Journal of Theoretical and Applied Information Technology
15th July 2024. Vol.102. No. 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SMOTE-2DCNN FOR ENHANCING SPEECH EMOTION RECOGNITION

NURUL NADHRAH KAMARUZAMAN[1], NOR AZURA HUSIN[2], NORWATI MUSTAPHA[3], RAZALI YAAKOB[4], MUHAMMAD MUDASSIR EJAZ[5]

[1234]Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia

[5]Universiti Teknologi Petronas Malaysia

E-mail: [1]nurul.nadhrah0111@gmail.com, [2]n_azura@upm.edu.my, [3]norwati@upm.edu.my, [4]razaliy@upm.edu.my, [5]mudassir@empirepixel.com

ABSTRACT

Speech emotion recognition (SER) is a specialized form of audio classification that aims to identify and classify emotional states expressed from spoken language or speech signals. In this study, the main objective is to propose an accurate audio classification model for the SER. This study primarily focuses on two key issues: the insufficient training data within each available dataset and the imbalanced distribution of data, both of which contribute to overfitting and negatively impact the accuracy of the audio classification model. Henceforth, we present the SMOTE-2DCNN, which is a combination of the Synthetic Minority Oversampling Technique (SMOTE) with a 2-Dimensional Convolutional Neural Network (2DCNN), designed to effectively address imbalanced data distributions and achieve accurate emotion classification. Our proposed SMOTE-2DCNN demonstrates outstanding performance with a UA rate of 81% and a WA rate of 80%. This represents a substantial enhancement, achieving approximately 15% higher accuracy compared to the leading state-of-the-art method.
Keywords: *Speech Emotion Recognition, Audio Classification, Deep Learning, SMOTE, Imbalanced Data*

1. INTRODUCTION

Verbal communication is naturally the most effective and efficient way of normal human interaction. This fact has led many researchers to believe that using speech signals to interact between humans and computers is a rapid and efficient method. As a result, Speech Emotion Recognition (SER) has played a significant role in the field of Human-Computer Interaction (HCI) as it is necessary for a computer to understand human emotions in interacting. For more than two decades, SER has become mostly advantageous for many applications that need interaction between human and computer such as calling centre conversation, online tutoring, medical analysis and many more [1], [2].

The main goal of SER is to automatically detect a speaker's emotional state from the tone of their voice. In other words, SER can be defined as a series of methods that analyse and classify speech signals to determine emotions that are hidden within them. As a part of audio classification, SER focuses specifically on classifying emotional states or sentiments expressed in speech. It typically involves the extraction of various acoustic features, such as pitch, intensity, and spectral characteristics, as well as linguistic features like word choice and prosody (intonation, rhythm, and tempo). Early approaches emphasized manually extracting features and using standard methods like Gaussian Mixture Models (GMM), Dynamic Time Warping (DTW), and Hidden Markov Models (HMM) [3]. In the deep learning era, methods like Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) have been widely implemented and achieved great performance.

The emergence of the deep learning model has filled the limitation of traditional handcrafted methods in extracting the high-level features from

multi modalities including audio which is indeed the primary modality in SER [4]. One of the advantages of deep learning is that, it is able to search for the best features to use for the purpose of recognition and classification. [5], [6]. Despite many researchers in the field having succeeded in improving the accuracy of classification by combining multiple architectures of deep learning such as CNN and RNN, the confusion and misclassification of emotional category is still occurring (Poria et al., 2016; Xu et al., 2020). This is mainly due to the fact that these methods are prone to overfitting and require a large amount of training data Therefore, a high-quality of annotated emotional speech database is a necessary to get a successful classification and for the SER to work accurately [10].

However, the main problem encountered in the SER field is the lack of labelled training data in each available dataset, which is especially necessary for effectively implementing deep learning techniques [11]. In addition, imbalance in data distribution such as uneven number of classes is also one of the contributing factors. With such a limited and unbalanced number of data, these issues have become crucial as the imbalance class would dramatically skew the performance of the classifier thus leading to the difficulty of achieving good results in classification. This consequence happened as it could exhibit bias towards the majority class and ignore the minority class altogether. As a result, the deep learning model will be overfit and the optimal result might not be achieved for the classification [12].

In this study, our main objective is to propose an accurate classification model to recognize emotion from humans' speech. This study primarily focusses on two key issues which are the insufficient training labelled data and the imbalanced data distribution. We exclude other modalities involved in SER such as text and facial expression and concentrate solely on audio classification as audio is the primary modality involved in SER. We introduce the SMOTE-2DCNN model which includes the Synthetic Minority Oversampling Technique (SMOTE) method with 2-Dimensional Convolutional Neural Network (2DCNN). Our proposed model was designed with two key modules which are imbalance handling and classification. By implementing SMOTE method, the main duty of the imbalance handling module is to resample the training dataset in order to reduce the bias brought on by the insufficient training labelled data and the imbalanced data distribution from the original dataset. Finally, in the classification module, we integrate 2DCNN into our development model.

The motivation behind incorporating SMOTE and 2DCNN in our research for enhancing audio classification in the context of speech emotion recognition is rooted in addressing the challenges posed by insufficient training labelled data and the imbalanced data distribution for SER. SMOTE is an oversampling technique used to balance the distribution of classes in a dataset by creating new synthetic data for the minority class. It creates synthetic samples by interpolating between existing instances of the minority class. As the synthetic samples are created based on the characteristics of the minority class, this helps to preserve the information present in the original dataset. Other oversampling methods, despite increasing sample volume, have the disadvantage of not providing any additional information or variance to the learning model. As a result, they may not be as effective in preserving the diversity of the minority class [13]. In addition, SMOTE balances class distributions well, avoiding overfitting and bias in models. Its algorithm-agnostic nature makes it compatible with a wide range of machine learning algorithms. SMOTE is also good at handling sparse, disjointed areas of the minority class in feature space.

