applied sciences

MDPI

Article

An Approach for Pronunciation Classification of Classical Arabic Phonemes Using Deep Learning

Amna Asif [1,*], Hamid Mukhtar [2], Fatimah Alqadheeb [3], Hafiz Farooq Ahmad [3] and Abdulaziz Alhumam [3]

[1] Information Systems Department, College of Computer Sciences and Information Technology (CCSIT), King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia
[2] Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; h.mukhtar@tu.edu.sa
[3] Computer Science Department, College of Computer Sciences and Information Technology (CCSIT), King Faisal University, P.O. Box 400, Al-Ahsa 31982, Saudi Arabia; 218002332@student.kfu.edu.sa (F.A.); hfahmad@kfu.edu.sa (H.F.A.); aahumam@kfu.edu.sa (A.A.)
* Correspondence: aarkhan@kfu.edu.sa

Abstract: A mispronunciation of Arabic short vowels can change the meaning of a complete sentence. For this reason, both the students and teachers of Classical Arabic (CA) are required extra practice for correcting students' pronunciation of Arabic short vowels. That makes the teaching and learning task cumbersome for both parties. An intelligent process of students' evaluation can make learning and teaching easier for both students and teachers. Given that online learning has become a norm these days, modern learning requires assessment by virtual teachers. In our case, the task is about recognizing the exact pronunciation of Arabic alphabets according to the standards. A major challenge in the recognition of precise pronunciation of Arabic alphabets is the correct identification of a large number of short vowels, which cannot be dealt with using traditional statistical audio processing techniques and machine learning models. Therefore, we developed a model that classifies Arabic short vowels using Deep Neural Networks (DNN). The model is constructed from scratch by: (i) collecting a new audio dataset, (ii) developing a neural network architecture, and (iii) optimizing and fine-tuning the developed model through several iterations to achieve high classification accuracy. Given a set of unseen audio samples of uttered short vowels, our proposed model has reached the testing accuracy of 95.77%. We can say that our results can be used by the experts and researchers for building better intelligent learning support systems in Arabic speech processing.

Keywords: deep learning; classical Arabic; short vowels; audio dataset; convolutional neural networks; optimization; regularization

Citation: Asif, A.; Mukhtar, H.; Alqadheeb, F.; Ahmad, H.F.; Alhumam, A. An Approach for Pronunciation Classification of Classical Arabic Phonemes Using Deep Learning. *Appl. Sci.* **2022**, *12*, 238. https://doi.org/10.3390/app12010238

Academic Editors: Aida Valls and Keun Ho Ryu

Received: 11 November 2021
Accepted: 21 December 2021
Published: 27 December 2021

1. Introduction

Speech processing technology has received considerable attention recently due to a variety of applications in the areas of automated speech recognition, information retrieval, and assisted communication. A lot of diverse research work has been done on speech processing for different human languages globally. As such, in recent years, deep learning has increasingly enabled autonomous speech processing including speech recognition and synthesis. However, the Arabic language has witnessed less research work in this domain due to its unique challenges.

Arabic is the fifth widely used language in the world and there are around 422 million speakers of Arabic as their first language on the globe [1]. In the broader term, Arabic language can be categorized into Classical Arabic (CA) and modern standard Arabic (MSA) dialects. MSA is a modified version of CA currently used in everyday communication in Arabic speaking countries. Classical Arabic is the language of the Holy Quran [2] and is still used largely in religious context and studies despite being more than 1400 years old.

Millions of Arabic and non-Arabic speakers around the world practice CA in their daily routine in the form of recitation of the Holy book or studying in a formal educational setting.

1.1. Arabic Phonemes and Their Pronunciation

A phoneme is the smallest unit of sound in human speech [3]. Phonemes include all the distinct units of sound spoken in a language. The Arabic language consists of 34 phonemes that can be broken down into 28 consonants, three short, and three long vowels [4] known by their common names of Fatha, Damma, and Khasra. Pronunciation is a general term that includes several distinct features present in human languages. The correct pronunciation is hard and challenging to measure because there is no universal definition for correctness in the context of human languages, but there has been some research in this domain [5,6].

