UNIVERSITY OF SAN FRANCISCO

CHANGE THE WORLD FROM HERE

MASTER OF DATA SCIENCE

Multiple-Instance Learning on Pathology Slides

Lori Sheng

UCSF Lab

Mentors:

Dmytro Lituiev, UCSF

Andrew Shaw, USF

Background

- **Computer Vision:**
  - Image classification
  - Segmentation

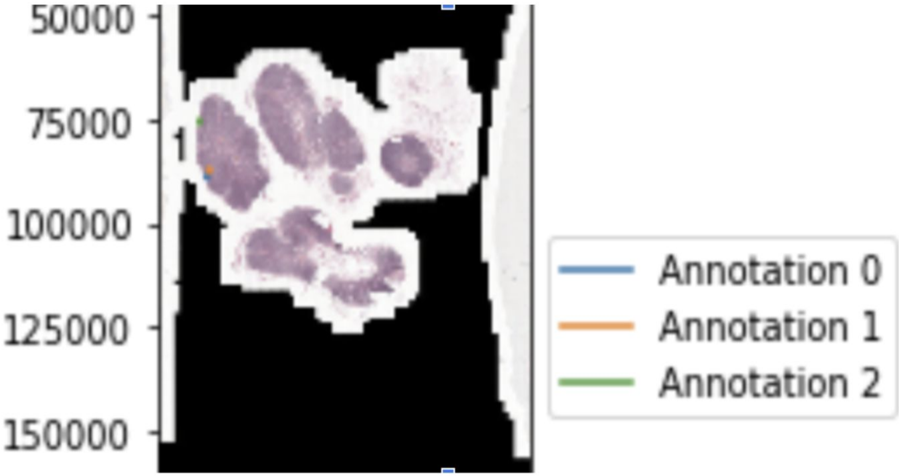


Figure 1: a slide from patient_004_node_004 (Camelyon17)

- **Problem for Pathology slides:**
  - Time-consuming and Labor-intensive (Obtaining labels)
  - Too large (200k x 100k pixels resolution)
  - Cannot downsample

Multiple-Instance Learning

- **Weakly supervised learning method:**
  - Takes a set of labeled bags containing many instances instead of receiving a set of labeled instances.
  - Slides = “bag” & Tiles = “instances”
- **Problem Statement:**
  - Each instance x_*i* in one bag has a label y_*i*. We define the label of the bag as:

Y = 1, if there exists a y_*i* such that y_*i* ==1

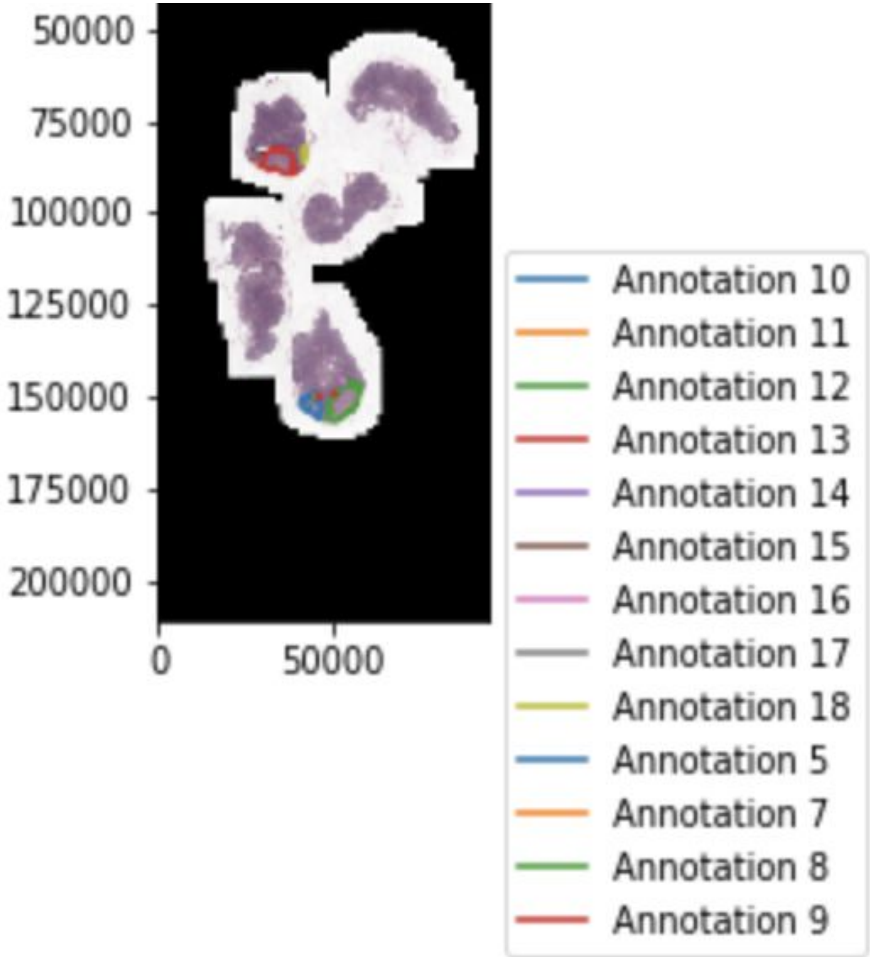
Y = 0, if for every y_*i* such that y_*i* == 0

Benefits for Pathology Slides

- **Save the labeling effort**
- **Leverage weakly labeled data**
- **Real life Cases:**
  - Camelyon17 Challenge
  - PANDA Challenge (Prostate Cancer)

Camelyon17

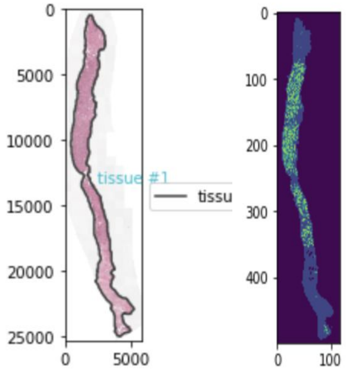
- Breast Cancer
- 1000 slides (.tiff file) from 5 medical centers
- Annotation in coordinates
- Each slide: 200k x 100k pixels
- Goal:
  - Predict “metastases” or “normal” on slide-level
  - Predict “metastases” or “normal” on patient-level

