USER MANUAL

Summary

Edit Dataset

- **EditDataSet.m**
  This script lets to edit an RGB Data Set converting it obtaining a 3D Grayscale, "Haematoxylin" or Reinhard Normalized new Data Set.
  Running this script first of all two windows for the selection of the training and test set folders will be opened as shown in Figure 1.

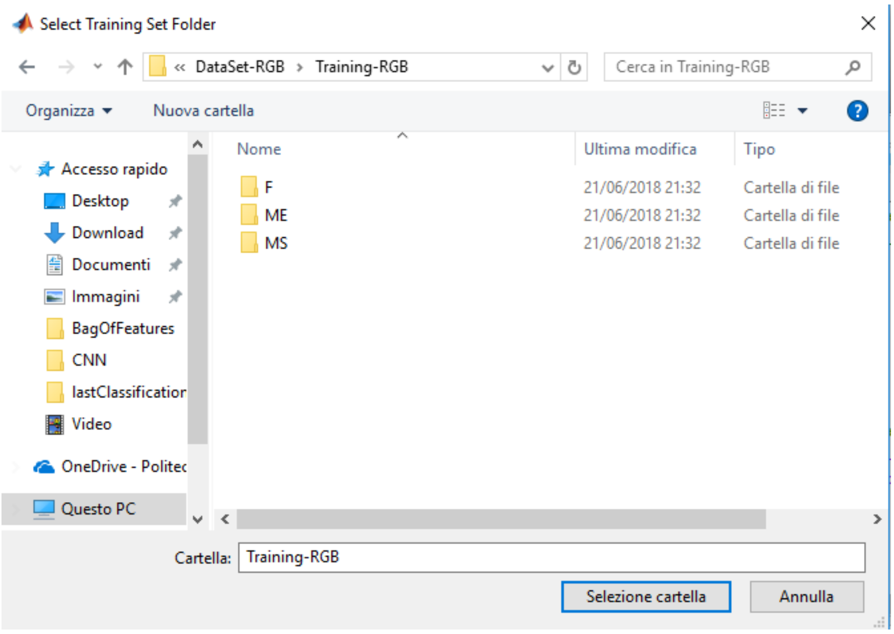


Figure 1

The directory tree structure must be as the one shown in *Figure 2* (Main Training (or Test) Folder → Categories Folders → Images).

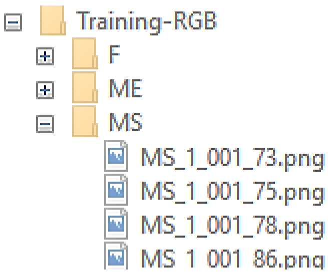


Figure 2

Select then the color for the new Data Set in the menu shown in Figure 3.

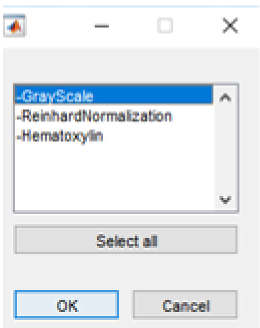


Figure 3 - Menu

Wait the end of the process (Figure 4).

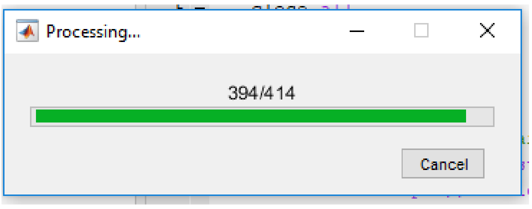


Figure 4 – Processing image

- **Examples**

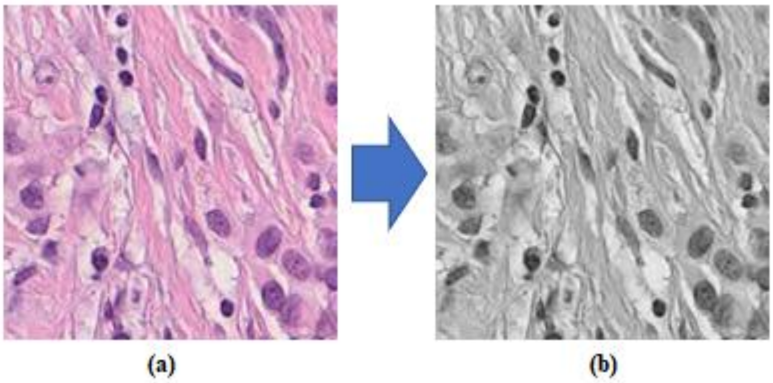


Figure 6 - RGB to 3D Grayscale: (a) RGB, (b) 3D Grayscale

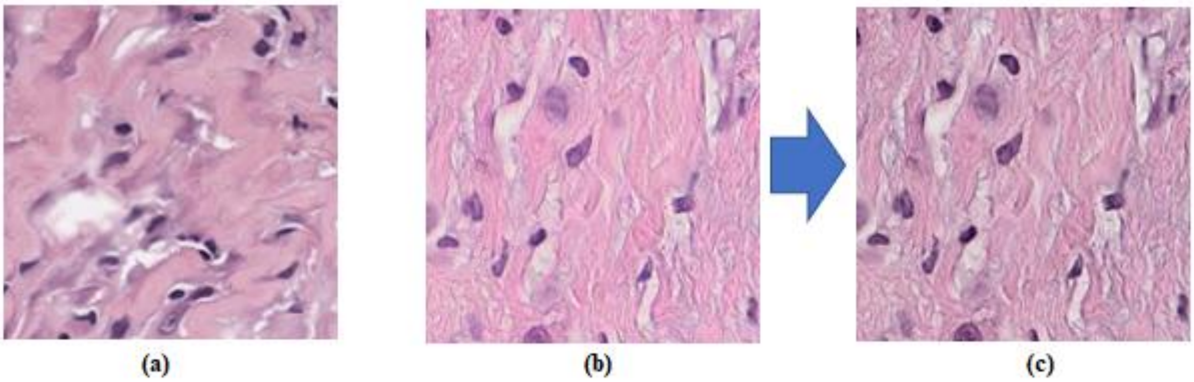


Figure 5 - Reinhard Normalization: (a) Target image, (b) Source image, (c) Normalized image

