EE583 Pattern Recognition Project
Inspection of *Deep Multilayer Perceptrons for Dimensional Speech Emotion Recognition*

Kutay Uğurlu 2232841

February 6, 2022

Contents

1 Introduction

Speech emotion recognition is the task of extracting the emotion category from either text or audio data. It has already some applications in security, medicine, entertainment and education [1].
This project report investigates the idea that Atmaja *et al.*stated in *Deep Multilayer Perceptrons for Dimensional Speech Emotion Recognition* [2]. The authors of the study discusses the need of utilizing modern computation units, such as Long Short Term Memory(LSTMs) and Convolutional Neural Networks(CNNs), in the neural networks for the task of dimensional speech emotion recognition.
The organization of the report is as follows:

- The problem and the proposed solutions to it are briefly introduced in this section.
- The theory of dimensional and categorical emotions are mentioned in Section 2.
- The methods, datasets and neural network architectures used in the implementation of the original paper along with modifications introduced in the speech emotion recognition tasks can be seen in Section 3.
- Section 4 is left for the presentation of the results from conducted experiments along with the reproduced results, and a discussion on them with a focus on comparison of the models.

2 Theory

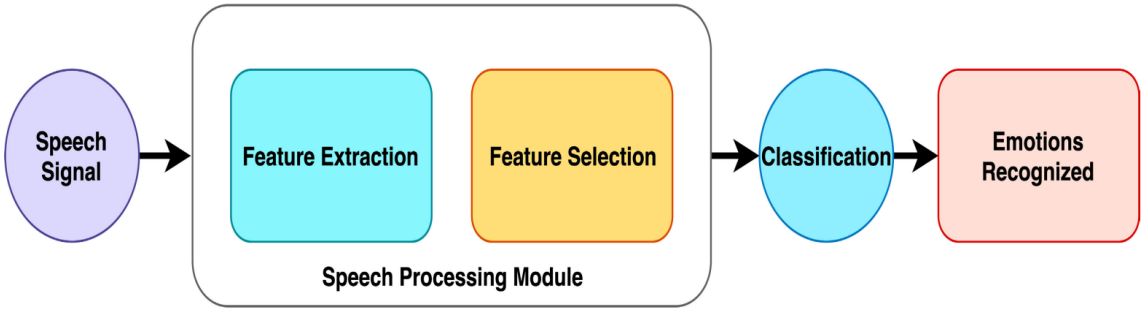


Figure 1: Traditional Speech Emotion Recognition Scheme [3]

The traditional algorithm for speech emotion recognition is demonstrated in Figure 1. Unlike these scheme, authors used to convert this categorical classification problem to a regression problem by using the dimensional emotion concept. In other words, they first predicted the magnitude of an utterance in each dimension. The remaining classification task afterwards is just deciding on which category an utterance belongs to in the dimensional emotion feature space.

The idea that emotion categories can be classified based on some emotional dimensions dates back to 1979. Russel argued in [4] that categorical emotions, namely sadness, anger, joy, etc., can be classified by the values they represent in three dimensions: Valence, Arousal and Dominance.

