Localizing Fluorescent Proteins Using Super-Resolution Neural Networks

Kyle Yee
Swarthmore College
kyee1@swarthmore.edu

Guy Hagen
University of Colorado
Colorado Springs
ghagen@uccs.edu

Jonathan Ventura
University of Colorado
Colorado Springs
jventura@uccs.edu

***Abstract*—Recent advances in microscopy and data analysis have allowed for the resolution of photoactivatable fluorescent protein (PA-FP) samples past the theoretical diffraction limit of 200 nm. To do this, several techniques have been developed which perform well on low density protein samples, but which have more difficulty resolving high density PA-FP images. This project seeks to super-resolve PA-FP samples using Convolutional Neural Networks (CNNs) by combining super-resolution and localization techniques. This implementation achieves good results on existing contest datasets and acts as a generalizable model to other protein samples. Notably, the neural network significantly outperforms other easily-accessible algorithms. Results from these experiments suggest that convolutional neural networks are a very promising method for single-molecule localization in a wide array of situations, with evaluation times much faster than existing methods.**

***Keywords*—Computational and artificial intelligence, Neural networks, Image processing, Machine vision, Object recognition**

I. Introduction

In conventional optics, the resolution achievable by an optical instrument is limited by the effects of light diffraction at small scales. In microscopy, this limit occurs around 200 nm, where objects or features smaller than this scale are not resolvable by the instrument alone. However, Betzig et al. [1] introduced a new technique for super-resolving images of photoactivatable fluorescent proteins (PA-FPs). These proteins are on the order of 10 nm and can be made to fluoresce at random intervals through exposure to laser light. The density of proteins flashing at one time can be varied by changing the frequency of laser pulses. By recording a signal from active PA-FPs, techniques such as PALM [1] and STORM [2] fit these signals with a single point-spread function (PSF). Through this process, PALM and STORM achieve good results in locating the position of a given PA-FP up to 20 nm, a factor of ten past the diffraction limit.

Despite this success of PALM and STORM, these methods have difficulty resolving PA-FP high-density fluorescence signals where diffraction patters occur simultaneously in close proximity. Instead, these techniques are limited to "sparse fields" where the majority of the sample is inactive at a given time [1]. However, when limited to sparse fields, sampling times must be long (on the order of 10 seconds) in order to capture an occurrence of each individual protein fluorescence, and such a limitation poses challenges to imaging living samples. Currently, there is an active area of research on developing techniques which can handle these high-density situations in the hope of decreasing the imaging time required for individual PA-FP samples. With this in mind, this research project seeks to use CNNs to resolve microscopic images through a machine learning approach, which has not yet been thoroughly explored.

II. Related Work

There are many algorithms which seek to accomplish the task of high-density localization microscopy. While none of the current leading algorithms use machine learning techniques, there are quite a few which improve on the original PALM and STORM methods. Some notable techniques are listed below.

Holden et al. [4] introduce DAOSTORM, an improvement on single PSF methods which is capable of fitting multiple PSFs to locations of high-density PA-FP signals. By doing this, DAOSTORM processes high-density signals with better precision than STORM and PALM methods. DAOSTORM achieves some of the best results to date on localization tasks involving high-density data.

Zhu et al. [5] build Compressed Sensing STORM (CSSTORM), which achieves higher density results than DAOSTORM and improves sampling time to 3 seconds. CSSTORM models PSFs as linear transformations on protein position data, and divides the output space into grid locations as small as one-eighth pixel size.

In the same year, Mukamel et al. [6] create deconvolutional STORM (deconSTORM). This technique also treats PSFs as reversible transformations on an original image, referred to as convolutions (distinct from convolutional neural networks). However, Mukamel et al. introduce non-linearity to this transformation to achieve good high-density results. By performing deconvolutions, deconSTORM is able to reconstruct super-resolution images.

In 2014, Ovensky et al. introduce ThunderSTORM [7], an ImageJ plugin which acts as a culmination of several different methods used to super-resolve proteins. ThunderSTORM has high performance in a variety of localization tasks, and is extremely accessible. Along with performing analysis, ThunderSTORM includes a realistic simulator of PA-FP data as well as an evaluator to analyze the performance of other methods.

