). Second, the trained detection CNN was used to detect emitters in the four

validation sets. Subsequently, the CNN returned probabilities for each frame, i.e. emitter and noise images, if there is an emitter located in the image or not. The probabilities were thresholded to obtain binary predictions for each frame. Thereafter, the threshold was tuned such that the number of FPs matched those of DoM, which subsequently allows for the comparison of TPs. Similarly, the threshold was tuned to match the TPs of both methods to compare the difference in FPs.

3.6 Localization of Motion-Blurred Emitters

Localization was performed on the frames where a TP detection was obtained. First, weighted centroid localization was already performed by DoM during the detection step (App. A.2). Subsequently, ROIs were made with sizes ranging from 5×5 to 17×17 for every TP detection, with incremental steps of two so that the width and length of the ROIs were always an uneven number. Thereafter, localization was performed on the ROIs by fitting the Gaussian models from eq. (12)–(14) with MLE and LSE utilizing Gputfit, published by Przybylski et al.[17]. Similarly, phasor fitting was performed for localization by using the script published in the supplementary information of Martens et al.[19] (App. A.3). Lastly, localization was performed using the original frames (60×60) of the TP detections utilizing the trained localization CNN. Subsequently, for every method the localization precision was calculated via eq. (6).

4. Results and Discussion

First, we discuss the non-simulated (nano-slit) experiment in Section 4.1 which was used to tune the diffusion constant and the SNR of the simulations. Second, we discuss in Section 4.2 the data processing of the dark and white light images to obtain the parameters needed to simulate our camera. Third, we report the performance of the detection CNN and compare it to published detection software. Lastly, we report the localization precisions of our localization CNN and various published localization software for when they were used to localize motion-blurred emitters of the validation sets.

4.1 Diffusion Constant and Emitter Intensity

Figure 18 shows a description of the non-simulated dataset containing a frame of the

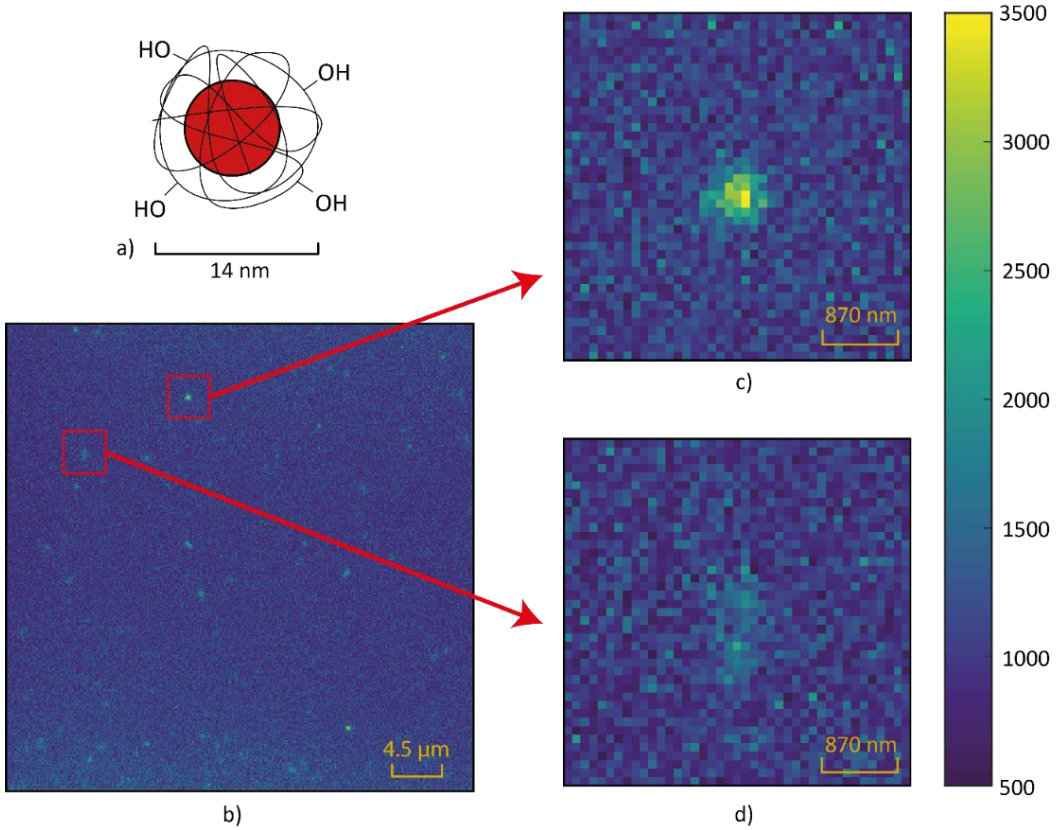


Figure 18: a) *A single frame of the SPT movie where quantum dots are diffusing in two dimensions. b) Stationary emitters are visible in the experiment. c) Diffusing emitters are visible in the experiment showing their motion-blur.*

recording[33]. The recording contains diffusing CdSe quantum dots covered in a layer of ZnS and a PEG 14 nm in diameter (Figure 18a). Their diffusion freedom is limited to 2D as they are confined in a nanoslit with a *z*-dimension of 150 nm. In the recording (Figure 18b) both stuck

and diffusing emitters can be obsered. The emission signal of the stuck emitters can be observed as non-deformed PSFs (Figure 18c), of which the position could be localized with nanometre precision[9,11,16,17]. However, the diffusing emitters are observed with motion-blur (Figure 18d). They observed in two different nanoslit experiments average diffusion constants of 4e−12 and 8e−13 m^2 s^{-1}[33]. Therefore, these are the diffusion constants used for the simulations to test the performance of CNNs.

Figure 19 shows the intensity histogram by isolating the stationary emitters observed in

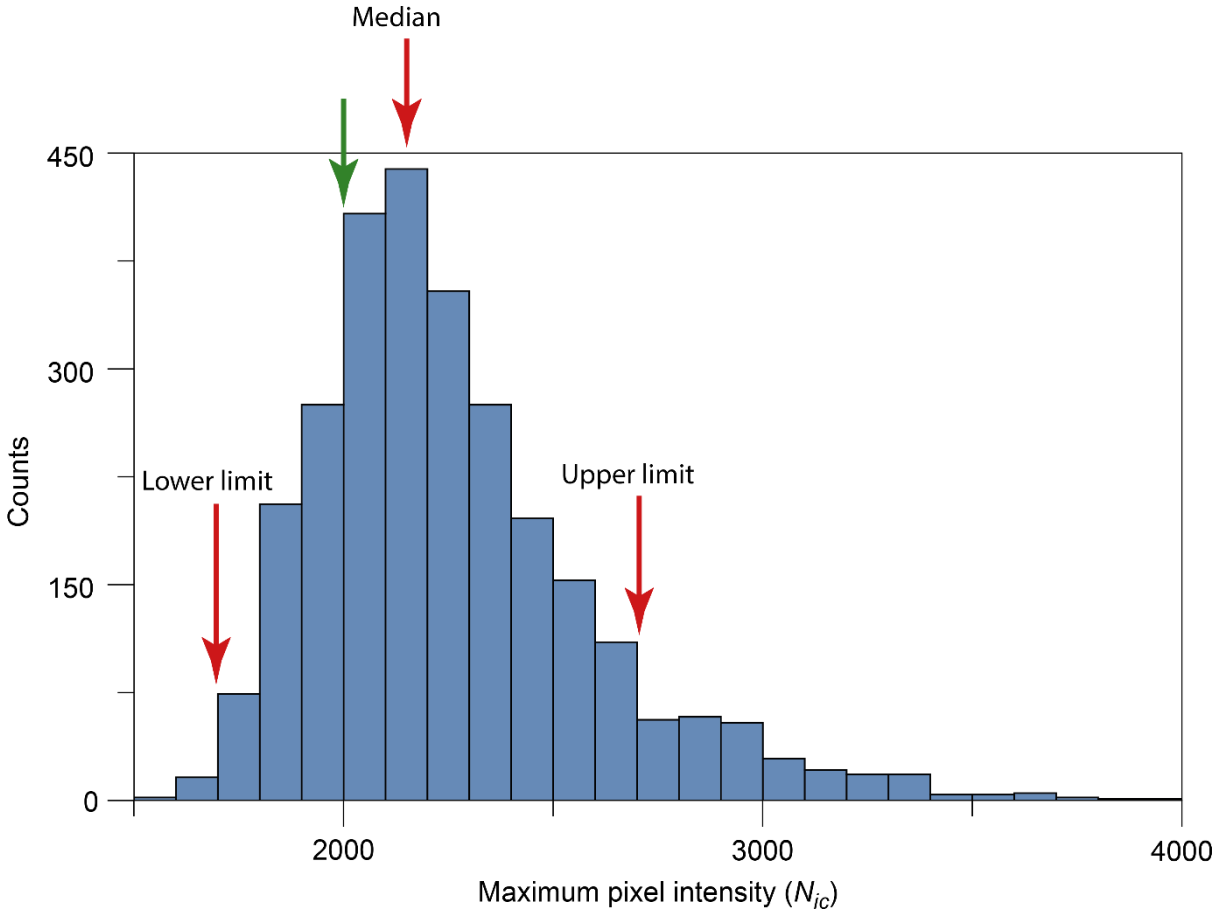


Figure 19: *A histogram of the brightness of the stationary emitters observed in the non-simulated dataset. In red the median and a self-defined lower and upper limit is indicated. The green arrow shows the new lower limit chosen to assure gradient descent during training of the detection CNN.*

the recording of Figure 19b. Subsequently, we tuned simulated stationary emitters to this histogram resulting in intensities of 5, 135 and 312 photons/emitter/frame to simulate equally bright emitters located at the lower limit, median and upper limit, respectively (Section 3.1). Therefore, we used these values in the training and validation set for the CNNs to test the performance of the CNN for detection and localization of the most common emitters (median), and to learn the performance in these extremes (lower to upper limit). We assume that the amount of photons emitted does change when emitter start moving, and therefore the amount

of photons emitted for diffusing emitters in the non-simulated dataset would emit photons at the same rate as our simulated diffusing emitters.

4.2 Camera Model

4.2.1 Parameter Acquisition

To approximate the parameters f and g we followed the methods reported in Hirsch et al[44]. Therefore, we plotted the mean image counts $\tilde{n}_{ic}$ of each pixel against the variance σ_{ic} (mean-variance test) for each of the recordings obtained from the experiments depicted in Figure 17 (Section 3.2). Consequently, Figure 20a shows the mean-variance test for the

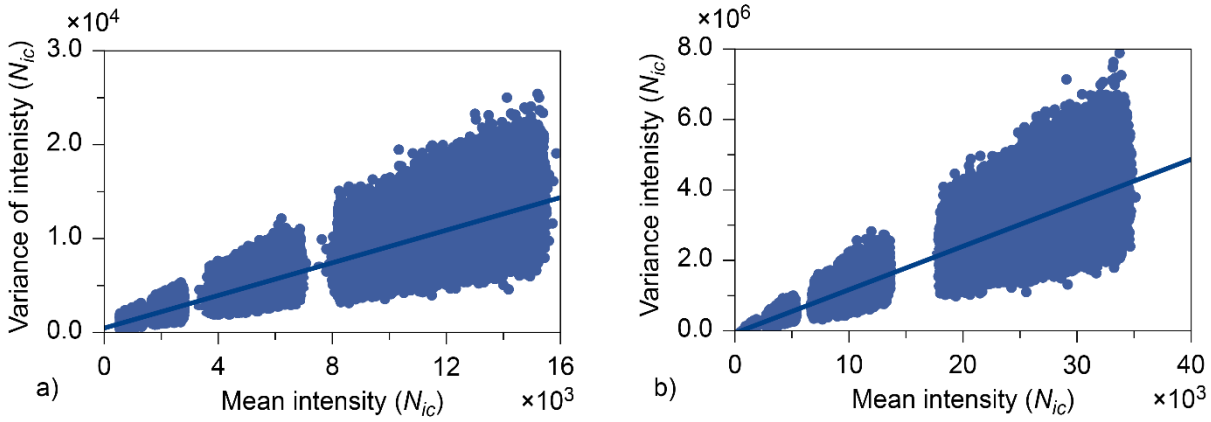


Figure 20: The mean-variance test of the experiment where the light intensity emitted on the camera N_{ic} was varied with a) the EM gain turned off and b) the EM gain turned on. A linear regression was fit to the experimental data.

experiment where the EM gain was turned off, and Figure 20b for the experiment with the EM gain of 300. As different transmission occurred in each recording, the resulting image counts differed resulting in the isolated regions of blue dots. Since we minimized the gain (Figure 20a) we can approximate eq. (5) as $p(n_{ic};i,q,f) \approx P(fn_{ic};iq)$, because the Poisson component dominates as no electron multiplication takes place. Subsequently, using this approximation and the definition of $f = n_{oe} / n_{ic}$, we know that the probability for n_{ic} is Poisson distributed with mean $\tilde{n}_{ic} = \tilde{n}_{oe} / f$ and variance $\sigma^2_{ic} = n_{oe} / f$, resulting in

$$\hat{f} = \frac{\tilde{n}_{ic}}{\sigma^2_{ic}} \tag{9}$$

for the approximation of f (circumflex stands for approximation). Subsequently, we used eq. (9) to derive an approximation for f by fitting a linear regression to Figure 20a, where the slope of the fit was equal to f (Table 3).

