Ryan Zurrin, Neha Goyal, Pablo Bendiksen, Muskaan Nanocha, Kendrick Kheav, Daniel Haehn

Building the World's Largest Mammography Database

More info: mpsych.org/omama

Simple Data API

```
1  import omama as O # importing the Omama package

1  omama_loader = O.OmamaLoader() # Subclass of DataLoader which helps set paths and env settings

1  data = O.Data(omama_loader, load_cache=True) # loading close to a million images

1  print(data) # easily print the data that was loaded using print
2  len(data) # easily print the total number of images loaded

Data(namespace(dh2=namespace(total=22144, 3D=9363, 2D=12781),
    dh0new=namespace(total=126460, 3D=62736, 2D=63724),
    dh_dcm_ast=namespace(total=819387, 3D=58, 2D=819329),
        total_all_dicoms=967991,
        total_2d_all=895834,
        total_3d_all=7157,
        total_2d_cancer=14965,
        total_2d_preindex=1915,
        total_2d_noncancer=855763,
        total_3d_cancer=376,
        total_3d_preindex=0,
        total_cancer=15341,
        total_preindex=1915,
        total_noncancer=922925,
        total_no_label=27810))

967991

1  # easily get 2D and 3D image
2  imgs2d = O.DataHelper.get2D(N=5, timing=True)
get2D        ...took    3.297618 seconds

1  O.DataHelper.view(imgs2d[0]) # view single images
```

```
1  imgs3d = O.DataHelper.get3D(N=5, timing=True)
get3D        ...took    73.524466 seconds

1  imgs = imgs2d + imgs3d # combine lists of images

1  O.DataHelper.view_grid(imgs, slice_num=22) # print lists of images easily too!

1  # Filter the data up to contain specific images
2
3  data.filter_data(3d=False, labels=["IndexCancer"], studies=["dh_dh2"], timing=True)
filter_images        ...took    0.163414 seconds

1  len(data) # Notice now there are only 37 images in the Data class
37

1  img_generator = data.next_image(_2d=True, timing=True) # build a generator with the images

1  # create a list of only the pixels from our images to use with ML algorithms
2  pixel_list = []
3  for i in range(36):
4      pixel_list.append(next(img_generator).pixels)

1  pixel_list[2]
array([[  0,   0,   0, ...,   0,   0,   0],
       [  0,   0,   0, ...,   0,   0,   0],
       [  0,   0,   0, ...,   0,   0,   0],
       ...,
       [2677, 2721, 2706, ...,   0,   0,   0],
       [2802, 2756, 2718, ...,   0,   0,   0],
       [2834, 2795, 2756, ...,   0,   0,   0]], dtype=uint16)

1  pixel_list[35]
array([[  0,   0,   0, ..., 441, 512, 527],
       [  0,   0,   0, ..., 481, 533, 588],
       [  0,   0,   0, ..., 496, 505, 580],
       ...,
       [  0,   0,   0, ...,   0,   0,   0],
       [  0,   0,   0, ...,   0,   0,   0],
       [  0,   0,   0, ...,   0,   0,   0]], dtype=uint16)

1  data.reset_data() # To reset the Data just use the reset_data() method

1  len(data) # Notice the length is back to almost 1 million images.
967991
```

Database Stats

- **175k+** 2D Mammograms
- **65k+** 3D Tomosynthesis
- **7k+** Cancer cases

Filtering for High Quality Data

Omama Mammography Dataset
Eliminating cases shown in red across the pipeline.

Pipeline steps:
- Remove 'No Label' and 'Both' cases.
- Remove uncommon image shape cases
- Remove minority manufacturer cases
- Remove study duplicates for 2D image cases (2D non-cancer only; no set overlap with 3D non-cancer)

Missing Label Filtering

895,834	2D Images
14,965	w/Index-Cancer
1,915	w/Pre-Cancer
72,157	3D Images
376	w/Index-Cancer
204,192	Total Studies
985,247	Total Images Start

−27,933 Images

Small Shape Size Filtering

872,521	2D Images
14,964	w/Index-Cancer
1,915	w/Pre-Cancer
67,538	3D Images
376	w/Index-Cancer
198,752	Total Studies
957,314	Total After 1st Step

−24,890 Images

Minority Manufacturer Filtering

852,290	2D Images
10,390	w/Index-Cancer
1,834	w/Pre-Cancer
67,534	3D Images
376	w/Index-Cancer
198,610	Total Studies
932,424	Total After 2nd Step

−3,825 Images

Preserve Unique Studies (2D Non-Cancer Only)

848,501	2D Images
10,368	w/Index-Cancer
1,820	w/Pre-Cancer
67,534	3D Images
376	w/Index-Cancer
198,610	Total Studies
928,599	Total After 3rd Step

