IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 May 2023, accepted 13 June 2023, date of publication 16 June 2023, date of current version 23 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3287093

RESEARCH ARTICLE

On the Effect of Log-Mel Spectrogram Parameter Tuning for Deep Learning-Based Speech Emotion Recognition

AZAMAT MUKHAMEDIYA[1], (Graduate Student Member, IEEE), SIAMAC FAZLI[2], AND AMIN ZOLLANVARI[1], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, School of Engineering and Digital Sciences, Nazarbayev University, Astana 010000, Kazakhstan
[2]Department of Computer Science, School of Engineering and Digital Sciences, Nazarbayev University, Astana 010000, Kazakhstan

Corresponding author: Amin Zollanvari (amin.zollanvari@nu.edu.kz)

This work was supported by the Nazarbayev University Faculty Development Competitive Research Grant under Award 021220FD1151 and Award 20122022FD4120.

ABSTRACT Speech emotion recognition (SER) has become a major area of investigation in human-computer interaction. Conventionally, SER is formulated as a classification problem that follows a common methodology: (i) extracting features from speech signals; and (ii) constructing an emotion classifier using extracted features. With the advent of deep learning, however, the former stage is integrated into the latter. That is to say, deep neural networks (DNNs), which are trained using log-Mel spectrograms (LMS) of audio waveforms, extract discriminative features from LMS. A critical issue, and one that is often overlooked, is that this procedure is done without relating the choice of LMS parameters to the performance of the trained DNN classifiers. It is commonplace in SER studies that practitioners assume some "usual" values for these parameters and devote major efforts to training and comparing various DNN architectures. In contrast with this common approach, in this work we choose a single lightweight pre-trained architecture, namely, SqueezeNet, and shift our main effort into tuning LMS parameters. Our empirical results using three publicly available SER datasets show that: (i) parameters of LMS can considerably affect the performance of DNNs; and (ii) by tuning LMS parameters, highly competitive classification performance can be achieved. In particular, treating LMS parameters as hyperparameters and tuning them led to $\sim$ 23%, $\sim$ 10%, and $\sim$ 11% improvement in contrast with the use of "usual" values of LMS parameters in EmoDB, IEMOCAP, and SAVEE datasets, respectively.

INDEX TERMS Log-Mel spectrogram, speech emotion recognition, SqueezeNet.

I. INTRODUCTION

Speech emotion recognition (SER) can be employed for accurate decoding of a speaker's physical and psychological conditions and, therefore, has the potential to play an important role within human-computer interfaces [1], [2], [3], [4]. Empowered by algorithmic and computational advances in recent years, deep neural networks (DNNs) have achieved breakthrough results in various areas [5], [6], [7], [8], [9], [10], and found eminent applications in SER [11], [12], [13]. It is commonplace in SER to train DNN classifiers with a log-Mel Spectrogram (LMS) generated from recorded audio signals. A critical issue, and one that is often overlooked, is that to generate LMS, one assumes some "usual" fixed values of LMS parameters such as the number of Mel-filter banks, the window and the hop length and then apply DNNs—for instance, see [11], [12], [13], [14], [15], and [16] to cite just a few articles.

From a machine learning perspective, LMS parameters should be treated as hyperparameters and require proper tuning. That being said, it may be argued that one can computationally benefit by fixing values of these parameters and, therefore, skipping hyperparameter tuning, if performances of trained DNNs are only marginally changed by varying these hyperparameters. However, whether these parameters

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

have negligible or, on the contrary, a considerable impact on the performance of DNN classifiers has been left unexplored so far. This is problematic because a great deal of research is focused on comparing various DNN architectures without concern for the impact of these parameters on the DNN performance.

In this paper, we are primarily concerned with the impact of LMS parameters on the performance of DNN classifiers. In this regard, and in contrast with earlier studies, we fix our choice of DNN classifier and vary the parameters of LMS. This practice allows us to focus our experiments on tuning parameters of LMS. In terms of DNN classifier, we use an architecture known as SqueezeNet [17]. In comparison with other pre-trained convolutional neural networks (CNNs), SqueezeNet has far fewer parameters; for example, compared to VGG16 [18], AlexNet [19], ResNet18 [20], and Inception [21], it is a lighter architecture by a factor of 115, 50, 9, and 5, respectively. This is especially attractive when considering deploying a DNN classifier on resource-limited embedded devices used in SER.

The main contribution of this paper is twofold: (i) we show that LMS parameters can considerably affect the performance of DNNs; and (ii) we empirically demonstrate that by tuning LMS parameters, highly competitive classification performance can be achieved.

The organization of this paper is as follows. In Section II, the literature review is presented. In Section III, the background material and the utilized model are introduced. In Section IV, the experimental setup and results are given. Sections V and VI provide the discussion and conclusion of the paper, respectively.

II. LITERATURE REVIEW

Human-centered signal processing research has placed considerable emphasis on SER, employing LMS-derived audio features to create precise models for emotion recognition. While early approaches focused on more classical machine learning-based approaches, a shift towards deep neural network-based models is evident [11], [12], [13], [14], [15], [16].

Meng et al. [11] proposed to combine a CNN with attention-based bidirectional long short-term memory (BiLSTM) for SER. LMS was computed using fixed parameters such as 40 Mel-filter banks, a temporal window length of 25ms, and a hop length of 10ms over 3s long segmented audio frames. This study extracted local features from LMS utilizing the proposed CNN architecture, while attention-based BiLSTM was used to learn long-term contextual dependencies from the extracted features.

Chen et al. [12] proposed an attention-based convolutional recurrent neural network (RNN) architecture and employed similar LMS parameters as in [11]. The convolutional RNN was used to extract high-level features from LMS, whereas the attention layer was utilized to learn relevant feature representations.

