Deep Temporal Multi-scale Net for Apparent Personality Analysis

Jingxi Xu

jx2324
Computer Science Department
Columbia University

Abstract. According to psychologists, each person's personality can be described by five big traits called the OCEAN model [1], and the impression of others can be formed within seconds. In this paper, we adopt LSTM and 3D-convolution to develop a temporal multi-scale deep neural network to automatically predict five scores representing each dimension of personality by letting the machine view a short video of the subject. Our main contributions are as follows: 1) We re-implement the bi-modal end-to-end deep neural network using temporally ordered audio and visual features proposed by [2] in TensorFlow. 2) Our re-implementation achieves $4 \sim 6\times$ speedup in the training process by adding multiprocessing. 3) We propose a new Deep Temporal Multi-scale Net which achieves the state-of-the-art results. In the project, all three team members contribute equally and I mainly focus on model design, feature preprocessing and extraction, and validation.

GitHub repo link: `https://github.com/JingxiXu/traits_analysis_tf_submission`

Keywords: LSTM, 3D-convolution, video processing, multi-scale model, deep learning, CNN, temporal model

1 Introduction

The ability of machines to interpret and analyze images and videos have made great strides in recent years as a result of the development of computer vision and deep learning techniques. While the state-of-the-art deep learning models have already made superhuman performance in image classification problems, the analysis of videos to generate insightful interpretation (clustering, classification, description, etc.) remains to be a harder problem and is attracting huge attention from computer vision and deep learning research communities.

In this project, we aim to leverage the cutting-edge video and audio processing techniques and deep learning models to automatically analyze videos of persons speaking in front of the camera to reveal their personalities, with regards to the *big five personality traits*.

Many contemporary personality psychologists believe that there are five basic dimensions of personality, often referred to as the big five personality traits or OCEAN model [1]. The five personalities are listed as follows:

- Openess: appreciation for art, emotion, adventure, unusual ideas, curiosity, and variety of experience.
- Conscientiousness: a tendency to be organized and dependable, show self-discipline, act dutifully, aim for achievement, and prefer planned rather than spontaneous behavior.
- Extraversion: energy, positive emotions, surgency, assertiveness, sociability and the tendency to seek stimulation in the company of others, and talkativeness.
- Agreeableness: a tendency to be compassionate and cooperative rather than suspicious and antagonistic towards others.
- Neuroticism: neuroticism identifies certain people who are more prone to psychological stress.

Our goal is to output five scores in the range of $[0, 1]$ (closed interval), one for each personality of the person speaking in the video, as shown in . We want a system to automatically do the job with deep learning techniques. This project can easily be adapted to other various applications. We can apply the learned scores from interview videos to help HR choose proper candidates, or even develop a learning system to make the decision from the five scores directly with the labels given by HR. In addition, from public security's perspective and with the help of psychologists, we might be able to apply similar technologies to tell whether a crime suspect is lying or not based on his facial expressions.

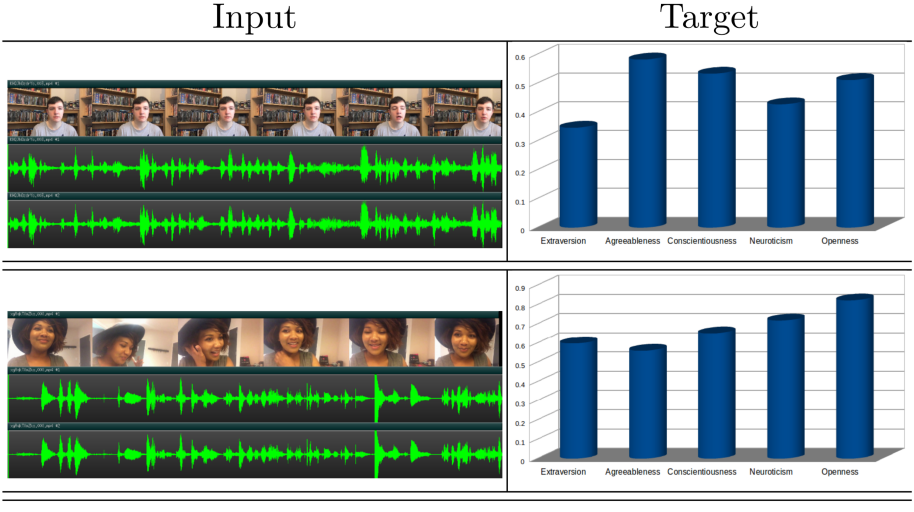


Fig. 1: Example of Input and Target. Input is the raw video with sound containing a person speaking, output will be the predicted personality-traits values. Imgae credit: [2].

