JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DESIGN OF EFFICIENT SPEECH EMOTION RECOGNITION SYSTEM

M. Hima Bindu,
Assistant Professor, Department of ECE,
Institute of Aeronautical Engineering

C V P Supradeepthi,
Assistant Professor, Department of ECE,
Institute of Aeronautical Engineering

Abstract:
Emotion recognition is one of the most active areas of research in the field of artificial intelligence. While emotions can be drawn from many modes like semantics, visual cues and even multimodal applications, speech alone carries a dominant part of emotions. To accomplish this task, many approaches have been formulated and applied over the years like Advanced signal processing, Machine learning and artificial neural networks. Although they are very reliable, they require a lot of resources in terms of training and testing which in turn increases the computational time and necessary hardware requirements. In this , a simple speech emotion recognition system was designed using a comparatively small data set and a simple deep neural network. An MFCC feature extractor was used to capture necessary pitch variations from the audio files and then these features were fed into a 1D convolutional neural network. This framework was further optimized by varying hyperparameters from both the feature extractor and the neural network which resulted in a decent accuracy over a less computational time and hardware.

Keywords: CNN, Deep neural networks, MFCC, Feature extraction.

Introduction:

Emotions are a complex set of expressions used by human beings to communicate with each other. Emotions can be conveyed in the form of various senses like Visuals, Speech, text etc. Speech is one of the main sources to produce emotions. Speech emotion can be further classified into Vocal emotion and Semantic emotion. Emotion recognition is an important task in the field of sciences and many product-based applications. It helps us to understand the working of social and intellectual aspects of the human brain. But extracting emotions and using them in those applications cannot be done by any biological interventions since the world is digitally driven. Extraction of emotions manually from possible sources and scenarios is also a very tedious and unappreciated practice. This is where the requirements of computer-based emotion recognition systems are used. While human beings are equipped with the brain to perform this task in an unsupervised yet efficient manner, it is hard for a machine to understand and interpret these emotions for humans. Conventional problem-solving techniques like data structures and corresponding optimization algorithms cannot be used to extract emotions from the speech data. Hence, data driven techniques like signal processing and machine learning are used to extract information that cannot be logically interpreted.

Extracting emotions from semantics is more inclined to the domain of Natural Language processing. They use pretrained text-emotion based approaches to capture emotions. But emotion recognition cannot be limited or biased to semantics because humans show complicated emotional queues like sarcasm which might contradict a text-based emotion prediction. This is where the prediction of vocal based emotions is used. This type of emotion recognition systems can capture variations in amplitudes and frequencies in the voice to capture emotions from the speech. This unlike NLP techniques can be accomplished by simple procedures like signal processing and basic machine learning. Many techniques have been formulated and applied over the years on vocal based speech emotion recognition.

The feature extraction is the heart of this speech emotion recognition pipeline. The absence of a feature extractor will cause the machine learning model to learn unnecessary patterns. The modern approach to exclude a feature extractor is to use a relevant deep learning framework. But deep learning frame works can also be used alongside a feature extractor in some situations. The job of a machine learning model is to train a complicated mathematical function that fits and models a set of data based on its labels through various algorithms. The inputs we offer to our machine learning model are speech data each mapped to a specific emotion. The machine learning model captures patterns from the data to predict the emotion with a given accuracy.

Methodology:

In this paper, deep learning (a subset of machine learning) is used to train the data and make predictions on some unseen data the model has never been trained on. Although, it is not a good practice to feed the data directly into a deep learning model since there will be a limitation to the capability of the model to make predicts from raw data. A better practice would be to give the deep learning model a set of relevant features that are extracted using a separate mechanism (apart from the machine learning model) so that the model can capture necessary patterns and make generic predictions.

