Journal of
*Imaging*

*Article*

# upU-Net Approaches for Background Emission Removal in Fluorescence Microscopy

**Alessandro Benfenati** [1,2]

[1] Environmental and Science Policy Department, Università degli Studi di Milano, Via Celoria 2, 20133 Milan, Italy; alessandro.benfenati@unimi.it

[2] Gruppo Nazionale Calcolo Scientifico, Istituto Nazionale di Alta Matematica, P.le Aldo Moro 5, 00185 Rome, Italy

**Abstract:** The physical process underlying microscopy imaging suffers from several issues: some of them include the blurring effect due to the Point Spread Function, the presence of Gaussian or Poisson noise, or even a mixture of these two types of perturbation. Among them, auto–fluorescence presents other artifacts in the registered image, and such fluorescence may be an important obstacle in correctly recognizing objects and organisms in the image. For example, particle tracking may suffer from the presence of this kind of perturbation. The objective of this work is to employ Deep Learning techniques, in the form of U-Nets like architectures, for background emission removal. Such fluorescence is modeled by Perlin noise, which reveals to be a suitable candidate for simulating such a phenomenon. The proposed architecture succeeds in removing the fluorescence, and at the same time, it acts as a denoiser for both Gaussian and Poisson noise. The performance of this approach is furthermore assessed on actual microscopy images and by employing the restored images for particle recognition.

**Keywords:** neural network; Perlin noise; U-Nets; particle estimation; deep learning; microscopy imaging

check for updates

**Citation:** Benfenati, A. upU-Net Approaches for Background Emission Removal in Fluorescence Microscopy. *J. Imaging* **2022**, *8*, 142. https://doi.org/10.3390/jimaging8050142

Academic Editors: Ashutosh Sharma and Paul R. Barber

Received: 22 April 2022
Accepted: 18 May 2022
Published: 20 May 2022

## 1. Introduction

The mathematical model underlying the physical process of registering fluorescence microscopy images relies on a linear model

$$\mathbf{gn} = \mathcal{N}(\mathbf{Hx}^\star + b) \tag{1}$$

where $\mathbf{x}^\star$ represents the clean image, $\mathbf{H}$ is the blurring linear operator which represents the Point Spread Function (PSF) that introduces perturbation in the recorded image, such as unsharp edges or motion artifacts. The scalar quantity $b$ is constant background emission, and $\mathbf{gn}$ represents the registered image by the microscope. Equation (1) refers to the discretization of a Fredholm integral of the first kind: the interested reader may find all the technical details in the seminal work [1]. In Equation (1) $\mathcal{N}(\cdot)$ represents the noise affecting the image: in case of additive noise Equation (1) reads as

$$\mathbf{gn} = \mathbf{Hx}^\star + b + \boldsymbol{\eta} \tag{2}$$

where $\boldsymbol{\eta}$ is a Gaussian variable of zero mean and variance equal to $\sigma^2$. In this case the noise is statistically independent of the data. A further noise affecting microscopy images is Poisson noise, which models the errors done in counting processes: indeed, the image acquisition process consists in counting the photons hitting the Charge-Coupled Device (CCD) of the camera. Such kind of noise is signal–dependent: low values on $\mathbf{Hx}^\star + b$ induce a high level of noise, while Poisson noise is not highly disruptive when the values of the recorded image are large. Typical microscopy images are affected by both noises:

the Gaussian statistics models the so–called *read noise*, which is the perturbation due to the electrical components of the camera, whilst the Poisson noise (called also *photon noise*) models the error on the photon–counting process. The details of the physical process and the subsequent mathematical treatment can be found in [1,2].

A classical approach for the restoration of microscopy images described by Equation (2) relies on a variational framework [1]: an estimation of $\mathbf{x}^\star$ is obtained by solving an optimization problem involving the sum of two functionals. The first term is the *fit–to–data* or *data fidelity*, which measures the discrepancy between the registered data and the recovered image. The choice for this functionality depends on the noise affecting the data: in presence of Gaussian noise the most suitable one is the Least Square functional [3], whilst in presence of Poisson noise the (generalized) KullBack–Leibler function [4,5] is widely employed. The second term occurring in the minimization is the so-called regularization term, which has the role of considering some characteristics of the desired image and controlling the influence of the noise in the reconstruction. Several options are available for such function: $\ell_2$ norm for diffuse components [6], $\ell_1$ for promoting sparse solution [7,8], Total Variation functional [9–11] (or its smooth counterpart [12]) for edge preserving. One may consider a composite regularization function, such as the Elastic–Net [13], or even non-convex options are available [3,14]. The sum of the data fidelity and the regularization function is balanced by the so-called regularization term, whose estimation is still an open problem [15,16]. Some mathematical manipulation of the regularization function [17,18] or the entire functional [19] may lead to remarkable results. Modern techniques aim to couple classical variational approaches with Deep Learning techniques [20–22].

In some cases the term $b$ is not constant, and it actually models various degrading contributions: undesired emission from auto–fluorescent objects (i.e., molecules, beads), fluorophores that spontaneously emit light but that are not the actual target of interest, the reflection of the light used to investigate the sample ([23, Figure 2]), diffraction due to the medium surrounding the objects of interest. Merging all the above observations on the influence of the statistical noise on the recorded image, one can hence write then Equation (2) as

$$\mathbf{gn} = \mathcal{P}(\mathbf{H}\mathbf{x}^\star + \mathbf{b}) + \boldsymbol{\eta} \tag{3}$$

where $\mathcal{P}(\cdot)$ models the Poisson noise, $\boldsymbol{\eta}$ is a Gaussian random variable of zero mean and variance $\sigma^2$ and $\mathbf{b}$ is no more a scalar but an array of suitable dimension that models the background emission. In some applications, such as particle position estimation [24–26], the presence of this artifact may worsen the performance of the algorithms employed for this task. Moreover, in order to test such algorithms, one must have at their disposal a large number of microscopy images with ground truth data to properly assess the performance of the developed strategy. One way to achieve this is to generate synthetic data: the blurring operator $\mathbf{H}$ can be modeled by a Gaussian PSF or by an Airy Disk [27–29], while the statistical noise (both Gaussian and Poisson types [30]) may be simulated by established software procedures.

The auto-fluorescence noise can be simulated in various ways. In [31] the authors employ a combination of vector and scalar-based PSF, polarization detection strategy, spectral crosstalk, and other physical phenomena. Montecarlo methods have been widely used in the literature [32–34], both for the reliable approximation of the light scattering and for the simple numerical implementation. The authors in [35] develop a quasi–analytical model for improving the brute force approach based on Monte Carlo methods. Intermittent fluorophores activation was considered in [36], where the spatial distribution of such fluorophores is sparse. The author of [37] analyses the theoretical framework of emission and scattering phenomenon in fluorescence microscopy.

In this work, the simulation of the background emission is based on the employment of Perlin noise [38]. This procedure is widely used on CGI (Computer–Generated Imagery), it has been employed to simulate clouds in satellite images [39–41] and recently has found several applications for the generation of synthetic microscopy images. In [42,43] Perlin noise at different frequencies is added to the ground truth images for the simulation of the background straining. The authors in [44] use a combination of Perlin noise and Gaussian functions for inserting diffuse spots of fluorescence in the background of synthetic images. Due to its flexibility and ability to simulate realistic texture in biological images, Perlin noise is employed for simulating both background emission and cell textures [45–48], and in [49] a combination of Voronoi diagrams and Perlin noise is used to generate realistic cytoplasm boundaries. In Figure 1, a comparison between an actual image (Figure 1a) and a synthetic one is provided (Figure 1b): the Perlin noise indeed provides a concrete model for the background emission. Recent works [50,51] develop a BGnet architecture to efficiently remove the structured background: the authors employ the Perlin noise for two main reasons: firstly, because it resembles the structured background that can be encountered under real–world experimental conditions and, secondly, because its structure can be controlled by properly setting its spatial frequency among the octaves.

To the best of the author's knowledge, there is no developed variational approach for background emission removal: in this work, a completely data–driven procedure is developed for the background emission removal. The proposed strategy relies on Deep Learning techniques, namely on U-net architectures.