2 Related Work

The most related work in literature would be the *ECCV 2016 workshop on automatic personality analysis and first impressions challenge* [3]. In this challenge, various teams have developed different deep learning models to predict five personality scores for a 15-second video of a person speaking.

[4] proposes a Deep Bimodal Regression (DBR) framework. For the visual modality, they modify the traditional convolutional neural networks for exploiting important visual clues. For the audio modality, they extract audio representations and build a linear regressor. To combine these complementary information, they ensemble these predicted regression scores by both early fusion and late fusion.

[2] proposes two end-to-end trained deep learning models that use audio features and face images for recognizing first impressions. The first model uses Volumetric (3D) convolution based deep neural network for determining personality traits. The second model formulates an LSTM based deep neural network for learning temporal patterns in the audio and visual features. The LSTM model slightly outperforms the 3D convolution based model.

3 Data Set

The data set can be obtained from the ChaLearn Looking at People website (`http://chalearnlap.cvc.uab.es/dataset/20/description/`).

There is a newly collected data set of 10000 15-second videos of people speaking to the camera collected from YouTube by Microsoft, annotated with personality trait scores by *Amazon Mechanical Turk* (AMT) workers. The groundtruth for each video is a vector of 5 real numbers in the range of $[0, 1]$, each representing a personality trait. Thus, this project can also be viewed as a clustering problem of videos.

People appearing in videos of this data set are of different genders, ages, nationalities, and ethnicities, and the content they are talking about can be quite random, from introducing themselves to advertising products or requesting users to subscribe to their YouTube channels.

4 Feature Extraction and Preprocessing

Compared to the features extracted and used in [2], we feed two more classes of features into our deep neural network - facial landmark and action units. To get all features below for a single video takes around 65s; therefore, total preprocessing and feature extraction time is about 1 week on a single machine.

We first use `ffmpeg` to extract an audio file (.wav) from each video. We will then divide both the video part (.mp4) and the audio part (.wav) into 6 partitions, and for each partition we extract the following features. The temporal sequence between 6 partitions will be later taken advantage of by LSTM.

Audio Features We use python library `pyAudioAnalysis` [5] to extract hand-crafted audio features of 68 dimensions (mean and standard deviation of 34 attributes described in figure 2).

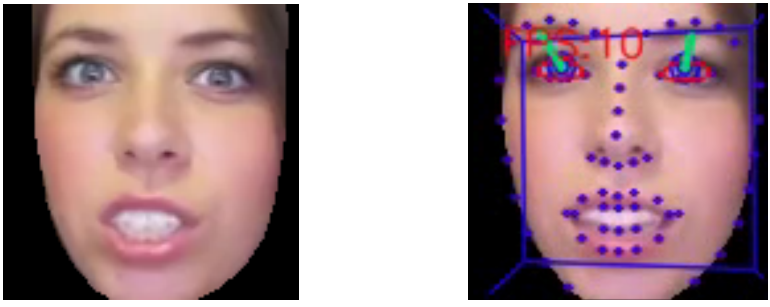
Feature ID	Feature Name	Description
1	Zero Crossing Rate	The rate of sign-changes of the signal during the duration of a particular frame.
2	Energy	The sum of squares of the signal values, normalized by the respective frame length.
3	Entropy of Energy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.
4	Spectral Centroid	The center of gravity of the spectrum.
5	Spectral Spread	The second central moment of the spectrum.
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames.
7	Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale.
22-33	Chroma Vector	A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
34	Chroma Deviation	The standard deviation of the 12 chroma coefficients.

Fig. 2: 34 hand-crafted audio feature attributes. Figure credit: `https://github.com/tyiannak/pyAudioAnalysis/wiki/3.-Feature-Extraction`

3D Aligned Cropped Face This feature along with facial landmark and action unit code listed below can all be obtained by using another python library `OpenFace` [5]. We further divide each partition into 8 intervals and for each interval, we randomly choose 1 aligned and cropped face returned by `OpenFace`, example image shown in figure 3a. Thus, for each partition, we have 8 frames in

temporal order which can be later used in 3D-convolution. Then for each video, we choose 6 partitions $\times$ 8 frames/partition = 48 frames in total.