Apart from that, the use of 2DCNN is motivated by the simplicity of the 2DCNN architecture which enhances to the efficacy of our approach. Unlike traditional methods that may require complex feature engineering, 2DCNNs operate directly on spectrogram representations of audio, capturing both spectral and temporal features simultaneously. This simplicity not only simplifies the model development process but also facilitates better performance, as the network automatically learns hierarchical features required for recognizing nuanced emotional patterns in speech. The 2DCNN's ability to exploit local patterns and hierarchical representations in the spectrogram contributes to its effectiveness in audio classification tasks, making it a powerful yet straightforward tool for enhancing speech emotion recognition[14]. Therefore, by synergistically incorporating SMOTE and 2DCNN, we hope to provide a robust and effective solution for improving the accuracy and reliability of speech emotion recognition systems in real-world applications

Our main contribution lies in addressing aforementioned issues which are the insufficient training labelled data and the imbalanced data

distribution, ultimately leading to a significant improvement in classification accuracy and mitigating the risk of overfitting. We believe that the proposed model must meet these two essential criteria in order to develop an accurate and effective audio classification for SER: (1) a simple classification model with effective imbalance class handling module, (2) a model with a compatible classifier and imbalance class handler. In this study, our proposed SMOTE-2DCNN model has successfully met both of the above criteria.

The remaining sections of this paper are organized as follows. Section 2 summarizes the related works done by previous researchers in the same area. Section 3 presents the detailed explanation of the methodology of our proposed method. Section 4 describes the evaluation of the [15] presents the result of the evaluation the proposed SMOTE-2DCNN method. Finally, Section 6 presents the discussion and conclusion, involving the limitation and future works of this study.

2. RELATED WORK

A wide range of classifiers for SER have been examined by researchers to improve and achieve better accuracy. In the traditional machine learning approach for SER, standard classifiers like GMM, HMM, ANN and K-NN were implemented after the extraction of features from the speech signal [2]. Since 2013, the use of DL methods for emotion recognition has gradually risen [16]. These days, the majority of SER models proposed by researchers employed DL methods and have achieved higher outcomes in terms of average accuracy and computational cost [11] [14].

According to the trend from year to year, the majority of the researchers have implemented CNN, followed by LSTM, and RNN DL approaches. In 2014, studies from [19], [20] implemented CNN with spectrograms in their work. The RNN classifier also has been employed by [21] in 2015 and achieved 81% of accuracy on the RECOLA database. The Deep Belief Network (DBN) classifier evaluated on the CASEC database produced the best accuracy of 94.60% in 2017 by [22] and the following year, [23] have achieved 92.71% of accuracy using DCNN on EMODB datasets. In 2020, the EMODB dataset was once again used by [24] with CNN and achieved 95% of accuracy. Each year, SER sees a rise in the use of hybrid approaches for example research from [25] in 2019 implement 2DCNN + LSTM with extracted features from spectrograms using EMODB and IEMOCAP dataset with 95.89% and 89.16% of accuracy, respectively.

The impact of environmental noise on deep learning models in audio classification for SER systems can be crucial which can potentially lead to the loss of fine-grained information [15]. Getting access to a large and well-annotated dataset is one of the significant obstacles in the development of an effective audio recognition system. Insufficient audio data makes deep neural network training very challenging because large amounts of training data are necessary for effective training and evaluation of audio systems[17]. Furthermore, a lack of data can lead to an imbalanced dataset as certain classes or categories may be underrepresented or insufficiently sampled, resulting in a skewed distribution that limits the model's ability to learn patterns from minority classes [18] .This imbalance can jeopardize the model's performance, leading to biased predictions and reduced accuracy for the minority classes.

The data level approach [26] , the cost sensitive approach [27], and the algorithm level approach [28] are the three main strategies that are generally used to address the class imbalance problem. The most common paradigm for managing imbalanced data is the data level approach. By applying data pre-processing prior to classification, data level algorithms, also known as over-sampling are typically used to increase the number of minority class samples [29]. On the other hand, under-sampling occurs when certain samples from the majority class are left out of the data [30]. One of the main advantages of the data level approach is its generality which allows it to be used with any classifier [31].

Many efforts have been made in the past to handle imbalanced data more effectively. A well-known Naïve Bayes classifier is primarily used in the under-sampling technique suggested in the study[32]. An innovative three-dimensional framework consisting of a discriminator, generator, and classifier in addition to decision boundary regularization was implemented in study [33]. Training a generator in conjunction with a classifier is the most noteworthy element of the suggested approach. In order to gradually remove samples from the majority class of imbalanced data, Xie et al. [34] proposed a novel under sampling technique that makes use of consecutive density peaks. Three distinct approaches, primarily based on genetic algorithms that automatically determine sample ratios for oversampling, under sampling, and hybrid

sampling techniques, were proposed in the study [35]. Two novel density-based techniques, density-based under sampling (DB_US) and density-based hybrid sampling (DB_HS), were used in the study [36] to completely eliminate the overlap between the majority class and the minority class in an unbalanced dataset and produce a balanced and normalized class distribution.

In this study, we primarily explore the Synthetic Minority Over-sampling Technique SMOTE method because of its popularity and competitive performance. SMOTE was created by [37] and is one of the most widely used over-sampling techniques. Using linear interpolation between a minority class point, the SMOTE method creates synthetic data. SMOTE is an effective over-sampling technique that has been extensively used in a number of previous studies. A novel oversampling algorithm based on the widely used SMOTE method was proposed by [38] for deep learning models. The study [39] used a novel hybrid method called CDSMOTE, which uses class decomposition and oversampling on the minority class samples to reduce the domination of the majority class samples. As a result of maintaining the majority class samples, this suggested method produces more balanced data than general under sampling algorithms. A new development of SMOTE was proposed in the research study [40] by combining it with the Kalman filter to filter out noisy samples from the resulting dataset that simultaneously includes the original data and the synthetically added samples. In order to address the noise issue, the study [40] used a novel oversampling algorithm called IR-SMOTE. The noise in minority class clusters is removed by sorting the majority class samples and using the k-means clustering algorithm. The number of synthetic samples is then suitably assigned to each cluster using the kernel density estimation method.

The aforementioned SMOTE pitfalls have one drawback of being sensitive to the classifier selection. Since SMOTE generates synthetic examples to balance class distribution, its effectiveness is influenced by the classifier's ability to handle these newly created instances. Certain classifiers may not be well-suited to accommodate the synthetic samples, potentially impacting overall classification performance.