Jiang et al. [14] proposed a parallelized convolutional RNN architecture for SER. LMS features were computed from entire audio waveforms. Similar to other approaches also here LMS was calculated with fixed parameters, namely 64 Mel-filter banks, a window length of 25ms, and a hop length of 10ms. The proposed parallelized architecture was designed to capture temporal-frequency correlations and simultaneously learn from variable length frame-level features.

Fan et al. [13] proposed the individual standardization network (ISNet), which is based on a CNN and a multi-layer perceptron. They standardize individual emotional representations using their CNN architecture aimed at improving the performance of SER systems. To obtain the spectrogram from raw audio signals, they used 128 Mel-filter banks with a window length of 1024 and a hop length of 512.

Maji et al. [15] proposed a model based on a convolutional "capsule" network (Conv-CapNet), which they combined with bidirectional gated recurrent units (BiGRU) to efficiently extract discriminant emotional features from speech signals with variable lengths. Also here LMS was computed with fixed parameters: 64 Mel-filter banks, window and hop lengths of 512.

Tang et al. [16] proposed a dilated CNN-based architecture. This study aimed at constructing a model which learns long temporal dependencies in speech utterances. They derived LMS using 40 Mel-filter banks, a window length of 40ms, and a hop length of 20ms over 3s long segmented audio frames.

The primary focus of the aforementioned studies was the construction of a reasonable architecture to increase the accuracy of SER. Interestingly, all above-mentioned studies used predefined values for LMS-specific parameters. In this paper, we study the impact of parameters of LMS on performance of DNNs for SER.

III. BACKGROUND

A. LOG-MEL SPECTROGRAM

One way to capture spectral information (i.e., spectrogram) from an audio sample is to compute the Short Time Fourier Transform (STFT). Given a recorded signal $x[n]$ with length T, $X_a[k]$, which is the STFT coefficient for the k^{th} frequency bin and the a^{th} time-frame, is calculated as [22]

$$\underset{0\le k\le N-1}{X_a[k]} = \sum_{n=0}^{N-1} x[n]\cdot w[n-aH]\cdot e^{-j2\pi kn/N}, \qquad (1)$$

where $w[n]$ is a window function with length L (e.g., Hamming window), H is the hop size, and N is the total number of DFT points (frequency bins).

The obtained spectrogram is then log-transformed and Mel scaled. The leverage of the Mel scale is inspired by its relation to human sound perception [23]. Specifically, it is often assumed that the Mel scale is linear up to 1000 Hz, and logarithmic above. To convert the spectrogram into Mel scale, the computed spectrogram passes through Mel-filter banks. The result, which is known as the *Mel spectrogram*, groups multiple STFT frequency bins into one Mel bin. In particular,

$LMS_a[m]$, which is the log-Mel spectrogram coefficient for the a^{th} frame and the m^{th} Mel-filter bank, is given by

$$\underset{0\le m\le M-1}{LMS_a[m]} = \log\left[\sum_{k=0}^{N-1} H_m[k]\cdot|X_a[k]|^2\right], \tag{2}$$

where $H_m[k]$ is the k^{th} coefficient for the m^{th} filter bank [24],

$$H_m[k] = \begin{cases} \frac{k-f[m-1]}{f[m]-f[m-1]} & f[m-1]\le k\le f[m] \\ \frac{f[m+1]-k}{f[m+1]-f[m]} & f[m]<k\le f[m+1] \\ 0 & otherwise \end{cases}, \tag{3}$$

$f[m]$ is the Mel bin point given by [24],

$$f[m] = \left\lfloor \frac{N+1}{F_s}\mathrm{B}^{-1}\left(f^l_{mel} + m\frac{f^h_{mel}-f^l_{mel}}{M+1}\right)\right\rfloor, \tag{4}$$

F_s is the sampling rate, M is the number of Mel-filter banks, f^h_{mel} and f^l_{mel} are highest and lowest frequencies of the filter banks in Mel, and B^{-1} is an operator that converts Mel into Hertz [23], [24]:

$$\mathrm{B}^{-1}(f) = (700\cdot(10^{f_{mel}/2595}-1)),\ (\mathrm{Hz}). \tag{5}$$

B. SqueezeNet

In an effort to achieve a comparable level of performance as in AlexNet but with a lighter architecture, Iandola et al. [17] constructed an architecture, namely, SqueezeNet, that is 50 times lighter than AlexNet. In comparison with other pre-trained architectures [18], [19], [20], [21], a lighter architecture not only leads to potentially faster inference, but can also facilitate usage on memory-limited embedded devices. Two types of SqueezeNet have been proposed: v1.0 and v1.1. Here we use v1.1, which has slightly fewer parameters in comparison with v1.0. SqueezeNet v1.1 consists of eight "Fire" modules that are built based on the following strategies [17]: (i) replacing 3×3 kernels with 1×1, hence reducing the number of parameters by a factor of 9 for each convolutional filter; and (ii) "squeezing" the number of input channels using a 1×1 kernel. As shown in Fig. 1, each "Fire" module has one "squeeze" convolutional layer with S filters of size 1×1, followed by two "expand" layers with E_1 and E_2 filters of size 1×1 and 3×3, respectively. The illustration of the utilized model is presented in Fig. 2.

