Arabic Emotion Speech Recognition Using Parallel CNN with LSTM and Attention Mechanisms

Ahmad Alsallaq[1], Besan Musallam[2], Ahmad Arekat[3], Sara Al-Araj[4]

***Abstract*—Emotion recognition from speech is a challenging task, particularly for languages like Arabic, where resources are limited. This paper presents a novel approach using a parallel Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) and attention mechanisms to classify emotions in Arabic speech. The Basic Arabic Vocal Emotions Dataset was used, consisting of three classes: low, neutral, and high emotions. The model achieved a significant accuracy of 96.91%, demonstrating the effectiveness of the proposed architecture.**

Introduction and Statement of Problem

Emotion recognition in speech plays a crucial role in human communication and interaction, with applications ranging from improving human-computer interaction [7] to aiding in psychological analysis and therapy. The ability to accurately identify emotions conveyed through speech can enhance various aspects of our daily lives, including customer service, virtual assistants, and mental health support systems. However, despite the advancements in machine learning and natural language processing, emotion recognition in Arabic speech remains a challenging task.[13]

The complexity of accurately capturing and interpreting emotional nuances in Arabic speech arises from several factors. Firstly, Arabic is a rich and diverse language with multiple dialects, regional variations, and cultural influences, making it challenging to develop models that generalize well across different contexts. Additionally, the limited availability of labeled datasets for Arabic speech emotion recognition poses a significant obstacle to the development and evaluation of effective models. Unlike some other languages with well-established datasets and benchmarks, such as English, Arabic lacks comprehensive resources for training and testing emotion recognition systems.[2]

Furthermore, the inherent variability in emotional expression across different speakers and cultural backgrounds adds another layer of complexity to the problem. Emotions are inherently subjective and context-dependent, making it challenging to create models that can accurately recognize and classify emotions across diverse populations and scenarios. Additionally, the subtle nuances and intricacies of emotional expression in Arabic speech, including variations in intonation, rhythm, and pronunciation, further complicate the task of building robust emotion recognition systems.[2]

In light of these challenges, this paper proposes a

Published by the IEEE Consumer Technology Society

novel approach to Arabic speech emotion recognition using a parallel Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) layers and attention mechanisms [6]. By leveraging the strengths of CNNs for feature extraction, LSTMs for sequence modeling, and attention mechanisms for focusing on relevant information, we aim to improve the accuracy and robustness of emotion recognition in Arabic speech. Our approach seeks to address the limitations of existing models by integrating advanced architectural components and leveraging recent advancements in deep learning techniques see Figure 5 for the pipeline used.

Through empirical evaluation on the Basic Arabic Vocal Emotions Dataset (BAVED), we demonstrate the effectiveness of our proposed model in accurately recognizing and classifying emotions in Arabic speech. By overcoming the challenges posed by limited datasets and the complexity of Arabic emotional expression, our work contributes to advancing the state-of-the-art in Arabic speech emotion recognition and lays the foundation for future research in this important area. [1]

In summary, this paper presents a comprehensive investigation into the challenges and opportunities in Arabic speech emotion recognition and proposes a novel model architecture to address these challenges. Our approach holds promise for enhancing various applications that rely on accurate emotion recognition in Arabic speech, ultimately improving human-computer interaction and facilitating deeper insights into human emotion and behavior.

LITERATURE REVIEW

• Recognizing emotions from speech signals, a pivotal aspect for enhancing natural humancomputer interaction, has garnered substantial attention in the realm of affective computing. Opting for Arabic Speech Emotion Recognition (SER) was necessitated by the limited availability of databases in this domain. This study utilized a dataset featuring 24 Arabic emotional sentences recorded twice by six performers, encompassing five emotions and totaling 1440 records. Employing Mel-Frequency Cepstral Coefficients (MFCC) with a high pass filter (HPF) for noise reduction, and Principal Component Analysis (PCA) for normalization and dimensionality reduction, we evaluated the performance of three models: Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Recurrent Neural Networks (RNN). SVM stood out with the highest accuracy at 93.12%, notably excelling in predicting sadness emotions without any misclassifications, in contrast to ANN (87.63%) and RNN (90.13%) performances. These findings underscore the effectiveness of SVM in Arabic SER, providing insights into emotion recognition accuracy and model distinctions. [5]

