Multimodal Personality Analysis for Candidate Screening

Team Members - Abhishek Phalak, Piyush Patil

Problem Statement

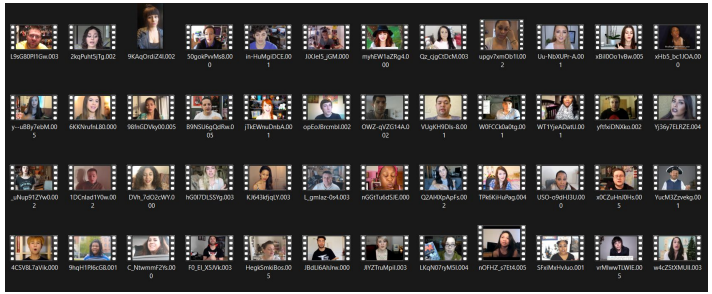
Determining human personality is crucial for understanding their internal and external states, which in turn determines a wide variety of daily and working behaviours. Our objective is to extract rich information from both visual and audio modalities of videos to accurately assess an individual's personality. Our proposed solution aims to aid in the candidate screening process, providing valuable insights that can be used to make informed decisions.

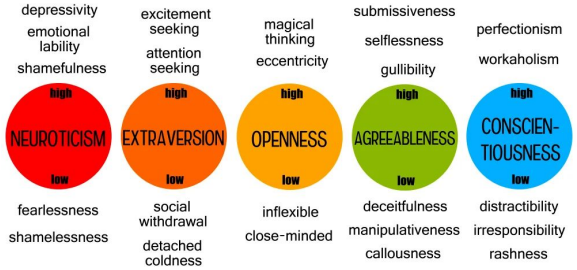
Background

- Subramaniam, A., Patel, V., Mishra, A., Balasubramanian, P., Mittal, A. (2016). Bi-modal First Impressions Recognition Using Temporally Ordered Deep Audio and Stochastic Visual Features. In: Hua, G., Jégou, H. (eds) Computer Vision – ECCV 2016 Workshops.
- X. -S. Wei, C. -L. Zhang, H. Zhang and J. Wu, "Deep Bimodal Regression of Apparent Personality Traits from Short Video Sequences," in IEEE Transactions on Affective Computing, vol. 9, no. 3, pp. 303-315, 1 July-Sept. 2018
- C. Ventura, D. Masip and A. Lapedriza, "Interpreting CNN Models for Apparent Personality Trait Regression," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 2017

Dataset

The Chalearn's First Impressions V2 (CVPR'17) data set, comprises 10000 clips (average duration 15s) extracted from more than 3,000 different YouTube high-definition (HD) videos of people facing and speaking in English to a camera. It models human personality along five dimensions: Extraversion, Agreeableness, Conscientiousness, Neuroticism & Openness.





Audio Preprocessing

Model 1

We extract audio from the video files in the form of numpy array using ffmpeg and perform following preprocessing steps -

1. Computing MFCC - To analyze the audio data, Mel-Frequency Cepstral Coefficients are computed with the n_mfcc parameter set to 24, resulting in the calculation of 24 coefficients for each audio frame.

2. Standardizing MFCC - The computed MFCC are then standardized to have zero mean and unit variance.

3. Padding and Reshaping - The standardized MFCC coefficients are padded with zeros to obtain a fixed size of (24, 1319)

Model 2

- For Audio preprocessing, to extract features we used pretrained model YAMNet, a convolutional neural network (CNN) architecture.
- We extracted audio features using YAMNet by loading an audio file, resampling it to 44100 Hz, and passing it through the YAMNet model.
- The resulting embeddings are flattened into a single vector and returned as a numpy array.

Video Preprocessing

Model 1

1. Extract Video Frames - we extracted randomly sampled frames to ensure that the extracted frames are representative of the overall content of the video

2. Resize Images - We resized each RGB image in the list to a fixed size of (248, 140) pixels

3. Crop Image window - we crop a small window of size 128 X 128 pixels ensuring that the window has a face and dropping frames which does not have a face.

4. Stacking arrays - . At last, we take the list of cropped RGB images and stack them into a single Numpy array with shape (num_of_samples=6, 128,128,3)

Model 2

- For video preprocessing, to extract features we used pretrained model VGGFace, a deep convolutional neural network that was specifically designed for face recognition tasks.
- A NumPy array representing an image is then passed as input to a function which returns a flattened feature vector representing the high-level features of the input image

Method

AUDIO MODELLING

- Input to model is 4D tensor
- We extract useful features from audio data by passing the input tensor through convolutional layers, and applying batch normalization and max pooling.
- Output is flattened and passed through dense layers to produce a feature vector that summarized the entire audio sample.

VIDEO MODELLING

- The Visual Subnetwork uses VGG16 CNN to extract features from each frame of the video
- The features extracted from each frame are then passed to an LSTM layer
- The LSTM layer summarizes the information across frames
- The output of the LSTM layer is a single feature vector that represents the entire video

- The Combined Network concatenates the outputs of two subnetworks.
- The concatenated output passes through two 256-unit dense layers.
- The final output is linear combination of the subnetwork outputs, with an output layer of 5 units.