IV. EXPERIMENTAL SETUP AND RESULTS

A. DATASETS

For experiments, we utilized three publicly available datasets, namely, EmoDB [25], IEMOCAP [26], and SAVEE [27]. The EmoDB dataset consists of 535 German utterances from ten actors (5 female and 5 male participants) and includes seven emotional classes: anger, anxiety, boredom, disgust, happiness, neutral, and sadness. In experiments for the IEMOCAP dataset, 2280 improvised English utterances for four emotional classes were employed: anger, happiness, neutral, and sadness. In this regard, and similar to [28] and [29], samples from happiness and excitement classes in the original dataset were merged into one class, namely, happiness. The recording was conducted in five dyadic sessions with one male and one female per session, thus ten actors in total. The SAVEE dataset has 480 acted English utterances recorded from 4 male actors and consists of seven emotional classes: anger, fear, disgust, happiness, neutral, sadness, and surprise. The utterances are taken from the standard TIMIT corpus [30]. For all three datasets, a 16 kHz sampling rate was used. Table 1 shows the number of data points across classes for each dataset.

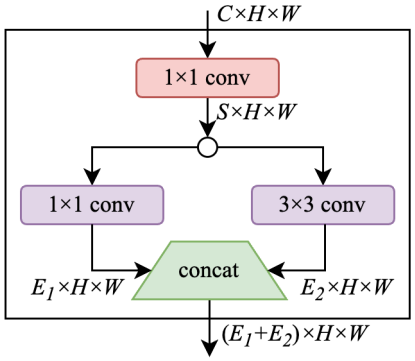


FIGURE 1. A "Fire" module used in SqueezeNet [17]. S is the number of 1×1 filters in a "squeeze" convolutional layer; E_1 and E_2 are the number of 1×1 and 3×3 filters in the "expand" layers, respectively; and C, H, and W are the number of channels, height, and weight of the input, respectively.

TABLE 1. Class-specific sample size.

	Emotion	SAVEE	EmoDB	IEMOCAP
1	Angry	60	127	289
2	Happy	60	71	947
3	Neutral	120	79	1099
4	Sadness	60	62	608
5	Anxiety / Fear	60	69	-
6	Boredom	-	81	-
7	Disgust	60	46	-
8	Surprise	60	-	-

B. EXPERIMENTAL SETUP

For each dataset, data was pooled (recorded waveforms) from all subjects and randomly split into two sets with a splitting ratio of 80:20%. The smaller subset was used as a test set and the larger subset was subsequently split into training and validation sets with the same splitting ratio. Splits were performed in a stratified manner to preserve the proportion of classes of the full dataset across training, validation, and test sets. In order to average out the effect of split-specific biases on results and subsequent conclusions, the process of randomly splitting the data into training, validation, and test sets was repeated $K = 10$ times in order to obtain average classification performance estimates — a procedure to which we refer as shuffle and split. In each repetition of shuffle and split, the training set is used for training the SqueezeNet architecture and the validation set is used for

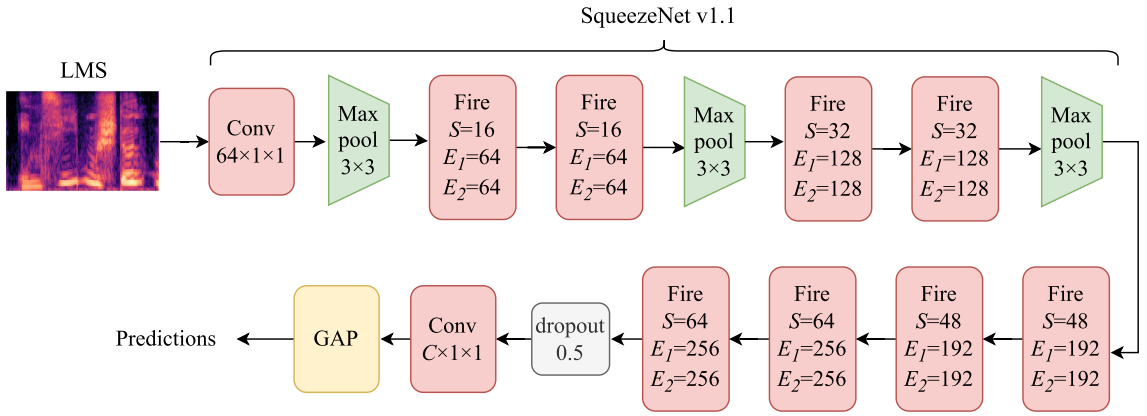


FIGURE 2. Architecture of SqueezeNet v1.1. In the figure, GAP and C indicate Global Average Pooling and a the number of classes, respectively. Every max pooling and convolutional layer has stride of 2.

model selection; that is, for tuning LMS-related hyperparameters (see Section IV-C). Finally, in each repetition, all models that resulted from the model selection stage were evaluated on the test set. We excluded the effect of learning the model training hyperparameters by fixing the Adam optimizer with a learning rate of 0.0001, a batch size of 16, a cross-entropy loss function, and 150 epochs. All experiments were conducted on a Windows 10 workstation with an Intel(R) Xeon(R) W-2145, (3.7 GHz) processor, 64 GB of RAM, and Nvidia TITAN RTX GPU (RAM = 24 GB, CUDA Cores: 4608). The entire workflow was implemented in PyTorch (version 1.9.1) with CUDA library version 11.1 and cuDNN library version 8.0. The source code for reproducing our experiments is available on Github.[1]

