CITS1401 Project 2 marking guide

Semester 1, 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 but different than sample.
   a. OP-1 **(6 marks):** A list of two dictionaries containing the facial asymmetry values between the original and mirrored face for the landmarks mentioned in Table-1 for each face F1 and F2 respectively.
   b. OP-2 **(6 marks):** A list of two dictionaries containing the facial distances (as mentioned in Table-2) for each face F1 and F2 respectively.
   c. OP-3 **(6 marks):** A list of Tuple sequences of the 5 faces having the lowest total face asymmetry.
   d. OP-4 **(4 marks):** The cosine similarity between faces F1 and F2 based on the six distances calculated above (OP2).

NOTE: **ONLY for Test Case 1:** For OP-1 and OP-2, give 6 marks if the results of both subject's match, give 3 marks if the results of only one subject matches. For OP-3, give 6 marks if all 5 tuple sequences match, give 3 marks if any three sequences match.

For all remaining Test Cases, there are no partial marks. Deduct the required marks even if a single entry in the outputs is incorrect.

Other test cases/ error-state cases:

You can run them one by one and test the below cases

2. Test Case 2: Fails to test if the file does not exist. **Deduct 1 mark**.
3. Test Case 3: Fails to test if the Adult ID is not case insensitive. This means that the code should return the same result whether the ID is in upper or lower case. **Deduct 1 mark**.
4. Test Case 4: Fails to test if the SubjID is a list or not. **Deduct 1 mark.**
5. Test Case 5: Fails if the columns are in a different order. **Deduct 2 marks**.
6. Test Case 6: Fails or return incorrect outputs if rows are in random order. **Deduct 5 marks**.
7. Test Case 7: Fails to detect a missing X, Y or Z coordinate. **Deduct 3 marks**.
8. Test Case 8: Fails to detect an out of bound X, Y or Z coordinate. **Deduct 3 marks.**
9. Test Case 9: Fails or returns incorrect outputs if one or more landmarks (rows) are missing. **Deduct 3 marks**.
10. Test Case 10: Fails or returns incorrect results if the asymmetry at nose tip is not zero. **Deduct 3 marks**. Note that the file for this test is large. Hence it can be used to test the efficiency of the algorithm.

   - If output of function is not returned in proper format as required and simply printed, then deduct 4 marks as mentioned in the end 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 and 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.

How to perform Self-Test: You have been provided with this grading rubric and 9 test files. Each file should be used for its corresponding test case (except Test-2) as shown below. Each file tests your code for only the specific condition given in the test case. TestData1.csv is the base-line test file which is exactly same as the sample file you were provided. If your code worked at the time of submission, it should work on TestData1.csv as well. Also **note** that TestData10.csv is very large and was used to test the efficiency of your code.

Sample outputs:

Test-1
```
OP = main('TestData1.csv',['D8328','E4996'])
>>> OP
([{'FT': 1.745, 'EX': 1.1983, 'EN': 2.139, 'AL': 1.3741, 'SBAL': 2.2901, 'CH': 1.6568}, {'FT': 1.6069, 'EX': 0.8966, 'EN': 0.9502, 'AL': 1.9318, 'SBAL': 1.4964, 'CH': 1.4435}], [{'EXEN': 30.0446, 'ENAL': 35.1095, 'ALEX': 46.2359, 'FTSBAL': 99.128, 'SBALCH': 28.0281, 'CHFT': 97.7108}, {'EXEN': 31.215, 'ENAL': 38.1904, 'ALEX': 49.7788, 'FTSBAL': 97.3245, 'SBALCH': 33.1105, 'CHFT': 99.8042}], [('I0951', 8.1453), ('E4996', 8.3254), ('H1178', 9.1597), ('F7831', 9.3268), ('J6687', 9.3878)], 0.9991)
```

Test-2
```
OP = main('TestDataNone.csv',['D8328','E4996'])
File not found error
```

Test-3
```
OP = main('TestData3.csv',['d8328','e4996'])
>>> OP
([{'FT': 1.745, 'EX': 1.1983, 'EN': 2.139, 'AL': 1.3741, 'SBAL': 2.2901, 'CH': 1.6568}, {'FT': 1.6069, 'EX': 0.8966, 'EN': 0.9502, 'AL': 1.9318, 'SBAL': 1.4964, 'CH': 1.4435}], [{'EXEN': 30.0446, 'ENAL': 35.1095, 'ALEX': 46.2359, 'FTSBAL': 99.128, 'SBALCH': 28.0281, 'CHFT': 97.7108}, {'EXEN': 31.215, 'ENAL': 38.1904, 'ALEX': 49.7788, 'FTSBAL': 97.3245, 'SBALCH': 33.1105, 'CHFT': 99.8042}], [('I0951', 8.1453), ('E4996', 8.3254), ('H1178', 9.1597), ('F7831', 9.3268), ('J6687', 9.3878)], 0.9991)
```