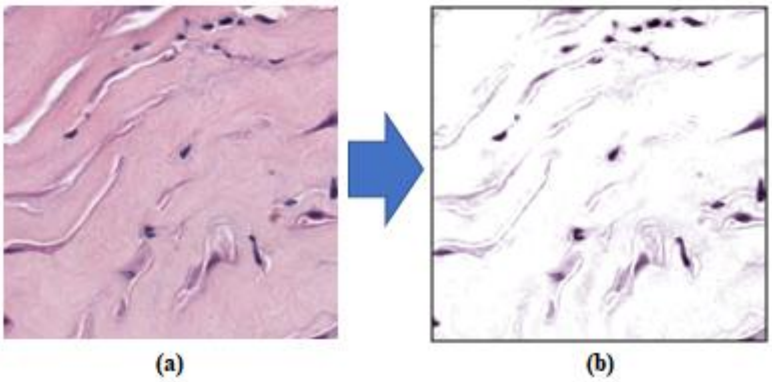


Figure 7 - RGB to "Hematoxylin": (a) RGB, (b) "Hematoxylin"

Features Extraction using Pre-trained CNN

- **FeaturesExtraction_GUI.py**
  Run this file to open the window that is shown in *Figure 1*.

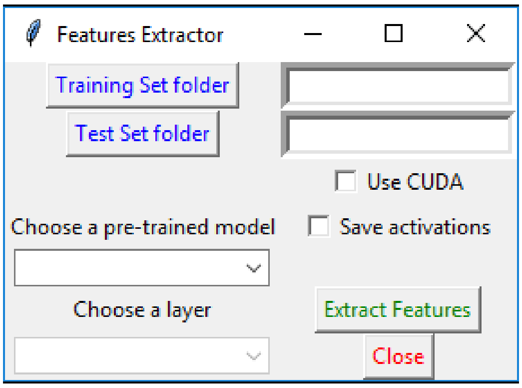


Figure 8 – Main window

- **Training Set folder**
  Click on this button to select the folder containing the images for the training set as shown in the example in *Figure 1*. The directory tree structure must be as the one shown in *Figure 2* (Main Training Folder → Categories Folders → Images). The path can't contain spaces.

- **Test Set folder**
  Click on this button to select the folder containing the images for the test set as shown in the example in *Figure 1*. The directory tree structure must be as the one shown in *Figure 2* (Main Test Folder → Categories Folders → Images). The path can't contain spaces.

- **Use CUDA**
  Tick it, as shown in *Figure 9*, to use GPU for the Features Extraction through CUDA drivers. The installation of the latest is needed to tick this tick. If the tick is off only CPU will be used.

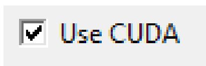


Figure 9 – Use CUDA ON

- **Save activations**
  Tick it, as shown in *Figure 10*, to save activations in Results/Activations as shown in *Figure 11*.

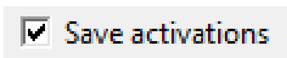


Figure 10 – Save activations ON

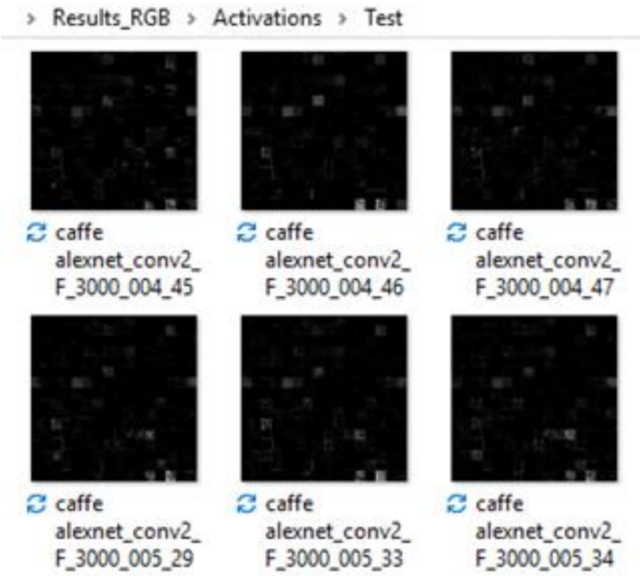


Figure 11 – Example of activations

- **Choose a pre-trained model**
  Choose a pre-trained model as shown in *Figure 12*. The features will be extracted using this pre-trained model through its weigths and its architectures.

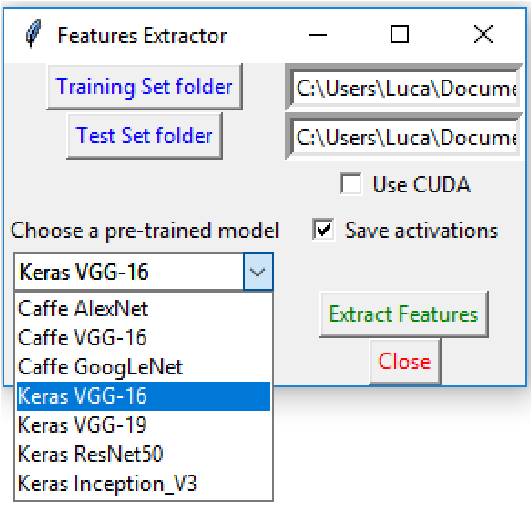


Figure 12 – Choice of ***a pre-trained model***

- **Choose a layer**
  Choose a layer from the avaiable layers for the selected pre-trained model as shown in *Figure 13*. The extracted features will be the output of this layer.

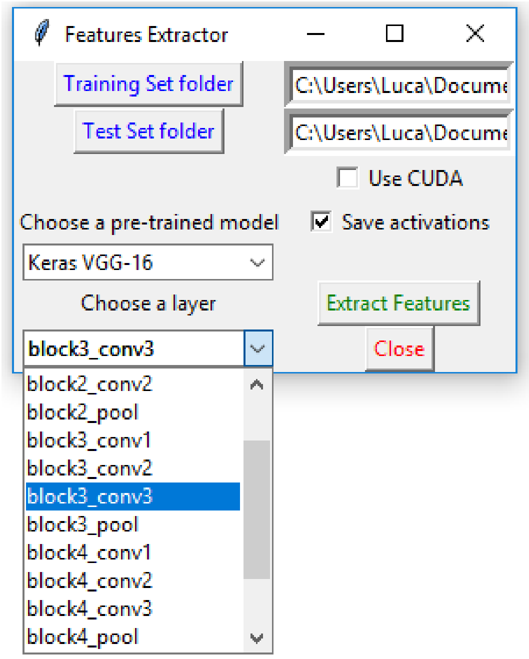


Figure 13 – Choice of a layer

- **Extract Features**
  Click on this button to open the window shown in *Figure* 14.