• This paper explores the impact of the Arabic language on physiological events and emotional behavior within the cultural context, with a primary focus on recognizing basic emotions in Arabic speech, including neutral, sadness, fear, anger, and happiness. Utilizing the REGIM_TES dataset and various descriptors such as Pitch of voice, Energy, MFCCs, Formant, LPC, and the spectrogram, the study employs the RBF kernel Support Vector Machines (SVMs) multiclass classifier for emotion classification. Notably, the segmentation of the Arabic corpus is conducted without relying on syntactic or delimiters linguistics, treating the acoustic flow as continuous to estimate emotional responses throughout conversations. The obtained precision rates for emotion classification showcase the effectiveness of the RBF kernel SVMs multiclass classifier: Anger: 93.65%, Fear: 96.62%, Happiness: 90.97%, Neutral: 86.36%, Sadness: 96.19%. These results emphasize the significance of accounting for cultural and linguistic nuances in the analysis of emotional behavior in Arabic language studies. [21]

• This paper introduces two deep learning architectures, Convolutional Neural Networks (CNNs) and Bi-directional Long Short-Term Memory (BLSTM), to achieve accurate emotion recognition from spoken Arabic while maintaining computational efficiency for real-world applications. The models were evaluated using the KSUEmotions dataset, and the results showed that the attention-based model outperformed a strong baseline deep CNN system, achieving a 2.2% improvement in accuracy. However, it was noted that the deep CNN model exhibited significantly faster training and execution times. The classification results, presented in Table 1 for different folds, demonstrate the comparative performance of CNN-BLSTM-DNN, CNN, and the overall accuracy achieved across multiple folds, providing valuable insights into the trade-off between accuracy and computational efficiency in Arabic speech emotion recognition see Table 1 for the Classification Results. [12]

• The research paper is centered around Arabic speaker emotion classification, employing rhythm met-

Table 1. Classification Results Using the Phase-mix dataset

Fold	Classification overall accuracy (%)	
	CNN-BLSTM-DNN	CNN
1	87.7	85.2
2	86.6	85.6
3	88.2	84.4
4	86.0	85.2
5	87.6	84.7
Average	87.2	85.9

rics such as Interval Measures (IMs) and Pairwise Variability Indices (PVIs) in conjunction with Multilayer Perceptron (MLP) neural network classifiers. Utilizing the King Saud University Emotions (KSUEmotions) Corpus recorded in two phases to simulate six emotions, the study achieved a maximum accuracy of 72.41% for both male and female speakers combined, utilizing all rhythm metrics. However, a noticeable discrepancy was observed, with lower accuracy for female speakers compared to their male counterparts. The results indicated that anger was the most readily recognized speaker emotion, while happiness posed a greater challenge for classification. Furthermore, the study highlighted that emotions expressed by male speakers were generally easier to discern than those of female speakers. In conclusion, the research underscores the effectiveness of neural networks in conjunction with rhythm metrics for Arabic speaker emotion recognition, particularly with a sufficiently large dataset. [22]

METHODOLOGY

1) Data collection

1.1)Data collection: data collection is an initial step for training our classifier. However, the lack of Acoustic data resources was an obstacle, but we managed to use the Basic Arabic Vocal Emotion Dataset, a comprehensive collection designed for basic Arabic speech recognition and vocal detection. [1]

1.2) Data annotation: This dataset comprises seven Arabic words, each recorded at three emotion levels to capture a range from low emotional expression (like feeling tired or down) to a standard neutral state, and finally, a high level of positive or negative emotions (happiness, joy, sadness, anger, etc.). The dataset consists of 1,935 records from 61 speakers, with 45 male and 16 female participants. The dataset's size is approximately 97.8 MB, and the recordings underwent normalization and formatting, resulting in audio WAV files with a sample rate of 16 kHz, single-channel mono audio, and a bit rate of 256 kbit/s. The dataset's naming convention includes speaker ID, gender, age, spoken word, spoken emotion, and record ID. [1]

1.3) Data Labeling: The metadata was extracted from filenames and stored in a Pandas DataFrame. The emotion labels were automatically extracted from the filenames of the audio files, which adhere to a standardized naming convention. This automated labeling process ensured efficiency and consistency in assigning emotion labels to the dataset.

