Example code for SPTnet

This software is distributed as accompanying software for the manuscript: ‘SPTnet: a deep learning framework for end-to-end single-particle tracking and motion dynamics analysis’ by Cheng Bi, Kevin L. Scrudders, Yue Zheng, Maryam Mahmoodi, Shalini T. Low-Nam, and Fang Huang

1. Files included in this package

Matlab scripts/file:

SPTnet_trainingdata_generator.m	Matlab code to generate simulated videos for SPTnet training
Visualize_SPTnet_Outputs_GUI.m	Matlab code to visualize the final output from SPTnet.
CRLB_H_D_frame.mat	Calculated CRLB matrix of Hurst exponent and generalized diffusion coefficient used in the loss function

Python scripts:

SPTnet_toolbox.py	Script for SPTnet architecture and other tools used in loading data and output result
SPTnet_training.py	Script to train the SPTnet
SPTnet_infernece.py	Script to use a trained model for inference
transformer.py	The transformer module used as spatial-T
transformer3D.py	The transformer module used as temporal-T

Folders:

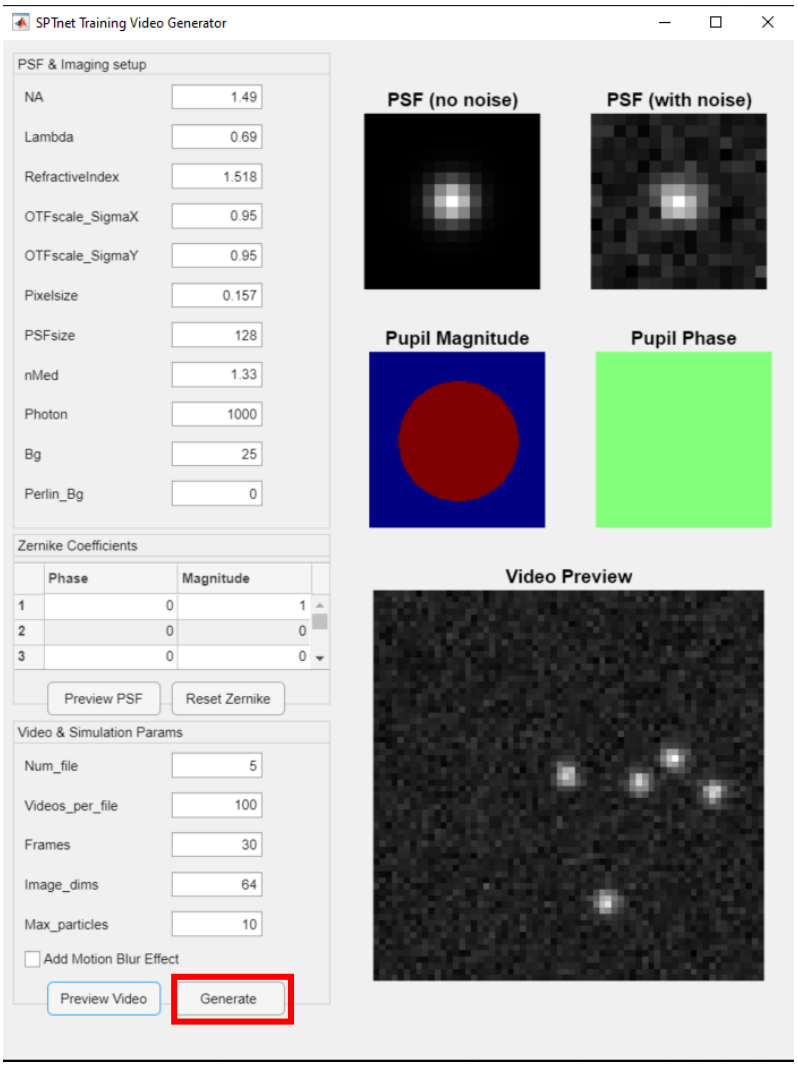
Example test data	Containing 10 test videos in one file, and a Tiff series from experimental data
PSF-toolbox	Toolbox used to simulate PSF through pupil function
Trained models	Containing a pre-trained model based on the parameters of a Nikon Ti2 TIRF microscope. (64x64 pixels, 30 frame)

DIPimage_2.9	Installation files of DIPimage 2.9 for both Windows and Linux

2. Instructions on generating training dataset

Change MATLAB current folder to the directory that contains “PSF-toolbox”.

Run ‘SPTnet_trainingdata_generator.m’ to generate the training dataset.



The default settings will generate 5 files each containing 100 videos.

