2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)
April 4-7, 2018, Washington, D.C., USA

DEEP NEURAL NETWORKS FOR DATA ASSOCIATION IN PARTICLE TRACKING

Yao Yao, Ihor Smal, Erik Meijering

Biomedical Imaging Group Rotterdam, Erasmus University Medical Center, Rotterdam, the Netherlands

ABSTRACT

An essential first step towards understanding intracellular dynamic processes using live-cell time-lapse microscopy imaging is to extract accurate trajectories of all relevant particles in the images. One of the key aspects of this task is to make accurate associations between particle detections across time frames. State-of-the-art methods for this purpose often have many user parameters, sometimes even without a clear biophysical meaning, and/or they use explicit particle motion models that may not reflect reality, making them unfavorable for non-expert users or specific applications. Here we present a novel approach to data association for particle tracking applications based on deep neural networks. Specifically, we propose a recurrent neural network that learns particle behavior from the data, and based on this it determines how to best extend trajectories from one frame to the next. The results of preliminary experiments indicate that our method performs comparable to state-of-the-art data association methods for particle tracking, but with the advantage that it does not require users to tune any parameters.

***Index Terms*—** Particle tracking, data association, deep learning, multi-task learning, neural networks.

1. INTRODUCTION

Particle tracking is an essential first step in many biological studies into the molecular mechanisms of intracellular dynamic processes imaged using time-lapse microscopy. It is often achieved by detecting particles in the individual frames of the image sequences and then linking detections across frames to construct trajectories in time. Many methods have been developed to perform these tasks automatically. Recent studies have comprehensively evaluated the performance of a wide range of particle tracking methods [1] as well as of data association methods for the linking step [2]. All of these methods have several (often many) free parameters that are sometimes hard to tune and do not always have a clear biophysical meaning. Moreover, many methods use explicit (and often simplistic) assumptions about the particle dynamics, which may not accurately reflect reality. These disadvantages make them difficult to apply by non-expert users and/or to a wider range of tracking problems. Both the applicability and usability of particle tracking methods could potentially be improved by employing methods that can autonomously learn from the data the underlying particle dynamics and how to best link detections across frames.

In this paper we explore this idea and present a novel approach to particle linking based on deep learning. In recent years deep learning has found widespread application in many areas of biomedical imaging [3, 4] and has also been previously used for particle and cell detection [5, 6] and segmentation [7]. However, to the best of our knowledge, it has not been investigated for data association in particle tracking. Specifically we propose a recurrent neural network (RNN) that learns and models particle behavior from given data in order to determine how to best extend trajectories from frame to frame during tracking. Thanks to its predictive power, our network allows for missing and spurious detections. We compare our method with current state-of-the-art linking approaches using realistic simulated data [1, 2].

2. RELATED WORKS

The task of data association is important not only in particle tracking but also in many other multi-object tracking problems, such as multi-person tracking, vehicle tracking, and cell tracking. In recent years many deep learning-based tracking methods have been proposed for single-object tracking [6, 8, 9] and multi-object tracking [10–12]. All of them extract high-level appearance features from individual frames using deep neural networks, mostly convolutional neural networks (CNN), and then compute the probability of whether two detections belong to the same trajectory or not.

Instead of using appearance models alone, the idea of combining motion dynamics of the objects with appearance to improve the performance has also been explored [9]. For multi-object tracking problems, linear integer programming is often used to assign detections to trajectories based on the calculated probabilities. However, contrary to people or natural objects, where rich appearance representations can be learned, the objects of interest in our applications (subresolution particles) have little or no distinctive appearance features that are helpful for data association. Therefore, in this work, we use hand-crafted dynamics features to learn whether a detection belongs to a trajectory or not.

