OMG – Empathy Challenge Submission – Alpha / City

Francesco Barbieri[1] Eric Guizzo[2] Federico Lucchesi[1] Giovanni Maffei[1] Fermín Moscoso del Prado Martín[1] Tillman Weyde[2]

***Abstract*— We describe our system for empathic emotion recognition. It is based on deep learning on multiple modalities in a late fusion architecture. We describe the modules of our system and discuss the evaluation results. Our code is also available for the research community[1]**

I. Introduction

This report describes our joint submission (Alpha and City) to the OMG – Empathy Challenge. Our goal in developing this system is to provide an early prototype for a more fully developed future multimodal emotion recognition systems that we aim to develop.

We propose a system that separately processes multiple data streams (modalities) which are integrated at a late stage, so-called late fusion. The motivations for this modular approach are ease of development, enabling different parts of the team to separately optimize the processing of different modalities, and future extensibility, facilitating the integration of additional data streams in future iterations of our system.

II. Methods

We integrate three different modalities (further broken down into five data streams) in the prediction of the valence ratings of the videos. These modalities are:

- *Image* information directly extracted from the videos.
- *Audio* information directly extracted from the videos.
- *Language* information obtained by automatic transcription of the audio data.

Figure 1 depicts a general schema of the model. Note that, in our model, the audio and language modalities each give rise to single data stream to be processed, whereas the image modality is further broken down into three input data streams: one corresponding to the full body of the subjects, another one focusing on the face of the subject, and a final one that further synthesizes specific landmarks extracted from the subjects' faces. Note that we employ specific architectures across the five resulting systems, which are specifically optimized, of each modality. Nevertheless, the loss function and training and validation sets where held constant across the five sub-systems.

In all five systems, instead of using Story #1 as the validation set and all other ones as the training set (as was suggested by the instructions), we chose instead to use Story #2 as our validation set and the remaining ones as our training set. The reason for this is that we found the statistical properties of Story #1 to be rather unrepresentative of the stories in the original training set. In particular, the main frequency at which the ratings oscillated between positive and negative was found to differ significantly from the others.

All models where trained to minimize $1 - \text{CCC}$ as loss function, where CCC is the *Concordance Correlation Coefficient*. For a sequence of valence predictions x and and a sequence ground truth valences y, the CCC is defined as

$$\text{CCC} = \frac{2\,\rho\,\sigma_x\,\sigma_y}{\sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2},$$

where σ_i, μ_i refer to the means and standard deviations of the subscripted sequences, and ρ is the *Pearson's correlation coefficient* between x and y. The CCC measure is a correlation coefficient that additionally penalizes divergences in either mean of variance between the two data sequences. As we will see below, this motivates some additional post-processing of the output data.

We experimented with several architectures and hyperparameter values for each module. For brevity, we present only the configurations that provided the best performance.

A. Audio Model

This model is based on the audio information extracted from the video files. Every audio file is pre-processed in 4 consecutive stages: pre-emphasis, segmentation, Fourier transform, and normalization. In order to discard low-frequency-noise, we first pass the signal through an 8th order Butterworth high-pass filter with 100 Hz cut-off frequency. Then we apply an emphasis filter based on the following equation

$$y(t) = \frac{2x(t) - x(t-1)}{3} \tag{1}$$

where $x(t)$ is an audio sample and $x(t-1)$ is the preceding sample. This acts as a gentle first order high pass filter that emphasizes the spectral range of speech, with the upper limit of 8kHz defined by the sampling frequency of 16kHz. Every file is segmented into 8-second slices with 20% overlap. Consequently, the STFT is computed for every slice using 16ms sliding windows with 10ms overlap. This results in exactly 4 STFT frames for each valence measure (since 1 valence every 40 milliseconds is provided). After this process, we discard the phase information and compute the power-law compression by exponentiating the spectrum magnitudes the power of 2/3 to approximate human perception [4]. This technique is borrowed from the method of calculation of

[1] Telefónica Innovation Alpha, Barcelona, Spain
[2] Department of Computer Science, City, University of London, UK
[1] https://github.com/omg-challenge-alpha/omg_challenge2018_submission_code