C. MODEL SELECTION AND EVALUATION

Let $\boldsymbol{\Theta}$ denote the space of LMS-related hyperparameters, which is determined based on the following parameters that were defined in Section III-A: (i) M, which is the number of Mel-filter banks; (ii) L, is the window length; (iii) the hop size H; and (iv) the audio segmentation length T. Candidate sets of these parameters that are used to specify $\boldsymbol{\Theta}$, which are subsequently used in the hyperparameter tuning, are presented in Table 2. Due to shorter maximum duration of recordings in the SAVEE dataset compared with the other two datasets (7 seconds for SAVEE, 9 and 29 seconds for EmoDB and IEMOCAP, respectively), the possible values of segmentation length for the SAVEE dataset was chosen differently. For the sake of comparison, we also added the "usual" values of LMS parameters, denoted $\boldsymbol{\theta}^{\bullet}$, as deployed in [11] and [12] to $\boldsymbol{\Theta}$; that is $\boldsymbol{\theta}^{\bullet} \triangleq \{L = 400$ (25ms × 16, 000 Hz sampling rate), $H = 160$ (10ms × 16, 000 Hz sampling rate), $M = 40$, $T = 3$s$\}$. The choice of hyperparameters indicated in Table 2 sets the cardinality of $\boldsymbol{\Theta}$ for each dataset to 17 $(M) \times 3$ $(L) \times 2$ $(H) \times 5$ $(T) + 1$ (due to $\boldsymbol{\theta}^{\bullet}$) = 511. We performed hyperparameter tuning by conducting a brute-force search within the specified $\boldsymbol{\Theta}$. The metric of model selection performance was weighted accuracy (WA) (also known as weighted average recall), defined as

$$\text{WA} = \frac{\sum_{i=1}^{c} N_{TP_i}}{\sum_{i=1}^{c} N_i}, \tag{6}$$

where c is the number of emotion classes, and N_i and N_{TP_i} denote the total number of evaluated instances and the number of correctly classified instances for class i, respectively.

Let $\mathcal{S} = \{(\mathbf{x}_j, y_j)\}, j = 1, \ldots, n$ denote a dataset with size n, where $\mathbf{x}_j \in \mathbb{R}^p$ is a p-dimensional feature vector and y_j is the corresponding class label. In each repetition $r = 1, \ldots, K$ of shuffle and split, we randomly split the dataset into training, validation, and test sets, denoted $\mathcal{S}_r^{tr}$, $\mathcal{S}_r^{val}$, $\mathcal{S}_r^{te}$, respectively, such that $\mathcal{S} = \mathcal{S}_r^{tr} \cup \mathcal{S}_r^{val} \cup \mathcal{S}_r^{te}$. Let $\text{M}_{\boldsymbol{\theta}}(\mathcal{D})$ denote a model built by applying a learning algorithm using a fixed set of hyperparemeters $\boldsymbol{\theta}$ on data $\mathcal{D}$. During the hyperparameter tuning for repetition r, we construct $\text{M}_{\boldsymbol{\theta}}(\mathcal{S}_r^{tr})$ for each $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, and evaluate the performance of the model on $\mathcal{S}_r^{val}$ to obtain $\text{WA}_r^{\boldsymbol{\theta}}$ [31].

Let $\boldsymbol{\theta}_r^*$ and $\boldsymbol{\theta}_r^\circ$, $r = 1, \ldots, K$, denote the values of the LMS hyperparameters that led to the highest WA (denoted WA_r^*) and lowest WA (denoted WA_r°) over (repetition-specific) validation set in the r^{th} repetition of shuffle and split, respectively; that is to say,

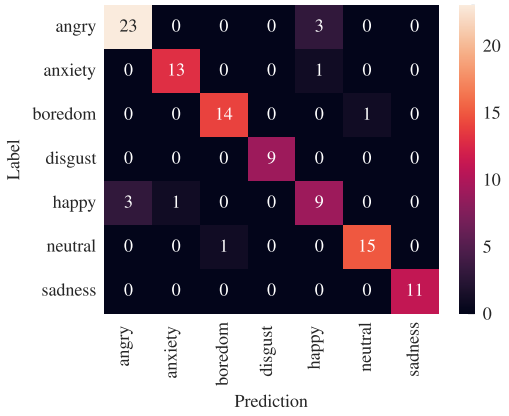
$$\boldsymbol{\theta}_r^* = \underset{\boldsymbol{\theta} \in \Theta}{\text{argmax}}\, \text{WA}_r^{\boldsymbol{\theta}}, \tag{7}$$

TABLE 2. The hyperparameter space $\boldsymbol{\Theta}$.

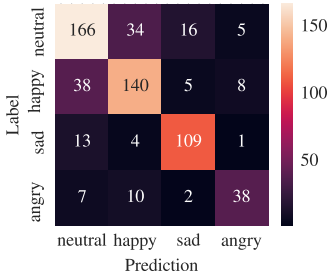
M, (Mel-filter banks)	L, (window length)	H, (hop length)	T (segmentation, length)[a]
128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, 248, 256	127, 255, 511	16, 32	{5, 6, 7, 8, 9}[b] {1, 2, 3, 4, 5}[c]

[a] in seconds
[b] for EmoDB and IEMOCAP datasets
[c] for SAVEE dataset

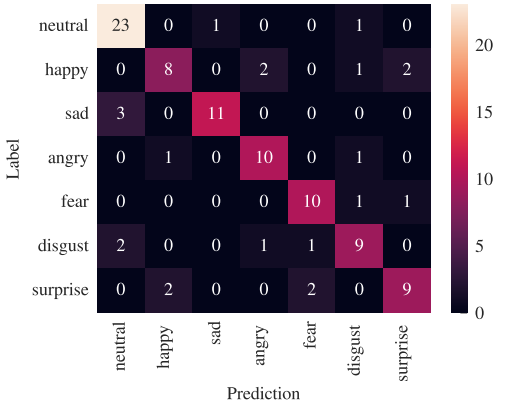
[1] https://github.com/Azamat-Mukhamediya/SER-LMS-SqueezeNet



(a)



(b)



(c)

