Report 1:

Automatic Particle Picking in Cryo-EM

--Machine Learning in Biology

Li Yuhui 2013012470

Abstract

This is one report of the course Computational Biology, where a whole process of automatic particle picking from electron micrographs is proposed based on machine learning techniques. The process include: image data preprocess, different machine learning techniques and deep learning techniques.

Key Word: Computational Biology, Machine Learning, Particle Picking, Cryo-EM, Computer Vision

1. Introduction

The current approach of interactive manual selection of biological molecules from digitized micrographs for single-particle averaging and three-dimensional (3D) reconstruction requires substantial effort and time. A fast procedure for automatic particle picking would greatly simplify and enhance this task. Therefore, what is need is an efficient programming pipeline based on machine learning techniques that allow objects to be identified regardless of their orientation and location in the digitized electron micrograph.

Here, this report proposes a pipeline of automatic particle picking based on computer vision techniques and machine learning techniques. The process and theory will be introduced in the following sections.

2. Materials and Methods

2.1 Materials

The most of the algorithms in this article are performed in python 2.7 in windows 10 only. MLP is performed in matlab. CNN is performed both in kares (one python library based on theano) and matlab. Codes are submitted as well.

The dataset used is pictures from the cryo-EM picture of TRPV1 provided by TA.

As the memory require is too much for laptop, the whole particle picking work is not fully done. And as the CNN needs lots of data (i.e. lots of computer memory), the work result is actually not quite good and persuasive.

2.2 Methods and Results

2.2.1 Data Preprocessing

The raw data has a very low signal-noise ratio (SNR), with different contrast and brightness. And several images have very distinct bubbles on it. Therefore, in the step of data preprocessing, the images are: 1) Normalized; 2) Gaussian filtered or median filtered; 3) Bubble detected and erased; 4) Histogram Equalized; 5) Sliding window cropped (see Figure 1).

1) Normalization

Normalization is a simple and useful first step. All the images are normalized by:

$$I = \frac{I - min}{max - min}$$

2) Gaussian Filtering and Median Filtering

To produce a better estimate based on the raw images, one way to compute is:

$$R = \sum_{k=1}^{mn} (W_k Z_k)$$

Where W is the coefficient of a filter with size of m * n, Z is the value of the gray scale of the corresponding image covered by the filter. In the process, the filters are frequently seen as convolution kernals, which convolutes and traversals with the image window.

Gaussian Filter is this kind of linear convolution operation with Gaussian kernel, while median filter is a kind of non-linear approach, to run through the signal entry by entry, replacing each entry with the median of neighboring entries.

3) Bubble Detection

There are two ways to detect bubbles in this project. One is by K-means algorithm as:

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (||x_j - \mu_i||^2)$$

Where μ is mean vector of the clusters.

The other way is by erosion. Erosion expands the minimal values of the seed image until it encounters a mask image. Thus, the seed image and mask image represent the maximum and minimum possible values of the reconstructed image.

4) Histogram Equalization

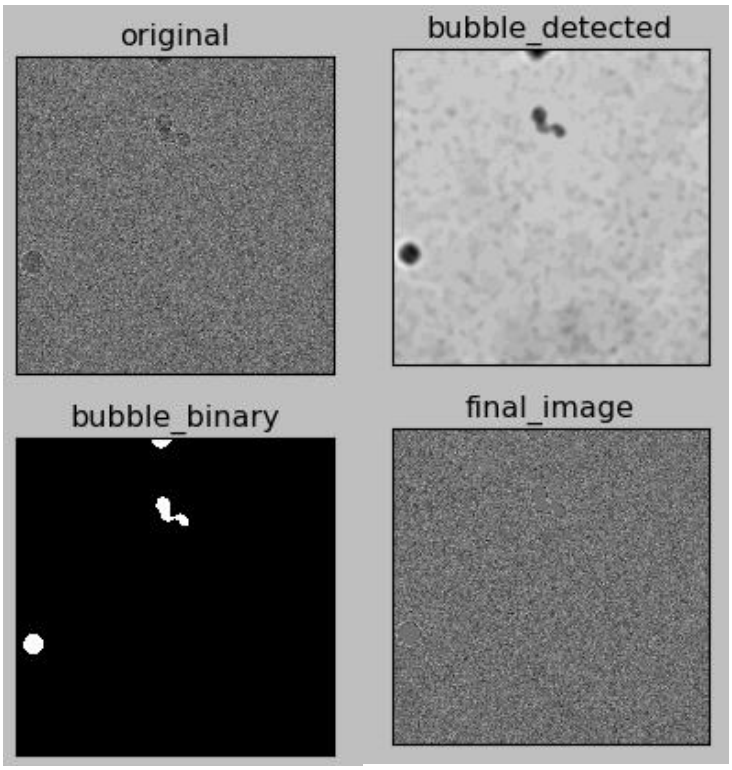
When an image has many gray scales and distributes uniformly, it frequently has high contrast and various hue. Therefore, if the most pixels of one gray scale are widened and those least are compressed, contrast of the image will be improved, thus lower the ambiguousness.