With high values for EM gain (Figure 20b) the Gamma component dominates in eq. (5). Subsequently, Hirsch et al. reports the approximation for *g* as

$$\hat{g} = f\ \frac{\sigma_{ic}^2}{2\tilde{n}_{ic}} \tag{10}$$

Subsequently, we used equation (10) to approximate *g* by fitting a linear regression to Figure 20b, of which the inverse slope was equal to *g* (Table 3). An assumption is made here that all spurious charge is created before the EM register to simplify the modelling, as a result *c* is not processed in the Gamma distribution of eq. (5). However, in reality spurious charge *is* created in the EM register and therefore affects to this approximation affects the models' performance[44].

To estimate parameters *c* and *r* we processed the recording of the dark images following the methods of Hirsch et al.[44]. First, we corrected the images following eq. (1)–(7) of Hirsch et al., to remove inhomogeneities from the images e.g. stripes of bright pixels[44]. This also included the removal (and determination) of c_b (Table 3). As no photons land on the camera grid, no photoelectrons are created. Therefore, the only created electrons contributing to pixel values in the recorded images originate from spurious charge. Eq. (5) can be modified to obtain the probability distribution function for dark images as

$$p(n_{ic}; q, c, g, r, f) = ((P(c) \circ G(g)) * N(r))(f n_{ic}) \tag{11}$$

by removing *i*. All parameters of eq. (11) are currently known, except for *c* and *r*. Therefore, we determined the remaining parameters by optimizing a fit of eq. (11) with a histogram of the dark images using log-likelihood fitting. Figure 21 shows the optimized fit of eq. (11) with the histogram of the dark images. First, as often zero electrons are created for certain pixels, the readout noise is the only component contributing to resulting pixel values of the dark images. Therefore, a normal distribution is expected with its peak located at zero. Consequently, the standard deviation of the peak can be used to describe the magnitude of the readout noise, and is therefore equal to parameter *r (*Table 3). To be able to fit eq. (11) to the histogram we added a bias of 1000 to all pixels, as negative values should be avoided due to the matrix operation, explaining the location of the peak. Second, when spurious charge is created, their signal is subsequently amplified in the EM register and results in higher image counts. Therefore, a '' tail'' is usually visible in the histogram of dark images attributed to by *c*[44]. However, the dark

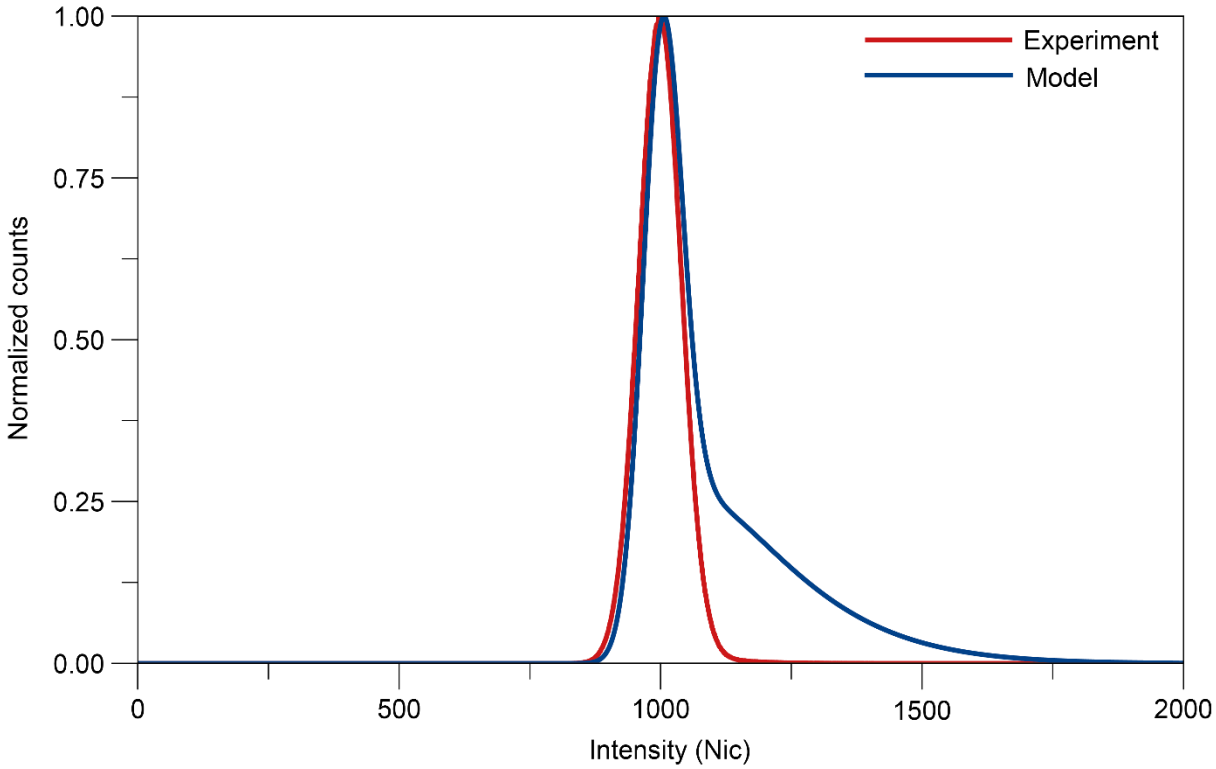


Figure 21: *Fit of eq. (11) with the dark images by optimizing parameters* c *and* r *using log-likelihood. The normal component showed an optimal fit. However, poor agreement was found for intensity values above 1100.*

images that we recorded did not contain a tail, and consequently resulted in a poor fit with the model for intensity values above 1100. Therefore, the value described in the manual of the camera for the spurious charge parameter *c* was used (Table 3)[55].

Table 3: *The resulting parameters obtained from data processing of the experiments, which can subsequently be used to model the camera noise of the camera used in the non-simulated dataset.*

Parameter	*Value*
f	1.1512
g	71.0067
r	37.3955
c	0.00025
b	488

4.2.2 Camera Model Validation

We fitted the camera model (eq. (5)) using the obtained parameters with the histograms of the noise recordings by optimizing *i* using MLE for the validation of our camera model. For this, the recordings from the experiment described in Figure 17 (Section 3.2) were used where the EM gain was turned on. Figure 22a–c shows the optimized fits of our camera model with the

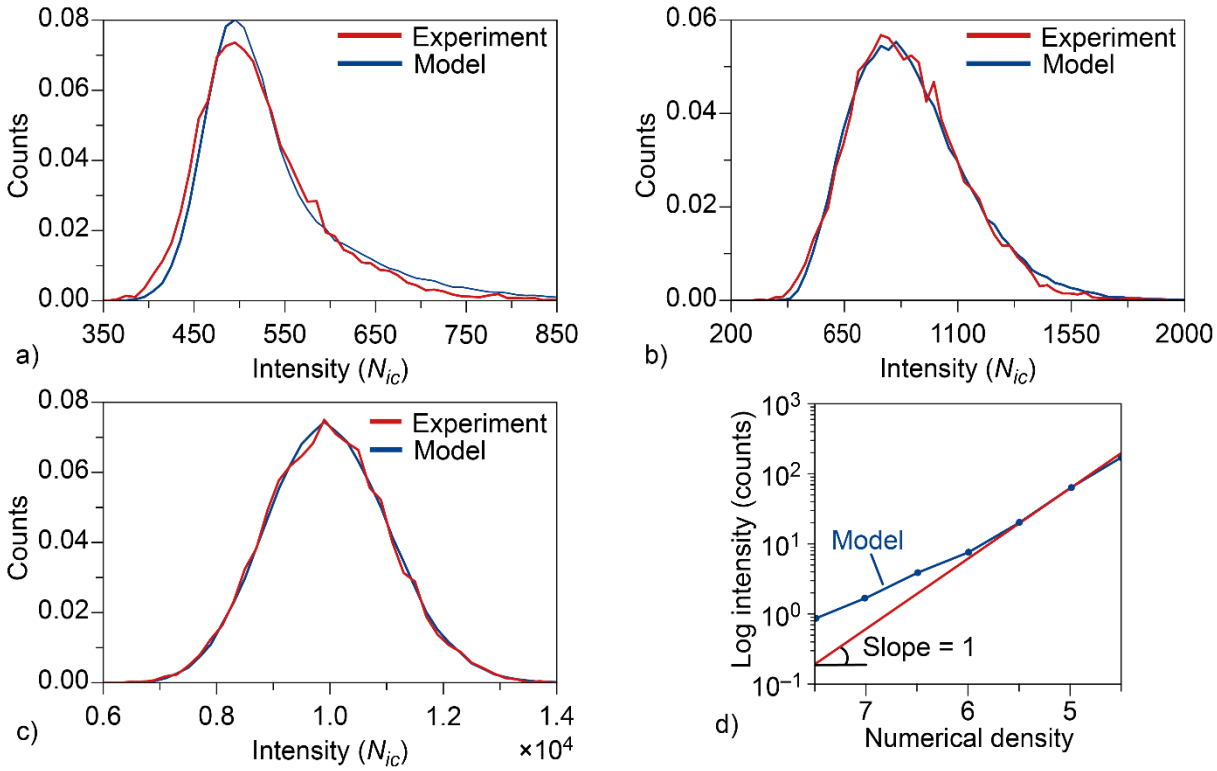


Figure 22: *The camera model fitted with the histograms of the different movies of the experiment, using maximum likelihood estimation which optimized variable i for light intensity. The histogram fits are shown where the filters used had a numerical density of 7.0, 5.5 and 4.0 for a), b) and c) respectively. d) shows the optimized parameter i plotted on a logarithmic scale versus the numerical densities of the filters used in the experiment.*

noise recordings where a ND filter of 7.0, 5.5 and 4.0 were used. As *i* increases when lower ND filters are used, we observe a shift in the shape of the distributions. The shift is caused by the Poisson component of eq. (5) dominating at low intensities, and the Gamma distribution dominating at high intensities. As can be seen the camera model shows a more optimal fit with higher with the distributions of higher *i* (additional fits in App A.4). Furthermore, in Figure 22d we plotted the optimized *i* values on a logarithmic scale. As *i* increases logarithmically with decreasing ND (Section 3.2), the plotted values for *i* should fit a straight line. However, only the higher *i* values fit with a straight line, implying that the camera model works optimal from intensities $\gtrapprox$8 photons.

An explanation for the deterioration of our the camera model for lower *i* could have been the use of a sequential set of filters to reach a desired ND. When a photon is reflected back on a filter located in the middle of the stack, it has a chance to be reflected a second time and still reach the camera. Consequently, *i* in the low light recordings are higher than expected. Therefore, when we fit the model to the histograms of these recordings, a higher *i* is required for a good fit than expected. Figure 22d shows an increasing error with lower NDs used. This can be explained by the amount of stacking increasing to reach higher NDs, consequently enhancing the effect of recording a higher *i* than expected. Therefore, *if* the stacking of filters is the reason for the poor fit in Figure 22d at lower intensities, our camera model performs better than expected as the expected values for *i* should lie at higher values in the graph. As a matter of fact, the dark images histogram fits with the camera model are only slightly less optimal than in the higher *i* regions (App. A.4). Furthermore, another potential explanation for the deterioration of our the camera model for lower *i* could be a wrong value used for parameter *c*. Figure 21 barely showed contributions of spurious charge to the histogram of the dark images. Instead of setting parameter *c* to a low value, we chose the value from the manual. As a consequence, the camera model has to fit a tail with the data that might not be there, potentially resulting in a slightly less optimal fit in Figure 22a. Lastly, the assumption was made that spurious charge is not created in the EM register, and that a more accurate camera model would process *c* in the Gamma distribution as well. However, this would mean that the camera model works less optimally for high *i*, as the Gamma distribution dominates at higher intensities. Since this is not the case, the assumption made in the camera model is not the cause of the less optimal fit in Figure 22a.

