

CITS1401 Project 1 marking guide

Semester 2, 2022

There are 22 marks for functionality which are checked through 11 tests. By default, each student will get full marks if they pass the first test. However, the rest of the tests inspect the functionality in special cases and failing these would result in deducting the marks. Please follow the guide below:

Output test cases:

1. **Test Case 1:** Check normal working of the code on standard data
 - a. OP-1 (**6 marks**): A list of lists containing the minimum (non-zero) and maximum GDis and LDis of each distance across the four expressions.
 - b. OP-2 (**4 marks**): A list of lists containing the difference between the geodesic and 3D Euclidean distances for each expression. There will be 4 lists inside the main list and each list will have eight elements.
 - c. OP-3 (**3 marks**): A list containing the average geodesic distance of the eight distances across the four expressions. This list will have 8 elements.
 - d. OP-4 (**3 marks**): A list containing the standard deviation of the 3D Euclidean distance of the eight distances across the four expressions. This list will have 8 elements. The formula to calculate standard deviation is provided at the end of this project sheet.
 - e. OP-5a (**3 marks**): The ID of the person with which the reference face has the maximum cosine similarity
 - f. OP-5b (**3 marks**): The maximum cosine similarity value. The formula to calculate cosine similarity is provided at the end of this project sheet.

NOTE: There is no partial grading unless marker thinks answers are very close.

Deductions:

2. Test Case 3: Fails or returns incorrect outputs if rows are in random order. Deduct 6 marks.
3. Test Case 4: Fails or returns incorrect outputs if any data entry is negative or zero. Deduct 2 marks.

NOTE: There is no partial credit for Test Cases 3-4. Even if a single output fails deduct full marks.

4. Other test cases / Error-state cases
You can run them one by one and test the below cases.
Below are error-state test cases:
 - If output of function is not returned in proper format as required and simply printed, then **deduct 4 marks** and grade it accordingly. This is considered as marker's intervention to fix student's code.
 - It sometimes happens that a student will make one mistake that prevents everything else from working. If you can see a quick fix and feel so inclined, please make the fix, run the code and get a run-time tests mark, but **deduct 4 marks** from that tally, as other assignments will not have the benefit of a staff assist.

Style and efficiency:

- Style (5/8) which involves intuitive variable and function names, consistent indentation, comments, etc.
 - Default is 5.
 - Deduct 1 mark if person's name or student id is not on the file to identify author of the code.
 - Deduct 2 marks if no, or scant comments are provided.
 - Deduct 2 marks if no functions other than main are created.
 - Deduct 2 mark if intuitive variable and function names, consistent indentation features are missing.Minimum: 0 mark
- Efficiency (3/8):
 - Assign 3 marks here if program compiles correctly given the constraints below.
 - Deduct 1 mark if readline() function is used in a loop or file is opened multiple times.
 - Deduct 2 marks if code include repeated blocks instead of loops or code has more loops than necessary.
 - Deduct 2 mark if program is taking long time or program is not doing anything significant.

If student has imported any other module, then comment it out and run the code and grade it accordingly. Assign zero mark in efficiency.

Sample outputs:

Test-1

```
>>> OP1, OP2, OP3, OP4 = main('ExpData_Test_1.csv','A002','stats')
>>> OP1,OP2,OP3,OP4
([[40.0213, 42.7344, 37.3563, 39.4616], [107.5395, 116.0008, 99.5575, 103.7238],
[38.4383, 41.1045, 37.8429, 40.6916], [71.795, 74.5215, 40.6003, 44.7781], [6.6755,
13.0617, 6.5871, 11.4378], [8.8589, 12.163, 8.3796, 12.1009], [153.3032, 180.1039,
120.9159, 128.3745], [149.9468, 177.1528, 106.6833, 116.8893]], [[2.6649, 9.4455,
0.1723, 28.0613, 0.1466, 0.5331, 37.6973, 43.2635], [3.6243, 12.7156, 0.7399,
31.1946, 0.0884, 0.0622, 51.7294, 65.6837], [3.2728, 9.1085, 1.5167, 29.1326, 0.8132,
0.4793, 32.3873, 45.6054], [2.7498, 7.9821, 0.4127, 29.7434, 1.6238, 1.2854, 38.501,
51.6976]], [41.4531, 111.4748, 39.8075, 72.7733, 9.2786, 11.0559, 163.9288,
163.7033], [0.7762, 1.8584, 1.1233, 1.5766, 1.8612, 1.3383, 2.7557, 3.6976])

>>> OP5a, OP5b = main('ExpData_Test_1.csv','A002','FR')
>>> OP5a, OP5b
('A033', 0.9995)
```

