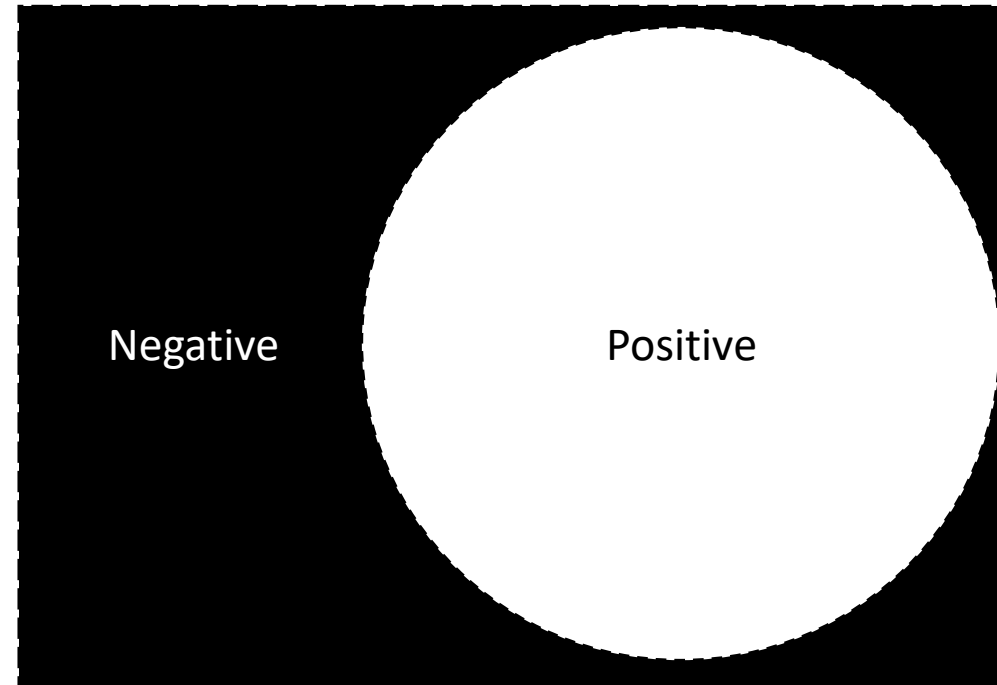


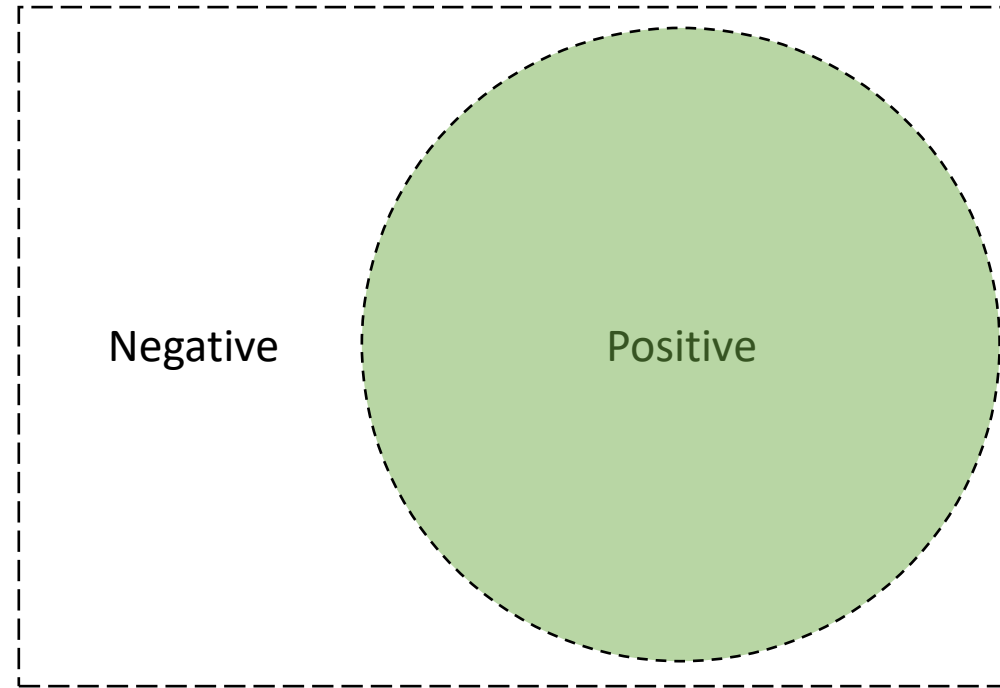
Segmentation quality estimation