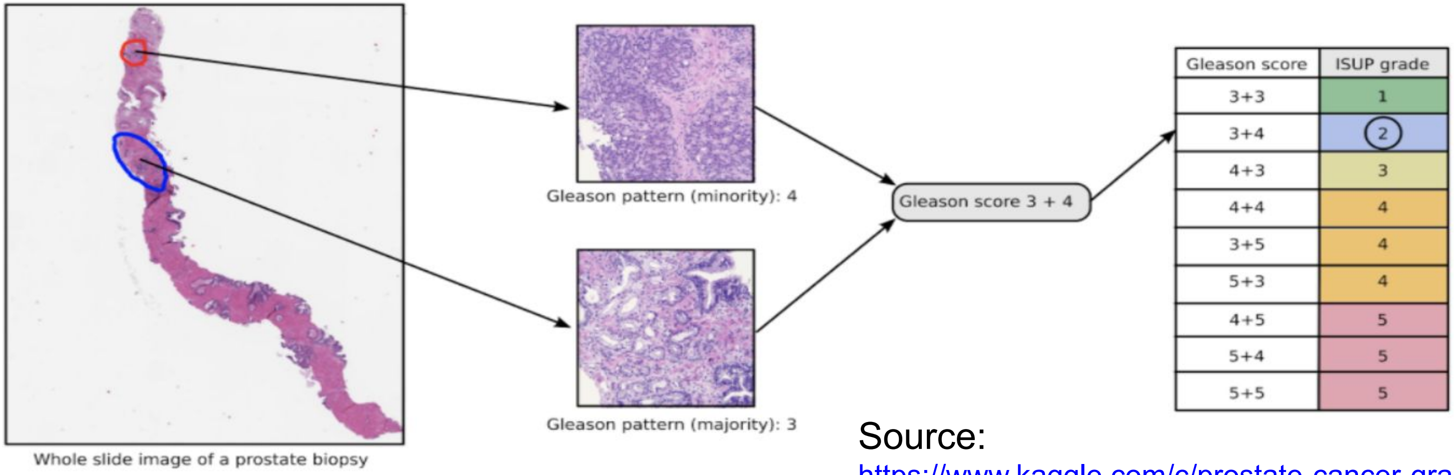


Source: https://camelyon17.grand-challenge.org/Data/

PANDA Challenge

- Prostate Cancer
- 11,000 slides (.tiff files) from 2 sources
- Masks are in red channel
- Goal:
  - Detect PCa on slides
  - Estimate severity based on Gleason Score





Whole slide image of a prostate biopsy
Gleason pattern (minority): 4
Gleason pattern (majority): 3
Gleason score 3 + 4

Gleason score	ISUP grade
3+3	1
3+4	2
4+3	3
4+4	4
3+5	4
5+3	4
4+5	5
5+4	5
5+5	5

Source:
https://www.kaggle.com/c/prostate-cancer-grade-assessment

Pipeline for Real-life Examples

1. Data Preparation
    a. pre-train and train data sets
    b. Read in .tiff files, divide into tiles, and give labels to each tile

2. Pre-train and Extract features from the model

3. Multiple-Instance Learning Training

4. Evaluation

Apply MIL on MNIST (toy example)

- **MNIST**
  - Images of handwritten digits
  - Each image has one label: 0 - 9
  - Training set: 60,000 images
  - Test set: 10,000 images
  - The size of each one: 28 x 28 and greyscale.