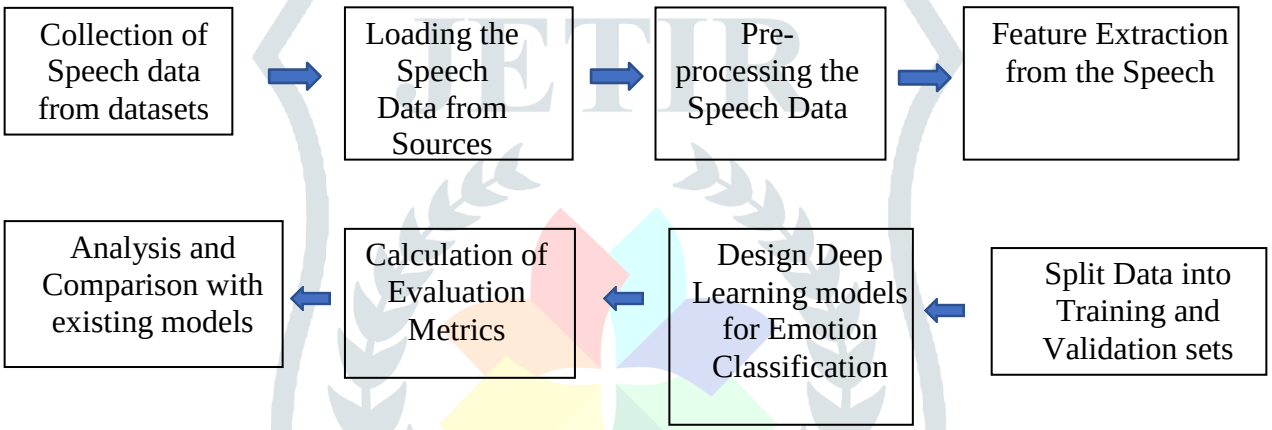

Fig:1 Flow diagram of Speech emotion recognition system

The steps for the Speech emotion recognition system are as follows

- Collection of labeled data mapping speech to emotions.
- Structuring the collected data for ease of access.
- Preprocessing the speech data.
- Feature extraction.
- Defining and building the deep learning framework.
- Train and validate the deep learning model.
- Evaluate the metrics.
- Tune the hyperparameters from feature extractor and the deep learning model.

The process between feature extraction and tuning hyperparameters is a cycle which converges at the maximum possible validation accuracy and minimum possible validation loss. This process is not automated in this project because the resources needed to do so were found to be significantly high. The development of the emotion recognition system involves changing the hyperparameters of both the feature extractor and the machine learning model to get better validation accuracy.

Optimization of the proposed framework:

The main objective of this paper is not only achieving a higher accuracy and other metrics on validation data, but also achieve a shorter computation time. The total computation time is calculated as the sum time taken to extract features of all the samples and the time taken to complete forward propagation of all the samples. So, the optimization needs to be done in such a way that our computation time decreases without reducing the validation accuracy and F1 score. The framework was started by calculating MFCC's at 44100Hz and 36 MFCC. These features were calculated and fed into neural network for training. Now, the hyperparameter of the neural network were varied until it gave a better accuracy. Here, accuracy was prioritized in the neural network because the computation time that a feed forward operation takes is significantly less compared to the computation time of feature extractor. This is due to the fact that feed forward operation in a neural network is taken up by the GPU of the host computer and the feature extractor is more intense in computation and runs on the host computer's CPU.
Tuned Hyper parameters in a Neural Network are:

- Learning Rate.
- Number of Epochs.
- Batch Size.
- Number of Nodes in Each layer.
- Dropout ratio.

After getting the optimal neural network with best validation accuracy and least computation time, The feature extractor hyperparameters which are Sampling Rate and MFCC count are tuned respectively and the model was re-trained every time the feature extractor hyperparameters were changed leaving the tuned neural network parameters unchanged. This gave the best model in terms of both validation accuracy and computation time.

Based on the framework and optimization scheme proposed in the methodology section of paper, the observations and results are obtained as following steps

- Visualization of Step-to-Step MFCC Evaluation of a random dataset sample.
- Tensorflow representation of the Convolutional Neural network (layer wise).
- Model metrics before optimization (both MFCC and CNN).
- MFCC Hyperparameter tuning results in terms of computation time.
- Graphs of Training and Validation Loss of the model with best MFCC hyperparameters.
- Graphs of Training and Validation Accuracy/ score of model with best MFCC hyperparameters.
- Confusion Matrix.
- Class-wise analysis of Precision, Recall and F1 scores.