Robert Haase

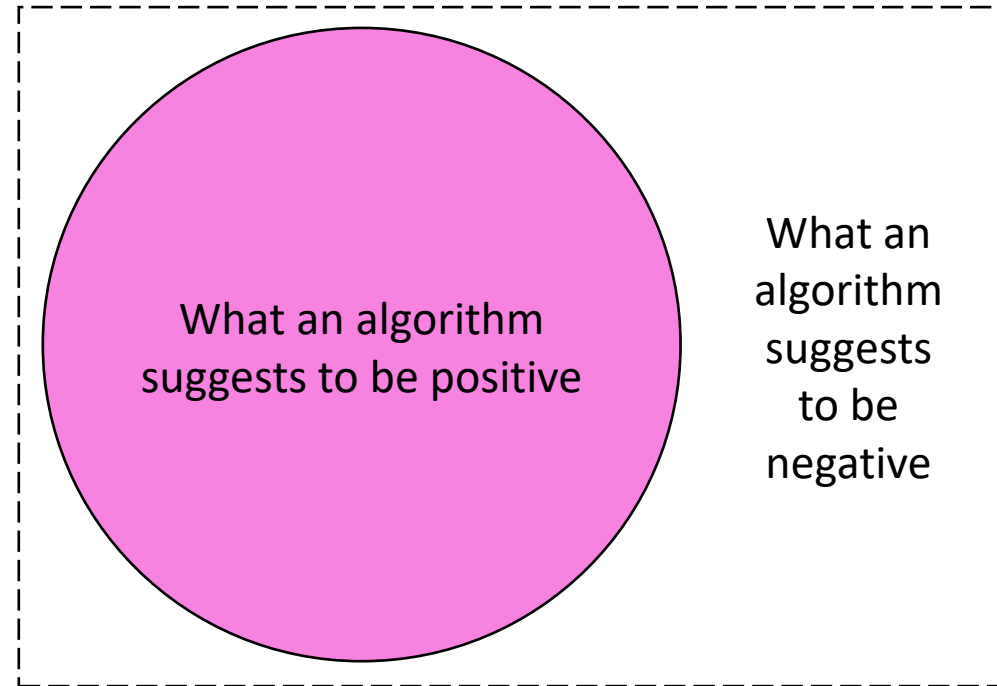
- In general
 - Define what's positive and what's negative.



