

Intro to Machine Learning

Marc Kohli, MD

Director of Clinical Informatics

UCSF Radiology and Biomedical Imaging

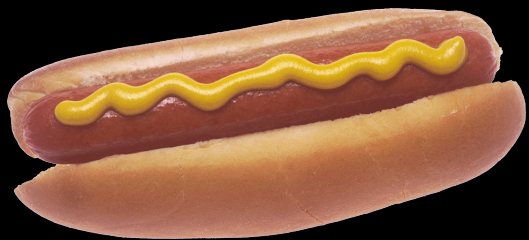
@shadowdoc

What is ML?

```
1 # super_breast_cancer_detector.py
2
3 class breast_cancer
4     # This class defines a set of functions and routines
5     # that look for features of breast cancer
6
7     def __init__(self, image):
8         mc_score = microcalcification_score(image)
9         ad_score = architectural_distortion(image)
10        self.cancer_prob = mc_score + ad_score
11
12    def microcalcification_score(image):
13        # evaluate all the pixels of the image looking
14        # clusters of high intensity
15        return score
16
17    def architectural_distortion(image):
18        # extract glandular tissue
19        # trace edges of glandular tissue
20        # compare against known good patterns
21        return score
22
```

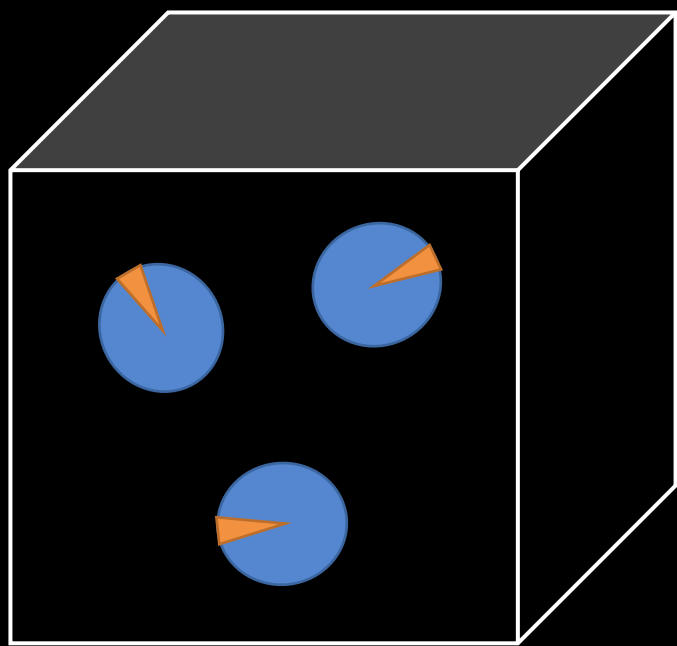
VS

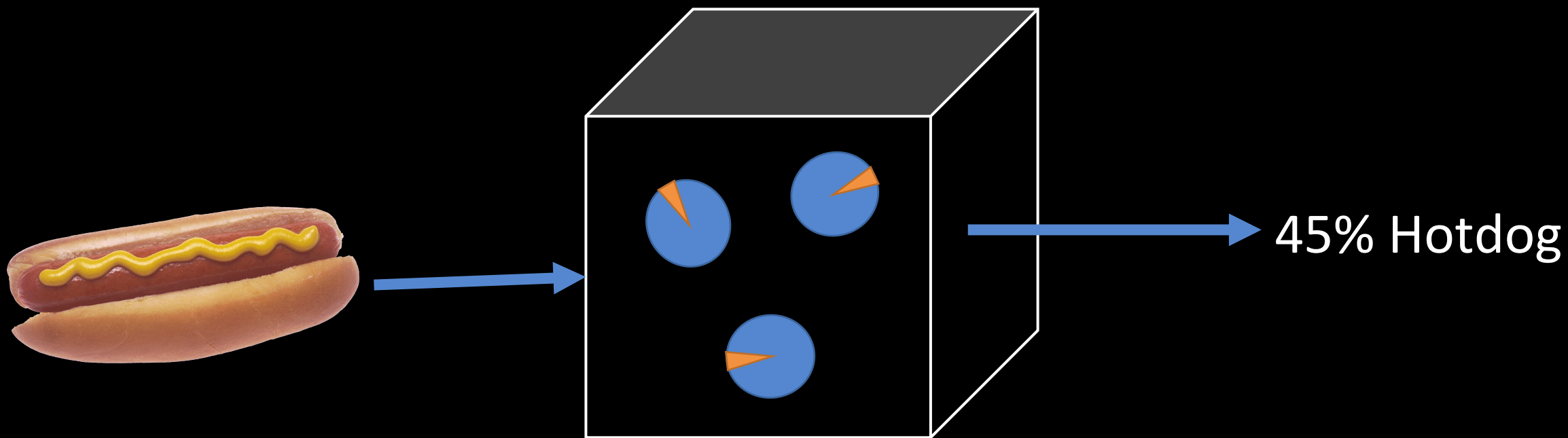


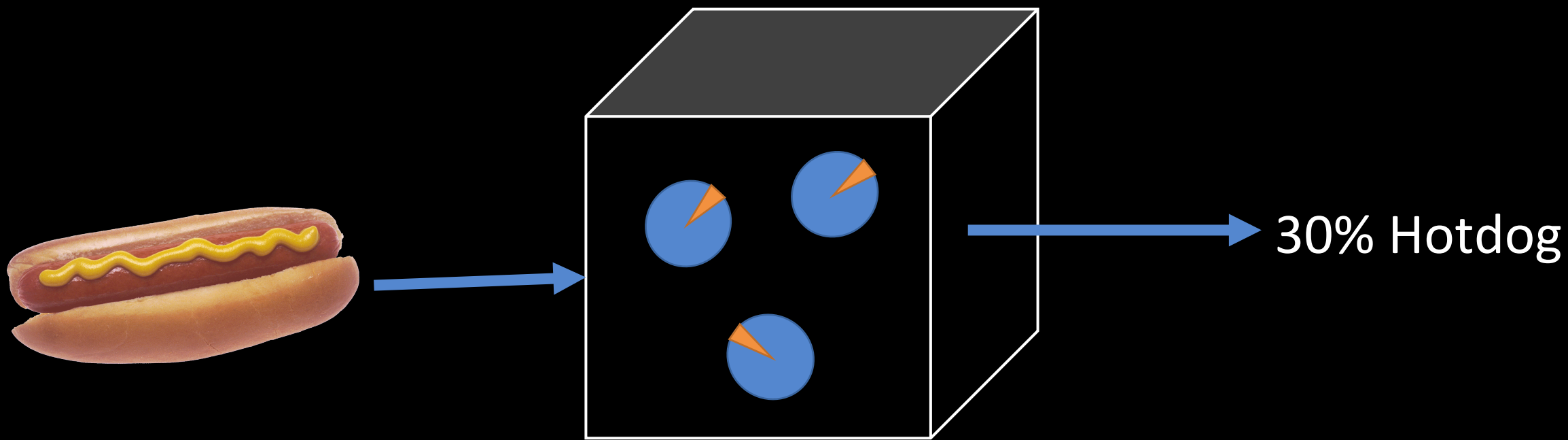


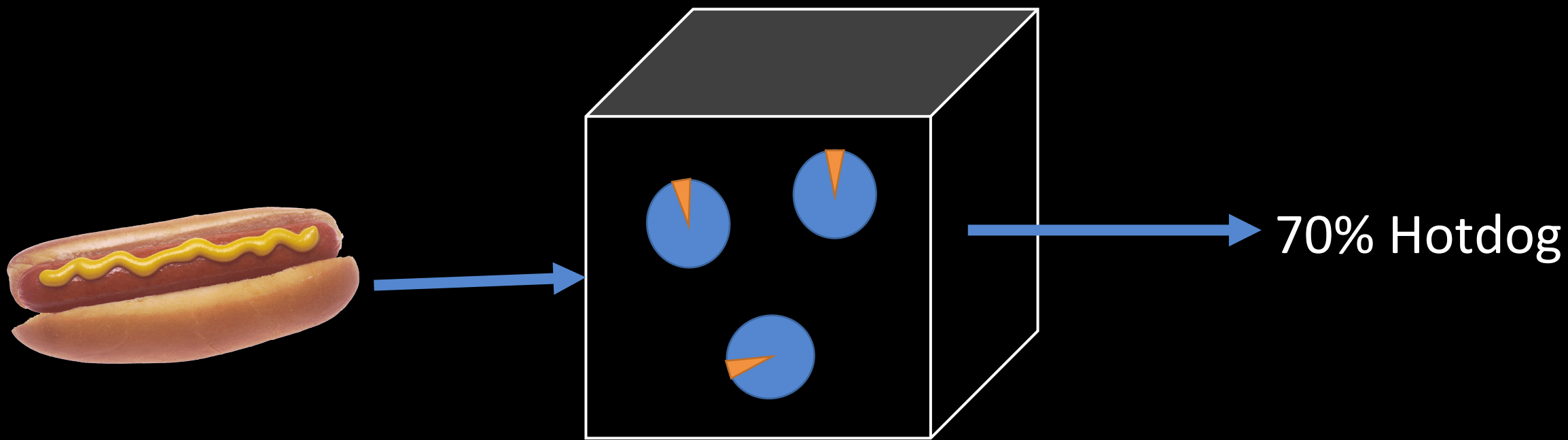
OR

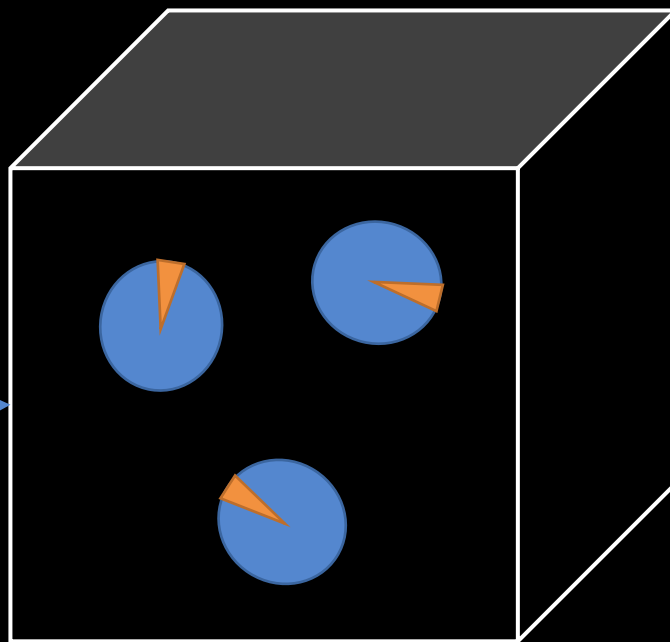
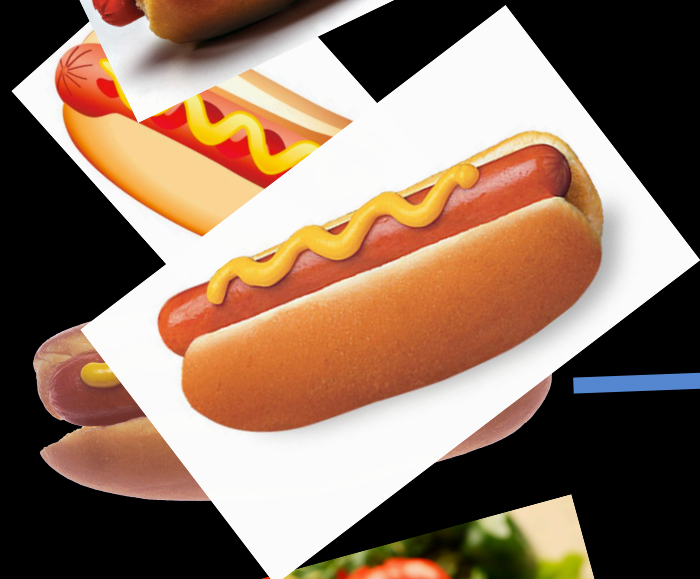