4.2.3 Background Noise

We determined the background noise parameter similar to how the validation of the camera model was performed. First, we selected four pixels of the non-simulated dataset where no emitters diffused by. Subsequently, we fitted the experimental distribution of n_{ie} with the camera model using a maximum likelihood estimation, which yielded the intensity of the incident light on the camera *i*. The fit is shown in Figure 23. As we selected a region in the movie were no emitters diffuse by, the only photons contributing to the resulting value for *i* came from background noise. Resulting in the background value parameter of 9.775 photons/pixel/frame.

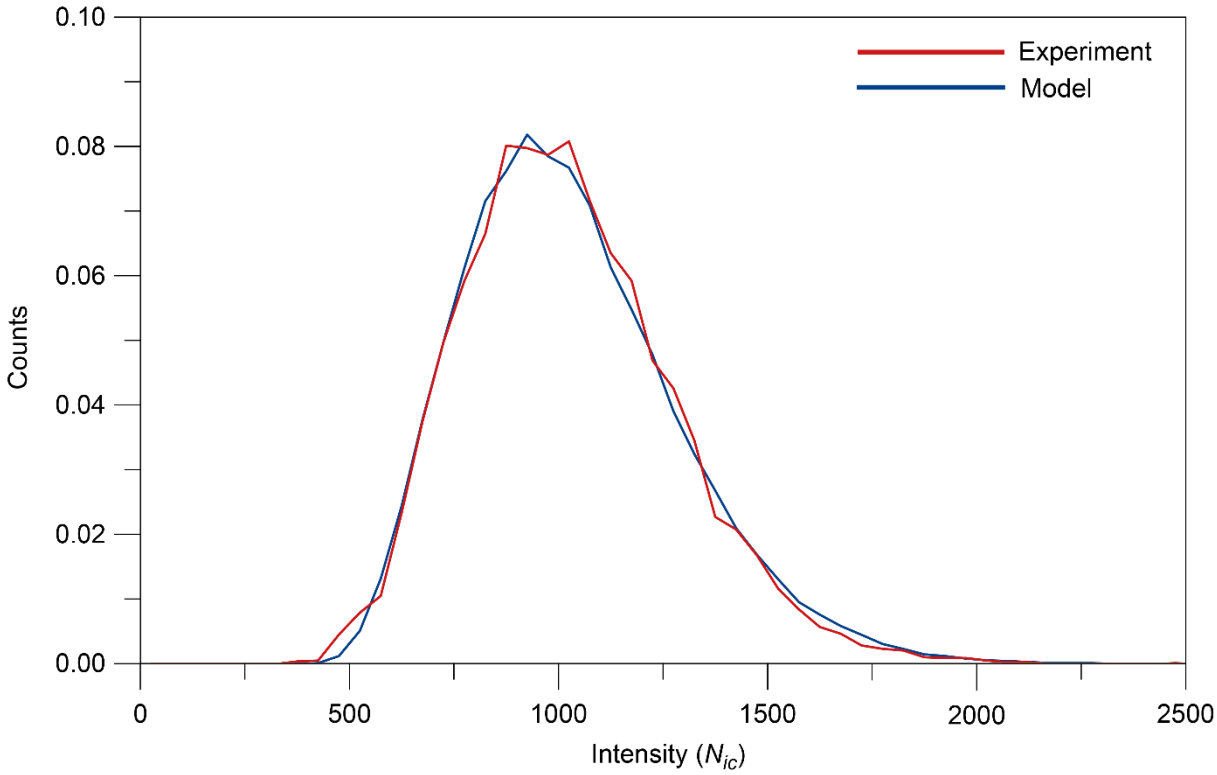


Figure 23: *The fit of the camera model with a histogram of four pixels of the non-simulated dataset by optimizing parameter* i *using a maximum likelihood estimation.*

As discussed, the camera model works optimally for values of *i* that lie on the straight line in Figure 22d, which contains the value of 9.775 photons/pixel/frame. Therefore, when we perform simulations using this background value, we can expected the camera noise to be simulated optimally. However, background noise differs per experiment. Therefore, we fitted our camera model with the distribution of n_{ie} of other experiments (App A.5) using a maximum likelihood estimation, of which the resulting background values were lower (App C.2) and therefore in the region of *i* were the camera model works less optimal in Figure 22d. The difference in background noise could be explained by the amount of emitters located in the recordings, where more emitters equals more scattering and consequently more background noise.

4.3 Detection Convolutional Neural Network

4.3.1 Training the Detection Neural Network

We trained the detection CNN with the training set described in Section 3.4. For this, we removed the images of <100 photons/emitter/frame to ensure gradient descent while training the CNN. Therefore, we trained the CNNs with emitter intensities ranging from the green arrow

to the upper limit in the intensity histogram of the non-simulated dataset (Figure 19, Section 4.1)

The training process of the detection CNN is shown in Figure 24 (Section 3.4). The observed

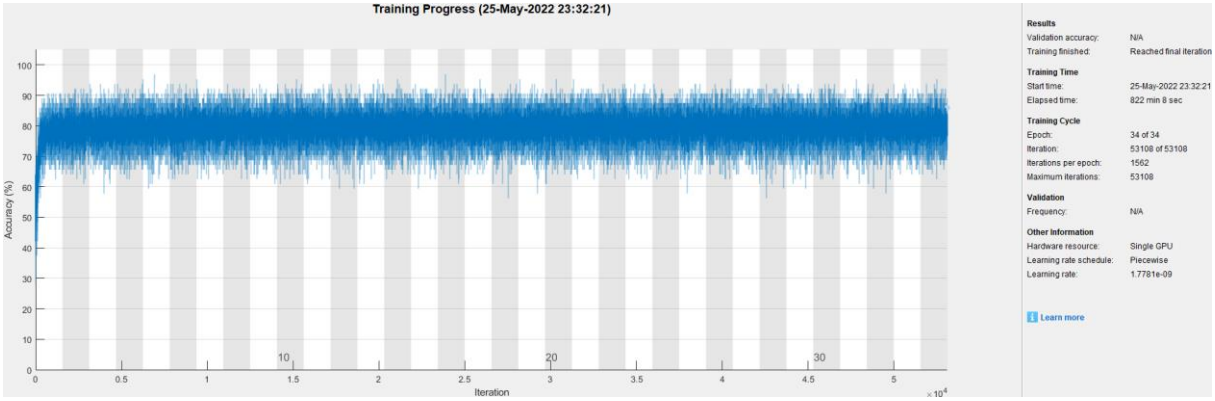


Figure 24: *The training process of the detection CNN.*

curve owes it shape to the decrease in learning rate overtime, where the learning rate was set high at the training start causing fast gradient descent, and subsequently was lowered showing smaller improvements overtime to eventually end up in the absolute global minimum of the loss function. Furthermore, we obtained a final accuracy of ~82% during training. At first, this value does not seem optimized. However, various emitters in the training set had intensities towards 100 photons and diffusion constants towards 4e−12 m^2 s^{-1}, of which consequently the SNR ratio might simply be too low to distinct emitters from noise. In fact, by reviewing the poor detection performance of DoM in these regimes (validation set 2, Section 4.3.2), a final training accuracy of 82~% seems satisfactory.

4.3.2 Validating the Detection Neural Network

We performed detection on the validation sets using our detection CNN and DoM as comparison. We introduced a method to compare the two methods in a fair manor (Section 3.5), but we observed that our CNN had trained itself to only detect emitters in a 30×30 box in the centre of the images (Section 4.4.2). This was a consequence of us deliberately choosing to only allow the average position of the motion-blurred emitters to be located in this region, to prevent emission signals from falling outside of the simulated frames. Therefore, the FPs that were detected in the noise frames by DoM, were only counted if they were located in the 30×30 box for a fair comparison to remain.

Figure 25 shows the detection results for validation set 1 where in Figure 25a the

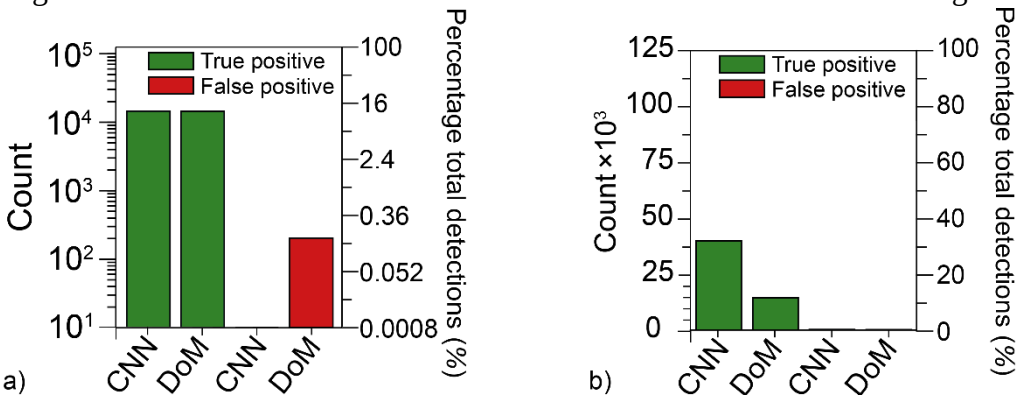


Figure 25: *Results of detections for validation set 1 using our detection CNN and DoM where in a) the TPs were equalized and b) the FPs were equalized. Our CNN resulted in ~250% increase of TPs compared to DoM.*

threshold of the CNN was tuned to equalize the TPs of both methods, whilst in Figure 25b the FPs were equalized. For each 125.000 frames containing an emitter, 125.000 TP detections are possible. Analogue to this, 125.000 FP detections are possible in the noise frames. Therefore, Figure 25 shows the absolute value of number of detections, whilst also showing the percentage of TP and FP detections compared to the maximum detections possible. As a result, Figure 25a can be used to compare the amount of FPs in a situation that both methods obtain as many TPs, whilst Figure 25b can be used to compare the TPs detections when both methods obtain as many FPs.

Figure 25 shows the detection results for validation set 2. Validation set 2 was the most

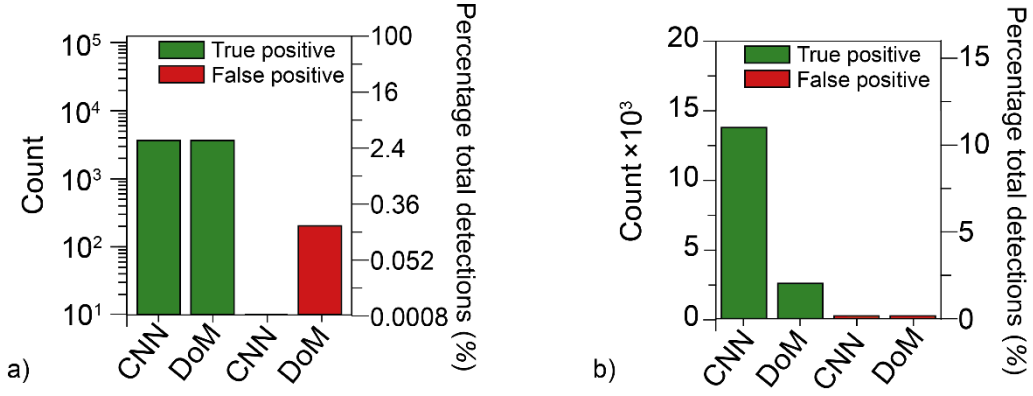


Figure 26: *Results of detections for validation set 2 using our detection CNN and DoM where in a) the TPs equalized and b) the FPs equalized. Our CNN performed better than DoM with an increase of TPs of ~500% for the TPs.*

complex set, as it contained emitters with the highest diffusion constant and lowest intensity, resulting in the lowest SNR visible in the frames. Interestingly, we observe an increase of

detection performance compared to DoM with the increase of the complexity of the validation sets (where validation sets 3 and 4 were also taken into consideration, App. A.6). This trend could be clarified because CNNs are able to approximate complex functions[21,48,49]. Therefore, it could have learnt very complex features of which attain to motion-blurred emitters for detection. Contradictory, DoM utilizes ‘’strels’’ in its algorithm for detection of emitters (simplified explanation, see App. A.2). The strels consist of simple features e.g. square, sphere, disk etc. As a result, DoM potentially does not recognize more complex motion-blurred signals, while the CNN does.

Of all validation sets, the results of validation set 1 and 2 are the most important. These two validation sets were tuned to the median of the non-simulated dataset, and therefore detections performed on the non-simulated dataset will mostly have increased performance of ~250% (8e−13 m^2 s^{-1}) and ~500% (4e−12 m^2 s^{-1}) compared to DoM.