Most recently, Min et al. [8] design the FAst Localization algorithm based on a CONtinuous-space formulation (FALCON). As opposed to previous methods, FALCON achieves continuous output space by fitting PA-FP signals using Taylor

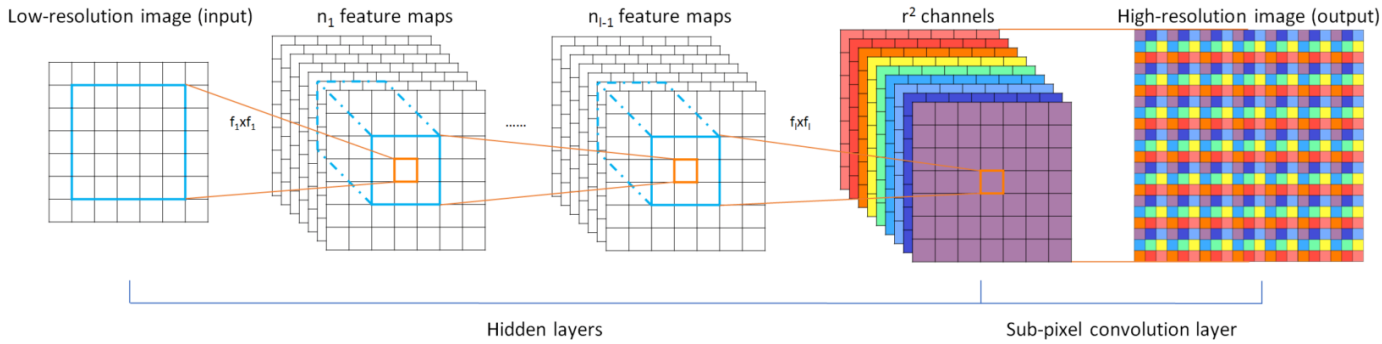

Fig. 1: Efficient implementation of the subpixel convolution layer (image taken without permission from Shi et al. [3]). This diagram shows two initial convolution layers followed by the subpixel convolution layer, which outputs a single-channel image by rearranging the channels in the subpixel layer to increase resolution.

approximations of PSFs. Thus, as compared to other methods, FALCON is able to achieve higher-precision localization without significantly increasing computation complexity. Compared to CSSTORM, FALCON also reduces sample time to 2.5 second temporal resolution. Along with DAOSTORM, FALCON performs very well in high-density localization tasks.

III. Methods

A. Super-Resolution

Recently, CNNs have achieved state-of-the-art results for super-resolution tasks [3], [9], [10]. In particular, Shi et al. [3] introduce an efficient method for learning super-resolution by upscaling in the network directly, rather than relying on other external upscaling methods. This particular architecture is based on the idea of subpixel convolutions, where a normal convolutional layer is used with fractional stride size in order to upscale the resulting output. However, as noted by Shi et al., subpixel convolution exponentially increases training time. Thus, they introduce a novel, efficient method for computing an output equivalent to that of subpixel convolutions. This is achieved by convolving a normal filter with $r*r*c$ channels and a stride of 1×1, where r is the upscaling factor. The resulting image has size $w \times h \times r*r*c$, and is then rearranged into an image of size $w*r \times h*r \times c$. This operation is represented by Figure 1.

Our network architecture implements this technique in order to map low-resolution microscope data to a high-resolution label space. By using this convolutional layer, we can train an end-to-end super-resolution network for localization with arbitrarily-large output resolution, thus increasing our localization accuracy.

B. Distance Transform Regression

The task of localization is closely related to the well-studied problem of counting. The current best methods for counting are based on the work of Lempitsky and Zisserman. [11], where regression models are trained to map objects to density maps. These density maps represent objects as Gaussian distributions, and are designed in such a way such that the discrete integral of the map is equal to the count of objects in the given image. While this technique has achieved state-of-the-art results for the task of counting on various datasets [12], it does not provide a good method for accurately localizing objects in these images.