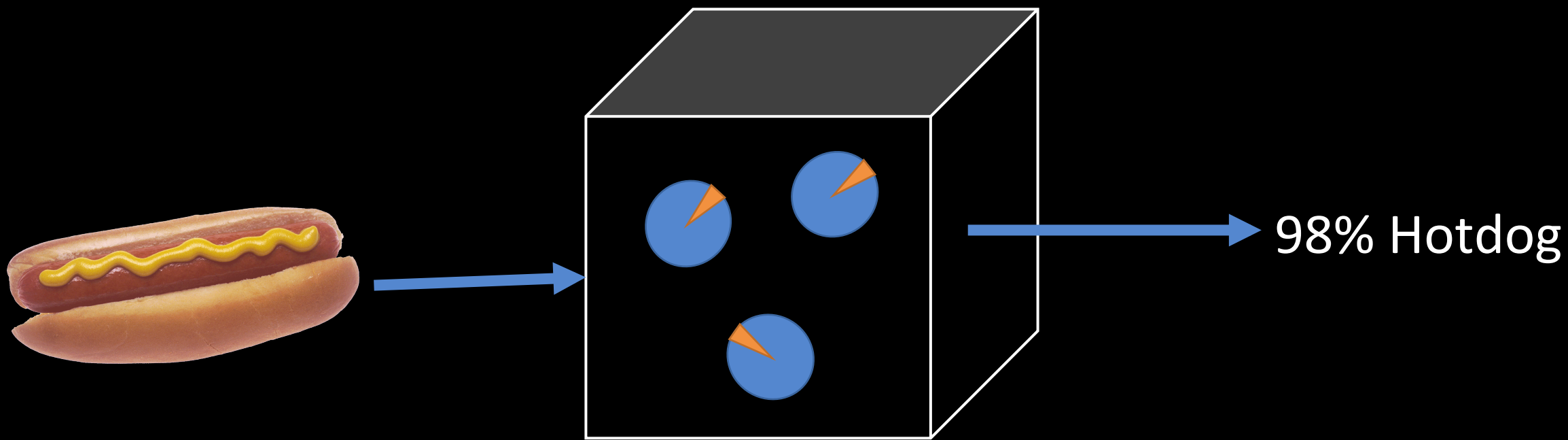


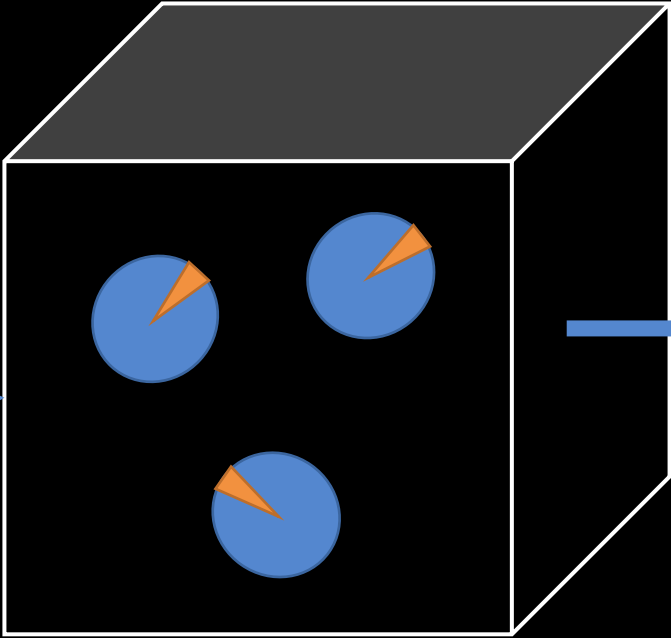




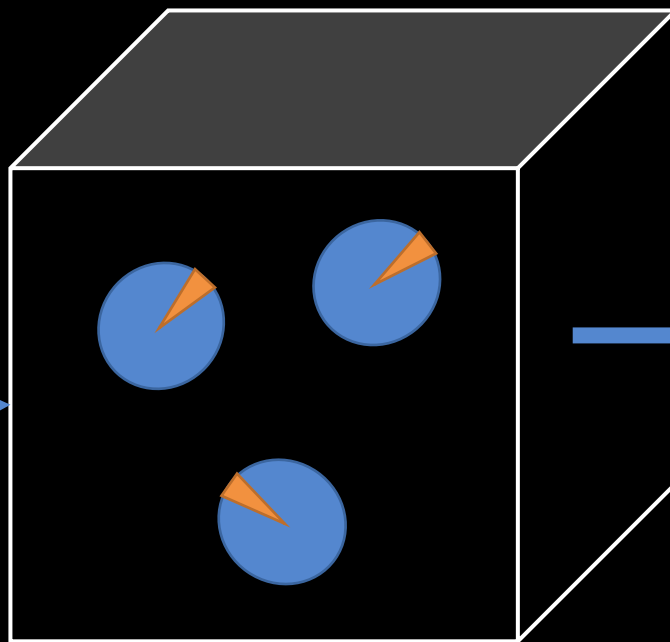