191,547	2D Images
10,368	w/Index-Cancer
1,820	w/Pre-Cancer
67,534	3D Images
376	w/Index-Cancer
197,785	Total Studies
259,081	Total Images

−669,518 Images

Bar chart labels:
- Missing Label Filtering: No Label Info, Index-Cancer, Pre-Cancer
- Small Shape Size Filtering: (512, 512), (787, 983), (794, 890)
- Minority Manufacturer Filtering: Lorad, MG, Kodak, MG, Siemens, MG, Fujifilm, Philips (Points scored)
- Preserve Unique Studies: Images From Unique Studies, Images From Repeat Studies (Points scored)

Digital Mammogram / 3D Tomosynthesis Mammogram

Digital Mammogram

3D Tomosynthesis Mammogram

Intelligent Annotation Framework

1: Machine
0: Human

Discriminator 1

Data B
Labeled by expert

re-label and add as training data

Annotations

Classifier

Data A
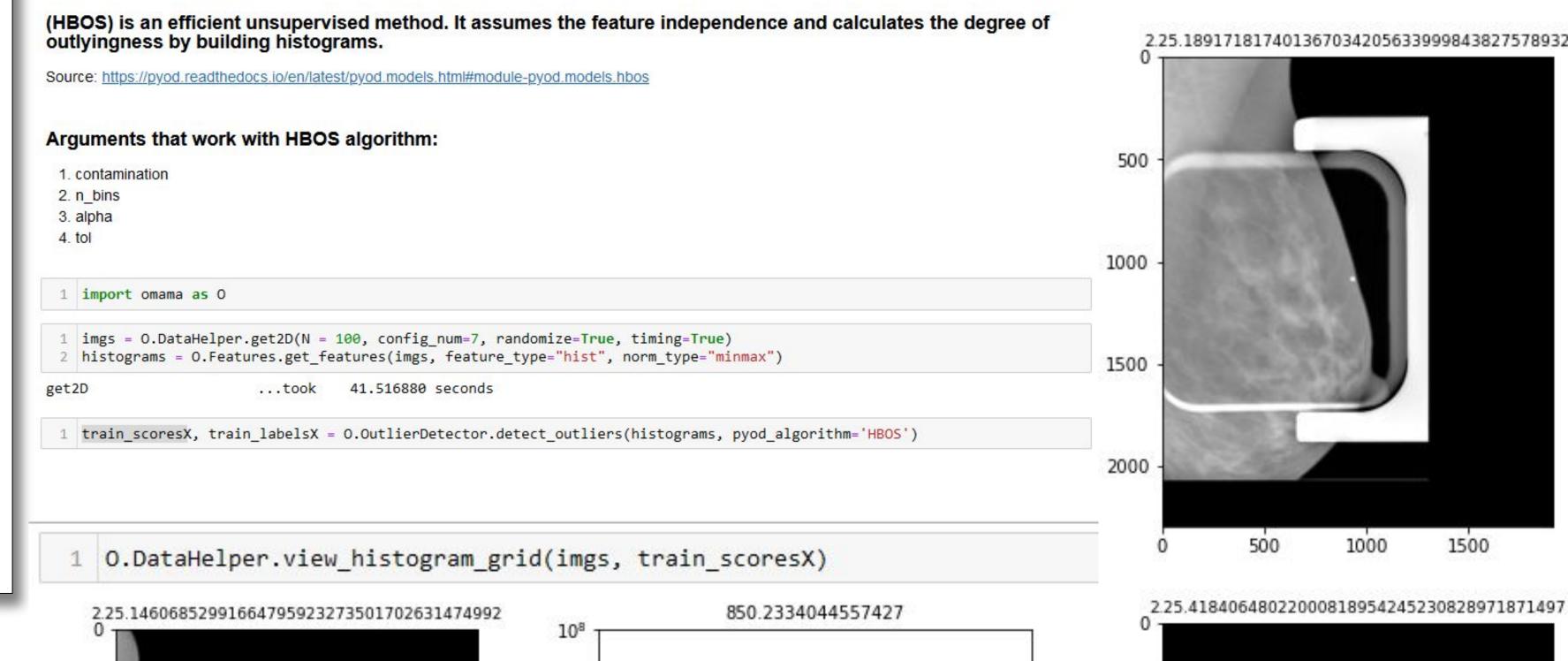
Web-based Annotation Tool

https://www.crowds-cure.org/

RMLO

```
Score ID
Quality: Good
0 - 5  Add Score
Z: Toggle Bad/Good
R: Reset image view
X: Reset tools/data
Left Click: ROI
Middle Click: Pan
Ctrl + Left Click: Window/Level
Space/Enter: Save and Load Next Image
Image_ID 2.25.124033402746662693819048297533826540049
```