FIGURE 3. Averaged confusion matrices across all repetition-specific test sets predictions for (a) EmoDB, (b) IEMOCAP, and (c) SAVEE datasets. To create the confusion matrices for each dataset and $\boldsymbol{\theta}_r^*$, all repetition-specific confusion matrices were averaged and rounded to a near integer.

$$\boldsymbol{\theta}_r^{\circ} = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmin}}\, \mathrm{WA}_r^{\boldsymbol{\theta}}, \tag{8}$$

$$\mathrm{WA}_r^{*} \triangleq \mathrm{WA}_r^{\boldsymbol{\theta}_r^*}, \tag{9}$$

$$\mathrm{WA}_r^{\circ} \triangleq \mathrm{WA}_r^{\boldsymbol{\theta}_r^{\circ}}. \tag{10}$$

Based on the common "winner-takes-all" model selection strategy, we select the SqueezeNet trained on the repetition-specific training data using $\boldsymbol{\theta}_r^*$ for further use. As a result, the selected SqueezeNet was subsequently evaluated on the repetition-specific test set. Table 3 shows the average

TABLE 3. Data-specific highest and lowest WA averaged over validation sets used in $K = 10$ repetitions of shuffle and split. Each entry is in the form of average ± standard deviation in percentage.

	EmoDB	IEMOCAP	SAVEE
$\bar{\mathrm{WA}}^{*}$	88.99 ± 1.89	76.08 ± 0.91	79.00 ± 1.42
$\bar{\mathrm{WA}}^{\circ}$	66.51 ± 6.15	67.93 ± 2.35	52.60 ± 4.03
$\bar{\mathrm{WA}}^{\bullet}$	69.78 ± 4.49	69.04 ± 1.01	65.50 ± 2.78

(over repetition-specific validation sets) highest and lowest WA for each dataset; that is, $\bar{\mathrm{WA}}^{*} \triangleq \frac{1}{K}\sum_{r=1}^{K} \mathrm{WA}_r^{*}$ and $\bar{\mathrm{WA}}^{\circ} \triangleq \frac{1}{K}\sum_{r=1}^{K} \mathrm{WA}_r^{\circ}$.

Two key observations are apparent from Table 3. First, the performance of SqueezeNet greatly depends on LMS hyperparameters. Although specific results depend on the dataset, in our experiments we see that by changing values of LMS hyperparameters the average classification performance of SqueezeNet changes in a wide range from $\sim$ 8% (for the IEMOCAP dataset) to $\sim$ 26% (for the SAVEE dataset). A second key observation is that the use of $\boldsymbol{\theta}^{\bullet}$ led to a relatively poor performance of SqueezeNet across all datasets. In particular, we see that for the all datasets, $\bar{\mathrm{WA}}^{\bullet}$, which denotes the average WA obtained by $\boldsymbol{\theta}^{\bullet}$ over repetition-specific validation sets, is closer to $\bar{\mathrm{WA}}^{\circ}$ than $\bar{\mathrm{WA}}^{*}$.

Let $\bar{\mathrm{WA}}_{\mathrm{test}}^{*}$ and $\bar{\mathrm{WA}}_{\mathrm{test}}^{\bullet}$ show dataset-specific average WA over repetition-specific test sets for the SqueezeNet trained using $\boldsymbol{\theta}_r^*$ and $\boldsymbol{\theta}_r^{\bullet}$, respectively. Table 4 shows $\bar{\mathrm{WA}}_{\mathrm{test}}^{*}$ and $\bar{\mathrm{WA}}_{\mathrm{test}}^{\bullet}$ for each dataset. The remarkable difference between $\bar{\mathrm{WA}}_{\mathrm{test}}^{*}$ and $\bar{\mathrm{WA}}_{\mathrm{test}}^{\bullet}$ shows that proper tuning of LMS hyperparameters can lead to substantial performance improvements of SqueezeNet with respect to deploying the usual parameters. Fig. 3 shows the averaged (rounded to a near integer) confusion matrices across all repetition-specific test sets using $\boldsymbol{\theta}_r^*$ for each dataset. We also present the weighted average (over classes) precision and weighted average f_1-score [32] metrics, defined as follows, averaged over all repetition-specific test sets using $\boldsymbol{\theta}_r^*$ (see Table 5):

$$\text{precision} = \frac{\sum_{i=1}^{c} \frac{N_{TP_i}}{N_{TP_i}+N_{FP_i}} \cdot N_i}{\sum_{i=1}^{c} N_i}, \tag{11}$$

$$\text{f}_1\text{-score} = \frac{\sum_{i=1}^{c} \frac{2N_{TP_i}}{2N_{TP_i}+N_{FP_i}+N_{FN_i}} \cdot N_i}{\sum_{i=1}^{c} N_i}, \tag{12}$$

where N_{FP_i} is the number of incorrectly classified instances (the predicted instance is class i, but the true instance is different class), N_{FN_i} is the number of incorrectly classified instances (the predicted instance is different class, but the true instance is class i).

V. DISCUSSION

In this study, we are primarily concerned with the impact of LMS-related parameters on the performance of DNN classifiers. This focus is warranted because many studies overlook the impact of these parameters on the performance of DNN

TABLE 4. Data-specific WA averaged over test sets used in $K = 10$ repetitions of shuffle and split. Each entry is in the form of average ± standard deviation in percentage.

	EmoDB	IEMOCAP	SAVEE
$\bar{\text{WA}}^{*}_{\text{test}}$	88.22 ± 1.87	75.97 ± 1.40	79.00 ± 1.24
$\bar{\text{WA}}^{\bullet}_{\text{test}}$	64.78 ± 4.12	65.31 ± 1.45	67.30 ± 2.45

TABLE 5. Data-specific weighted average precision and weighted average f_1-score averaged over test sets used in $K = 10$ repetitions of shuffle and split. Each entry is in the form of average ± standard deviation.

