International Journal of Advancement in Electronics and Computer Engineering (IJAECE)
Volume 13, Issue 5, May. 2024, pp.101-107, ISSN 2278 -1412

Ensemble Learning-Based Speech Emotion Recognition for Children with Autism Spectrum Disorder: A Solution for Emotion Identification

[1]*Rajat Tiwari,* [2]*Dr. Sneha Soni*
[1]*M.tech Scholar,* [2]*Professor*
[1,2] *Computer Science & Engineering,*
[1,2] *Sagar Institute of Research and Technology- Excellence (M.P)*

Abstract *– This paper presents Speech emotion recognition is an important field of study aimed at developing systems capable of automatically identifying and classifying emotions from speech signals. In this research, we propose a deep learning-based approach using LSTM (Long Short-Term Memory) neural networks for speech emotion recognition. The study utilizes a dataset of speech recordings with labeled emotion annotations. The preprocessing stage involves segmenting the speech data into frames and extracting relevant acoustic features like Mel-frequency cepstral coefficients (MFCCs).*

The LSTM architecture is designed to capture temporal dependencies and patterns in the speech data. Experimental results demonstrate the effectiveness of the proposed method, achieving an accuracy of 85% in classifying emotions such as happiness, sadness, anger, and neutral. The findings indicate the potential of deep learning and LSTM models in accurately recognizing emotions from speech signals.

Further improvements and future research directions are discussed, including fine-tuning techniques and real-time deployment. The proposed method contributes to the advancement of speech emotion recognition systems, which have applications in fields such as affective computing, human-computer interaction, and healthcare.

Keywords: Autism Spectrum Disorder, Speech Emotion Recognition, Ensemble Learning, Human Emotion Identification, Social Interaction, Communication Skills.

I. INTRODUCTION

Speech Emotion Recognition (SER) is a field of study within the broader domain of affective computing that focuses on the analysis and interpretation of human emotions expressed through speech. Emotions play a crucial role in human communication, and accurately recognizing and understanding these emotions from speech signals has numerous practical applications, such as improving human-computer interaction, call center monitoring, mental health assessment, and virtual agent design [3]. SER has relied on extracting various acoustic features from speech signals, such as pitch, energy, spectral features, and prosody, and employing machine learning algorithms to classify the emotional state. However, these handcrafted feature-based approaches often require domain knowledge, are time-consuming, and may not capture the full complexity of emotional expression.

Deep Learning, a subfield of machine learning, has gained significant attention in recent years due to its ability to automatically learn high-level representations from raw data. In the context of SER, deep learning models have shown promising results by directly processing raw speech signals without relying on manual feature extraction [1], One popular approach in deep learning for SER is the use of Convolutional Neural Networks (CNNs), which can learn hierarchical representations by applying convolutional filters to local patterns in the speech signal. CNNs have demonstrated their effectiveness in capturing both low-level acoustic cues and higher-level contextual information [8].

To train deep learning models for SER, a large annotated dataset is required. These datasets typically consist of recordings of individuals expressing different emotions, along with corresponding emotion labels. The models are trained to map the raw speech signals to the corresponding emotional categories.

Once trained, the deep learning models can be used to predict emotions from unseen speech data by feeding the raw signal into the trained network. The output of the model is a probability distribution over the different emotional categories, indicating the likelihood of each emotion being present in the input speech.

A. Speech

Speech refers to the verbal expression of language through the production and articulation of sounds. It is a primary mode of communication for humans, allowing individuals to convey thoughts, ideas, information, and emotions to others.

Speech involves the coordination of various physiological processes and organs involved in

DOI Member 112.145.383

producing sounds. These include the respiratory system, which provides the necessary airflow, the larynx (voice box), where the vocal cords are located and vibrate to produce sound, and the articulatory system, including the tongue, lips, teeth, and palate, which shape and modify the airflow to produce specific speech sounds.

Speech is composed of distinct units called phonemes, which are the smallest meaningful units of sound in a particular language. Phonemes combine to form syllables, which, in turn, combine to form words, phrases, and sentences. The arrangement and combination of these units follow the rules and structures of a specific language, known as phonetics and phonology.

