%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Implementation details

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

The code is to test a residual network with adaptive content suppression for image steganalysis. The implementation is based on the MatConvNet platform.

The residual network is a CNN model equipped with many batch normalization layers. In our implementation, each testing image accompanied with 20 cover images and their stego images forms a batch and is fed into the trained model. The model then gives a predicted label for the testing image. The images for accompanying testing images are stored in the "batch\_images" folder. There are totally 600 cover images and their stego images (suniward, wow, hill, mipod stego images) in this folder. Thus, the code can generate 30 predictions for each testing image. According to our experimental result, detection accuracies for four adaptive steganographic algorithms are listed as follows (10000 BOSS testing images, 50% are cover images and 50% are stego images):

-- wow\_04: 92%~95%

-- suniward\_04: 88% ~ 91%

-- hill\_04: 87% ~ 92%

-- mipod\_04: 92% ~ 94%

Based on 30 results, you can use the voting method to predict the label of your images. According to our test, voting can bring 3% ~ 5% performance improvements.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Files

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

**test\_resnet**: this is the main function, which can be used to test a trained model. In current version, we have provided four well trained models,including suniward, wow, hill, and mipod at 0.4 bpp, which are trained on BOSS dataset (models are in the "trained\_models" file).

**cnn\_steganalysis\_setup\_data**: the function to determine testing images.

**cnn\_train\_dag\_check**: the function to validate the accuracy of the model.

**getBatchFn, getDagNNBatch**: the function to read images from specified paths.

**setup**: the function to setup environment for the proposed model.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Folders

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

**dependencies**: this folder contains basic functions of constructing a CNN model with the MatConvNet platform. It contains two sub-folders, i.e. matconvnet and vlfeat. These files can be downloaded from following links:

vlfeat: http://www.vlfeat.org/

matconvnet: http://www.vlfeat.org/matconvnet/

**test\_images**: the images for testing.

**predictions**: the folder saves the prediction results for your given images. '1' represents the cover image, '2' represents the stego image.

**batch\_images**: the folder contains 600 cover images and their suniward, wow, hill, mipod stego images. When you test the model, please assure that the accompanying images are consistent with the trained model.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Variable

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

**multi.mat**: the external variable to determine which images are used for accompanying the input testing image. It selects images with index ranging from (multi-1)\*20+1 to multi\*20 for accompanying a testing image. When run the code, please reset this value to 1 initially.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Hints

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Before using the code, you should first compile the function “vl\_compilenn” under the folder dependencies\matconvnet\matlab to generate many mex64 files.

Our models are trained based on the cropped version of the BOSS dataset: the original 512x512 image are cropped into 4 non-overlapping 256x256 images. When you test the model, please make sure that the size of input image is 256x256.

We have provided an example for you to test. In the folder “test\_images”, there are 10 images and their true labels. You can run the file “test\_resnet”, which can generate 10 predicted labels in the “predictions” folder.