3. METHODOLOGY

The implementation of SMOTE-2DCNN is illustrated in Figure 1. We implemented an experimental research design in this study. The research unfolds through systematic steps: (1) Preprocessing involves data cleaning and feature extraction from audio signals, (2) Application of SMOTE to augment the minority class instances, (3) Construction of the 2DCNN model architecture tailored for speech emotion recognition, (4) Training the model using the augmented dataset, and (5) Evaluation and validation using appropriate metrics to assess the model's performance.

The first step involves preprocessing the audio data to select the best form of input set features to fit in with the model. In order to handle imbalanced data distribution, the SMOTE has been employed to generate synthetic samples to augment minority class instances. Subsequently, the augmented dataset is fed into the customized 2DCNN model architecture, which is designed for more robust emotion recognition. The experimental design will include training and evaluating the SMOTE-2DCNN model on the augmented dataset, comparing its performance with baseline models, and conducting statistical analyses to validate the effectiveness of the proposed approach in mitigating overfitting and enhancing the accuracy of audio classification for speech emotion classification.

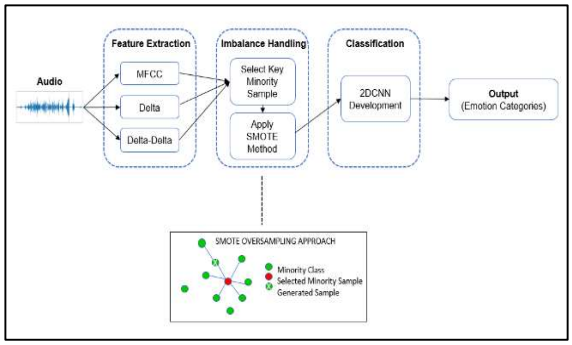

Figure 1: Illustration of SMOTE-2DCNN Implementation

3.1 Audio Pre-Processing

This section describes in detail on audio data pre-processing which is a crucial step before training and testing the model. The main challenge of this part is to find the top-notch features of audio data that are appropriate for the emotion recognition task. We implement a deep learning approach for the development of our classification model. For that reason, we choses to incorporate the unstructured audio representation of MFCC features. This MFCC feature has the ability to extract the patterns on its

own since the feature extraction process is done automatically before it will then be fed directly into the deep learning-based model.

However, the MFCC feature vector only describes the power spectral envelope of a single frame, but speech appears to have information in the dynamics such as the trajectories of the MFCC coefficients over time. Consequently, we have included delta (differential) and delta-delta (acceleration) features as feature inputs in this study. The purpose of using delta and delta-delta coefficients is based on the idea that better speech recognition requires an understanding on how the coefficients change over time. The recognition performance was shown to be significantly improved by calculating the MFCC trajectories and appending them to the original feature vector.

3.2 SMOTE-2DCNN Detailed Description

This section describes in detail on our proposed SMOTE-2DCNN model for audio classification of SER. The SMOTE-2DCNN is designed with two key modules which are imbalance handling and classification. The main duty of the imbalance handling module is to resample the training dataset in order to reduce the bias brought on by the imbalance in the original dataset to the experimental results. For that, we proposed to incorporate the SMOTE method into our model development in order to carry out the task effectively. Finally, in the classification module, we integrate 2DCNN into our development model.

A 2DCNN is a type of neural network that is used to process data in two-dimensional arrays such as images and data with height and width. The network applies a series of filters to the input which is also known as kernels or weights. Each filter is a small 2D array with a height and width that are often smaller than the input like 3 X 3 or 5 X 5 in size.

3.2.1 Imbalance Handling Module with SMOTE

In order to generate new synthetic data, SMOTE employs the k-nearest neighbor technique. The deep insight of the SMOTE algorithm works in three steps. The first step starts by setting a minority class with set A_{minor}. For each $x \in A_{minor}$, the k-nearest neighbors of x are determined by calculating the Euclidean distance between x and each other element in set A_{minor}. The second step involves the construction of set A_1 by selecting N samples (i.e., $x_1, x_2,...x_N$) at random from each $x \in A_{minor}$'s k-nearest neighbours. The third step which is the final step, the following formula was used to generate a new synthetic example:

For each $x_k \in A_1$ ($k=1,2,3,...N$),

$$x' = x + rand(0,1) * |x - x_k| \quad (1)$$

Each of the aforementioned steps should be repeated until both the minority and majority classes are distributed equally.

3.2.2 Classification Module With 2DCNN

In this study, the 2DCNN will receive the 2-dimensional input involving time and features of MFCC, Delta and Delta-Delta. The convolution process of all the inputs can be expressed by this formula equation:

$$y = x * h(n_1, n_2) = \sum_{k1=\infty}^{\infty} \sum_{k2=\infty}^{\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2) \quad (2)$$

where, $h\ (k_1, k_2)$ represents the filter and $x(k_1, k_2)$ indicates the input vector of this convolutional layer. The filter $h(k_1, k_2)$ is first turned into $h(-k_1, -k_2)$ and translated by n_1 and n_2 which causes the filter $h(k_1, k_2)$ to eventually become filter $h(n_1 - k_1, n_2 - k_2)$. This is what the negative sign in the filter $h(n_1 - k_1, n_2 - k_2)$ means. Finally, multiply the input $x(k_1, k_2)$ with the result to obtain the value $y(n_1, n_2)$.

Each convolutional layer is followed by the addition of an activation function to enhance the model's interpretation ability and non-linearity. We selected the rectified linear unit (ReLU) as the activation function in order to avoid the vanishing gradient issue. The ReLU function can be seen in the following equation:

$$f(x) = \{x, 0, \quad x > 0 \ or \ otherwise \quad (3)$$

Each convolution block ends with the addition of a max-pooling layer. By obtaining the maximum value of the input vector within the predetermined range, the max-pooling layer eliminates redundant information from the input vector and extracts significant features. In our case, we applied a 2 x 2 max pooling filter and left the stride length at its default value of 2, which is equal to the max pooling filter size. We also included batch normalization

between the layers to normalize the inputs of the layers in order to speed up and stabilize the network training. Batch Normalization can be calculated by applying the following mathematical formulas:

$$\mu_B \leftarrow \frac{1}{3}\sum_{i=1}^{m} x_i \tag{4}$$

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{m}^{i=1}(x_i - \mu_B)^2 \tag{5}$$

where the mean and variance of batch data input are represented by μ_B and σ^2_B. Each dimension of input is then separately normalized with the implementation of following formula equation:
For $x = (x_1 \ldots x_d)$,

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{6}$$

For numerical stability, a tiny constant ε is added to the denominator in order to avoid the occurrence of dividing by value 0. By scaling and shifting the regularized value data $\hat{X}_i$, it allows the batch normalization to reinstate the power of the network. This process can be expressed by the following formula equation:

$$y_i \leftarrow \gamma\hat{x}_i + \beta \tag{7}$$

where ɣ and **β** are the parameters that later will be learned during the optimization process.