5) Sliding Window

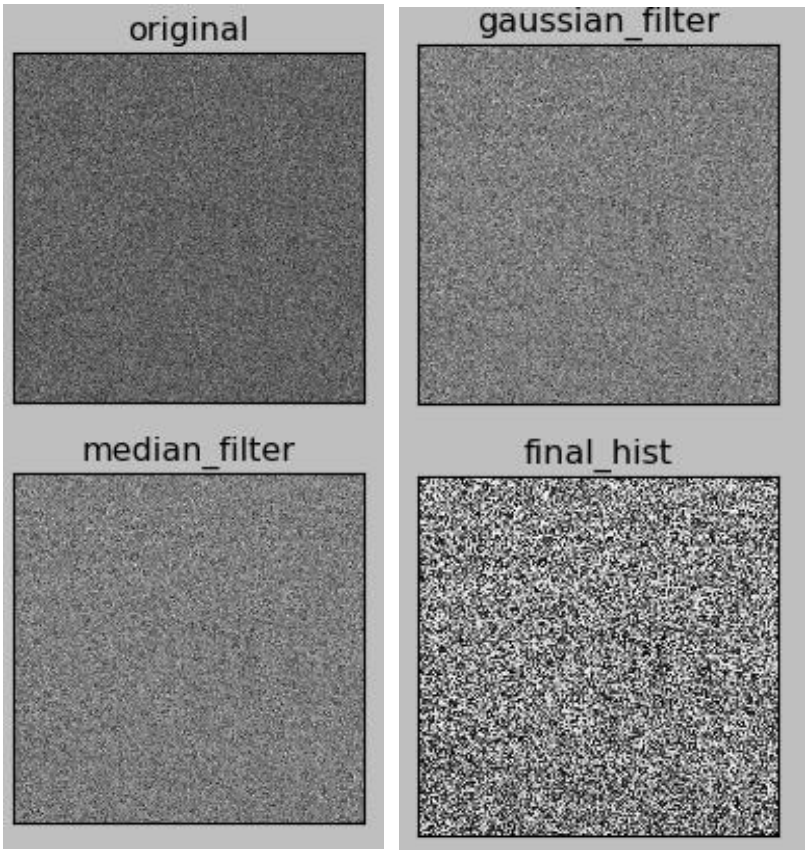
Finally, to extract the feature of data, sliding windows with size of 180 x 180 pixels are cropped from the images preprocessed by the above methods (3710 x 3710 pixels) and then resized to 64 x 64 pixels. Based on rotational invariance, all the images are

rotated by 90, 180, 270 degrees. Threshold of 36 pixels are chosen to decide which of the sliding windows are positive samples.

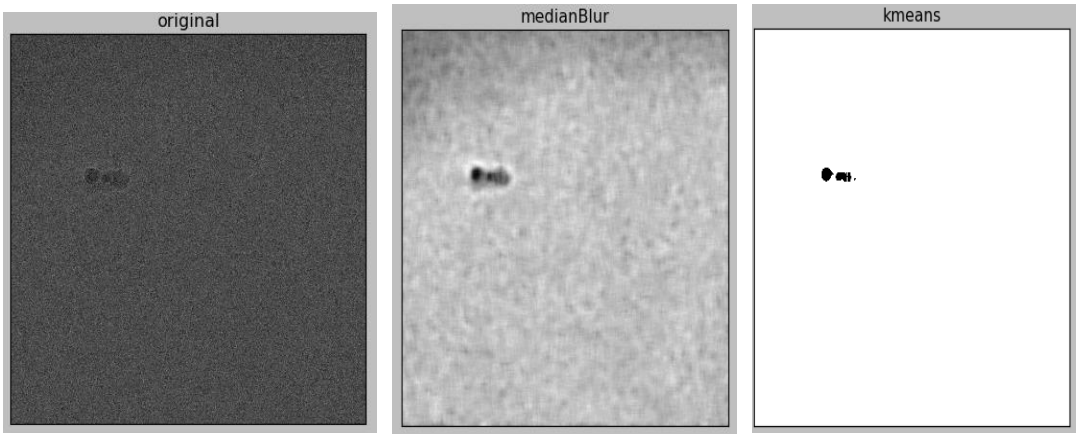
By the above approaches, total amount of data could reach 500,000.