The novelty in this paper is brought by the development phase and choosing the hyperparameters from both feature extractor and the model. Any improvement will be counted as the parameter difference or it's percentage difference from the initial observation. There are no specific units for the training or validation loss. The frequency units are not common to every spectrogram as a nonlinear frequency scaling (Mel) is introduced. Its either Mels or Hertz. The Memory utilized to finish the computations inside the RAM (CPU or GPU) are in Giga Bytes (GB). The Time taken to finish the computations is MM: SS format, where MM is minutes and SS is seconds. The parameter count is also a unit less yet significant quantity.

Results :

In the feature extractor optimization, there are only 2 hyperparameters to be tuned:

- Signal Sampling Rate.
- MFCC Feature Count (Number of Mel Bands).

The initial sampling rate and MFCC feature count are 44100 Hz and 36 features. The neural network was trained every time upon optimization of the feature extractor. The validation accuracy score and computation time are calculated upon each change in hyperparameters. First, the sampling rate of the feature extractor was changed followed by the number of MFCC features. Also, samples with "Disgust" emotion were dropped here.

Table 1 : Results of the model by varying sample rate of the MFCC extractor

Sampling Rate (Hz)	MFCC Feature Count	Validation Accuracy Score	F1 - Score	Time (Minutes) (MFCC + CNN)
44100	36	0.7553	0.81	09:05 + 01:24
22050	36	0.7743	0.77	08:14 + 01:19
16000	36	0.7690	0.79	08:05 + 01:03
8000	36	0.8010	0.84	08:01 + 00:58

Since the best sampling rate for the feature extractor that suits the CNN model is 8000 Hz, the MFCC count is varied in the next step keeping this best sampling rate common.

Table 2 : Results of the model by varying feature count of the MFCC extractor

Sampling Rate (Hz)	MFCC Feature Count	Validation Accuracy Score	F1- Score	Time (Minutes) (MFCC + CNN)
8000	30	0.9017	0.9014	05:47 + 00:45
8000	25	0.7826	0.82	05:58 + 00:57
8000	20	0.7703	0.77	05:45 + 00:38
8000	10	0.7948	0.80	05:30 + 00:33

After varying the MFCC count hyperparameter, the highest validation accuracy score is produced when feature count is equal to 30. The increase in validation accuracy score compared to plain network optimization score is 0.1464 (14.64%).The increase in validation accuracy score compared to un-optimized model is 0.1856 (18.56%). The improvement in computation time between un-optimized model and final model is from 10:29 minutes to 06:02 minutes. It is 04:27 minutes faster (42.44%) than the prior model.

After feature extraction, the data is split into training and validation sets with training data being 80% and validation data being 20%. Since the split is done randomly, it is stratified with respect to the emotions so that the balance of emotions is not different in train and validation sets. The Deep learning architecture used in this paper is a trainable 1D Convolutional Neural network that has 2 convolutional blocks and 3 Dense layers bridged by a Global Average pooling layer. The third dense layer is the output softmax layer with 6 nodes each representing an emotion (using a label encoder). Each Convolution block has a 1D Convolution layer followed by a dropout layer and a Max-pooling layer. The Loss function used in this paper is the Categorical Cross Entropy and the Optimization algorithm used is Adaptive Moment Estimation (Adam) which is an improvised version of the traditional gradient descent.

The confusion matrix was tabulated over validation data here. The principal diagonal has a majority of samples which indicates that the model has higher validation accuracy.

Table 3 : Confusion Matrix of the best model over validation data

	Angry	Fear	Happy	Neutral	Sad	Surprised
Angry	126 (15.5%)	2 (0.2%)	1 (0.1%)	1 (0.1%)	1 (0.1%)	0 (0.0%)
Fear	5 (0.6%)	115 (14.1%)	1 (0.1%)	3 (0.4%)	6 (0.7%)	0 (0.0%)
Happy	9 (1.1%)	11 (1.4%)	104 (12.8%)	1 (0.1%)	1 (0.1%)	4 (0.5%)
Neutral	1 (0.1%)	1 (0.1%)	2 (0.2%)	152 (18.7%)	4 (0.5%)	2 (0.2%)
Sad	0	8 (1.0%)	2 (0.2%)	6 (0.7%)	117 (14.4%)	0 (0.0%)
Surprised	2 (0.2%)	3 (0.4%)	4 (0.5%)	0 (0.0%)	1 (0.1%)	120 (14.7%)

The development part of this paper was totally focused on optimization in terms of validation accuracy and time. But it is also important to keep the "relevance" metrics in check. The development process also retained a weighted validation F1 score of 0.9010 which shows a sign of consistency with the produced validation accuracy.