Facial Landmark Facial landmark [6] is an important class of feature in describing facial expression, whose movement can then reflect personalities. For each frame, we use `OpenFace` to extract 68 2D facial landmarks directly from the cropped face images generated by the step above, as shown by figure 3b.



(a) Aligned cropped face image (b) Face image with facial landmarks

Fig. 3: Cropped face and facial landmark visualization

Facial Action Unit Facial Action Coding System (FACS) [7] is a system to taxonomize human facial movements by their appearance on the face. Facial Action Units (AUs) is another important way to describe human facial expression. `OpenFace` is able to recognize a subset of AUs [8], specifically: 1, 2, 4, 5, 6, 7, 9, 10, 12, 14, 15, 17, 20, 23, 25, 26, 28, and 45. For each of the selected 48 cropped face frames, we will also obtain their AUs.

5 Models

This section is originally written by Qiangeng Xu.

Personality analysis has been a very challenging problem even for psychologists. To make the machine be able to automatically predict similar results as human does, it has to be trained to learn from various kinds of features and more importantly, take advantage of temporal patterns (head movement, facial expression, etc.) of the video it sees. The characteristic that distinguishes videos from images is the temporal relationship between video frames whereas image classification problem does not have. Moreover, we should also consider the intrinsic correlation among features across multiple modalities. In this section, we will analyze two existing models and propose a third novel model.

5.1 Segregated Convolution Model

[2] and [9] has proposed models that processes audio and visual branch separately and does not fuse the results until last layer. These two models will first learn a low dimension semantic vector for each branch separately and then concatenate these two vectors together and then feed the concatenated vector into a MLP. We consider them to be in the same class - segregated convolution model, because they do not use temporal pattern/sequence across different modalities until the last MLP, even though they adopt slightly different methods (3D convolution and 2D convolution with residual connection respectively). The network of [9] is shown in figure 4.

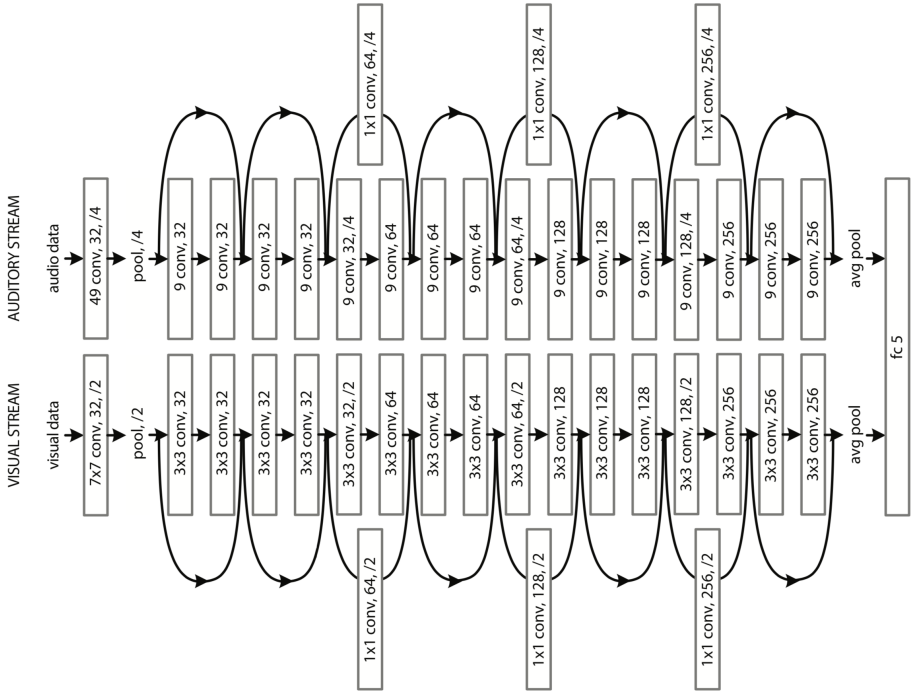


Fig. 4: 2D convolution residual model. Image credit: [9]

5.2 LSTM Baseline Model

To improve the performance of segregated convolution models, [2] adopts LSTM to better capture the temporal patterns both inside modality and across different modalities. This class of models usually first splits the video into multiple non-overlapping partitions. In each partition, the model learns a low-dimension

feature vector through CNN and after that, keeping the temporal sequence unchanged, feed the vector of each partition into an LSTM model. Each output of the LSTM then goes through the same shared weight MLP to predict the five scores for each video. In the end, the model performs average/max pooling to fuse 6 5D scores into 1 5D scores. There are also some weaknesses for this class of models: 1) it only takes cropped faces as visual information and 2) it ignores the temporal visual information inside each partition as it only samples 1 frame randomly for each partition.

