**DeepEM Manual**

Copyright © 2016 | Peking University, in association with Intel® Parallel Computing Center for Structural Biology at Dana-Farber Cancer Institute.

1. **Introduction**

The MATLAB script package DeepEM is an automated single-particle recognition program in cryo-electron microscopy. It is a supervised machine learning method based on convolutional neural network (CNN) and realized based on the DeepLearnToolbox [1] in Matlab. The micrographs in mrc format are used as input image and output the box files which can be directly used in EMAN2 [2].

1. **Requirement and dependency**

* Matlab installation (R2014b recommended) with Image Processing Toolbox.
* DeepLearnToolbox: our algorithm is dependent on this toolbox, which is a deep learning tool box including Deep Belief Nets, Stacked Autoencoders, Convolutional Neural Nets, Convolutional Autoencoders and vanilla Neural Nets.

1. **Code structure**

The script files contain the following:

* DeepLearnToolbox: which contains the basic CNN framework
* Create\_CNN.m: to create the Convolutional neural network(CNN)
* Select\_CNN.m: to do particle recognition using the created CNN.
* Read\_train\_data.m: to read particle images in the box files from mrc file.
* mrcs\_read.m: to read mrc files from micrographs.
* Create\_19s.m Select\_19s.m: to run DeepEM for the example of proteasomal 19S complex.
* Create\_KLH.m and Select\_KLH.m: to run DeepEM for the example of KLH.

1. **Testing datasets**

19Sdata KLHdata: two example datasets to run the DeepEM algorithm. Dat file folder contains the original micrographs in dat format and mic file folder contains micrographs in mrc format converted using SPIDER [3] for the convenience of read by Matlab. The negative and positive file folders contain box files in EMAN2 of training datasets.

1. **DeepEM inputs and outputs**

Inputs:

1. Macrographs in MRC format, which are converted from .dat format files in the script in SPIDER.
2. Selected training dataset in the box file in EMAN2, which would be used to read the particle images.

Outputs:

1. In the training (create\_CNN) step, it would output the .mat file which is the trained CNN for the corresponding sample.
2. In the particle recognition (select\_CNN) step, it would generate nine result files representing the output boxfiles for each micrograph with different threshold values. To generate more boxfiles based on more threshold values, the users can modify the output part in the Select\_CNN.m

1. **Algorithm control parameters**
   1. **Parameters in the training step**

**The parameters of the micrographs:**

* dim\_x and dim\_y: The size of the micrographs
* name\_length, name\_prefix=The parameter of the micrograph name
* mic\_path=The path of the micrographs in MRC format

**The parameters of the training dataset:**

* Boxsize: The size of the particle images
* num\_positive1: The number of original positive dataset
* num\_negative1:The number of original negative dataset
* positive1\_box\_path: The path of the boxfile of the original positive dataset
* negative1\_box\_path: The path of the boxfile of the original negative dataset
* positive1\_mic\_start\_num: The number of start micrograph of original positive dataset
* positive1\_mic\_end\_num: The number of end micrograph of original positive dataset
* negative1\_mic\_start\_num: The number of start micrograph of original negative dataset
* negative1\_mic\_end\_num: The number of end micrograph of original negative dataset
* do\_train\_again: Flag (0/1) for whether to train the CNN again, detailed in workflow chart in the supplementary Fig.1.
* num\_positive2: The number of adding positive dataset
* num\_negative2: The number of adding negative dataset
* positive2\_box\_path: The path of the boxfile of the adding positive dataset
* negative2\_box\_path: The path of the boxfile of the adding negative dataset
* positive2\_mic\_start\_num: The number of start micrograph of adding positive dataset
* positive2\_mic\_end\_num: The number of end micrograph of adding positive dataset
* negative2\_mic\_start\_num: The number of start micrograph of adding negative dataset
* negative2\_mic\_end\_num: The number of end micrograph of adding negative dataset
* rotation\_angel: The angel to rotate the image to do augmentation
* rotation\_n=360/rotation\_angel

**The parameters of the testing dataset in the step:**

* num\_p\_test: The number of positive testing data
* num\_n\_test: The number of negative testing data

**The control parameters for the CNN:**

* FL\_kernelsize: The convolutional kernelsize of the first layer
* TL\_kernalsize: The convolutional kernelsize of the third layer
* FIL\_kernalsize: The convolutional kernelsize of the fifth layer
* SL\_poolingsize: The subsampling size of the second layer
* FOL\_poolingsize: The subsampling size of the fourth layer
* SIL\_poolingsize: The subsampling size of the sixth layer
* FL\_feature\_map: The number of feature maps in the first layer
* TL\_feature\_map: The number of feature maps in the third layer
* FIL\_feature\_map: The number of feature maps in the fifth layer

**The output name of the trained CNN:**

* OUTPUT\_CNN\_NAME: The name of the trained CNN
  1. **Parameters in the particle recognition step**
* load(‘CNN\_name.mat’): Load the trained CNN
* data\_path: The path of the data
* boxsize: The size of the particle image
* start\_mic\_num: The start number of the micrograph to do particle recognition
* end\_mic\_num: The end number of the micrograph to do particle recognition
* dim\_x, dim\_y: The size of the micrographs
* scan\_step: The step to scan the micrographs when doing particle recognition
* range1, range2:The parameters to select local maximum value
* min\_std, max\_std: The range of the standard deviation to do filter, which are determined by the micrographs, we usually use a relatively large range value at first and then gradually reduce to an appropriate one.
* rotation\_angel: The angel to rotate the image to do augmentation
* rotation\_n=360/rotation\_angel

1. **How to run the example dataset**

Two example datasets, KLH and 19S proteasomes, are included with the code download. The details about the datasets are included in the paper. To run the code using the example dataset:

1. Run the corresponding creat\_KLM.m code file to create CNN
2. Run the corresponding select\_KLH.m code file to do particle recognition

**Reference**

1. Palm, R. Prediction as a candidate for learning deep hierarchical models of data. IMM2012-06284 (2012)
2. Tang, G. et al. EMAN2: an extensible image processing suite for electron microscopy. *J. Struct. Biol.*, 157(1), 38-46 (2007)
3. Shaikh, T.R, *et al,* SPIDER image processing for single-particle reconstruction of biological macromolecules from electron micrographs. *Nature Protocols*. 3, 1941-1974 (2008)