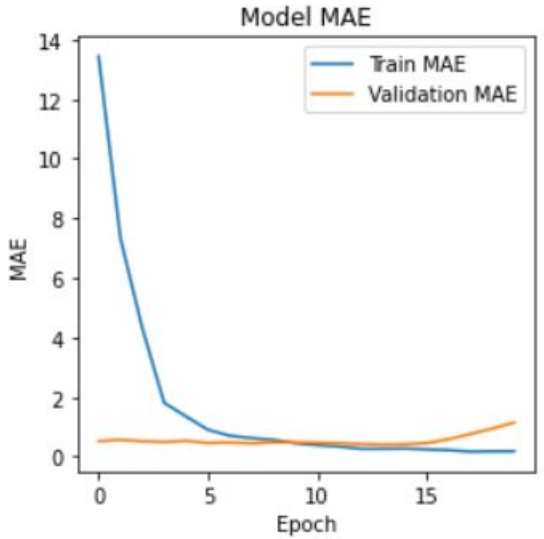
```
Model: "model_2"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_6241 (InputLayer)        [(None, 186)]        0           []

 reshape (Reshape)              (None, 186, 1)       0           ['input_6241[0][0]']

 conv1d (Conv1D)                (None, 184, 32)      128         ['reshape[0][0]']

 batch_normalization (BatchNorm  (None, 184, 32)     128         ['conv1d[0][0]']
 alization)

 max_pooling1d (MaxPooling1D)   (None, 92, 32)       0           ['batch_normalization[0][0]']

 flatten (Flatten)              (None, 2944)         0           ['max_pooling1d[0][0]']

 input_6242 (InputLayer)        [(None, 6, 25088)]   0           []

 dense (Dense)                  (None, 128)          376960      ['flatten[0][0]']

 sequential (Sequential)        (None, 1)            6439233     ['input_6242[0][0]']

 concatenate (Concatenate)      (None, 129)          0           ['dense[0][0]',
                                                                  'sequential[0][0]']

 dense_2 (Dense)                (None, 256)          33280       ['concatenate[0][0]']

 dense_3 (Dense)                (None, 5)            1285        ['dense_2[0][0]']

==================================================================================================
Total params: 6,851,014
Trainable params: 6,850,950
Non-trainable params: 64
```

Baseline

BU-NKU SYSTEM

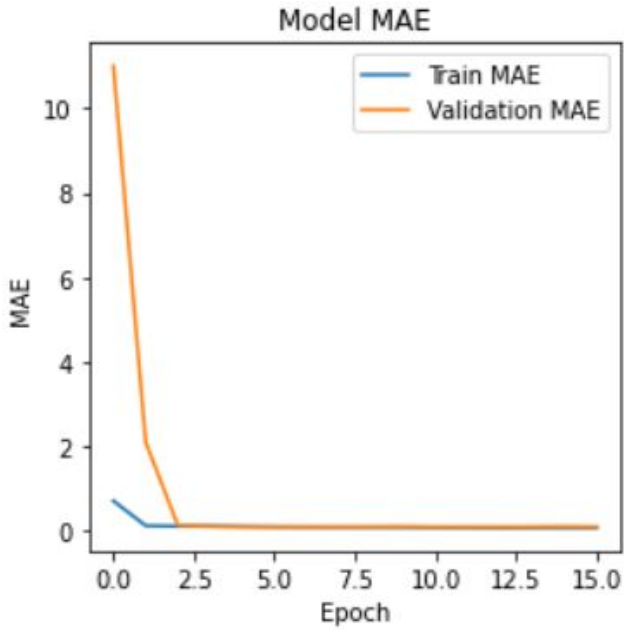
Team	Agreeableness	Conscientiousness	Extraversion	Neuroticism	Openness
BU-NKU	0.913731	0.919769	0.921289	0.914613	0.917014
Baseline	0.91123	0.915228	0.91122	0.910378	0.911123
PML	0.910312	0.913775	0.91551	0.908297	0.910078
ROCHCI	0.903216	0.894914	0.90266	0.901147	0.904709
FDMB	0.891004	0.865975	0.878842	0.863237	0.874761