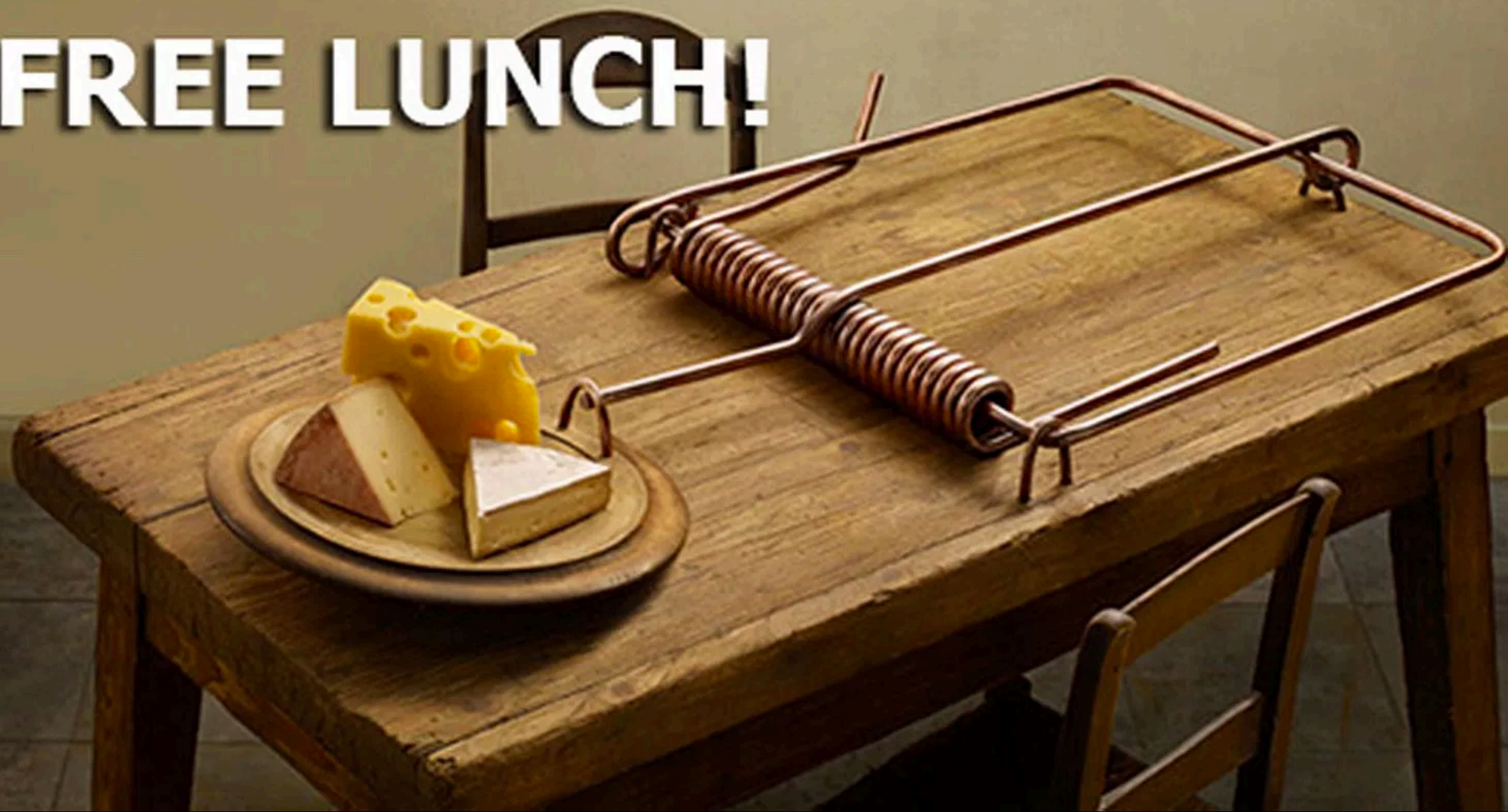


93% Hotdog



0.2% hotdog

FREE LUNCH!