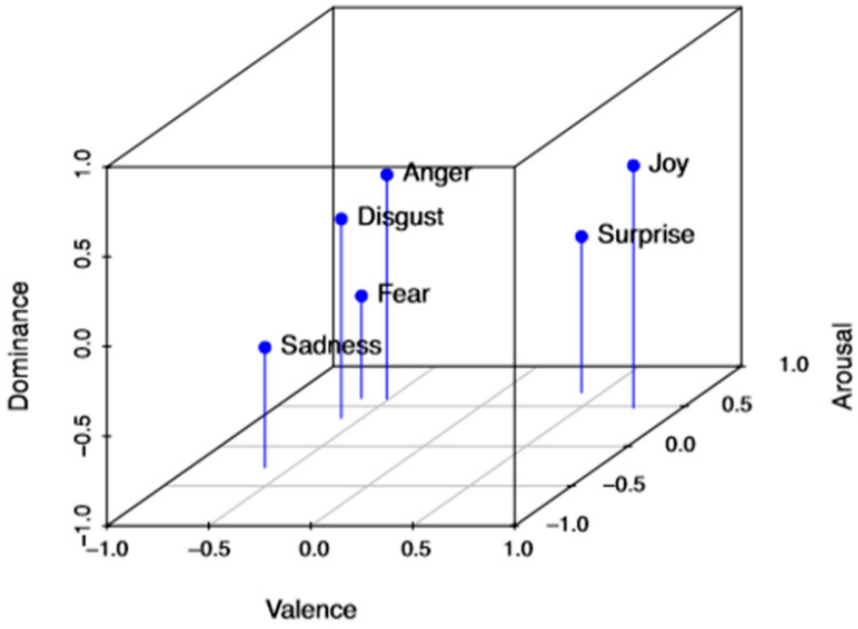


Figure 2: Categorical Emotions in VAD space [5]

Table 1: VAD Dimensions of 6 basic emotions [5].

	Valence	Arousal	Dominance
Anger	-0.43	0.67	0.34
Joy	0.76	0.48	0.35
Surprise	0.4	0.67	-0.13
Disgust	-0.6	0.35	0.11
Fear	-0.64	0.6	-0.43
Sadness	-0.63	0.27	-0.33

The categorical emotions of a given utterance can be decided based on its dimensional features' location in the Valence-Arousal-Dominance space. The authors of the study show that by using this idea, the Multilayer Perceptrons networks achieve more success, although being obsolete and computationally less costly, compared to more modern computational units such as LSTMs and CNNs, provided that the same number of architecture units are used and resultant number of parameters are not significantly different.

3 Implementation

This section composes of two sections describing the implementation of the original authors and slight changes made.

3.1 Original Paper

3.1.1 Data

3.1.1.1 Datasets

There are mainly two datasets utilized in the paper.

1. **IEMOCAP(The Interactive emotional Dyadic Motion Capture database):** 12 hours of speech data consisting of 10039 utterances is used [6].
2. **MSP-IMPROV:** 18 hours of speech data consisting of 8438 utterances is used [7].

3.1.1.2 Data Preprocessing

The shared data in [2] are already preprocessed and the scripts and tools the authors utilized was not explicitly shared. However, the types of features Atmaja *et al.* used is shared in the paper. The audio in the dataset are used to extract 47 features per utterance. These features are obtained in a 2-level process. First, the Low Level Descriptors defined in [8] are calculated by the `opensmile` software. These Low Level Descriptors are as follows:

- Intensity
- Alpha ratio
- Hammarberg index
- Spectral slope 0-500 Hz
- Spectral slope 500-1500 Hz
- Spectral flux
- 4 MFCCs
- f_0
- jitter
- shimmer
- Harmonics-to-Noise Ratio (HNR)
- Harmonic difference H1-H2
- Harmonic difference H1-A3
- F1
- F1 bandwidth
- F1 amplitude
- F2
- F2 amplitude
- F3
- F3 amplitude

Then, 47 features, High Statistical Functions of these 23 features are calculated in two sets by utilizing the mean and standard deviation. In addition, authors defined an extra feature: Silence. The silence is defined as the ratio of the silent frames per utterance.

$$S = \frac{N_s}{N_t} \tag{1}$$

where N_s is the number of silent frames and N_t is the number of total frames. The frames are labelled as silent by being compared to a threshold.

