
Munge Documentation

Release 0.0.1

Srivathsa Pasumarthi

Feb 15, 2018

Contents:

1	Dataset	1
2	DataElement	3
3	DataLoader	5
4	ImageThresholder	7
5	Utils	9
6	Indices and tables	11
	Python Module Index	13

Class to represent a dataset as a whole or for each study

```
class munge.Dataset.Dataset (config_file='config.json')  
    Bases: object
```

Dataset class can be instantiated with the following args

- **parameters, types, return and return types:**

Parameters **config_file** (*string*) – full path of the application config file

asarray ()

Returns the array representation of all the data points in this dataset

Returns array of data and labels of this dataset

get_all ()

Maps the images with the contours and returns a generator with data points

Returns generator of instances of `DataElement` having the corresponding image and contour

get_by_study (*patient_id*)

Maps the images with contours and returns a generator with data points, for the given study

Parameters **patient_id** – unique ID of the study

Returns generator of instances of `DataElement` having the corresponding image and contour, for the given study

plot_verification_for_study (*patient_id, filename=None, rows=5, columns=5*)

Plots a series of images with the corresponding contour patches for the given study

Parameters

- **patient_id** – unique ID of the study
- **filename** – filename to save the plot in
- **rows** – number of rows in the plot

- **columns** – number of columns in the plot

to_dict (*patient_id=None*)

Returns the Dict representation of the dataset

Returns Dict having id, dcm_path and contour_path attributes of the data points in this dataset

Class to represent a data point in the dataset with relevant features and methods

class munge.DataElement.DataElement (dicom_path, icontour_path, ocontour_path=None)
Bases: object

DataElement class can be instantiated with the following args

- **parameters, types, return and return types:**

Parameters

- **dicom_path** (*string*) – full path of the DICOM image
- **contour_path** (*string*) – full path of the corresponding contour file

asarray()

Returns the DataElement in the form of (data, label)

Returns array of data and labels

get_area_in_sqmm (roi='icontour')

Gets the area of the ROI in sq.mm. The conversion is done using the PixelSpacing tag of the DICOM image.

Returns area in sq.mm

get_image_icontour_overlay (window=30, patch_color=[255, 0, 0])

Gets a bounding box around the inner contour with and without the i-contour overlaid (horizontally stacked). This will be useful for manual verification of the annotation

Parameters

- **window** – size of bounding box required around the marked contour
- **patch_color** – [r, g, b] value of the color in which the patch should be overlaid

Returns horizontally stacked array with left image being original and the right with the patch drawn

get_image_ocontour_overlay (*window=30, patch_color=[255, 0, 0]*)

Gets a bounding box around the outer contour with and without the o-contour overlaid (horizontally stacked). This will be useful for manual verification of the annotation

Parameters

- **window** – size of bounding box required around the marked contour
- **patch_color** – [r, g, b] value of the color in which the patch should be overlaid

Returns horizontally stacked array with left image being original and the right with the patch drawn

get_roi_avg_relative_intensity (*roi='icontour'*)

Gets the relative intensity (%) of the ROI. Relative intensity is w.r.t the maximum intensity of the image

Returns average intensity in percentage

overlay_contours (*window=30, patch_colors=[[0, 0, 255], [255, 0, 0]]*)

Overlays both inner and outer contours for visualization

Parameters

- **window** – Bounding box window size around the ROI
- **patch_colors** – Array of colors for the outer and inner contours

Class to load data in the second stage of the pipeline

```
class munge.DataLoader.DataLoader(dataset)  
    Bases: object
```

DataLoader class can be instantiated with the following args

- **parameters, types, return and return types:**

Parameters **dataset** – instance of Dataset class

```
load_train_data(epochs=10, batch_size=8, log_file='data_loader.log')  
    Returns an array of DataElement instances split into batches and epochs
```

Parameters

- **epochs** – number of epochs needed
- **batch_size** – number of images to be used per batch
- **log_file** – path to the log_file

Returns array of dimension epochs x (data_size/batch_size) x batch_size containing instances of DataElement

```
static plot_random_epoch(data, epoch_size=10, filename=None)  
    Method to plot the images from a randomly selected epoch
```

Parameters

- **data** – return value of load_train_data function
- **epoch_size** – size of epoch
- **filename** – file to which the plot should be saved

ImageThresholder

Class to threshold an image and plot necessary figures related to thresholding

```
class munge.ImageThresholder.ImageThresholder (data_element, n_components=2,  
                                              method='gmm', postprocess=False)
```

Bases: `object`

ImageThresholder class can be instantiated with the following args

- **parameters, types, return and return types:**

Parameters

- **data_element** – Instance of `DataElement` class
- **n_components** – Number of components to the model fit

dilate (*thresholded_img*)

Performs binary dilation on the given image using a disk-shaped structural element of arbitrary radius 3.