CAD vs. ML

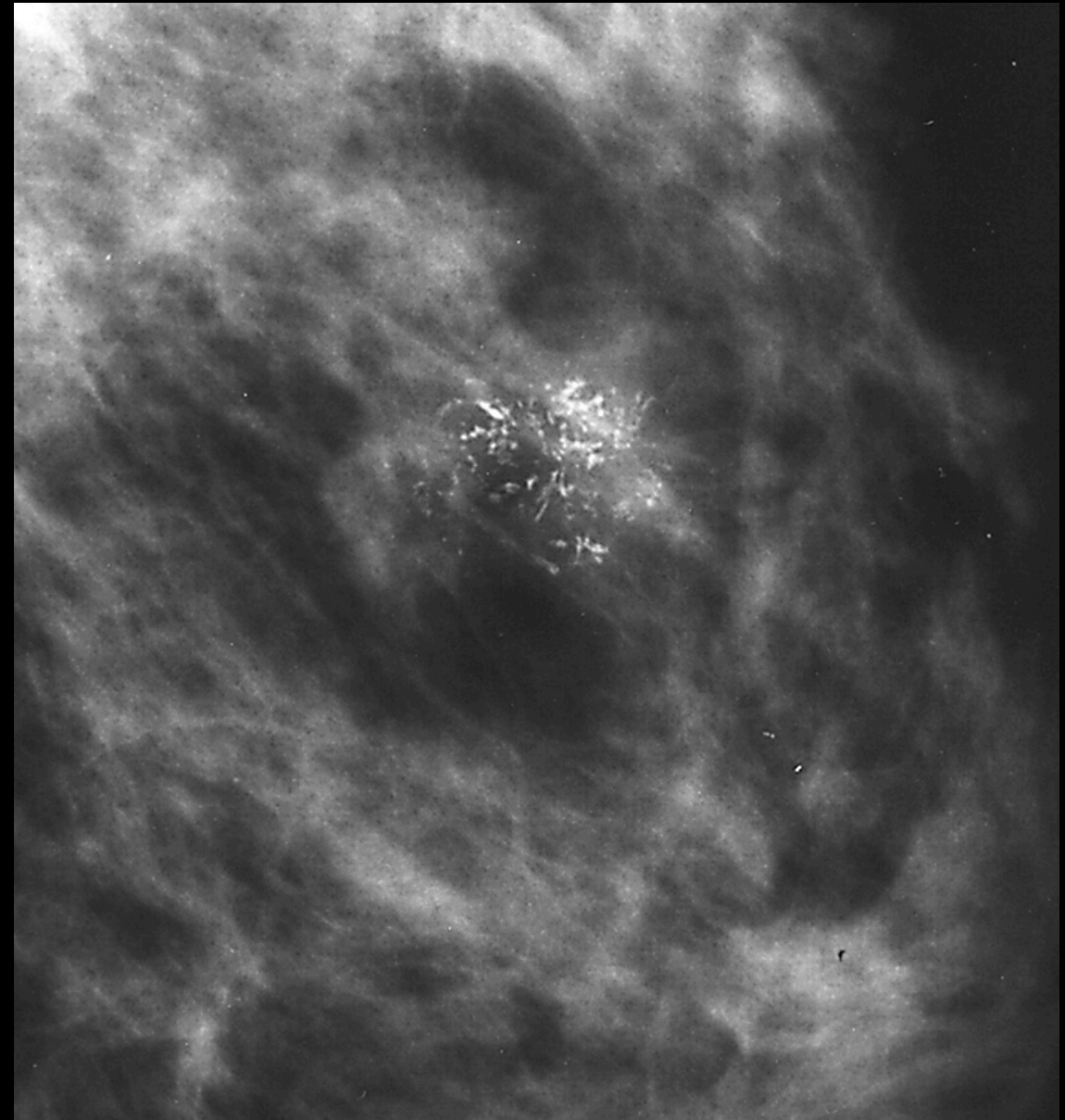


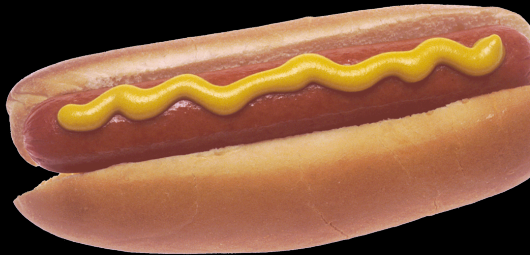


```
1  # not_hotdog.py
2
3  class hotdog
4      # This class defines a set of functions and routines
5      # that look for features of breast cancer
6
7      def __init__(self, image):
8          mustard_score = mustard(image)
9          tubular_meat_score = tubular_meat(image)
10         bun_score = bun(image)
11         if (mustard_score + tubular_meat_score + bun_score > 100):
12             self.hotdog = true
13         else:
14             self.hotdog = false
15
16         def mustard(image):
17             #take all the pixels that are yellow
18             #look for squiggly line pattern
19             return mustard_score
20
21         def tubular_meat(image):
22             #look at all of the redish pixels
23             #identify a tubular structure
24             return tubular_meat_score
25
26         def bun(image):
27             #look for light brown pixels that are arranged
28             #in a linear bun shape
29             return bun_score
```



```
1  # super_breast_cancer_detector.py
2
3  class breast_cancer
4      # This class defines a set of functions and routines
5      # that look for features of breast cancer
6
7      def __init__(self, image):
8          mc_score = microcalcification_score(image)
9          ad_score = architectural_distortion(image)
10         mass_score =
11         if mc_score + ad_score + mass_score > 225:
12             return true
13         else:
14             return false
15
16     def microcalcification_score(image):
17         # evaluate all the pixels of the image looking
18         # clusters of high intensity
19         return score
20
21     def architectural_distortion(image):
22         # extract glandular tissue
23         # trace edges of glandular tissue
24         # compare against known good patterns
25         return score
26
27     def mass(image):
28         # identify whether there are any masslike
29         # regions in the image
```





```
1 # not_hotdog.py
2
3 class hotdog
4     # This class defines a set of functions and routines
5     # that look for features of breast cancer
6
7     def __init__(self, image):
8         mustard_score = mustard(image)
9         tubular_meat_score = tubular_meat(image)
10        bun_score = bun(image)
11        if (mustard_score + tubular_meat_score + bun_score > 100):
12            self.hotdog = true
13        else:
14            self.hotdog = false
15
16    def mustard(image):
17        #take all the pixels that are yellow
18        #look for squiggly line pattern
19        return mustard_score
20
21    def tubular_meat(image):
22        #look at all of the redish pixels
23        #identify a tubular structure
24        return tubular_meat_score
25
26    def bun(image):
27        #look for light brown pixels that are arranged
28        #in a linear bun shape
29        return bun_score
```



```
1 # not_hotdog.py
2
3 class hotdog
4     # This class defines a set of functions and routines
5     # that look for features of breast cancer
6
7     def __init__(self, image):
8         mustard_score = mustard(image)
9         tubular_meat_score = tubular_meat(image)
10        bun_score = bun(image)
11        if (mustard_score + tubular_meat_score + bun_score > 100):
12            self.hotdog = true
13        else:
14            self.hotdog = false
15
16    def mustard(image):
17        #take all the pixels that are yellow
18        #look for squiggly line pattern
19        return mustard_score
20
21    def tubular_meat(image):
22        #look at all of the redish pixels
23        #identify a tubular structure
24        return tubular_meat_score
25
26    def bun(image):
27        #look for light brown pixels that are arranged
28        #in a linear bun shape
29        return bun_score
```



```
1 # not_hotdog.py
2
3 class hotdog
4     # This class defines a set of functions and routines
5     # that look for features of breast cancer
6
7     def __init__(self, image):
8         mustard_score = mustard(image)
9         tubular_meat_score = tubular_meat(image)