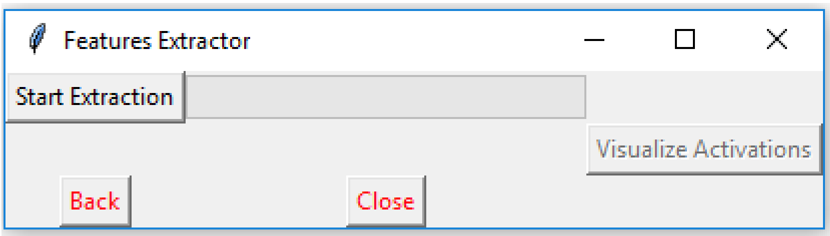


Figure 14

- **Start Extraction**
  Click on this button to start the features extraction. In *Figure 15* a running features extraction is shown.

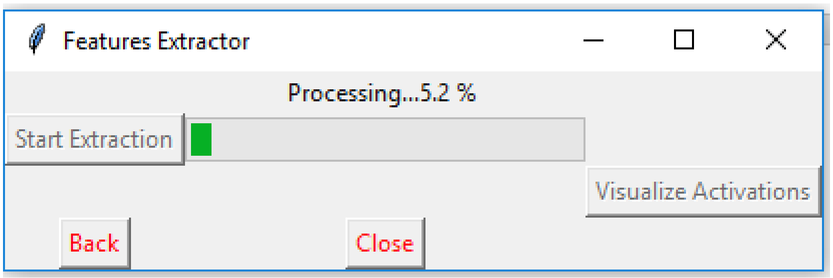


Figure 15 - A features extraction running process

- **Visualize Activations**
  At the end of the features extraction process, this button will be enabled, as shown in *Figure 16*, if the “Save activations” tick was set ON. Clicking on this button it is possible to visualize the activations as shown in *Figure 17*. Click on + button to zoom.

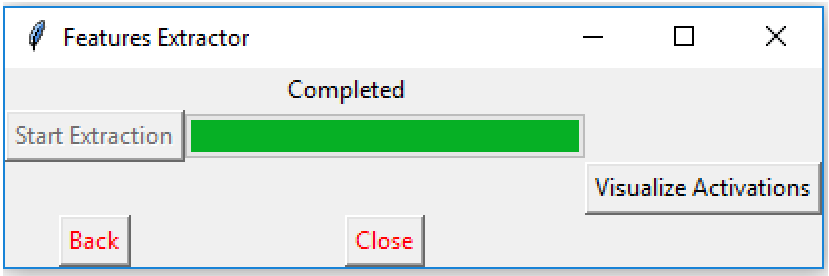


Figure 16 – Features extraction completed

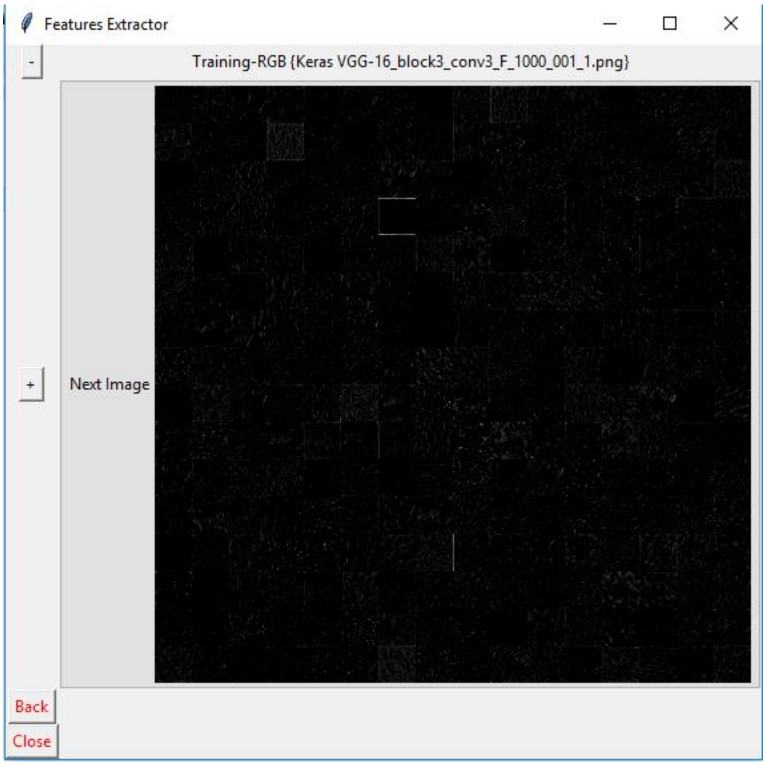


Figure 17 – Example of activation

- **Results**
  The files containing the extracted features are saved in Results/Features folder with names <Model>_<Layer>_<Training or Test main folder>.h5 as shown in *Figure 18*.
  It is possible to open these files using the MATLAB script **h5reader.m**.

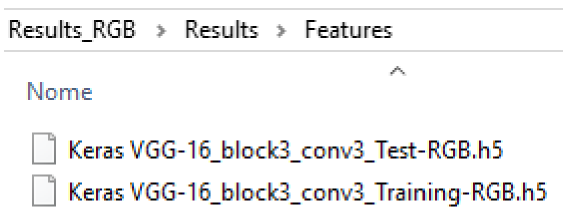


Figure 18

Dimensionality Reduction

- **DimensionalityReduction_GUI.py**
  Run this file to open the window that is shown in *Figure 19*.