1.4) Exploratory Data Analysis: The distribution of emotions and gender in the dataset was visualized using bar plots to ensure balanced representation

gender distribution: There were more male sample than female sample as shown in Figure 1.

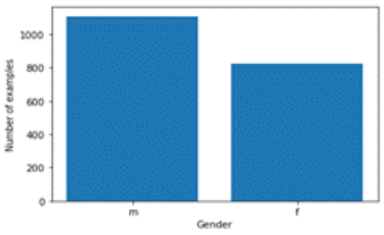

Figure 1. Gender Distribution of Speakers

Emotion Distribution: The dataset contained more samples labeled as 'low' compared to 'neutral' and 'high' emotions, with only slight differences between the counts of 'low' and 'neutral' emotions as shown in Figure 2.

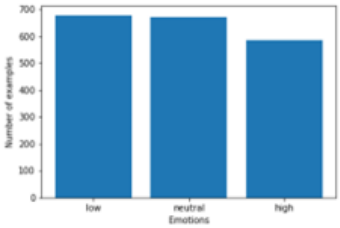

Figure 2. Distribution of emotions in the dataset

1.5) Data split: The dataset was split into 80% for training, 10% for validation, and 10% for testing to evaluate the model's performance.

2) Preprocessing

The preprocessing of the dataset involved several steps to prepare the audio data for model training.

2.1) Mel Spectrogram

The mel spectrogram is a representation of the frequency content of an audio signal over time. It is derived from the traditional spectrogram but is modified to better align with human auditory perception. Here's a breakdown of the steps involved in generating a mel spectrogram:

1) Frame the Audio : The audio signal is divided into short, overlapping frames using a windowing function such as the Hamming window. This process helps capture the spectral content of the audio over short time intervals.

2) Compute the Short-Time Fourier Transform (STFT) : For each frame, the Fourier Transform is applied to obtain the frequency content of the signal. The resulting spectrogram represents the magnitude of the frequency components over time.

3) Apply the Mel Filterbank : The linear frequency scale of the traditional spectrogram is converted to the mel scale, which better reflects human auditory perception. This conversion is achieved using a series of triangular filters spaced evenly on the mel scale.

4) Compute the Logarithm : The magnitude values in the mel spectrogram are typically converted to decibels (dB) using a logarithmic scale. This helps emphasize lower intensity sounds and improves the dynamic range of the spectrogram.

The mel spectrogram is widely used in audio processing tasks such as speech recognition and music analysis due to its ability to capture relevant acoustic features in a format that closely resembles human auditory perception. [27]

To extract the Mel spectrogram, we implemented the `getMELspectrogram` function using the Librosa library. The audio files were resampled to a uniform sample rate of 16,000 Hz. The Mel spectrograms were computed using a Hamming window with a window size of 512 samples and a hop length of 256 samples, resulting in 128 Mel frequency bins. Specifically, the Mel spectrograms were computed using `librosa.feature.melspectrogram` and subsequently converted to decibel units with `librosa.power_to_db`, producing a 2D array representing the Mel spectrogram. The Mel spectrogram visualization is shown in Figure 3.

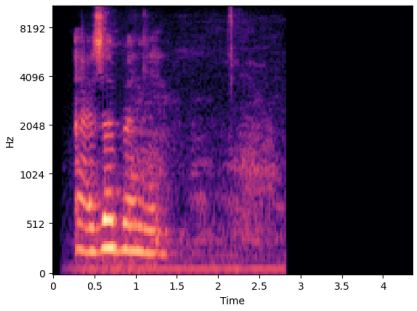

Figure 3. Mel spectrogram visualization

2.2) Additive White Gaussian Noise (AWGN)

Additive White Gaussian Noise (AWGN) is a type of random noise that is often added to audio signals during preprocessing to augment the dataset and improve model robustness.

1) Additive : AWGN is added to the original audio signal, meaning it is combined with the existing signal without modifying it.

2) White : AWGN has a flat frequency spectrum, meaning it contains equal power at all frequencies. This property makes it suitable for simulating various types of background noise.