(a)



(b)



(c)

Figure 1. Data Preprocessing. (a)Bubble Detection and Erasion by Erosion Method. (b) Data Denoising by Gaussian Filter and Median Filter, and Finally Histgram Equalization. (c) Bubble Detection by K-means. Here in bubble detection process, raw images are first filtered to blur everything to make the bubble more distinct, thus imporving the detecting accuracy.

2.2.2 Machine Learning Techniques

After data preprocessing, this problem can be transformed to a kind of supervised learning problem. Therefore, several machine learning models can be used such as support vector machine, logistic regression, multilayer perceptron, random forest, support vector machine and convolutional neural network. (See Table 1 and Figure 2)

1) Support Vector Machine (SVM)

Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. The linear SVM optimization can be written as:

$$y_i(\vec{w} \cdot \overrightarrow{x_i} - b) \geq 1$$

Where w is parameters of model, yi is output and xi is training samples. To extend to cases where the data are not linearly seperatable, hinge loss function is introduced as:

$$\max\left(0, 1 - y_i(\vec{w} \cdot \overrightarrow{x_i} - b)\right)$$

And to minimize:

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1-y_i(\vec{w}\cdot\overrightarrow{x_i}-b)\right)\right]+\lambda\|\vec{w}\|^2$$

Where the parameter λ determines the tradeoff between increasing the margin-size and ensuring that the xi lie on the correct side of the margin.

In addition, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

2) Logistic Regression

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution:

$$F(x)=\frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

Note that F(x) is interpreted as the probability of the dependent variable equaling a "success" or "case" rather than a failure or non-case.

3) Random Forest

Random forests is a notion of the general technique of random decision forest that are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

The training algorithm for random forests applies the general technique of bootstrap aggregating. Given a training set X with responses Y, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples. After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':

$$\hat{f}=\frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(x')$$

Random forests differ in only one way from this general scheme: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. This process is sometimes called "feature bagging". The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the trees, causing them to become correlated.

4) Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple

layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training the network.

In this work, a 2-layer MLP is performed with one hidden layer to those data.

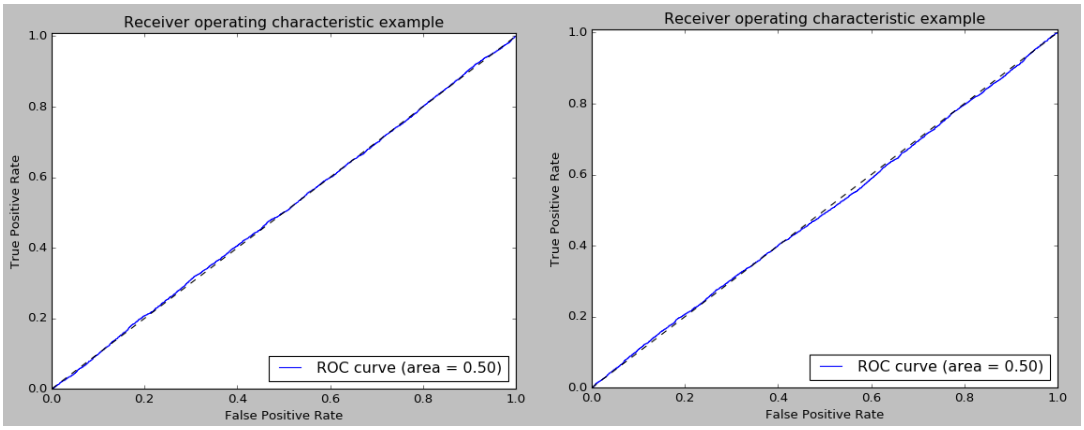
5) Convolutional Neural Network (CNN)

Convolutional neural networks are biologically inspired variants of multilayer perceptrons, designed to emulate the behaviour of a visual cortex. These models mitigate the challenges posed by the MLP architecture by exploiting the strong spatially local correlation present in natural images. As opposed to MLPs, CNN have the following distinguishing features: 1) 3D volumes of neurons. The layers of a CNN have neurons arranged in 3 dimensions: width, height and depth. 2) Local connectivity. CNNs exploit spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. 3) Shared weights. In CNNs, each filter is replicated across the entire visual field. These replicated units share the same parameterization (weight vector and bias) and form a feature map.