Another aspect of particle tracking is the need to deal with initialization and/or termination of trajectories as well as missing and/or spurious detections due to noisy output of the particle detector. These challenges have recently also been explored for applications in high-energy physics [12] but require further investigation. To solve these problems, in going from frame to frame, we need a method that not only assesses possible trajectory-detection associations, but also predicts the next locations of trajectories in order to detect possible gaps. A simple approach to deal with both tasks would be to train two separate networks. However, since the two tasks may not be mutually independent, a better approach seems to be to use some form of multi-task learning, which has been shown to be beneficial in other applications [6, 13]. Here we build on this idea and propose a multi-task deep learning architecture for particle tracking.

3. PROPOSED METHOD

3.1. Data Association Framework

The proposed method uses a temporal sliding window for solving the association problem from frame to frame (Fig. 1). Specifically, at any given frame k, we have a set of established tracklets, which are fragments of particle trajectories in the past $S+1$ frames until k (inclusive), and we have a set of candidate detections in frame $k+1$. To simplify the association problem, for each established tracklet, the candidate detections are all possible detections in frame $k+1$ whose spatial distance with respect to the tracklet position in frame k is less than a specified value (gating). Then we extract dynamics features from each tracklet and its candidates. Of the many possible dynamics features that could be computed [14], we use instantaneous displacement vectors, instantaneous directions, displacements to the start of sliding window, distances to the start of sliding window, and the locations of the individual detections. These features are passed on to a deep neural network (described next), which for each tracklet outputs an association score for each of its candidates, as well as a prediction of the tracklet's position in frame $k+1$ based on its behavior in the foregoing $S+1$ frames. Next, to obtain optimal assignments between tracklets and candidates based on the scores, mixed integer programming is applied using Gurobi (http://www.gurobi.com/). Finally, in a post-processing step, each established tracklet is extended with its optimal candidate or, if no candidate was found, with a virtual one corresponding to the predicted position. The latter implies there either was a gap in the data or the tracklet terminated. Any unassigned nonconflicting candidates in frame $k+1$ are taken to be newly starting tracklets. Then, the sliding window is moved by one frame to solve the assignment problem for that frame, and so on.

3.2. Neural Network Architecture

Our deep neural network (Fig. 2) performs two separate tasks: association estimation and motion prediction. The association

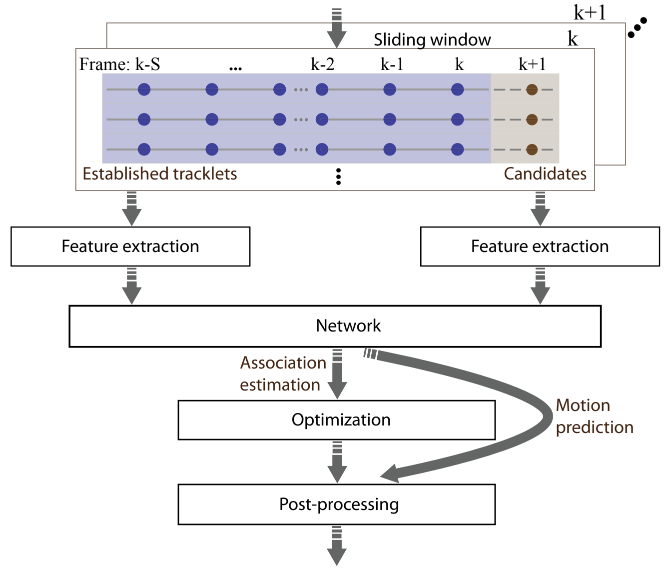

Fig. 1. Overview of our data association framework. Dynamics features are extracted from established tracklets and candidate detections in a sliding window. These are subsequently used by a deep learning network to score each pair of established tracklet and candidate and to predict the position of each established tracklet in the next frame. An optimization algorithm is used to find the optimal assignment based on the scores. In a post-processing step, each tracklet is extended with its optimal candidate, or with the predicted position if no candidate was found.