	EmoDB	IEMOCAP	SAVEE
precision	0.89 ± 0.02	0.76 ± 0.01	0.80 ± 0.01
f_1-score	0.88 ± 0.02	0.76 ± 0.01	0.79 ± 0.01

classifiers (see Introduction). Using SqueezeNet as archetypal, our results obtained in Section IV show that: 1) LMS hyperparameters can considerably affect the performance of DNN classifiers used in SER; and 2) the values of LMS hyperparameters that are commonly used in the literature lead to relatively poor performance of DNN classifiers. The former point can be observed in the results of Table 3, which shows that varying values of LMS hyperparameters can considerably change the average WA of the SqueezeNet. The latter point is evident in the results of Table 4 where it can be seen that in contrast with the use of "usual" values of LMS parameters, treating them as hyperparameters and tuning them led to a $\sim 23\%$, $\sim 10\%$, and $\sim 11\%$ improvement in average WA in EmoDB, IEMOCAP, and SAVEE datasets, respectively.

Fig. 4 shows the heat maps of the dataset-specific average (over repetition-specific validation sets) WA across all combinations of LMS hyperparameters. To create the heat maps, we divided the four LMS hyperparameters into two tuples: (T, L) combinations on the horizontal axis (x-axis), and (M, H) on the vertical axis (y-axis). At the same time, we applied hierarchical clustering on both axes to better group data points in each heat map. Using heat maps one can visualize the considerable impact of LMS parameters on the performance of SqueezeNet. Said that, an interesting observation in the heat maps is the transition of the results across the y-axis for the EmoDB dataset, and across the x-axis for the remaining two datasets. In particular, in Fig. 4 (a) we observe that a higher average WA is *generally* achieved for lower values of M regardless of other parameters; that is, for $M \in \{128, 136, 144, 152\}$. In contrast, for the other two datasets, a better average WA is generally achieved for lower range of T and higher values of L. These interesting observations further confirm that acceptable LMS parameters depend on the specific dataset—and that is when hyperparameter tuning is required.

A question that remains is how our results, obtained by taking a single SqueezeNet architecture with LMS hyperparameter tuning, compare to other classification results obtained using other, more advanced types of DNNs reported in the literature for these datasets. To answer this question,

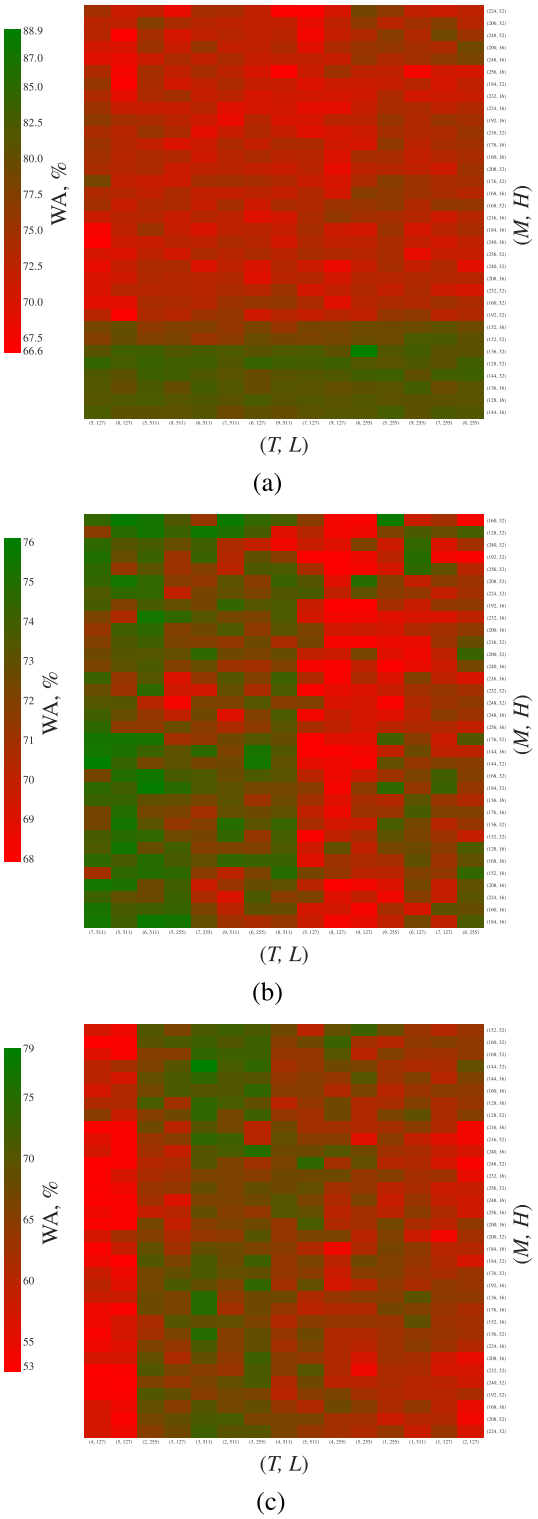


FIGURE 4. Heat maps of average (over repetition-specific validation sets) WA across all combinations of LMS hyperparameters for (a) EmoDB, (b) IEMOCAP, and (c) SAVEE datasets. To create the heat maps, the four LMS hyperparameters were divided into two tuples: (T, L) combinations on the horizontal axis, and (M, H) on the vertical axis.

we summarize state-of-the-art results obtained in several other studies in Table 6. For ease of comparison, our results,

TABLE 6. Comparison with state-of-art results on test set.

