README

Albert Buchard, Mariella Kast, Remy Joseph

12/22/2016

Description

This package contains two models that aim at classifying roads from satellite images: a Fully Convolutional Network, and a Convolutional Neural Network which led to the best prediction

CNN

CNN is the model that we used for our best kaggle prediction. To run this code, you need to open run_CNN.py and execute it. There are two main variables to deal with: RESTORE and GENERATE to modify in the code.

Run the predictions

In case the user is interested only in reproducing the submission from the model, we advise to use the following configuration:

```
RESTORE = True
GENERATE = True
model_name = "./model_8_50epoch_0.86.ckpt"
submission_name = 'submission_reproduce.csv'
```

GENERATE: if set to one, generate will produce and save the images for the extended training set.

If this is the first time you run this code, GENERATE has to be set to 1. The rotated images will be produced in a matter of minutes

RESTORE: determines whether the code is to be used to train the network or to make predictions.

If set to False, the network will train itself on the training set and save the model. If set to True, Predictions will be made on each image of the test set as well as on the rotated versions that have to be previously generated Once this is done, a post-processing step is ran and produces the submission file.

Training

If one is interested in running the whole pipeline, we advise to run the routines twice.

```
RESTORE = False
GENERATE = True
model_name = "./model_8_50epoch_0.86_reproduce.ckpt"
submission_name = 'submission_reproduce.csv' # Does not matter actually on the first run
```

Then, to load the newly trained model and predict:

```
RESTORE = True
GENERATE = False
model_name = "./model_8_50epoch_0.86_reproduce.ckpt"
submission_name = 'submission_reproduce.csv'
```

Files

- Training_run.py contains all the routines used in the pre-processing, the training and the testing of our neural network.
- project2_run_win.py contains the post-processing routines.

FCN MODEL

All the code relative to the FCN model construction, training and testing is in run_FCN.py.
Two arguments can be given in the terminal : * -p or –phase : set either to “train”, or “test” * -m or –model : the model filename (should be of format *.h5 using KERAS)

To run training

In the terminal:

```
$ python run_FCN.py -p "train"
```

Will save the new model as “newly_trained_fcn.h5”

To run testing

In the terminal:

```
$ python run_FCN.py -p "test"
```

The default model will be “final_fcn_model.h5”

To set a custom model :

```
$ python run_FCN.py -p "test" -m "custom.h5"
```

Dependencies

- Core: os, sys, getopt, gzip, urllib, warning
- Graphic/images: matplotlib, PIL
- Machine Learning: scipy, tensorflow, keras
- Custom: mask_to_submission

MCA

You will also find a mw_transform.py file that contains the code to perfom MCA as explained in the report. To run the MCA on an image, simply use the function:

```
Ridge, Star = mMCA(img,20,5)
```

Ridge will contain the ridgelet version of the input image img and Star will contain its Starlet counterpart.