As the network gets deeper, a flatten layer is applied to combine all layers into one single layer, transform each dimensional array into a single lengthy continuous linear vector and provide them as inputs to the subsequent fully connected layers, also known as dense layers. A couple of fully connected layers have been added to wrap up the classification model. By multiplying the input vector by the weights matrix and then adding the bias vector, the fully connected layers are in charge of creating a linear transformation to the input vector. The product is then subjected to a non-linear transformation using the non-linear activation function *f* which is ReLU. The process is repeated for each fully connected layer and the calculation of this process can be expressed by the following equation:

$$y = f(Wx + b) \tag{8}$$

where, *x* is input vector, *W* is weight vector, *b* is bias and *f* is activation function.

We included a dropout layer as the regularization technique to prevent overfitting of the model. Following the fully connected layers, the last layer employs the softmax activation function in order to determine the probability that the input belongs to a particular class. Below is the softmax activation function equation:

$$softmax\,(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{9}$$

Apart from that, we employ categorical cross-entropy as loss function for training model in which can be expressed by the following equation:

$$Loss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log_2(\hat{y}_{i,j}) \tag{10}$$

where, *N* denotes number of samples, *C* is the number of classes. y_i is the one-hot encoded vector representing the true class label of the input sample in the form of 0s 1s and $\hat{y}_{i,j}$ is a vector representing the predicted probabilities for each class.

3.3 SMOTE-2DCNN Model Architecture

This section describes in detail about the SMOTE-2DCNN model architecture. Following is the architecture of our SMOTE-2DCNN network:

- **CONV 1:** The input size of the first convolutional layer is 16 x 8 x 1. This layer implements 64 kernels with spatial size of 3 x 3. The activation function used is the Leaky Rectified Linear Unit (ReLU) with the padding type "same". Followed by a 2 x 2 stride step max-pooling function with Batch Normalization.

- **CONV 2:** The second layer implements 64 kernels with spatial size of 3 x 3. The activation function used is the Leaky Rectified Linear Unit (ReLU) with the padding type "same". Followed by a 2 x 2 stride step max-pooling function with Batch Normalization.

- **CONV 3:** The third layer implements 64 kernels with spatial size of 3 x 3. The activation function used is the Leaky Rectified Linear Unit (ReLU) with the padding type "same". Followed by a 2 x 2 stride step max-pooling function.

- **DENSE 1:** The first dense layer sets 64 units with the implementation of Leaky Rectified Linear Unit (ReLU) as the activation function. This layer uses the kernel regularizer and bias regularizer that applies the L2 regularization penalty with the default value used is l2=0.01. This layer also includes the dropout with the rate of 0.5.

- **DENSE 2:** The last dense layer sets 4 units with the implementation of Softmax as the activation function.

Figure 2 presents the proposed 2DCNN model architectures.

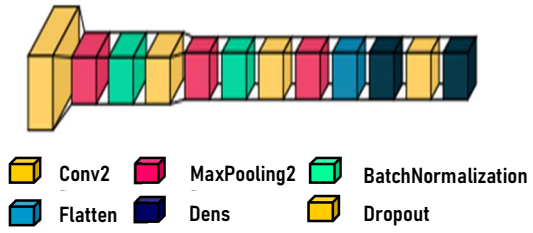

Figure 2: Proposed SMOTE-2DCNN Model Architectures

3.4 SMOTE-2DCNN Model Algorithm

The proposed SMOTE-2DCNN model set of steps is outlined in Algorithm 1. Let D be the set of data, where: $D = \{(x_1, y_1), \ldots, (x_A, y_A)\}$. Here, (x_A, y_A), represents the feature set and label of audio data A. For $x \in A_{minor}$, we determine the k-nearest neighbour of x and using Equation 1, we generate the synthetic samples A_1, to balance the class distribution. We then split the dataset into training and testing sets with an 80:20 ratio.

Next, we build our 2DCNN model. We define the model architecture, add convolutional layers, max pooling layers, flatten the output, y of the convolutional layers, add dense layers and apply softmax activation function appropriate to the classification task. We compile the model with appropriate loss and optimizer and train the network using backpropagation and gradient descent on a training set of labelled data. All the hyperparameters optimization involved are described in Section 4.4.2. Lastly, we evaluate the performance of the model on the testing set and the predicted class will be the final output. Algorithm 1 presents the algorithm for SMOTE-2DCNN.

4.0 EVALUATIONS AND RESULTS

In this section, the evaluation procedure for the SMOTE-2DCNN model and its outcome are thoroughly discussed. We used the audio IEMOCAP dataset to evaluate our proposed SMOTE-2DCNN [41]. To maintain consistency with previous research, we specifically selected the emotions of angry, happy, sad, and neutral from among all the categorical emotion labels that were annotated in the IEMOCAP dataset. We first defined the 80:20 using the "train_test_split()" function to split the dataset for training and testing model. In this instance, 80% of the dataset is designated as the training dataset, and 20% as the test dataset. We implemented stratified k-fold cross-validation by setting the stratify parameter to ensure that the distribution of

Algorithm 1: SMOTE-2DCNN Model	
Input:	D = $\{(x_1, y_1), \ldots, (x_A, y_A)\}$ A_{minor} = Minority class instances, k = Number of nearest neighbors to use, N = Number of synthetic samples to generate
Output:	A_1 = Synthetic minority class samples $FinalOutput_{Test}$ = Label class predicted on the provided testing sample $Test$ acc_{M1} = Percentage accuracy of M_1 fusion model M_1 represent model 1 which is the proposed SMOTE_2DCNN.
1) Split D into training and testing	
2) for each sample x in A_{minor}:	Compute the k- nearest neighbors of x in A_{minor}, excluding x itself.