Parameters **thresholded_img** – thresholded image

Returns dilated image

get_jaccard_coeff ()

Gets the jaccard similarity co-efficient of the reference and the detected region

Returns jaccard coefficient

get_thresholded_contour_mask ()

Thresholds the o-contour region and returns a mask

Returns Boolean mask containing the thresholded image

plot_model_fit (*filename=None*)

Plots the histogram of the o-contour region with the gaussians that were used to fit the model and the selected threshold value

Parameters **filename** – File path to save the plot

plot_thresholding_result (*filename=None*)

Plots the thresholded region along with the ground truth region overlaid in different colors

Parameters **filename** – File path to save the plot

Contour related util functions

`munge.utils.contour.get_dcm_num_for_contour(contour_file_name)`

Gets the DICOM series number for a given contour file name or full file path

Ex: For 'data/contourfiles/SC-HF-I-1/i-contours/IM-0001-0048-icontour-manual.txt' the return value will be 48

Parameters `contour_file_name` – name of the contour file

Returns Integer corresponding to the DICOM series number

`munge.utils.contour.parse_contour_file(filename)`

Parse the given contour filename

Parameters `filename` – filepath to the contourfile to parse

Returns list of tuples holding x, y coordinates of the contour

`munge.utils.contour.poly_to_mask(polygon, width, height)`

Convert polygon to mask

Parameters

- **polygon** – list of pairs of x, y coords [(x1, y1), (x2, y2), ...] in units of pixels
- **width** – scalar image width
- **height** – scalar image height

Returns Boolean mask of shape (height, width)

Image related util functions

`munge.utils.image.get_dcm_resolution(dcm_img)`

Gets the resolution of the DICOM image

Parameters `dcm_img` – pydicom instance of DICOM image

Returns Resolution of the DICOM image i.e equivalent spacing of 1 pixel in millimeters

`munge.utils.image.grayscale_to_rgb(img_raw)`

Converts the given grayscale image to a three channel image

Parameters `img_raw` – the raw grayscale image

Returns 3 Channel RGB image

`munge.utils.image.parse_dicom_file(filename)`

Parse the given DICOM filename

Parameters `filename` – filepath to the DICOM file to parse

Returns dictionary with DICOM image data

Miscellaneous util functions

`munge.utils.misc.csv2dict(csv_file)`

Converts and returns a CSV file to Dict

Parameters `csv_file` – Path to the CSV file

Returns Dict representation of the CSV file

`munge.utils.misc.get_app_config(config_file)`

Gets the application configuration as a dict from the given file

Parameters `config_file` – path to the configuration file

Returns Dict containing the application configuration

`munge.utils.misc.get_bounding_box_coords(contour, window=30)`

Given a contour and window, get the min and max co-ordinates of a bounding box around that window

Parameters

- **contour** – Array of co-ordinates defining the contour
- **window** – The window size of the bounding box

Returns min_x, max_x, min_y and max_y of the bounding box

`munge.utils.misc.get_ocontour_for_icontour(icontour_file, ocontour_dir)`

Gets the ocontour file corresponding to the given icontour_path. If the ocontour file does not exist, *None* is returned.

Parameters `icontour_path` – Full path to the icontour file

Returns Path of the corresponding ocontour file, if exists or *None*

`munge.utils.misc.get_uuid()`

Generates and returns a random GUID

Returns Random GUID V4

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

m

- `munge.DataElement`, [3](#)
- `munge.DataLoader`, [5](#)
- `munge.Dataset`, [1](#)
- `munge.ImageThresholder`, [7](#)
- `munge.utils.contour`, [9](#)
- `munge.utils.image`, [9](#)
- `munge.utils.misc`, [10](#)

A

asarray() (munge.DataElement.DataElement method), 3
asarray() (munge.Dataset.Dataset method), 1

C

csv2dict() (in module munge.utils.misc), 10

D

DataElement (class in munge.DataElement), 3
DataLoader (class in munge.DataLoader), 5
Dataset (class in munge.Dataset), 1
dilate() (munge.ImageThreshold.ImageThreshold method), 7

G

get_all() (munge.Dataset.Dataset method), 1
get_app_config() (in module munge.utils.misc), 10
get_area_in_sqmm() (munge.DataElement.DataElement method), 3
get_bounding_box_coords() (in module munge.utils.misc), 10
get_by_study() (munge.Dataset.Dataset method), 1
get_dcm_num_for_contour() (in module munge.utils.contour), 9
get_dcm_resolution() (in module munge.utils.image), 9
get_image_icontour_overlay() (munge.DataElement.DataElement method), 3
get_image_ointour_overlay() (munge.DataElement.DataElement method), 3
get_jaccard_coeff() (munge.ImageThreshold.ImageThreshold method), 7
get_ointour_for_icontour() (in module munge.utils.misc), 10
get_roi_avg_relative_intensity() (munge.DataElement.DataElement method), 4
get_thresholded_contour_mask() (munge.ImageThreshold.ImageThreshold method), 7
get_uuid() (in module munge.utils.misc), 10
grayscale_to_rgb() (in module munge.utils.image), 9

I

ImageThreshold (class in munge.ImageThreshold), 7

L

load_train_data() (munge.DataLoader.DataLoader method), 5

M

munge.DataElement (module), 3
munge.DataLoader (module), 5
munge.Dataset (module), 1
munge.ImageThreshold (module), 7
munge.utils.contour (module), 9
munge.utils.image (module), 9
munge.utils.misc (module), 10

O

overlay_contours() (munge.DataElement.DataElement method), 4

P

parse_contour_file() (in module munge.utils.contour), 9
parse_dicom_file() (in module munge.utils.image), 10
plot_model_fit() (munge.ImageThreshold.ImageThreshold method), 7
plot_random_epoch() (munge.DataLoader.DataLoader static method), 5
plot_thresholding_result() (munge.ImageThreshold.ImageThreshold method), 7
plot_verification_for_study() (munge.Dataset.Dataset method), 1
poly_to_mask() (in module munge.utils.contour), 9

T

to_dict() (munge.Dataset.Dataset method), 2