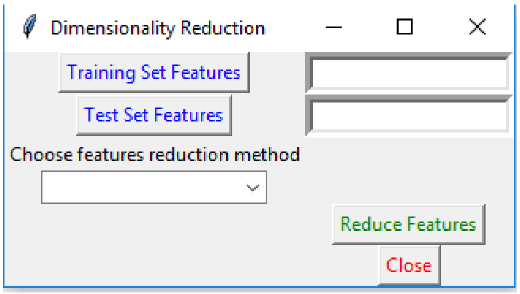


Figure 19

- **Training Set Features**

  Click on this button to select the .h5 file of the training features to reduce. The path can't contain spaces.
  It must contain at least the following datasets:
    - 'feats': array-like, shape (n_samples, n_features)
    - 'labels': array-like, shape (n_samples,)
    - 'img_ids': array-like, shape (n_samples,)

  A file with these characteristics is the output of FeaturesExtraction_GUI.py

- **Test Set Features**

  Click on this button to select the .h5 file of the test features to reduce. Its structure must be like the training features .h5 file. The path can't contain spaces.

- **Choose features reduction method**
  Choose a features reduction method. The avaiable methods are shown in *Figure 20*.

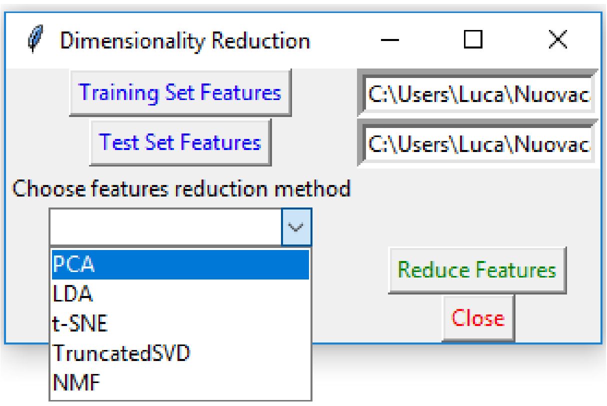


Figure 20

- **Reduce Features**
  Clicking on this button user will be redirected to the proper parameters page. Pages for each method are shown in Figure 21. Click on the blue text to get more information about parameters.

Clicking on Start Reduction button, the dimensionality reduction will be performed.

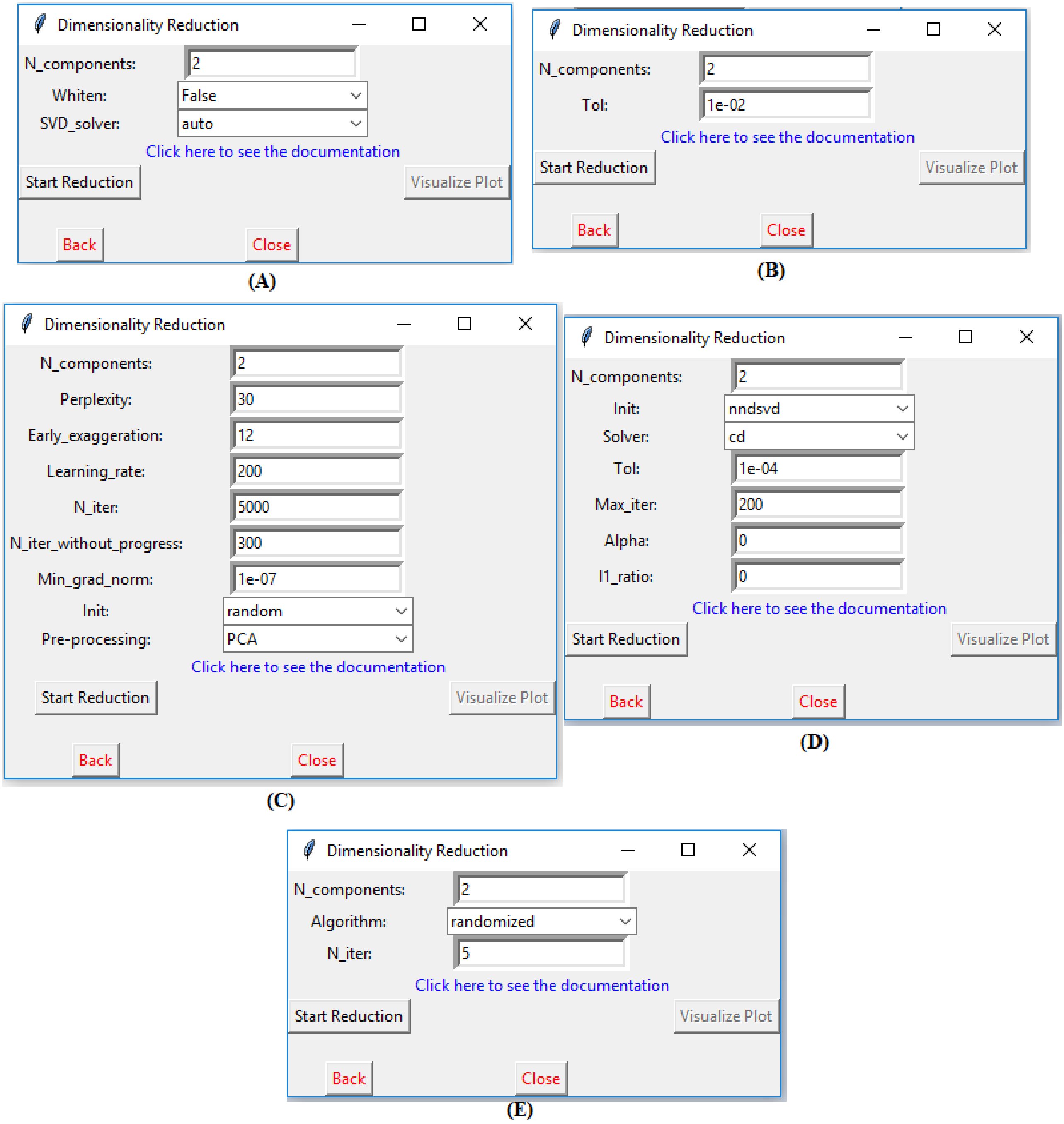


Figure 21 - PCA (A), LDA (B), t-SNE (C), NNMF (D), TruncatedSVD (E)

- **Visualize Plot**

  If the selected number of components is minor than 4, Visualize Plot button will be enabled. Clicking on this button a scatter plot of the reduced features will be shown. This plot will be saved in Results/Plots directory.