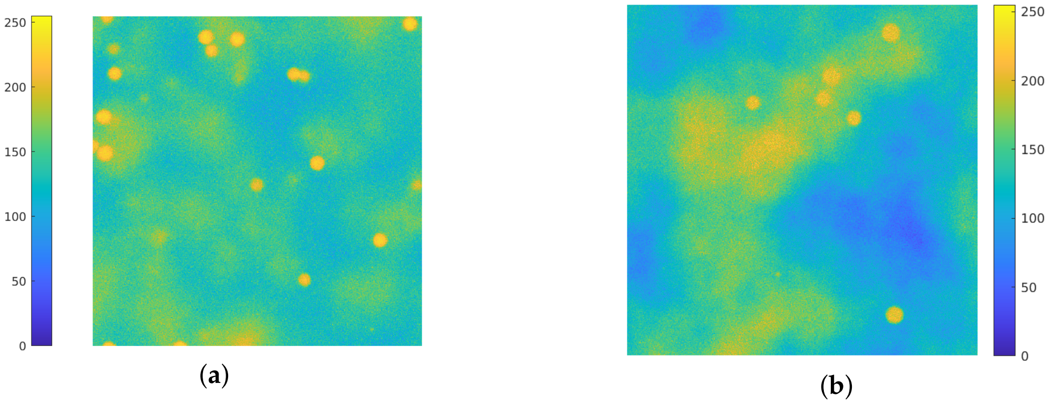


**Figure 1.** (**a**): an actual fluorescence microscopy image, of dimension 512 × 512. (**b**): a synthetic image, where the background fluorescence is simulated via Perlin noise. Both images are displayed with values in $[0, 255]$.

U-net architectures consist of two main structures: the first one has a contracting behavior, and it aims to analyze the input data at different levels. It corresponds to the encoder (or analysis path) of traditional autoencoders (AEs), and in the original work [52] this path was developed for searching for meaningful information for segmentation task. It presents several levels, and each level lowers the dimension of the input: for example, a contracting path with 5 layers may reduce a 256 × 256 image to a 128 × 128 at the first level, then to 64 × 64 at the second and the last later will output an 8 × 8 image. Usually, each stop of the contracting path halves the dimension of the input.

The second part of a U-net consists of the expansion path, or synthesis path, which corresponds to the decoder of an AE. The original aim of this part is to learn localized information for pixel–classification. The layers of this path will increase the size of the output, reaching at its final layer the original dimension. Moreover, each level of the contracting path is connected with the relative layer in the expansion path: this allows us to propagate information through the network. The symmetric u–shape gives the name to this type of network. The interested reader may find complete reviews on U-Nets and their application in medical imaging in [53,54].

Such architectures have been introduced to biomedical imaging in the seminal work [52] and revealed to possess remarkable performances in segmentation tasks [55–57] but also in generating synthetic images when coupled with Generative Adversarial Networks (GANs) [58]. To the best of the author's knowledge, the approach of applying U-net like architectures for removing background emission in microscopy imaging is novel: the main idea is to provide as an input to the neural network a corrupted image, namely **gn** from

Equation (3), and require as output an image cleansed by the background emission, namely $\mathcal{P}(\mathbf{H}\mathbf{x}^\star) + \boldsymbol{\eta}$. Due to the typical behavior of U-net-like architecture and autoencoders, one expects smoothed images as outputs: this is the reason why the neural network is trained with noisy images as output and not with completely clean images.

The rest of the paper is organized as follows. Section 2 depicts the architecture employed for addressing the task of background emission removal and Section 3 provides the details of the synthetic data generation. In Section 4 numerical experiments are carried on to assess the performance of the proposed procedure on images that are blurred by a Gaussian PSF and which are affected by Gaussian, Poisson, and Perlin noise. Moreover, this procedure is tested on actual microscopy images and an application to a practical problem, namely the particle's position estimation, is shown. Section 5 draws the conclusions and presents the future perspectives of this work. The code used in this work is available on https://github.com/AleBenfe/upU-net_Perlin (accessed on 13 May 2022).

## 2. upU-net Architecture

The proposed architecture is a slight modification of classical U-Nets and it is depicted in Figure 2. The main difference is that it is not employed for the classification task, but for Perlin noise removal: Figure 3 shows the original image and the desired output image, where the background emission is not present anymore, while the statistical noise is conserved.

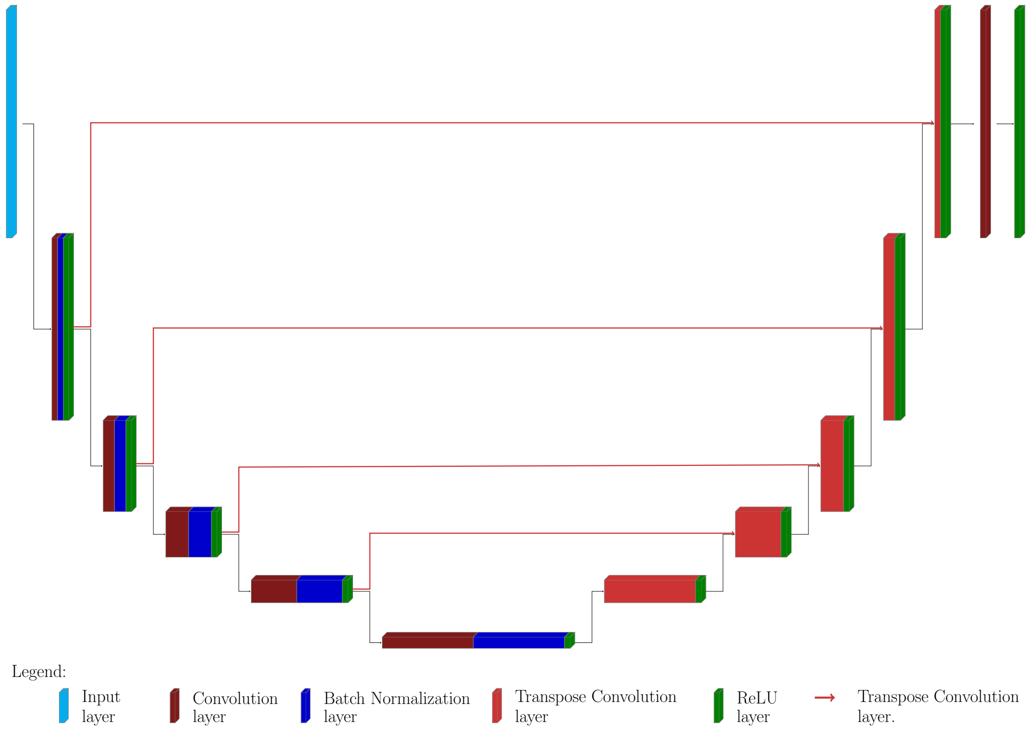

Legend:
Input layer
Convolution layer
Batch Normalization layer
Transpose Convolution layer
ReLU layer
Transpose Convolution layer.

**Figure 2.** The Proposed upU-net architecture. The contracting path consists in several blocks, 5 in the example provided in the Figure. Each block is composed by convolution layer (dark red), batch normalization layer (dark blue) and a Rectified Linear Unit (ReLU) layer (green). The expansive path's blocks have just two parts: up-convolution layer (light red) and a ReLu layer. All the features of the last block are merged via a 1d1 convolution layer, followed by a further ReLu. The red lines are connecting the outputs of each block from the contracting path with the block of the expansive path of higher dimension: this seems to induce a smoother behavior of the net.

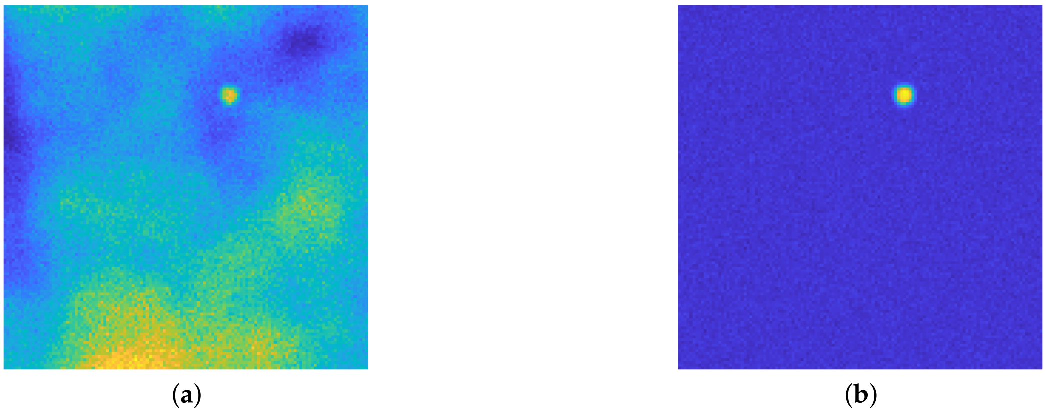


**Figure 3.** (**a**): the acquired image, affected by several artifact due to the blurring operator, Gaussian and Poisson noise and background emission, modeled by Perlin noise. (**b**): the desired output. Both images are displayed with values in $[0,255]$. The aim of the proposed upU-net is to remove only the Perlin noise: the experimental results will show that the typical behavior of such AE–like architecture will remove also the statistical noise. See Section 3 for the details about the generation of such image.