Dataset	Paper	WA, % ± std.
EmoDB	Maji *et al.* [15] (2022)	90.31
	Andayani *et al.* [33] (2022)	87.72
	Nagarajan *et al.* [34] (2020)	86.96
	Chen *et al.* [35] (2021)	85.42
	Ours	88.22 ± 1.87
IEMOCAP	Nediyanchath *et al.* [36] (2020)	76.40
	Xu *et al.* [28] (2021)	76.18
	Yenigalla *et al.* [37] (2018)	73.90
	Tarantino *et al.* [29] (2019)	70.17
	Ours	75.97 ± 1.40
SAVEE	Singh *et al.* [38] (2021)	81.70
	Avots *et al.* [39] (2019)	77.40
	Nagarajan *et al.* [34] (2020)	77.08
	Liu *et al.* [40] (2018)	76.40
	Ours	79.00 ± 1.24

which were reported in the first row of Table 4, have been added to this table. As can be seen, our results are comparable to other studies across all three datasets.

VI. CONCLUSION

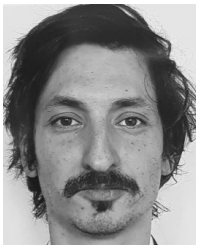
A salient issue and one not explicitly stated in DNN-based speech emotion recognition is whether LMS parameters (number of Mel-filter banks, hop size, and window and segmentation lengths) have a major impact on the performance of trained classifiers. In contrast with many other studies in SER that concentrate on constructing new DNN architectures that are superior to existing ones, in this study we fix our choice of DNN classifier by adopting a lightweight architecture among other pre-trained CNNs, namely, SqueezeNet, and instead treat LMS parameters as the classifier hyperparameters that require tuning. Our empirical results on three publicly available datasets show that: 1) LMS hyperparameters can considerably affect the performance of DNN classifiers used in SER; and 2) compared with the utility of some common values of LMS parameters that are found in literature, one can achieve a major improvement in classification performance if instead these parameters are treated as the classifier hyperparameters and tuned in a model selection phase. We further showed that in our experiments this practice led to comparable results with state-of-the-art DNN architectures used on similar datasets. An interesting research issue for the future is to examine whether jointly tuning of the LMS and DNN architecture parameters can lead to a considerable improvement in classification performance over existing results.

REFERENCES

[1] M. B. Er, "A novel approach for classification of speech emotions based on deep and acoustic features," *IEEE Access*, vol. 8, pp. 221640–221653, 2020.
[2] M. Dhuheir, A. Albaseer, E. Baccour, A. Erbad, M. Abdallah, and M. Hamdi, "Emotion recognition for healthcare surveillance systems using neural networks: A survey," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2021, pp. 681–687.
[3] K. Bahreini, R. Nadolski, and W. Westera, "Towards real-time speech emotion recognition for affective e-learning," *Educ. Inf. Technol.*, vol. 21, no. 5, pp. 1367–1386, Sep. 2016, doi: 10.1007/s10639-015-9388-2.
[4] M. Bojanić, V. Delić, and A. Karpov, "Call redistribution for a call center based on speech emotion recognition," *Appl. Sci.*, vol. 10, no. 13, p. 4653, Jul. 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/13/4653
[5] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li, and H. Shi, "Escaping the big data paradigm with compact transformers," 2021, *arXiv:2104.05704*.
[6] D. Amodei, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn. (ICML)*, vol. 48. New York, NY, USA: JMLR.org, 2016, pp. 173–182.
[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.
[8] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2018.
[9] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K. Müller, "Explaining deep neural networks and beyond: A review of methods and applications," *Proc. IEEE*, vol. 109, no. 3, pp. 247–278, Mar. 2021.
[10] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.
[11] H. Meng, T. Yan, F. Yuan, and H. Wei, "Speech emotion recognition from 3D Log-Mel spectrograms with deep learning network," *IEEE Access*, vol. 7, pp. 125868–125881, 2019.
[12] M. Chen, X. He, J. Yang, and H. Zhang, "3-D convolutional recurrent neural networks with attention model for speech emotion recognition," *IEEE Signal Process. Lett.*, vol. 25, no. 10, pp. 1440–1444, Oct. 2018.
[13] W. Fan, X. Xu, B. Cai, and X. Xing, "ISNet: Individual standardization network for speech emotion recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 30, pp. 1803–1814, 2022.
[14] P. Jiang, H. Fu, H. Tao, P. Lei, and L. Zhao, "Parallelized convolutional recurrent neural network with spectral features for speech emotion recognition," *IEEE Access*, vol. 7, pp. 90368–90377, 2019.
[15] B. Maji, M. Swain, and M. Mustaqeem, "Advanced fusion-based speech emotion recognition system using a dual-attention mechanism with conv-caps and bi-GRU features," *Electronics*, vol. 11, no. 9, p. 1328, Apr. 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/9/1328
[16] D. Tang, P. Kuppens, L. Geurts, and T. van Waterschoot, "End-to-end speech emotion recognition using a novel context-stacking dilated convolution neural network," *EURASIP J. Audio, Speech, Music Process.*, vol. 2021, no. 1, p. 18, May 2021, doi: 10.1186/s13636-021-00208-5.
[17] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size," 2016, *arXiv:1602.07360*.
[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent., (ICLR)*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, 2015, pp. 1–14.
[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, vol. 1. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105.
[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[22] N. Kehtarnavaz, "Frequency domain processing," in *Digital Signal Processing System Design*, 2nd ed., N. Kehtarnavaz, Ed. New York, NY, USA: Academic, 2008, pp. 175–196. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123744906000076
[23] D. O'Shaughnessy, *Hearing*. Hoboken, NJ, USA: Wiley, 2000, pp. 109–139.
[24] X. Huang, A. Acero, H.-W. Hon, and R. Reddy, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, 1st ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.
[25] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of German emotional speech," in *Proc. Interspeech*, Sep. 2005, pp. 1517–1520.
[26] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Lang. Resour. Eval.*, vol. 42, no. 4, pp. 335–359, Dec. 2008, doi: 10.1007/s10579-008-9076-6.