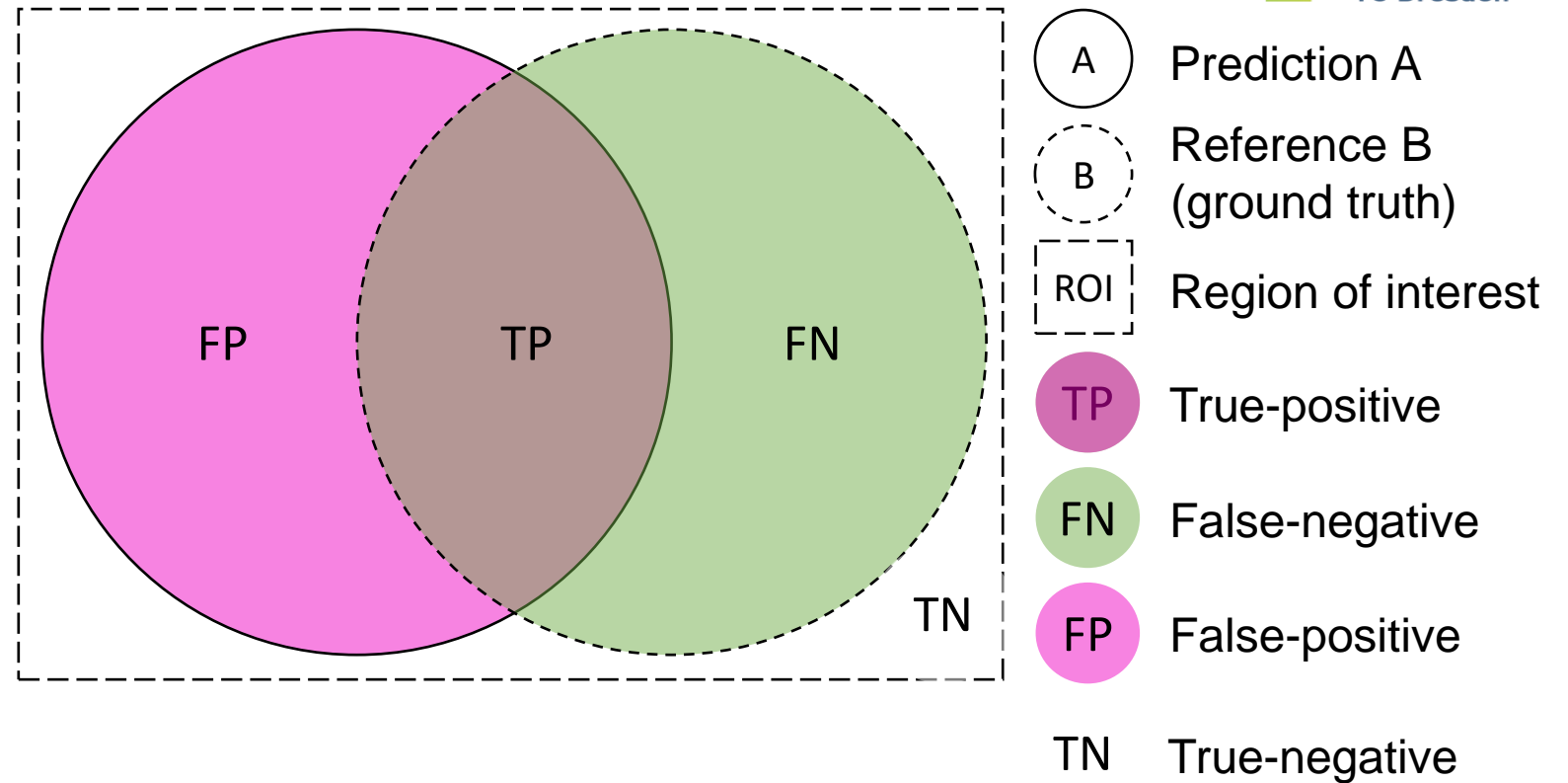
- In general
 - Define what's positive and what's negative.



- In general
 - Define what's positive and what's negative.



- In general
 - Define what's positive and what's negative.
 - Compare with a reference to figure out what was true and false
- Welcome to the Theory of Sets



Precision

$$\frac{TP}{TP + FP}$$

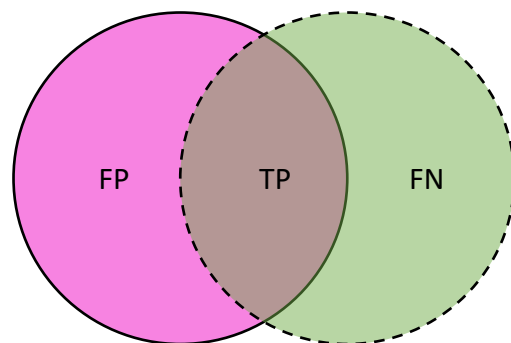
What fraction of points that were predicted as positives were really positive?

Recall
(a.k.a. sensitivity)

$$\frac{TP}{TP + FN}$$

What fraction of positives points were predicted as positives?

- Pixel wise: Segmentation quality

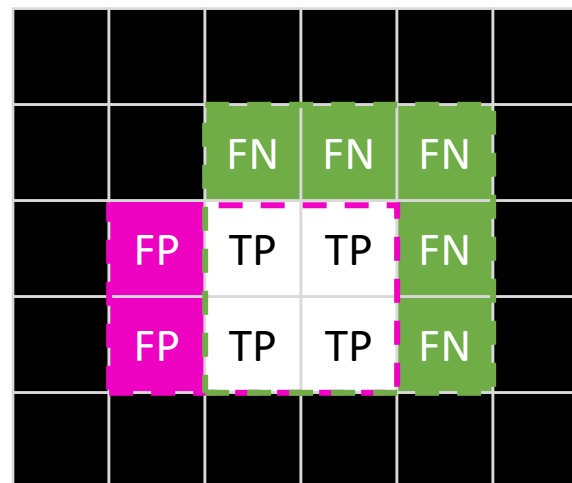


Precision

$$\frac{TP}{TP + FP}$$

Recall
(a.k.a. sensitivity)