The upU-net presents a contracting and expanding path: each block of the former consists of three different layers:

- A convolution 2D layer, with stride 2 (in both dimensions) and padding 1 (in both dimensions). The boundary conditions are set to symmetric, excluding the image edge.
- A batch normalization layer.
- A ReLU layer.

The dimension of each convolution layer is $3 \times 3 \times 2^{2+l}$, where $l$ is the block level: for example, for a net with 5 blocks, the activation of the convolution layer of the last block is then $3 \times 3 \times 128$.

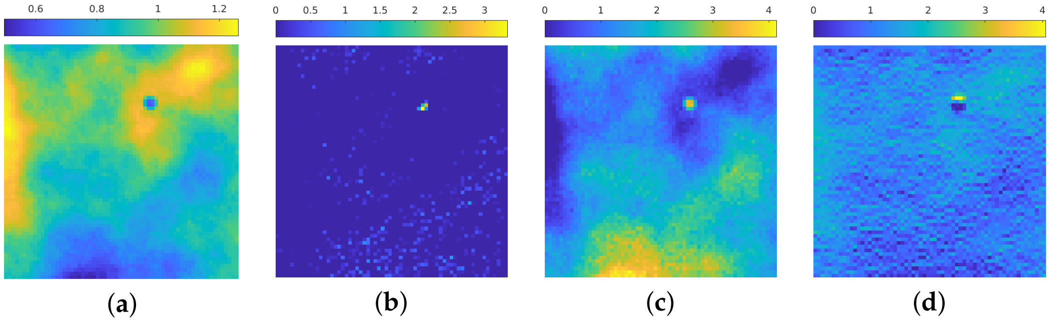
The synthesis path of the upU-net architecture presents the symmetric blocks, each one consists in

- An up–convolution layer. The scope of this type of layer is to increase the size of the input, and the last one will provide data of the same size as the input.
- A ReLU layer.

The actual task of up-convolution layers is to increase the size of the input by performing an interpolation: training the neural network means that such interpolation is not *a–priori* selected (e.g., linear or cubic), but the weights and the bias are learned, tailoring the network to the particular data in consideration. ReLU layers are employed to assure non–negative data through the network: this is a physical requirement, since images (RGB or gray–level ones) have non-negative pixel values. The last two layers consist in a $1 \times 1$ convolution layer, in order to collapse all the filters in the 2D image, and in a final ReLU layer to ensure again non-negative values on the final output.

The major novelty presented in this approach is that the connections between the corresponding blocks of the two paths do not follow the classical scheme. Indeed, the block at level $l$ in the contracting path is connected to the $(l+1)$-th block of the expansive path, and not to the related $l$-th one, and this connection is made via other up–convolution layers (see Figure 2). In this way, the propagation of information is smoother, and the reconstruction will suffer less from the presence of statistical noise, since the up–convolution layers act as smoothing filters.

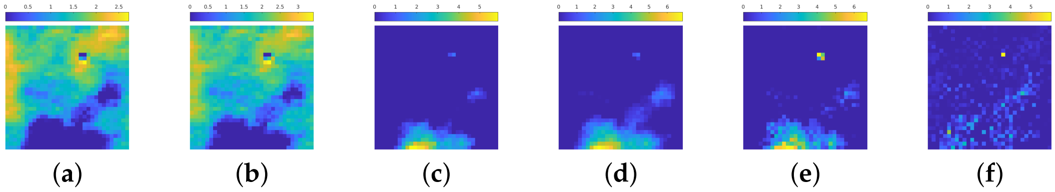
Figure 4 presents a visual inspection of some activations of the net trained in Section 4.1 for each block of the contracting path for the image in Figure 3a: we see that the first blocks aim to discern the areas in which the emission background has more influence, and at the same time can roughly recognize the position of the isolated particles and recognize the presence of statistical noise. In particular, Figure 4a recognizes the areas in the image where the emission background is not present, at the same time Figure 4c shows this activation can find the areas in which such emission is stronger. Figure 4b seems to find the position of the particle, still in a very approximate way; Figure 4d shows that the neural network, at this stage, can find the zones in which the statistical noise is predominant: indeed, the higher activation corresponds to the dark areas in the original image, where Poisson noise is present at a higher level due to low photon count.



(a) (b) (c) (d)

**Figure 4.** (**a**–**d**): 4 out of 8 activation of the 1st block in the contracting path. The images are displayed in different color scale, since the ranges of the activation values differ by a large amount between filters.

This behavior is maintained also in deeper blocks, as confirmed by the inspection of some of the activations of the second block in Figure 5. The net tries to separate the zones in which the emission background is present, while searching for the position of particles, if any, and recognizing the presence of statistical noise.



(a) (b) (c) (d) (e) (f)

**Figure 5.** (**a**–**f**): 6 out of 16 activation of the 2nd block in the contracting path. The color scale is not the same among the images, due to different value ranges among the filters.

## 3. Data Generation

The synthetic images employed in the numerical test of Section 4 are generated as follows.

The position of $p$ particles of radius $r$ is randomly drawn, from an uniform distribution, in a continuous volume if dimension $D_x \times D_y \times D_z$. The procedure checks that there is no intersection between the particles, given the position and the radius. The volume is then discretized, by choosing the resolution of the final image: denote with $n \times n \times n_z$ the number of voxels on each dimension, then each voxel will cover a volume equal to $\mathrm{d}x \times \mathrm{d}y \times \mathrm{d}z$, being $\mathrm{d}x = D_x/n$, $\mathrm{d}y = D_y/n$, $\mathrm{d}z = D_z/n_z$,. The horizontal frames are assumed to be square. Each voxel within a distance less than $r$ is set to 229.5. This particular value comes from the choice of setting the value of the beads equal to 90% of the maximum value registered: since the aim is to simulate tiff images, such value is 255. One horizontal slice is then considered a registered image: this choice has been made for avoiding to have circular profiles with the same radius. Indeed, since the particles have been randomly positioned in a 3D space all the beads' centers will have different $z$ coordinates: this means that considering just one slice of the 3D volume allows to intersect the particles at different heights, thus in a single frame the profiles will have a radius that ranges from 0 (no intersection at all) to 2 µm (maximum intersection at the equator). The steps for particles' simulation are depicted in Figure 6.

A Perlin noise [38] pattern is then added to the image: the generation of this perturbation is pursued by following the improved procedure described in [59,60]. Each instance of the noise is generated by using 8 octaves, an initial frequency of 1, an initial amplitude of 1, and persistence of 0.5. At each octave added after the first, the frequency is doubled and the amplitude is multiplied by the persistence.

A Gaussian PSF with variance $\sigma^2$ is employed to blur the image, for simulating the physical process of image acquisition in fluorescence confocal microscopy.

The last steps regard the statistical noise perturbation:

- Gaussian noise is added following the procedure depicted in [61]: a multivalued random Gaussian variable $\boldsymbol{\eta}$ is generated, then a noise level $\sigma_n$ is chosen. Being $\mathbf{g}$ the noise–free image, the perturbed one is then

$$\mathbf{gn} = \mathbf{g} + \sigma_n \frac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|_{\mathrm{F}}} \|\mathbf{g}\|_{\mathrm{F}} \tag{4}$$

where $\| \cdot \|_{\mathrm{F}}$ denotes the Frobenius norm. In this way, one has that $\|\mathbf{g} - \mathbf{gn}\|_{\mathrm{F}} = \sigma_n$.
- Poisson noise is added, using built-in functions of the software used (see Section 4 for details).

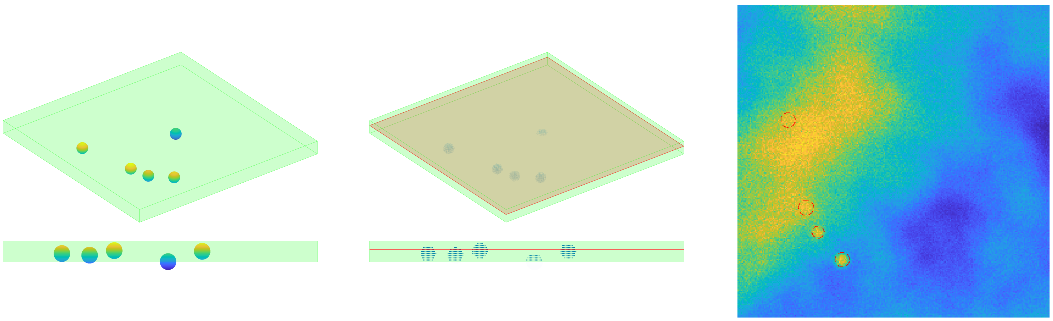

(a) (b) (c)

**Figure 6.** Visual inspection for the synthetic data generation. (**a**): the particles' position is randomly chosen in a continuous space, and the green shaded volume represents the volume taken into consideration. Notice that a particle may not be completely included in the volume, see the bottom part of the panel (**a**). (**b**): once the resolution is chosen, the particles volume is discretized and only the voxels inside the volume are considered. One slice of the whole volume is considered: such slice is perturbed with Perlin noise, then the action of a PSF induces a blur effect and eventually Gaussian and Poisson noise are added. (**c**): the final simulated image, displayed with values in $[0, 255]$. The dashed red lines represent the profile of the particles.

