CODENSCIOUS

Codenscious Technologies Pvt. Ltd.
www.codenscious.com
+91 731 498-4984

Date: November 28, 2018

MODEL TRAINING GUIDE

1. Deep learning Libraries setup

Step 1: Install Python
You can download the latest version of Python from here :
https://www.python.org/downloads/

Step 2: Download Anaconda
https://www.anaconda.com/download/

Step 3: Install OpenCV

```
pip install opencv-python
```

All the libraries that we are using will be covered under this and included under conda. Even if at the time of executing code, it demands installation of any package, it could be easily installed using command-
" pip install 'package-name' ".

2.Environment setup

You can create an environment which is like an isolated virtual container where you can install python 3.6 and install your libraries.

conda list envs — Lists all the environments created in Anaconda

conda activate env_name — To activate and start working in a particular environment
I have activated "tensorFlow_p36" for our project.

Command for that would be -
source activate tensorflow_p36 , or
conda activate tensorflow

3. Folder Architecture-

Initially our directory will look like:

```
Classification-Model-
        -train_test_split.py
        -Data_prepration.py
        -train.py
        -test.py
        -positive_paths.txt (Contains paths of folders having positive images)
        -negative_paths.txt (Contains path of folders having negative images)
```

Later after compiling python files in sequence,some files would automatically get created. At the end our folder would look like-

```
Classification-Model
        -train_test_split.py
        -positive_training.txt
        -positive_test.txt
        -negative_training.txt
        -negative_test.txt
        -Data_prepration.py
        -train.py
        -test.py
        -Accuracy.jpg (Graph that shows accuracies of model while training)
```

-X_concrete.pickle
-y_concrete.pickle
-Concrete_Crack_Classification_model.model (our model after creation)
-logs.json (it's a json file containing model history)
-positive_paths.txt
-negative_paths.txt
-Residuals.txt (Contains paths for images which are detected False positive, False negatives and those who couldn’t be complied due to some error.)

4.Running Model

Step 1:In Pipeline, there should be two text files. Each of them containing corresponding paths for folders containing Positive and Negative Images.

Step 2: In terminal, traverse into the Pipeline folder.

Step 3: Activated the Environment as explained in point-2
ex. source activate tensorflow_p36

Now we just had to run this four files in given order

-train_test_split.py --pos= *positive txt file* --neg= *negative txt file*
-Data_prepration.py
-train.py
-test.py

Step 5: Splitting data into train and test
terminal command :

```
python train_test_split.py --pos=*positive txt file* --neg=*negative txt file*
```

example :

```
python train_test_split.py --pos=pospath.txt --neg=negpath.txt
```

It will create 4 text files for positive and negative images for both training and testing respectively.

-positive_training.txt (Contains paths of positive images for Training)
-positive_test.txt (Contains paths of positive images for Testing)
-negative_training.txt (Contains paths of negative images for Training)
-negative_test.txt (Contains paths of negative images for Testing)

Note - Currently splitting into 80% train data and 20% test data ,it could be changed accordingly.

Step 6: Data Preprocessing-

```
terminal command : python Data_prepration.py
```

It will create two pickle files from training data.
-X_concrete.pickle
-y_concrete.pickle

Step 7: Initiate Training

```
terminal command : python train.py
```

It will start training our model, and generate an '.jpg' image file displaying accuracy graph.
-Accuracy.jpg

Note- Currently Epochs are set to 10 and validation split to 20 percent, it could be changed accordingly.

Step 8: Testing on sample

```
terminal command : python test.py
```

It will show the inferences, Accuracy of our model at the end. It will also create a text file "Residuals.txt" which would contain paths for images which are predicted false positive, false negative and images which gave error while compiling.

- Residuals.txt

All Done!!