$$Threshold = 0.3 \times X_{RMS} \tag{2}$$

and

$$X_{RMS} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x[i]^2} \tag{3}$$

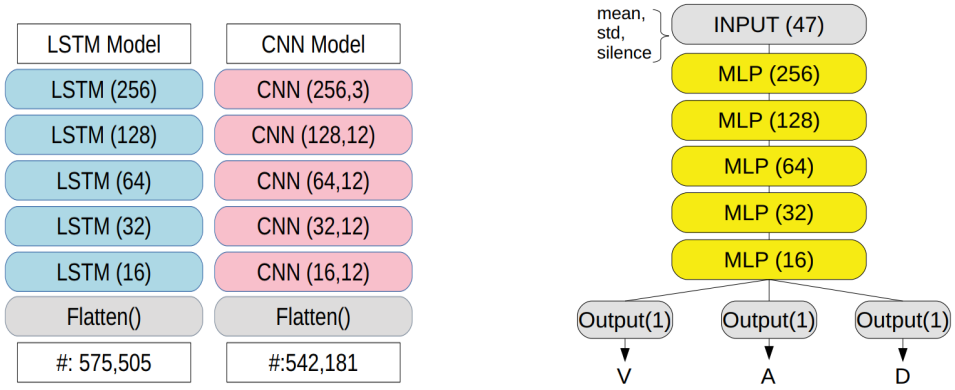
where the factor 0.3 is determined empirically.

3.1.1.3 Scaling

Both the data and the labels that were originally labelled in the range $[-5, 5]$ are scaled to the range [-1,1] in ReLU activated CNNs and LSTM networks.

3.1.2 Models

The benchmark models which authors use to compare the model performance are composed of CNN and LSTM units. The authors use the same number of units and layers in the model. In Figure 3, it can be seen that the models have approximately the same number of trainable parameters.
The multiplayer perceptron models are implemented with scikit-learn's MLP Regressor [9] and the remaining CNN and LSTM models are implemented with Tensorflow [10].



(a) Benchmark model architectures (b) Proposed Multilayer MLP architecture

Figure 3: Model architectures utilized in [2]

3.1.2.1 Loss Function: CCC Loss

To optimize the network, Concordance Correlation Coefficient in Eqn. 4 is used.

$$CCC = \frac{2\rho\sigma_x\sigma_y}{{\sigma_x}^2 + {\sigma_y}^2 + (\mu_x - \mu_y)^2} \tag{4}$$

Then, the loss can be found as $CCCL = 1 - CCC$.
The total loss with respect to this loss is defined to be the sum of loss of individual classes follows:

$$CCCL_T = CCCL_V + CCCL_A + CCCL_D \tag{5}$$

However, the authors state that they could not find a way to implement this loss function to optimize MLP models. Hence, they used MSE Loss instead in the training process, and CCC Loss again in the evaluation.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{6}$$

3.2 Modifications

The software provided in the repository [1] is copied via `git clone` and some additional changes made on the repository. The modified version is shared here [2].

3.2.1 Data

3.2.1.1 Datasets

The authors of the study made use of the most of the databases whose utterances are labelled by multiple annotators in the Valence-Arousal-Dominance dimension. Although there are many available datasets on Speech Emotion Recognition, only a few of them are labelled in the same manner. However, these datasets were not available for open access. The remaining available datasets are either labelled categorically, *i.e.* sad, joy, anger etc. or labelled only in 2 dimension: Valence and Arousal. Since the latter option requires one to re-train the model, does not provide one with the chance of evaluating the pretrained model with the new test data and results in an unfair evaluation of the results since the dimension of the problem is changed, categorical dataset, MELD [2], was used for the evaluation. The dataset consists of utterances in mp4 format from the TV series "Friends" and the categorical emotions regarding these utterances. The number of labeled utterance is 2610. However, the number of utterances labelled as "neutral" in the emotions, which corresponds to the origin of the Valence-Arousal-Dominance space, was exceedingly higher than the others, and it was causing problems in both class-imbalance sense and in the learning part of the regression networks, hence those samples were dropped from the dataset.

3.2.1.2 Data Preprocessing

To augment the dimensional, *i.e.* numerical, labels from the categorical labels. This is conducted by mapping the categorical labels to points in the Valence-Arousal-Dominance space via the mapping given in Table 1 first, and perturbing the points with normally distributed noise with standard deviation 0.01, to reflect the effect of multiple annotators. To get the time series data in WAV format in the same datarate of 44.1 kHz from MP4 format, `FFMPEG` is utilized.
The whole data formation process can be seen in Figure 4.