Another advantage of the CNN over DoM, is that the threshold of the CNN is set *after* all frames have been processed for detection. This allows for quickly changing the threshold until the desired true positive/false positive rate is reached, i.e. tune it to optimize the Jaccard index (Outlook).

A potential bias might be present in the detection CNN: the diffusion constant and intensity of the emitters were varied during training, but the background noise was not. Therefore, the CNN has been optimized in feature detection for limited cases. This means that if the CNN *was* trained with varying background noise, it could have learnt additional features. These additional features might have been visible in the noise images of our validation sets, and could have resulted in an increase of FPs. However, *if* training the CNN with varying background noise results in an increase of FPs, this problem could be omitted by first measuring the background noise in a non-simulated experiment, and subsequently using our CNN specifically trained on this background noise intensity (assuming constant background noise during the experiments).

4.4 Localization Convolutional Neural Network

4.4.1 Training the Localization Neural Network

We used our detection CNN to perform detection in the training set simulated for the localization CNN. Subsequently, we set a threshold of 0.5 to with as a result 511.085 TPs

detected. Subsequently, we used frames of the TPs to train the localization CNN, with the process depicted in Figure 27.

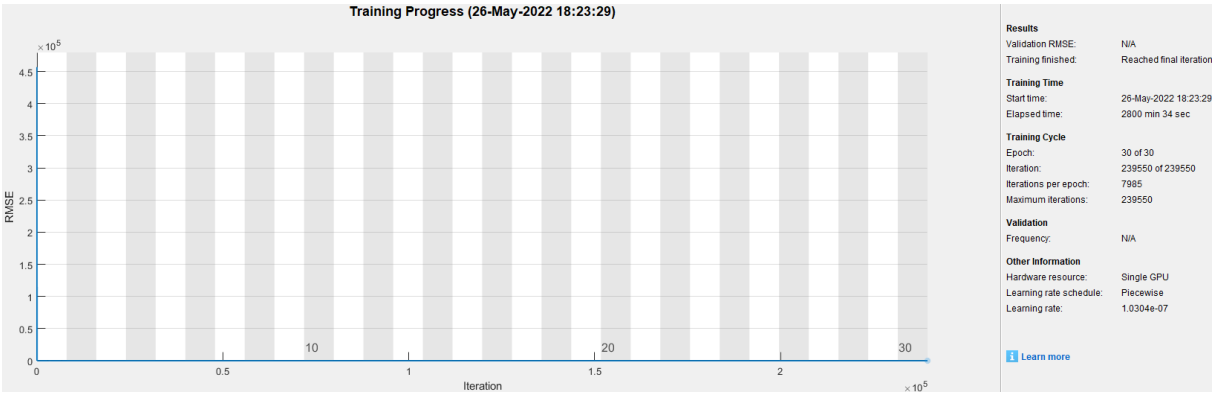


Figure 27: *The training process of the localization CNN.*

As can be seen in Figure 27, the loss function was optimized relatively fast. The same reason as for the detection CNN contributes to this: a high learning rate is used at the start of the training session which is subsequently decreased to ensure gradient descent to the absolute global minimum of the loss function.

4.4.2 Validating the Localization Neural Network

The trained localization CNN was subsequently validated. First, to validate the localization CNN, a detection step had to be performed on the validation sets. This was performed with DoM and the detection CNN, and the TPs were equalized (Figure 25a, Figure 25a, App. A.6a and c). The frames detected as TPs the validation sets by using the detection CNN were subsequently used to validate the localization CNN with. Similarly, the frames resulting from DoM detection were used for testing the benchmark localization methods. As the amount of TPs were equalised, we assumed that the resulting validation sets contained the same ‘’difficulty’’ of frames to localize motion-blurred emitters in. Subsequently, localizations in the frames of the validation sets were performed. Figure 28 shows the resulting localization precisions of the methods used for validation set 1, and Figure 28b for validation set 2.

Even though the localization CNN performed worse in both cases than the benchmark, the same pattern can be observed as with the detection CNN: the localization CNN performs relatively better in the complex situation. However, due to time constraints optimization of the localization CNN was not fully explored (see Outlook).

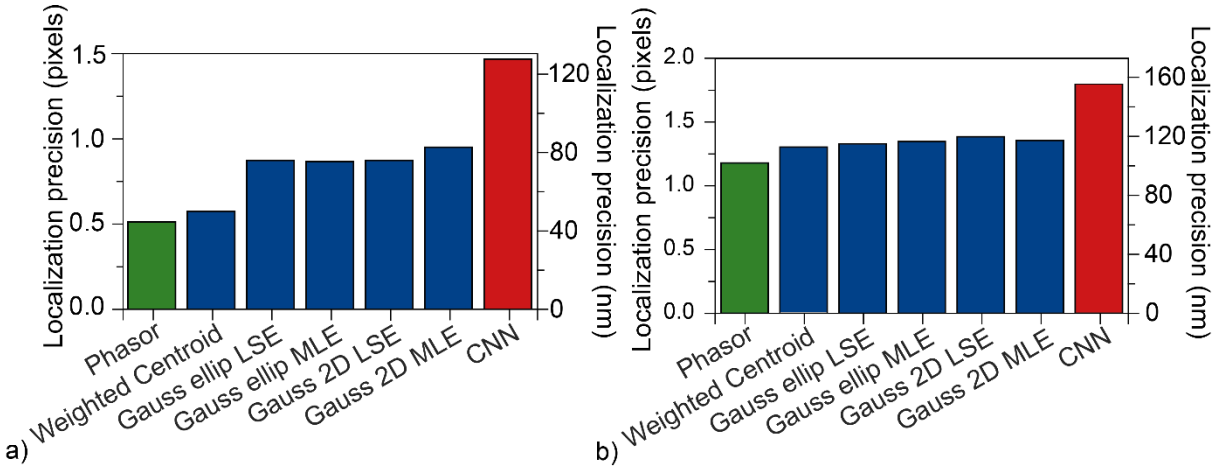


Figure 28: *The localization precisions of all methods used for in a) validation set 1, and b) validation set 2. All benchmark localization software outperformed the localization CNN, with phasor localization having the highest precision.*

The phasor benchmark method resulted in the best performing benchmark method. However, as described by Martens et al.[19], the localization precision of the phasor method depends on the region of interest size. Subsequently, we performed a ROI study, and the localization precision of every method for every validation set with respect to the region of interest size can be found in A.7. We observed the same ROI dependence of the phasor method as reported by Martens et al.[19] Subsequently, in Figure 28 the localization precisions were reported of all methods for were the optimal ROI size for the phasor method was used. The ROI dependence arises as localization errors are higher when either not all pixels containing signal of the emitters are located in the ROI, or when the ROI is too large and extra pixels only containing noise are used in the localization. The ROI dependence can be a drawback when unknown what the ROI size should be (determined by diffusion constant and SNR).

The weighted centroid was the second best performing localization method. Even though the localization precision is worse than that of the phasor method, it is independent of ROI size. That means when analysing a sample with emitters that have different diffusion coefficients, and determining optimal region of interest sizes is difficult, the weighted centroid method could remain better choice than the phasor localization method. Additional research towards the weighted centroid was performed, and can be found in Appendix E, where we confirmed the known bias for the method towards the middle of the images[18]. The weighted centroid shows a bias towards the middle of the images. However, this bias can be reduced by erosion operations, as utilized in DoM [54].

The Gaussian fitting methods performed most poorly of the benchmark methods. The rotational ellipsoidal Gaussian model was not included in Figure 28, as the fits did not converge. Convergence is drawback of Gaussian fitting, as localizations are lost due to the models often not converging on the motion-blurred signal (App. A8). As can be seen in Figure 28, the fitting with an ellipsoidal 2D Gaussian has slightly lower localization precisions than using a regular 2D Gaussian, which was also shown by Deschout et al.[15].

To conclude, the benchmark localization methods outperformed the localization CNN with respect to localization precision. The phasor method showed the best localization precision, has the drawback of being dependent on ROI size. The second best performing method is the weighted centroid, which contradictory to phasor fitting method has the advantage of independence of ROI size. And lastly, the Gaussian fitting methods performed the worst of the benchmark localization methods, and have an additional disadvantage that localizations are lost because of models not converging with the motion-blur signal. The performance of localization methods could be summarized as: phasor fitting > weighted centroid estimator > Gaussian ellipsoid fitting > Gaussian 2D fitting > CNN > Rotational ellipsoidal Gaussian fitting.

5. Conclusion

In this thesis, we investigated the use of convolutional neural networks for the detection and localization of the average position of motion-blurred emitters diffusing in two dimensions. Simulations were performed of 60×60 frames containing a motion-blurred emitter for training and validation of the CNNs. We tuned the parameters of the simulation software to a non-simulated dataset to obtain reliable performances for if the CNNs are used on the non-simulated dataset (App. C). The parameters tuned to the non-simulated datasets contained the diffusion constant, emitter intensity and background noise/

First, the parameter for the brightness of the emitters were obtained by analysing a non-simulated dataset obtained from previous research[33]. We found a distribution of brightnesses in the experimental data set. We selected a lower, median and upper limit of 5, 135 and 312 photons emitted by a emitter in a single frame, respectively. Second, parameters were obtained to simulate camera noise. This was performed by following a method described by Hirsch et al., which involved three different experiments using the EMCCD camera that was also used for the real SPT experiment[44]. The first two experiments involved taking a movie with the camera while the EM gain turned on and off, and by changing the light intensity by interchanging reflective filters. Parameter f *a*nd g for the camera model were successfully determined by performing a mean-variance test on the movies of the experiments. Parameter c_b was found by a third experiment which included taking a movie with a closed shutter, resulting in a movie of dark images. A probability distribution function of the camera for dark images was fit with the histogram of the movie using log-likelihood estimation, which resulted in the last two parameters, c, r. However, the fit was not optimal and therefore the spurious charge parameter c was obtained from the manual of the camera[55].

The camera model was subsequently validated, and it was shown that the model worked optimally for modelling high light intensity experiments, but deteriorated for low light experiments. A factor contributing to the lower performance on low light intensity experiments could be that for these movies several filters had to be stacked. This could have led to a higher transmission of photons than expected, explaining the intensity difference between the observed and expected intensity i, subsequently ensuring that the camera model work better than expected.

Simulations were performed to acquire a training set for the localization CNN (900.000 frames of motion-blurred emitters) and detection CNN (250.000 frames 1:1 motion-blurred emitters : noise images). Subsequently, validation sets (250.000 1:1 motion-blurred emitters : noise images) were simulated to test the performance of our CNNs. The diffusion constant for the emitters simulated in the training sets ranged from 8e−13 and 4e−12 m^2 s^{-1}, randomly assigned for each emitter. Similarly, the intensity ranged from 0 and 312 photons per emitter. The four validation sets contained combinations of the diffusion constants and intensity of 135 and 312 photons. 135 photons used in the simulations equalized to the median intensity observed in the non-simulated dataset, and 312 photons was a self-defined upper limit.

Gradient descent was first not observed during training of the detection CNN. Therefore, frames with emitters emitting <100 photons were removed, and subsequently gradient descent was observed. Validation of the detection CNN resulted an increase of performance with respect to published detection software as the validation sets got more complex, with ~500% more detections when compared to the most complex dataset, i.e. fast diffusion and low intensity. This result was expected as CNNs are powerful for approximating complex functions, and therefore it seems reasonable that its performance increases compared to a regular algorithm such as the benchmark detection software when the problem gets more complex.

The localization CNN was trained, validated and compared to published localization software. However, the benchmark localization methods outperformed the localization CNN in terms of localization precision. Interestingly, however, the same pattern for the detection CNN was observed, where the localization precision of the CNN relative to the benchmark methods increased as the validation sets get more complex. The phasor localization method resulted in the best localization precision. However, the method is dependent on region of interest. The second best method was the weighted centroid method, which in contrast with the phasor method has the advantage of being independent of the region of interest size. The Gaussian fitting methods resulted in relatively poor localization precisions, and have the additional disadvantage of losing localizations when the models do not fit with the data..

6. Future Research

There are several options for extending this research. The training images used are 60×60, which is relatively large for a localization step. The phasor method shows that optimal region of interest sizes, where all the pixels containing information about the emitter, are in a region of 13×13 (in the case of the most complex validation set). There is a large difference in surface area for the amount of noise in the training images and the amount of pixels containing information of the motion-blurred emitter. Therefore, a CNN could perhaps be trained more efficiently by only giving it the region of interest images. As discussed earlier, this method is prone to a bias towards the middle of images, as this is a local minimum in the loss function. It would be interesting to see how optimized the loss function can get when a localization CNN is trained on region of interest data when first a learning rate test is performed, and thus higher values of the learning rate are used at the start of the training session to find the global minimum of the loss function. Additionally, the CNNs could be trained with a larger training set where also the background noise is varied, since for every experiment the background noise differs. Consequently, the CNNs could be used for a wider range of experiments as long as the same camera is used.