However, density-map regression can be slightly altered in order to better-localize proteins. Rather than learning regression model from objects to Gaussians, we attempt to map a pixel to its distance from the nearest protein location, as proposed by Kainz et al. [13]. Each pixel in a label sample is assigned a value d based on its Euclidean distance to the nearest localization. Then, the following operation is performed on the label data:

$$f(d) = \begin{cases} e^{\alpha(1-\frac{d}{d_{max}})} - 1 & 0 \leq d < d_{max} \\ 0 & d \geq d_{max} \end{cases} \quad (1)$$

Through this operation, called the *distance transform*, each protein is assigned an exact peak with the value e^{α}. After training our model to regress protein images to this function, we can find local maxima of the network output, thresholded at some lower-bound. These maxima correspond to protein locations and are accurate up to the resolution of the output image. Furthermore, these locations can be refined slightly by fitting a quadratic to a small neighborhood about any local maxima, thus producing a continuous output space.

IV. Experiments

A. Datasets

Test data for this experiment comes from the Single-Molecule Localization Microscopy (SMLM) Symposium challenges from 2013 and 2016. These challenge datasets provide simulated PA-FP microscope samples at varying densities as well as ground-truth locations for these simulated proteins. These simulations are designed to model proteins in various tubulin structures.

The SMLM challenges provide leaderboards showing results from a number of existing techniques (including those mentioned above) on contest datasets. While the ground truth localizations for these datasets are not provided, our results may be sent in for evaluation, allowing us to make general comparisons between our method and various state-of-the-art algorithms.

