LCAT Documentation

Release 0.1.0

Lung Cancer Action Team

Jun 12, 2017

Contents

Contents:

CHAPTER 1

lcat package

Subpackages

lcat.analysis package

Submodules

lcat.analysis.tracheal_distance module

BMI 260: Final Project Bronchi segmentation/identification

lcat.analysis.tracheal_distance.**get_seed_boundary**(*lung_segmentation*)
Given a *lung_segmentation*, identify and return the top slice of the trachea.

lcat.analysis.tracheal_distance.**get_tracheal_distances**(*scan*, *lung_segmentation*)
Calculate a distance map for every accessible voxel in the lung segmentation of the distance from the top of the trachea to the voxel. Returns a masked numpy array representing the distance assigned to each voxel.

Module contents

Analysis package of the Lung Cancer Action Team toolkit.

lcat.featurization package

Submodules

lcat.featurization.body_depth module

Body depth featurization module.

lcat.featurization.body_depth.**featurize_center**(*scan*)
Featurize the given scan, returning body depth statistics.

lcat.featurization.center module

X, Y and Z featurization module.

lcat.featurization.center.**featurize_center**(*scan*)
Featurize the given scan, returning tracheal distance statistics.

lcat.featurization.characteristics module

Characteristics featurization module.

lcat.featurization.characteristics.**featurize_characteristics**(*scan*)
Featurize the given scan, returning nodule characteristics.

lcat.featurization.region_properties module

X, Y and Z featurization module.

lcat.featurization.region_properties.**calculate_equivalent_diameter**(*nodule_mask*, *unit_cell*)
Calculate and return the equivalent diameter of a sphere with the same volume as the given nodule.

lcat.featurization.region_properties.**calculate_max_intensity**(*nodule_mask*, *intensity_image*)
Calculate and return the maximum intensity of the nodule represented by *nodule_mask* in *intensity_image*

lcat.featurization.region_properties.**calculate_mean_intensity**(*nodule_mask*, *intensity_image*)
Calculate and return the average intensity of the nodule represented by *nodule_mask* in *intensity_image*

lcat.featurization.region_properties.**calculate_min_intensity**(*nodule_mask*, *intensity_image*)
Calculate and return the minimum intensity of the nodule represented by *nodule_mask* in *intensity_image*

lcat.featurization.region_properties.**calculate_volume**(*nodule_mask*, *unit_cell*)
Calculate and return the volume occupied by the nodule represented by the given nodule_mask.

lcat.featurization.region_properties.**featurize_region_properties**(*scan*)
Featurize the given scan, returning tracheal distance statistics.

lcat.featurization.registry module

Feature registry module of Lung Cancer Action Team package.

lcat.featurization.registry.**featurize_scan**(*scan*)
Featurize the given scan, using all available featurizers. Returns a pandas DataFrame with all featurizer results, indexed by patient_id and nodule_id using a MultiIndex.

lcat.featurization.registry.**featurize_scan_single**(*scan*, *featurizer_name*)
Featurize the given scan, using the specified featurizer. Returns a pandas DataFrame with all featurizer results, indexed by patient_id and nodule_id using a MultiIndex.

lcat.featurization.registry.**register_featurizer**(*featurizer_name*)
Function decorator which registers the given function under the argument *featurizer_name* as a featurizer. The function must accept a single scan and return a pandas DataFrame containing columns representing features and rows representing nodules. The Index must specify the nodule_id for each nodule.

lcat.featurization.tracheal_distance module

Tracheal distance featurization module.

lcat.featurization.tracheal_distance.**featurize_tracheal_distance**(*scan*)
Featurize the given scan, returning tracheal distance statistics.

Module contents

Featurization package of the Lung Cancer Action Team toolkit.

lcat.loading package

Submodules

lcat.loading.annotations module

BMI 260: Final Project Load chest CT scan annotations from radiologist xml files.

class lcat.loading.annotations.**Nodule**(*nodule_id*, *characteristics*, *origin*, *mask*)
Bases: tuple

characteristics
Alias for field number 1

mask
Alias for field number 3

nodule_id
Alias for field number 0

origin
Alias for field number 2

lcat.loading.annotations.**get_mask_region**(*read*, *dimensions*, *sop_instance_uids*)
Returns a full representation of the region represented by the given nodule read as a mask.

lcat.loading.annotations.**get_nodule_information**(*read*, *dimensions*, *sop_instance_uids*)
Given an unblindedReadNodule element, create a Nodule object representing the nodule's characteristics and vertices.

lcat.loading.annotations.**get_read_characteristics**(*read*)
Get the characteristics from a read as recorded by the radiologist. Returns an empty dictionary if no characteristics were recorded.

lcat.loading.annotations.**get_read_mask**(*read*, *dimensions*, *sop_instance_uids*)
Get a 3D array representing the region described by the specific read, prefaced by an origin specifying its placement in the image (in index coordinates).

lcat.loading.annotations.**get_read_nodule_id**(*read*)

lcat.loading.annotations.**load_radiologist_annotations**(*dicom_folder*, *dimensions*, *sop_instance_uids*)
Load radiologist annotations (namely nodule characteristics and regions) from the xml files present in *dicom_folder*. Returns an array of Nodule objects representing all nodules found in the radiologist annotations.

lcat.loading.annotations.**main**()
Command-line invocation routine.

lcat.loading.annotations.**mark_region**(*mask*, *roi_elem*, *sop_instance_uids*)
Mark the region of interest encoded by *roi_elem* in *mask*. *sop_instance_uids* is used to determine the slices referenced by *roi_elem*.

lcat.loading.images module

Authors: Connor Brinton and Scotty Fleming Load a CT scan from a series of dicom files.