This is the model that we re-implement in TensorFlow [10] and use as a baseline to evaluate our own model's performance. The model structure is shown in figure 5.

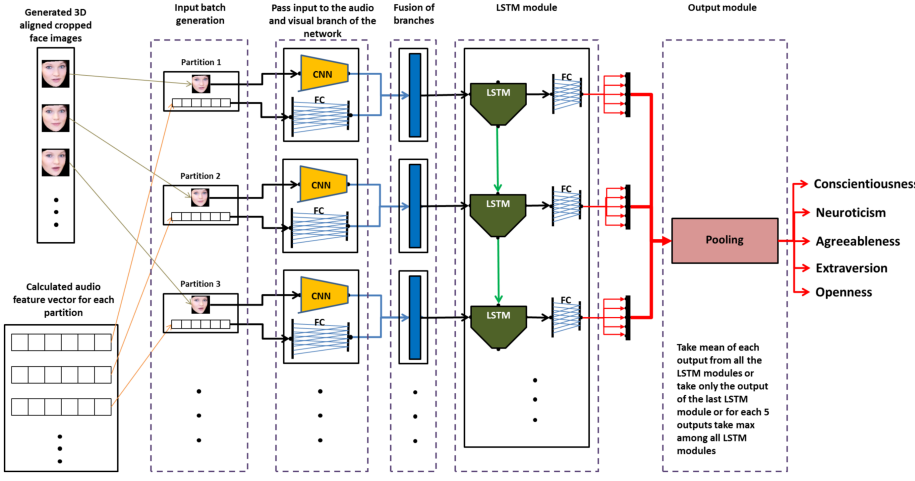


Fig. 5: LSTM baseline model. Image credit: [2]

5.3 Deep Temporal Multi-scale Net

Using 3D aligned and cropped face frames can help us remove irrelevant information for personality analysis but only recognizing the facial expression from RGB face frames is probably too hard, which is one of the motivations of us adding more related and useful facial features. In addition, the current studies such as [11] and [12] have used very deep network or even inception net [13] for the expression recognition task, because extracting personality information from raw cropped face images will cast too much burden on the network. Therefore, to handle this challenge, we also propose more complex network structures by adding facial landmarks with the RGB channels to form a fourth dimension. Moreover, we incorporate the facial action units [14] and process them as a third branch/modality.

Inside each partition, the LSTM baseline model only takes one frame by random to represent the whole partition, which is a not ideal. The most ideal case would be to use entire frames in a partition to bring in all information. However, that way may also introduce redundant information and is also not feasible in terms of training time. In our model, we balance these two cases by taking multiple frames inside a partition in sequence. This is done by further divide the partition into multiple non-overlapping intervals and randomly pick a frame from each interval. The frames in sequence inside a single partition will be used by 3D-convolution to extract low-dimension feature vectors. These low-dimension vectors from each partition will then be concatenated and feed to LSTM to learn temporal information among sequences.

In addition, the video frame rate could vary from the video source and different traits could depend more on certain temporal scale than the others. Here we adapt the multi-scale learning methodology from many vision tasks such as object detection [15, 16], frame prediction [17] and action recognition [18]. We apply their spacial multi-scale training method on the temporal dimension by fanning out three 3D convolution visual branches with different temporal strides. In the end, we concatenate feature vectors from all modalities and different scales and put them into the LSTM unit. Our model is illustrated in figure 6, and its network structure details are shown in figure 7.

6 Experiments and Implementation

This section is originally written by Xuefeng Hu.

6.1 Network Structure

As mentioned before, we re-implement the bi-modal LSTM network and the new proposed Temporal Multi-scale model in TensorFlow. The network structures are shown in figure 5 and figure 6 respectively.

6.2 Data Split

We use the ChaLearn dataset [3] for training and evaluation as introduced in section 3. However, because this data set is from the 2016 ECCV workshop competition, the test set labels are not released and therefore not available. In our experiment, we take the original 2000 video validation set as the test set and further split the original 6000-video training set into a 5000-video training set and a 1000-video validation set.

