# README file for the '`RMT-MTLLSSVM`' Matlab and Julia Package

**Abstract**

This document explains how to use the code implementing the Random Matrix Improved Multi-Task Learning Least Square Support Vector Machine (RMT-MTLLSSVM) proposed in the article '`Deciphering and Optimizing Multi-Task Learning: a Random Matrix Approach`'.

## 1 For Matlab users

### 1.1 Archive content

- The function implementing the multi class extension is called `RMTMTLLSSVM_train.m` which trains the MTL LS-SVM proposed algorithm.
- The function implementing the binary classification is called `MTLLSSVMTrain_binary.m` which trains the MTL LS-SVM proposed binary algorithm.
- The main script comparing all algorithms for synthetic data/real data is `CompareMTL.m` for the multi class classification for SURF-BOW features.
- The main script comparing all algorithms for synthetic data/real data is `CompareMTL2.m` for the multi class classification for VGG features.
- The main script illustrating the binary classification with fixed probability of false alarm is `PFA.m`.
- Folder `utils`: containing alternative MTL algorithms among which MMDT algorithm, CDLS algorithm, ILS algorithm and LS-SVM on source or target[1] and other functions used for the proposed method.
- Folder `datasets`: containing Office+Caltech dataset Mit-Bih dataset and Mnist dataset.

[1] To use these codes, one needs to have a Matlab compiler for Mex files

### 1.2 Code `binary_experiments`

The different options proposed to execute the script `binary_experiments` are as follows:

- The training size $n_s$
- The test size *nst*
- The hyperparameters $\lambda$ and $\gamma$ and the value of $\beta$ relating the tasks.

### 1.3 Code `CompareMTL.m` and `CompareMTL2.m`

The different options proposed to execute the script `CompareMTL.m` and `CompareMTL2.m` comparing the different Multi Task algorithms are as follows:

- **"dataset"** to be chosen as 'a-d' such that a (for the source) and d (for target) are chosen between *A for Amazon*, *Ca for Caltech*, *D for DSLR*, *W for Webcam*. For illustration, for a MTL with source task Amazon and target task Caltech, the setting is "dataset='A-Ca'"
- **"number_trials"** which represents the number of trials to be tested for each dataset in order to average performances.

### 1.4 Code `PFA.m`

The different options proposed to execute the script `PFA.m` are as follows:

- **"data"** to be chosen between *'synthetic'* and *'real'* to test the binary classification with fixed probability of false alarm either on synthetic data or real data.
- **"n_training_sample"** is the number of training examples to be sampled from the overall training set.
- Parameters for synthetic data (the dimension of the sample $\mathbf{p}$, the number of samples of each class of the 2 tasks $[n_{11}, n_{12}, n_{21}, n_{22}]$ as well as the mean of each class of each task $[\mu_{11}, \mu_{12}, \mu_{21}, \mu_{22}]$).

## 2 For Julia users

### 2.1 Archive content

- The main script for experiments on binary classification is `binary_classification.jl`.
- The main script comparing the classical/improved MTL-LSSVM algorithm and the for synthetic data/real data is `CompareMTL.jl` for the multi class classification for SURF-BOW and VGG features.

- The main script illustrating the binary classification with fixed probability of false alarm is `PFA.jl`.
- The file `utils`: containing the function implementing the proposed MTL LS-SVM algorithm and some other function for generating the data, etc.

### 2.2 Code `CompareMTL.jl`

The various options proposed to execute the script are:

- The `data_type` among which we can choose `"OFFICE"`, `"OFFICE_VGG"` or `"synthetic"`.
- **"dataset"** to be chosen as 'a-d' such that a (for the source) and d (for target) are chosen between *A for Amazon*, *Ca for Caltech*, *D for DSLR*, *W for Webcam*. For illustration, for a MTL with source task Amazon and target task Caltech, the setting is "dataset='A-Ca'"

Note that the script doesn't contain the implementation of alternative methods (MMDT, CDLS, ILS). Their implementation can be found in the matlab script.

## References