

CAP 4628/5627 Affective Computing
Project 1 – 3D Expression Recognition
Due 3/20 by 11:59pm

For this project, you will analyze 3D facial landmarks by running them through 3 classic machine learning algorithms (Random Forest, SVM, and Decision Tree). You will turn in a report of your experimental results, which includes discussion (i.e., answering questions). See below for detailed instructions.

Instructions

- Your program must be written in Python. You can use any modules needed to complete this project (import). Create a python file named Project1.py, however, you are free to split up your program into multiple files.
- Run the following machine learning classifiers:
 - Random Forest (RF)
 - Support Vector Machine (SVM)
 - Decision Tree (TREE)
- For each of the above classifiers, you must run 10-fold cross validation.
- The data to train and evaluate the listed classifiers using 10-fold subject independent cross validation is available in the zip file FacialLandmarks.zip. The zip file is about 105 MB and uncompressed it is approximately 200 MB. The directory structure is as follows:

/BU4DFE_BND_V1.1

/Subject directories (e.g., F001)

/Expressions (e.g., Angry)

.BND files (text data files)

- The .bnd files have an index and one facial landmark per row; example below. You can ignore/skip the index. You only need the x, y, z coordinate from each row. There are 83 (or 84) rows in total, one for each of the 83 landmarks. If there is an 84th row, it is empty. Some (all?) may have this.

Index x y z

- The following three experiments need to be run. Each will be done with 10-fold cross validation and each of three classifiers (RF, SVM, TREE). Read the data and then perform these operations on the data before using them to train and evaluate your classifier.
 - **Use the raw 3D landmarks.** There are 83 (x, y, z) coordinates in total (249 data points). Your feature vector will be a size of 249.
 - **Translate the raw 3D landmarks (face) to the origin (0, 0, 0).** To translate a face to the origin, first calculate the average 3D landmark (for x, y, and z). This will give you the middle of the face (approximately). Once you have the average, subtract it from each of the 3D landmarks. This will center (approximately) the face at the origin. Once you have translated the face, do the same experiments as above.
 - **Rotate the raw 3D landmarks 180 degrees.** Like translations you must rotate every 3D point over the x-axis, y-axis, and z-axis, 180 degrees. You can use the following math to do this. Note: to calculate PI in python you can use the following (this will calculate PI to 3 decimal places):
From math import acos
Pi=round(2*acos(0.0), 3)

$$\begin{aligned} \text{x-axis: } & \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\pi) & \sin(\pi) \\ 0 & -\sin(\pi) & \cos(\pi) \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ \text{y-axis: } & \begin{bmatrix} \cos(\pi) & 0 & -\sin(\pi) \\ 0 & 1 & 0 \\ \sin(\pi) & 0 & \cos(\pi) \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ \text{z-axis: } & \begin{bmatrix} \cos(\pi) & \sin(\pi) & 0 \\ -\sin(\pi) & \cos(\pi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} \end{aligned}$$

- For each of these experiments you need to calculate the confusion matrix, accuracy, precision, and recall. Since you will have multiple of these, you will also need to take the average for each experiment/classifier. For the confusion matrices, you can take the average of each cell across the 10 folds. For example, you will have 10 cross-validation runs for a random forest and the original raw data. So, you will have 10 for this classifier. Take the average of each for this classifier. You must do the same for all three classifiers.
- In total, you will have 15 experiments to run, each with 10-fold cross validation – 3 classifiers X 5 different data types (original, translated, rotated across 3 axes).
- Make sure to save the results from each of your classifier runs (e.g., save to text file, or some other format).
- Your program must take the following command parameters. The exact format of the input is given below as well – python Project1.py <classifier to run> <data type to use> <data directory>
 - Classifier to run
 - **RF** – random forest
 - **SVM** – support vector machine
 - **TREE** – decision tree
 - Data type to use
 - **o** - original
 - **t** - translated
 - **x** – rotate x
 - **y** – rotate y
 - **z** – rotate z
 - Data directory

Example python run: python Project1.py RF x ./BU4DFE_BND_V1.1

This example will use the random forest classifier on rotated (over x axis) data.

- NOTE: Your python scripts should be well commented. It should be clear what each function is doing.
- All code should be modular, well-designed functions. Your python code (functions) should only run when they are called.

Report Questions

1. What are the classification results for each experiment? (Create a table of this – sample template available at end of document)
2. Which of the classifiers worked the best for each data type (original, translated, rotated)? Why? If multiple had the same, then why did this happen? Note: this question is left intentionally vague.

3. For the top classifier, for each data type, describe the misclassification for each classifier. What was misclassified as what (e.g., sad looks like happy)? Based on your intuition about how expressions look, do the misclassifications make sense (i.e., do you think the expressions are similar – not based on any specific example)?
4. Why do you think you got the results that you got for each of the different data types/classifiers (i.e., why are they different, or why are they the same)? For example, if SVM and RF have different results, why are they different? If they are the same – why are they the same?
5. You must plot one sample of each data type – original, translated, rotated (1 each for x, y, and z). In total, you will have 5 figures/plots. Note: if you are unsure how to do this, look up 3D scatter plot in python.
6. **(CAP 5627 only)** Describe, in your own words, what Random Forest, SVM, and Decision Tree classifiers are/how they work. You may need to look this information up. Your description should be approximately 100-150 words in length for each classifier.

1. Answer the above questions.
2. Show the confusion matrices for the best classifier for each data type. Make a new table, do not copy paste images from command line.
3. There is no template for the report, but it should be well done and easy to read. *You should have a section for each classifier and detailed results about the experiments in each.*

- a. PDF of your report
 - b. All python files
 - c. README file that lists any packages that you imported
1. Code/comments – 60 points
2. Report – 40 points

[illegible]

Figure 1. 83 features on face.