10        bun_score = bun(image)
11        if (mustard_score + tubular_meat_score + bun_score > 100):
12            self.hotdog = true
13        else:
14            self.hotdog = false
15
16    def mustard(image):
17        #take all the pixels that are yellow
18        #look for squiggly line pattern
19        return mustard_score
20
21    def tubular_meat(image):
22        #look at all of the redish pixels
23        #identify a tubular structure
24        return tubular_meat_score
25
26    def bun(image):
27        #look for light brown pixels that are arranged
28        #in a linear bun shape
29        return bun_score
```



```
1 # not_hotdog.py
2
3 class hotdog
4     # This class defines a set of functions and routines
5     # that look for features of breast cancer
6
7     def __init__(self, image):
8         mustard_score = mustard(image)
9         tubular_meat_score = tubular_meat(image)
10        bun_score = bun(image)
11        if (mustard_score + tubular_meat_score + bun_score > 100):
12            self.hotdog = true
13        else:
14            self.hotdog = false
15
16    def mustard(image):
17        #take all the pixels that are yellow
18        #look for squiggly line pattern
19        return mustard_score
20
21    def tubular_meat(image):
22        #look at all of the redish pixels
23        #identify a tubular structure
24        return tubular_meat_score
25
26    def bun(image):
27        #look for light brown pixels that are arranged
28        #in a linear bun shape
29        return bun_score
```



```
1 # not_hotdog.py
2
3 class hotdog
4     # This class defines a set of functions and routines
5     # that look for features of breast cancer
6
7     def __init__(self, image):
8         mustard_score = mustard(image)
9         tubular_meat_score = tubular_meat(image)
10        bun_score = bun(image)
11        if (mustard_score + tubular_meat_score + bun_score > 100):
12            self.hotdog = true
13        else:
14            self.hotdog = false
15
16    def mustard(image):
17        #take all the pixels that are yellow
18        #look for squiggly line pattern
19        return mustard_score
20
21    def tubular_meat(image):
22        #look at all of the redish pixels
23        #identify a tubular structure
24        return tubular_meat_score
25
26    def bun(image):
27        #look for light brown pixels that are arranged
28        #in a linear bun shape
29        return bun_score
```



```
1 # not_hotdog.py
2
3 class hotdog
4     # This class defines a set of functions and routines
5     # that look for features of breast cancer
6
7     def __init__(self, image):
8         mustard_score = mustard(image)
9         tubular_meat_score = tubular_meat(image)
10        bun_score = bun(image)
11        if (mustard_score + tubular_meat_score + bun_score > 100):
12            self.hotdog = true
13        else:
14            self.hotdog = false
15
16    def mustard(image):
17        #take all the pixels that are yellow
18        #look for squiggly line pattern
19        return mustard_score
20
21    def tubular_meat(image):
22        #look at all of the redish pixels
23        #identify a tubular structure
24        return tubular_meat_score
25
26    def bun(image):
27        #look for light brown pixels that are arranged
28        #in a linear bun shape
29        return bun_score
```

CAD = human defined

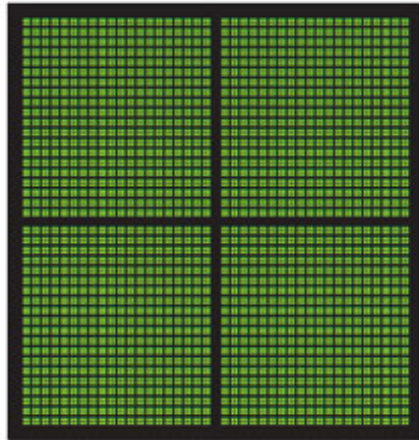
ML = machine detected



Why did imaging ML get so hot right now?



CPU
MULTIPLE CORES



GPU
THOUSANDS OF CORES

IMAGENET Large Scale Visual Recognition Challenge (ILSVRC)

Competition

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and image classification at large scale. One high level motivation is to allow researchers to compare progress in detection across a wider variety of objects -- taking advantage of the quite expensive labeling effort. Another motivation is to measure the progress of computer vision for large scale image indexing for retrieval and annotation.

For details about each challenge please refer to the corresponding page.