Furthermore, to really extend the world of SPT, the detection CNN in combination with phasor localization could already be tested. The combination of these two methods would result in the most amount of true positive detections + the most optimal localization error. Moreover, as discussed briefly in the introduction, Lindén et al. showed that knowing the localization precision optimized the estimation of the diffusion coefficient precision[10]. It could be interesting to not only train the localization CNN to give coordinates for emitters, but also to return a localization precision. The localization precision could then be used to estimate the diffusion coefficient with higher precision. Instead of these consecutive steps, a CNN could also be investigated that immediately returns diffusion coefficients for emitters.

Another extension of this study is allowing the emitters to move in three dimensions, which causes more complex cases due to intensity and shape change of the PSF. A CNN could perhaps work better than the benchmark localization methods, as we have seen from this thesis that CNNs work increasingly better compared to regular algorithms when the problems gets more complex. Furthermore, the threshold for the benchmark detection software was set to a value where simply few false positives were detected, as this is undesirable in SPT. The

threshold of the CNN was subsequently tuned so that false positives or true positives were equalized. The threshold of the CNN was thus influenced by the threshold of the benchmark detection software. Another method to quantify how well both methods work without having to equalise true or false positives, is to calculate the maximum Jaccard index possible for both methods. The Jaccard index is a measure of how well a detection method works, and depends on the number of true positives, false positives and false negatives[56]. As a sidenote, the CNN also showed less false negatives than the benchmark detection software. The thresholds of both methods could subsequently be optimized separately, by tuning their threshold for which the Jaccard index is maximized. The Jaccard index lies between 0 and 1, where methods with a Jaccard index over 0.7 are considered good working methods[57].

Bibliography

[1] C. Smith, A. K. Hill, L. Torrente-Murciano, *Energy Environ. Sci.* **2020**, *13*, 331–344.

[2] H. Liu, *Chinese J. Catal.* **2014**, *35*, 1619–1640.

[3] M. Gambino, A. E. Nieuwelink, F. Reints, M. Veselý, M. Filez, D. Ferreira Sanchez, D. Grolimund, N. Nesterenko, D. Minoux, F. Meirer, B. M. Weckhuysen, *J. Catal.* **2021**, *404*, 634–646.

[4] E. T. C. Vogt, B. M. Weckhuysen, *Chem. Soc. Rev.* **2015**, *44*, 7342–7370.

[5] O. Deutschmann, H. Knözinger, K. Kochloefl, T. Turek, *Heterogeneous Catalysis and Solid Catalysts, 3. Industrial Applications*, **2011**.

[6] M. Campanati, G. Fornasari, A. Vaccari, *Catal. Today* **2003**, *77*, 299–314.

[7] N. F. Bin Dong, Nourhan Mansour, Teng-Xiang Huang, Wenyu Huang, *Chem. Soc. Rev.* **2021**, *50*, 6483–6506.

[8] F. C. Hendriks, F. Meirer, A. V. Kubarev, Z. Ristanović, M. B. J. Roeffaers, E. T. C. Vogt, P. C. A. Bruijnincx, B. M. Weckhuysen, *J. Am. Chem. Soc.* **2017**, *139*, 13632–13635.

[9] J. J. E. Maris, D. Fu, F. Meirer, B. M. Weckhuysen, *Adsorption* **2021**, *27*, 423–452.

[10] M. Lindén, V. Ćurić, E. Amselem, J. Elf, *Nat. Commun.* **2017**, *8*, 15115.

[11] C. Manzo, M. F. Garcia-Parajo, *Reports Prog. Phys.* **2015**, *78*, 124601.

[12] C. L. Vestergaard, *Phys. Rev. E* **2016**, *94*, 022401.

[13] B. Heit, *Fluorescent Microscopy*, **2011**.

[14] N. Naredi-Rainer, J. Prescher, A. Hartschuh, D. C. Lamb, *Confocal Microscopy*, **2013**.

[15] H. Deschout, K. Neyts, K. Braeckmans, *J. Biophotonics* **2012**, *5*, 97–109.

[16] K. I. Mortensen, L. S. Churchman, J. A. Spudich, H. Flyvbjerg, *Nat. Methods* **2010**, *7*, 377–381.

[17] A. Przybylski, B. Thiel, J. Keller-Findeisen, B. Stock, M. Bates, *Sci. Rep.* **2017**, *7*, 15722.

[18] L. von Diezmann, Y. Shechtman, W. E. Moerner, *Chem. Rev.* **2017**, *117*, 7244–7275.

[19] K. J. A. Martens, A. N. Bader, S. Baas, B. Rieger, J. Hohlbein, *J. Chem. Phys.* **2018**, *148*, 123311.

[20] K. O'Shea, R. Nash, **2015**, 1–11.

[21] T. Hastie, R. Tibshirani, G. James, D. Witten, *Springer texts* **2021**, *102*, 618.

[22] B. Huang, W. Wang, M. Bates, X. Zhuang, *Science* **2008**, *319*, 810–813.

[23] S. Jia, J. C. Vaughan, X. Zhuang, *Nat. Photonics* **2014**, *8*, 302–306.

[24] S. R. P. Pavani, M. A. Thompson, J. S. Biteen, S. J. Lord, N. Liu, R. J. Twieg, R. Piestun, W. E. Moerner, *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106*, 2995–2999.

[25] Y. Shechtman, L. E. Weiss, A. S. Backer, S. J. Sahl, W. E. Moerner, *Nano Lett.* **2015**, *15*, 4194–4199.

[26] Y. Shechtman, L. E. Weiss, A. S. Backer, M. Y. Lee, W. E. Moerner, *Nat. Photonics* **2016**, *10*, 590–594.

[27] C. Smith, M. Huisman, M. Siemons, D. Grünwald, S. Stallinga, *Opt. Express* **2016**, *24*, 4996-5013.

[28] A. S. Backer, W. E. Moerner, *J. Phys. Chem. B* **2014**, *118*, 8313–8329.

[29] T. Kim, S. Moon, K. Xu, *Nat. Commun.* **2019**, *10*, 1996.

[30] P. Zhang, S. Liu, A. Chaurasia, D. Ma, M. J. Mlodzianoski, E. Culurciello, F. Huang, *Nat. Methods* **2018**, *15*, 913–916.

[31] N. Boyd, E. Jonas, H. Babcock, B. Recht, *bioRxiv* **2018**, 267096.

[32] P. Zelger, K. Kaser, B. Rossboth, L. Velas, G. J. Schütz, A. Jesacher, *Opt. Express* **2018**, *26*, 33166-33179.

[33] F. M.R. Mayorga Gonzalez, J. J. E. Maris, M. Wagner, Y. Ganjkhanlou, J.G. Bomer, M.J. Werny, M. Odijk, F.T. Rabouw, A. van den Berg, B. M. Weckhuysen, *Prep.* **n.d.**

[34] J. W. Dobrucki, *Fluorescence Microscopy*, *Journal of Biomedical Optics*, **2013,** *19*, 97-142.

[35] D. Roy, K. Majhi, M. K. Mondal, S. K. Saha, S. Sinha, P. Chowdhury, *ACS Omega* **2018**, *3*, 7613–7620.

[36] J. Vangindertael, R. Camacho, W. Sempels, H. Mizuno, P. Dedecker, K. P. F. Janssen, *Methods Appl. Fluoresc.* **2018**, *6*, 022003.

[37] H. Shen, L. J. Tauzin, R. Baiyasi, W. Wang, N. Moringo, B. Shuang, C. F. Landes, *Chem. Rev.* **2017**, *117*, 7331–7376.

[38] S. E. Manahan, *Green Chemistry and the Ten Commandments of Sustainability*, **2005**.

[39] J. Wood, L. F. Gladden, *Appl. Catal. A Gen.* **2003**, *249*, 241–253.

[40] J. Kärger, D. M. Ruthven, D. N. Theodorou, J. Corma, A. Zones, C. Lamb, D. Carroll, *Diffusion as a Random Walk*, Wiley-VCH, Weinheim, **2012**.

[41] E. S. Machlin, *An Introd. to Asp. Thermodyn. Kinet. Relev. to Mater. Sci.* **2007**, 225–262.

[42] L. J. Friedman, J. Chung, J. Gelles, *Biophys. J.* **2006**, *91*, 1023–1031.

[43] G. Bottiroli, A. C. Croce, *Photochem. Photobiol. Sci.* **2004**, *3*, 189–210.

[44] M. Hirsch, R. J. Wareham, M. L. Martin-Fernandez, M. P. Hobson, D. J. Rolfe, *PLoS One*, **2013**, *8*, e53671.

[45] K. B. W. Harpsøe, M. I. Andersen, P. Kjægaard, *Astron. Astrophys.* **2012**, *537*, A50.

[46] M. Lelek, M. T. Gyparaki, G. Beliu, F. Schueder, J. Griffié, S. Manley, R. Jungmann, M. Sauer, M. Lakadamyali, C. Zimmer, *Nat. Rev. Methods Prim.* **2021**, *1*, 39.

[47] H. Deschout, F. C. Zanacchi, M. Mlodzianoski, A. Diaspro, J. Bewersdorf, S. T. Hess,

K. Braeckmans, *Nat. Methods* **2014**, *11*, 253–266.

[48] L. Möckl, A. R. Roy, W. E. Moerner, *Biomed. Opt. Express* **2020**, *11*, 1633.

[49] P. Kim, *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*, **2017**.

[50] S. E. Feller, *Methods Mol. Biol.* **2007**, *400*, 89–102.

[51] M. Fränzl, F. Cichos, *Sci. Rep.* **2020**, *10*, 12571.

[52] D. P. Kingma, J. L. Ba, *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* **2015**, 1–15.

[53] C. Smith, M. Huisman, M. Siemons, D. Grünwald, S. Stallinga, *Opt. Express* **2016**, *24*, 4996.

[54] E. S. Machlin, *Diffusion*, **2007**.

[55] "iXon Ultra 888," can be found under https://andor.oxinst.com/products/ixon-emccd-camera-series/ixon-ultra-888.

[56] S. C. Stein, J. Thiart, *Sci. Rep.* **2016**, *6*, 37947.

[57] J. Lorenzo-Navarro, M. Castrillón-Santana, E. Sánchez-Nielsen, B. Zarco, A. Herrera, I. Martínez, M. Gómez, *Sci. Total Environ.* **2021**, *765*, 142728.

Appendix A. Supplementary Figures

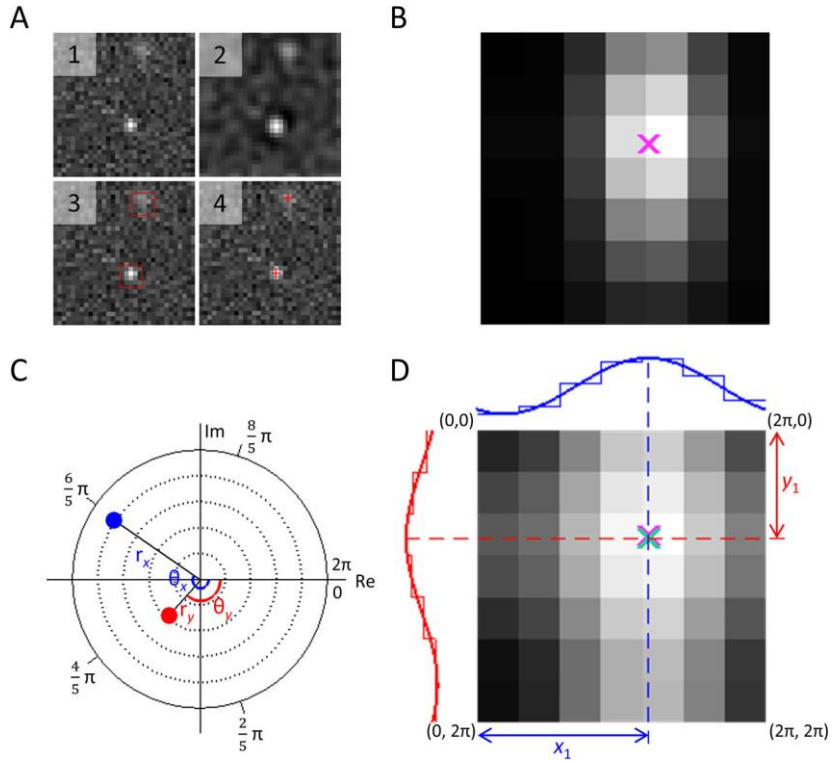


Figure A.1: a) Original experimental images containing the emission of fluorescent emitters. b) a ROI cut out of the experimental images. c) Phase vectors of the point spread function with their angles visible. d) The angles can be used to localize the emitters. (image from Martens et al.[19])