## 4. Results

This section is devoted to assessing the performance of the proposed approach. All the experiments were done on a Linux machine (Ubuntu 20.04) with an Intel(R) Core(TM) i5–8250U CPU (1.60 GHz), 16 GiB RAM (Intel, Santa Clara, CA, USA) and under MATLAB R2022a environment (MathWorks, Natick, MA, USA), employing the Parallel and the Deep Learning toolboxes [62,63]. The code is available at https://github.com/AleBenfe/upU-net_Perlin (accessed on 13 May 2022). The first run of experiments is devoted to testing the proposed architecture on small-size images and a comparison with a more classic U-net architecture is pursued. The third part of this section is aimed to test the net on a different dataset to the one it was trained on, and Sections 4.4 and 4.5 are devoted to employing the reconstruction obtained by the network for particle estimation problems. Finally, the last part shows the achieved results on real images.

### *4.1. Assessing the Performance of upU-net*

The first experiment is carried out on small–sized images: this has been done for providing the proof of concept of the reliable performances of the proposed method. The architecture employed in this experiment has the following structure:

- 5 blocks in the contracting part, each one containing a 2D convolution layer, a batch normalization, and a ReLU layer (see Figure 2). The number of filters in the first block is 8, then it is doubled in each subsequent block.
- 5 blocks in the expansive path, consisting of an up–convolution layer and a ReLU layer. The number of filters starts from 128 and then it is halved, mimicking the structure of the contracting path.
- 5 up-convolution layers that connect the relative blocks in the two paths.
- A final 1D convolution layer for merging all the filtered data, coupled with the last ReLU layer.

The dataset for this experiment contains 500 images: 90% of them are used for the training, 5% for the validation, and the remaining 5% as the test set. All the images are blurred by a Gaussian PSF with $\sigma = 2$ and they are affected by Gaussian noise with $\sigma_n = 0.03$ and by Poisson noise. The training is done by giving an input to the noisy image net (see Figure 3a) and requiring as output Perlin-free images (see Figure 3b). The loss function employed for the training is the least square function. The chosen solver is the ADAM algorithm [64] with default settings, with an initial learning rate of 0.01, a drop factor of 0.2 every 25 epochs with a piecewise constant strategy, a minibatch size of 50 items, and 200 maximum epochs. Moreover, the loss function consists of a regression function and it is augmented with an $\ell_2$ regularization term, balanced by a parameter equal to 0.1.

The dataset for this experiment has been generated under the following setting. A volume of $76 \times 76 \times 5$ µm has been considered, and a random number of particles between 10 and 15 is randomly chosen. The particles' radius is set to 2 µm, and the single horizontal frame has $128 \times 128$ pixels. Each slice is perturbed with Perlin noise, then a Gaussian PSF with variance equal to 2 is used to induce a blur effect on the image. Subsequentely, Gaussian noise at level 3% is added to the image and the MATLAB's function `imnoise` is employed for adding Poisson noise (a scaling of $10^{-12}$ on the input and a subsequent re-scaling by $10^{12}$ is pursued in order to obtain reliable results: the interested reader should check the function documentation for more technical detail about this scaling [65]).

Figure 7 presents the results of the first run of tests on some items of the test dataset: the first row presents the noisy images, the second row the ground truth and the last one shows the reconstruction obtained by the proposed architecture.

All the reconstructed images present a smoother background with respect to the ground truths: this is not surprising, as one expects exactly this kind of behavior when using an autoencoder–like structure, such as the proposed one. It is quite clear that the background emission is completely removed from all the images. Moreover, the blurring effect is still present in the recovered images, since the upU-net is trained on blurred ones.

The first image (Figure 7a) seems to present just one particle: the ground truth in Figure 7f reveals that there are two more particles in the diffuse, yellow area. The upU-net is capable to recognize the presence not only of the most visible bead, but also of the two hidden particles, even if their shape is not completely circular.

The second image (Figure 7b) presents three main issues: one hidden particle in the diffuse area (as in the previous case), one faint particle on the right boundary of the image, and one bead close to the bottom yellow cloud. As one can see in Figure 7l, the upU-Net scheme can recognize the presence of both the hidden and the faint particle, and at the same time it does not confuse the bead at the bottom with the emission background.

Figure 7c does not present particular problems: indeed all the particles are well recognized, but it is evident that a small artifact may arise, even if it is rare. Figure 7n shows again as upU-net is capable to recognize particles close to intense emission, whilst Figure 7o shows that faint particles merged inside low-intensity background emission can be easily recognized.

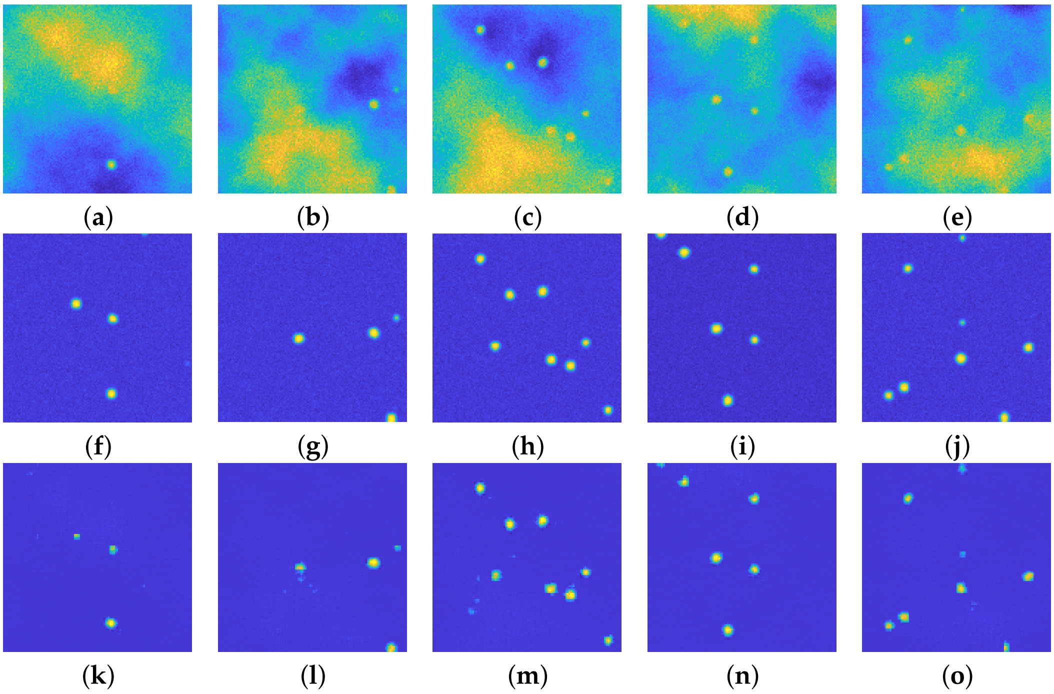


**Figure 7.** (**a**–**e**): Perturbed images. (**f**–**j**): relative ground truths. (**k**–**o**): recovered images via the proposed approach. In each of the 5 example cases, the particles are well recognized, even the faint ones. Small artifacts arise (see (**l**,**m**) due to the presence of background emission of high intensity. All the images are displayed with values in $[0, 255]$.

### *4.2. Comparison with Classical U-net*

This section is devoted to assessing the difference in performance between the proposed approach and a more classical U-net architecture.

