pyleaf Documentation

Release 1.0.0

Vishal Sonawane

Dec 05, 2018

CONTENTS

sphinx-quickstart on Mon Dec 3 19:04:20 2018. You can adapt this file completely to your liking, but it should at least contain the root *toctree* directive.

CHAPTER

ONE

LEAFAREACALCULATOR

1.1 basefunctions module

basefunctions.**CNNModel**()

Convolutional Neural network model. Input is 256 vector input dimensional array as Embedding layer. Additional layers: Conv1d, MaxPooling, Dense, Flatten,Dense.

Type of class label is categorical crossentropy. Adam Optimizer is selected. Display metric is accuracy.

Returns CNN model which is a prototype of the classification model specified in this method.

Return type keras.model

basefunctions.**PROCESS_IMAGE**(*sample_name_list*, *path_*)

Accept the image file name from LeafAreaCalculator and prcoess the passed image by cobverting it to model input vector and processing all pixels in the image. CNN model ouptput class color labels for every pixel. Simple math is then used to calculate leaf area(green pixels) from green:red ratio of all classified labels.

Parameters

- **sample_name_list** (*string*) – image file name to process
- **path** (*string*) – default path to search for image

Returns list of useful parameters: image_file_name, area in cm^2 and ** optional use parameters.

Return type List

basefunctions.**RETRAIN_MODEL_FROM_SCRATCH**()

Retrain the CNN model process entirely from scratch. This is action taken when use uses Tools->Retrain model from GUI menu.

basefunctions.**alter_calc**(*image*)

EXtract pixel level information from the passed image. Retrieved (r,g,b) pixel values for every pixel of the image and convert them to hsv values, for easier seperation of green colored pixels.

Parameters **image** (*string*) – name of image file

Returns List of pixels scanned across the image left top right, top to bottom.

Return type List of tuples

basefunctions.**generateClassLabels**(*sample*, *color*)

Generates class labels for all the pixels extracted from an image. Example: if red.png was fetched, all pixels of this image are labelled 'red'. Likewise, for all images in the default training folder (test_images/) In our default environment, we have green for all kinds of leaf images, red for all red squared, and any other is irrelevant to pur computations.

basefunctions.**getPixelFromLabel**(*label*)
Function to get the pizel from the class label. This reverse mapping is use to generate an image file fom available class labels.

Parameters **label** (*int*) – class label

Returns (r,g,b) equivalent of class label color.

Return type tuple

basefunctions.**image_area_calculator**(*basepath*, *sample*)
Process the desired image, and extract the pixel level information of every pixel in the image in (h,s,v) format. Utlizes alter_calc method to generate (h,s,v) values.

Parameters

- **basepath** (*string*) – default path of images
- **sample** (*string*) – name of image file

Returns returns the vector embedding which will be input to CNN model.

Return type list of lists

basefunctions.**kerasTokenizerUnit**(*topbestwords*)
Convert the (h,s,v) value of every pixel to a 256 dimensional feature vector that serves as an input to the CNN EMbedding layer.

Parameters **topbestwords** – maximum dimesion of vector, default: 256

:type integer :return: 256 dimensional vector :rtype: np.array

basefunctions.**load_model**()
Loads the already available model previously saved on disk.

basefunctions.**processInputTrain**()
Function that extracts all training images from default traininf images foled (test_images/) and converts them into required input vector format for CNN embedding layer.

basefunctions.**readFromDisk**()
Reads the previously sored pixel/class label values saved as pickled objects.

basefunctions.**regular**(*string_list*, *class_labels_norm*, *finalSequence_*)
Trains CNN model with the passed embedding vectors of the image. CNN model is trained with 600*600 image dimesions, so 360000 data pointes per training image, for all images in default training images directory, for 10 iterations. Note: At the time of documenting, the acuarcy of model achieved was 99.5%.

Parameters

- **string_list** – Dummy variable. Not used.
- **class_labels_norm** (*Vector*) – Normalized class labels for the colors.
- **finalSequence** (*List of vectors.*) – Input embedding vectors for image

basefunctions.**saveModelToDisk**(*model*)
Saves the currenly used CNN model configuration to disk.

Parameters **model** (*keras.CNN model*) – current CNN model

basefunctions.**saveToDisk**(*string_list*, *class_labels_norm*, *finalSequence_*)
Saves CNN model configuration and input/output pixel/class_label values to disk for later use. Utilizes pickle objects for compressed serialized storage.

1.2 leaf_area_calculator_gui module

CHAPTER

TWO

INSTALLATION

Download the source package from GitHub repository:

```
https://github.com/vishal11582285/pyleaf.git
```

Once downloaded, you should be able to locate the below files and directories:

```
all_photos : All raw images that can be used for processing
limited_images: Set of limited chosen images for testing purpose
saved_images: This will be the default save directory. <Do not change name of this␣
↪folder>
test_images: Contains all the test images for training the CNN model for␣
↪classification task.

model.h5, model.json: Already trained model. Ready for immediate use.
```

```
Locate the setup.py file
In command prompt, type:
python setup.py build
python setup.py install
```

You should now be able to see the package pyleaf in site_packages of your Python interpreter.

CHAPTER

THREE

USAGE

Open up the Python interpreter:

```
>>import pyleaf
should open up the pyleaf GUI for use.
```

CHAPTER

FOUR

GUI FEATURES

On the GUI Application, you will find various options to get you started right away:

```
Batch Process:
Select default image folder from File->Select Default Image Path
Then select Tools->Batch Process to run leaf area analysis on all images in chosen␣
↪folder.

Single Image Analysis:
Use Select Images button to select one or more image(s) and click Process Images to␣
↪view the results.

Use the navigation page under original window preview to load all selected images.

Reset Workspace button clears the workspace for a fresh start.

Tools->Retrain Model to re-train CNN clasification model based on updated training␣
↪images in test_images/ folder.

Tools->View Recent Results to open up a window and view the image and their␣
↪calculated areas in tabular format.
```

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

b