In the Arabic language, the accurate pronunciation of phonemes is required in learning the language to void change in the meaning of the sentence. In many countries around the world, the CA is part of the schools' syllabus. Correct pronunciation of phonemes as per defined rules is an essential requirement to preserve the meaning of the words [7,8]. Mispronunciation results in two types of errors; firstly, it changes the meaning of the word completely. Secondly, it defies the rules of pronunciation, and both errors are forbidden in CA. Learning the correct pronunciation of Arabic alphabets is challenging, and it requires each learner to follow an individual teacher, who listens and corrects the mistakes separately for each learner [9]. These corrections belong to the pronunciation of all the 84 variations of phonemes. A major distinction of the CA from MSA is the emphasis on the correct use of vowels. When speaking CA, the speakers tend to produce all the sounds according to the existing rules for CA pronunciation. CA has precise and explicitly defined rules for correct pronunciation to conserve the accurate meaning of the words and provides a framework to facilitate both the natives and non-natives learning the language [10]. These rules are standardized, widely available, and recognized by the Arabic speaking world [11]. In CA, alphabets' articulation points and characteristics and massive practicing of vocals play a significant role in correct pronunciation. On the contrary, it has been studied previously that in the case of MSA, the different dialects such as Saudi, Egyptian, and Sudanese have different qualities of vowel pronunciation [12].

1.2. Motivation for Pronunciation Classification

In a typical CA learning institute, a single teacher may be responsible for listening to and correcting the mistakes of several dozen learners on a single day. This is quite a laborious job leading to fatigue and is prone to diminished error correction by the instructor over time. Moreover, each student has to wait for their turn to speak or read the learned parts. If this job can be automated by letting the students' work be evaluated through an automated mechanism, it will improve the efficiency of the learning process and the overall productivity of the students and instructors. Moreover, this will open new ways of distant and home-based learning with little reliance on the presence of an instructor. As learning can take place anytime and anywhere, it is required to develop automatic approaches that can detect pronunciation errors and give feedback to the learner.

It is worth mentioning that in the Arabic language, the correct pronunciation is vital and requires a lot of practice by the learner of the language. Currently, the students not only rely on in-class participation but also practice using digital technology. In this regard, we have evidence of the development of many applications supporting the students' learning process. There is a need for research in developing systems and applications for improving Arabic reading and speaking knowledge. In various parts of the world when there is an unavailability of real instructors, such applications play an important role in supporting the student's learning process. The improved model can easily be integrated with existing applications to increase the accuracy of Arabic short vowels classification in the learning process. The Arabic-speaking countries are a recent place of interest for many tourists around the world. Therefore, many people are interested to learn this

language for their survival in this part of the world. Any applications developed using better speech recognition models will be beneficial for people who want to learn Arabic using the available online resources.

1.3. Objectives and Contributions

This research aims to develop a classification system for correct and incorrect pronunciation of the Arabic short vowels. It is a novel research idea focusing on the subtle pronunciation differences in Arabic speech processing. The outcomes of this research can be useful in developing advanced systems that can autonomously classify words and sentences to effectively facilitate CA learning with accurate pronunciation. As there are 28 alphabets in the Arabic language and each alphabet has three possible vowel states, they make a total of 84 unique phonemes. Thus, given audio that utters any of these 84 phonemes, our task is to accurately map the audio to the correct phoneme.

Although there are a few existing datasets related to Arabic pronunciation, there is no dataset that serves our purpose of containing the different possible vowel pronunciations. Thus, we created a new dataset of the recordings of the Arabic alphabets through an online audio recording system. After passing the data through various stages of preprocessing, we augmented the available audio data by generating synthetic audio to make a sufficiently bigger audio dataset. We trained a deep convolutional neural network (CNN) over this data for audio classification. The trained network can classify unseen audio data with a testing accuracy of 95.77% into one of the 84 classes. Our research is different from the previous works in terms of the dataset, features engineering, proposed architecture, and evaluation performance.

The major contributions of this work are:

1. Collection of an audio dataset for the Arabic alphabet focused on the three states of vowels for each alphabet.
2. Classification of Arabic short vowels by recognizing the correct short vowels from a recorded phoneme.
3. Constructing a general CNN architecture for phoneme classification. This allows replicating the architecture for similar tasks or a different number of classes.
4. Sharing our experience of model optimization and fine-tuning with the researchers and practitioners to aid their knowledge of building better models in the future.

The remainder of the article is organized as follows. Section 2 discusses the related work. Section 3 describes the materials and methods used in this research. Section 4 explains the development of the CNN model for Arabic short vowel classification and describes the techniques of data augmentation and fine-tuning applied to improve the model's performance. Section 5 reports the results of the classification. Section 6 contains discussions in view of the related work and implications of our work. Section 7 concludes this article.

2. Related Work

There are several research studies on using machine learning and deep learning for speech recognition and pronunciation detection of various types of Arabic datasets. Many of such techniques used audio features for further audio processing and classification. Several tools have been designed and developed to support Arabic learning and teaching. In the following, we report the previous research contributions in the above-mentioned areas.