Results




```
Epoch 7/20
3/3 [==============================] - 111s 44s/step - loss: 0.7060 - mae: 0.7060 - val_loss: 0.4783 - val_mae: 0.4783
Epoch 8/20
3/3 [==============================] - 110s 43s/step - loss: 0.6230 - mae: 0.6230 - val_loss: 0.4482 - val_mae: 0.4482
Epoch 9/20
3/3 [==============================] - 110s 44s/step - loss: 0.5637 - mae: 0.5637 - val_loss: 0.4787 - val_mae: 0.4787
Epoch 10/20
3/3 [==============================] - 109s 43s/step - loss: 0.4526 - mae: 0.4526 - val_loss: 0.4854 - val_mae: 0.4854
Epoch 11/20
3/3 [==============================] - 107s 42s/step - loss: 0.3908 - mae: 0.3908 - val_loss: 0.4696 - val_mae: 0.4696
Epoch 12/20
3/3 [==============================] - 109s 43s/step - loss: 0.3409 - mae: 0.3409 - val_loss: 0.4499 - val_mae: 0.4499
Epoch 13/20
3/3 [==============================] - 120s 48s/step - loss: 0.2663 - mae: 0.2663 - val_loss: 0.4262 - val_mae: 0.4262
Epoch 14/20
3/3 [==============================] - 111s 44s/step - loss: 0.2665 - mae: 0.2665 - val_loss: 0.4075 - val_mae: 0.4075
Epoch 15/20
3/3 [==============================] - 109s 43s/step - loss: 0.2754 - mae: 0.2754 - val_loss: 0.4189 - val_mae: 0.4189
Epoch 16/20
3/3 [==============================] - 109s 43s/step - loss: 0.2456 - mae: 0.2456 - val_loss: 0.4561 - val_mae: 0.4561
Epoch 17/20
3/3 [==============================] - 109s 43s/step - loss: 0.2234 - mae: 0.2234 - val_loss: 0.5876 - val_mae: 0.5876
Epoch 18/20
3/3 [==============================] - 110s 43s/step - loss: 0.1715 - mae: 0.1715 - val_loss: 0.7632 - val_mae: 0.7632
Epoch 19/20
3/3 [==============================] - 109s 43s/step - loss: 0.1818 - mae: 0.1818 - val_loss: 0.9433 - val_mae: 0.9433
Epoch 20/20
3/3 [==============================] - 108s 43s/step - loss: 0.1840 - mae: 0.1840 - val_loss: 1.1473 - val_mae: 1.1473
```

Results




```
Train on 6000 samples, validate on 2000 samples
Epoch 1/20
6000/6000 [==============================] - 65s 11ms/sample - loss: 0.7185 - mae: 0.7185 - val_loss: 11.0003 - val_mae: 11.00034 - mae: 0.
Epoch 2/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.1328 - mae: 0.1328 - val_loss: 2.1111 - val_mae: 2.1111 loss: 0.1330 - mae: 0.13 -
ETA: 1s - loss: 0.1329 - mae: 0.13 - ETA: 1s - loss: 0.1329 - mae:
Epoch 3/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.1198 - mae: 0.1198 - val_loss: 0.1377 - val_mae: 0.1377
Epoch 4/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.1225 - mae: 0.1225 - val_loss: 0.1123 - val_mae: 0.1123
Epoch 5/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.1126 - mae: 0.1126 - val_loss: 0.1051 - val_mae: 0.1051
Epoch 6/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.1040 - mae: 0.1040 - val_loss: 0.1051 - val_mae: 0.10516 - mae - ETA: 2s - loss:
0.1043 -  - ETA: 0s - loss: 0.1040 - mae: 0.10
Epoch 7/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.0992 - mae: 0.0992 - val_loss: 0.1063 - val_mae: 0.1063 - ETA: 4s - loss: 0.0990 -
ETA: 1s - loss: 0.0992 - mae - ETA: 0s - loss: 0.0992 - mae: 0.099
Epoch 8/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.0977 - mae: 0.0977 - val_loss: 0.1077 - val_mae: 0.1077: 0.0975 - mae: 0.0 - ETA:
5s
Epoch 9/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.0970 - mae: 0.0970 - val_loss: 0.1086 - val_mae: 0.1086
Epoch 10/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.0946 - mae: 0.0946 - val_loss: 0.1147 - val_mae: 0.1147 loss: 0.0946 - mae: 0.09
Epoch 11/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.0907 - mae: 0.0907 - val_loss: 0.1069 - val_mae: 0.1069
Epoch 12/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.0892 - mae: 0.0892 - val_loss: 0.1057 - val_mae: 0.1057
Epoch 13/20
6000/6000 [==============================] - 52s 9ms/sample - loss: 0.0858 - mae: 0.0858 - val_loss: 0.1066 - val_mae: 0.1066
Epoch 14/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.0869 - mae: 0.0869 - val_loss: 0.1078 - val_mae: 0.1078 lo - ETA: 24s  - ETA: 17s
- loss: 0.08 - ETA: 15s - loss: 0.0873 - mae: 0.08 - ETA: 15s  - ETA: 6s - loss: 0.08 - ETA: 3s - loss: 0.0873 - mae: 0 - ETA: 2s - loss: 0.0872
Epoch 15/20
6000/6000 [==============================] - 51s 8ms/sample - loss: 0.0859 - mae: 0.0859 - val_loss: 0.1140 - val_mae: 0.1140
Epoch 16/20
6000/6000 [==============================] - 52s 9ms/sample - loss: 0.0855 - mae: 0.0855 - val_loss: 0.1075 - val_mae: 0.1075
```

Summary

We proposed a method to access personality traits using audio-visual modalities for candidate screening.

Abhishek preprocessed the audio input using pre-trained YAMNet model and created an audio subnetwork. Piyush preprocessed video data using a pre-trained VGG16 model and created a visual subnetwork. Both members contributed equally to the combined network, presentation well as report.

THANK YOU