Speech perception refers to the process of interpreting and understanding spoken language. Listeners rely on auditory cues to recognize and differentiate between different speech sounds and patterns. They also rely on other contextual cues, such as body language, facial expressions, and intonation, to understand the meaning and intent behind the spoken words.

Speech has several characteristics that can vary across individuals, cultures, and languages. These include factors such as pitch, loudness, tempo, rhythm, intonation, and articulation. Variations in these aspects contribute to the diversity of accents, dialects, and languages spoken around the world

II. METHOD

A. Experimental Procedure

Three speech emotion corpora were collected from the internet - the Ryerson Audio-Visual Database of Emotional Speech and Songs (RAVDESS), the Toronto Emotional Speech Set (TESS), the Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D) to create the speech emotion recognition system. At first, the RAVDESS corpus was selected for training and optimizing the support vector machine (SVM), the multilayer perceptron (MLP), and the recurrent neural network (RNN) model. The RAVDESS dataset contains data from all the seven emotion classes used in this work, and there is an equal number of male and female actors. Also, the recording quality in RAVDESS is better compared to CREMA-D. Furthermore, training the models on a single dataset was faster than training them on all three. All these factors made RAVDESS an excellent first choice. However, after using the RAVDESS hyperparameter settings on the other two datasets, the results were inferior. This was because the RAVDESS dataset had very few data, and so the machine had low generalization capability when tested on other datasets. For this reason, the model tuning strategy was changed to a new strategy described below.

Three different noise samples were used in order to modify all three speech emotion corpora. This was done to train the models on speech data in the presence of noise. Most everyday conversations happen with some noise in the background. The characteristics of the background noise depend on the surrounding environment of the speakers. If the speech emotion system is only trained on clean speech data, it will not perform well in real-world applications because the system will pick up noise along with the speech signal and try to process the audio with the added noise. The audio features extracted from the audio will be misleading as they will contain components of the noise. This will eventually lead to low classification accuracies. Therefore, the models were trained and evaluated with datasets containing background noise to create a robust speech emotion recognizer. The three background noises selected are the sound of children playing in a playground, the ambiance in a shopping mall, and the sound of cars passing by on the streets. These three noise samples represent three completely different scenarios. Three different SNR values were selected for adding these noise samples to the clean speech – 0 dB, 5 dB, and 10 dB – which introduces a lot more variety to the original clean speech datasets. Different sections of the noise files were added to different clean speech files to avoid teaching the background noise's machine features during training. From RAVDESS, TESS, and CREMA-D, each clean speech file was combined with one of the three noise samples in one of the three SNRs to create a noise-added file.

Dataset naming convention followed in this research.

Name	*Description*	*Data Samples (balanced)*
RAVDESS_Clean	Original RAVDESS corpus	1,344
RAVDESS_Clean_Noise	Original RAVDESS corpus along with the noise-added versions	2,688
TESS_Clean	Original TESS corpus	2,800
TESS_Clean_Noise	Original TESS corpus along with the noise-added versions	5,600
CREMA-D_Clean	Original CREMA-D corpus	7,626
CREMA-D_Clean_Noise	Original CREMA-D corpus along with the noise-added versions	15,252
Complete_Clean	Original RAVDESS, TESS, and CREMA-D corpora	13,041
Complete_Clean_Noise	Original RAVDESS, TESS, and CREMA-D corpora along with the noise-added versions	26,082

The three machine learning algorithms (SVM, MLP, and RNN) on only the RAVDESS dataset, they were trained on a bigger dataset that contains the RAVDESS recordings (clean speech), the TESS recordings (clean

speech), and the CREMA-D recordings (clean speech), along with the noise-added versions of these clean-speech files. A special naming convention is used from this point forward to simplify referencing these different datasets. This naming convention is explained in Table. The neutral class in RAVDESS and CREMA-D had fewer data samples than the other classes, so it was resampled to match the other classes. The surprise class was missing in CREMA-D, so it was resampled when all three datasets were combined

