# Lung Nodule Classification with Multi-model Contextual 3D Convolutional Neural Networks

Xiaotao Chen, Zhaoning Zhang, Zheng Qin, Hao Yu, Dongsheng Li
Parallel and Distributed Laboratory
Dept. of Computer Science and Engineering,
National University of Defense Technology

## Method Description:

For the false positive reduction track of the LUNA16 Challenge, we formulate it as a binary classification problem and have developed 3D convolutional neural network (3D CNN) models to solve it. we leverage multi-model contextual 3D CNNs to confront the challenge and effectively reduce the false positive rate. The method is inspired by the paper named *Multi-level Contextual 3D CNNs for False Positive Reduction in Pulmonary Nodule Detection*, which published by the champion of this competition. Our method is implemented with some improvements, for example, we added more architectures of the multi-scale object detector. We use 4 different scales with 20*20*6, 30*30*10, 30*30*18, and 40*40*26 in this submission.

## The pre-processing data stage:

1. According to the 754975 records in the candidate CV2.csv, the 3D data is intercepted in the original graph with different sizes: 20*20*6, 30*30*10, 30*30*18, and 40*40*26, which are used in the four architectures: arch0, arch1, arch2, arch3.
2. To solve the problem of data imbalance. 1) To expand the positive samples, according to the 1186 positive samples given in annonations.csv, the 3D data is intercepted in the original graph with four different sizes. 2) Down sampling the negative data when training.
3. We made ten train-test data sets combinations for the cross-validation, and trained the ten different models for each subset as the test set.

**The Training model stage:**

1. We train the four architectures with different dimensions of arch0: 20*20*6, arch1: 30*30*10, arch2: 30*30*18 and arch3: 40*40*26.
2. We trained each architectures with 20 epochs.
3. We extracted 10% train data for validation randomly. The three models trained by minimizing the cross-entropy loss with the optimizer named “adagrad”. The learning rate was initialized by default. And we set batch size to 100. Dropout were used during the training procedure. And Batch Normalization layer was added behind each convolutional layer and full connected layer. The method was implemented with Python based on the deep learning library of Keras.
4. We ensembled the final model with the ratio of arch0, arch1,arch2 and Arch3 for 0.2, 0.2, 0.2 and 0.4.