

scene_class

The following is a *.md file and there are syntax insertions that may be misread as content.

To view the file correctly, please go to https://github.com/chrizandr/scene_class

The following are the dependencies required before the code can be properly executed: Each dependency is stated alongside the module that is dependent on it Parts of the code may be run without installing all the dependencies. The only dependency that all the modules need is the Python interpreter.

Python 2.7.x

OpenCV 3.1 : feature_extract, normalize, Gabor

scikit-learn 0.16.1 : LDA

python-numpy 1.11.2 : feature_extract, normalize, Gabor, segment

python-weka-wrapper 0.3.9 : classifier

Instructions for each of the modules are now explained in detail:

feature_extract : The module accesses the dataset and extract features from each image in the dataset. The dataset can be downloaded from <http://cvcl.mit.edu/database.htm> It is advised that all the .zip files that are downloaded be extracted in a single folder. The module is programmed to access the "base_dir" folder on the system where each of the dataset folders are located. The folder names of the dataset are stored in the "sub_dirs" variable. The values are the default names of the folders. Any change in the names must also be changed accordingly in the module. The package by default has a "data/" folder where the results of the feature extraction process are stored in "data.csv" The name of the file can be changed by modifying the "name" variable in the module.

Gabor, segment, normalize : These modules do not perform any function on their own, but are used by the feature_extract module and must be kept in the same package as that of the feature_extract module. The __init__.py file specifies the folder to be treated as a python package.

LDA : The LDA module performs Linear Discriminant analysis for data subspace reduction for a labeled set of features. The module uses the shuffled data present in the *_r.csv file. The transformed data is written into the *_r_LDA.csv file in the same folder. The path for both the files is specified in the "path" variable.

csv2arff : The weka library uses a different data representation than other machine learning libraries, The weka module uses the .arff format for storing and retrieving data. Although it is possible to use .csv files as well, the library recognizes the .arff format more efficiently. The csv2arff module, takes a .csv file and converts it into the corresponding .arf file. All values are treated as numeric and the last entry of every row is assumed to be the class label, although this can be easily changed by modifying the code a little.

classifier : The weka library is a well known data mining and machine learning library with a full set of tools for immediate research applications without much set up. The library is written in java and as such the code uses a Java Virtual Machine to emulate the classifier. The module uses data from the .arff for classification The variable "data_dir" is used to specify the path to the folder containing the data file and the "filename" variable is used to specify the name of the .arff file. **shuffle** : The shuffle module was necessary to enable cross-validation evaluation of the data. The module takes the *.csv file stored in the "filename" variable which is located in the "path" filepath. The shuffled data is stored in the same filepath under the *_r.csv file

The classification process can be invoked by execution of the classifiers module directly. There is precomputed data present in the "data/" folder that can be used for classification.

The order in which execution of the Python files was done to extract the features are:

feature_extract: To extract the features (feature_extract uses Gabor, segment and normalize to extract the features) **LDA**: to transform the extracted features according to the best linear combination. **shuffle**: to shuffle the computed data to enable 10-fold cross-validation of the resultant data **classifier**: to classify the data according to Nearest Neighbour Classifier with 10 fold

cross-validation

This project is a joint effort by Team 1 for the DIP course at IIIT Sri City.