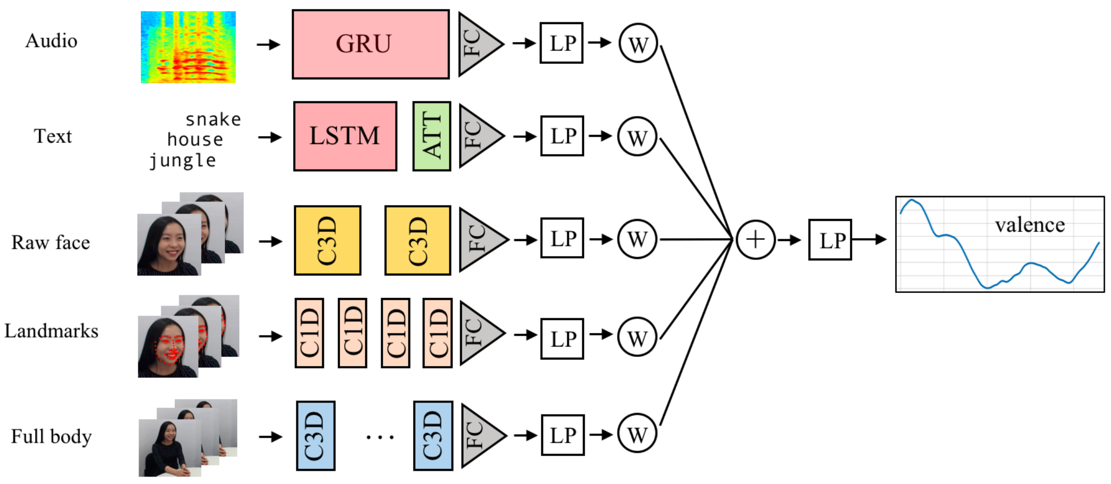


Fig. 1: Schematic view of the whole system.

the Perceptual Linear Prediction Coefficients [5]. Finally, we normalize the spectra to zero mean and unit standard deviation.

This neural network has a sequence-to-sequence design based on a Recurrent Neural Network. The model's task is to predict time sequences of 200 valence samples for 8 seconds of input. First, we apply a layer of bi-directional Gated Recurrent Units (GRU) with 250 neurons for the forward and 250 neurons for the backward representation of the input data. Batch normalization and Dropout at 30% are applied to the GRUs output to reduce overfitting. Then the signal is propagated into a fully-connected layer with 200 neurons using linear activation. We trained the model with a batch size of 50 samples, using the ADAM optimizer [6] and applying Early Stopping. The mean CCC obtained by this model in the validation set is 0.32.

B. Language-based Model

The model for processing the linguistic input stream consists of a recurrent network that processes the dialogue transcript, which was obtained with the Amazon Transcribe service [2]. Beyond transcription, no further textual preprocessing is applied. The transcription results in a sequence of words, together with time-stamps indicating when each word starts and ends. Each word spans several frames, hence more than one valence value. We address this by associating each word to the average valence score of all the valence scores within its span. Each word is represented as a vector of 11 dimensions, consisting of the features extracted from two emotional lexicons [10], [11].

An LSTM network [9] is used to predict a valence score after each word. The time window used (what the LSTM *sees* at each step) is a window of 100 words. The hidden vectors of the LSTM are merged with a weighted average implemented with the following attention module as in [12]:

$$z_i = w_a h_i + b_a$$
$$\alpha_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$$
$$s = \sum_{j=1}^{N} \alpha_j h_j$$

where $h_i \in \mathbb{R}^d$ is the hidden representation of the LSTM corresponding to the i^{th} word, with N the total number of words in the window. The weight vector $w_a \in \mathbb{R}^d$ and bias term $b_a \in \mathbb{R}$ map this hidden representation to a value that reflects the importance of this state for final valence. The values $z_1, ..., z_n$ are then normalized using a softmax function, yielding the attention weights α_i. The word sequence representation s (at each time frame) is defined as a weighted average of the vectors h_i.

