Comparing Performance of Classifiers on Labeling Emotions in Speech

GROUP 1

Berk Bayraktaroglu (u66609)

Maksims Habirovs (u254095)

Daniils Krasovskis (u657456)

Linda Selles (u212878)

Vladislavs Sokolovs (u718659)

Abstract

In this paper three different machine learning models, namely a CNN, CNN-LTSM and SVM are compared on their performance on Speech Emotion Recognition (SER) on the dataset RAVDESS. The performance is measured with the accuracy of the models on the classification task.

Keywords: Machine Learning; Emotion; SVM; CNN; CNN-LSTM; Speech Emotion Recognition

Introduction

Speech recognition is already implemented in services that are used daily such as Alexa, Siri and many more applications. However, the recognition of emotion in speech is a more recent development. Recognizing an emotion can be important for communication. Since different emotions require different responses.

The challenge in the recognition of emotion in speech is the model and the dataset that is used for this recognition. Many researchers have been studying this particular subject, starting with the founding work "Affective Computing", 2000 by Rosalind Picard and the RAVDESS (in American English), IEMOCAP (in American English) and EMODB (in German) datasets which are used a lot among Speech Emotion Recognition (SER) researchers nowadays to apply Machine Learning and Deep Learning algorithms (Singh & Goel, 2022; Asiya & Kiran, 2021; Issa et al., 2020; Jha et. al., 2022; Livingston & Russo, 2018; Busso et. al., 2008;). There are several different models that could be used but the question is what model performs best. For example k-Nearest Neighbors performs worse than Support Vector Machine according to Jha et al. (2022).

Thus, is there a significant difference in performance when using the Support Vector Machine (SVM), the Convolutional Neural Network (CNN) or the CNN with Long Short-Term Memory (LSTM) on the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) developed by Livingstone and Russo (2018)?

Methods

Three models were compared in terms of accuracy metrics, namely - Support Vector Machine, Deep Convolutional Neural Network and Hybrid Convolutional Neural Network with Long Short-Term Memory described by Zhao et al. (2018).

These models will be compared to the performance of Puri et al. (2018) who did a CNN approach and got an accuracy of 98% and compared to the Support Vector Model of Jha et al. (2022) that had an accuracy of 78%. Both papers classified emotions on the RAVDESS dataset.

The SVM model was assembled using the Sklearn package in Python. Both Deep Learning models were assembled using Keras from a Python library called TensorFlow. The aforementioned paper contained 2 proposed CNN-LSTM models that were interesting. There was a 1-dimensional and a 2-dimensional model. However, due to lack of computational power and use of locally connected convolutional layers, only the 1-dimensional model could be run, trained and compared.

Dataset The dataset that was used to train and measure the models on, is the Ryerson Audio-Visual Database of Emotional Speech and Song developed by Livingstone and Russo (2018). From this dataset only the speech files were used. This dataset had speech files of 24 actors of which every actor has 60 speech recordings. There are 12 female actors and 12 male actors, making this dataset balanced. The statements that the actors spoke contain different emotions such as anger, fear, calmness, happiness, sadness, disgust, surprise and neutral. The statements also differ in emotional intensity with levels "normal" and "strong".

Data Splitting and Preprocessing The data for all the mentioned models was strategically split to make sure that the proportions of emotion and gender are preserved equally. 80% of the data was used as training data and the remaining 20% of the data was used as test data due to the RAVDESS dataset being relatively small. There also was an attempt to strategically split the data on the intensity of the emotions which is provided in the RAVDESS dataset, but it made the models classify worse than they did before the additional split, the reasons for that may be a big amount of classes to classify or intensity does not contribute a lot to the distinction between same emotion intensity variability.

Feature Extraction Different features were used for the SVM and both CNN with CNN-LSTM. For the SVM model the features that were used are: Mel-frequency Cepstral Coefficients (MFCC's) and delta-mfcc's concatenated, where the amount of the first 20 MFCC's we took was 13, and chroma(pitch) features. All the features were padded to make them uniform in length. For both the CNN and CNN-LSTM there was only one feature used, namely the Mel Spectrogram with 128 mel-frequency bands. We used Mel Spectrograms for both models because they are very useful in both case scenarios: either it is convoluting over an image which the Mel Spectrogram is or using it as a sequence of frequencies that are distributed over time which they are too.