3) for k = 1 to N	a) Choose a random neighbor x_k from the k nearest neighbors of x. b) Generate a new synthetic sample x' by following the Equation 1. c) Add the new synthetic sample x' sample to Synthetic minority class samples, A_1 d) Return A_1
4) Define 2DCNN Model architecture	
5) Initialize weights and biases randomly	
6) Set hyperparameters; learning_rate, num_epochs, batch_size, optimizer	
7) Train model: for epoch= 1 to num_epochs	a) Loop over batches of data b) Get batch of input data and corresponding labels c)Following Equation 2, 3, 4, 5, 6, 7, 8, 9, 10 perform forward pass d) End for
8) Perform backward pass: for layer in reversed layers (starting from last layer moving backwards)	a) Compute gradients of output w.r.t. input b) Compute gradients of loss w.r.t. layer weights and bias c) Update layer weights and bias using gradients and optimizer d) Update gradients with respect to output of previous layer e) End for
9) Compute the output of proposed SMOTE-2DCNN model using testing data Ti, $i = 1…nc$ (nc represents number of class which in this case = 4) to get the predicted label class, $FinalOutput_{Test}$	
10) Calculate the metrics such as accuracy, precision, recall, and F1 score	
11) Get the final percentage accuracy of proposed SMOTE-2DCNN model, acc_{M1}	
12) End	

target classes are consistent across folds, which helps to reduce bias and improve the accuracy of our model. Table 1 shows the details of the dataset division after splitting.

Table 1: Details of Dataset Division (Audio)

IEMOCAP (Audio) *(Involving 4 label class; angry, happy, sad, and neutral)*	
Number of Samples for Training	5465
Number of Samples for Testing	1367

4.1 Evaluation Performance Measure

The performance of the model will be measured based on accuracy, precision, recall, F1-score, and confusion matrix. Accuracy has been extensively accepted as a benchmark for assessing the effectiveness of emotion recognition systems based on audio speech. In addition, Precision, Recall and F1- score has been also widely used to evaluate performance with respect to how reliable the model is in classifying an uneven class distribution. We therefore compute and compare the accuracy, precision, recall, and F1-Score to evaluate the performance of our proposed model and to fairly compare it with several current state-of-the-art audio speech emotion recognition models using the same above-mentioned measures.

4.2 Hyperparameter Optimization for SMOTE-2DCNN

Prior to conducting any experiment, the optimum hyperparameters used in our proposed SMOTE-2DCNN is necessary to be identified in advance. The process of hyperparameter tuning in SMOTE-2DCNN is crucial that can significantly impact the performance of the model. Therefore, the aim of hyperparameter tuning is to identify the best set of hyperparameters that will maximize the model's accuracy on the testing set.

Theoretical analysis and mathematical equations are insufficient for determining the ideal hyperparameter settings, which vary depending on the specific environment and cannot be predetermined. Consequently, even when utilizing the same algorithm, the actual performance may differ across different cases. To address this challenge, the optimal hyperparameter settings for our proposed SMOTE-2DCNN are determined through a trial-and-error approach.

We consider multiple essential hyperparameters when tuning our proposed SMOTE-2DCNN. Table 2 summarizes the best tuned hyperparameter values for our proposed SMOTE-2DCNN model

Table 2: Hyperparameters Tuning Value (SMOTE-2DCNN)

Used Hyperparameter	Values
Number of Epochs	100
Batch Size	64
Learning Rate	0.01
Kernel	3
Dropout Rate	0.5
Optimizer	Adam
Regularized	kernel regularizer = L2 (0.01) bias regularizer = L2 (0.01)

4.3 Experimental Set Up

In this section, we have developed a thorough experimental set up consisting of three experimental designs to assess the efficacy of our proposed SMOTE-2DCNN model.

The first experimental design involves training our designed 2DCNN model with two different sets of feature input. The first feature input set, we calculate the MFCC, Delta, Delta-Delta features and take the mean of them in order to reduce the size of features. Meanwhile, in the second feature input set, we consider the entire arrays of MFCC, Delta, Delta-Delta features without taking the mean only.

The second experimental design involves comparing the performance of our designed 2DCNN with and without SMOTE approach. We train our 2DCNN model with the best outcome from the previous experiment as our input feature set and make a comparison between the two results.

The third experimental design involves comparing the performance of our proposed SMOTE-2DCNN method with other state-of-the-art methods in the audio classification model for emotion recognition. The state-of-the-art methods involved have also been trained using the same dataset from IEMOCAP audio data.

5.0 RESULT

This section presents the experiment results and findings of our proposed SMOTE-2DCNN on the aforementioned sets experimental design.

5.1 Experiment 1: Comparison Between Two Feature Input Sets

We conducted the first experiment to verify the best form of input set feature. By comparing different forms of input feature sets will contribute more performance improvement to our proposed model.

Table 3: Comparison of Different Form of Input Feature Set

Input Feature Set	Accuracy (%)
Mean of MFCC, Delta, Delta-Delta	70
Entire Arrays of MFCC, Delta, Delta-Delta	75

Table 3 shows the experiment results on a comparison of different input feature sets. Based on the result, by using the entire arrays of MFCC, Delta, Delta-Delta, the 2DCNN model achieves higher accuracy with 75% accuracy, which outperforms the result of using the mean of MFCC, Delta, Delta-Delta with 70% accuracy.

5.2 Experiment 2: Comparison Between with SMOTE and without SMOTE

The input feature set of entire arrays of MFCC, Delta, Delta-Delta was picked for the final experiment with our SMOTE-2DCNN model because of the higher accuracy result achieved in the previous experiment.

To show the robustness of our proposed method, we compared the result of our designed 2DCNN with SMOTE and without SMOTE. Table 4 shows the percentage accuracy of our designed 2DCNN with and without SMOTE. Figure 3 illustrates the accuracy per epoch and loss per epoch of our designed 2DCNN with SMOTE and Figure 4 illustrates the accuracy per epoch and loss per epoch of the 2DCNN model without SMOTE.

