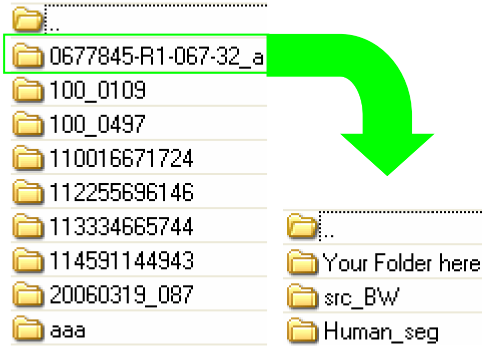
1 Introduction

This document describes the mode of operation of the evaluation code supplied with the Weizmann segmentation evaluation database.

2 Basic setup

The database contains one folder for each image, where each of which contains the human segmentations and the source image in separate folders. To use the functions supplied one must create a folder in each image folder that contains the output of the segmentations one whishes to evaluate.

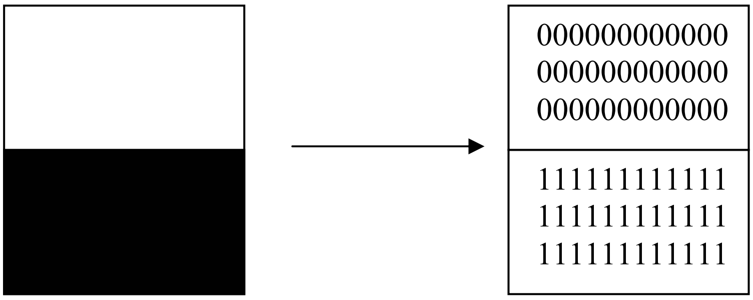


..
0677845-R1-067-32_a
100_0109
100_0497
110016671724
112255696146
113334665744
114591144943
20060319_087
aaa

..
Your Folder here
src_BW
Human_seg

Figure 1: Segmentation evaluation folders tree

The output of the evaluated method should be saved in image format where each pixel value represents the class number to whom it belongs. It is recommended to use either 8-bit or 16-bit PGM format.



```
000000000000
000000000000
000000000000
111111111111
111111111111
111111111111
```

Figure 2: Example of the evaluation output format

Note that if the results folder contains more then one output files the evaluation methods selects the one with the highest F-measure score.

3 Evaluations functions

Two Matlab evaluation methods are supplied with the database. The methods implements the two tests described in our CVPR paper.

3.1 ComputeFMeasure.m

This method implements the one segment covering test, which searches for the best single segment in terms of the F-measure score. The function usage is as follows:

```
Syntax:

       [Results]=ComputeFMeasure(DBpath,SegResultsSubPath,SysType)

Input:
       DBpath - The directory of the entire evaluation Database

       SegResultsSubPath - The name of the sub-directory  in which
      the results of the algorithm to be evaluated  are placed.

       SysType - The type of system in use, this determines the path
       separation char. There are two optional values  'win' or
       'unix' if no value is specified the default is set to 'win'.

Output:
       Results - An 100X3 matrix where Results(i,1) holds the best F-
       score for a single segment. Results(i,2) and Results(i,3)
       holds the corresponding Recall and Precision scores.
Example:
       [Results]=ComputeFMeasure('c:\Evaluation_DB','MyRes','win');
```

3.2 ComputeFMeasureMultiSeg.m

This method implements the fragmentation test, which calculates the F-measure score for a union of several segments covering the foreground object.

```
Compute the F-score for multi segments
Syntax:
       [Results]=
       ComputeFMeasureMultiSeg(DBpath,SegResultsSubPath,SysType)
Input:
       DBpath - The directory of the entire evaluation Database

       SegResultsSubPath - The name of the sub-directory in which
      results of the algorithm to be evaluated are placed.

       SysType - The type of system in use, this determines the path
                 separation char. There are two optional values 'win'
                 or 'unix', if no value is specified the default is
                 set to 'win'.

Output:
       Results - An 100X4 matrix where Results(i,1) holds the best f-
      score for a single segment. Results(i,2) and Results(i,3) holds
      the corresponding Recall and Precision scores.
      Results(i,4) holds the amount of fragmentation.
```