Test-4
```
OP = main('TestData4.csv',('D8328','E4996'))
```

```
>>> OP
(None, None, None, None)
```

Test-5

```
OP = main('TestData5.csv',['D8328','E4996'])
>>> OP
([{'FT': 1.745, 'EX': 1.1983, 'EN': 2.139, 'AL': 1.3741, 'SBAL': 2.2901, 'CH': 1.6568}, {'FT': 1.6069, 'EX': 0.8966, 'EN': 0.9502, 'AL': 1.9318, 'SBAL': 1.4964, 'CH': 1.4435}], [{'EXEN': 30.0446, 'ENAL': 35.1095, 'ALEX': 46.2359, 'FTSBAL': 99.128, 'SBALCH': 28.0281, 'CHFT': 97.7108}, {'EXEN': 31.215, 'ENAL': 38.1904, 'ALEX': 49.7788, 'FTSBAL': 97.3245, 'SBALCH': 33.1105, 'CHFT': 99.8042}], [('I0951', 8.1453), ('E4996', 8.3254), ('H1178', 9.1597), ('F7831', 9.3268), ('J6687', 9.3878)], 0.9991)
```

Test-6

```
OP = main('TestData6.csv',['D8328','E4996'])
>>> OP
([{'FT': 1.745, 'EX': 1.1983, 'EN': 2.139, 'AL': 1.3741, 'SBAL': 2.2901, 'CH': 1.6568}, {'FT': 1.6069, 'EX': 0.8966, 'EN': 0.9502, 'AL': 1.9318, 'SBAL': 1.4964, 'CH': 1.4435}], [{'EXEN': 30.0446, 'ENAL': 35.1095, 'ALEX': 46.2359, 'FTSBAL': 99.128, 'SBALCH': 28.0281, 'CHFT': 97.7108}, {'EXEN': 31.215, 'ENAL': 38.1904, 'ALEX': 49.7788, 'FTSBAL': 97.3245, 'SBALCH': 33.1105, 'CHFT': 99.8042}], [('I0951', 8.1453), ('E4996', 8.3254), ('H1178', 9.1597), ('F7831', 9.3268), ('J6687', 9.3878)], 0.9991)
```

Test-7

```
OP = main('TestData7.csv',['D8328','E4996'])
>>> OP
([None, {'FT': 1.6069, 'EX': 0.8966, 'EN': 0.9502, 'AL': 1.9318, 'SBAL': 1.4964, 'CH': 1.4435}], [None, {'EXEN': 31.215, 'ENAL': 38.1904, 'ALEX': 49.7788, 'FTSBAL': 97.3245, 'SBALCH': 33.1105, 'CHFT': 99.8042}], [('E4996', 8.3254), ('F7831', 9.3268), ('J6687', 9.3878), ('B7033', 10.6282), ('P9721', 10.8609)], None)
```

Test-8

```
OP = main('TestData8.csv',['D8328','E4996'])
>>> OP
([None, {'FT': 1.6069, 'EX': 0.8966, 'EN': 0.9502, 'AL': 1.9318, 'SBAL': 1.4964, 'CH': 1.4435}], [None, {'EXEN': 31.215, 'ENAL': 38.1904, 'ALEX': 49.7788, 'FTSBAL': 97.3245, 'SBALCH': 33.1105, 'CHFT': 99.8042}], [('E4996', 8.3254), ('F7831', 9.3268), ('J6687', 9.3878), ('B7033', 10.6282), ('P9721', 10.8609)], None)
```

Test-9

```
OP = main('TestData9.csv',['D8328','E4996'])
>>> OP
([{'FT': 1.745, 'EX': 1.1983, 'EN': 2.139, 'AL': 1.3741, 'SBAL': 2.2901, 'CH': 1.6568}, None], [{'EXEN': 30.0446, 'ENAL': 35.1095, 'ALEX': 46.2359, 'FTSBAL': 99.128, 'SBALCH': 28.0281, 'CHFT': 97.7108}, None], [('H1178', 9.1597), ('F7831', 9.3268), ('J6687', 9.3878), ('K6431', 9.6359), ('D8328', 10.4033)], None)
```

Test-10

```
OP = main('TestData10.csv',['D8328','E4996'])
>>> OP
([None, {'FT': 1.6069, 'EX': 0.8966, 'EN': 0.9502, 'AL': 1.9318, 'SBAL': 1.4964, 'CH': 1.4435}], [None, {'EXEN': 31.215, 'ENAL': 38.1904, 'ALEX': 49.7788, 'FTSBAL': 97.3245, 'SBALCH': 33.1105, 'CHFT': 99.8042}], [('I2742', 5.2126), ('N2779', 5.4834), ('E5756', 5.877), ('Z0787', 5.9875), ('T0114', 6.0518)], None)
```