III. Result & DISCUSSION

The *Complete_Clean* dataset is comprised of all the original clean speech utterances of the Ryerson Audio-Visual Database of Emotional Speech and Songs (RAVDESS), the Toronto Emotional Speech Set (TESS), and the Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D). The *Complete_Clean_Noise* dataset was created by adding three noise samples in three different SNR values to the *Complete_Clean* dataset. The three noise samples used for background inclusion in the samples are a recording of children playing in a playground, a recording of a shopping mall, and a recording of cars passing by on the streets. SNRs values used are 0 dB, 5 dB, and 10 dB. The noise samples were added to the clean speech utterances using MATLAB, and for each clean speech, a noise sample, an SNR value, and a specific section of the noise file were randomly picked by the MATLAB code. Since the *Complete_Clean_Noise* corpora had class imbalance, the minority classes were resampled (with replacement) to match the sample count of the majority classes. The neutral class samples were lower in both RAVDESS and CREMA-D, and the surprise class was missing from CREMA-D. The hyperparameters of all the models discussed in this section were tuned while being trained on the *Complete_Clean_Noise* dataset. The data split of 80:10:10 was used, where 80% of the dataset was used in training the models, while 10% was used for validation and the other 10% was used for testing. Each data split was *stratified*, meaning that there were equal number of data samples per emotion class in each of the three data splits (training, validation, and test).

A. Experiments with Custom Feature Set

The customization included 36 low-level audio descriptors - the Mel-frequency cepstral coefficients (MFCCs), the root-mean-square (RMS) energy, the spectral contrast, and the polynomial coefficients. Among these low-level descriptors, the MFCCs and the RMS energy were used in most of the prior speech emotion recognition related work. The other descriptors were mainly used in music classification tasks. However, they have shown to yield good classification accuracies when applied to emotion classification task in this work. A total of 62 audio features were created using the four low-level audio descriptors of the custom feature set for the SVM and MLP models. They are 26 mean values of first 26 MFCCs across all audio frames, 26 standard deviations of first 26 MFCCs across all audio frames, one mean RMS energy across all audio frames, seven mean values of spectral contrast across all audio frames, and two mean values of polynomial coefficients across all audio frames. No functionals were applied for the RNN model since the low- level descriptors extracted per frame are the *sequences* that the RNN learns from. The low- level descriptors were directly used as the audio features for the RNN.

SVM Model with Custom Feature Set

The learning curves were plotted for the SVM model trained on the *Complete_Clean_Noise* dataset using the custom feature set. Figure 1 shows the confusion matrix for this model, a summary of the results. For this model, the radial basis function (RBF) kernel was used, with C=10.0 and γ=0.01. The *Scikit-learn* library was utilized to develop the SVM model in Python.

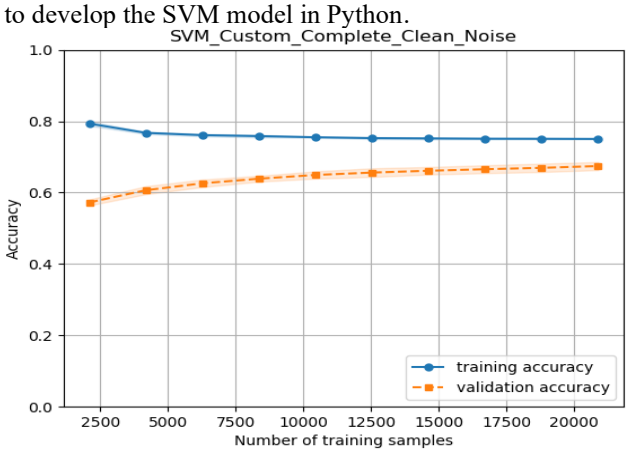

Fig 1 Learning curves for the SVM model trained on the Complete_Clean_Noise corpus, using the custom feature set.