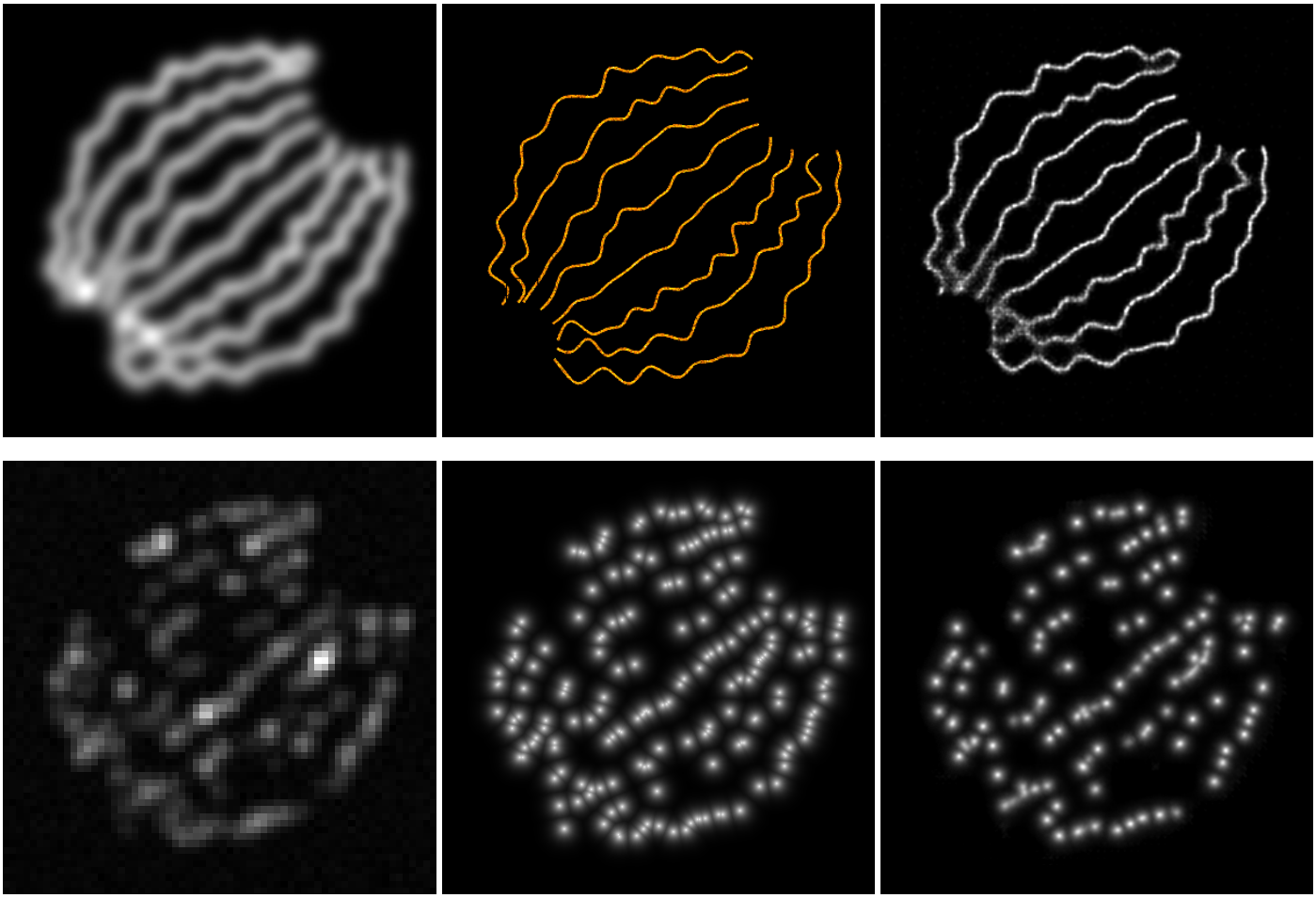

Fig. 2: Results from the 2013 SMLM dataset using the distance transform. The top row shows the 2013 dataset averaged over all frames (left), the ground-truth protein positions (middle), and a histogram of our network protein location predictions (right), where a brighter value signifies more-frequent occurrences of protein locations. The bottom row contains a sample testing image (left), its label data using the distance transform (middle), and our network prediction (right)

1) 2013: The 2013 challenge is focused on high-density localization. From this challenge, we use the "Bundled Tubes High Density" dataset, which models 8 tubulins with a diameter of 30 nm, in a 100 nm thick microscope slide. This dataset has 81049 fluorophores contained in 168 frames, making it the highest density dataset used in our experiments.

2) 2016: The primary focus of the 2016 challenge is on 3D localization. However, the challenge contains multiple 2D training datasets. In order to distinguish these datasets from those in the 2013 challenge, these 2D localization datasets simulate a thicker microscope slide of 1500 nm, meaning that approximately half of the proteins appear out of focus. From the 2016 challenge, we use the MT0.N1.HD and MT0.N2.HD datasets, both of which represent high-density samples. Proteins in these datasets occur at a slightly lower density than the 2013 samples, modeling three microtubules over 2500 frames and a total of 11172 flashes. These datasets simulate identical protein flashes and locations within each frame, differing only in signal-to-noise (SNR) ratios. The MT0.N1.HD dataset has a high peak SNR average of 22.597, while the MT0.N2.HD has a lower peak SNR average of 19.425. We will refer to these datasets as high-SNR and low-SNR 2016 datasets respectively.

3) Evaluation: Training data is generated using ThunderSTORM's simulator, which creates artificial microscopic images together with ground-truth locations. This simulator incorporates a number of parameters which specify the background noise, density of proteins, and camera detector settings. In order to evaluate the neural network performance, we use ThunderSTORM's built-in evaluation program. This program computes statistics such as the precision, recall, Jaccard Index, F1-Measure, and root-mean-squared (RMS) error of a result compared to ground-truth data. In this evaluation, one specifies the maximum distance allowed for a localization to be considered correct, which we refer to as the tolerance. Our neural network predictions are generated using the methods described in the following subsection.

B. 2013 Experiments

For the 2013 dataset, we use an architecture of 9 convolutional layers with 3×3 filter-size and 32 feature-detectors, 1 subpixel layer with an upscaling factor $r = 7$ and 32 feature