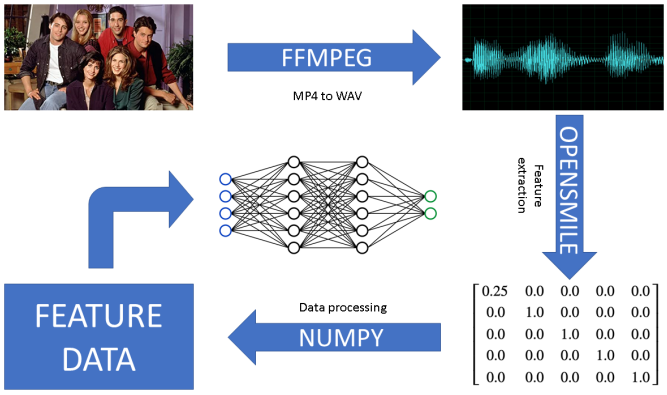


Figure 4: The data formation steps used to augment data from MELD dataset

The data processing steps shown above visually in Figure 4 is explained in detail in section 3.1.1.2.

[1] https://github.com/bagustris/deep_mlp_ser
[2] https://github.com/kutay-ugurlu/583-Project

3.2.2 Models

The pre-trained models are evaluated using the new test data. Furthermore, to make better conclusions on the models' capability to fit the data, the data is split to train, test and validation sets and the model is evaluated after being retrained. The architectures of the model were not changed. The only change made was the number of training and patience epoch for early stopping. The comparison was only made for speaker independent case due to the requirement of splitting the newly introduced test data in a speaker-wise manner and train the models using mixed datasets in the speaker dependent case and lack of available continuous data from the same speaker.

4 Results and Discussion

Table 2: CCC results of the conducted experiments, TS1:IEMOCAP, TS2:IMPROV, M:MIXED

	Original Mean CCC	Tested Valence CCC	Tested Arousal CCC	Tested Dominance CCC	Trained Mean CCC
CNN TS1	0.330	0.022	0.043	0.043	0.130
CNN TS2	0.372	-0.000	-0.000	-0.000	0.103
CNN M	0.204	0.032	0.046	0.046	0.119
LSTM TS1	0.359	0.013	-0.000	-0.000	0.118
LSTM TS2	0.338	-0.001	-0.003	-0.003	0.113
LSTM M	0.215	-0.008	-0.002	-0.002	0.118
MLP TS1	0.434	0.042	0.020	0.020	0.005
MLP TS2	0.419	0.035	-0.058	-0.058	0.005
MLP M	0.294	0.104	0.050	0.050	0.005

The CCC values obtained from the experiments are summarized in Table 2. The **Original Mean CCC** columns represent the reproduced mean of three dimensional CCC results in the original paper [2]. Following three columns represent the values of the three dimensional evaluation results with the new test data in 3.2.1. The last column represent the values obtained from the experiments where the new test data is used both in training and evaluation.

One may notice the performance difference between the dataset by comparing the numbers in the 1st column with the following three ones. These results imply that the models trained with the datasets used in the study do not generalize to the other test data. The mean of the results of the **Trained Mean CCC** case is used to replace the different results with k-fold cross validation results.
The **Trained Mean CCC** results slightly varies among the experiments. The results are obtained via 3 different train-test splits and the mean of them can be considered as the 3-fold cross validation CCC.

$$CNN_{avg} = 0.117 \qquad LSTM_{avg} = 0.116 \qquad MLP_{avg} = 0.005$$

4.1 Number of samples in the training set

The difference between the training and evaluation within the same datasets, *i.e.*, **Original Mean CCC vs Trained Mean CCC**, can be explained by the number of utterances utilized in the datasets. There are 9-fold difference in the size of the datasets between the cases, and within-dataset performances rely highly on the data that is provided to the network. The size difference directly affects to which extent the model can fit the conditional probability distribution.

4.2 Number of training data distributions

One noticeable result is that mixing corpora, *i.e.* forcing models to learn 2 different data distributions, resulted in poorer performance in the original study. This is expected, since the same model is being fitted to the two distributions, hence the prediction performance gets poorer in the test data from the same distributions. However, when we inspect the Multilayer perceptrons models in the second case, we observe a significantly increased generalization capability by three times. This may be due to the fact that this model has been trained on multiple different samples from multiple distributions, performing better in a never-seen sample. The same situation is also applicable for CNNs for the valence dimension of the regression.