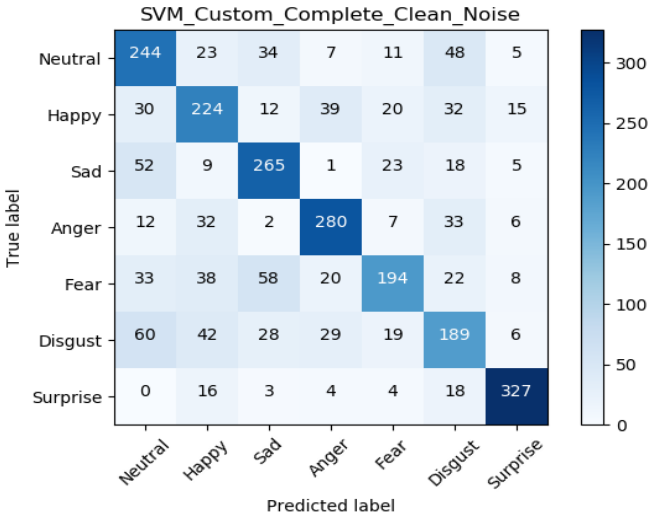

Fig 2 Confusion matrix for the SVM model trained on the Complete_Clean_Noise corpus, using the custom feature set

MLP Model with Custom Feature Set

The number of artificial neuron units used in an artificial neural network and the number of layers are hyperparameters that can be tuned for getting high accuracies. There is no golden rule for selecting the number of neurons or layers. Researchers usually experiment with these parameters and select values that provide the highest performance. A common convention among computer scientists is to use *log* BASE-2 number, like 64, 128, and 256. Another convention is to use increments of 50 or hundred, like 50, 100, and 200. The number of input-layer neurons is equal to the number of input features, and the number of output-layer neurons is equal to the number of classes in the dataset. Even though there is no rule for selecting the number of neurons in the hidden layer(s), there are some rules-of-thumb that can be followed, according to. To design the MLP architecture of this model, number of neurons, such as 10, 50, 100, 200, and 500, were selected for each layer

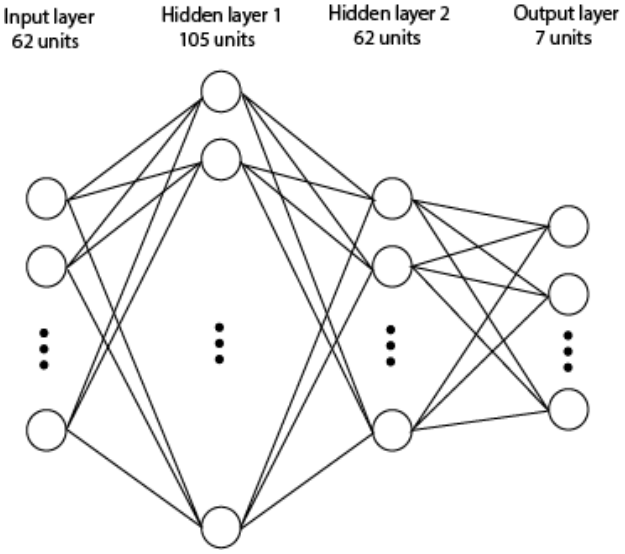

Fig 3. Architecture of the MLP used with the custom feature set

The Adam optimizer was used in order to minimize the loss function, which in this case is the categorical cross-entropy loss for the MLP model. The rectified linear unit (ReLU) activation function was used and for the output units, the Softmax activation function was used, which provides the prediction accuracies for each class for the hidden layer units. Instead of using a fixed learning rate, a learning rate scheduler was used to change the learning rate as the training progressed. An inverse time decay function was used as the learning rate schedule, with an initial learning rate of 0.01, 1000 decay steps, and a decay rate of 80%. The training, validation, and testing data were each divided into batches of size sixteen, and 50 epochs were used during training

RNN Model with Custom Feature Set

The RNN layers created using Keras requires a tensor as the input, compared to the 2D-structured inputs in MLP (batch, features). A tensor is a three-dimensional array of numbers. For RNNs, the three dimensions are the number of data samples, the number of features, and the number of time steps (batch, time steps, features). The audio frames were used as the time steps, while the features were the low-level descriptors extracted per frame to process the sequential data. For this reason, all the low-level descriptors extracted using the custom feature set were used as the audio features. In this case, each low-level descriptor value is extracted for each audio frame, which forms a sequence of
data suitable to be processed by an RNN. The audio frames represent the time steps of the input data. Meaning, once the current audio frame has been processed, with all the low- level descriptors extracted and fed to the network, the next audio frame is processed. Using a sampling rate of 16 kHz and a frame length of 512 samples (32 ms), around 150 audio frames were processed per audio file. Figure 4 shows the architecture of the RNN model. The LSTM cells are represented by a *recurrent edge* on the units