- [ILSVRC 2015](#)
- [ILSVRC 2014](#)
- [ILSVRC 2013](#)
- [ILSVRC 2012](#)
- [ILSVRC 2011](#)
- [ILSVRC 2010](#)

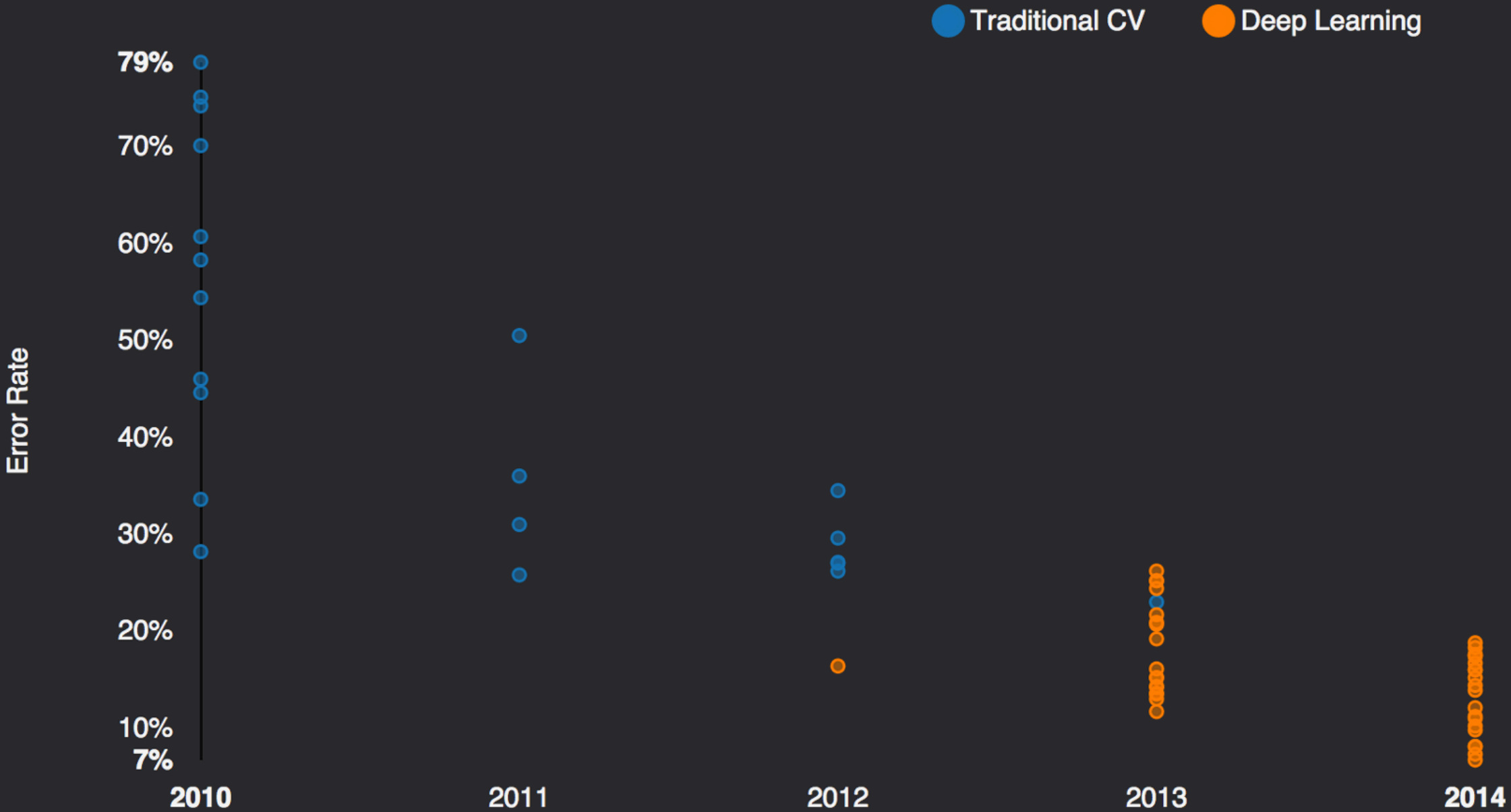
Workshop

Every year of the challenge there is a corresponding workshop at one of the premier computer vision conferences. The purpose of the workshop is to present the methods and results of the challenge. Challenge participants with the most successful and innovative entries are invited to present. Please visit the current challenge page for the workshop schedule and information.

Citation

When reporting results of the challenges or using the datasets, please cite:

Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) **ImageNet Large Scale Visual Recognition Challenge**. *IJCV*, 2015. [paper](#) | [bibtex](#) | [paper content on arxiv](#) | [attribute annotations](#)



Your query "mountain" matches 79 synsets.

Rank by [Popularity](#) [Depth](#)



Synset: [range](#), [mountain range](#), [range of mountains](#), [chain](#), [mountain chain](#), [chain of mountains](#)

Definition: a series of hills or mountains; "the valley was between two ranges of hills"; "the plains lay just beyond the mountain range".

Popularity percentile: 95%

Depth in WordNet: 3



Synset: [mountain](#), [mount](#)

Definition: a land mass that projects well above its surroundings; higher than a hill.

Popularity percentile: 95%

Depth in WordNet: 4



Synset: [mountain tent](#)

Definition: a lightweight tent with a floor; flaps close with a zipper.

Popularity percentile: 72%

Depth in WordNet: 7



Synset: [mountain bike](#), [all-terrain bike](#), [off-roader](#)

Definition: a bicycle with a sturdy frame and fat tires; originally designed for riding in mountainous country.

Popularity percentile: 71%

Depth in WordNet: 8



Synset: [western white pine](#), [silver pine](#), [mountain pine](#), [Pinus monticola](#)

Definition: tall pine of western North America with stout blue-green needles; bark is grey-brown with rectangular plates when mature.