Model Architectures The SVM model used the SVC() function from the Sklearn, the kernel was RBF, gamma value was set to 'auto' and the C value of the classifier was set to 1000.

The CNN model has three convolutional layers, three pooling layers, 4 dropout layers, 1 after each pooling layer and 1 before the dense layer. In between all convolutional and pooling layers, there is a batch normalization layer. Also the model has 1 flatten layer after all the convolutional and pooling layers and before 2 dense layers, where the first dense layer serves as a layer to learn the patterns from the data and the second one is a layer that classifies the emotions. We also implemented a kernel regularizer for the first dense layer because we encountered a problem of overfitting and L2 regularization contributed to solving this issue.

The CNN LSTM network is constructed by stacking four Local Feature Learning Blocks (LFLB), as they are called in the paper by Zhao et al. (2018), one LSTM layer and one fully connected layer. The LFLB was designed as a substitute for a traditional Convolutional Neural Network by Zhao et al. (2018) to extract emotion-related features. An individual LFLB includes a convolutional layer, a batch normalization layer, an Exponential Linear Unit (ELU) layer, and a max-pooling layer. The essential components of the LFLB are the convolution and pooling layers. The 1-dimensional CNN-LSTM architecture is constructed by connecting four of these LFLBs, followed by an LSTM layer, and finally, a dense layer.

Feature Scaling and Dimensionality Reduction To scale the features for the SVM model we used Standard Scaler and MinMaxScaler in different runs. To reduce the dimensionality of the data Principal Component Analysis (PCA) was used to keep 95% of the data's explained variance. To find the best algorithms that suited the task, we used different combinations of scalers either with PCA or without it. In our final run we used Standard Scaler and no PCA, that was the best combination of the algorithms we could possibly find.

Hyperparameters of the Deep Learning Models To increase the overall effectiveness of the search for the best hyperparameters for the CNN and CNN-LSTM models, we used a Bayesian Optimisation which Zhao et al.(2018) also used to enhance the overall performance of their models. We applied this approach to our CNN model.

The search space of the CNN model's hyperparameters was as follows:

- Dropout rate after each convolution layer: 0.0 - 0.2
- Dropout rate after first dense layer: 0.0 - 0.5
- All optimizers available in Keras, namely: Adam, SGD, RMSprop, Adagrad, Adadelta, Ftrl, Nadam, AdamW
- Optimizers learning rate: 0.001 - 0.000001
- First dense layer's kernel regularizers L2 value: 0.1 - 0.0001

The range of the search for the dropout rate was instantiated according to Hinton et. al., 2012.

The search space of the CNN-LSTM model's hyperparameters was as follows:

- All optimizers available in Keras, namely: Adam, SGD, RMSprop, Adagrad, Adadelta, Ftrl, Nadam, AdamW
- Optimizers learning rate: 0.001 - 0.000001

Results

Support Vector Machine Model

The SVM model was assembled using the Scikit-Learn package.
During the preprocessing of the data and running the model we used several different methods such as delta2 MFCC, librosa.trim to remove silence, PCA to reduce dimensionality were used but these methods showed no improvement or even made the model perform worse than our first run without any of these methods. However, the standard scaler and minmax scaler showed a slight improvement of about 1-2 %, whereas setting the 'C' value to 1000 showed a great improvement of 15% accuracy. Using features such as Mel Spectrogram and Spectral Contrast as features showed no improvement of the model.

The best performing SVM model had an accuracy of 62% when the C value of the classifier was set at 1000, MFCC and MFCC-delta concatenated and Chroma was used as features; also we used a StandardScaler. Data was split strategically on gender and emotion. Figure 1 shows the confusion matrix of the SVM model. On the x-axis are the predicted emotions and on the y-axis are the true emotions. In table 1 the emotions related on the axis can be seen.