estimation, which outputs a score indicating the likelihood of an established tracklet and a candidate detection belonging to the same trajectory, is formulated as a classification problem. The motion prediction, on the other hand, which for a given tracklet outputs the expected spatial position in the next frame, is formulated as a regression problem. The network takes a pair of feature vectors, one from an established tracklet in S preceding frames plus the current frame, and the other from a candidate in the next frame. The established tracklet feature vector, which is the input of an RNN layer more precisely a long short-term memory (LSTM) layer, is used in both tasks. The candidate feature vector, which is the input of a fully-connected (FC) layer, is used only in the association estimation task. For the latter task, the output of the LSTM layer is concatenated with the FC output, and the result is fed into a sequence of FC layers. Next, a softmax classifier is applied, which outputs the probabilities for the positive and negative classes, where the positive class indicates that the candidate should be associated with the established tracklet, and the negative class that it should not. For the motion prediction task, the LSTM output is passed through two FC layers, which output the best possible location of the tracklet in the next frame. For all FC layers except the last, we use dropout [15] with fixed probability $p_d = 0.8$ as regularization to reduce overfitting in the training stage.

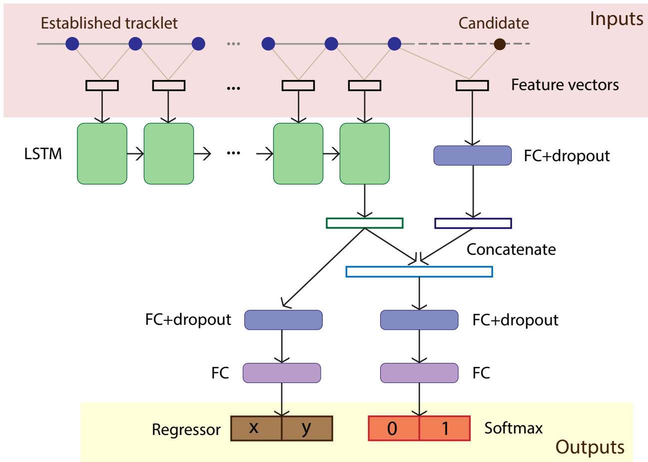

Fig. 2. Overview of our neural network architecture. The established tracklet features are passed through an LSTM layer, and the candidate features through an FC layer. The result vectors are concatenated and passed through two FC layers to produce probabilities using softmax for the association estimation task. Meanwhile, the output of the LSTM layer is also passed through another two FC layers, to produce a tracklet position in the next frame for the motion prediction task.

3.3. Multi-Task Loss Function

To avoid learning tasks separately and managing a pipeline for each, our network is trained in a multi-task fashion. The output of the association estimation branch of the network are probabilities $\rho = \{\rho_0, \rho_1\}$, corresponding to whether a tracklet and candidate are positively (ρ_1) or negatively (ρ_0) associated, and the output of the motion prediction branch are coordinates $\mu = \{\mu_x, \mu_y\}$. In our experiments, each training sample is labeled with a ground-truth classification $u = \{u_0, u_1\}$ and a ground-truth regression $v = \{v_x, v_y\}$, drawn from the corresponding ground-truth trajectory. We use a multi-task loss $\mathcal{L}$ to jointly train for the two tasks:

$$\mathcal{L} = \mathcal{L}_{\mathrm{cls}}(\rho, u) + \lambda \mathcal{L}_{\mathrm{reg}}(\mu, v), \tag{1}$$

where the first term is the cross-entropy loss for the network predicted probabilities ρ under true probabilities u:

$$\mathcal{L}_{\mathrm{cls}}(\rho, u) = -\sum_{i \in \{0,1\}} u_i \log(\rho_i). \tag{2}$$

The regression task loss $\mathcal{L}_{\mathrm{reg}}$ is defined over the tuple v of true positions in the next frame and a network predicted tuple μ. The commonly used L2 loss increases sharply with the difference between the target v and the estimated μ, which makes it too sensitive to unlabeled training samples. Therefore we use the more robust Huber loss function [16]:

$$\mathcal{L}_{\mathrm{reg}}(\mu, v) = \sum_{i \in \{x,y\}} \mathcal{H}(v_i - \mu_i), \tag{3}$$

$$\mathcal{H}(r) = \begin{cases} 0.5r^2 & \text{if } |r| < \delta, \\ \delta r - 0.5\delta^2 & \text{otherwise.} \end{cases} \tag{4}$$

In our experiments we used $\delta = 1$, and the hyperparameter in (1) controlling the balance between the two task losses was fixed to $\lambda = 0.1$, which gave good results.

