A Guide to the Datasets

Experiment 1

To examine classification of the videos using their physical characteristics, we ran a Naïve Bayes (NB) classifier on the data. For each movie in the SEED dataset we held back that movie and trained the classifier on the remaining movies, then we asked the classifier to predict the emotion for the movie that we held back and recorded whether or not the prediction was correct. By repeating this process for every movie in the dataset we could record the average classification accuracy, which quantified how well our classifier is able to predict the emotion label from the physical characteristics. We repeated this process, holding back each movie in turn, to provide classification rates averaged over all movies. This process calculated classification between the movies labelled as having positive or negative valence, the same comparison as that carried out by Zheng and colleagues (see our paper)

**All the data described above, and the Python files for integrating different data matrices and running the analysis, can be found within the folder entitled “Python code for movie classification”**

**Note that within this folder is the data and Python files for the analysis of the SEED dataset and our own dataset** (using the NIMSTIM and ADFES-BIV stimuli).

Experiment 2

*Oscillatory EEG and ERP Analysis: Initial Processing of Data*

For each individual EEG epoch we computed the average power for each channel in each frequency band 4-8Hz (theta), 8-14Hz (alpha), 14-30Hz (beta) and >30Hz (gamma) and in the time intervals 0-0.5, 0.5-1 and 1-1.5 seconds from onset. The intervals were used to capture earlier to later onset activity. This resulted in 63\*4\*3=756 different features for each epoch.

We averaged these features within the subject\*block\*emotion groups, which contain roughly 40 epochs each. We call these averaged groups *grand epochs*.

Next, we took the log of all of the averaged power features. This makes the data more normally distributed and more suitable for the classification algorithms that we are going to apply. This resulted in 756 logged and averaged power features for each grand epoch.

Additionally, ERP features were calculated by taking grand averages of the raw signals for each channel and then measuring the peak amplitude, mean amplitude and peak latency for P1, N1 and P2 for each channel. These ERPs were selected due to their relevance to early face and emotion processing (Ibanez et al., 2012). Furthermore, taking early components helps constrains attention influences as 200ms or less limits attention to no more than 1 saccade. Thus, time frames of 200ms or less helps control for attention effects. This resulted in 63\*3\*3=567 ERP features for each grand epoch.

This results in a dataset consisting of 24\*6\*7=1008 grand epochs. Each grand epoch is attributed to one of the 24 subjects, one of the 6 blocks and one of the 7 emotions. Each grand epoch has 756+567=1323 features.

**All the data described above, and the MATLAB files for integrating different data matrices and running the analysis, can be found within the folder entitled “MATLAB code for processing data”**

 *Multi-class and Binary Classifications*

We analysed the data with two different tests. The first approach included all seven emotion categories from our experiment. The second approach was a binary comparison, creating classification rates for a single emotion versus another single emotion with all permutations tested. Data was collapsed across conditions and presentation type; image or video.

The aim for both classification problems was to use machine learning to construct a classifier from the data that could predict the emotion of a grand epoch from a previously unseen test subject. To estimate the accuracy of the classifier, we held back one test subject and used the remaining data (the training data) to train a classifier and then test the resulting classifier on the data of the held back subject (the test data). By repeating this process for each subject and averaging the results we obtained a valid estimate for how accurately we could classify a new, unseen, test subject.

**All the data described above, and the Python files for integrating different data matrices and running the analysis, can be found within the folder entitled “Python code for EEG\_classification”**