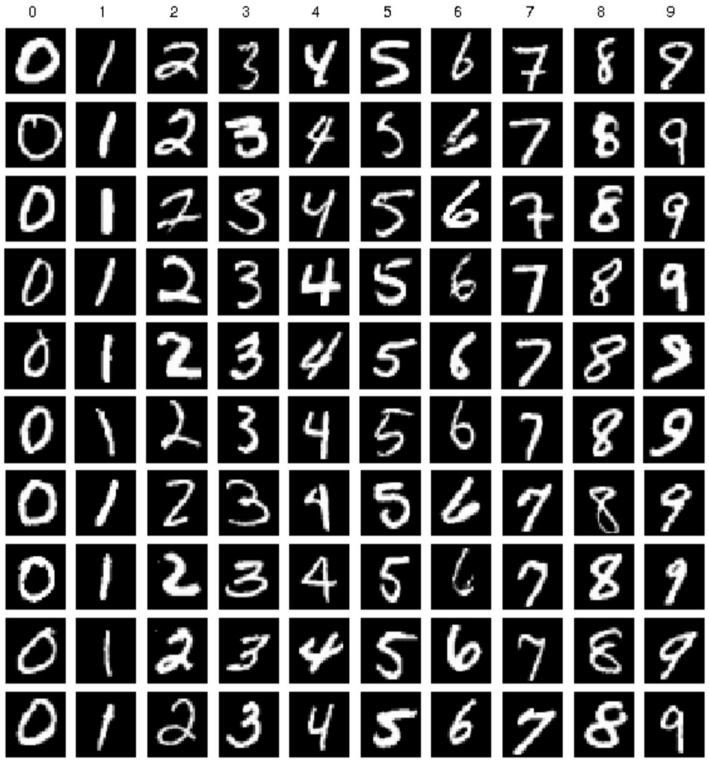


Figure 2: MNIST sample

Create MIL-MNIST Toy Dataset

- Randomly assign
- Label each bag

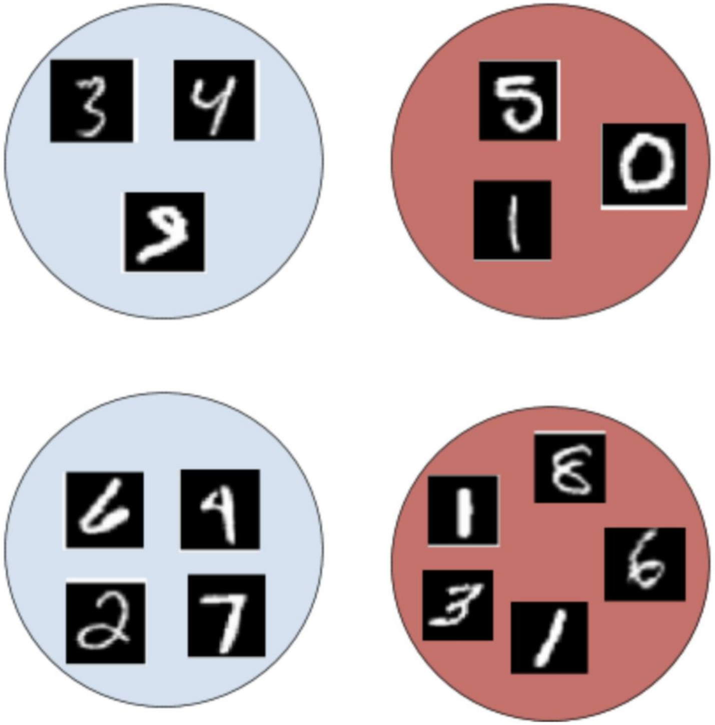
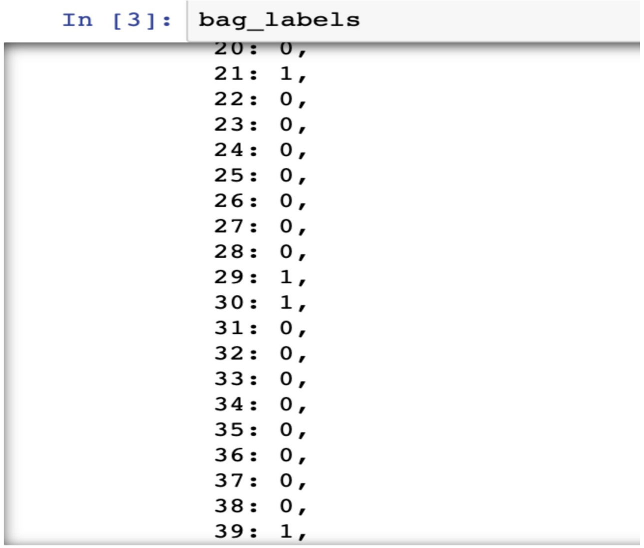