4. EXPERIMENTAL RESULTS

To evaluate the performance of our method we followed the same procedure as in the most comprehensive evaluation study published to date [2]. Simulated data were taken from the particle tracking challenge [1] and processed to obtain a ground-truth (GT) dataset with different levels of false-negative (FN) detections (0%, 5%, 10%, 15%, 20%) and false-positive (FP) detections (0%, 20%). Neighboring detections within the Rayleigh distance were replaced by a single detection to get the 0% FN dataset, and from there on detections were randomly removed or added to get datasets with specific percentages of FN and FP. Performance was quantified using the α, β, and JSC$_\gamma$ measures [1, 2].

We compared our method to 11 alternative data association methods [2], including a two-frame (2D) greedy nearest-neighbor method based on distance (GNN-D) or velocity (GNN-V), a 2D interacting multiple models (IMM) filtering approach [17], a two-step linear assignment procedure (LAP) [18], a noniterative greedy assignment (NGA) procedure [19] using two (2D), three (3D), or four (4D) frames and convex costs (CC), and a multi-frame assignment procedure (MAP) [20] using three (3D) or four (4D) frames and IMM or CC for computing costs. The free parameters of all methods were optimized as published [2]. For our proposed RNN-based method we found that setting the window-size parameter to $S = 5$ gives good results.

Due to space constraints we show only the results for the low-density microtubule dataset (Fig. 3) but very similar results were obtained with all other datasets. From the results we observe that, as expected, the multi-frame (3D and 4D) data association approaches in most cases perform better than the two-frame (2D) approaches, although which specific approach (NGA or MAP and using IMM or CC) performs best depends on the measure. Generally our proposed method outperforms the 2D approaches and performs comparably to the best multi-frame approaches.

5. DISCUSSION AND CONCLUSIONS

In this paper we have presented a novel deep-learning based data association method for particle tracking. Our method exploits LSTM to learn the dynamic behavior of particles directly from the data, and hence it is able to capture complex motion patterns, which would be difficult to model with traditional methods. It optimizes its parameters using the training data, avoiding the need for manual tweaking, which makes

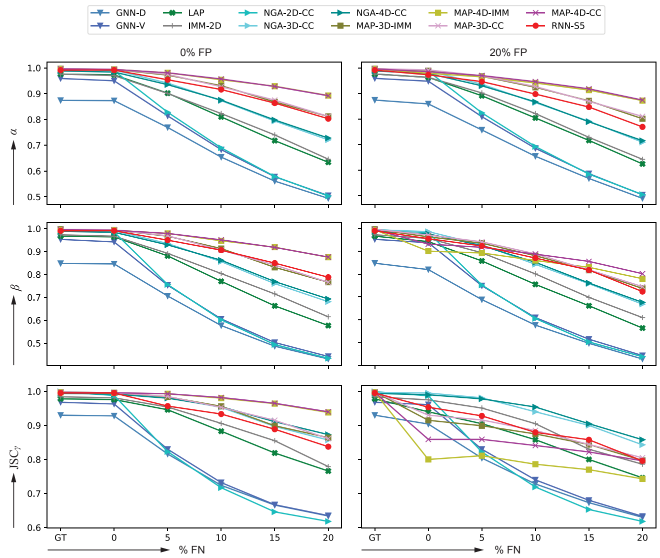

Fig. 3. Performance of our method compared to the state of the art. Results are shown for the low-density microtubule scenario and the α, β, JSC$_\gamma$ measures [1] with (a)-(c) 0% and (d)-(f) 20% false-positives (FP) as a function of the false-negative (FN) percentage. GT denotes the original ground-truth data before reduction using the Rayleigh criterion [2]. RNN-S5 denotes the proposed method (S = 5). See main text for a brief discussion of the other methods.

our method potentially more objective and applicable to a wider range of tracking problems and usage by non-experts. The presented experimental results show that our method is better than two-frame assignment methods and is comparable to state-of-the-art multi-frame methods.