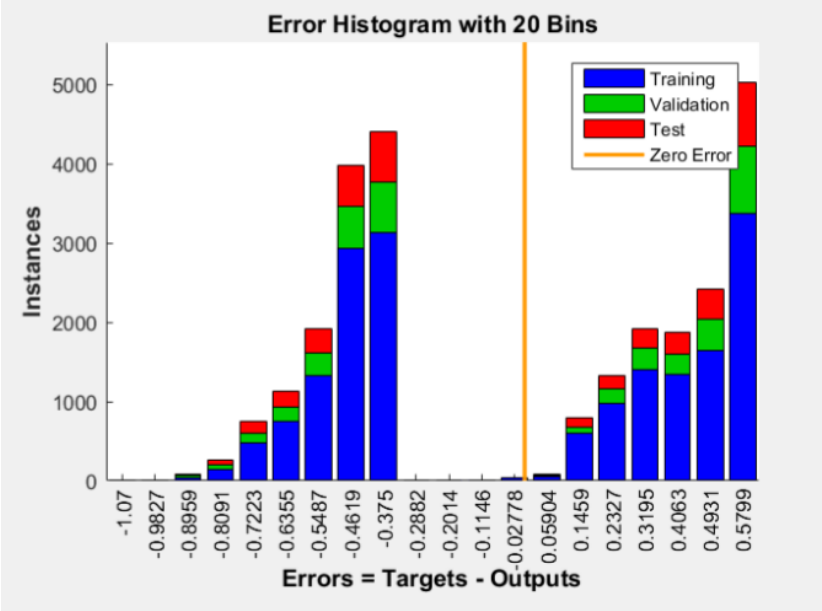
In this work, a VGG-like CNN structure is built. With 3 x 3 convolution filter size and 2 x 2 max pooling followed. The activation function is relu.

Table 1. Result of Machine Learning Techniques. Four of the above machine learning algorithms results are shown in the Table. Note that as the data is too large (500,000) for my computer memory to hold, only some of the data is randomly extracted for training and testing. Therefore, as the CNN performs not good both in two-class-classification and small amount of data, it is not fully done. Nonetheless, the codes of CNN (kares or matlab) can be run.

Methods	**Train**			**Test**		
	Precision	*Recall*	*Accuracy*	*Precision*	*Recall*	*Accuracy*
SVM	67.74%	72.37%	66.32%	48.78%	61.12%	49.96%
LR	65.54%	68.82%	65.10%	46.65%	55.73%	48.90%
RF	97.81%	88.50%	93.02%	47.46%	29.34%	51.19%
MLP	78.77%	82.31%	81.12%	52.26%	48.78%	51.76%



(a) (b)



(c)

Figure 2. The Result of the Machine Learning Algorithms. (a)The ROC curve of SVM. (b) The ROC curve of logstic regression. (c) The error historgram of MLP.

3. Conclusion

The whole process of the automatic particle picking process is performed based on machine learning techniques including deep learning. Although the result is actually not well enough, as precision and recall are both in a low level (0.5 and 0.6), the work is encouraging. Here is certain aspects which can be improved: 1) Use certain stronger algorithm to denoise the images. 2) Use data in a larger scale. 3) Use stronger machine learning algorithms with paramaters adjustment. 4) Extract feature in some other ways. By experiments, it is promising that the machine learning techniques will solve the time and work consuming particle picking work.

Acknowledgement

I give my acknowledgement to professor Zeng Jianyang's computational biology course and many resources on the Internet. Due to Zeng's teaching and many algorithms and tutorials found on the Internet, this project can be done smoothly and completely.

Reference

[1] R. C. Gonzalez and R. E. Woods, Digital image processing, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 2002.
[2] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[3] R. Collobert and S. Bengio (2004). Links between Perceptrons, MLPs and SVMs. Proc. Int'l Conf. on Machine Learning (ICML).
[4] Yifan Cheng, Nikolaus Grigorieff, Pawel A. Penczek, and Thomas Walz. A primer to single-particle cryo-electron microscopy. Cell, 161(3):438 – 449, 2015.
[5] Robert Langlois, Jesper Pallesen, Jordan T. Ash, Danny Nam Ho, John L. Rubinstein, and Joachim Frank. Au tomated particle picking for low-contrast macromolecules in cryo-electron microscopy. Journal of Structural Biology, 186(1):1 – 7, 2014.
[6] Wikipedia, deep learning tutorial.