A U-net architecture with 3 layers of depth is firstly generated by using the `unet` MATLAB command: a slight modification of the output of this function is pursued, such modification consists in inserting a batch normalization layer between the ReLU layer and the successive convolution one. This addition is required otherwise the network, in its original form, is not able to learn any parameters and outputs only zeros images. The encoder's structure consists of 3 blocks, each one containing

- a convolution layer
- a ReLU layer
- a batch normalization layer
- a convolution layer
- a ReLU layer
- a batch normalization layer
- a max-pooling layer

The deepest level of the contracting part presents a dropout layer with a probability of dropout of 0.5, before the pooling. The decoder's part consists in

- an upconvolution layer
- a ReLU layer
- a concatenation layer, for connecting with the relative encoding part
- a convolution layer
- a ReLU layer
- a batch normalization layer
- a convolution layer
- a ReLU layer
- a batch normalization layer

The bridge structure of the U-net has the same structure as an encoder's block, i.e., 2× (a convolution layer + ReLU + batch normalization), without the max pooling operation. The initial number of $3 \times 3$ filters is 8.

Such network is trained on the $128 \times 128$ image dataset, the choice of 3 levels of depth is done to have the same number of learnable parameters, with respect to the upU-net used in Section 4.1: approximately 295,500 the former, approximately 121,300 the latter. The first difference between the proposed approach and the U-net is the training time: the upU-net required around 15 min whilst the U-net required around 55 min, despite having a low number of learnables. In terms of achieved results, the U-net introduces artifacts in some of the recovered images, as evident in Figure 8.

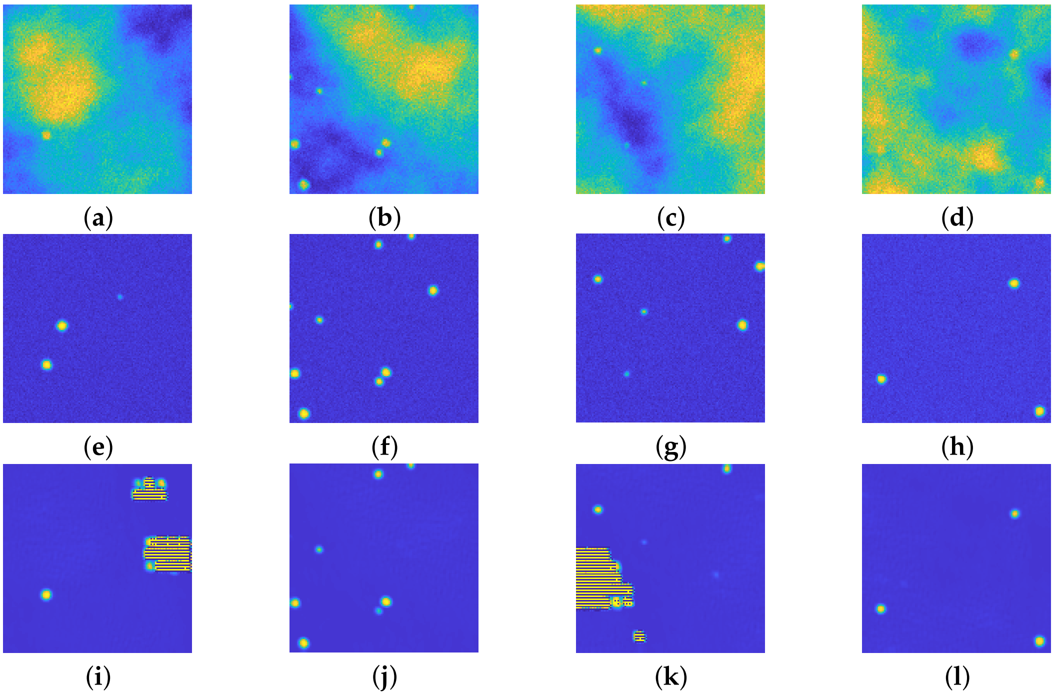


**Figure 8.** (**a**–**d**): Synthetic images. (**e**–**h**): ground truths. (**i**–**l**): recovered images via the U-net. This network is able to recover some of the images, but unfortunately it introduces some undesired artifacts. All the images are displayed with values in $[0, 255]$.

The higher computational time required for training (despite its lower number of parameters to be learned) and the rising of such artifacts in the recovered images show that the proposed upU-net is a valid competitor of classical U-net approaches.

### *4.3. upU-net for Different Radius Particles*

In this section, another upU-net is trained for recovering images of size $256 \times 256$: the training parameters are the same as the net studied in Section 4.1, whilst the structure of the net consists of 4 blocks instead of 5. The training dataset has the same structure as the one used in Section 4.1, the sole difference is the dimension of the images. The training time was around 40 min.

Another dataset is then created, where the number of particles ranges from 0 to 15 and where their radius lies in the interval $[0.5, 4]$ µm. Figure 9 presents the visual inspection of the achieved results: even if the net has been trained in images generated by volumes with particles of the same radius, it can reconstruct beads of different radii. This is due to the fact that each image of the training dataset is generated by considering a frame intersecting the volume at a fixed height, hence the registered profiles have a different radius (see Figure 6 for a detailed visual inspection).

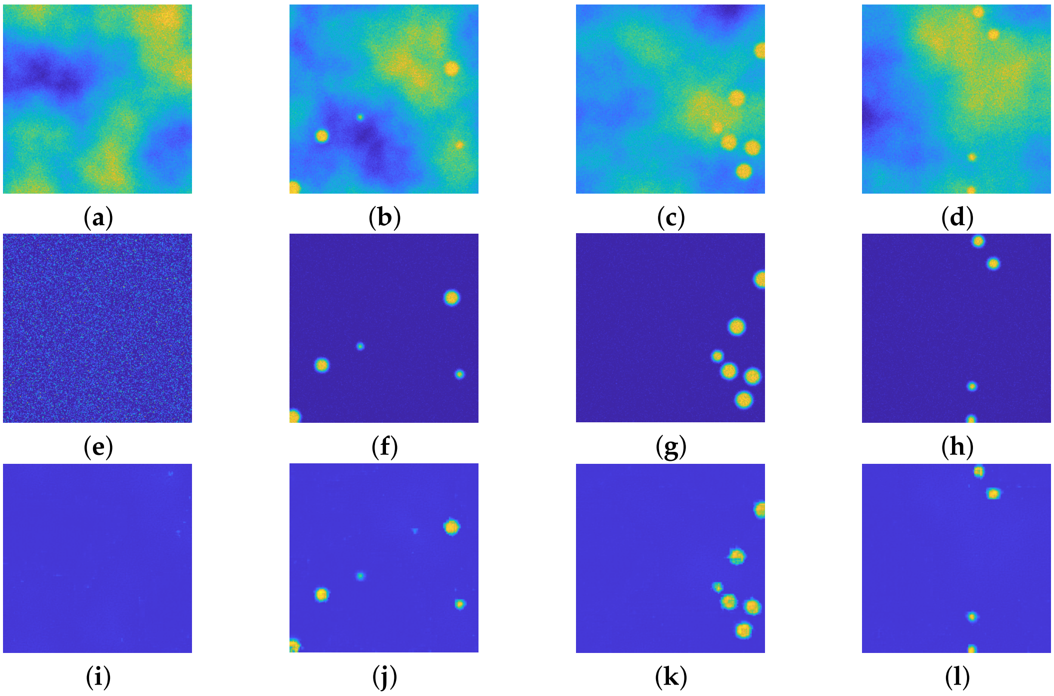


**Figure 9.** (**a**–**d**): Synthetic images with beads with radius in $[0.5, 4]$ µm. (**e**–**h**): ground truths. (**i**–**l**): recovered images. he upU-net is able to reconstruct images with particles with different radius with respect with its training. All the images are displayed with values in $[0, 255]$.

### *4.4. upU-net for Particle Estimation Task*

This last run of experiments refers to the estimation of the particle position when the background emission heavily affects this task.

Beside the upU-Nets presented in Sections 4.1 and 4.3, a further one is trained for recovering 512 × 512–size images. The training parameters are the same used for the previous nets, the sole change regards the number of epochs, fixed to 60: this leads to a training time of about one hour. Moreover, further 100 synthetic images with dimensions of 128, 256, 512 are generated, slightly different from the ones used for training and validation. In this new dataset the Perlin noise has a large influence on the final registered data. The task of recognizing the particles' center is pursued by employing the algorithm depicted in [66], slightly modified and tailored for handling 2D images. Figure 10 shows the potentiality of the proposed procedure: particles embedded in large portions with background emissions with high intensities are easily recognized. The employed procedure, moreover, allow us to estimate both the center and the radius in the frame. Due to some distortion provoked by the reconstruction of the upU-net, the exact position is not completely recovered: anyway, there is a clear reckoning of the presence of particles even in difficult imaging situations.