This code has been tested on the following systems and packages:

Microsoft Windows 10 Education, Matlab R2021a, DIPimage 2.9

(http://www.diplib.org/)

3. Instructions on training neural network for SPTnet

The code has been tested on the following systems and packages:

Ubuntu20.04LTS, Python3.9.12, Pytorch1.11.0, CUDA11.3, MatlabR2021a

To start training,

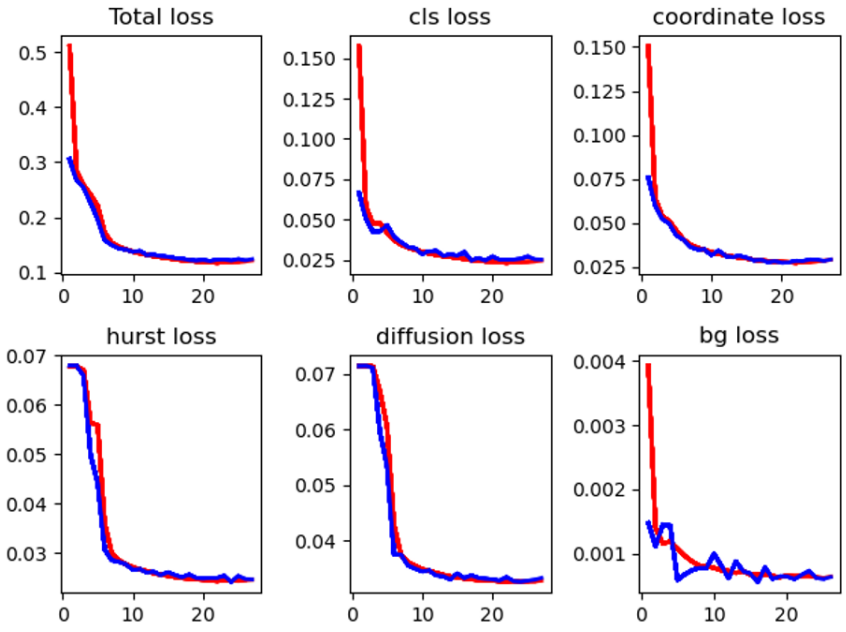
(1) Install and activate the environment for SPTnet, see “Installation of SPTnet.docx”

(2) Navigate to the folder with SPTnet_training.py file and type the following command in the terminal: python SPTnet_training.py

(3) Select the folder to save the trained model

(4) Select the training data files.

(5) During the training, the model with the minimal validation loss will be saved as ‘trained_model’ onto the selected folder in step (2), together with an image of the training loss and validation loss changes along with training epoch.



4. Instructions on inference using a trained model

To test the trained model,

(1) Type the following command in terminal: python SPTnet_inference.py

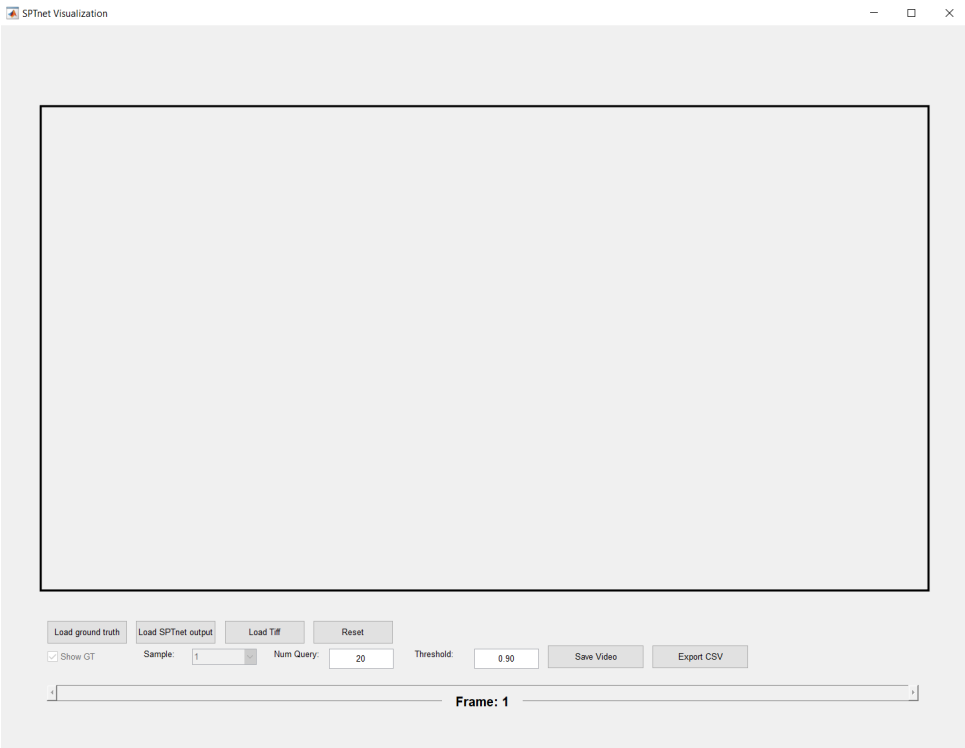
(2) Select the trained model that will be used to conduct the inference