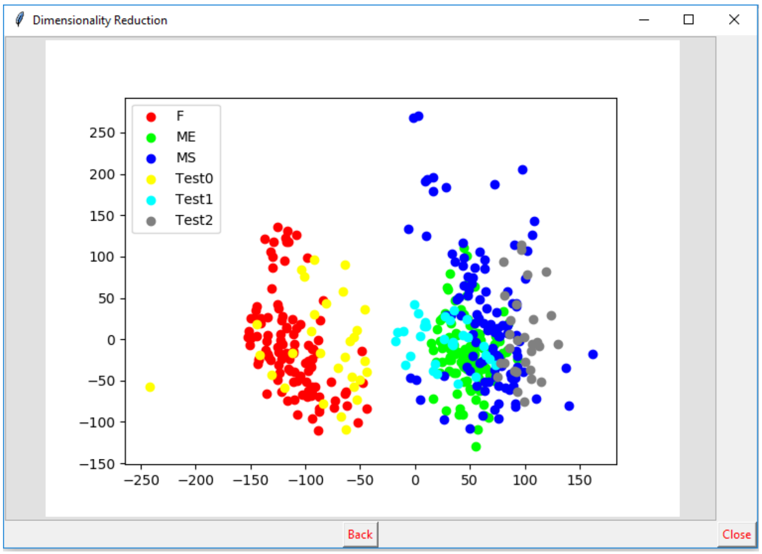


Figure 22 – Example of visualization of reduced features

- **Results**
  The files containing the reduced features are saved in Results/ReducedFeatures folder with names <Dimensionality reduction method><n_components>_<Model>_<Layer>_<Training or Test main folder>.h5.
  It is possible to open these files using the MATLAB script **h5reader.m**.

Classification

- **Classification_GUI.py**
  Run this file to open the window that is shown in *Figure 23*.

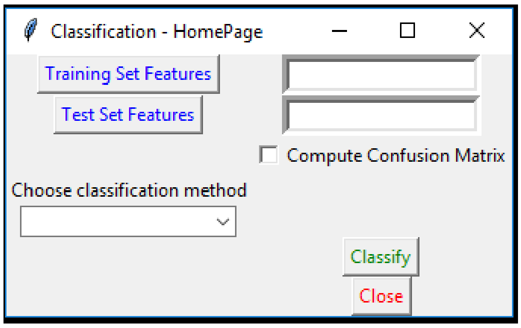


Figure 23 – HomePage Classification_GUI

- **Training Set Features**

  Click on this button to select the .h5 file of the training features to reduce.
  It must contain at least the following datasets:
    - One between 'pca', 'tsne', 'tsvd', 'lda', 'nmf', 'feats': array-like, shape (n_samples, n_features)
    - 'labels': array-like, shape (n_samples,)
    - 'img_ids': array-like, shape (n_samples,)

  A file with these characteristics is the output of DimensionalityReduction_GUI.py or FeaturesExtraction_GUI.py.
  The path can't contain spaces.

- **Test Set Features**

  Click on this button to select the .h5 file of the test features to reduce. Its structure must be like the training features .h5 file. The path can't contain spaces.

- **Compute Confusion Matrix**

  Tick this checkbox if you are performing a supervised classification, otherwise it doesn't make sense.

- **Choose classification method**
  Choose a classification method. The avaiable methods are shown in *Figure 24*.

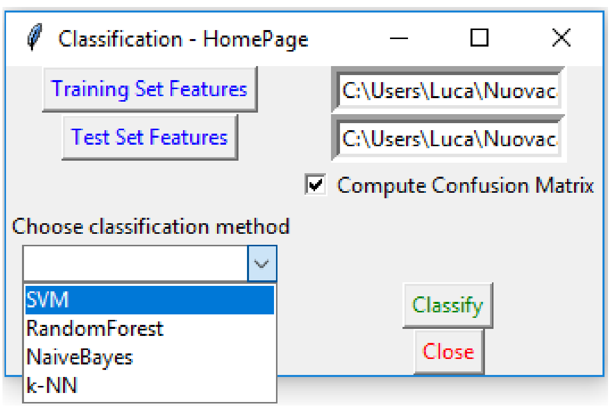


Figure 24 – Select classification method

- **Classify**

  Clicking on this button user will be redirected to the proper classifier parameters page. Pages for each classification method are shown in Figure 25. Click on the blue text to get more information about parameters.

  It is possible to perform tuning on red surrounded parameters typing many values comma separated.

  Clicking on Start Classification button, the classification will be performed.