Figure 3: bags and instance labels

Load Pre-trained Model and Extract Features

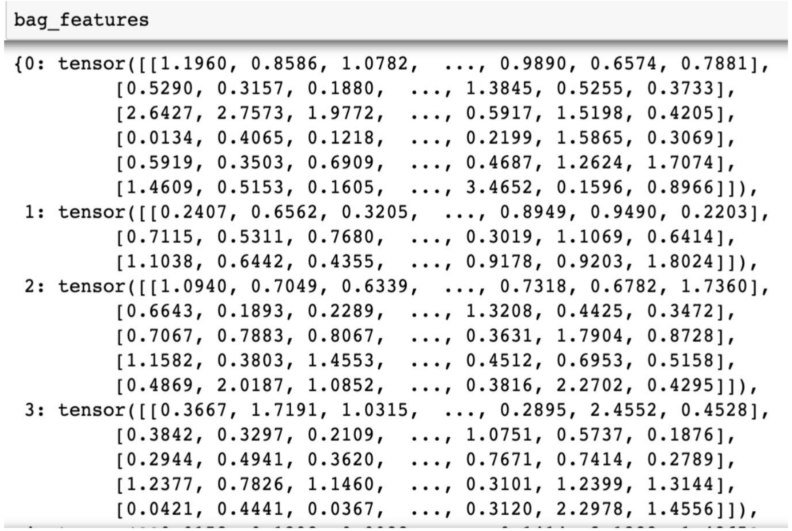
- Train, Validation, and Test set
- Get features
- Get bag_indices and bag_labels
- Map bag_indices with features based on indices and create bag_features



```
In [3]: bag_labels
```

```
        20: 0,
        21: 1,
        22: 0,
        23: 0,
        24: 0,
        25: 0,
        26: 0,
        27: 0,
        28: 0,
        29: 1,
        30: 1,
        31: 0,
        32: 0,
        33: 0,
        34: 0,
        35: 0,
        36: 0,
        37: 0,
        38: 0,
        39: 1,
```

Figure 8: Bag labels



```
bag_features
```

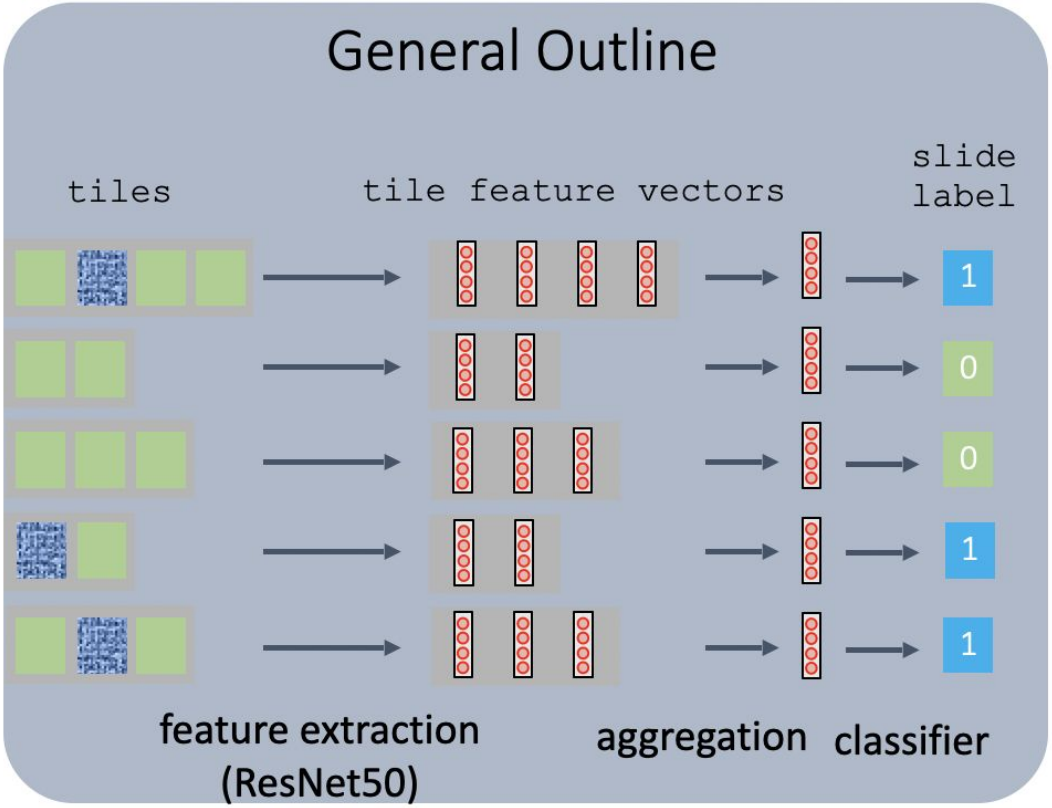
```
{0: tensor([[1.1960, 0.8586, 1.0782,  ..., 0.9890, 0.6574, 0.7881],
        [0.5290, 0.3157, 0.1880,  ..., 1.3845, 0.5255, 0.3733],
        [2.6427, 2.7573, 1.9772,  ..., 0.5917, 1.5198, 0.4205],
        [0.0134, 0.4065, 0.1218,  ..., 0.2199, 1.5865, 0.3069],
        [0.5919, 0.3503, 0.6909,  ..., 0.4687, 1.2624, 1.7074],
        [1.4609, 0.5153, 0.1605,  ..., 3.4652, 0.1596, 0.8966]]),
 1: tensor([[0.2407, 0.6562, 0.3205,  ..., 0.8949, 0.9490, 0.2203],
        [0.7115, 0.5311, 0.7680,  ..., 0.3019, 1.1069, 0.6414],
        [1.1038, 0.6442, 0.4355,  ..., 0.9178, 0.9203, 1.8024]]),
 2: tensor([[1.0940, 0.7049, 0.6339,  ..., 0.7318, 0.6782, 1.7360],
        [0.6643, 0.1893, 0.2289,  ..., 1.3208, 0.4425, 0.3472],
        [0.7067, 0.7883, 0.8067,  ..., 0.3631, 1.7904, 0.8728],
        [1.1582, 0.3803, 1.4553,  ..., 0.4512, 0.6953, 0.5158],
        [0.4869, 2.0187, 1.0852,  ..., 0.3816, 2.2702, 0.4295]]),
 3: tensor([[0.3667, 1.7191, 1.0315,  ..., 0.2895, 2.4552, 0.4528],
        [0.3842, 0.3297, 0.2109,  ..., 1.0751, 0.5737, 0.1876],
        [0.2944, 0.4941, 0.3620,  ..., 0.7671, 0.7414, 0.2789],
        [1.2377, 0.7826, 1.1460,  ..., 0.3101, 1.2399, 1.3144],
        [0.0421, 0.4441, 0.0367,  ..., 0.3120, 2.2978, 1.4556]]),
```