3) Gaussian : AWGN follows a Gaussian (normal) distribution, meaning the amplitude of the noise at any given time is sampled from a normal distribution with a mean of zero and a specified standard deviation.

By adding AWGN to the audio data, the model is exposed to a wider range of acoustic variations, helping it become more robust to noise and other environmental factors. This augmentation technique can improve the model's generalization performance and make it more effective in real-world scenarios where the audio may contain background noise or other disturbances.

Overall, both the mel spectrogram and AWGN play crucial roles in preprocessing audio data, helping to extract relevant features and enhance the model's ability to learn and generalize from the training data.[23]

3) Modeling Approach

The model architecture is designed to leverage both spatial and temporal features inherent in audio data for robust emotion classification. It employs a parallel structure, integrating Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTMs) networks, augmented with an attention mechanism. [29]

3.1 Parallel CNN-LSTM Architecture

- **Convolutional Blocks (CNN) [30]:** The model incorporates four sequential convolutional blocks. Each block consists of:
  - A 2D convolutional layer with a 3x3 kernel size and an increasing number of filters (16, 32, 64, 64) in each subsequent block. This enables the extraction of progressively higher-level features from the Mel spectrograms.
  - Batch normalization to stabilize and accelerate training by normalizing the activations within each mini-batch.
  - A Rectified Linear Unit (ReLU) [9] activation function to introduce non-linearity and improve model expressiveness.
  - A max-pooling layer with a 2x2 or 4x4 kernel size (depending on the block) to downsample the feature maps, reducing computational complexity and aiding in translational invariance.
  - A dropout layer with a probability of 0.3 to mitigate overfitting by randomly deactivating neurons during training.
- **LSTM Block [15]:**
  - A max-pooling layer with a kernel size of [2, 4] and a stride of [2, 4] is applied to the input Mel spectrograms to reduce their dimensionality before feeding them into the LSTM layer.
  - A bidirectional LSTM layer with 128 hidden units is employed to capture temporal dependencies in the audio data. The bidirectional nature allows the model to learn from both past and future contexts, enhancing its ability to understand the sequential nature of emotions in speech.
  - A dropout layer with a probability of 0.1 is applied to the LSTM output to further regularize the model and prevent overfitting.

3.2 Attention Mechanism [24] The model incorporates a multi-head self-attention mechanism [28] to weigh the importance of different time steps in the LSTM output sequence. Self-attention allows the model to attend to different parts of the input sequence when processing each time step, enabling it to capture dependencies and relationships between different elements in the sequence.

The multi-head self-attention mechanism operates as follows:

1) **Linear Projections:** The input sequence (output of the bidirectional LSTM) is linearly projected into three distinct representations: queries (Q), keys (K), and values (V). Each projection is learned through separate weight matrices.
2) **Scaled Dot-Product Attention:** Scaled dot-product attention is computed for each head. This involves calculating the dot product between the query and all keys, scaling it down by the square root of the dimension of the keys, and applying a softmax function to obtain normalized attention weights. These weights determine the contribution of each value to the final output for that head.
3) **Multi-Head Attention:** The attention mechanism is applied multiple times in parallel with different linear projections (heads). This allows the model to capture different aspects of the input sequence simultaneously.
4) **Concatenation and Output Projection:** The outputs from all attention heads are concatenated and then combined with the original LSTM output. This combined output is then transformed by a final linear layer to produce the final attention output.

The output of the multi-head attention layer is then combined with the original LSTM output to provide a richer representation of the input sequence, which is subsequently used for emotion classification.

Figure 4. The construction of the CNN+LSTM+Self-Attention+LSTM model

3.3 Output Layer

- The flattened output from the final convolutional block (CNN features) is concatenated with the attention context vector.
- This concatenated representation is then passed through a linear layer to produce logits for the three emotion classes (low, neutral, high).
- A softmax activation function is applied to the logits to obtain the final class probabilities.

3.4 Loss Function and Optimizer

- The model is trained using the cross-entropy loss function [18], a standard choice for multi-class classification tasks.
- The Adam optimizer [14] is employed to update the model's parameters during training, leveraging adaptive learning rates for each parameter.
- A ReduceLROnPlateau [8]learning rate scheduler is utilized to dynamically adjust the learning rate based on the validation loss, promoting stable convergence and preventing overfitting.