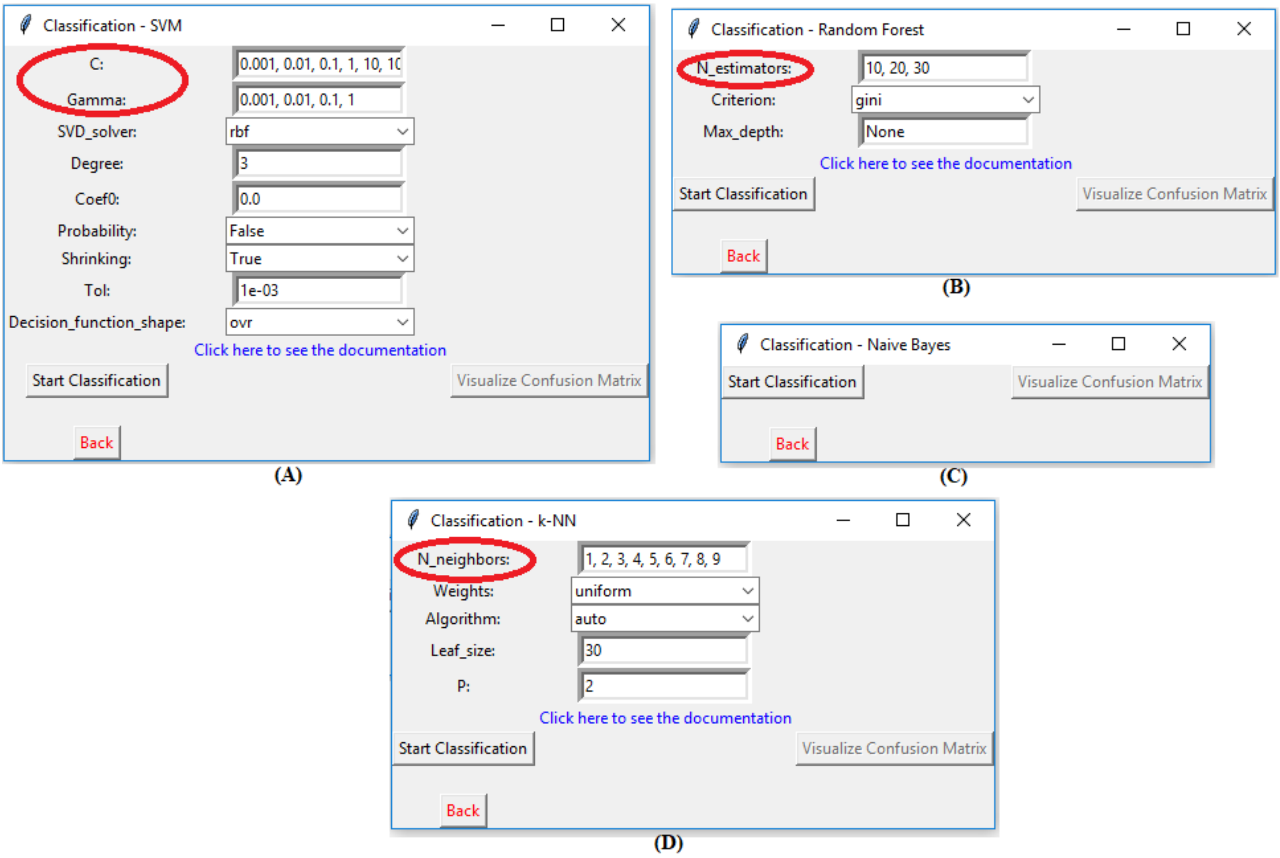


Figure 25 - SVM (A), Random Forest (B), Naive Bayes (C), k-NN (D)

- **Visualize Confusion Matrix**

  If Compute Confusion Matrix checkbox is ticked, Visualize Confusion Matrix button will be enabled at the end of the classification process. Click on it to visualize the computed confusion matrix. The shown figure will also be saved in Results\Classification \ConfusionMatrix directory.

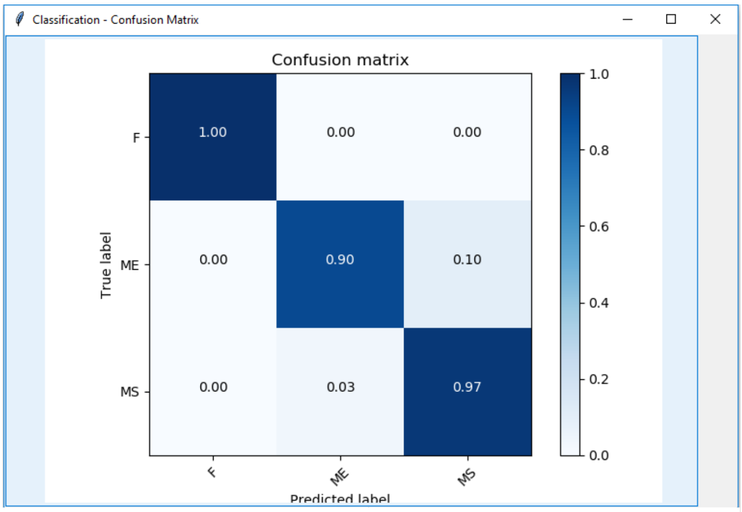
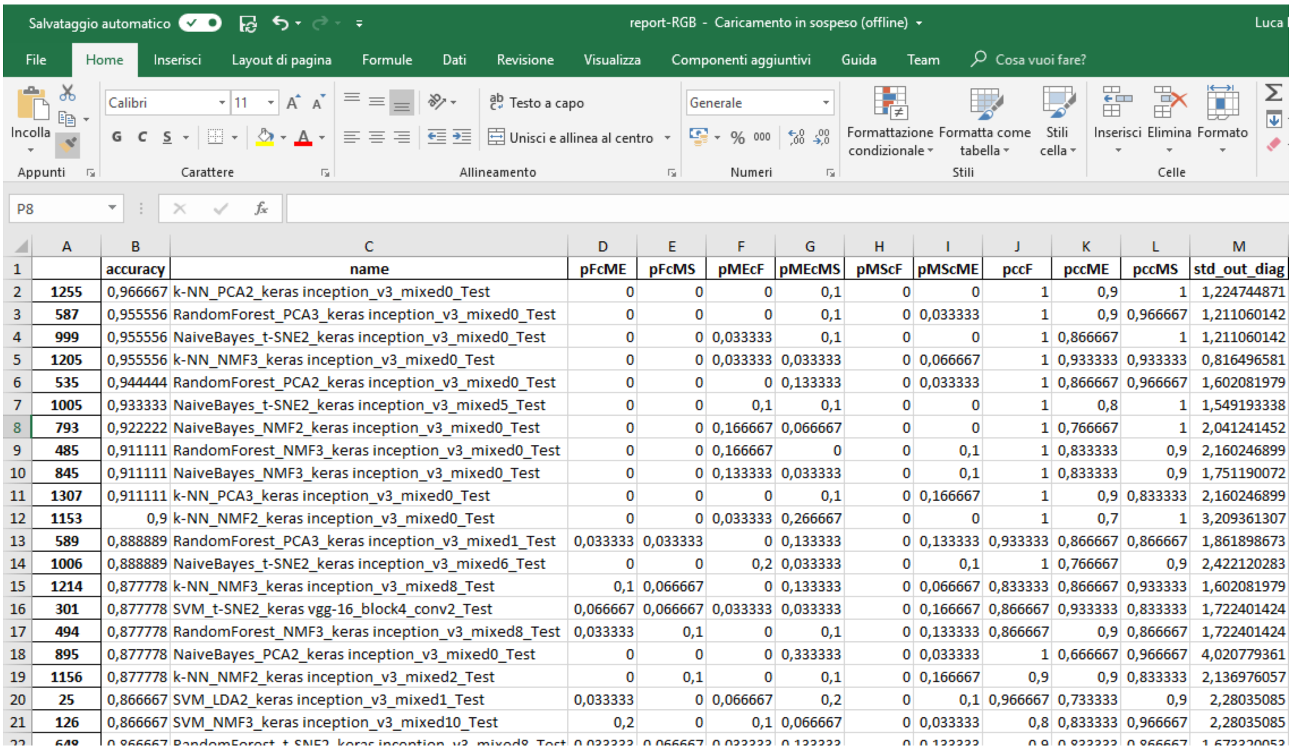


Figure 26 – Example of confusion matrix

- **Results**
  The file containing the results of the classification is saved in Results/Predictions folder with names <Classification method>_<Dimensionality reduction method><n_components>_<Model>_<Layer>_<Training or Test main folder>.xlsx.