Test-2

```
>>> OP5a, OP5b = main('ExpData_Test_1.csv','a012','FR')
>>> OP5a, OP5b
('B083', 0.9995)
```

Test-3

```
>>> OP1, OP2, OP3, OP4 = main('ExpData_Test_3.csv','B011','stats')
>>> OP1,OP2,OP3,OP4
([[37.4066, 39.2145, 35.8783, 37.0626], [100.616, 110.9733, 97.5039, 103.7748],
[37.19, 39.5387, 37.4681, 39.6036], [61.793, 65.3944, 34.4908, 39.6219], [10.641,
13.5974, 9.6986, 11.8726], [9.6133, 16.2964, 9.4765, 12.8275], [163.4311, 172.4022,
```

```
129.8188, 134.5737], [165.814, 183.1507, 127.8445, 130.2315]], [[2.1518, 5.6381, -
0.0649, 25.9978, 1.4018, 0.7809, 35.3823, 50.4792], [2.8449, 8.8113, 0.0355, 27.3022,
1.1626, 0.1368, 37.8285, 52.9192], [3.1116, 7.1985, -0.2781, 25.3537, 0.9423, 3.4689,
33.6979, 46.0873], [1.5284, 3.1121, -0.1001, 25.8582, 1.7248, 0.0738, 33.6123,
37.4413]], [38.628, 106.0859, 38.2652, 64.0193, 12.1915, 12.3723, 166.9751,
175.4044], [0.4891, 2.3827, 0.8254, 2.0775, 0.8684, 1.191, 1.7518, 0.9208]]
```

Test-4

```
>>> OP1, OP2, OP3, OP4 = main('ExpData_Test_4.csv','A002','stats')
```

```
>>> OP1,OP2,OP3,OP4
```

```
([[40.0213, 42.7344, 37.3563, 50], [107.5395, 116.0008, 99.5575, 103.7238],
[38.4383, 41.1045, 37.8429, 40.6916], [50, 74.5215, 40.6003, 50], [6.6755, 50, 6.5871,
11.4378], [8.8589, 12.163, 8.3796, 12.1009], [153.3032, 180.1039, 120.9159,
128.3745], [149.9468, 177.1528, 106.6833, 116.8893]], [[2.6649, 9.4455, 0.1723,
28.0613, 42.6415, 0.5331, 37.6973, 43.2635], [-8.342, 12.7156, 0.7399, 9.3997,
0.0884, 0.0622, 51.7294, 65.6837], [3.2728, 9.1085, 1.5167, 22.9003, 0.8132, 0.4793,
32.3873, 45.6054], [2.7498, 7.9821, 0.4127, 29.7434, 1.6238, 1.2854, 38.501,
51.6976]], [41.4531, 111.4748, 39.8075, 67.3245, 19.9023, 11.0559, 163.9288,
163.7033], [5.0407, 1.8584, 1.1233, 3.3782, 1.8612, 1.3383, 2.7557, 3.6976])
```

```
>>> OP5a, OP5b = main('ExpData_Test_4.csv','A002','FR')
```

```
>>> OP5a, OP5b
```

```
('E050', 0.9907)
```