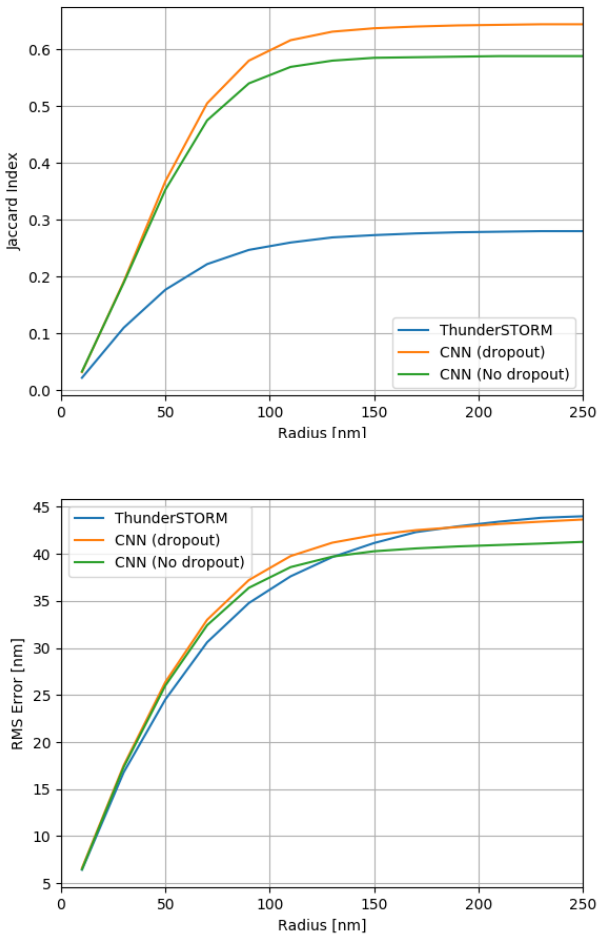

Fig. 3: Jaccard Index (top) and RMSE (bottom) for our 2013 neural network predictions at a range of tolerances. These plots show our neural network results with and without dropout layers, as well as the results from ThunderSTORM's analysis

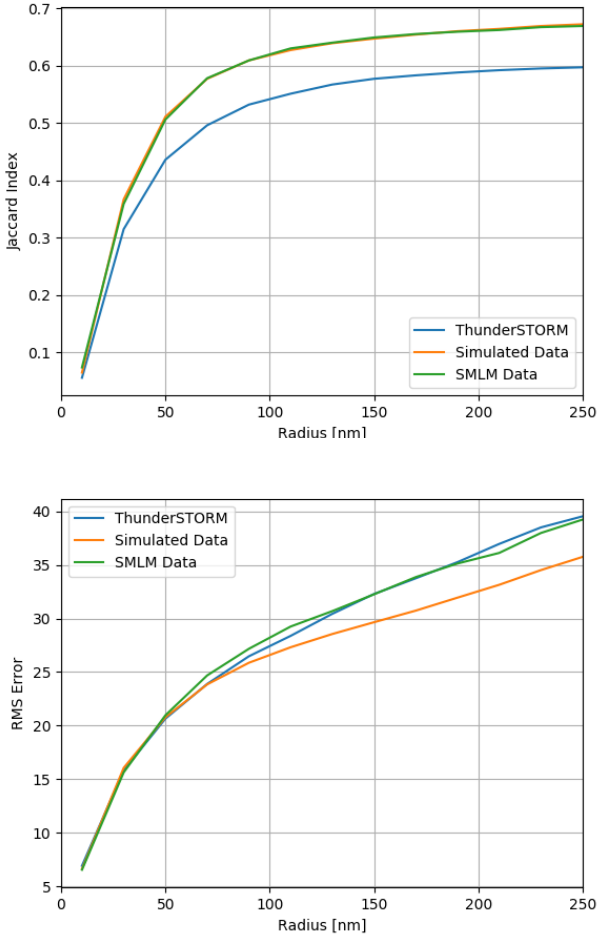

Fig. 4: Jaccard Index (top) and RMSE (bottom) on the 2016 high-SNR SMLM dataset. Here, we show results when training the CNN on data simulated from ThunderSTORM and on SMLM contest data. We also show ThunderSTORM results.

detectors, and a 1×1 filter size flattening convolutional layer which outputs gray scale image. We train this network on 1000 64×64 images with a pixel-size of 100×100 nm generated with ThunderSTORM. These images are designed to have density and background noise similar to that found in the 2013 SMLM HD dataset, but with completely random distribution rather than tubulin structure. For the distance transformation, we set $\alpha = 7$ and $d = 35$ pixels. We experiment with this configuration using a network with no dropout as well as one with a dropout rate of .5%. Visual results of the resulting regression map are shown in Figure 2.

When evaluating these results, we choose a threshold of 300 for local-maxima. After evaluating the Jaccard Index and RMS error of our results at various tolerances between 10 and 250 nm, we also analyze the 2013 dataset using ThunderSTORM's built-in protein localization software. A comparison between these methods is shown in Figure 3, which includes both dropout and non-dropout network architectures.