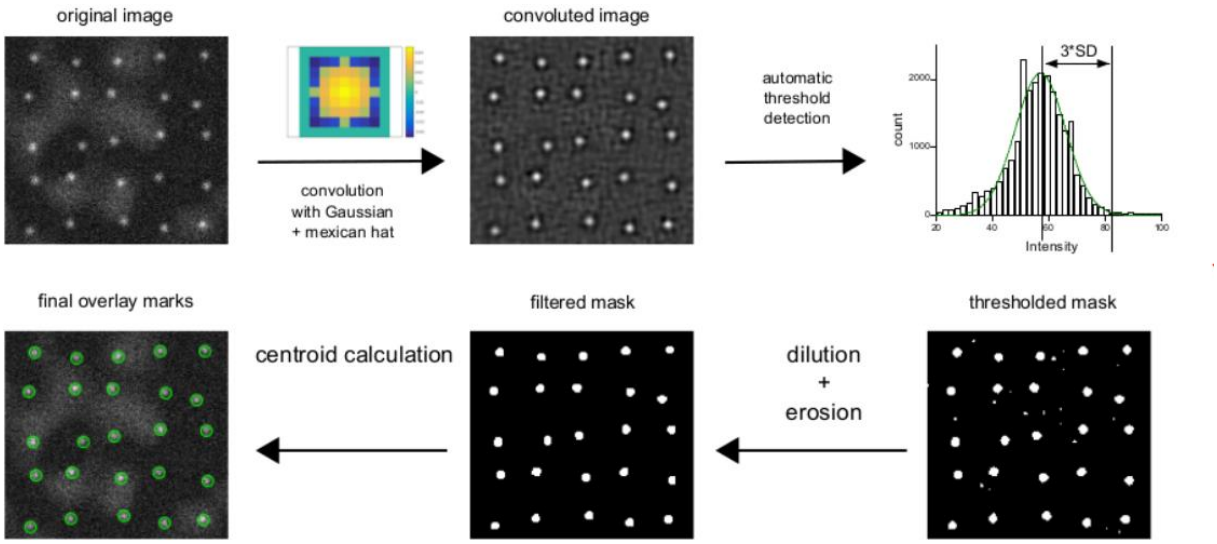


Figure A.2: Workflow of DoM detection software, where the image is convoluted with a Mexican hat filter, thresholded, diluted, thresholded, and finally the centroid positions of remaining spots are calculated. (DoM v1.2.2 by Katrukha et al.[54])

```
%% Perform a 2D Fourier transformation on the complete ROI.
fft_values = fft2(ROI);

%Get the size of the matrix
WindowPixelSize = size(ROI,1);

%Calculate the angle of the X-phasor from the first Fourier coefficient in
X
angX = angle(fft_values(1,2));
%Correct the angle
if (angX>0) angX=angX-2*pi; end;
%Normalize the angle by 2pi and the amount of pixels of the ROI
PositionX = (abs(angX)/(2*pi/WindowPixelSize) + 1);
%Calculate the angle of the Y-phasor from the first Fourier coefficient in
Y
angY = angle(fft_values(2,1));
%Correct the angle
if (angY>0) angY=angY-2*pi; end;
%Normalize the angle by 2pi and the amount of pixels of the ROI
PositionY = (abs(angY)/(2*pi/WindowPixelSize) + 1);

%Calculate the magnitude of the X and Y phasors by taking the absolute
value of the first Fourier coefficient in X and Y
MagnitudeX = abs(fft_values(1,2));
MagnitudeY = abs(fft_values(2,1));

%Print a line with results
fprintf('\nPosition found at X=%.2f, Y=%.2f, with phasor magnitude in
X=%.2f, phasor magnitude in Y=%.2f\n', PositionX, PositionY, MagnitudeX,
MagnitudeY);
```

Figure A.3: MATLAB script snippet used for phasor fitting. (provided in supplementary information of Martens et al.[19])

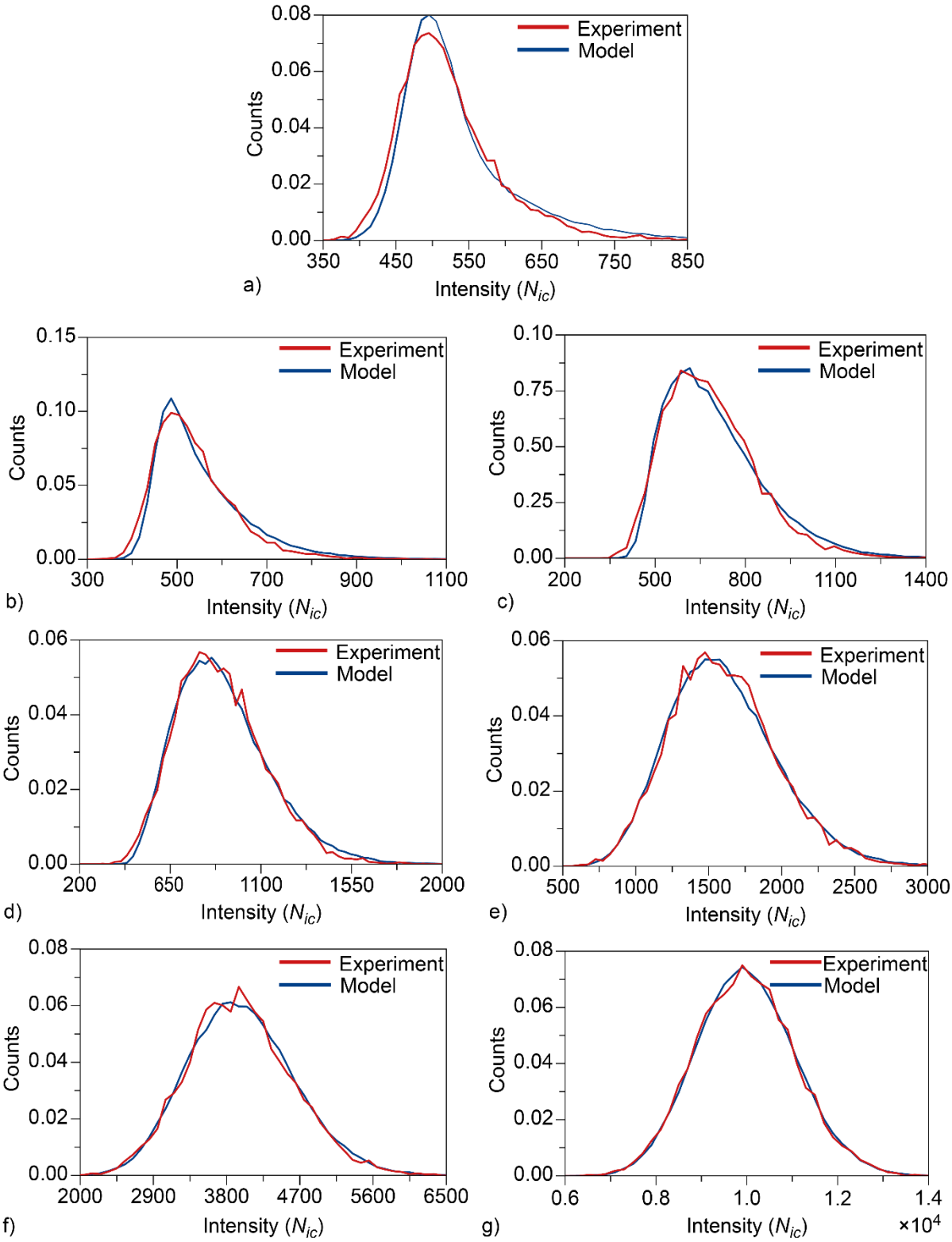


Figure A.4: *histogram fits of the other movies with the camera model where the numerical densities of the filters where a) 7.5, b) 7.0, c) 6.5, d) 6.0 e) 5.5 f) 5.0 g) 4.5.*

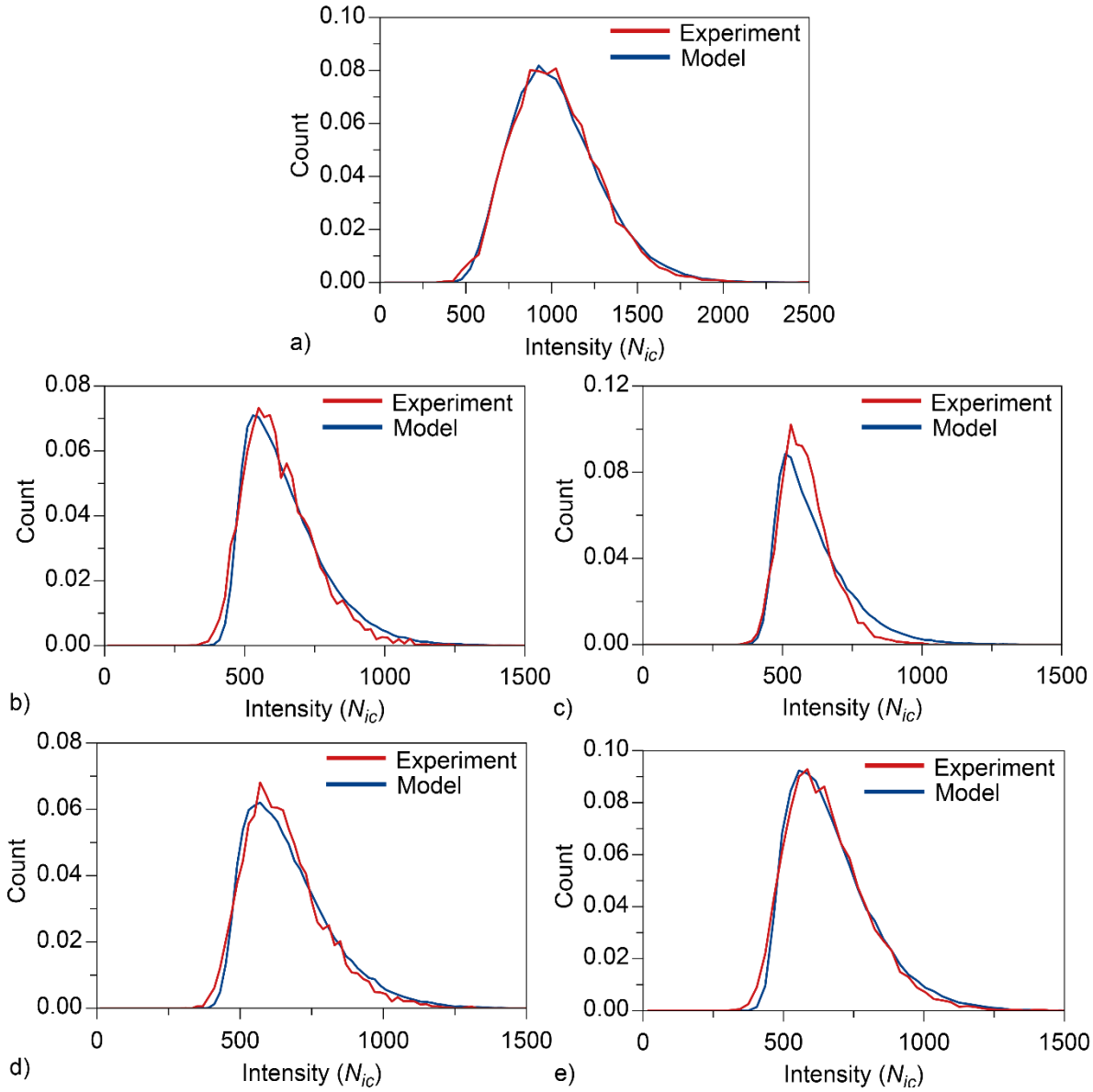


Figure A.5: *the resulting fits of the camera model with other SPT experiments using maximum likelihood estimation to optimized parameter i. a) The experiment which was used for parameter acquisition in this thesis, where quantum dots confined in 150 nm and 0.0075 NaOH M. b) confined in 150 nm and 0.02 NaOH M. c) confined in 50 nm. d) confined in 100 nm. e) confined in 150 nm.*