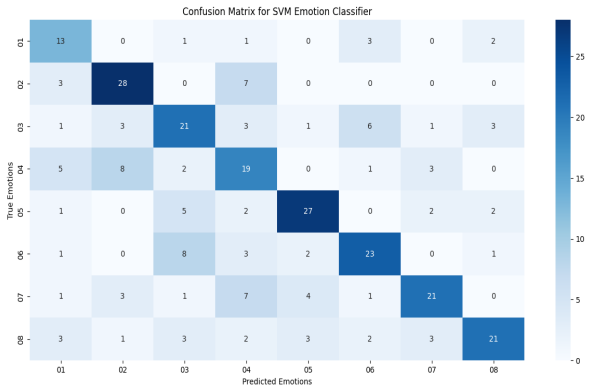

Figure 1: Confusion Matrix of the SVM emotion classifier

Table 1: The emotions related to numbers in the confusion matrices

Number	Emotion	Number	Emotion
01	Neutral	05	Angry
02	Calm	06	Fearful
03	Happy	07	Disgust
04	Sad	08	Surprised

Convolutional Neural Network Model

For the CNN model a TensorFlow Keras package was used. The model often got stuck at 60% due to overfitting. The model did however perform with an accuracy of 71% on epoch 759. Figure 2 shows the confusion matrix of the CNN model, which in some degree have differences compared with the SVM and CNN-LSTM. The model was trained for 1000 epochs.

CNN-LSTM Model

For the CNN-LSTM model a TensorFlow Keras package. With this model we wanted to recreate the architecture of the model proposed by Zhao et al.(2018) and achieve comparable results, but in the paper the highest accuracy for 1D-CNN-LSTM model is not mentioned, either because of being in the mix with mean accuracies of the many model runs together with 2D version, or simply not given enough attention. The highest accuracy that was reached was 59% accuracy on epoch 141. The model converged fast and the accuracy did not improve later. The confusion matrix can be seen in Figure 3. The model was trained for 1000 epochs.

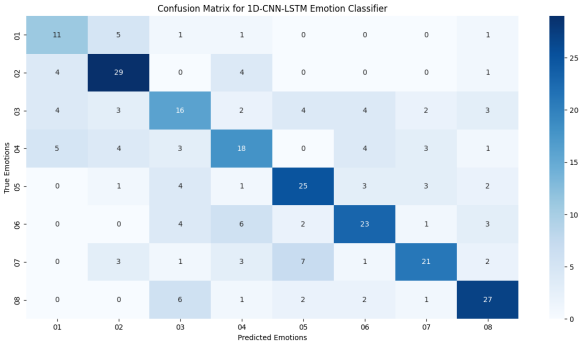

Figure 2: Confusion Matrix of the best performing CNN-LSTM emotion classifier

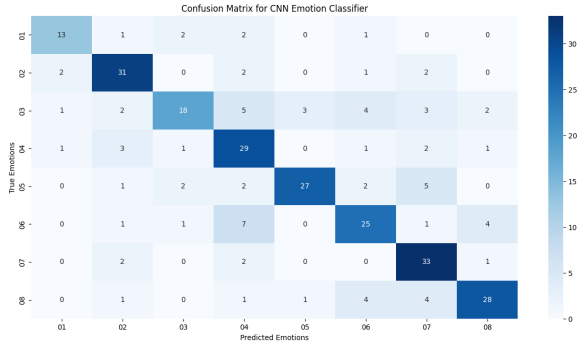

Figure 3. Confusion Matrix of the best performing CNN emotion classifier

Table 2: Performance Measures for the best performing SVM, CNN and CNN-LTSM models

Emotion	SVM Precision	SVM Recall	CNN Precision	CNN Recall	LSTM Precision	LSTM Recall
Neutral	0.50	0.40	0.76	0.68	0.46	0.58
Calm	0.60	0.82	0.74	0.82	0.64	0.76
Happy	0.48	0.61	0.75	0.47	0.46	0.42
Sad	0.55	0.41	0.58	0.76	0.50	0.47
Angry	0.62	0.66	0.87	0.69	0.62	0.64
Fearful	0.79	0.58	0.66	0.64	0.62	0.59
Disgust	0.60	0.72	0.66	0.87	0.68	0.55
Surprised	0.71	0.53	0.78	0.72	0.68	0.69

Discussion

A note to make, these models were subject-dependent, so the same actors appeared both in train and test data in a random manner. For future developments and as a part of the same discussion, it would be a good idea to rerun all the models making them subject-independent with no reappearing actors in train and test data.

It is unfortunate that the 2D-CNN-LSTM model could not be run, because this model's accuracy on the Berlin EmoDB dataset was 95.89% (subject-independent) and 95.33% (subject-dependent), also it would be more plausible to compare with the results that are achieved in this paper to the proposed architecture (Zhao et al., 2018).