Figure 9: Bag features

Train the Model for MIL

Model for Multiple Instance Learning:

1. Transforms instances to an embedding of low-dimensionality. (fixed)
2. Passes the embedding through a permutation-invariant aggregation function. (optimizable)
3. Transforms into the bag probability. (optimizable)



Dmytro S Lituiev, Sung Jik Cha, Andrew Bishara, Jae Ho Sohn, Eugenia Rutenberg, Dexter Hadley, Zoltan Laszik. “A Deep Learning Model for Identifying Rejection in Transplant Kidney Biopsies”. Poster presentation at USCAP 2020, Los Angeles

Result: ~ 0.99 accuracy

Blog post for MIL on MNIST with end-to-end jupyter notebook:

https://medium.com/swlh/multiple-instance-learning-c49bd21f5620

Reference:

Ilse, Maximilian, et al. "Attention-Based Deep Multiple Instance Learning." *ArXiv.org*, 28 June 2018, https://arxiv.org/abs/1802.04712.

Carbonneau, Marc-André, et al. "Multiple Instance Learning: A Survey of Problem Characteristics and Applications." *ArXiv.org*, 11 Dec. 2016, https://arxiv.org/abs/1612.03365.

LeCun, Yann, et al. "THE MNIST DATABASE." MNIST Handwritten Digit Database, Yann LeCun, Corinna Cortes, and Chris Burges, http://yann.lecun.com/exdb/mnist/.

https://www.researchgate.net/figure/Example-images-from-the-MNIST-dataset_fig1_306056875

Dmytro S Lituiev, Sung Jik Cha, Andrew Bishara, Jae Ho Sohn, Eugenia Rutenberg, Dexter Hadley, Zoltan Laszik. "A Deep Learning Model for Identifying Rejection in Transplant Kidney Biopsies". Poster presentation at USCAP 2020, Los Angeles