Categorizing images using Self-Organizing Map

1.0 About

This project is to categorize wallpapers in to two groups, beaches and forests. This uses a self-organizing map to categorize images into two categories.

2.0 Implementation

The SOM uses 23 classes (can be changes) to categorize a wallpaper. 14 of those classes are for beaches and 9 classes are for forests. The SOM starts with random weights. Then it is trained using images. First when a new image is entered the SOM tries to find the node which gives the maximum output (weighted sum). Then if that node is not already allocated for a class it change its weights to give more larger value as output. If the node is already allocates it take another node update its weights.

The input dataset is created using the RBG pixel values of a image resized in to 80x60. That means that input dataset has 80x60x = 14400. It is also the size of the weight vector of a node.

More information on SOM, http://en.wikipedia.org/wiki/Self-organizing_map

3.0 Pre-Requirements

This program is written using 'Python'. It need 'Python', 'PyQt', and 'pySide' to run this scripts.
Use Synaptic package manager in Linux to install them.

4.0 Running

This contains three main programs. First we need to run 'trainer.py' to train the SOM. Use,

python trainer.py

to run the program.

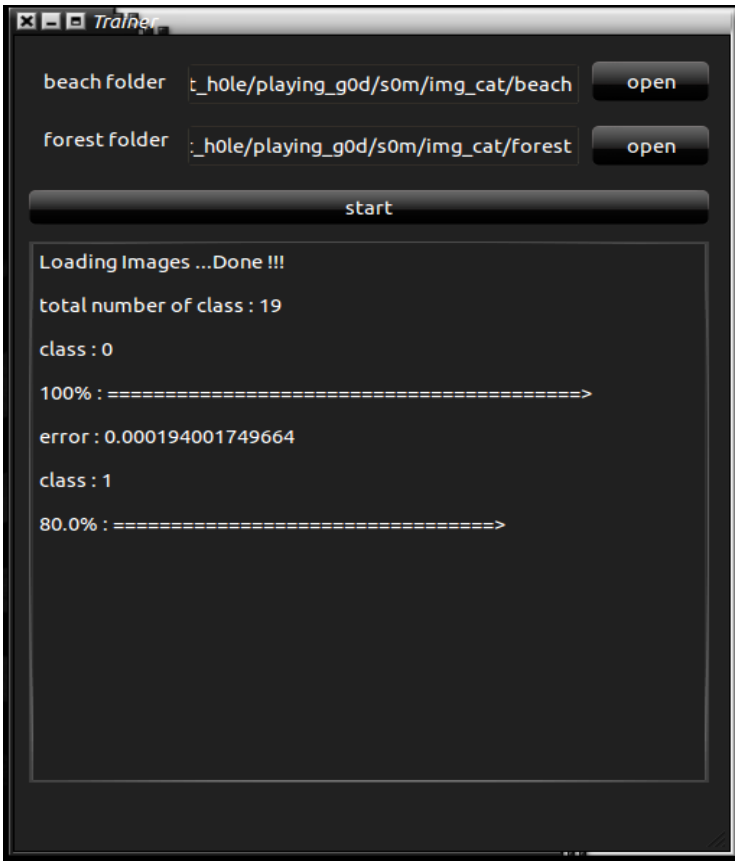

Figure 4.1 Trainer

First choose the two folders which contains training data for beaches and forests. Then use start to start the training. After training the weights of the SOM will be saved in the 'weights' file. ('weights' file already containing trained weights are also included in source code folder).

Then to test the SOM we can use 'tester.py' program. To run it use,

python tester.py

command.

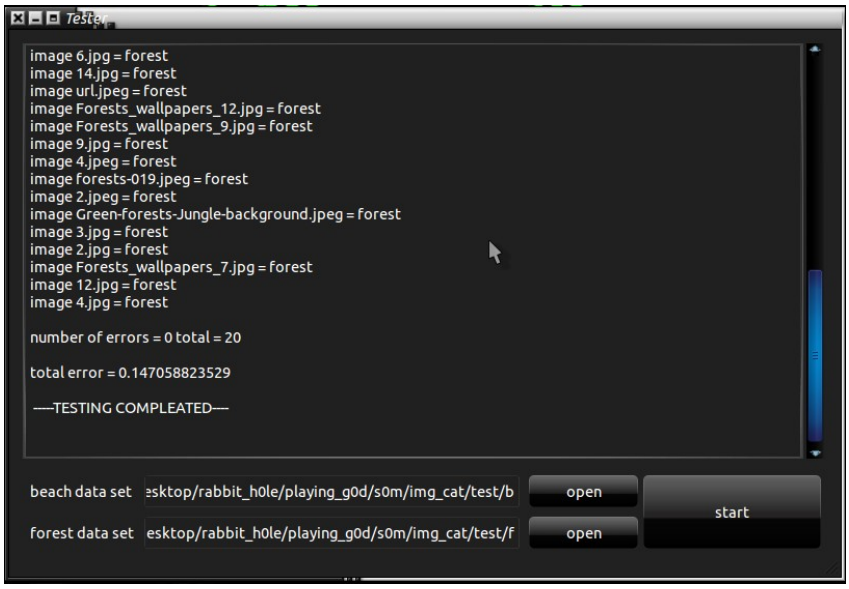

Figure 4.2 Tester

In the program choose test data folders for beaches and forests. Then start the training. It will show the classification of the SOM.

Then to categorize a image use 'categorizer.py'. Run it and open a image using 'open' button. Then use 'find' to classify it.

Figure 4.3 Categorizer

5.0 Results

- Number of classes = 23 (14 - beaches, 9 - forests)
- Total Test images = 50 (25 – beaches, 25 - forests)
- Total misclassified images = 2 (1 – beaches, 1 - forests)
- Total error = 0.04

6.0 More Information

- tcg.galahena@gmail.com
- http://www.inf0warri0r.blogspot.com

7.0 Licenses