Image segmentation – evaluation metrics

In computer vision, **image segmentation** is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. [1][2] Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. [3]

I. Evaluation metrics:

When trying out different segmentation methods, how do you know which one is best? If you have a *ground truth* or *gold standard* segmentation, you can use various metrics to check how close each automated method comes to the truth.

1. **Pixel accuracy** - the percent of pixels in the image that are classified correctly [5]

$$\boldsymbol{Pixel_{accuracy}} = \frac{\sum_i n_{ii}}{\sum_i t_i} = \frac{TP+TN}{TP+TN+FP+FN}$$

2. **Mean accuracy over classes** [5]

$$\boldsymbol{Mean\_accuracy} = \frac{1}{n_{cl}} \cdot \frac{\sum_i n_{ii}}{t_i}$$

3. **Mean IoU (Mean Intersection over Union)** [5] - also known as the Jaccard Index, is one of the most commonly used metrics in image segmentation. IoU is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth. This metric ranges from 0–1 (0–100%) with 0 signifying no overlap and 1 signifying perfectly overlapping segmentation. For **binary** (two classes) or **multi-class segmentation**, the mean IoU of the image is calculated by **taking the IoU of each class and averaging them**.

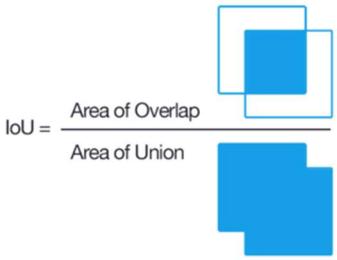

Figure 1. IoU calculation visualized. Source: Wikipedia

$$\boldsymbol{IoU} = \frac{1}{n_{cl}} \cdot \frac{\sum_i n_{ii}}{t_i + \sum_j n_{ij} - n_{ii}} = \frac{TP}{TP + FN + FP}$$

Where, n_{ij} – number of pixels of class i predicted to belong to class j, n_{cl} – number of classes, t_i – total number of pixels of class i.

4. **Precision -** a metric of exactness or quality [4],

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

5. **Recall -** a metric of completeness or quantity [4]

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}}$$

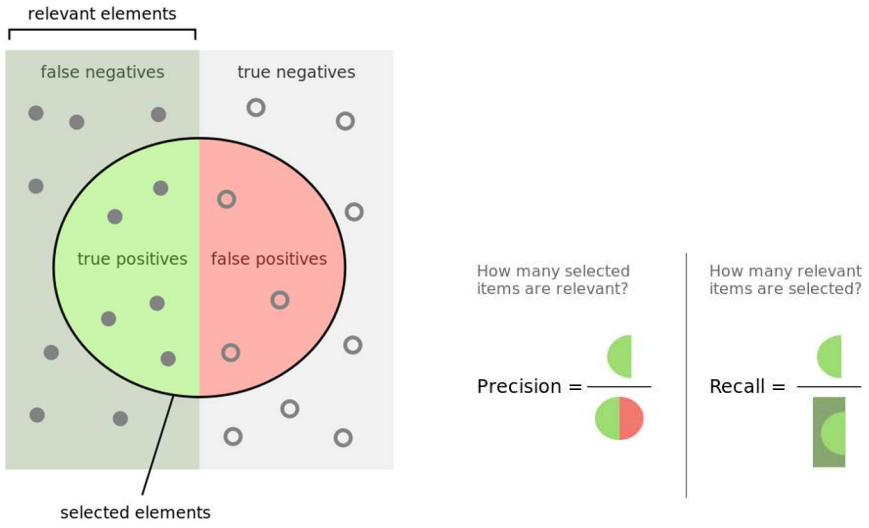

Figure 2 Precision and Recall

6. **F-measure**, also known as *Sørensen–Dice* coefficient or *Dice similarity coefficient* [4]

$$\boldsymbol{F - measure} = \frac{\boldsymbol{2 * Recall * Precision}}{\boldsymbol{Recall + Precision}}$$