$$\frac{TP}{TP + FN}$$



True-positive: 4

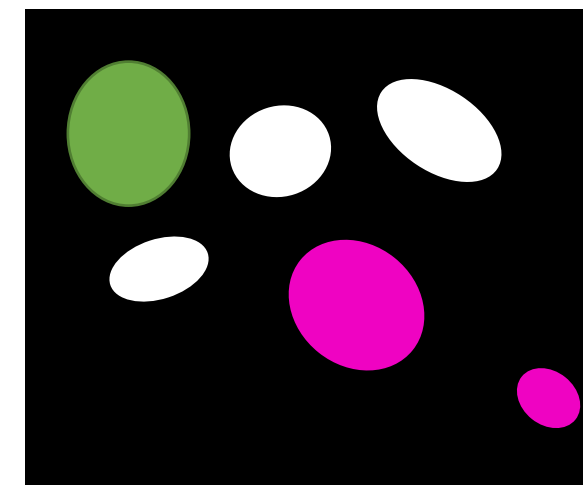
False-negative: 5

False-positive: 2

Precision: $4/6 = 66\%$

Recall: $4/9 = 44\%$

- Object wise: Detection quality



True-positive: 3

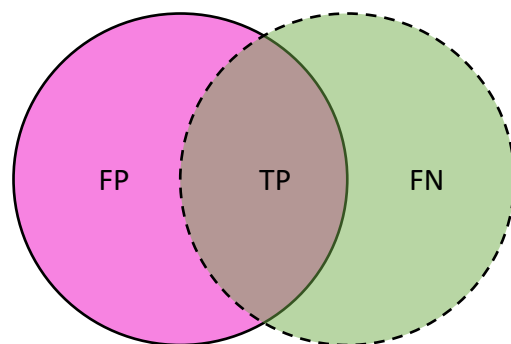
False-negative: 1

False-positive: 2

Precision: $3/4 = 75\%$

Recall: $3/5 = 60\%$

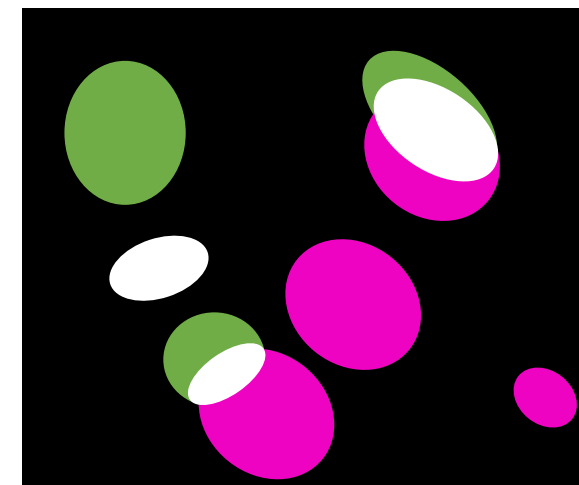
- In practice: Mixed



Intersection-over-union
(a.k.a. Jaccard index) $\frac{TP}{TP + FN + FP}$

Precision $\frac{TP}{TP + FP}$