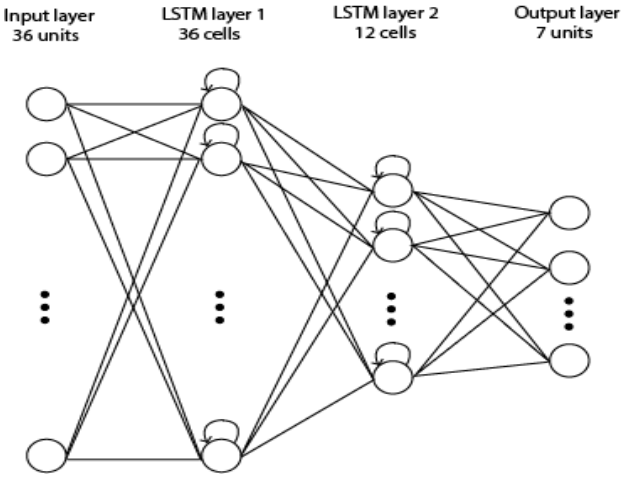

Fig 4. Architecture of the RNN used with the custom feature set

The Adam optimizer was used along with the categorical cross-entropy loss function. The hyperbolic tangent (tanh) function was the activation function for the LSTM cells, and the sigmoid (σ) function was the recurrent activation function. The Softmax function was used as the activation for the output units. The inverse time decay function is the learning rate scheduler, with 0.01 as the initial learning rate, 1,000 as the decay steps, and 80% as the decay rate. A batch size of sixteen was used, and the total number of epochs used during training was 50. A 30% dropout was placed between the two LSTM layers and a 30% dropout between the second LSTM layer and the output. A 20% recurrent dropout was placed for the LSTM cells in the first layer. Figure 5 shows the accuracy curves of the RNN model, and Figure 7 shows the loss curves. The confusion matrix for this model is shown in Figure 8.

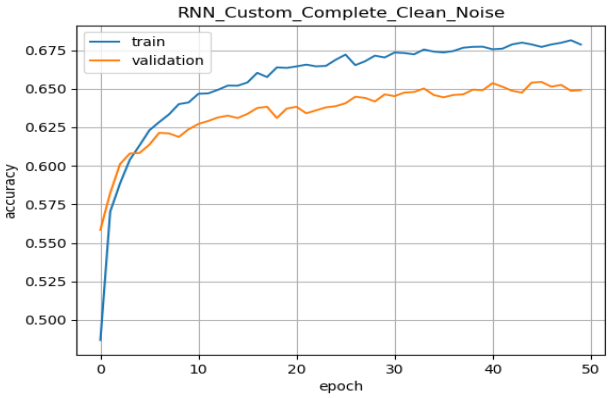

Fig 6 Accuracy curves for the RNN model trained on the Complete_Clean_Noise corpus, using the custom feature set

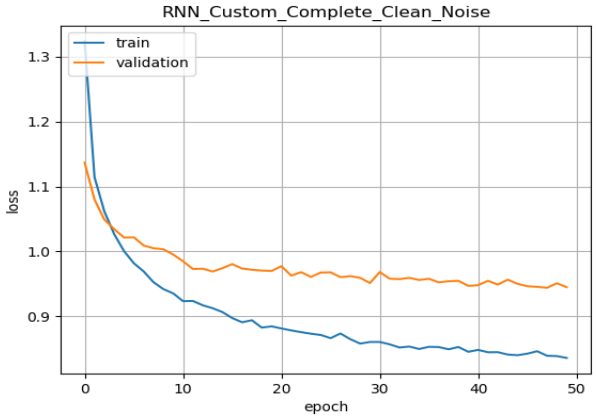

Fig 7 Loss curves for the RNN model trained on the Complete_Clean_Noise corpus, using the custom feature set