6.3 Training Details

The training details of our two experiments on bi-modal LSTM and Deep Temporal Multi-scale Net are as follows:

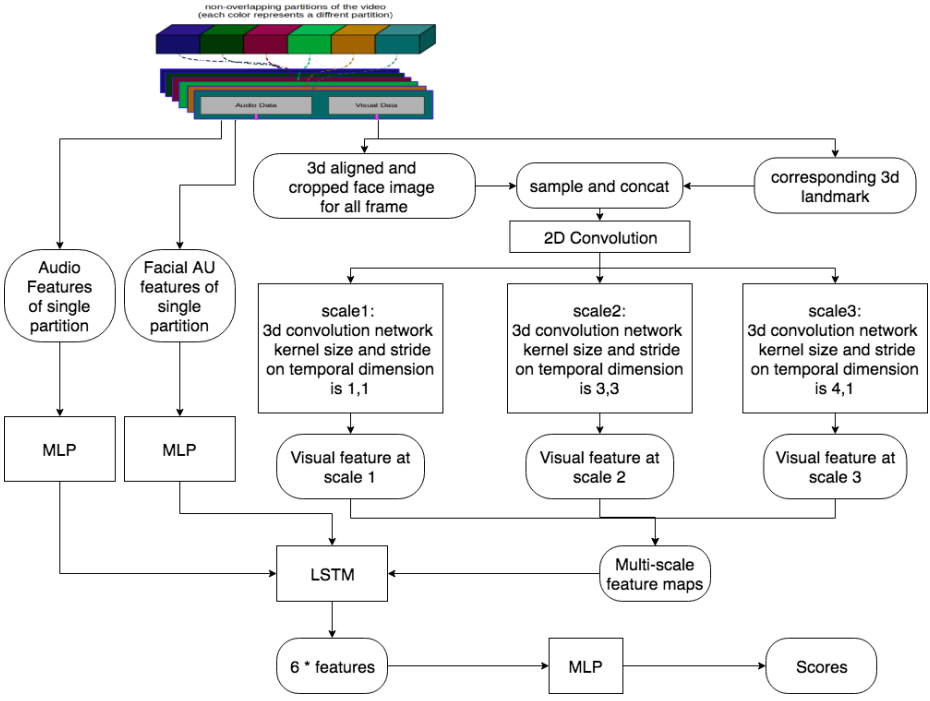


Fig. 6: Temporal Multi-scale model. Three branches including the audio stream branch on the left, facial AU code branch in the middle and the visual stream branch on the right. The visual stream branch has three sub branches of different temporal scales.

Bi-modal LSTM Net We have re-implemented the Bi-modal LSTM Net in TensorFlow which is originally implemented in Torch. We follow most of the original training settings as in [2]:

- Loss Function: mean squared error between the ground truth and the prediction:

$$\frac{1}{n}\sum_{i=0}^{n}\sum_{j=0}^{4}\left(Pred(i,j)-GT(i,j)\right)^2$$

  where $Pred(i,j)$ is the prediction produced by bi-modal LSTM net, $GT(i,j)$ is the ground truth of the j-th trait of sample i and n is the batch size.
- Optimization: As proposed in [2], we use Stochastic Gradient Descent optimizer to do the back-propagation, with a 0.9 momentum, an initial learning rate at 0.05 and exponentially decay by 0.96 every 2500 iterations.
- 128 sample per batch with 24000 iterations ($\sim$ 615 epochs).
- Other Detail: To better monitor the training process, we show training loss every 100 iterations, validation loss every 500 iterations and save model every 1000 iterations.