[27] S. Haq and P. Jackson, *Machine Audition: Principles, Algorithms and Systems*. Hershey, PA, USA: IGI Global, Aug. 2010, ch. Multimodal Emotion Recognition, pp. 398–423.
[28] M. Xu, F. Zhang, and W. Zhang, "Head fusion: Improving the accuracy and robustness of speech emotion recognition on the IEMOCAP and RAVDESS dataset," *IEEE Access*, vol. 9, pp. 74539–74549, 2021.
[29] L. Tarantino, P. N. Garner, and A. Lazaridis, "Self-attention for speech emotion recognition," in *Proc. Interspeech*, Sep. 2019, pp. 2578–2582.
[30] L. F. Lamel, R. H. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Proc. Speech Input/Output Assessment Speech Databases*, 1989, pp. 161–170.
[31] M. Feurer and F. Hutter, *Hyperparameter Optimization*. Cham, Switzerland: Springer, 2019, pp. 3–33, doi: 10.1007/978-3-030-05318-5_1.
[32] J. Ashok Kumar, S. Abirami, A. Ghosh, and T. E. Trueman, "A C-LSTM with attention mechanism for question categorization," in *Machine Learning and Metaheuristics Algorithms, and Applications*, S. M. Thampi, L. Trajkovic, K.-C. Li, S. Das, M. Wozniak, and S. Berretti, Eds. Singapore: Springer, 2020, pp. 234–244.
[33] F. Andayani, L. B. Theng, M. T. Tsun, and C. Chua, "Hybrid LSTM-transformer model for emotion recognition from speech audio files," *IEEE Access*, vol. 10, pp. 36018–36027, 2022.
[34] S. Nagarajan, S. S. S. Nettimi, L. S. Kumar, M. K. Nath, and A. Kanhe, "Speech emotion recognition using cepstral features extracted with novel triangular filter banks based on bark and ERB frequency scales," *Digit. Signal Process.*, vol. 104, Sep. 2020, Art. no. 102763. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200420301081
[35] S. Chen, M. Zhang, X. Yang, Z. Zhao, T. Zou, and X. Sun, "The impact of attention mechanisms on speech emotion recognition," *Sensors*, vol. 21, no. 22, p. 7530, Nov. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/22/7530
[36] A. Nediyanchath, P. Paramasivam, and P. Yenigalla, "Multi-head attention for speech emotion recognition with auxiliary learning of gender recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 7179–7183.
[37] P. Yenigalla, A. Kumar, S. Tripathi, C. Singh, S. Kar, and J. Vepa, "Speech emotion recognition using spectrogram & phoneme embedding," in *Proc. Interspeech*, Sep. 2018, pp. 3688–3692.
[38] P. Singh, R. Srivastava, K. P. S. Rana, and V. Kumar, "A multimodal hierarchical approach to speech emotion recognition from audio and text," *Knowl.-Based Syst.*, vol. 229, Oct. 2021, Art. no. 107316. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705121005785
[39] E. Avots, T. Sapiński, M. Bachmann, and D. Kamińska, "Audiovisual emotion recognition in wild," *Mach. Vis. Appl.*, vol. 30, no. 5, pp. 975–985, Jul. 2019, doi: 10.1007/s00138-018-0960-9.
[40] Z.-T. Liu, Q. Xie, M. Wu, W.-H. Cao, Y. Mei, and J.-W. Mao, "Speech emotion recognition based on an improved brain emotion learning model," *Neurocomputing*, vol. 309, pp. 145–156, Oct. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218305344

AZAMAT MUKHAMEDIYA (Graduate Student Member, IEEE) received the bachelor's (Hons.) and master's degrees in radio engineering, electronics and telecommunications from L. N. Gumilyov Eurasian National University, Kazakhstan, in 2019 and 2021, respectively. He is currently pursuing the Ph.D. degree in machine learning with a focus on speech emotion recognition applications.



SIAMAC FAZLI received the B.Sc. degree in physics from the University of Exeter, Exeter, U.K., in 2002, the M.Sc. degree in medical neurosciences from the Humboldt University of Berlin, Berlin, Germany, in 2004, and the Ph.D. degree from the Berlin Institute of Technology, Berlin, in 2011.

From 2011 to 2013, he was a Postdoctoral Researcher with the Berlin Institute of Technology, Bernstein Focus Neurotechnology. In 2013, he was appointed as an Assistant Professor with Korea University, Seoul, South Korea. From 2016 to 2017, he was a Group Leader with the Fraunhofer Institute for Telecommunications, Berlin. In 2018, he joined the Computer Science Department, Nazarbazev University, as an Associate Professor. His current research interests include machine learning, neuroscience, multimodal neuroimaging, and brain–computer interfacing.

AMIN ZOLLANVARI (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Shiraz University, Iran, and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, TX, USA, in 2010. He held a postdoctoral position with the Harvard Medical School and Brigham and Women's Hospital, Boston, MA, USA (2010–2012), and then he joined the Department of Statistics, Texas A&M University, as an Assistant Research Scientist (2012–2014). Since 2015, he has been with Nazarbayev University, where he is currently an Associate Professor with the Department of Electrical and Computer Engineering. He is the author of the textbook *Machine Learning with Python: Theory and Implementation* (Springer, 2023). His research interests include machine learning, statistical signal processing, and biomedical informatics.