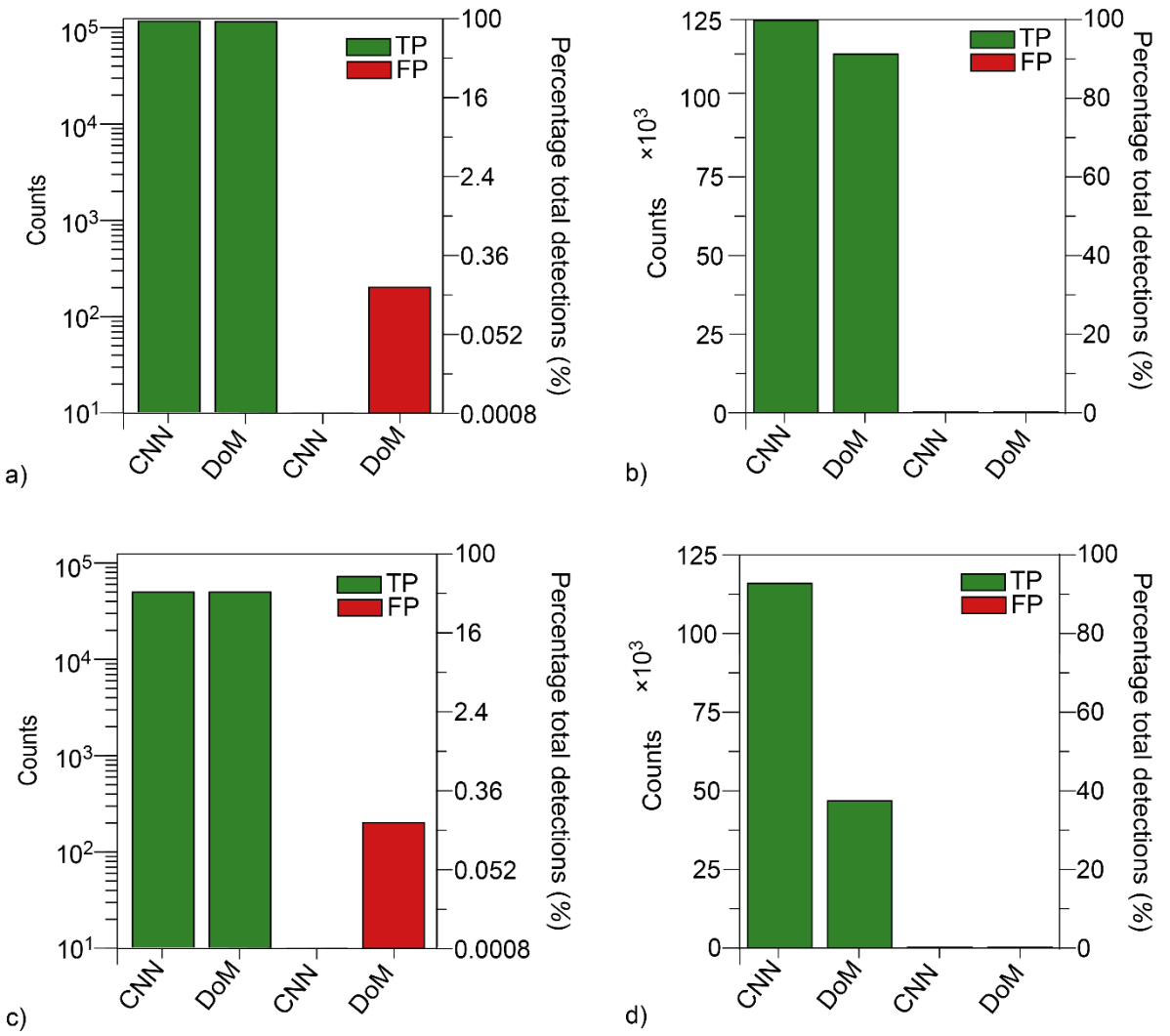


Figure A.6: *Results of detections using our detection CNN and DoM where in a) validation set 3 with the TPs equalized, b) validation set 3 with the FPs equalized, c) validation set 4 with the TPs equalized, d) validation set 4 with the FPs equalized.*

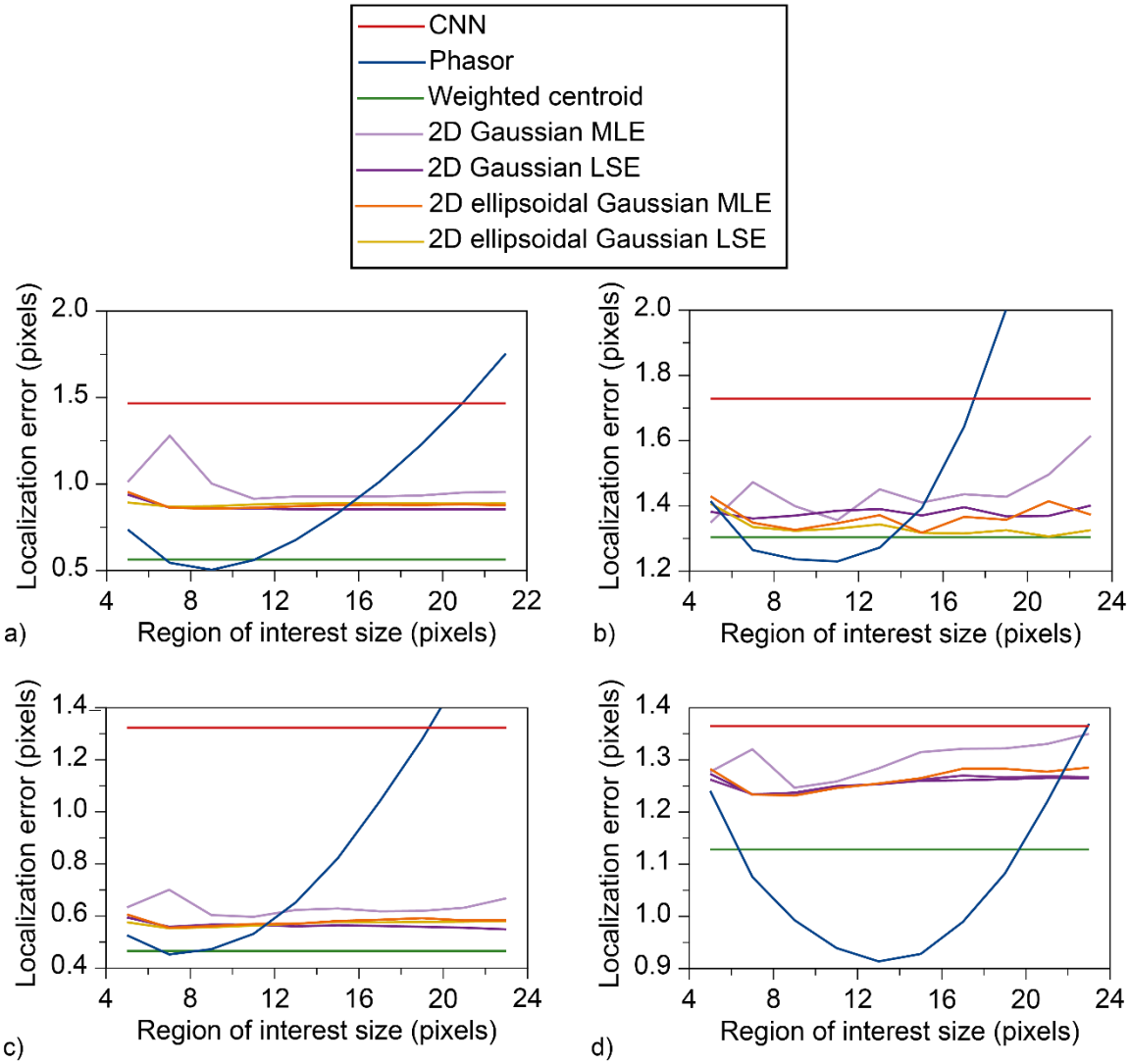


Figure A.7: *The localization precision of all localization methods as function of ROI size when localization was performed on validation sets a) 1, b) 2, c) 3 and d) 4. As can be seen, only the phasor method depends on the region of interest size.*

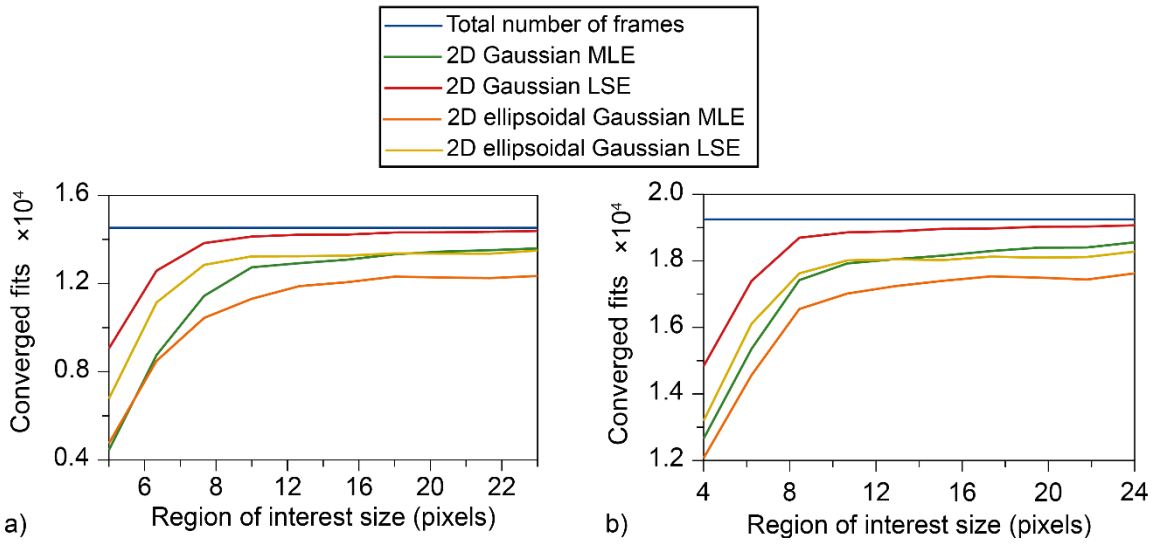


Figure A.8: *Only converged with of Gaussian models with the ROIs are counted as localizations. Here the amount of converged fits are reported for the different Gaussian models using MLE and LSE. The blue line shows the total frames localized. a) shows the convergence for validation set 1, and b) the convergence of validation set 2.*

Appendix B. Gaussian Models

The 2D-Gaussian model is mathematically described as

$$g(x, y, \vec{p}) = p_0 exp(-((x - p_1)^2 + (y - p_2)^2)/(2p_3^2))) + p_4 \quad (12)$$

with $\vec{p}$ a vector containing all parameters *p*, p_0 the amplitude of the emission signal, p_1 the centre coordinate *x*, p_2 the centre coordinate *y*, p_3 the standard deviation of the emission signal in *x*, p_4 the offset of the emission signal (originating from camera bias and background noise).

The ellipsoidal version of the 2D-Gaussian model is mathematically described as

$$g(x, y, \vec{p}) = p_0 exp\left(-\frac{1}{2}\left(\frac{(x - p_1)^2}{p_3^2} + \frac{(y - p_2)^2}{p_5^2}\right)\right) + p_4 \quad (13)$$

with p_5 the standard deviation of the emission signal in the *y*-dimension.

The ellipsoidal version of the 2D-Gaussian model is mathematically described as

$$g(x, y, \vec{p}) = p_0 exp\left(-\frac{1}{2}\left(\frac{((x - p_1)^2 \cos p_6 - (y - p_2) \sin p_6)^2}{p_3^2} + \frac{((x - p_1)^2 \sin p_6 + (y - p_2) \cos p_6)^2}{p_5^2}\right)\right) + p_4 \quad (14)$$

with p_6 the rotation angle in radians with respect to the *x* and *y* coordinate axes.

Appendix C. Table of All Parameters

Parameter	Value
Simulation (fixed parameters)	
Field of view	5.2×5.2 µm
Camera grid	60×60 pixels
Pixel size	86.67 nm
Frametime	0.05 s
Number of emitters/frame	1
Background noise	9.775 photons pixel^{-1} frame^{-1} (195.5 photons s^{-1} pixel^{-1})
PSF	
PSF sampling x, y	10 nm
PSF sampling z	10 nm
Convolution resolution	1 nm
Cut-off in z	1 µm
Wavelength	600 nm
Refractive index	1.518
Magnification microscope	91
Numerical aperture	1.40
Camera	
Quantum efficiency (q)	0.9
Readout noise (r)	37.3955
Spurious charge (c)	0.00025[55]
Electron magnification gain (g)	71.0067
Camera bias (c_b)	488

Electrons per ADC (*f*)	1.1512
Simulation training set	
Number of frames	1.000.000
Diffusion constant	Random between 4e−12 and 8e−13 $m^2 s^{-1}$
Emitter intensity	Random between 0 and 312 photons/emitter/frame
Simulation validation set 1	
Diffusion constant	8e−13 $m^2 s^{-1}$
Emitter intensity	135 photons/emitter/frame
Simulation validation set 2	
Diffusion constant	4e−12 $m^2 s^{-1}$
Emitter intensity	135 photons/emitter/frame
Simulation validation set 3	
Diffusion constant	8e−13 $m^2 s^{-1}$
Emitter intensity	312 photons/emitter/frame
Simulation validation set 4	
Diffusion constant	4e−12 $m^2 s^{-1}$
Emitter intensity	312 photons/emitter/frame
Detection step[54]	
PSF sigma	2 pixels
Pixel size	86.67 nm
Threshold	3×SD (See A.X)
Gaussian fitting	
p_0	Max pixel value image – 1030 (avg. noise value)
p_1	*x* position obtained from detection step
p_2	*y* position obtained from detection step

p_3	3 pixels
p_4	1030 (avg. noise value)
p_5	3 pixels
p_6	0 rad

Table C.2: Parameter *i* determined for other non-simulated datasets.