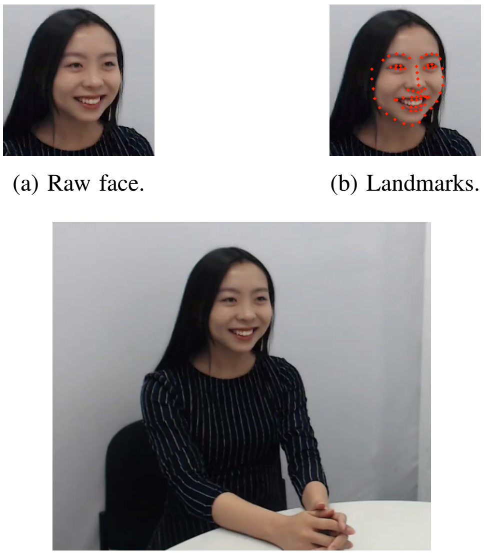
For each input also the listener subject information is given in the input to the network, since the story transcript can be very similar across videos, but the labeling can be highly different depending on the subject. The subject feature is implemented as a trainable vector of size two (one vector for each of the 10 subjects).

Finally, vector s, computed with the attention module, is concatenated with the subject vector, and a final affine tranformation is used to shrink the concatenated vector to one dimension (the final valence prediction of the model). This module achieves an average CCC of 0.32 across the validation set.

C. Vision Model

The visual model includes features extracted from the subjects who were listening to the story. Visual features are extracted from the face only, in order to capture facial expressions, but also from the whole body, in order to model the subject body reactions to the story (i.e. posture and gestures). This leads to three modules for vision as follows.

[2] https://aws.amazon.com/transcribe/



(a) Raw face. (b) Landmarks.

(c) Full body.

Fig. 2: Vision data examples.

1) Raw Face: This module is dedicated to the prediction of valence from the subject's facial expression. It takes into account both the temporal evolution of the data and a subject feature vector. The cropped face images are obtained using the pre-processing script provided on the competition repository[3], using a temporal resolution of 10 frames for the face detection algorithm (Figure 2a). The obtained crops are subsequently turned to grey-scale, downsampled to a resolution of 48x48 pixels and normalized to have zero mean and unit standard deviation. The images are further organized with a 10 frame sliding window to obtain samples of shape 10x48x48. Each sample is matched with the valence label corresponding to the 10th frame. To predict valence from sequences of faces we use a neural network architecture composed of one 3 dimensional convolutional block of output shape 32 (two 3D convolutional layers with kernel size 3x3x3 and ReLU activation followed by a max-pooling layer and batch normalization) followed by a second 3D convolutional block of output shape 64. In addition we provide a subject feature vector encoding information about the subject and implemented as a trainable vector of size three (one vector for each of the 10 subjects), as in the text model. The concatenated layer is finally mapped to a fully connected ReLU layer of size 128, followed by a fully connected ReLU layer of size 32 and a single unit with linear activation. The network is trained with a batch size of 64 samples, using the ADAM optimizer. This module achieves a mean CCC of 0.14 on the validation set.

2) Face Landmarks: This module is dedicated to the prediction of valence from features extracted from image data, taking into account their temporal evolution. The data pre-processing consists of a facial landmark detection, performed frame-by-frame using the *dlib* library [7]. We detect 68 landmarks points per frame on the subjects face, as shown in Figure 2b. Each point is defined by its (x, y) coordinates. This characterizes each video with 136 time series describing the temporal evolution of the landmarks points. Each time series is subsequently processed to have zero mean and unit standard deviation. The time series are further organized into 25 frames sliding sequences so to obtain samples of shape 25x136. Each sample is matched with the valence label corresponding to the 25th frame. For the first 25 frames, we perform constant-value padding. To predict valence, we use the 25-sample-long time series as inputs for a 1D Convolutional Neural Network architecture, composed of a first convolutional layer with 100 kernels, followed by a batch normalization layer and a convolutional block of three convolutional layers with 100, 160, and 160 kernels respectively. A 1D global average pooling is then applied, followed by a Fully Connected layer of size 32 that is mapped to a single unit with linear activation. All the layers - except for the output layer - have ReLU activation functions, and a kernel size of 4. The network is trained with a batch size of 512 samples, using the ADAM optimizer. This module achieves a 0.12 CCC on the validation set.