4.3 Labelling and Human Bias

The labelling process of the data used in the study was conducted by multiple annotators. The annotators scored the utterances in Valence-Arousal-Dominanca dimension and as the ground truth mean of the scores are utilized. In the newly introduced test data, the labels are augmented from categorical labels and perturbed with Gaussian noise with standard deviation 0.01, to introduce the human bias to the artificial labels. In the first labelling process, due to nature of the speech emotion recognition task, the ground truth was not clear and indisputable by the annotators. That is to say, the utterances may have been scored significantly different among the annotators and datasets. This is because the task of perceiving the emotion in an utterance not an objective task such as a vision task where people classify the vehicles they see in images. This may be another contributing factor to the performance difference in cross-dataset scenarios. In addition, simply taking the mean of the labels scored by annotators may be causing some bias among datasets. In other words, the closer categorical labels, such as surprise and joy, may have been simply labelled the same by coincidence or there could be some disagreement on these measures. For this problem, even a statistical technique Cohen's Kappa(κ) was introduced [11], to measure the inter-rater reliability.

4.4 Architecture

Inspecting the **Trained Mean CCC** results, one can conclude that the modern computational units such as LSTM and convolutional layers outperformed the classical MLP. Although, the extracted features mentioned in Section 3.1.1.2 do not have a straightforward time-series relation like the one observed in raw audio data, CNNs and LSTMs have an advantage over capturing the mutual (or sequential), relationship that is not easily comprehensible by humans. On the other hand, Multilayer Perceptron models regards these features as if they are completely independent and there is not a spatial relationship between the consecutive ones. This may be the reason why they are outperformed by the more modern architectures. The converse situation that is observed in the original paper seems to be a specific result of hyperparameter optimization along with the used datasets.

References

[1] L. Cen, F. Wu, Z. L. Yu, and F. Hu, "A real-time speech emotion recognition system and its application in online learning", p. 27–46, 2016.

[2] B. T. Atmaja and M. Akagi, "Deep multilayer perceptrons for dimensional speech emotion recognition", *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, p. 325–331, 2020.

[3] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, and T. Alhussain, "Speech emotion recognition using deep learning techniques: A review", *IEEE Access*, Vol. 7, p. 117327–117345, 2019. DOI: 10.1109/ACCESS.2019.2936124.

[4] J. A. Russell, "Affective space is bipolar.", *Journal of personality and social psychology*, Vol. 37, No. 3, p. 345, 1979.

[5] O. Blan, G. Moise, L. Petrescu, A. Moldoveanu, M. Leordeanu, and F. Moldoveanu, "Emotion classification based on biophysical signals and machine learning techniques", *Symmetry*, Vol. 12, No. 1, p. 21, 2020.

[6] C. Busso, M. Bulut, C.-C. Lee, "Iemocap: Interactive emotional dyadic motion capture database", *Language resources and evaluation*, Vol. 42, No. 4, p. 335–359, 2008.

[7] C. Busso, S. Parthasarathy, A. Burmania, M. AbdelWahab, N. Sadoughi, and E. M. Provost, "Msp-improv: An acted corpus of dyadic interactions to study emotion perception", *IEEE Transactions on Affective Computing*, Vol. 8, No. 1, p. 67–80, 2016.

[8] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: The munich versatile and fast open-source audio feature extractor", in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, p. 1459–1462.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, Vol. 12, p. 2825–2830, 2011.

[10] Martín Abadi, Ashish Agarwal, Paul Barham, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: https://www.tensorflow.org/.

[11] J. Cohen, "A coefficient of agreement for nominal scales", *Educational and Psychological Measurement*, Vol. 20, No. 1, p. 37–46, 1960. DOI: 10.1177/001316446002000104. eprint: https://doi.org/10.1177/001316446002000104. [Online]. Available: https://doi.org/10.1177/001316446002000104.