C. 2016 Experiments

Due to the thicker nature of the 2016 SMLM datasets, not all protein flashes are in focus in the SMLM training datasets. Because of this, our training method for the 2016 data differ slightly from those in the 2013 experiments. For both low- and high-SNR datasets, we train on 4000 simulated ThunderSTORM images. These simulated images are made using gradient density masks, which increase protein density from top to bottom of the image, as well as gradient noise masks, which increase the noise signal in the simulations from the left to the right. During preprocessing, we flip and rotate these images in cycles of 8 to achieve all possible orientations of these density and noise masks. In both datasets, we use an a model with 11 convolutional layers (32 feature detectors), 1 subpixel layer with a factor $r = 7$ and 10 feature detectors, and a final 1×1 convolutional layer. In both of these models, we use a value of $\alpha = 7$ and $d = 42$.

We compare these results to the ThunderSTORM evaluation of these datasets. We also train our model on low-SNR and high-SNR training sets directly in order to compare against the performance using simulated training data. Before training

on these datasets directly, we set aside the first 500 frames of both datasets for validation. The results from all of these experiments are shown in Figures 4 and 5, all tested on the 500 validation frames.

D. Results

In many of these experiments, we see that our super-resolving CNN outperforms ThunderSTORM in both Jaccard Index and RMS error. The exception to this is in the 2016 low-SNR RMS error, where ThunderSTORM obtains a slightly lower value. Despite this, our implementation improves significantly on ThunderSTORM's results in terms of Jaccard Index (see Tables I and II)

From these experiments, we see that our network learns a generalizable model for protein localization. After training on stochastically-placed protein signals and testing on data with a tubulin structure, our network achieves good results in localizing high-density proteins. Furthermore, in the 2016 experiments, there is no significant difference between CNN performance when trained on ThunderSTORM-simulated data as opposed to contest data directly. Thus, our experiments suggest that this network can be applied to a wide array of microscope data, by simply changing the parameters of the ThunderSTORM simulations used for training.

Finally, note that evaluation times are significantly reduced in the neural network method when compared to methods such as ThunderSTORM. After training, our network evaluates the 500-frame 2016 validation sets in approximately 90 seconds, where the majority of this time is spent loading the images and writing out the results. This time itself could be improved significantly by using a solid-state drive

	ThunderSTORM	CNN (dropout)	CNN (no dropout)
Jaccard Index	.280	**.644**	.588
RMSE [nm]	44.0	43.7	**41.3**

TABLE I: 2013 results

	ThunderSTORM	Simulated Data	SMLM Data
Jaccard Index	.597/.487	.672/.594	**.669/.597**
RMSE [nm]	39.5/**48.4**	**35.7**/52.4	39.2/52.2

TABLE II: 2016 results (high-SNR/low-SNR)

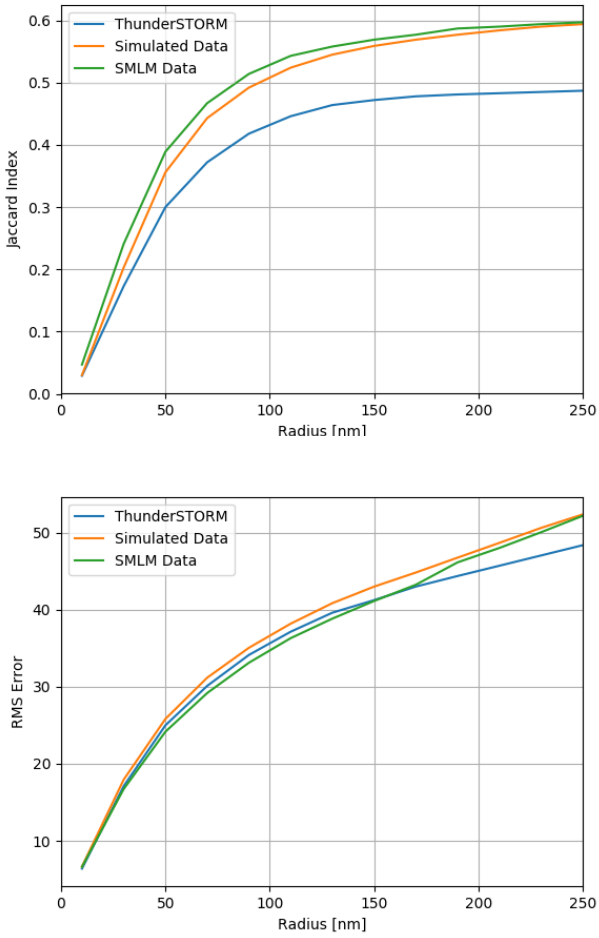

Fig. 5: Jaccard Index (top) and RMSE (bottom) on the 2016 low-SNR SMLM dataset. As in Figure 4, we show ThunderSTORM analysis as well as results when training the network on ThunderSTORM simulated data and on SMLM data.

