gaussianMLFitting.m

```
function [Means, Covs] = gaussianMLFitting(TrainSet)
% GAUSSIANMLFITTING Maximum Likelihood Multivariate Normal
 Distribution Fitting
% INPT: TrainSet: LxNxD matrix. A training set of L classes, N samples and D
 dimension.
% OUPT: Means: 1xL cell array. Each cell contains the Mean of a class.
%Covs: 1xL cell raary. Each cell contains the Covariance matrix of a class.

Means = cell(1, 1);
Covs = cell(1, 1);

for i = 1 : length(TrainSet)
    [SampleMean, SampleCov] = mvnML(TrainSet{i});
    Means{i} = SampleMean;
    Covs{i} = SampleCov;
end

end
```

```
function [sMean, sCov] = mvnML(dataSet)
% MVNML Get Mean and Cov of Multivariate Normal Dist of a dataset

setSize = size(dataSet, 1);

% Get mean of each column - 1xD
sMean = sum(dataSet,1) ./ setSize;

% Calculate covariance matrix %

% You can do it according to the formula
% For some reason this approach takes a lot of time
dataSetSubMean = double(dataSet)-repmat(sMean,size(dataSet,1),1);
sCov = (dataSetSubMean.' * dataSetSubMean);

% Another way is to use cov(double(dataSet)) directly
% sCov = cov(double(dataSet));

% Another one
% sCov = zeros (size(dataSet,2), size(dataSet,2));
% for i = 1 : setSize
%     M = double(dataSet(i, :)) - sMean;
%     M = M' * M;
%     sCov = sCov + M;
% end

if setSize > 1
     sCov = sCov ./ (setSize-1);
end

end
```