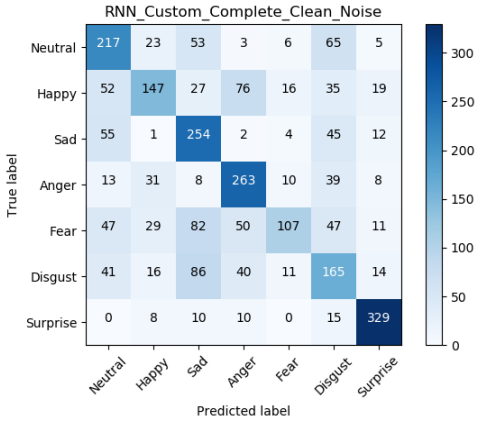

Fig 8 Confusion matrix for the RNN model trained on the Complete_Clean_Noise corpus, using the custom feature set

Comparing Models from Both Feature Sets

The performance metrics for the models created using the custom feature set and the models created using the partial GeMAPS feature set. The models listed in this table were trained and evaluated on the Complete_Clean_Noise dataset. From Table, it can be seen that the custom feature set models have outperformed the partial GeMAPS feature set in all metrics. This can be attributed to the significantly lower number of features in the partial GeMAPS feature set since the lower number of features could not capture the variations in the training data. The model that showed the best performance among all the models studied in this work was the MLP model trained on the Complete_Clean_Noise dataset using the custom feature set. It showed the highest classification accuracies and good average precision and average recall scores as well. Figure 8 shows the results of stratified 10-fold cross-validation for the models on the Complete_Clean_Noise dataset. After separating the test set from the training set, the training set was split into ten equal parts or folds. The model evaluation was performed ten times, and each time one out of the ten parts was used as the validation set while the remaining nine parts were used for the test set. Each time, a different fold was selected for validation split, and the training and validation accuracy were calculated for the ten experiments. The accuracy scores shown in the bar plot of Figure 9 were calculated by computing the mean of all the classification accuracies over the ten experiments.

Comparing the models created using the two different feature sets, for the Complete_Clean_Noise corpus

Classifier	*Feature Set*	*Training %*	*Valid. %*	*Test %*	*Precision*	*Recall*
SVM	Custom	75.0 %	65.2 %	66.1 %	66.4 %	66.1 %
SVM	P.GeMAPS	68.0 %	58.7 %	58.7 %	58.2 %	58.7 %
MLP	Custom	68.1 %	65.9 %	65.7 %	83.3 %	50.3 %
MLP	P.GeMAPS	58.3 %	58.2 %	57.9 %	79.5%	38.2 %
RNN	Custom	67.9 %	64.9 %	63.7 %	75.4 %	53.7 %
RNN	P.GeMAPS	60.2 %	59.9 %	57.9 %	75.7 %	41.9 %

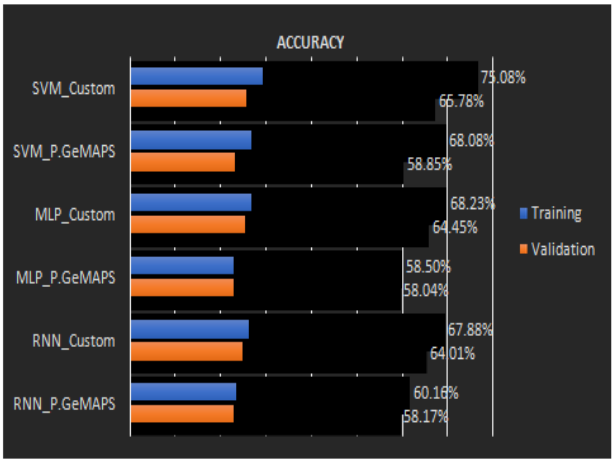

Fig 9 fold cross-validation results on the Complete_Clean_Noise corpus for the models listed on the vertical axis