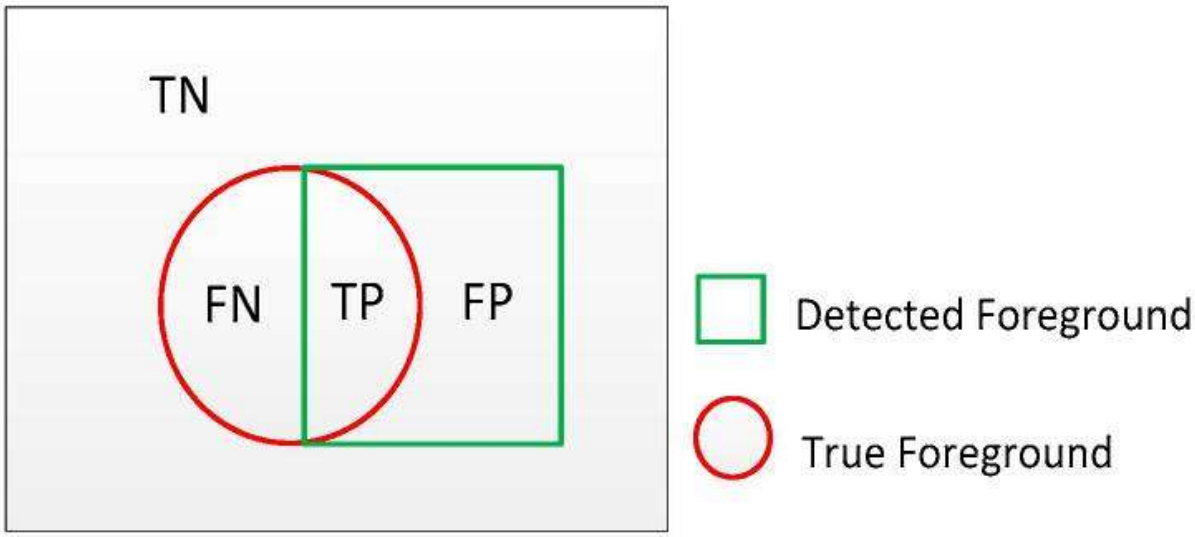

Figure 3 TP, TN, FP and FN

The scores of **precision** and **recall** should be both high, but there is an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other. The **F-Measure** which is a harmonic mean of precision and recall can be viewed as a compromise between precision and recall. It balances the precision and recall with equal weighs, and it is high only when both recall and precision are high. A higher score of F-Measure means that the performance of the algorithm is better.

TP is the number of true positives, TN is the number of true negatives, FN is the number of false negatives and FP is the number of false positives.

II. Datasets

1. **Broad Bioimage Benchmark Collection -** https://data.broadinstitute.org/bbbc/
    a. Human HT29 colon-cancer cells
    b. Synthetic cells (BBBC004, BBBC005)
    c. Human U2OS cells (out of focus)
    d. 3D HL60 Cell line (synthetic data)
    e. 3D Colon Tissue (synthetic data)
    f. Polymerized structures

III. Compute TP, TN, FP, FN using OpenCV

```
struct result_t { int TP;
                  int FP;
                  int FN;
                  int TN; };


result_t conf_mat_2c(cv::Mat1b truth, cv::Mat1b detections) {
      CV_Assert(truth.size == detections.size);
      result_t result = { 0 };
      cv::Mat inv_truth(~truth);
      cv::Mat inv_detections(~detections);
      cv::Mat temp;
      cv::bitwise_and(detections, truth, temp);
      result.TP = cv::countNonZero(temp);
      cv::bitwise_and(detections, inv_truth, temp);
      result.FP = cv::countNonZero(temp);
      cv::bitwise_and(inv_detections, truth, temp);
      result.FN = cv::countNonZero(temp);
      cv::bitwise_and(inv_detections, inv_truth, temp);
      result.TN = cv::countNonZero(temp);
      return result;
}
```

Bibliography

[1] Linda G. Shapiro and George C. Stockman (2001), “Computer Vision”, pp 279-325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3

[2] Barghout, Lauren, and Lawrence W. Lee, “Perceptual information processing system”, Paravue Inc. U.S. Patent Application 10/618,543, filed July 11, 2003

[3] https://en.wikipedia.org/

[4] Yao G, Lei T, Zhong J, Jiang P, Jia W.,“Comparative Evaluation of Background Subtraction Algorithms in Remote Scene Videos Captured by MWIR Sensors”, *Sensors (Basel)*. 2017;17(9):1945. Published 2017 Aug 24. doi:10.3390/s17091945

[5] J. Long, E. Shelhamer and T. Darrell, “Fully convolutional networks for semantic segmentation”, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 3431-3440. doi: 10.1109/CVPR.2015.7298965