3) Full-Body: This module is dedicated to predicting valence ratings out of full body subject images and it takes into account the sequential nature of the dataset. The full-body crop images are obtained using a pre-processing script that applies a cropping box manually selected to capture the position of the subject. An example is shown in Figure 2c. The obtained crops are then turned to grey-scale, downsampled to a resolution of 128x128 pixels and normalized by subtracting the mean and dividing by the standard deviation. Further the images are organized into 16 frame sliding windows so to obtain samples of shape 16x128x128 and each sample matched with the label corresponding to the 16th frame. Note that in the final setup the video sequences and respective labels are downsampled by a factor of 5 in order to expand the temporal window to approx. 3 seconds while maintaining a fixed sample shape. To predict valence ratings from full body video snippets we use a neural network architecture based on a version of the ResNet 16 [8] architecture adapted to 3-dimensional data [4]. This architecture was further modified for regression by replacing the last 3 fully connected ReLU layers with a 512 unit layer connected to a 32 unit layer followed by a single unit output. The network is trained with a batch size of 128 samples, using the ADAM optimizer. This module achieves a mean CCC of 0.31 on the validation set.

D. Postprocessing & Multimodal Integration

The predictions from each module are post-processed using a first order Butterworth low-pass filter and different cutoff frequencies adjusted for each individual module, ranging from 0.004 to 0.01 Hz. The filtered predictions $\hat{x}$ are then re-centered and re-scaled so that they match the training set in terms of per-subject means and standard deviations. This

[3] https://github.com/knowledgetechnologyuhh/OMGEmpathyChallenge

[4] https://github.com/JihongJu/keras-resnet3d

is a relatively ad-hoc procedure designed to optimize CCCs, under the assumption that the mean and standard deviations of the ground truth valences in the training set provide an approximation of those in the testing and validation sets.

Our best final predictions were obtained with a weighted average of the post-processed predictions of the single models. We set the weights approximately proportional to the CCC validation score of each modality. Audio, Text and Fullbody have similar performances, hence same weights (0.29), while the weights of Landmarks and Rawface are respectively 0.1 and 0.03. The average predictions are then filtered using a Butterworth low-pass filter of order 1 and cutoff frequency 0.01 Hz.

III. Results & Conclusions

Figure 3 plots the predictions of the five different input streams and final integrated signal for Story #2 (validation) for Subjects 7 and 3 (the worst and best performing, respectively). We observed that accurate predictions are associated with low disagreement across modules. Some models performed in average better than others, but the weighting scheme roughly proportional to the prediction accuracies of each input stream provided optimal results. This simple multimodal integration method performed less effectively in situations of perceptual ambiguity, where different modules predicted different, sometimes opposite, estimations.

In the final evaluation results, our model's performance varied substantially across subjects and stories as is shown in Table I. Performance was very good on Stories #3 and #6, but very poor on Story #7. Similarly, for two out of ten subjects the average CCC is negative, while for the rest the average CCC ranges from 0.09 up to 0.34.

As we indicated in the introduction, our system should be taken as a first approximation to a multimodal integration system. Although we have spent considerable time and effort in optimizing the valence predictions from the individual input streams, we believe there is yet much space for improving the method of integrating the predictions across modalities and modules. For instance, the weighted average we have used, could probably be improved upon by more sophisticated machine learning models (although our initial experiments failed to achieve this). Nevertheless, our work on this system has provided us with some valuable insights.

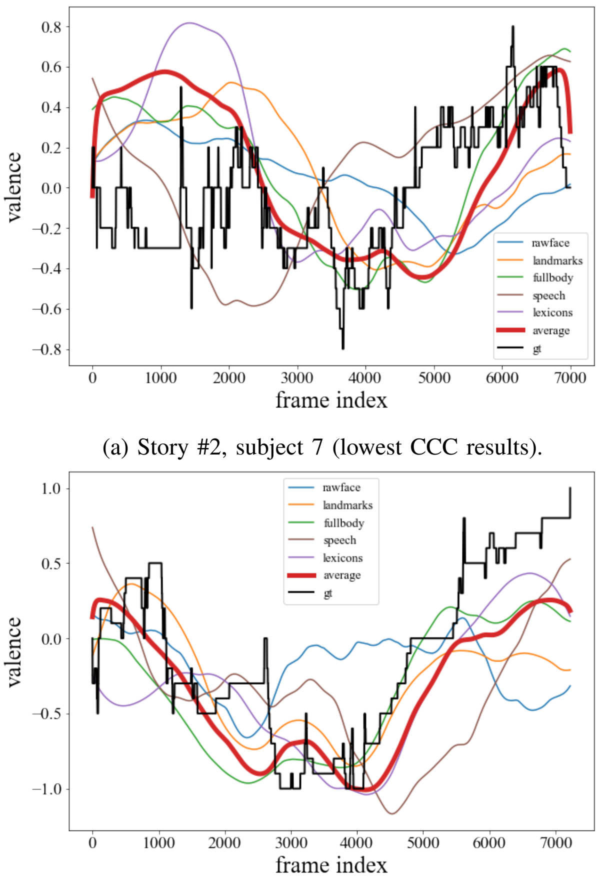
The first of this concerns the nature of the ground truth data themselves. As we have been training systems to try to match exactly the ground truth (and therefore optimize the CCC), we noticed that these data contain a large amount of high-frequency components The values oscillate between positive and negative several times within a few hundred milliseconds, which we suspect to be not solely reflective of emotional responses from the user, but caused by the input method. A more rigorous way of addressing this effect may help to estimate emotions better and may make the machine lerning more effective.