When comparing the confusion matrices for the models trained on the Complete_Clean_Noise dataset, it can be seen that the surprise emotion was the class that was most accurately predicted. This can be due to the fact that the surprise class was missing from the CREMA-D dataset, and when the three datasets were joined to create the Complete_Clean_Noise dataset, this class was heavily resampled from RAVDESS and TESS. Also, the neutral class samples were lower in the RAVDESS and CREMA-D datasets, so it was also resampled when all three datasets were joined. However, resampling was done after the training, validation, and test samples were separated, which prevented the repetition of minority data samples in the three splits. Besides surprise and neutral emotions, the top two most accurately classified emotions were sad emotions and anger. For the RAVDESS dataset, the surprise emotion was the most accurately predicted class. For the CREMA-D dataset, the anger emotion was the most accurately predicted class. For TESS, the neutral emotion was the most accurately predicted class. For all models, the two most challenging emotions to classify were the happy and the fear emotion.

IV. CONCLUSION

This paper has focused on a speech emotion recognition solution was created for helping children with autism spectrum disorder (ASD) identify emotions in social interactions. Children with ASD have difficulty identifying emotional cues in social interactions. The objective was to develop a tool that could help these children better detect emotions when conversing with people around them. The speech emotion recognizer was developed in Python using ensemble learning, a technique used to combine multiple machine learning algorithms to get a more accurate prediction.

Three machine learning algorithms were used – a support vector machine (SVM), a multilayer perceptron (MLP), and a recurrent neural network (RNN). The datasets used to train these algorithms include the Ryerson Audio-Visual Database of Emotional Speech and Songs (RAVDESS), the Toronto Emotional Speech Set (TESS), the Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D), and the noise-added versions of the three datasets. Three noise samples were selected - a noise file containing a recording of children playing in the background, a noise file containing a recording of shopping mall ambiance, and a noise file containing a recording of cars passing by on the streets. Each clean speech utterance was added to one of the three types of noise files using MATLAB, in one of three SNR values – 0 dB, 5 dB, or 10 dB.

Two separate speech emotion recognition models were developed – one to be used indoors and the other to be used outdoors. The model created for indoor use was trained on only clean speech data from all three datasets, and the model created to be used outdoors was trained on the final dataset, which included clean speech and noise-added files from all three datasets. This was done so that if the speech emotion model was implemented on a mobile application, users could select the model they want to use based on their environment. Finally, a multimodal emotion classifier was created by joining the speech emotion recognition model with a facial expression recognition model. This produced four emotion recognition classifiers – three speech emotion recognition classifiers and one facial expression recognition classifier. The Python program was written so that if predictions from the four classifiers are unique, the facial expression recognition solution's prediction would be used, as it had better classification accuracy than the speech emotion recognition models

References

[1]. Fatma M. Talaat, Zainab H. Ali, Reham R. Mostafa & Nora El-Rashidy. (2024). Real-time facial emotion recognition model based on kernel autoencoder and convolutional neural network for autism children.

[2]. Ayşe Tuna (2023) “Recognising and Expressing Emotions: Difficulties of Children with Autism Spectrum Disorder in Learning a Foreign Language and How to Resolve Them” c e p s Journal | Vol.13 | No 4 |

[3]. Rodolfo Pavez, Jaime Diaz, Jeferson Arango Lopez, Danay Ahumada, Carolina Mendez-Sandoval & Fernando Moreira (2023) “Emo-mirror: a proposal to support emotion recognition in children with autism spectrum disorders” Volume 35, pages 7913–7924.

[4]. Yuri Matveev, Anton Matveev, Olga Frolova, Elena Lyakso and Nersisson Ruban (2022). Automatic Speech Emotion Recognition of Younger School Age Children. Volume 10, Issue 14.

[5]. Mohammad Ariff Rashidan , Shahrul Na’im Sidek Hazlina Md. Yusof Madihah Khalid ,Ahmad Aidil Arafat Dzulkarnain, Aimi Shazwani Ghazali,Afiqah Mohd Zabidi , And Faizanah Abdul Alim Sidique (2021) “Technology-Assisted Emotion Recognition for Autism Spectrum Disorder (ASD) Children” Volume 9.

[6]. Uzma Abid Siddiqui, Farman Ullah, Asif Iqbal, Ajmal Khan, Rehmat Ullah, Sheroz Paracha, Hassan Shahzad and Kyung-Sup Kwak (2021) “Wearable-Sensors-Based Platform for Gesture Recognition of Autism Spectrum Disorder Children Using Machine Learning Algorithms” Volume 21 Issue 10.