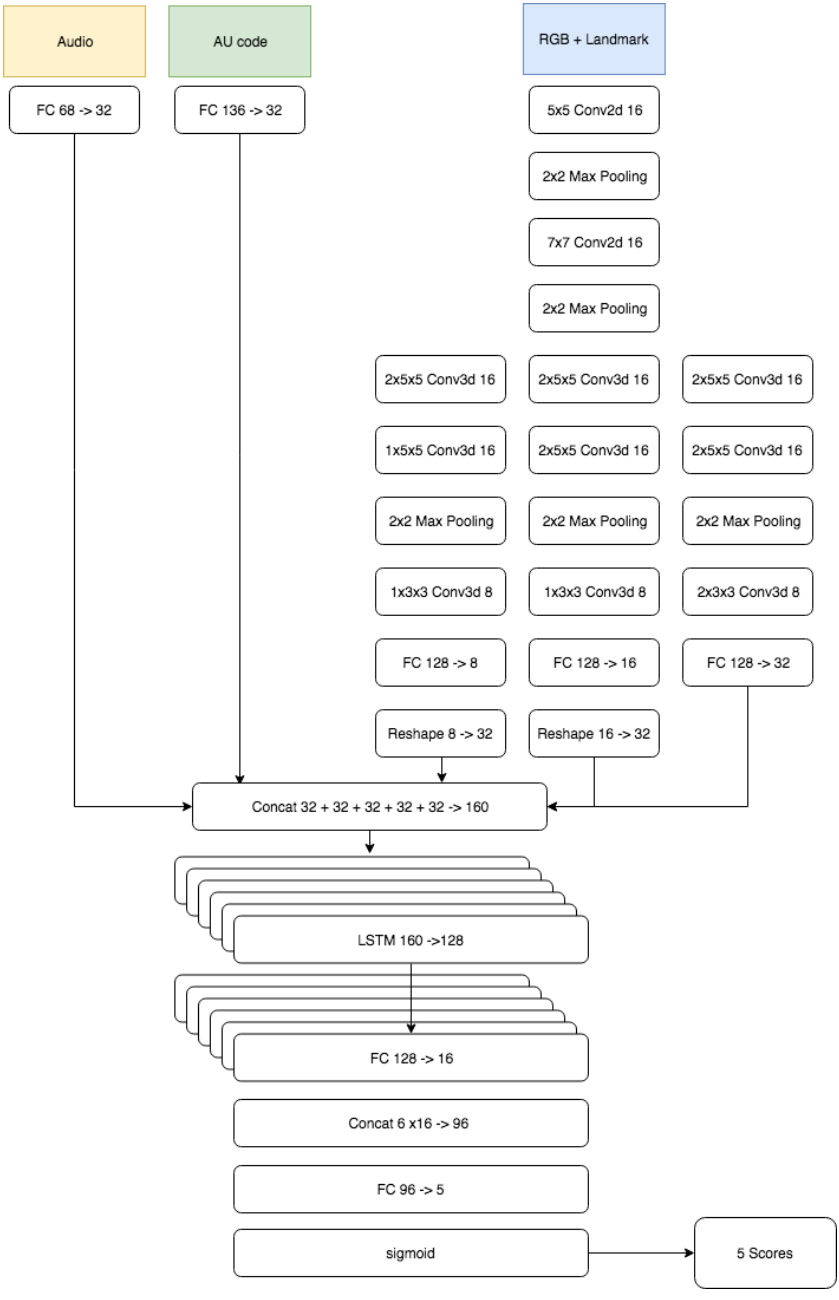


Fig. 7: Temporal Multi-scale structure

Temporal Multi-scale Net The training details of our novel Temporal Multi-scale Net are as follows:

- Loss Function: mean squared error between the ground truth and the prediction:
$$\frac{1}{n}\sum_{i=0}^{n}\sum_{j=0}^{4}\left(Pred(i,j)-GT(i,j)\right)^2$$
where $Pred(i,j)$ is the prediction produced by bi-modal LSTM net, $GT(i,j)$ is the ground truth of the j-th trait of sample i and n is the batch size.
- Optimization: as proposed in [2], we still use Stochastic Gradient Descent optimizer to do the back-propagation, with a 0.9 momentum, an initial learning rate of 0.05 and exponentially decay by 0.96 every 2500 iterations.
- Due to the the increase of model size and the memory restriction, we reduced the batch size to 64 with 50000 iterations ($\sim$ 641 epochs).
- Other details: to better monitor the training process, we show training loss every 100 iterations, validation loss every 500 iterations and save model every 1000 iterations.

6.4 Evaluation Metric

For evaluation, we adopt **Average Similarity** metric as defined in [3]:

$$\text{Average Similarity} = \frac{1}{N}\sum_{i=0}^{N}\sum_{j=0}^{4}1-\left|Pred(i,j)-GT(i,j)\right|$$

where $Pred(i,j)$ is the prediction produced by bi-modal LSTM net, $GT(i,j)$ is the ground truth of the j-th trait of sample i and N is the size of test samples.

7 Result and Discussion

This section is originally written by Xuefeng Hu and Qiangeng Xu.

We present the results of our re-implemented model and new model in this section. We select the best models using 1000-video validation set and report their performance on the 2000-video test set (which used to be the 2000-video validation set in 2016 ECCV workshop). The performance of other models are their reported performance on the same set as mentioned in [2, 3, 9, 19].