In real cases, a DNN is used to substitute the role of a feature extractor in a machine learning problem. But a DNN combined with a feature extractor is used, which is not a conventional practice. But 1D CNN captured reasonable patterns due to their application specific functionality. Although DNNs are used in this paper, we can still use other machine learning algorithms like Naïve Bayes, Hidden Markov models and Support Vector Machines. They also yield very promising results without heavy optimization but limit the performance at some stages.

Conclusion:

The usage of 1D convolution neural networks along with an MFCC feature extractor has proved to be accurate in terms of predictions and efficient in terms of computation time and memory. Since, MFCC extraction has flexibility in varying hyperparameters, the number of features needed for the neural network to capture the exact number of patterns could be found. Also, the neural network training was very fast due to its less parameter count which is contrary to the fact that it could still produce a great accuracy. The GPU acceleration has proved to be very use but, it's not reliable in real time product-based applications since graphic acceleration cannot be integrated to small scale chips with power limitations. The model although being extremely useful is limited to speech-based emotion recognition. It cannot be used in multimodal applications, but it does reduce the computational resources required when compared to multimodal emotion recognition systems. Architectures like reinforcement learning and pre trained embedding are avoided to reduce computational complexity.

The pipeline is trained using a noise free data. But real-life applications in most cases are infused with noisy speech. So, the paper pipeline can be either prefaced with a noise reduction technique and then trained or prefaced with a noise reduction technique at the stage of application deployment. Also, there are many pretrained emotion specific speech embeddings that can capture reasonable features from the speech but huge in terms of computation resources. Dimensionality reduction techniques and feature selection pipelines can be used to reduce the parameters and preserve the features.

Referrences:

1. Lalitha, S., Mudupu, A., Nandyala, B.V. and Munagala, R., 2015, December. Speech emotion recognition using DWT. In 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (pp. 1-4). IEEE.
2. Khalil, R.A., Jones, E., Babar, M.I., Jan, T., Zafar, M.H. and Alhussain, T., 2019. Speech emotion recognition using deep learning techniques: A review. IEEE Access, 7, pp.117327-117345.
3. Muda, L., Begam, M. and Elamvazuthi, I., 2010. Voice recognition algorithms using mel frequency cepstral

coefficient (MFCC) and dynamic time warping (DTW) techniques. arXiv preprint arXiv:1003.4083.

4. Livingstone, S.R. and 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), p.e0196391..
5. Pichora-Fuller, M.K. and Dupuis, K., 2020. Toronto emotional speech set (TESS). Scholars Portal Dataverse, 1.
6. Mustaqeem and Kwon, S., 2021. 1D-CNN: Speech emotion recognition system using a stacked network with dilated CNN features. CMC-COMPUTERS MATERIALS & CONTINUA, 67(3), pp.4039-4059.
7. Siriwardhana, S., Reis, A., Weerasekera, R. and Nanayakkara, S., 2020. Jointly fine- tuning" bert-like" self-supervised models to improve multimodal speech emotion recognition. arXiv preprint arXiv:2008.06682.
8. Kiranyaz, S., Ince, T., Abdeljaber, O., Avci, O. and Gabbouj, M., 2019, May. 1-D convolutional neural networks for signal processing applications. In ICASSP 2019- 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp.
9. Iliev, A.I., Scordilis, M.S., Papa, J.P. and Falcão, A.X., 2010. Spoken emotion recognition through optimum-path forest classification using glottal features. Computer Speech & Language, 24(3), pp.445-460.
10. Hossin, M. and Sulaiman, M.N., 2015. A review on evaluation metrics for data classification evaluations. International journal of data mining & knowledge management process, 5(2), p.1.