V. Conclusion

Our network achieves very promising results on high-density datasets. With further refinement, we expect our results to be competitive with other top methods aimed at localizing high-density proteins. Although CNNs have not been thoroughly explored in this context, initial results from this project indicate their applicability to the field of nanometer-scale microscopy. By implementing techniques such as subpixel super-resolution and distance transform regression, we have shown that neural networks are a fast and accurate method for imaging living samples beyond the diffraction limit of 200 nanometers.

VI. Future Work

In the upcoming months, we propose further modifications to our model. Currently, our architecture only uses one scaling layer in order to upscale the image. We propose to experiment with multiple scaling layers with smaller scale factors, thus introducing non-linearities between phases of scaling. One example architecture consists of three scaling layers with a scale-factor $r = 2$, thus resulting in a total upscaling of 8. Furthermore, we have up until now chosen to focus on 2D localization tasks. However, the 2016 SMLM Challenge primarily focuses on 3D localization, and thus has several 3D datasets available. In the coming months, we propose to extend our model to 3D localization tasks. Finally, we plan to submit our results to the SMLM 2016 Challenge in order to more-directly compare our method against current state-of-the-art algorithms.

VII. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1359275 and 1659788. Furthermore, we acknowledge Diptotip Deb and Sridhama Prakhya for their helpful conversations and insights during the research process.