- **Analize results of classification**
  Run ConfusionMatrixAnalyzer.py and select the JSON file that was generated during the classification process in Results\Classification\ConfusionMatrix folder.

  A report.xlsx file will be generated.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1		accuracy	name	pFcME	pFcMS	pMEcF	pMEcMS	pMScF	pMScME	pccF	pccME	pccMS	std_out_diag
2	1255	0,966667	k-NN_PCA2_keras inception_v3_mixed0_Test	0	0	0	0,1	0	0	1	0,9	1	1,224744871
3	587	0,955556	RandomForest_PCA3_keras inception_v3_mixed0_Test	0	0	0	0,1	0	0,033333	1	0,9	0,966667	1,211060142
4	999	0,955556	NaiveBayes_t-SNE2_keras inception_v3_mixed0_Test	0	0	0,033333	0,1	0	0	1	0,866667	1	1,211060142
5	1205	0,955556	k-NN_NMF3_keras inception_v3_mixed0_Test	0	0	0,033333	0,033333	0	0,066667	1	0,933333	0,933333	0,816496581
6	535	0,944444	RandomForest_PCA2_keras inception_v3_mixed0_Test	0	0	0	0,133333	0	0,033333	1	0,866667	0,966667	1,602081979
7	1005	0,933333	NaiveBayes_t-SNE2_keras inception_v3_mixed5_Test	0	0	0,1	0,1	0	0	1	0,8	1	1,549193338
8	793	0,922222	NaiveBayes_NMF2_keras inception_v3_mixed0_Test	0	0	0,166667	0,066667	0	0	1	0,766667	1	2,041241452
9	485	0,911111	RandomForest_NMF3_keras inception_v3_mixed0_Test	0	0	0,166667	0	0	0,1	1	0,833333	0,9	2,160246899
10	845	0,911111	NaiveBayes_NMF3_keras inception_v3_mixed0_Test	0	0	0,133333	0,033333	0	0,1	1	0,833333	0,9	1,751190072
11	1307	0,911111	k-NN_PCA3_keras inception_v3_mixed0_Test	0	0	0	0,1	0	0,166667	1	0,9	0,833333	2,160246899
12	1153	0,9	k-NN_NMF2_keras inception_v3_mixed0_Test	0	0	0,033333	0,266667	0	0	1	0,7	1	3,209361307
13	589	0,888889	RandomForest_PCA3_keras inception_v3_mixed1_Test	0,033333	0,033333	0	0,133333	0	0,133333	0,933333	0,866667	0,866667	1,861898673
14	1006	0,888889	NaiveBayes_t-SNE2_keras inception_v3_mixed6_Test	0	0	0,2	0,033333	0	0,1	1	0,766667	0,9	2,422120283
15	1214	0,877778	k-NN_NMF3_keras inception_v3_mixed8_Test	0,1	0,066667	0	0,133333	0	0,066667	0,833333	0,866667	0,933333	1,602081979
16	301	0,877778	SVM_t-SNE2_keras vgg-16_block4_conv2_Test	0,066667	0,066667	0,033333	0,033333	0	0,166667	0,866667	0,933333	0,833333	1,722401424
17	494	0,877778	RandomForest_NMF3_keras inception_v3_mixed8_Test	0,033333	0,1	0	0,1	0	0,133333	0,866667	0,9	0,866667	1,722401424
18	895	0,877778	NaiveBayes_PCA2_keras inception_v3_mixed0_Test	0	0	0	0,333333	0	0,033333	1	0,666667	0,966667	4,020779361
19	1156	0,877778	k-NN_NMF2_keras inception_v3_mixed2_Test	0	0,1	0	0,1	0	0,166667	0,9	0,9	0,833333	2,136976057
20	25	0,866667	SVM_LDA2_keras inception_v3_mixed1_Test	0,033333	0	0,066667	0,2	0	0,1	0,966667	0,733333	0,9	2,28035085
21	126	0,866667	SVM_NMF3_keras inception_v3_mixed10_Test	0,2	0	0,1	0,066667	0	0,033333	0,8	0,833333	0,966667	2,28035085

Figure 27 – Example of report.xlsx

Features Reduction using Bag of Features and classification using SVM

- **BagOfFeatures.m**
  Run this script to extract a bag of features from a Training Set, encode the images of the Test Set using this bag and finally classify Test Set images using SVM.

  First, select the parameters shown in Figure 28.

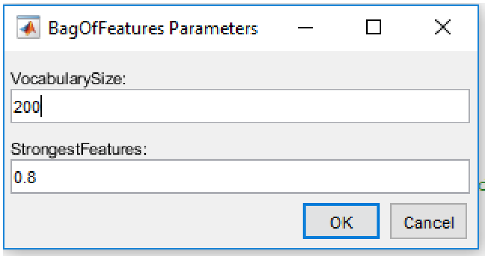


Figure 28 – Parameters

  Then, select Training Set folder. The directory tree structure must be as the one shown in *Figures 1 and 2* (Main Training Folder → Categories Folders → Images). The path can't contain spaces.
  Do the same for Test Set folder.
  Wait the end of all processes.

- **Results**
  Predictions and other results will be saved in Results VocabularySize_<VocabularySize> folder in a file called Results-<DataSet name>.mat.

- **Examples of features vector histograms and words on images**
  Some of these examples will be saved in

  Results VocabularySize_<VocabularySize>\Examples\<DataSet name> folder.