4) Computational Infrastructure and Software Tools:

4.1) Hardware:

Processor: Intel(R) Core (TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz Installed RAM: 16.0 GB (15.8 GB usable), System type: 64-bit operating system, x64-based processor

4.2) Software:

1. Development Environment: JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality [26].

2. Programming Language: Python is the programming language used for data processing and model development which offers a rich ecosystem of libraries and frameworks for machine learning and data science.

3. Deep Learning Framework: PyTorch 1.8.0 [3]: Used for building and training the neural network models.

4. Audio Processing: Librosa 0.8.0 [19]: Used for feature extraction.

5. Data Processing and Analysis: we used Scikit-learn 0.24.1 [25] for data preprocessing and evaluation metrics, Pandas 1.2.3 [20] for handling and processing the dataset, and NumPy 1.20.1 [10] for numerical computations, array operations, and data augmentation.

6. Pandas 1.2.3 [20]**:** Used for handling and processing the dataset.

7. Version Control:

• Git: a distributed version control system that tracks changes in any set of computer files

• GitHub: a web-based interface that uses Git, the open source version control software that lets multiple people make separate changes to web pages at the same time.

5) Evaluation Matrices

We opted to use the standard classification metrics, such as accuracy, precision, recall, f1 score, and confusion matrix [4], to evaluate the models. For binary classification tasks, such metrics are simple and can be obtained using Equations (1)–(4).

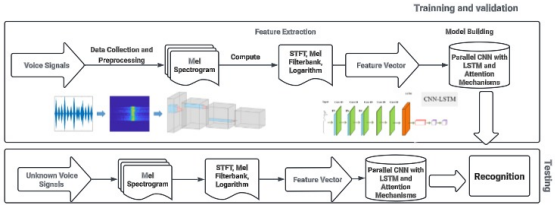

Figure 5. A pipeline for voice recognition using a Parallel CNN with LSTM and Attention Mechanisms model

RESULTS

The initial pre-built model[11] was trained using a learning rate of 0.01 and did not utilize the ReduceLROnPlateau scheduler [8]. This model achieved an accuracy of 93.81% with a loss of 0.166. By changing the learning rate and introducing the ReduceLROnPlateau scheduler, the model's performance improved significantly.

Learning Rate Adjustment

Learning Rate Increase: Increasing the learning rate from 0.01 to 0.02 accelerated the convergence process during training [17]. The higher learning rate allowed the model to make larger updates to the weights, which helped in escaping local minima more effectively. Seeing the improvement after increasing the learning rate from 0.01 to 0.02, and noting that the results were better, made us consider further testing with higher learning rates. Further testing was con-

ducted by adjusting the learning rate to 0.05, 0.075, and 0.1. The results were as follows:

- Test accuracy for 0.02: 94.16% with a loss of 0.158
- Test accuracy for 0.05: 96.91% with a loss of 0.101
- Test accuracy for 0.075: 96.56% with a loss of 0.112
- Test accuracy for 0.1: 96.39% with a loss of 0.126

These results indicate that higher learning rates produce models with better accuracy, with the optimal learning rate being around 0.05.

Learning Rate Scheduler

ReduceLROnPlateau: The ReduceLROnPlateau scheduler [8] dynamically adjusted the learning rate based on the validation loss. If the validation loss did not improve for a specified number of epochs (patience), the scheduler reduced the learning rate by a factor. This adaptive approach allowed the model to start with a higher learning rate for rapid initial learning and then reduce the rate to fine-tune the weights more delicately. The introduction of this scheduler prevented overfitting by avoiding excessive adjustments when the validation loss plateaued. It also helped in finding a more stable and lower loss region in the loss landscape.

The combined effect of these changes led to an improvement in accuracy from 93.81% to 96.91% and a reduction in loss from 0.166 to 0.101. These adjustments facilitated better weight updates, improved the model's generalization, and enhanced its ability to capture subtle variations in the speech data.

The performance metrics included precision, recall, and F1-score, all of which indicated a balanced and robust classification across the three emotion classes (low, neutral, high). The average F1-score of 0.97 confirmed the model's effectiveness in both identifying true emotions and minimizing misclassifications. See Figure 6 for the Confusion Matrix.