Recall
(a.k.a. sensitivity) $\frac{TP}{TP + FN}$



Objects with at least 50%
pixel-wise overlap between
P and GT

True positive: 2
False positives: 3
False negatives: 2

Precision: $2/5 = 40\%$
Recall: $2/4 = 50\%$

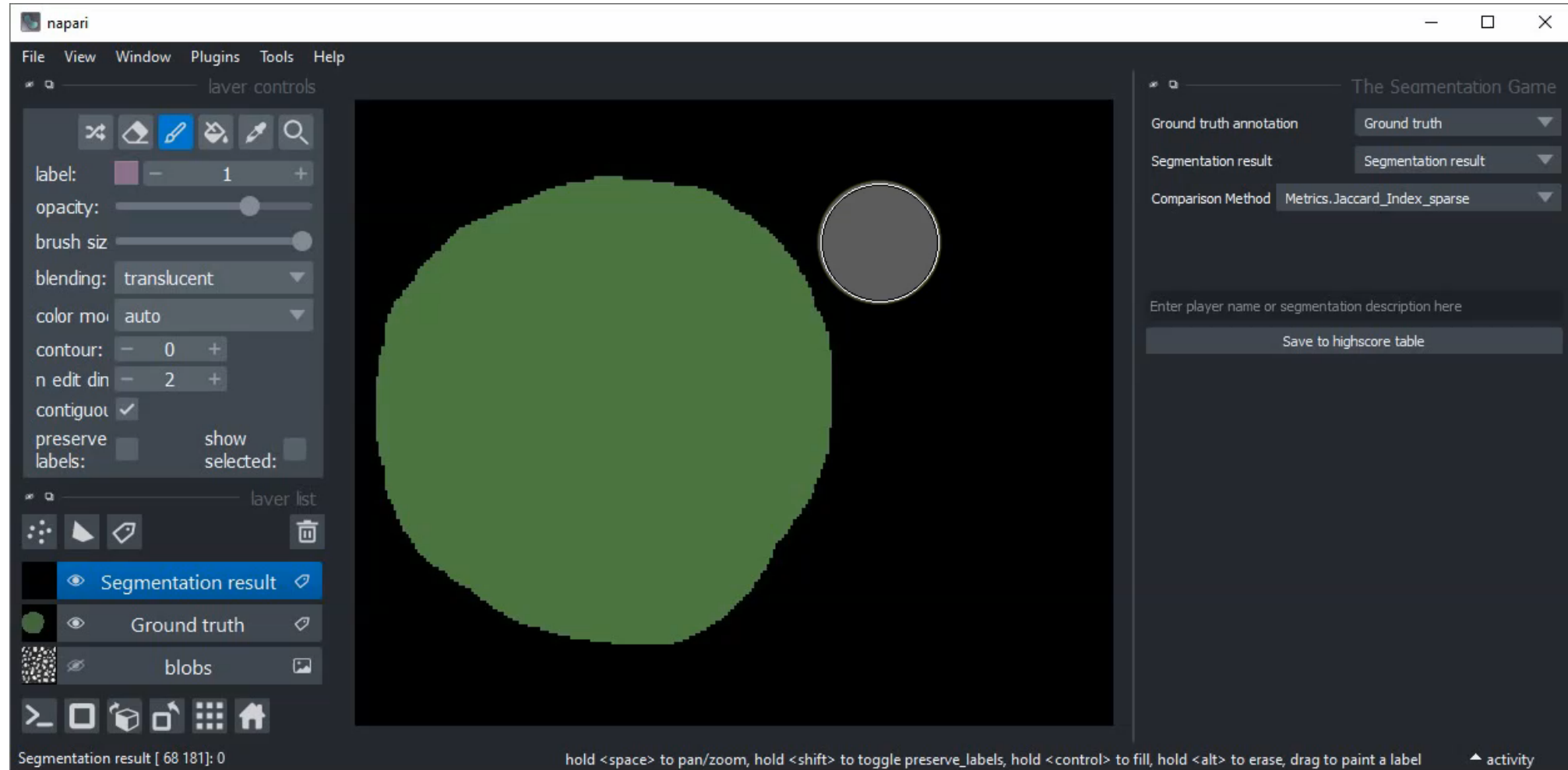
Average Overlap for all
ground-truth objects:

$$(0 + 1 + 0.5 + 0.2) / 4$$

$\approx 43\%$

Pixel-wise versus Object-wise evaluation

- Average Overlap for all ground-truth objects
- <https://github.com/haesleinhuepf/the-segmentation-game>