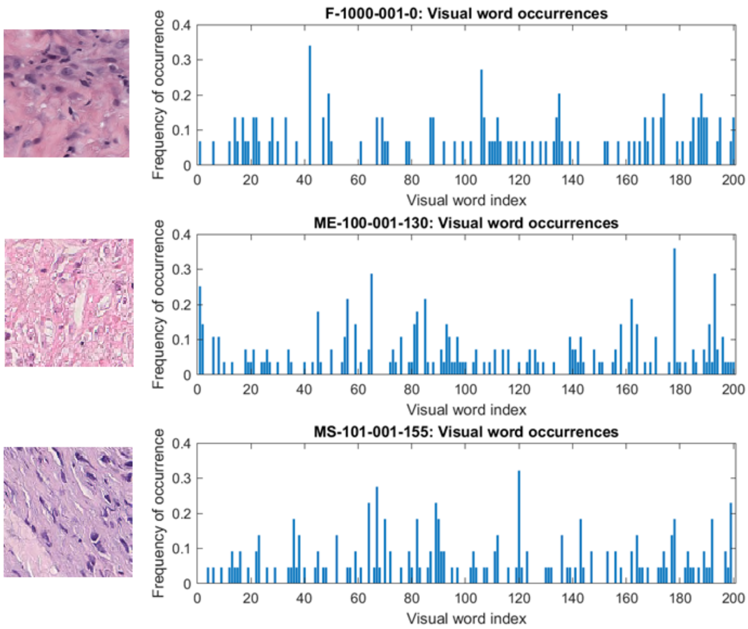


Figure 29 - Features vector histograms

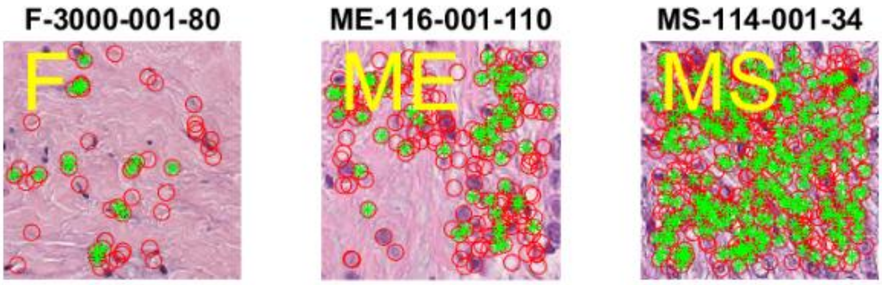
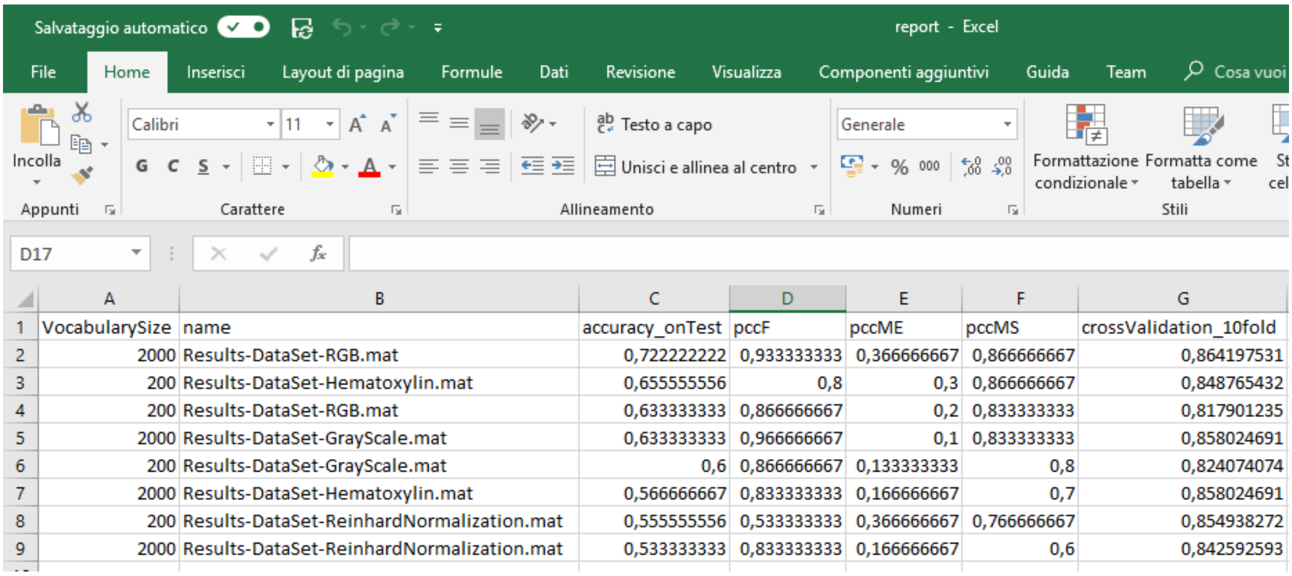


Figure 30 – Words on images

- **Analize results of classification**

  Run AnalyzeResults.m. A report.xlsx file will be generated.



	A	B	C	D	E	F	G
1	VocabularySize	name	accuracy_onTest	pccF	pccME	pccMS	crossValidation_10fold
2	2000	Results-DataSet-RGB.mat	0,722222222	0,933333333	0,366666667	0,866666667	0,864197531
3	200	Results-DataSet-Hematoxylin.mat	0,655555556	0,8	0,3	0,866666667	0,848765432
4	200	Results-DataSet-RGB.mat	0,633333333	0,866666667	0,2	0,833333333	0,817901235
5	2000	Results-DataSet-GrayScale.mat	0,633333333	0,966666667	0,1	0,833333333	0,858024691
6	200	Results-DataSet-GrayScale.mat	0,6	0,866666667	0,133333333	0,8	0,824074074
7	2000	Results-DataSet-Hematoxylin.mat	0,566666667	0,833333333	0,166666667	0,7	0,858024691
8	200	Results-DataSet-ReinhardNormalization.mat	0,555555556	0,533333333	0,366666667	0,766666667	0,854938272
9	2000	Results-DataSet-ReinhardNormalization.mat	0,533333333	0,833333333	0,166666667	0,6	0,842592593

Figure 31 – Example of report.xlsx