Also it would be nice to try our models with different data, because depending on the dataset, the accuracy may vary significantly. Also considering the fact that the CNN-LSTM model performed worse than other models, it may be possible to infer that either sometimes machine learning algorithms may be suited for a task better than deep learning ones or the DL algorithm is not well-suited for the task by itself.

Which model is actually the best not only on the accuracy, but also on time that is required for it to produce desired results is a food-for-thought type of question. Whether it is worth spending 1000 epochs to get 10% more accuracy rather than using an SVM is not something that can be answered directly and depends on the task. Although, if dealing with anything as sensitive as human emotions, it may be worth delving into "quality" rather than "quantity", aiming for the best possible accuracy.

Conclusion

This paper explores how different ML algorithms perform on a speech recognition task. Three models were chosen for the task, 2 of them are our own work, namely SVM and CNN, and the other is a 1D-CNN-LSTM model from the paper by Zhao, et al. (2018).

In terms of accuracy, the best model was our CNN model with accuracy of 71%, then SVM with 62% accuracy and lastly a 1D-CNN-LSTM with 59%. This means that our CNN model performed not as good as the CNN model of Puri et al. (2018), they reached an accuracy of 98%, as well as the SVM model by Jha et al. (2022), who got an accuracy of 78%. However, what is consistent with all of these papers is that the CNN model performs better on classifying emotions in speech than the SVM.

Statement of Technology

As mentioned in the paper the data that we use is the RAVDESS which is constructed by Livingstone and Russo (2018). For writing the code, training the models and generating the figures and tables, as an IDE PyCharm was used; as programming language Python was used and the libraries: scikit-learn, librosa, tensorflow, seaborn, matplotlib, numpy packages. For the reference list the website scribbr.com was used as a guiding tool.

CRediT

Berk Bayraktaroglu: Conceptualization, Resources, Methodology, Writing- Review and Editing. **Maksims Habirovs:** Conceptualization, Investigation, Validation,, Resources. **Daniils Krasovskis:** Conceptualization, Investigation, Project administration, Resources. **Linda Selles:**

Conceptualization, Resources, Writing - Original Draft, Writing - Review & Editing, Visualisation, Project Administration. **Vladislavs Sokolovs:** Conceptualization, Methodology, Software, Resources, Visualisation, Writing - Original Draft, Writing - Review & Editing, Project Administration.

References

Asiya, A. U., & Kiran, K. K. (2021). Speech Emotion Recognition-A Deep Learning Approach. In *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. https://doi.org/10.1109/i-smac52330.2021.9640995

Busso, C., Bulut, M., Lee, C., Kazemzadeh, A., Mower, E., Kim, S. S., Chang, J. N., Lee, S., & Narayanan, S. S. (2008). IEMOCAP: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, *42*(4), 335–359. https://doi.org/10.1007/s10579-008-9076-6

Issa, D., Demirci, M. F., & Yazici, A. (2020). Speech emotion recognition with deep convolutional neural networks. *Biomedical Signal Processing and Control*, *59*, 101894. https://doi.org/10.1016/j.bspc.2020.101894

Jha, T., Kavya, R., Christopher, J., & Arunachalam, V. (2022). Machine learning techniques for speech emotion recognition using paralinguistic acoustic features. *International Journal of Speech Technology*, *25*(3), 707–725. https://doi.org/10.1007/s10772-022-09985-6

Livingstone, S. R., & Russo, F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLOS ONE*, *13*(5), e0196391. https://doi.org/10.1371/journal.pone.0196391

Picard, R. W. (2000). *Affective Computing*. MIT Press. https://doi.org/10.7551/mitpress/1140.001.0001

Puri, T. A., Soni, M., Dhiman, G., Khalaf, O. I., Alazzam, M. B., & Khan, I. R. (2022). *Detection of Emotion of Speech for RAVDESS Audio Using Hybrid Convolution Neural Network* (Vol. 2022). Hindawi Publishing Corporation. https://doi.org/10.1155/2022/8472947

Singh, Y. B., & Goel, S. (2022). A systematic literature review of speech emotion recognition approaches. *Neurocomputing*, *492*, 245–263. https://doi.org/10.1016/j.neucom.2022.04.028

Zhao, J., Mao, X., & Chen, L. (2019). Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomedical Signal Processing and Control*, *47*, 312–323. https://doi.org/10.1016/j.bspc.2018.08.035