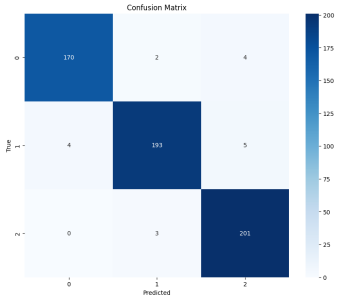

Figure 6. Confusions Matrix for Learning Rate 0.05

Table 2. Model Performance Metrics per Class (Macro-Averaged)

Class	Accuracy	Precision	Recall	F1-Score
0	0.9542	0.966	0.977	0.971
1	0.9646	0.955	0.975	0.965
2	0.9851	0.985	1.0	0.992
Overall (Macro-Averaged)	0.972	0.969	0.984	0.976

EQUATIONS FOR METRICS AND MODELS

1. Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

2. Precision

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

3. Recall

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

4. F1-Score

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

5. CNN Equation (Convolutional Layer)

The convolution operation in a CNN layer can be expressed as:

$$z_{i,j,k} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{i+m,j+n} w_{m,n,k} + b_k \tag{5}$$

where:

* $z_{i,j,k}$: Output feature map at location (i, j) for channel k. * $x_{i+m,j+n}$: Input feature map at location $(i+m, j+n)$. * $w_{m,n,k}$: Weight of the filter (kernel) at location (m, n) for channel k. * b_k: Bias term for channel k.

6. ReLU Equation (Activation Function)

The Rectified Linear Unit (ReLU) activation function is a popular choice in CNNs due to its simplicity and effectiveness. It is defined as:

$$ReLU(x) = \max(0, x) \tag{6}$$

7. LSTM Equations (Long Short-Term Memory)

LSTMs are designed to capture long-term dependencies in sequential data. Their core equations are:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{7}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{8}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{9}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{10}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{11}$$

$$h_t = o_t * \tanh(C_t) \tag{12}$$

where:

* f_t: Forget gate (controls how much past information to forget) * i_t: Input gate (controls how much new information to let in) * $\tilde{C}_t$: Candidate cell state (potential new information to store) * C_t: Cell state (the LSTM's internal memory) * o_t: Output gate (controls how much of the cell state to output) * h_t: Hidden state (the LSTM's output) * W_f, W_i, W_C, W_o: Weight matrices * b_f, b_i, b_C, b_o: Bias vectors

FUTURE WORK

Future work will extend beyond the current scope to explore advanced architectures, such as Kolmogorov-Arnold networks [16], aimed at further enhancing emotion recognition accuracy. Additionally, efforts will be directed towards expanding the dataset to encompass a broader spectrum of nuanced emotions, thereby enriching the model's understanding and classification capabilities. Furthermore, research endeavors will delve into real-time emotion recognition, aiming to enhance the model's applicability in dynamic environments and interactive systems.

Moreover, beyond emotion recognition, there is potential for exploring additional tasks such as gender recognition utilizing the same model architecture. By leveraging the rich features extracted from speech data, our model could be adapted and fine-tuned to accurately classify gender, thereby broadening its utility and applicability in various domains. This expansion into multi-modal recognition tasks represents an exciting avenue for future research, promising advancements in both accuracy and versatility.

CONCLUSION

This study presents a thorough investigation into the effectiveness of a parallel CNN-LSTM model with attention mechanisms for emotion recognition [6] in Arabic speech. Through the utilization of the Basic Arabic Vocal Emotions Dataset [1] and the implementation of preprocessing methods, our model exhibited remarkable accuracy and resilience in classifying emotions. Noteworthy adjustments to the learning rate and the integration of the ReduceLROnPlateau scheduler [8] played pivotal roles in fine-tuning the model's performance, highlighting their significance in enhancing emotion recognition accuracy.

ACKNOWLEDGMENT

We would like to express our deepest gratitude to Dr. Tmam Alsarhan for her invaluable guidance and support throughout the process of writing this paper. Her unwavering belief in our potential, combined with her exceptional teaching skills, allowed us to delve into the complexities of pattern recognition and truly grasp its significance. Her patience, encouragement, and willingness to share her extensive knowledge have been instrumental in our success. We are immensely grateful for the opportunity to have learned from such a remarkable mentor.