(3) Select the video file or Tiff series that will be analyzed by SPTnet

(4) An output '.mat' file will be generated under the 'inference_results' folder located in the directory of the selected model in step (2), which contains all the estimated trajectories, detection probabilities, Hurst exponents, and generalized diffusion coefficients ready for analysis or visualization.

5. Instructions on visualizing the SPTnet outputs overlay with ground truth in a videos

(1) Run 'Visualize_SPTnet_Outputs.m'

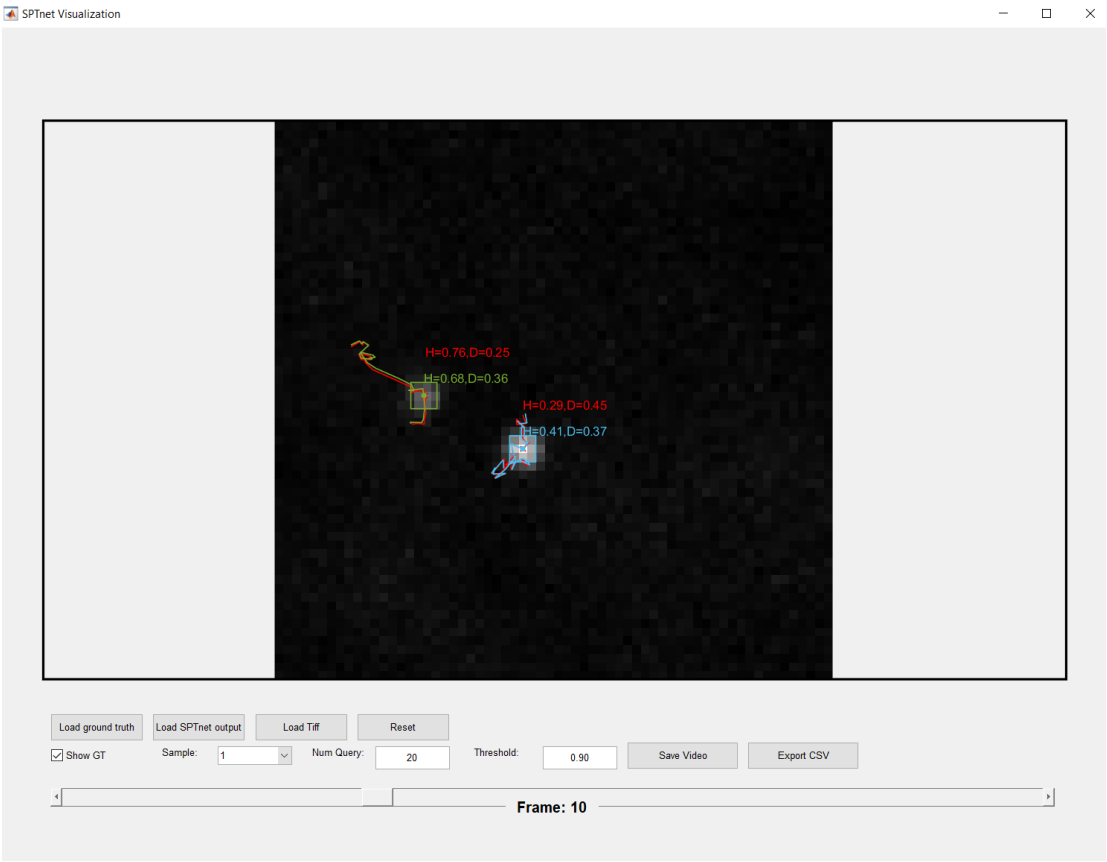


(2) Select the "Load ground truth" for files used for testing the model

(3) Select the "Load SPTnet output" for SPTnet inference results.

(4) By default, the tested videos with ground truth trajectories, Hurst exponent, and generalized diffusion coefficient will be shown in red, and the SPTnet estimation results

will show different colors for different tracks. An example frame from the visualization result is attached below.



Note: All necessary packages required to run this example code are listed in the file “package required for SPTnet.txt”

We also recommend to install all the packages through the “SPTnet_environment.yml”