Although the proposed method is promising in multiple ways, essentially it still uses a two-frame assignment approach, despite the fact that by employing LSTM we do use the history of particle movements to learn the dynamics. To further improve the potential of our method we aim to further extend it to a multi-frame assignment method by considering candidates in multiple succeeding frames compared to only using the next frame as in this work. Also, while we focused our evaluation on simulated datasets in order to easily compare with existing methods, we are in the process of testing our method for real cell and particle tracking applications. Finally, our future work will also involve using deep learning for particle detection, and building an end-to-end solution for the particle tracking problem, directly producing optimal trajectories from raw image sequences.

6. REFERENCES

[1] N. Chenouard et al., "Objective comparison of particle tracking methods," *Nature Methods*, vol. 11, pp. 281–289, 2014.

[2] I. Smal and E. Meijering, "Quantitative comparison of multiframe data association techniques for particle tracking in time-lapse fluorescence microscopy," *Medical Image Analysis*, vol. 24, pp. 163–189, 2015.

[3] H. Greenspan, B. van Ginneken, and R.M. Summers, "Deep learning in medical imaging: overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 1153–1159, 2016.

[4] G. Litjens et al., "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[5] J.M. Newby, A.M. Schaefer, P.T. Lee, M.G. Forest, and S.K. Lai, "Deep neural networks automate detection for tracking of submicron scale particles in 2D and 3D," *arXiv:1704.03009*, 2017.

[6] T. He, H. Mao, J. Guo, and Z. Yi, "Cell tracking using deep neural networks with multi-task learning," *Image and Vision Computing*, vol. 60, pp. 142–153, 2017.

[7] O.Z. Kraus, J.L. Ba, and B.J. Frey, "Classifying and segmenting microscopy images with deep multiple instance learning," *Bioinformatics*, vol. 32, pp. i52–i59, 2016.

[8] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, and P.H.S. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proceedings of the European Conference Computer Vision Workshops: Part II*, 2016, pp. 850–865.

[9] G. Ning et al., "Spatially supervised recurrent convolutional neural networks for visual object tracking," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2017, pp. 1–4.

[10] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 418–425.

[11] A. Milan, S.H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, pp. 4225–4232.

[12] S. Farrell et al., "The HEP.TrkX Project: Deep neural networks for HL-LHC online and offline tracking," *EPJ Web of Conferences*, vol. 150, pp. 00003, 2017.

[13] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[14] Y. Yao, I. Smal, I. Grigoriev, M. Martin, A. Akhmanova, and E. Meijering, "Automated analysis of intracellular dynamic processes," *Methods in Molecular Biology*, vol. 1563, no. 14, pp. 209–228, 2017.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[16] P.J. Huber, "Robust estimation of a location parameter," *Annals of Mathematical Statistics*, vol. 35, pp. 73–101, 1964.

[17] H.A.P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, pp. 780–783, 1988.

[18] K. Jaqaman et al., "Robust single-particle tracking in live-cell time-lapse sequences," *Nature Methods*, vol. 5, pp. 695–702, 2008.

[19] A. Jaiswal, W.J. Godinez, R. Elis, M.J. Lehmann, and K. Rohr, "Tracking virus particles in fluorescence microscopy images using two-step multi-frame association," in *Proceedings of the IEEE International Symposium on Biomedical Imaging*, 2012, pp. 664–667.

[20] L. Feng, Y. Xu, Y. Yang, and X. Zheng, "Multiple dense particle tracking in fluorescence microscopy images based on multidimensional assignment," *Journal of Structural Biology*, vol. 173, pp. 219–228, 2011.