It worth noting that our performance is worse than [2] even though we re-implement the same net-work structure. The reasons could be: 1) the difference in numerical methods adopted by Tensorflow [10] and Torch [20]; 2) smaller training set we have because we split the original training set into training and validation; 3) since our test set is their validation set, it makes sense because the model they report is the best on this set.

On the other hand, our Temporal Multi-scale model outperforms all the comparable models due to the two more modalities it incorporates and the more comprehensive learning strategy on multiple time scales. The only two personality

Model	Average	Open	Consci	Extrav	Agree	Neuro
Audiovisual Residual [9]	0.913	0.913	0.913	0.915	0.916	0.910
Bi-model LSTM [2]	0.912	0.912	0.913	0.910	0.916	0.910
Bi-model LSTM Re-implementation	0.909	0.908	0.909	0.910	0.912	0.905
Temporal Multi-scale	0.915	0.920	0.918	0.908	0.906	0.921

Table 1: Experiment results on test set

traits on which the model achieve less satisfactory performance are Extraversion and Agreeableness. Since those two traits are relatively easier to be determined by a single static frame, we suppose the information from longer time scales affect the accuracy.

8 Conclusion

In this project, we develop models for automatically predicting big five personality traits from short videos. We re-implement the bi-modal LSTM model proposed in [2] in TensorFlow. We achieve a speedup of around $4 \sim 6\times$ due to our multiprocessing. We then analyze its weaknesses and then propose a novel model called Deep Temporal Multi-scale Net which incorporates extra features - facial landmarks and action units. Our model is also more complex in network architecture by adding one more modality using action units and using 3 scales of 3D-convolution with more frames in each partition.

References

1. contributors, W.: Big five personality traits — wikipedia, the free encyclopedia (2018) [Online; accessed 22-March-2018].
2. Subramaniam, A., Patel, V., Mishra, A., Balasubramanian, P., Mittal, A.: Bi-modal first impressions recognition using temporally ordered deep audio and stochastic visual features. In: European Conference on Computer Vision, Springer (2016) 337–348
3. Ponce-López, V., Chen, B., Oliu, M., Corneanu, C., Clapés, A., Guyon, I., Baró, X., Escalante, H.J., Escalera, S.: Chalearn lap 2016: First round challenge on first impressions-dataset and results. In: European Conference on Computer Vision, Springer (2016) 400–418
4. Wei, X.S., Zhang, C.L., Zhang, H., Wu, J.: Deep bimodal regression of apparent personality traits from short video sequences. IEEE Transactions on Affective Computing (2017)
5. Giannakopoulos, T.: pyaudioanalysis: An open-source python library for audio signal analysis. PloS one **10**(12) (2015)
6. Baltrusaitis, T., Robinson, P., Morency, L.P.: Constrained local neural fields for robust facial landmark detection in the wild. In: Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, IEEE (2013) 354–361
7. Ekman, P.: Facial action coding system (facs). A human face (2002)

8. Wood, E., Baltrusaitis, T., Zhang, X., Sugano, Y., Robinson, P., Bulling, A.: Rendering of eyes for eye-shape registration and gaze estimation. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 3756–3764
9. Güçlütürk, Y., Güçlü, U., van Gerven, M.A., van Lier, R.: Deep impression: Audiovisual deep residual networks for multimodal apparent personality trait recognition. In: European Conference on Computer Vision, Springer (2016) 349–358
10. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
11. Xu, Q., Qin, Z., Wan, T.: Generative cooperative net for image generation and data augmentation. arXiv preprint arXiv:1705.02887 (2017)
12. Mollahosseini, A., Chan, D., Mahoor, M.H.: Going deeper in facial expression recognition using deep neural networks. In: Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, IEEE (2016) 1–10
13. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., et al.: Going deeper with convolutions, Cvpr (2015)
14. Friesen, E., Ekman, P.: Facial action coding system: a technique for the measurement of facial movement. Palo Alto (1978)
15. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. (2015) 91–99
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 779–788
17. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440 (2015)
18. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: Cdc: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2017) 1417–1426
19. Zhang, C.L., Zhang, H., Wei, X.S., Wu, J.: Deep bimodal regression for apparent personality analysis. In: European Conference on Computer Vision, Springer (2016) 311–324
20. Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: A matlab-like environment for machine learning. In: BigLearn, NIPS Workshop. (2011)