Experiment	***i* in photons/pixel/frame**
0.0075 NaOH M and quantum dots confined in 150 nm (experiment used for parameter acquisition)	9.7747
0.02 NaOH M and quantum dots confined in 150 nm	2.8140
Quantum dots confined in 50 nm	2.1910
Quantum dots confined in 100 nm	3.3486
Quantum dots confined in 150 nm	3.3351

Appendix D. Resizing the PSF

For the PSF to be convolved with sub-pixel precision, the simulated PSFs are shifted and resized to fit with the dimensions of the pixel grid (Section 3.1). However, during resizing it is important that the amount of photons are kept the same, for the accuracy of the SNR. One of the methods interpolates pixel values of the PSF to end up at the correct dimensions, which is therefore inaccurate. Different methods for resizing were tested by convolving the resized PSFs with 10 photons. The absolute PSFs after resizing and convolution using both methods are depicted in Figure D.1a. The absolute and relative difference between the two resized PSFs are depicted in D.1b.

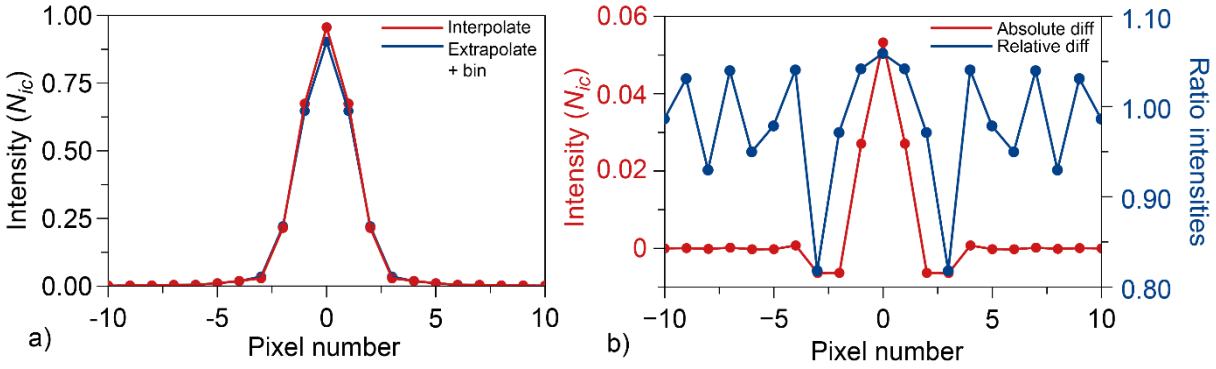
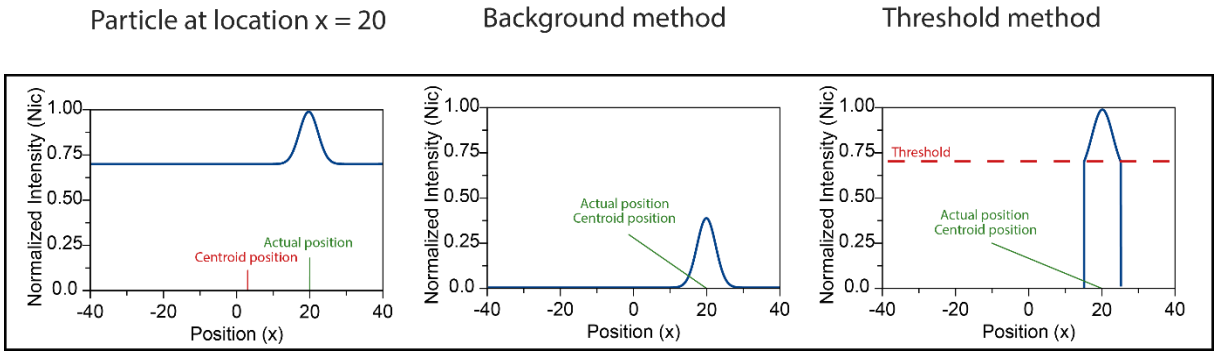


Figure D.1: *a) The PSF after resizing and subsequent convolution of 10 photons, using the two different methods. b) The absolute and relative difference between the PSFs in a).*

As can be seen in Figure D.1, the methods result in slightly different resized PSFs. The reason is that when the PSF size is reduced using interpolation, the photon intensity at a pixel can differ drastically. An enlargement of the PSF to a size so that the PSF can be subsequently binned into the desired smaller PSF solves this problem. Figure D.b shows that there is only a small difference in the middle region of the resized PSFs (the region between −4 and 4 pixels). However, the region in the middle only differs about 0 to 0.05 pixel counts, which is a negligible difference. As the more precise extrapolation + binning requires an extra step, the computation time of both methods differ. The interpolation method required 56.97 seconds to resize and convolve 5000 PSFs, whilst the more precise extrapolation + binning method required 206.40 seconds. The minor difference in the intensity values for the two different resizing methods and the major difference in computation time made us choose to work with the faster interpolation method for the simulations.

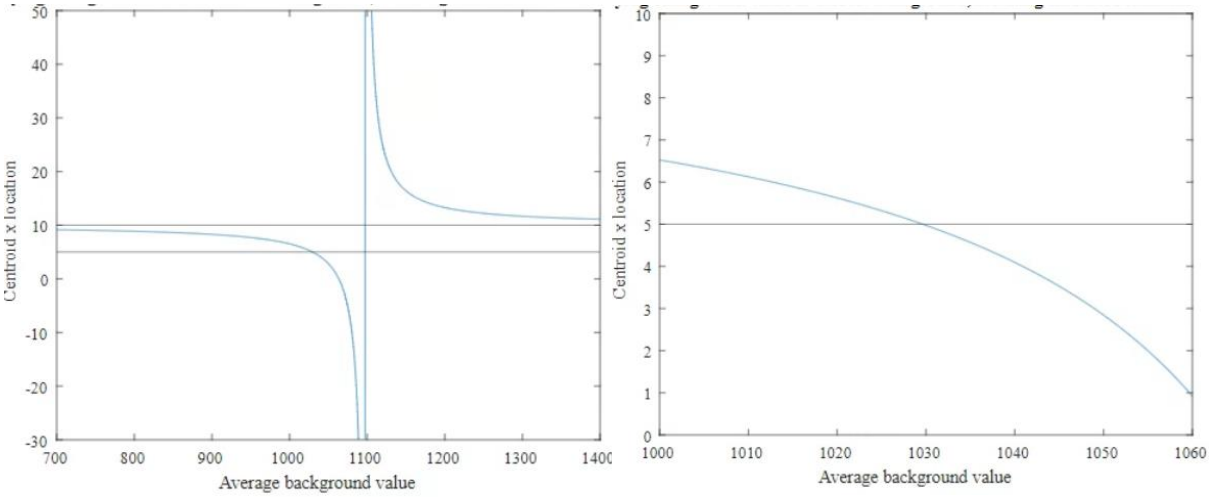
Appendix E. Centroid Estimator

Additional research was performed towards localization using a centroid estimator. Two methods for centroid localization were developed: referred to as the ‘’background method’’ and



the ‘’threshold method’’. Figure E.1 shows the working mechanism of both methods, were in Figure E.1a centroid localization is performed using eq. (8) on the unprocessed signal of a 1D emitter, resulting in a wrongly predicted centroid position estimation. Moreover, Figure E.2 shows the working mechanism of the background method, where the average noise values are removed from the 1D image, resulting in a correctly predicted centroid location. Lastly, the threshold method is described in Figure E.2, where all values below a threshold are set to zero, resulting in a correct centroid positions if the threshold is set well enough above the average noise values.

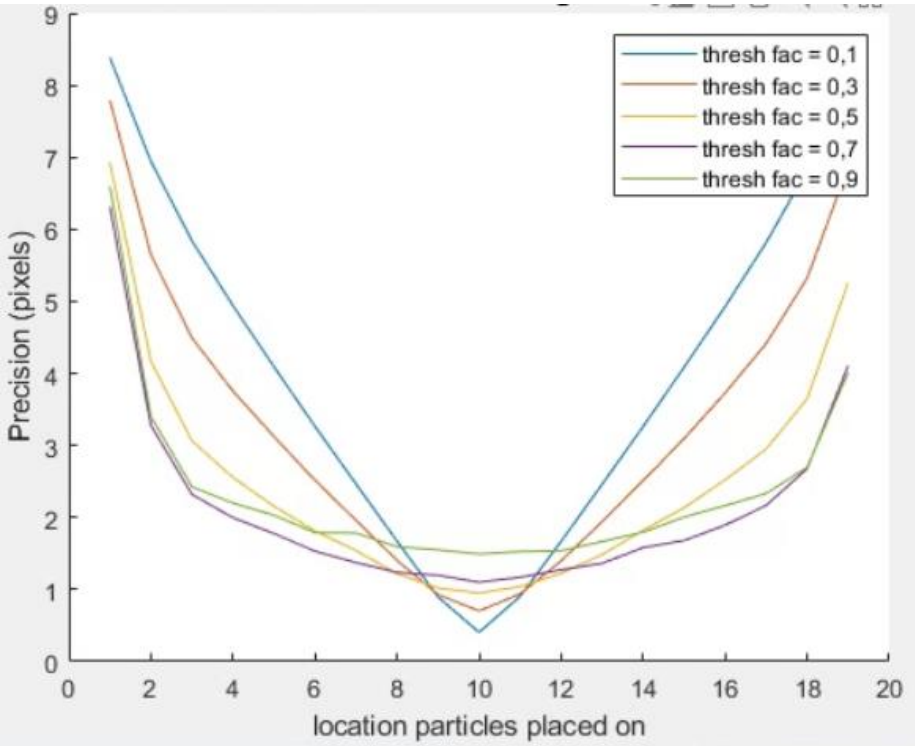
The background method was tested by performing a simulation of a single motion-blurred emitter located at position $x = 10$ in a pixel grid of 19×19. Subsequently, the background method was used to detect the emitter as a function of average background value. The correct average background value was contained in the range of values used. Figure E.2a shows the



estimated centroid positions for the average background values used, with a line intercept drawn

for the correct centroid position for *x*, 10. As can be seen, not using the exact average background value results in a localization error. Furthermore, the horizontal asymptotes approach the centre coordinate of the image, introducing the known bias of the centroid method[18].

The centroid method was tested by performing a simulation of 95.000 motion-blurred emitters in a 19×19 pixel grid, where sets of 5.000 emitters each had their average coordinates located exactly in the full range of *x*-pixels in the camera grid, i.e. (1,10), (2,10),… (19,10). Subsequently, the threshold centroid method was performed for each set of emitters, while also varying the threshold value, summarized in Figure E.3. The threshold was defined as a value



between the average noise and the amplitude of the signal, i.e. a threshold of 0 equals the avg. noise, threshold of 0.5 equals the value between avg. noise and amplitude, and threshold of 1 equals the amplitude of the signal. Consequently, Figure E.3 shows that the threshold centroid method has a precision that is dependent on location of the emitter in the image. Further investigation showed that using a lower threshold results in an increase of noise pixels not being thresholded. As these pixels are evenly distributed through the image, their centroid position is located in the middle of the image. Therefore, by adding these pixels in the calculation of the centroid position of the emitter, the estimation will move towards the middle of the image. Therefore, as can be seen in Figure E.3, when approximating the centroid position of an emitter located on the side of an image, e.g. $x = 5$, the centroid method with a lower threshold will show

a higher bias towards the middle, resulting in a larger localization error, and therefore worse localization precisions.

DoM localization uses a complicated algorithm that utilizes the threshold method[54]. In addition, they perform dilation and erosion operations in order to remove the isolated noisy pixels, to try to emit the bias towards the middle of the images.