Our second conclusion concerns the overall shape of the curves. We found that when the training data have similar overall shapes and the neural networks do worse when that shape changes. This suggest that time-warping procedures for training data enrichment might be useful to ensure that the systems generalize better.

Subject	Story			Subj. Avg
	3	6	7	
1	0.43	0.62	-0.03	*0.34*
2	0.28	0.22	0.00	*0.17*
3	0.18	0.57	0.09	*0.28*
4	-0.11	0.62	0.04	*0.18*
5	0.09	0.86	-0.05	*0.30*
6	0.09	0.47	0.09	*0.22*
7	0.11	0.36	-0.21	*0.09*
8	0.16	-0.22	-0.01	*-0.02*
9	-0.11	0.11	-0.22	*-0.07*
10	0.08	0.66	0.09	*0.27*
Story Avg	*0.12*	*0.43*	*0.02*	*0.17*

TABLE I: Experimental results on the test set. Final results of Personalized and Generalized track are both 0.17 as the model submitted to the two tasks was the same.

Overall, the results show that the prediction of empathic emotional reactions is still a challenging task that deserves further investigation.



(a) Story #2, subject 7 (lowest CCC results).

(b) Story #2, subject 3 (highest CCC result).

Fig. 3: Example predictions: *speech* is for the audio module, *lexicons* for the language module, *gt* for the ground truth data and *average* for the overall prediction.

REFERENCES

[1] J.G.F. Francis, The QR Transformation I, *Comput. J.*, vol. 4, 1961, pp 265-271.

[2] H. Kwakernaak and R. Sivan, *Modern Signals and Systems*, Prentice Hall, Englewood Cliffs, NJ; 1991.

[3] D. Boley and R. Maier, ”A Parallel QR Algorithm for the Non-Symmetric Eigenvalue Algorithm”, *in Third SIAM Conference on Applied Linear Algebra*, Madison, WI, 1988, pp. A20.

[4] F. Weninger, J.R. Hershey, J. Le Roux and B. Shuller, ”Discriminatively Trained Recurrent Neural Networks for Single-Channel Speech Separation”, *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Atlanta, GA, 2014, pp. 577-581.

[5] H. Hermansky, ”Perceptual Linear Predictive Analysis for Speech”, *The Journal of The Acoustical Society of America (JASA)*, vol. 87, pp. 1738-1752, 1990.

[6] D.P. Kingma and J. Ba, ”Adam: A Method for Stochastic Optimization”, *3rd International Conference for Learning Representations*, 2015

[7] Davis E. King, ”Dlib-ml: A Machine Learning Toolkit”, *Journal of Machine Learning Research*, 2009

[8] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. ”Deep residual learning for image recognition.” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[9] Sepp Hochreiter and Jurgen Schmidhuber. ”Long short-term memory.” Neural Computation, 9(8):17351780. 1997.

[10] Warriner, A.B., Kuperman, V., and Brysbaert, M. ”Norms of valence, arousal, and dominance for 13,915 English lemmas.” Behavior Research Methods, 45, 1191-1207, 2013

[11] Staiano, Jacopo, and Marco Guerini. ”Depechemood: a lexicon for emotion analysis from crowd-annotated news.” Association Computational Linguistics, 2014.

[12] Yang, Zichao, et al. ”Hierarchical attention networks for document classification.” Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016.