The procedure depicted in [66] allows also to assess the performance of the upU-net approach. Two measurements are employed for each image: the True Positive Ratio (TPR)

$$\text{TPR} = \frac{\#\{E_p \cap P\}}{\#\{P\}} \tag{5}$$

where $E_p$ is the set of estimated particles and $P$ is the set of actual particles present in the image. TPR measures the fraction of the recognized particles that are actually in the frame in examination. Given a set $A$, $\#A$ stands for the number of elements of the set $A$. The second measurement is the number of false particles

$$\varphi = \frac{1}{N} \sum_{i=1}^{N} \#\{F_i\} \tag{6}$$

where $F_i$ is the set representing the false particles recognized in the $i$-th frame. The quantity in Equation (6) represents then the average number of recognized particles that are not actually present in the dataset, and $N$ is the number of images in the dataset. Table 1 show the performance of the nets.

**Table 1.** Performance assessment of the proposed procedure in view of particle estimation task. TPR is the true positive ration depicted in (5), while $\varphi$ is the average number of particles erroneously recognized (see (6)).

| Image Size | TPR | $\varphi$ |
|---|---|---|
| 128 × 128 | 90.87% | 0.1 |
| 256 × 256 | 90.19% | 0.19 |
| 512 × 512 | 92.29% | 0.69 |

The average TPR on all datasets (100 images each) is greater than 90%, whilst the average number of particles erroneously recognized is less than one in each test case: this means that the total number of particles is slightly underestimated, and it is very rare that the artifacts are recognized as beads in the image.

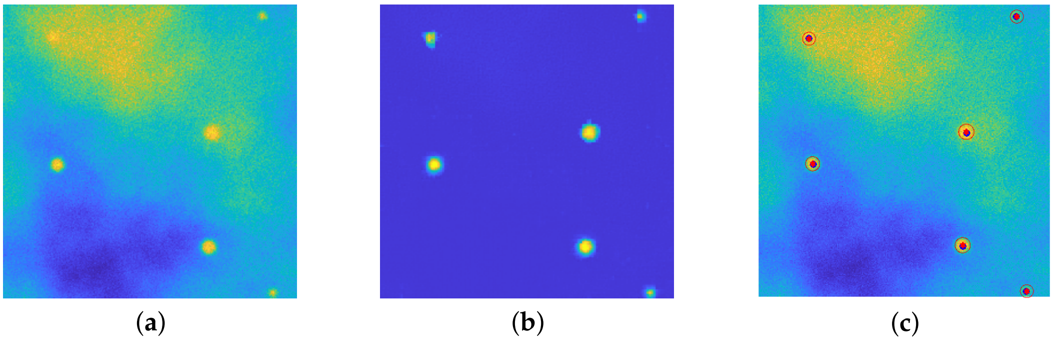


**Figure 10.** Particle Estimation task. (**a**): original image. (**b**): recovered image by upU-net. (**c**): superposition of the true particles' center (blue dots), of the estimated center (red dots) and of the estimated profile (red lines). The values range in the interval $[0, 255]$. In this situation one has several particles that are easily recognizable, a faint one (in the top right corner) and a last bead inside a diffuse emission area (top left corner.)

### 4.5. Volume Reconstruction

A further experiment considers the reconstruction of the particles' position in 3D space. A volume of 76 × 76 × 6 is discretized in an array of 512 × 512 × 22 voxels, and several particles are randomly put in such volume. Each registered frame is processed by the appropriate upU-net, the procedure can be summarized in two steps:

1. Apply to each frame the up–UNet, in order to remove the noise artifacts.
2. Apply the procedure in [66] to the restored array.

Figure 11 provides a three–dimensional visual inspection of the same procedure applied to one reconstruction of the particle distribution in space, with a superposition of some of the layers of the 3D image.

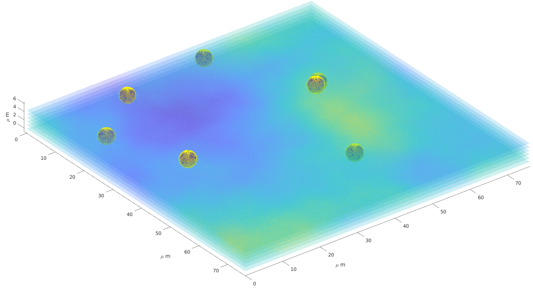
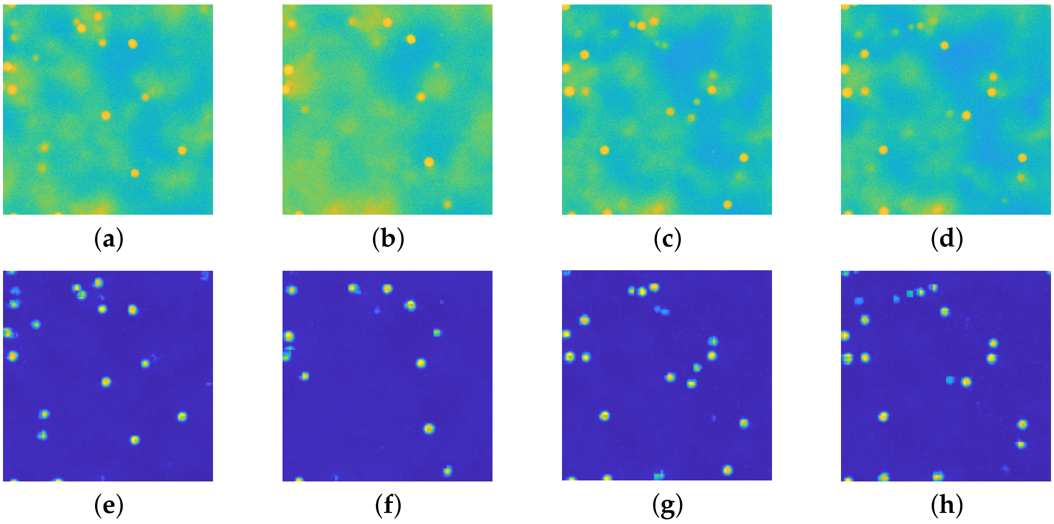


**Figure 11.** Three–dimension reconstruction of particles distribution. Six layers (out of 22) are superimposed to the sphere representing the particles.

### *4.6. Real Dataset*

This section is devoted to presenting the performances of the proposed architecture on real images. The microscopy images refer to a 3D volume with dimension $D_x = D_y = 64$ μm, $D_z \ = \ 4.1$ μm which has been discretized in a 3D array of $512 \times 512 \times 10$ voxels. The radius of the beads is 1.5 μm and they are suspended in a ∼70–30% glycerol/water mixture (viscosity of approximately 0.017 Pa s). The microscope used to acquire this data is aZeiss LSM 700 with a 100 × NA 1.4 oil immersion objective (Zeiss Plan–APOCHROMAT) (ZEISS, Oberkochen, Germany). The volume has been scanned 50 times, and the acquisition time for each frame is 0.5 s. A visual inspection of the obtained results is in Figure 12: the background fluorescence emission has been successfully removed, the beads' profiles are evident in the recovered images. This experiment provides empirical proof that Perlin noise does simulate in a reliable way the fluorescence emission, even if its connection with the actual physical phenomenon is faint.


(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

**Figure 12.** (**a**–**d**): registered images displayed in $[0, 255]$. (**e**–**h**): recovered images via the upU-net architecture displayed in $[0, 255]$. The fluorescence has been successfully removed and the particle position is evident from the restored images.

## 5. Conclusions

This work presents the novel application of a U-net–like architecture for removing background emissions from fluorescence confocal microscopy images. The background emission is simulated by Perlin noise, which has a high degree of similarity with the actual emission. Moreover, the images were perturbed with Gaussian and Poisson noise, after having been blurred via a Gaussian PSF. Such realistic images, with different dimensions, were then used for the training and the test of the upU-net architecture: the visual inspection of the results of both experiments presents a remarkable performance, showing that the proposed approach efficiently removes the background emission and at the same time it exploits the typical behavior of autoencoders, i.e., the recovered images present a more smooth appearance and the both Gaussian and Poisson noise are removed. A comparison with classical U-net approaches shows that the proposed strategy overcomes the classical one, due to its robustness, its performance, and its lower computational time for reaching convergence. The proposed procedure is then coupled with a particle estimation algorithm: the numerical experience showed that the performance of reckoning the beads in a microscopy image is greater than 90% and moreover there is a really low probability that artifacts are mistaken for actual particles. Finally, the procedure is applied to real images. This last experiment provides two main pieces of evidence: Perlin noise is a valid candidate for simulating the background fluorescence noise and that the upU-net can remove such kind of perturbation also on real microscopy images.

