

---

# **Crystal Pattern Recognition**

***Release 0.1***

**Michael Barbier**

**Apr 07, 2020**



# CONTENTS

<b>1</b>	<b>Package mlpy</b>	<b>1</b>
<b>2</b>	<b>Module plots</b>	<b>3</b>
<b>3</b>	<b>Module detection</b>	<b>5</b>
<b>4</b>	<b>Module patterns</b>	<b>7</b>
<b>5</b>	<b>Module mio</b>	<b>9</b>
<b>6</b>	<b>Indices and tables</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>
	<b>Index</b>	<b>15</b>



**PACKAGE MLPY**

Crystal pattern recognition in images.



## MODULE PLOTS

Module for plots, graphs, annotation of images

`mlpy.plots.draw_circles(circle_list, im, radius)`

Draw circles on an image with a specific radius.

**Parameters**

- **circle\_list** – List of [x,y,radius] of the circle
- **im** – The input image
- **radius** – The radius of the circles

**Returns** The image with the circles annotated (writes on the original image)

**Return type** image as uint8 rgb numpy matrix

`mlpy.plots.plot_lindemann_histogram(lindemann_parameter_list, n_bins)`

Plots the Lindemann histogram TODO





## MODULE DETECTION

Module for particle/spot detection in an image/video

`mlpy.detection.detection` (*orig, method, saturation\_perc, radius*)

Detection of particles as centers and radii. Uses a specified method and does some pre-processing of the data.

**Parameters**

- **orig** – The original image (can be RGB or gray-valued)
- **method** – On of the valid methods: ['CHT', 'Laplace']
- **saturation\_perc** – Saturation percentage
- **radius** – Expected radius

**Returns** [circle\_list, im\_gray, im\_norm, im\_blur] circle\_list:

`mlpy.detection.detection_cht` (*im, radius*)

Detect particles using the Circular Hough Transform (CHT)

**Parameters**

- **im** – The input image which should be grey-valued
- **radius** – The radius of the particles used by the CHT algorithm

**Returns** [1,2]: (1) A list with [x, y, radius] values, (2) The smoothed image used as input to the CHT algorithm

`mlpy.detection.detection_laplace` (*im, radius*)

**Parameters**

- **im** –
- **radius** –

**Returns**



## MODULE PATTERNS

The machine learning methods to recognize various crystal structures.

`mlpy.patterns.compute_lindemann_parameter(circle_list, radius)`

Draw circles on an image with a specific radius.

### Parameters

- **circle\_list** – list of [x, y, radius], describing the circles
- **radius** – The radius describing the size of the local region around the point/particle of interest

**Returns** None



## MODULE MIO

Module for input/output of images, text files, etc

`mlpy.mio.get_metadata(file_path)`

Extracts the meta data of an image/video using ffmpeg and puts it into a specific format (dictionary)

**Parameters** `file_path` – File path of the image/video

**Returns** The meta data as a dictionary

`mlpy.mio.print_metadata(file_path)`

Prints the meta data as extracted by ffmpeg and returns this raw meta data.

**Parameters** `file_path` – File path of the image/video

**Returns** The meta data as extracted by ffmpeg

`mlpy.mio.read_frames(file_path, frame_list)`

Loads video data using the OpenCV library (reads in a specified list of frames).

**Parameters**

- `file_path` – File path of the video
- `frame_list` – List of frames (the indices) of interest

**Returns** A list with the frames as numpy? data



## INDICES AND TABLES

- genindex
- modindex
- search





## PYTHON MODULE INDEX

### d

`detection`, 5

### m

`mio`, 9

`mlpy`, 1

`mlpy.detection`, 3

`mlpy.mio`, 7

`mlpy.patterns`, 5

`mlpy.plots`, 1

### p

`patterns`, 7

`plots`, 3



## INDEX

### C

`compute_lindemann_parameter()` (in module `mlpy.patterns`), 7

### D

`detection`  
module, 5

`detection()` (in module `mlpy.detection`), 5

`detection_cht()` (in module `mlpy.detection`), 5

`detection_laplace()` (in module `mlpy.detection`),  
5

`draw_circles()` (in module `mlpy.plots`), 3

### G

`get_metadata()` (in module `mlpy.mio`), 9

### M

`mio`  
module, 9

`mlpy`  
module, 1

`mlpy.detection`  
module, 3

`mlpy.mio`  
module, 7

`mlpy.patterns`  
module, 5

`mlpy.plots`  
module, 1

`module`  
  `detection`, 5  
  `mio`, 9  
  `mlpy`, 1  
  `mlpy.detection`, 3  
  `mlpy.mio`, 7  
  `mlpy.patterns`, 5  
  `mlpy.plots`, 1  
  `patterns`, 7  
  `plots`, 3

### P

`patterns`

module, 7

`plot_lindemann_histogram()` (in module  
`mlpy.plots`), 3

`plots`  
module, 3

`print_metadata()` (in module `mlpy.mio`), 9

### R

`read_frames()` (in module `mlpy.mio`), 9