lcat.loading.images.**get_single_value**(*values*)
Given a sequence of values, checks that all values are the same, and then returns the single value. The values must satisfy transitive equality.

lcat.loading.images.**load_folder**(*dicom_folder*)
Given a folder of dicom files, load them and return a 3D numpy array representing the scan. Pixel values are converted to Houndsfield units using the rescaling slope and intercept encoded in each dicom file.

lcat.loading.scans module

Load data for a chest CT scan from both dicom and radiologist xml files.

class lcat.loading.scans.**Scan**(*patient_id*, *voxels*, *nodules*, *unit_cell*)
Bases: tuple

nodules
Alias for field number 2

patient_id
Alias for field number 0

unit_cell
Alias for field number 3

voxels
Alias for field number 1

lcat.loading.scans.**cubify_scan**(*scan*)
Given a scan, interpolate the data to make the unit cell cubic. The dimension(s) with the smallest magnitude(s) in the unit cell will remain the same.

lcat.loading.scans.**get_scaling_factors**(*scan*)
Returns the scaling factors for the given scan.

lcat.loading.scans.**interpolate_array**(*array*, *new_shape*, *output=None*, *mode='nearest'*)
Given an array and a target shape, perform an orthogonal transformation to the new shape using a spline interpolation.

lcat.loading.scans.**interpolate_array_by_axis**(*array*, *axis_coordinates*, *output=None*, *mode='nearest'*)

lcat.loading.scans.**load_scan**(*scan_folder*, *cubify=False*)
Loads the CT scan as a 3d voxel array, then loads the segmentation in the given dicom_folder by reading any xml

files located in the folder and referencing dicom files as necessary. Returns a 3D mask with the same dimensions as the CT scan.

TODO: Combine nodules referring to the same entity. Unfortunately there is currently no unique ID for each nodule, meaning that multiple radiologist reads result in multiple almost-identical nodules in the scan metadata.

lcat.loading.scans.**rescale_nodule**(*nodule*, *scaling_factors*)
This interpolation process is trickier than the voxel interpolation process, because we're not only rescaling the mask, we're also moving the origin of the mask. In order to properly perform this interpolation, we perform the following steps: (1) Find the scaled origin as real numbers (not rounded) (2) Find the scaled anti-origin (opposite corner) as real numbers (not rounded) (3) Count the number of scaled cells along each dimension for mask (4) Convert rounded extents to original space (5) Perform rescaling by axis coordinates

lcat.loading.scans.**zoom_array**(*array*, *factor*, *mode='nearest'*)
Rescale the given *array* along each axis by *factor*. If *factor* is a sequence, each axis will be zoomed by its corresponding factor in *factor*.

Module contents

Loading package of the Lung Cancer Action Team toolkit.

lcat.segmentation package

Submodules

lcat.segmentation.body module

Segments a body from a CT scan.

lcat.segmentation.body.**get_body_envelope**(*body_mask*)
Given a mask representing thresholded lung values, obtain an envelope containing the lung region with no interior holes.

lcat.segmentation.body.**get_body_segmentation**(*scan*)
Given a *Scan* object representing a chest CT scan, return a binary mask representing the region occupied by the body.

lcat.segmentation.body.**get_largest_volume**(*labels*)
Return a binary mask equivalent to the component with the largest volumes in the array *labels*.

lcat.segmentation.body.**get_top_value**(*arr*)
Given an ndarray, return the value which occurs most frequently.

lcat.segmentation.lungs module

Authors: Connor Brinton and Scotty Fleming Segment lungs from a chest CT scan.

lcat.segmentation.lungs.**get_largest_volume**(*labels*)
Return a binary mask equivalent to the component with the largest volumes in the array *labels*.

lcat.segmentation.lungs.**get_lung_envelope**(*lung_mask*)
Given a mask representing thresholded lung values, obtain an envelope containing the lung region with no interior holes.

lcat.segmentation.lungs.**get_lung_segmentation**(*scan*)
Given a *Scan* object representing a chest CT scan, return a binary mask representing the lungs (not including the air within the lungs).

lcat.segmentation.lungs.**get_top_value**(*arr*)
Given an ndarray, return the value which occurs most frequently.

Module contents

Segmentation package of the Lung Cancer Action Team toolkit.

Submodules

lcat.util module

Utility functions for the lcat toolkit.

lcat.util.**clear_border**(*labels*, *axis=None*, *in_place=False*)
Clears any labeled components touching either border along the given axes (in *axis*).

lcat.util.**compress_nodule_mask**(*mask*)

lcat.util.**crop_to_box**(*arr*, *box*)
Given an array *arr* and boundaries *boundaries* (such as is returned by *get_bounding_box*), return *arr* cropped to *boundaries*.

lcat.util.**get_bounding_box**(*arr*)
Given an array of values, returns an list of tuples, where each tuple represents the extent of the non-zero values in *arr* along a particular axis.

Inspired by http://stackoverflow.com/questions/31400769/bounding-box-of-numpy-array

lcat.util.**get_full_nodule_mask**(*nodule*, *scan_shape*)

lcat.util.**image_from_mask**(*mask*)
Convert a binary mask into a PIL image.

lcat.util.**plot_slices**(*voxels*, *rows=6*, *columns=6*, *cmap=None*)
Given a 3D array of voxels, plots slices.

lcat.util.**save_slices**(*voxels*, *destination_folder*, *prefix='slice'*)
Given a voxel array, save each slice as a tiff file with the name *prefix###.tiff* in the given *destination_folder*.

Module contents

Main package of the Lung Cancer Action Team toolkit.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

l

Index

M

N

O

P

R

S

U

V

Z