Table 4: Percentage Accuracy of 2DCNN with and without SMOTE

Method	Accuracy (%)
With SMOTE	80
Without SMOTE	75

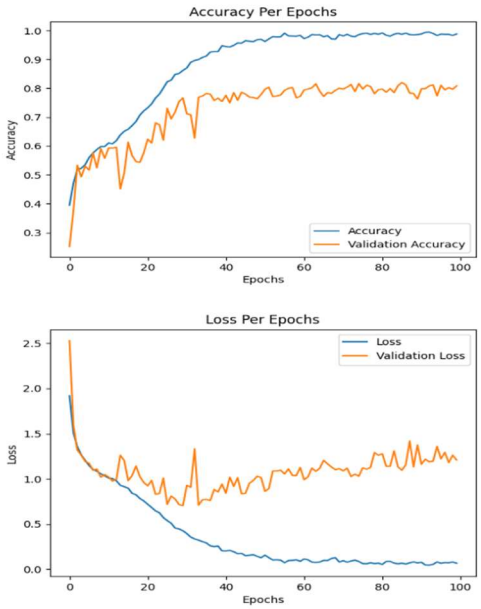

Figure 3: Accuracy Per Epoch and Loss Per Epoch of 2DCNN with SMOTE

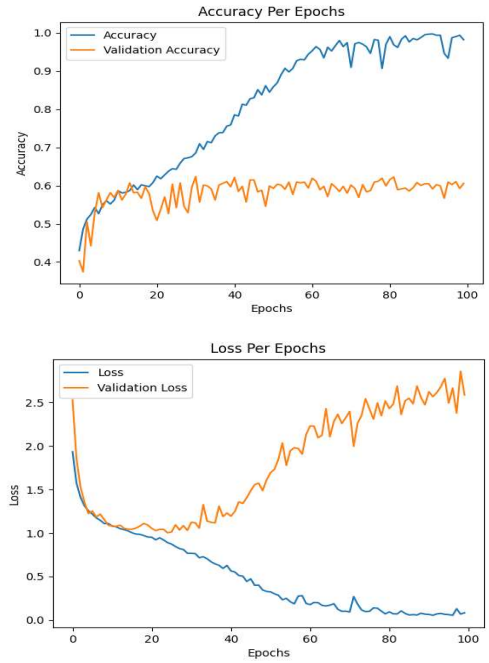

Figure 4: Accuracy Per Epoch and Loss Per Epoch of 2DCNN without SMOTE

As illustrated in Figure 5, our proposed SMOTE-2DCNN achieves 80% of accuracy with loss = 1.3. This accuracy is 5% significantly higher than the accuracy achieved by 2DCNN model without including the SMOTE method. Moreover, based on the confusion matrix illustrated in Figure 8, our model also achieves high True Positive Rate in most of the emotions such as angry, happy, sad with their corresponding TPR = 82%, 90%, 85%, whereas the model performs slightly less effectively on neutral emotion with TPR = 71%.

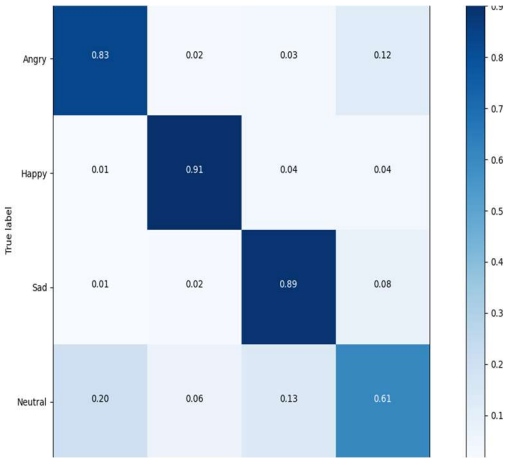

Figure 5 : Confusion Matrix of SMOTE-2DCNN

Table 5 shows the result of precision, recall and F1 score of each emotion label; angry, happiness, sadness and neutral. Among all, happy emotion scores highest precision with 89%. Same goes for recall and F1-scores, happy emotion scores the highest with 91% and 90%, respectively.

Table 5: Precision, Recall and F1-Score of SMOTE-2DCNN

Emotion	Precision (%)	Recall (%)	F1-Score (%)
Angry	78	83	81
Happy	**89**	**91**	**90**
Sad	82	89	85
Neutral	72	61	66

5.3 Experiment 3: Comparison with State-of-The- Art

Table 6 and Figure 6 present the comparison of percentage accuracy with the graphical representation for state-of-the-art methods and our proposed SMOTE-2DCNN in audio classification model for emotion recognition. We considered the un-weighted accuracy (UA) and weighted accuracy (WA) in the record of the percentage accuracy results in order to make it consistent with the prior studies of the state-of-the-art methods. As presented in Table 6, our proposed SMOTE-2DCNN yields the highest percentage of UA and WA of 81% and 80%, respectively. It shows

a significant improvement in approximately 15% of the accuracy achieved by method that performed the best result among other state-of-the-art.

Table 6: Comparison SMOTE-2DCNN with State-Of-The-Art

Method	UA (%)	WA (%)
BiLSTM +Attn [42]	51.2	55.6
BLSTM + SelfAttn [43]	76.8	76.6
LSTM + Attn [44]	57.4	63.4
CNN + LSTM [45]	59.4	68.0
2TransformerEncoder [46]	57.1	63.6
SMOTE-2DCNN	81.0	80.0