Popularity percentile: 69%

Depth in WordNet: 13

Your query "mountain" matches 79 synsets.

Rank by [Popularity](#) [Depth](#)



[Synset: range, mountain range, range of mountains, chain, mountain](#)

Definition: a series of hills or mountains; "the valley between the mountain range".

Popularity percentile:: 95%

Depth in WordNet: 3

[Synset: mountain, mount](#)

Definition: a land mass

Popularity percentile:: 95%

Depth in WordNet: 4

[Synset: mountain](#)

Definition: a light

Popularity percentile:: 95%

Depth in WordNet: 7

[Synset: mountain bike,](#)

Definition: a bicycle with a seat and handlebars, designed for riding on mountainous country.

Popularity percentile:: 71%

Depth in WordNet: 8

[Synset: western white pine, silver pine, mountain pine, Pinus monticola](#)

Definition: tall pine of western North America with stout blue-green needles; bark is grey-brown with rectangular plates when mature.

Popularity percentile:: 69%

Depth in WordNet: 13

Powered by
Enterprise Imaging

Challenges



FINGER POINTING

When artificial intelligence botches your medical diagnosis, who's to blame?



Who owns the medical record?

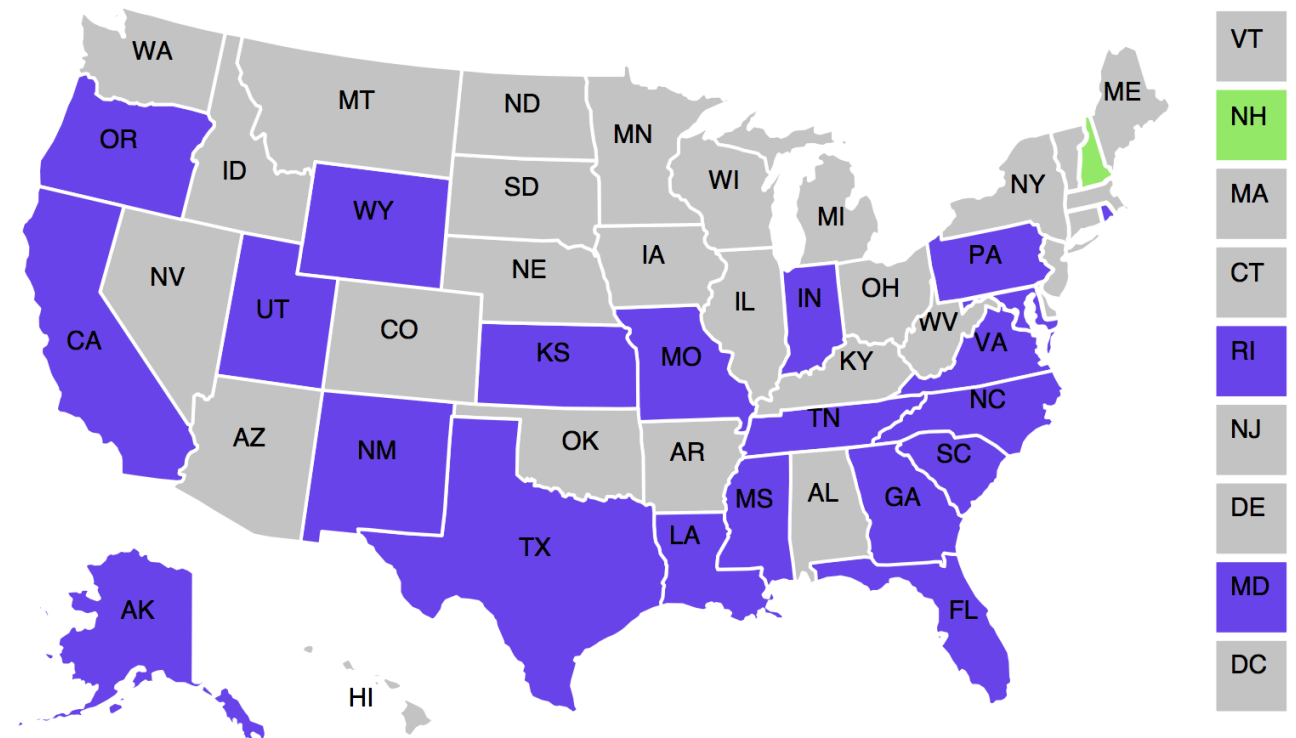
Notes:

* Provider ownership of medical record is referenced in language of law

± Ownership is of the physical conveyance for the medical information

‡ Ownership is of the information contained in the record

[Last Updated 08/20/15]



Medical Record Ownership Laws



Hospital and/or physician owns medical record



Patient owns information in medical record



No law identified conferring specific ownership or property right to medical record

Medicine

NHS gave DeepMind patient records on an 'inappropriate legal basis'

A top UK government advisor is questioning a deal that granted DeepMind access to 1.6 million patient records.



Nick Summers, 05.16.17

A data-sharing deal between DeepMind and London's Royal Free Hospital Trust was struck on an "inappropriate legal basis," a top UK government advisor has said. In April 2016, NewScientist revealed that the company had received 1.6 million patient records to develop an app called "Streams." While there are strict rules regarding patient data and confidentiality, common law states that consent is "implied" if the information is being used for "direct care." Google's AI division used this line of

Health data privacy group medConfidential takes a different view, however: "This response by Google shows that DeepMind has learnt nothing. There may well be lawful reasons for third party IT providers to process data for direct care for 1.6 million patients - unfortunately for Google's AI division, developing an app is not one of them."