[7]. O. Korn, L. Stamm, and G. Moeckel, (2017) “Designing authentic emotions for non-human characters. A study evaluating virtual affective behavior”, Designing Interactive Systems, pp. 477-487, Jun

[8]. P. Ekman And W. V. Friesen, (1971) "Constants Across Cultures In The Face And Emotion", Journal Of Personality And Social Psychology, Vol. 17, No. 2, Pp. 124–129.

[9]. S. Schelinski and K. V. Kriegstein ,(2018) "The relation between vocal pitch and vocal emotion recognition abilities in people with autism spectrum disorder and typical development," Journal of Autism and Developmental Disorders, vol. 49, pp. 68-82, Jul.

[10]. S. Sadhu, R. Li, and H. Hermansky (2019), "M-vectors: sub-band based energy modulation features for multi-stream automatic speech recognition," 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6545- 6549, May.

[11]. W. Liu et al,(Aug 2019) "State-time-alignment phone clustering based language-independent phone recognizer front-end for phonotactic language recognition," 14th Int. Conf. on Computer Science & Education, pp. 863-867.

[12]. G. Shanmugasundaram, S. Yazhini, E. Hemapratha, and S. Nithya,(Mar 2019) "A comprehensive review on stress detection techniques," International Conference on System, Computation, Automation and Networking, pp. 1-6.

[13]. A. K. Oryina and A. O. Adedolapo, "Emotion recognition for user centred e- learning, (2016)" 40th Annual International Computer Software and Applications Conference, vol. 2, pp. 509-514,.

[14]. N. Kurpukdee, S. Kasuriya, V. Chunwijitra, C. Wutiwiwatchai and P. Lamsrichan, (2017) "A study of support vector machines for emotional speech recognition," 2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES), pp. 1-6,.

[15]. A. Meftah, Y. Alotaibi and S. Selouani,(2016) "Emotional speech recognition: A multilingual perspective," 2016 International Conference on Bio-engineering for Smart Technologies (BioSMART), pp. 1-4,.

[16]. G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller and S. Zafeiriou,(2016) "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5200-5204,.

[17]. B. Schuller, E. Marchi, S. Baron-Cohen, A. Lassalle, H. O'Reilly, D. Pigat, P. Robinson, I. Davies, T. Baltrusaitis and M. Mahmoud,(2015) "Recent developments and results of ASC-Inclusion: An Integrated Internet-Based Environment for Social Inclusion of Children with Autism Spectrum Conditions," IDGEI, (No pagination provided).

[18]. How do i calculated the number of overlapping frames an given audio file has?" (2020) [Online] Available: https://math.stackexchange.com/questions/2249977/how-do-i- compute-thenumber-of-overlapping-frames-an-given-audio-file-has.

[19]. S. B. Davis and P. Mermelstein,(1980) "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", IEEE Transactions on Acoustic, Speech and Signal Processing, vol. 28, pp. 357–366,.

[20]. M. Farrús, J. Hernando and P. Ejarque,(2007) "Jitter and shimmer measurements for speaker recognition," 8th Annual Conference of the International Speech Communication Association, Interspeech, vol. 2, pp. 778-781,.

[21]. D. Jiang, L. Lu, H. Zhang, J. Tao and L. Cai, (2002) "Music type classification by spectral contrast feature," Proceedings. IEEE International Conference on Multimedia and Expo, vol. 1, pp. 113-116.

[22]. O. Agcaoglu, B. Santhanam and M. Hayat, (2013) "Improved spectrograms using the discrete Fractional Fourier transform," IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), pp. 80-85,.

[23]. A. A. Bashit, (2019)"A comprehensive solar powered remote monitoring and identification of Houston Toad call automatic recognizing device system design", Master's Thesis, Engineering, Texas State University, San Marcos, TX, USA.

[24]. A. A. Bashit and D. Valles,(2018) "A mel-filterbank and MFCC-based neural network approach to train the Houston Toad call detection system design," 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 438-443.