- Voxel-wise Youden-Index

$$YI = p_{TP} + p_{TN} - 1$$

- Volume error

$$\Delta_V = V_A - V_B$$

$$\delta_V = \frac{\Delta_V}{V_B}$$

- Dice Index

$$DI(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

- Jaccard Index

$$JI(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{DI}{2 - DI}$$

- Contour distance

$$d_{e,min}(a, B) = \min(d_e(a, b) | b \in B)$$

$$\bar{d}_c(A, B) = \frac{\sum_{\forall a \in C(A)} d_{e,min}(a, C(B))}{|C(A)|}$$

$$\bar{d}_{bil,c}(A, B) = \frac{\bar{d}_c(A, B) + \bar{d}_c(B, A)}{2}$$

- Hausdorff distance

$$d_H(A, B) = \max(d_{e,min}(a, B) | a \in A)$$

$$d_{bil,H}(A, B) = \max(d_H(A, B), d_H(B, A))$$

- Simplified Hausdorff distance

$$d_H(A, B) = \max(d_{e,min}(a, C(B)) | a \in C(A))$$

- Volume standard deviation

$$\delta_{\bar{V}} = 2 \frac{|V_A - V_B|}{|V_A + V_B|}$$

- Classification error

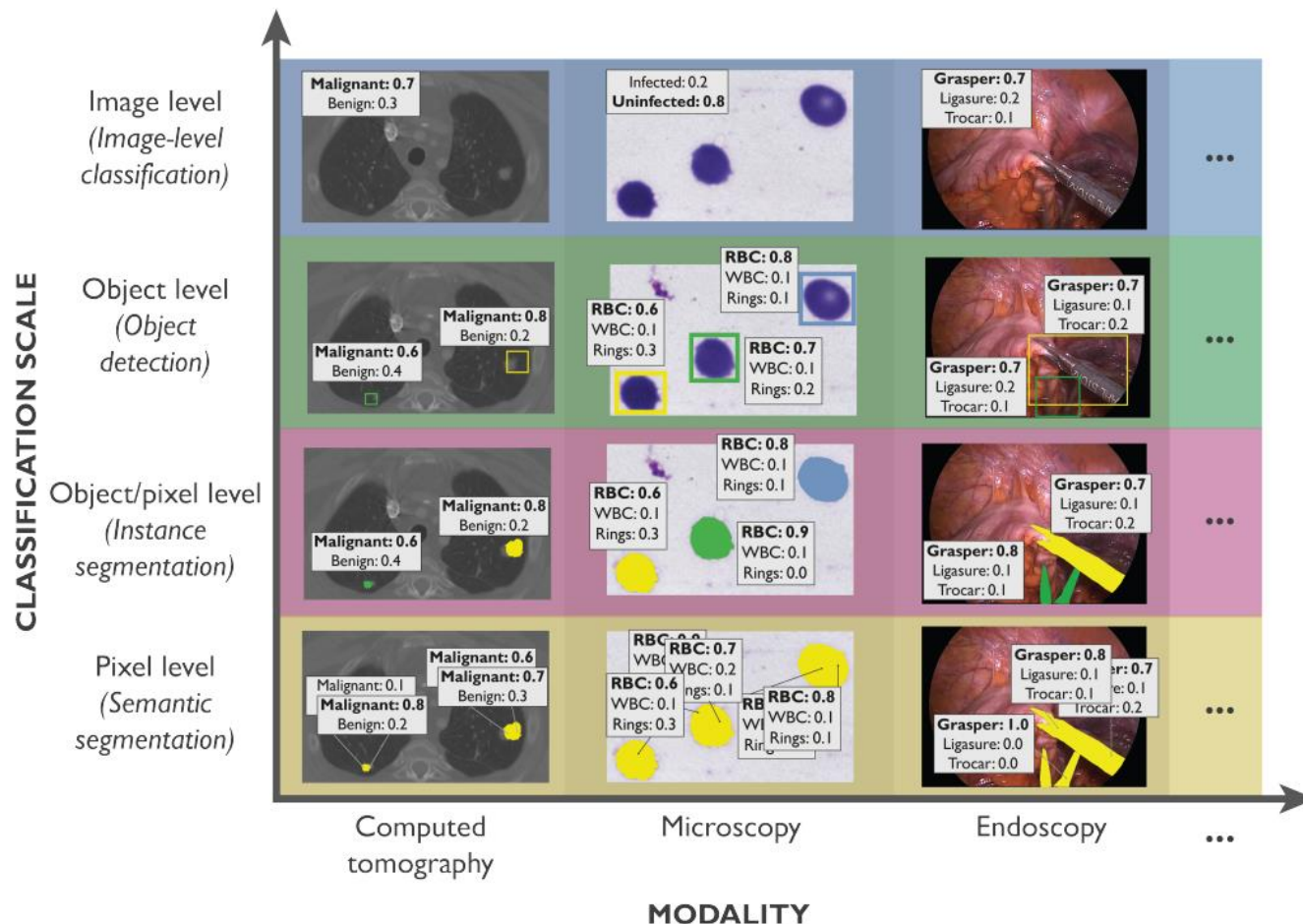
$$e_{Class} = \frac{H}{|TP| + |FN|}$$

- Hamming distance