REFERENCES

1. a13x10. Basic arabic vocal emotions dataset, 2024.
2. Sherif Mahdy Abdou and Abdullah M Moussa. Arabic speech recognition: Challenges and state of the art. *Computational linguistics, speech and image processing for arabic language*, pages 1–27, 2019.
3. Soumith Chitta Gregory Chan Edward Yang Zachary DeVito Zachary Lin PyTorch contributors Adam Paszke, Sam Gross. Pytorch: An imperative style, high-performance deep learning library, 2019.
4. KLU AI. Accuracy, precision, recall, f1. https://klu.ai/glossary/accuracy-precision-recall-f1#:~:text=Accuracy%20measures%20the%20overall%20correctness,metric%20for%20evaluating%20classification%20models., 2023.
5. Abdallah Al-Faham and Nada Ghneim. Towards enhanced arabic speech emotion recognition: comparison between three methodologies. *Asian J. Sci. Technol*, 7(3):2665–2669, 2016.
6. Shouyan Chen, Mingyan Zhang, Xiaofen Yang, Zhijia Zhao, Tao Zou, and Xinqi Sun. The impact of attention

mechanisms on speech emotion recognition. *Sensors*, 21:7530, 11 2021.

7. Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John G Taylor. Emotion recognition in human-computer interaction. *IEEE Signal processing magazine*, 18(1):32–80, 2001.
8. PyTorch Developers. Reducelronplateau — pytorch 2.3 documentation. 2024.
9. Kazuyuki Hara, Daisuke Saito, and Hayaru Shouno. Analysis of function of rectified linear unit used in deep learning. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2015.
10. Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
11. Mohammed Hassanain. Baved: Parallel cnn-attention-lstm, 2024.
12. Yasser Hifny and Ahmed Ali. Efficient arabic emotion recognition using deep neural networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6710–6714. IEEE, 2019.
13. IEEE Signal Processing Society. What are the benefits of speech recognition technology? *IEEE Signal Processing Society Blog*, 2024.
14. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
15. Lei Li, Yabin Wu, Yihang Ou, Qi Li, Yanquan Zhou, and Daoxin Chen. Research on machine learning algorithms and feature extraction for time series. In *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*, pages 1–5. IEEE, 2017.
16. Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.
17. Ekaterina Lobacheva, Eduard Pockonechnyy, Maxim Kodryan, and Dmitry Vetrov. Large learning rates improve generalization: But how large are we talking about? *arXiv preprint arXiv:2311.11303*, 2023.
18. Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *International Conference on Machine Learning*, pages 23803–23828. PMLR, 2023.
19. Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python, 2015.
20. Wes McKinney and The pandas development team. pandas: Powerful python data analysis toolkit, 2024.
21. Mohamed Meddeb, Hichem Karray, and Adel M Alimi. Speech emotion recognition based on arabic features. In *2015 15th international conference on intelligent systems design and applications (ISDA)*, pages 46–51. IEEE, 2015.
22. Ali Mefiah, Yousef A Alotaibi, and Sid-Ahmed Selouani. Arabic speaker emotion classification using rhythm metrics and neural networks. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1426–1430. IEEE, 2015.
23. Maury Microwave. What is additive white gaussian noise why is it important for test measurement? *Maury Microwave Blog*.
24. Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.
25. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python, 2011.
26. Project Jupyter. Jupyter notebook. https://jupyter.org/, Accessed 2024.
27. BZJLS Thornton. Audio recognition using mel spectrograms and convolution neural networks. 2019.
28. Yue Wang, Guanci Yang, Shaobo Li, Yang Li, Ling He, and Dan Liu. Arrhythmia classification algorithm based on multi-head self-attention mechanism. *Biomedical Signal Processing and Control*, 79:104206, 2023.
29. Xiaochun Yin, Zengguang Liu, Deyong Liu, and Xiaojun Ren. A novel cnn-based bi-lstm parallel model with attention mechanism for human activity recognition with noisy data. *Scientific Reports*, 12(1):7878, 2022.
30. Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference,*

Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13, pages 818–833. Springer, 2014.