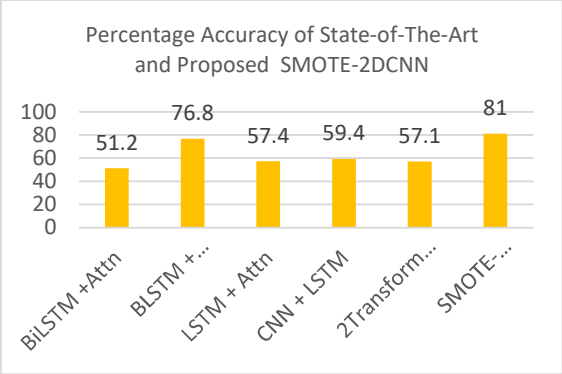

Figure 6: Percentage Accuracy of State-of-The-Art and Proposed SMOTE-2DCNN

6.0 DISCUSSION AND CONCLUSION

Our proposed SMOTE-2DCNN achieves significantly better performance compared to typical 2DCNN and several current state-of-the-art audio speech emotion recognition models (refer Table 6). The reason of this outstanding achievement is because of several factors which are discussed below:

i. *Simple Model with Effective Imbalance Class Handling Module:* Our SMOTE-2DCNN implement Synthetic Minority Oversampling Technique method in order to handle imbalance class issue in IEMOCAP dataset. Unlike typical random oversampling technique, the SMOTE algorithm creates artificial samples based on the feature space similarities rather than using data space similarities between actual minority samples. Additionally, SMOTE uses k-nearest neighbors to choose which synthetic samples to generate thus helps to ensure that the synthetic samples are similar to the current minority class samples. This approach effectively assists in improving the overfitting issue imposed by typical random oversampling thus, enhance classification accuracy and reduces bias toward majority classes.

ii. *Compatibility on the Classifier and Imbalance Class Handler:* Our SMOTE-2DCNN is composed of two key modules: Imbalance Handling Module with SMOTE and Classification Module with 2DCNN. The combination of both modules is appropriate and compatible when it is capable in handling extreme imbalance class distribution and being robust to small variance sample size of data. Besides that, SMOTE is a reliable imbalance class handler which is suitable for enhancing a deep learning classifier of 2DCNN model, deployed specifically for emotion recognition tasks. The effectiveness of this combination lies not only in the successful handling of imbalanced data but also in the synergy between SMOTE and the classifier. The compatibility between SMOTE and 2DCNN is paramount for achieving optimal results. In other word, the 2DCNN model benefits from the enhanced representation of minority class instances provided by SMOTE, resulting in improved classification accuracy and robustness.

In conclusion, our proposed SMOTE-2DCNN model has proven to be a valuable strategy for addressing both the issues of insufficient and imbalanced data distribution in the context of audio classification, specifically in the domain of SER. Furthermore, it is crucial to acknowledge that the efficacy of SMOTE hinges on the classifier's adaptability to the synthetic instances added into the training set. As highlighted earlier, certain classifiers may encounter difficulties in accommodating these artificially generated instances, potentially leading to suboptimal performance in classification tasks. Therefore, the choice of a classifier plays a pivotal role in determining the overall success of the SMOTE implementation.

In this case, the implementation of 2DCNN is inherently capable of leveraging the augmented

dataset. Indeed, adapting to the synthetic instance and generalizing well to unseen data is essential for achieving heightened accuracy in speech emotion recognition. By addressing data imbalance and compatibility issues, researchers can pave the way for more effective and accurate models in audio classification tasks.

Our future work will focus on combining SMOTE with other techniques for data augmentation and ensemble learning. Hybrid models that leverage the strengths of multiple techniques could be a promising direction as the strategy may provide improved generalization and resilience to class imbalance.

ACKNOWLEDGEMENT

This paper is part of a research funded by the Universiti Putra Malaysia Grant (GP-IPS), under Project Vote Number: 9682000

REFERENCES:

[1] B. W. Schuller, "Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends," *Commun ACM*, vol. 61, no. 5, pp. 90–99, 2018.

[2] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognit*, vol. 44, no. 3, pp. 572–587, 2011.

[3] S. Karpagavalli and E. Chandra, "A review on automatic speech recognition architecture and approaches," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 4, pp. 393–404, 2016.

[4] S. Poria, E. Cambria, and A. Gelbukh, "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 2539–2544.

[5] P. Tzirakis, J. Zhang, and B. W. Schuller, "End-to-end speech emotion recognition using deep neural networks," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2018, pp. 5089–5093.

[6] T. Zhang, W. Zheng, Z. Cui, Y. Zong, and Y. Li, "Spatial–temporal recurrent neural network for emotion recognition," *IEEE Trans Cybern*, vol. 49, no. 3, pp. 839–847, 2018.

[7] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, "Convolutional MKL based multimodal emotion recognition and sentiment analysis," in *2016 IEEE 16th international conference on data mining (ICDM)*, IEEE, 2016, pp. 439–448.

[8] G. Xu, W. Li, and J. Liu, "A social emotion classification approach using multi-model fusion," *Future Generation Computer Systems*, vol. 102, pp. 347–356, 2020.

[9] R. Kosti, J. Alvarez, A. Recasens, and A. Lapedriza, "Context based emotion recognition using emotic dataset," *IEEE Trans Pattern Anal Mach Intell*, 2019.

[10] D. Ververidis and C. Kotropoulos, "Emotional speech recognition: Resources, features, and methods," *Speech Commun*, vol. 48, no. 9, pp. 1162–1181, 2006.

[11] J. Boigne, B. Liyanage, and T. Östrem, "Recognizing more emotions with less data using self-supervised transfer learning," *arXiv preprint arXiv:2011.05585*, 2020.

[12] K. Ghosh, C. Bellinger, R. Corizzo, P. Branco, B. Krawczyk, and N. Japkowicz, "The class imbalance problem in deep learning," *Mach Learn*, pp. 1–57, 2022.

[13] R. Dubey, J. Zhou, Y. Wang, P. M. Thompson, J. Ye, and A. D. N. Initiative, "Analysis of sampling techniques for imbalanced data: An n= 648 ADNI study," *Neuroimage*, vol. 87, pp. 220–241, 2014.

[14] H.-C. Chu, Y.-L. Zhang, and H.-C. Chiang, "A CNN Sound Classification Mechanism Using Data Augmentation," *Sensors*, vol. 23, no. 15, p. 6972, 2023.

[15] J. Meyer, L. Dentel, and F. Meunier, "Speech recognition in natural background noise," *PLoS One*, vol. 8, no. 11, p. e79279, 2013.

[16] Y. B. Singh and S. Goel, "A systematic literature review of speech emotion recognition approaches," *Neurocomputing*, vol. 492, pp. 245–263, 2022.