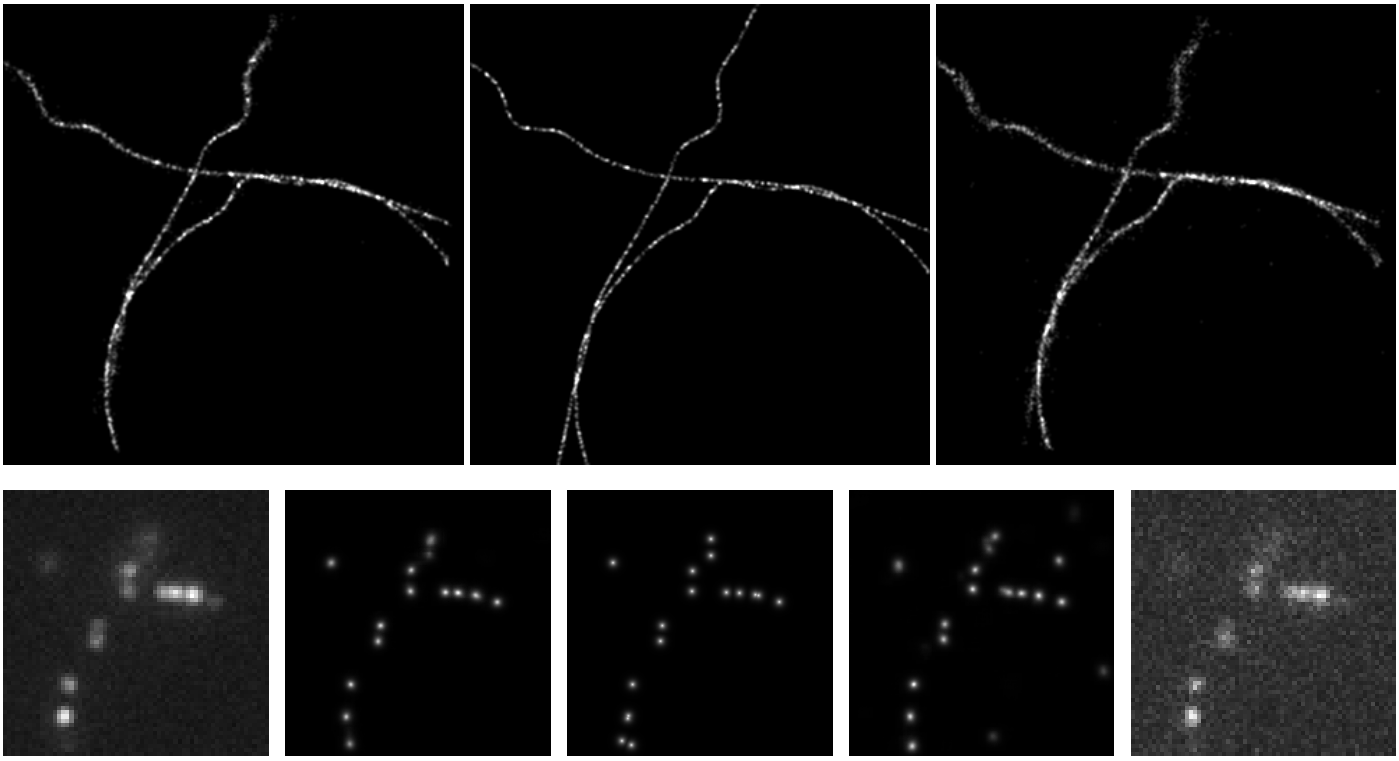

Fig. 6: Results from the 2016 low- and high-SNR datasets. The top row shows average histograms from the network predictions, where the center shows the ground truth, the left shows high-SNR results and the right shows low-SNR results. The bottom row displays a high-SNR testing image (left), the high-SNR network prediction (left-middle), the ground-truth label (middle), the low-SNR network prediction (right-middle), and the low-SNR testing image (right)

REFERENCES

[1] E. Betzig, G. H. Patterson, R. Sougrat, O. W. Lindwasser, S. Olenych, J. S. Bonifacino, M. W. Davidson, J. Lippincott-Schwartz, and H. F. Hess, "Imaging Intracellular Fluorescent Proteins at Nanometer Resolution," *Science*, vol. 313, no. 5793, 2006. [Online]. Available: http://science.sciencemag.org/content/313/5793/1642

[2] M. J. Rust, M. Bates, and X. Zhuang, "Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (storm)," *Nat Meth*, vol. 3, no. 10, pp. 793–796, 10 2006. [Online]. Available: http://dx.doi.org/10.1038/nmeth929

[3] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1874–1883, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7780576/

[4] S. J. Holden, S. Uphoff, and A. N. Kapanidis, "DAOSTORM: an algorithm for high- density super-resolution microscopy," *Nature Methods*, vol. 8, no. 4, pp. 279–280, 2011. [Online]. Available: http://www.nature.com/doifinder/10.1038/nmeth0411-279

[5] L. Zhu, W. Zhang, D. Elnatan, and B. Huang, "Faster STORM using compressed sensing," *Nature Methods*, vol. 9, no. 7, pp. 721–723, 2012. [Online]. Available: http://www.nature.com/doifinder/10.1038/nmeth.1978

[6] E. A. Mukamel, H. Babcock, and X. Zhuang, "Statistical deconvolution for superresolution fluorescence microscopy," *Biophysical Journal*, vol. 102, no. 10, pp. 2391–2400, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.bpj.2012.03.070

[7] M. Ovesný, P. Kížek, J. Borkovec, Z. Švindrych, and G. M. Hagen, "ThunderSTORM: A comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging," *Bioinformatics*, vol. 30, no. 16, pp. 2389–2390, 2014.

[8] J. Min, C. Vonesch, H. Kirshner, L. Carlini, N. Olivier, S. Holden, S. Manley, J. C. Ye, and M. Unser, "FALCON: fast and unbiased reconstruction of high-density super-resolution microscopy data," *Scientific Reports*, vol. 4, no. 1, p. 4577, 2015. [Online]. Available: http://www.nature.com/articles/srep04577

[9] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*, vol. 8689, 2014, pp. 184–199. [Online]. Available: http://link.springer.com/10.1007/978-3-319-10593-2_13

[10] J. Kim, J. K. Lee, and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[11] V. Lempitsky and A. Zisserman, "Learning To Count Objects in Images," *Advances in Neural Information Processing Systems*, pp. 1324–1332, 2010. [Online]. Available: http://papers.nips.cc/paper/4043-learning-to-count-objects-in-images.pdf

[12] D. Onoro-Rubio and R. J. Lopez-Sastre, "Towards Perspective-Free Object Counting with Deep Learning," in *ECCV 2016: 14th European Conference, Amsterdam, The Netherlands*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, 2016, pp. 615–629. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46478-7_38

[13] P. Kainz, M. Urschler, S. Schulter, P. Wohlhart, and V. Lepetit, "You Should Use Regression to Detect Cells," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 276–283, 2015.