This work may pave the way for several new approaches. On one hand, the Perlin noise can be investigated more for the simulation of the fluorescence and the background emission, by encompassing some physical properties in the generation of the noise, such as the presence of the particles. Moreover, instead of considering a simple 2D frame, 3D data can be easily generated, employing the same procedure for the simulation of 3D Perlin noise (see Section 4.5) and convolution techniques for 3D PSFs. A further investigation of the theoretical aspect regarding the reasons why this particular architecture provides such good results will be the object of future work. The neural network architecture can be furthermore employed in more general imaging problems: for example, the encoder part can be employed as the regularization term [67] in a variational framework, or the recovered images ease a segmentation task [68].

**Funding:** This research received support from the PRECISION project, funded by the University of Milan.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The code employed for the numerical experiments can be found at https://github.com/AleBenfe/upU-net_Perlin (accessed on 21 April 2022).

**Acknowledgments:** The author would like to thank H. Talbot, T. Bourouina, F. Bonacci and Labex MMCD, (Université Paris-Est Creteil, Université Gustave Eiffel, École des Ponts Paris Tech) for the microscopy images employed in Section 4.6.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Bertero, M.; Boccacci, P.; Desiderà, G.; Vicidomini, G. Image deblurring with Poisson data: From cells to galaxies. *Inverse Probl.* **2009**, *25*, 123006. [CrossRef]
2. Bertero, M.; Boccacci, P.; Ruggiero, V. *Inverse Imaging with Poisson Data*; IOP Publishing: Bristol, UK, 2018; pp. 2053–2563.
3. Bechensteen, A.; Rebegoldi, S.; Aubert, G.; Blanc-Féraud, L. $\ell 2 - \ell 0$ optimization for single molecule localization microscopy. In *Proceedings of the Imaging and Applied Optics 2018 (3D, AO, AIO, COSI, DH, IS, LACSEA, LS&C, MATH, pcAOP), Orlando, FL, USA*; Optica Publishing Group: Orlando, FL, USA, 2018; p. MW2D.1.
4. Benfenati, A.; Ruggiero, V. Image regularization for Poisson data. *J. Phys. Conf. Ser.* **2015**, *657*, 012011. [CrossRef]
5. di Serafino, D.; Landi, G.; Viola, M. Directional TGV-Based Image Restoration under Poisson Noise. *J. Imaging* **2021**, *7*, 99. [CrossRef]
6. Pragliola, M.; Calatroni, L.; Lanza, A.; Sgallari, F. Residual Whiteness Principle for Automatic Parameter Selection in $\ell_2 - \ell_2$ Image Super-Resolution Problems. In Proceedings of the International Conference on Scale Space and Variational Methods in Computer Vision, Lège-Cap Ferret, France, 16–20 May 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 476–488.
7. Benfenati, A.; Chouzenoux, E.; Pesquet, J.C. Proximal approaches for matrix optimization problems: Application to robust precision matrix estimation. *Signal Process.* **2020**, *169*, 107417. [CrossRef]
8. Beck, A.; Teboulle, M. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [CrossRef]
9. Bonettini, S.; Benfenati, A.; Ruggiero, V. Primal-dual first order methods for total variation image restoration in presence of poisson noise. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 4156–4160.

10. Cascarano, P.; Sebastiani, A.; Comes, M.C.; Franchini, G.; Porta, F. Combining Weighted Total Variation and Deep Image Prior for natural and medical image restoration via ADMM. In Proceedings of the 2021 21st International Conference on Computational Science and Its Applications (ICCSA), Cagliari, Italy, 13–16 September 2021; pp. 39–46.
11. Nasonov, A.; Krylov, A. An Improvement of BM3D Image Denoising and Deblurring Algorithm by Generalized Total Variation. In Proceedings of the 2018 7th European Workshop on Visual Information Processing (EUVIP), Tampere, Finland, 26–28 November 2018; pp. 1–4.
12. Benfenati, A.; Camera, A.L.; Carbillet, M. Deconvolution of post-adaptive optics images of faint circumstellar environments by means of the inexact Bregman procedure. *Astron. Astrophys.* **2016**, *586*, A16. [CrossRef]
13. Benfenati, A.; Causin, P.; Lupieri, M.G.; Naldi, G. Regularization Techniques for Inverse Problem in DOT Applications. *J. Phys. Conf. Ser.* **2020**, *1476*, 012007. [CrossRef]
14. Zhang, M.; Desrosiers, C. High-quality Image Restoration Using Low-Rank Patch Regularization and Global Structure Sparsity. *IEEE Trans. Image Process.* **2019**, *28*, 868–879. [CrossRef]
15. Bevilacqua, F.; Lanza, A.; Pragliola, M.; Sgallari, F. Nearly Exact Discrepancy Principle for Low-Count Poisson Image Restoration. *J. Imaging* **2022**, *8*, 1. [CrossRef]
16. Zanni, L.; Benfenati, A.; Bertero, M.; Ruggiero, V. Numerical Methods for Parameter Estimation in Poisson Data Inversion. *J. Math. Imaging Vis.* **2015**, *52*, 397–413. [CrossRef]
17. Benfenati, A.; Ruggiero, V. Inexact Bregman iteration with an application to Poisson data reconstruction. *Inverse Probl.* **2013**, *29*, 065016. [CrossRef]
18. Benfenati, A.; Ruggiero, V. Inexact Bregman iteration for deconvolution of superimposed extended and point sources. *Commun. Nonlinear Sci. Numer. Simul.* **2015**, *20*, 882–896. [CrossRef]
19. Cadoni, S.; Chouzenoux, E.; Pesquet, J.C.; Chaux, C. A block parallel majorize-minimize memory gradient algorithm. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3194–3198.
20. Cascarano, P.; Comes, M.C.; Mencattini, A.; Parrini, M.C.; Piccolomini, E.L.; Martinelli, E. Recursive Deep Prior Video: A super resolution algorithm for time-lapse microscopy of organ-on-chip experiments. *Med Image Anal.* **2021**, *72*, 102124. [CrossRef]
21. Liu, J.; Sun, Y.; Xu, X.; Kamilov, U.S. Image Restoration Using Total Variation Regularized Deep Image Prior. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 7715–7719.
22. Cascarano, P.; Piccolomini, E.L.; Morotti, E.; Sebastiani, A. Plug-and-Play gradient-based denoisers applied to CT image enhancement. *Appl. Math. Comput.* **2022**, *422*, 126967. [CrossRef]
23. Puybareau, É.; Carlinet, E.; Benfenati, A.; Talbot, H. Spherical Fluorescent Particle Segmentation and Tracking in 3D Confocal Microscopy. In Proceedings of the Mathematical Morphology and Its Applications to Signal and Image Processing, Fontainebleau, France, May 15–17; Burgeth, B., Kleefeld, A., Naegel, B., Passat, N., Perret, B., Eds.; Springer International Publishing: Cham, Swizerland, 2019; pp. 520–531.
24. Josephson, L.L.; Swan, J.W.; Furst, E.M. In situ measurement of localization error in particle tracking microrheology. *Rheol. Acta* **2018**, *57*, 793–800. [CrossRef]
25. Chu, K.K.; Mojahed, D.; Fernandez, C.M.; Li, Y.; Liu, L.; Wilsterman, E.J.; Diephuis, B.; Birket, S.E.; Bowers, H.; Solomon, G.M.; et al. Particle-Tracking Microrheology Using Micro-Optical Coherence Tomography. *Biophys. J.* **2016**, *111*, 1053–1063. [CrossRef]
26. Godinez, W.J.; Rohr, K. Tracking Multiple Particles in Fluorescence Time-Lapse Microscopy Images via Probabilistic Data Association. *IEEE Trans. Med. Imaging* **2015**, *34*, 415–432. [CrossRef]
27. Gardill, A.; Kemeny, I.; Li, Y.; Zahedian, M.; Cambria, M.C.; Xu, X.; Lordi, V.; Gali, A.; Maze, J.R.; Choy, J.T.; et al. Super-Resolution Airy Disk Microscopy of Individual Color Centers in Diamond. *arXiv* **2022**, arXiv:2203.11859.
28. Richter, V.; Piper, M.; Wagner, M.; Schneckenburger, H. Increasing Resolution in Live Cell Microscopy by Structured Illumination (SIM). *Appl. Sci.* **2019**, *9*, 1188. [CrossRef]
29. Kim, B. DVDeconv: An Open-Source MATLAB Toolbox for Depth-Variant Asymmetric Deconvolution of Fluorescence Micrographs. *Cells* **2021**, *10*, 397. [CrossRef]
30. Jezierska, A.; Chouzenoux, E.; Pesquet, J.C.; Talbot, H. A primal-dual proximal splitting approach for restoring data corrupted with poisson-gaussian noise. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 1085–1088.
31. Novák, T.; Gajdos, T.; Sinkó, J.; Szabó, G.; Erdélyi, M. TestSTORM: Versatile simulator software for multimodal super-resolution localization fluorescence microscopy. *Sci. Rep.* **2017**, *7*, 951. [CrossRef]
32. Flier, B.M.; Baier, M.; Huber, J.; Müllen, K.; Mecking, S.; Zumbusch, A.; Wöll, D. Single molecule fluorescence microscopy investigations on heterogeneity of translational diffusion in thin polymer films. *Phys. Chem. Chem. Phys.* **2011**, *13*, 1770–1775. [CrossRef]
33. Abdellah, M.; Bilgili, A.; Eilemann, S.; Markram, H.; Schürmann, F. A Computational Model of Light-Sheet Fluorescence Microscopy using Physically-based Rendering. In Proceedings of the Eurographics (Posters), Los Angeles, CA, USA, 7–9 August 2015; pp. 15–16.
34. Carles, G.; Zammit, P.; Harvey, A.R. Holistic Monte-Carlo optical modelling of biological imaging. *Sci. Rep.* **2019**, *9*, 15832. [CrossRef]