Area: 14.311.91 px²
Mean: 2,887.43
Std Dev: 153.67

Data Science / GPU Compute

Annotations

Data Science

GPU Compute

2x NVIDIA DGX A100

SYSTEM SPECIFICATIONS

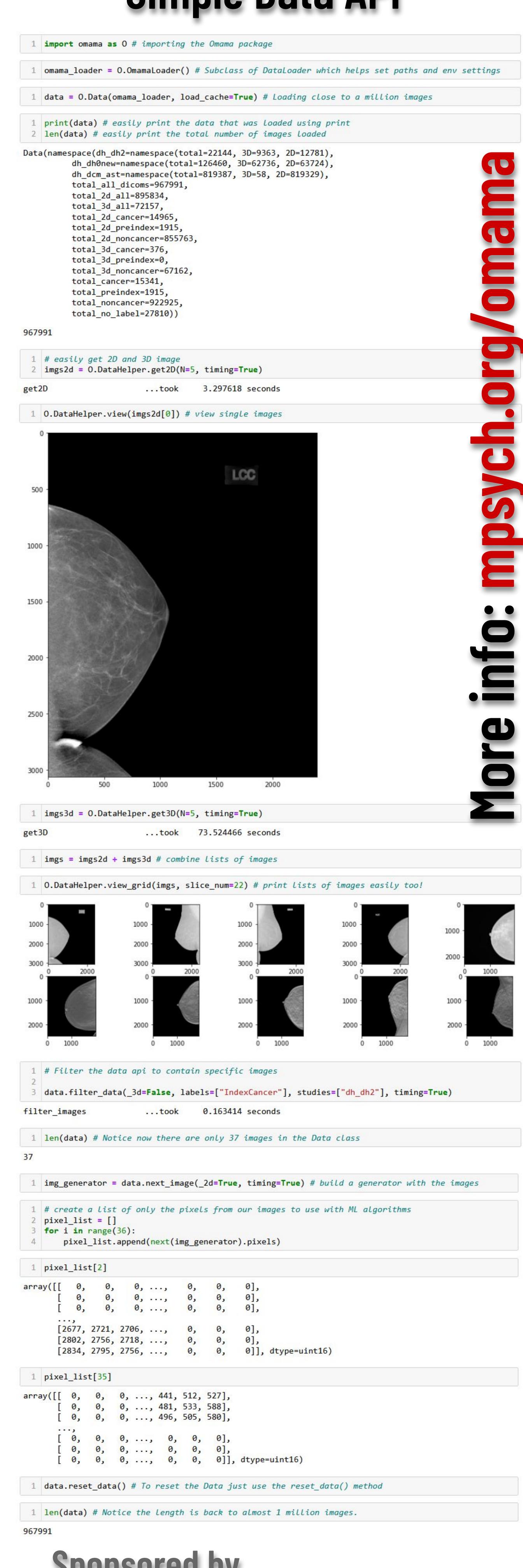|	
GPUs	8x NVIDIA A100 Tensor Core GPUs
GPU Memory	320 GB total
Performance	5 petaFLOPS AI 10 petaOPS INT8
NVIDIA NVSwitches	6
CPU	Dual AMD Rome 7742, 128 cores total, 2.25 GHz (base), 3.4 GHz (max boost)
System Memory	1TB
Networking	8x Single-Port Mellanox ConnectX-6 VPI 200Gb/s HDR InfiniBand 1x Dual-Port Mellanox ConnectX-6 VPI 10/25/50/100/200Gb/s Ethernet
Storage	OS: 2x 1.92TB M.2 NVME drives Internal Storage: 30TB (4x 7.5TB) U.2 NVME drives

Unsupervised Outlier Detection

HBOS - Histogram- based outlier detection

(HBOS) is an efficient unsupervised method. It assumes the feature independence and calculates the degree of outlyingness by building histograms.

Source: https://pyod.readthedocs.io/en/latest/pyod.models.html#module-pyod.models.hbos

Arguments that work with HBOS algorithm:
1. contamination
2. n_bins
3. alpha
4. tol

```
1  import omama as O

1  imgs = O.DataHelper.get2D(N = 100, config_num=7, randomize=True, timing=True)
2  histograms = O.Features.get_features(imgs, feature_type="hist", norm_type="minmax")
get2D        ...took    41.516080 seconds

1  train_scoresX, train_labelsX = O.OutlierDetector.detect_outliers(histograms, pyod_algorithm='HBOS')
```

```
1  O.DataHelper.view_histogram_grid(imgs, train_scoresX)
```

UMass Boston

DeepHealth — RadNet AI Solutions

Made with ♥ in MA

Everything Open-Source!