[17] E. Tsalera, A. Papadakis, and M. Samarakou, "Comparison of pre-trained CNNs for audio classification using transfer learning," *Journal of Sensor and Actuator Networks*, vol. 10, no. 4, p. 72, 2021.

[18] O. O. Abayomi-Alli, R. Damaševičius, A. Qazi, M. Adedoyin-Olowe, and S. Misra,

"Data augmentation and deep learning methods in sound classification: A systematic review," *Electronics (Basel)*, vol. 11, no. 22, p. 3795, 2022.

[19] Z. Huang, M. Dong, Q. Mao, and Y. Zhan, "Speech emotion recognition using CNN," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 801–804.

[20] Q. Mao, M. Dong, Z. Huang, and Y. Zhan, "Learning salient features for speech emotion recognition using convolutional neural networks," *IEEE Trans Multimedia*, vol. 16, no. 8, pp. 2203–2213, 2014.

[21] S. Chen and Q. Jin, "Multi-modal dimensional emotion recognition using recurrent neural networks," in *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, 2015, pp. 49–56.

[22] W. Zhang *et al.*, "Deep learning and SVM-based emotion recognition from Chinese speech for smart affective services," *Softw Pract Exp*, vol. 47, no. 8, pp. 1127–1138, 2017.

[23] J. Zhao, X. Mao, and L. Chen, "Learning deep features to recognise speech emotion using merged deep CNN," *IET Signal Processing*, vol. 12, no. 6, pp. 713–721, 2018.

[24] T. Anvarjon, Mustaqeem, and S. Kwon, "Deep-net: A lightweight CNN-based speech emotion recognition system using deep frequency features," *Sensors*, vol. 20, no. 18, p. 5212, 2020.

[25] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," *Biomed Signal Process Control*, vol. 47, pp. 312–323, 2019.

[26] A. Guzmán-Ponce, J. S. Sánchez, R. M. Valdovinos, and J. R. Marcial-Romero, "DBIG-US: A two-stage under-sampling algorithm to face the class imbalance problem," *Expert Syst Appl*, vol. 168, p. 114301, 2021.

[27] D. Devi, S. K. Biswas, and B. Purkayastha, "Correlation-based oversampling aided cost sensitive ensemble learning technique for treatment of class imbalance," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 34, no. 1, pp. 143–174, 2022.

[28] M. A. Ganaie, M. Tanveer, and A. D. N. Initiative, "Fuzzy least squares projection twin support vector machines for class imbalance learning," *Appl Soft Comput*, vol. 113, p. 107933, 2021.

[29] M. Koziarski, C. Bellinger, and M. Woźniak, "RB-CCR: Radial-Based Combined Cleaning and Resampling algorithm for imbalanced data classification," *Mach Learn*, vol. 110, pp. 3059–3093, 2021.

[30] V. K. Chennuru and S. R. Timmapareddy, "Simulated annealing based undersampling (SAUS): A hybrid multi-objective optimization method to tackle class imbalance," *Applied Intelligence*, vol. 52, no. 2, pp. 2092–2110, 2022.

[31] D. Elreedy, A. F. Atiya, and F. Kamalov, "A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning," *Mach Learn*, pp. 1–21, 2023.

[32] C. K. Aridas, S. Karlos, V. G. Kanas, N. Fazakis, and S. B. Kotsiantis, "Uncertainty based under-sampling for learning naive bayes classifiers under imbalanced data sets," *IEEE Access*, vol. 8, pp. 2122–2133, 2019.

[33] H.-S. Choi, D. Jung, S. Kim, and S. Yoon, "Imbalanced data classification via cooperative interaction between classifier and generator," *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 8, pp. 3343–3356, 2021.

[34] X. Xie, H. Liu, S. Zeng, L. Lin, and W. Li, "A novel progressively undersampling method based on the density peaks sequence for imbalanced data," *Knowl Based Syst*, vol. 213, p. 106689, 2021.

[35] M. Zheng *et al.*, "An automatic sampling ratio detection method based on genetic algorithm for imbalanced data classification," *Knowl Based Syst*, vol. 216, p. 106800, 2021.

[36] S. Mayabadi and H. Saadatfar, "Two density-based sampling approaches for imbalanced and overlapping data," *Knowl Based Syst*, vol. 241, p. 108217, 2022.

[37] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[38] D. Dablain, B. Krawczyk, and N. V Chawla, "DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data," *IEEE Trans Neural Netw Learn Syst*, 2022.

[39] E. Elyan, C. F. Moreno-Garcia, and C. Jayne, "CDSMOTE: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification," *Neural Comput Appl*, vol. 33, pp. 2839–2851, 2021.

[40] G. S. Thejas, Y. Hariprasad, S. S. Iyengar, N. R. Sunitha, P. Badrinath, and S. Chennupati, "An extension of Synthetic Minority Oversampling Technique based on Kalman filter for imbalanced datasets," *Machine Learning with Applications*, vol. 8, p. 100267, 2022.

[41] C. Busso *et al.*, "IEMOCAP: Interactive emotional dyadic motion capture database," *Lang Resour Eval*, vol. 42, pp. 335–359, 2008.

[42] S. Tripathi, S. Tripathi, and H. Beigi, "MULTI-MODAL EMOTION RECOGNITION ON IEMOCAP WITH NEURAL NETWORKS.," *arXiv preprint arXiv:1804.05788*.

[43] J. Santoso, T. Yamada, K. Ishizuka, T. Hashimoto, and S. Makino, "Speech Emotion Recognition Based on Self-Attention Weight Correction for Acoustic and Text Features," *IEEE Access*, vol. 10, pp. 115732–115743, 2022.

[44] H. Xu, H. Zhang, K. Han, Y. Wang, Y. Peng, and X. Li, "Learning alignment for multimodal emotion recognition from speech," *arXiv preprint arXiv:1909.05645*, 2019.

[45] A. Satt, S. Rozenberg, and R. Hoory, "Efficient emotion recognition from speech using deep learning on spectrograms.," in *Interspeech*, 2017, pp. 1089–1093.

[46] J. Zhang, L. Xing, Z. Tan, H. Wang, and K. Wang, "Multi-head attention fusion networks for multi-modal speech emotion recognition," *Comput Ind Eng*, vol. 168, p. 108078, 2022.