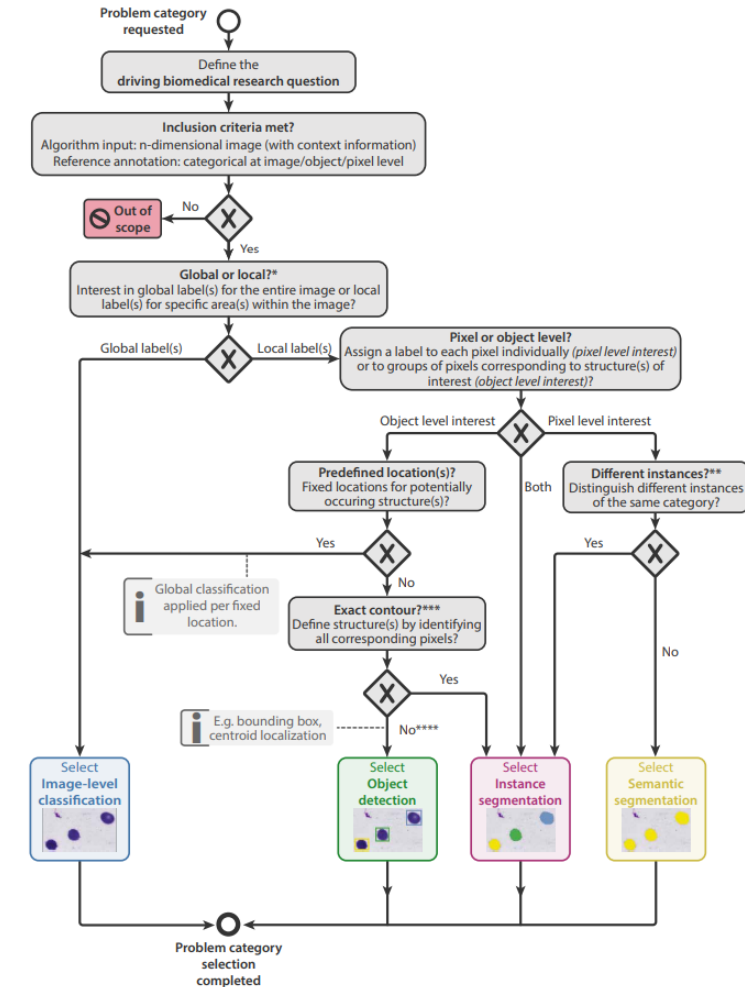
$$\begin{aligned} d_h &= |A \cup B| - |A \cap B| \\ &= |FP| + |FN| \end{aligned}$$

What metric to use when?

- “Metrics reloaded: Pitfalls and recommendations for image analysis validation”
 Maier-Hein, Reinke et al. <https://arxiv.org/abs/2206.01653>



+ S1



- Use *The Segmentation Game* from Python to measure the quality of a segmentation algorithm applied to a folder of images.

```
metrics.jaccard_index_sparse(sparse_labels, labels)
```

```
0.8357392602053431
```

```
for image_filename in os.listdir(image_folder):  
    print(image_folder + image_filename)
```

```
../../../../data/BBBC007_batch/17P1_POS0013_D_1UL.tif  
../../../../data/BBBC007_batch/20P1_POS0005_D_1UL.tif  
../../../../data/BBBC007_batch/20P1_POS0007_D_1UL.tif  
../../../../data/BBBC007_batch/20P1_POS0010_D_1UL.tif  
../../../../data/BBBC007_batch/A9_p7d.tif  
../../../../data/BBBC007_batch/AS_09125_040701150004_A02f00d0.tif
```