35. Lu, X.; Rodenko, O.; Zhang, Y.; Gross, H. Efficient simulation of autofluorescence effects in microscope lenses. *Appl. Opt.* **2019**, *58*, 3589–3596. [CrossRef]
36. Mukamel, E.; Babcock, H.; Zhuang, X. Statistical Deconvolution for Superresolution Fluorescence Microscopy. *Biophys. J.* **2012**, *102*, 2391–2400. [CrossRef]
37. Axelrod, D. Evanescent Excitation and Emission in Fluorescence Microscopy. *Biophys. J.* **2013**, *104*, 1401–1409. [CrossRef]
38. Perlin, K. An image synthesizer. *ACM Siggraph Comput. Graph.* **1985**, *19*, 287–296. [CrossRef]
39. Meraner, A.; Ebel, P.; Zhu, X.X.; Schmitt, M. Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR-optical data fusion. *ISPRS J. Photogramm. Remote Sens.* **2020**, *166*, 333–346. [CrossRef]
40. Lee, K.Y.; Sim, J.Y. Cloud Removal of Satellite Images Using Convolutional Neural Network With Reliable Cloudy Image Synthesis Model. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 3581–3585.
41. Enomoto, K.; Sakurada, K.; Wang, W.; Fukui, H.; Matsuoka, M.; Nakamura, R.; Kawaguchi, N. Filmy Cloud Removal on Satellite Imagery with Multispectral Conditional Generative Adversarial Nets. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1533–1541.
42. Štěpka, K.; Matula, P.; Matula, P.; Wörz, S.; Rohr, K.; Kozubek, M. Performance and sensitivity evaluation of 3D spot detection methods in confocal microscopy. *Cytom. Part A* **2015**, *87*, 759–772. [CrossRef]
43. Rolfe, D.J.; McLachlan, C.I.; Hirsch, M.; Needham, S.R.; Tynan, C.J.; Webb, S.E.; Martin-Fernandez, M.L.; Hobson, M.P. Automated multidimensional single molecule fluorescence microscopy feature detection and tracking. *Eur. Biophys. J.* **2011**, *40*, 1167–1186. [CrossRef]
44. Woelk, L.M.; Kannabiran , S.A.; Brock , V.J.; Gee, C.E.; Lohr, C.; Guse, A.H.; Diercks, B.P.; Werner, R. Time-Dependent Image Restoration of Low-SNR Live-Cell Ca2 Fluorescence Microscopy Data. *Int. J. Mol. Sci.* **2021**, *22*, 11792. [CrossRef]
45. Abdolhoseini, M.; Kluge, M.G.; Walker, F.R.; Johnson, S.J. Neuron Image Synthesizer Via Gaussian Mixture Model and Perlin Noise. In Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019; pp. 530–533.
46. Sorokin, D.V.; Peterlík, I.; Ulman, V.; Svoboda, D.; Nečasová, T.; Morgaenko, K.; Eiselleová, L.; Tesařová, L.; Maška, M. FiloGen: A Model-Based Generator of Synthetic 3-D Time-Lapse Sequences of Single Motile Cells With Growing and Branching Filopodia. *IEEE Trans. Med. Imaging* **2018**, *37*, 2630–2641. [CrossRef]
47. Sorokin, D.V.; Peterlík, I.; Ulman, V.; Svoboda, D.; Maška, M. Model-based generation of synthetic 3D time-lapse sequences of motile cells with growing filopodia. In Proceedings of the 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), Melbourne, VIC, Australia, 18–21 April 2017; pp. 822–826.
48. Ghaye, J.; De Micheli, G.; Carrara, S. Simulated biological cells for receptor counting in fluorescence imaging. *BioNanoScience* **2012**, *2*, 94–103. [CrossRef]
49. Malm, P.; Brun, A.; Bengtsson, E. Papsynth: Simulated bright-field images of cervical smears. In Proceedings of the 2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Rotterdam, The Netherlands, 14–17 April 2010; pp. 117–120.
50. Möckl, L.; Roy, A.R.; Petrov, P.N.; Moerner, W. Accurate and rapid background estimation in single-molecule localization microscopy using the deep neural network BGnet. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 60–67. [CrossRef]
51. Möckl, L.; Roy, A.R.; Moerner, W.E. Deep learning in single-molecule microscopy: Fundamentals, caveats, and recent developments. *Biomed. Opt. Express* **2020**, *11*, 1633–1661. [CrossRef]
52. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Swizerland, 2015; pp. 234–241.
53. Siddique, N.; Paheding, S.; Elkin, C.P.; Devabhaktuni, V. U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. *IEEE Access* **2021**, *9*, 82031–82057. [CrossRef]
54. Punn, N.S.; Agarwal, S. Modality specific U-Net variants for biomedical image segmentation: A survey. *Artif. Intell. Rev.* **2022**, *2022*, 1–45. [CrossRef]
55. Long, F. Microscopy cell nuclei segmentation with enhanced U-Net. *BMC Bioinform.* **2020**, *21*, 8. [CrossRef]
56. Ibtehaz, N.; Rahman, M.S. MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation. *Neural Netw.* **2020**, *121*, 74–87. [CrossRef]
57. Tran, S.T.; Cheng, C.H.; Nguyen, T.T.; Le, M.H.; Liu, D.G. TMD-Unet: Triple-Unet with Multi-Scale Input Features and Dense Skip Connection for Medical Image Segmentation. *Healthcare* **2021**, *9*, 54. [CrossRef]
58. Fu, C.; Lee, S.; Joon Ho, D.; Han, S.; Salama, P.; Dunn, K.W.; Delp, E.J. Three Dimensional Fluorescence Microscopy Image Synthesis and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, 18–23 June 2018.
59. Perlin, K. Improving Noise. *ACM Trans. Graph.* **2002**, *21*, 681–682. [CrossRef]
60. Lagae, A.; Lefebvre, S.; Cook, R.; DeRose, T.; Drettakis, G.; Ebert, D.S.; Lewis, J.P.; Perlin, K.; Zwicker, M. State of the Art in Procedural Noise Functions. In *Proceedings of the EG 2010–State of the Art Reports*; Hauser, H., Reinhard, E., Eds.; Eurographics, Eurographics Association: Norrkoping, Sweden, 2010.

61. Hansen, P.; Nagy, J.; O'Leary, D. *Deblurring Images: Matrices, Spectra, and Filtering*; Fundamentals of Algorithms, Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2006.
62. MatLab. 2022. Available online: https://it.mathworks.com/products/parallel-computing.html (accessed on 13 May 2022).
63. MatLab. 2022. Available online: https://www.mathworks.com/products/deep-learning.html (accessed on 13 May 2022).
64. Bock, S.; Weiß, M. A Proof of Local Convergence for the Adam Optimizer. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
65. MatLab. Imnoise Documentation. 2022. Available online: https://it.mathworks.com/help/images/ref/imnoise.html (accessed on 13 May 2022).
66. Benfenati, A.; Bonacci, F.; Bourouina, T.; Talbot, H. Efficient Position Estimation of 3D Fluorescent Spherical Beads in Confocal Microscopy via Poisson Denoising. *J. Math. Imaging Vis.* **2021**, *63*, 56–72. [CrossRef]
67. Li, H.; Schwab, J.; Antholzer, S.; Haltmeier, M. NETT: Solving inverse problems with deep neural networks. *Inverse Probl.* **2020**, *36*, 065005. [CrossRef]
68. Aletti, G.; Benfenati, A.; Naldi, G. A Semiautomatic Multi-Label Color Image Segmentation Coupling Dirichlet Problem and Colour Distances. *J. Imaging* **2021**, *7*, 208. [CrossRef]