2.1. Pronunciation Detection

In the direction of pronunciation detection (correct pronunciation and mispronunciation), the previous studies have investigated the machine learning techniques of SVM (support vector machine), KNN (k-nearest neighbors) and NN (neural network) [13], and DNN (deep neural network) techniques of CNN (convolutional neural networks) with transfer learning, AlexNet, BLSTM (bidirectional long short-term memory) [10]. The machine learning algorithms for pronunciation detection had achieved an accuracy of 74.37%

for KNN, 83.50% for SVM, and 90% for NN. Similarly, the DNN techniques reported that CNN, AlexNet, and BLSTM give 95.95%, 98.41%, and 88.32% accuracy for recognizing each alphabet, respectively. As well as detecting the quality of pronunciation of each alphabet using CNN, AlexNet, and BLSTM attained the accuracy of 97.88%, 99.14%, and 77.71%, respectively. In DNNs experiments, the dataset was limited to 29 classes of pronunciation, excluding the vowels. Therefore, it is required to fill this gap to support teaching and learning of CA basics. DNN algorithms and models such as CNN consist of multiple hidden layers capable of efficiently extracting important features from a large set of data. These features can be transformed from one layer to the next in a series of several layers with varying weights in the neural network until it results in a set of layers that can be used to initialize a deep learning algorithm for speech recognition [14]. A proposed method of a DNN based on articulatory models, with a multi-label learning scheme, shows promising results in speech error detection [15]. The researchers consider measuring all attributes responsible for generating the sound related to the movement of the tongue, lips, and other organs. This experiment observes 74% of accuracy for pronunciation error detection. We have found significant improvements in pronunciation detection accuracy in DNN techniques compared to machine learning algorithms from the results.

Besides machine learning and DNN techniques, pronunciation detection was performed using statistical methods. In [16], a system was proposed to detect how badly an Arabic word was pronounced using different scores of pronunciation measurement. The system used the GLL (Global Average Log Likelihood) score, the LLL (Local Average Log Likelihood) score, the RoS (Rate of Speech score), and the RoA (Rate of Articulation) score to assess the pronunciation quality of the learner quantitatively. The dataset consisted of three Arabic corpora, spoken by six young Algerian learners. The pronunciation of expert learners is used as a benchmark to assess the other five learners. Evaluation measurement was used to decide whether the score calculated by the system could detect mispronunciation; the researchers used the CA (Correct Acceptance), CR (Correct Rejection), FA (False Acceptance), and FR (False Rejection). The results showed that the system could detect mispronunciation, using the GLL score method, with 86.66% of correct rejection, and the GLL had the higher CA + CR (76.66%) and the lower FA + FR (23.32%).

In the direction of pronunciation detection of non-native speakers, mispronunciation of Arabic phonemes has been investigated for non-native speakers by analyzing the Arabic speech of Pakistani and Indian speakers from the KSU (King Saud University) database [17]. Research findings of this study highlight that non-native Arabic speakers often mispronounce five Arabic phonemes. The system trained with native and non-native speakers of Arabic phonemes and tested with only non-native speakers. A threshold was set to be compared with the calculated score of GOP (Goodness of Pronunciation) to decide whether the phoneme was pronounced correctly or not. Five experiments were conducted to set up the suitable parameters for the system using HMM (Hidden Markov model). The system used 16 mixtures with 19 HMM re-estimation. Moreover, it extracted 12 MFCC (Mel-frequency Cepstral Coefficients) from sound data. The result of the GOP showed a high accuracy from 87% to 100%, and the false rejection was zero to less than 10%. HMM for automatic speech recognition system has been proposed to help improve the pronunciation for Malaysian teachers of the Arabic language [18]. The aim is to develop a computer system for standard Arabic pronunciation learning by estimating the pronunciation score based on the HMM log-likelihood probability model. The system is designed to extract feature vectors from speech utterances using the MFCC technique; then, the Baum Welch Algorithm is applied to train the system and build the HMMs set. The pronunciation scoring system uses HMMs with the test speech features to perform classification of the speech utterances by applying the Viterbi Algorithm to calculate word pronunciation score. The dataset consisted of 200 words recorded by 20 native Arabic speakers and 10 non-native speakers. The accuracy performance of the proposed system was 89.69%.

2.2. Audio Features

In audio classification, features identification and extraction for audio processing is one of the most popular techniques. A system was developed in [19] to recognize the *Makhraj* (the areas of the mouth from which the Arabic alphabets are pronounced) pronunciation using MFCC features extraction techniques to build a database of features from the audio dataset. Then, the SVM classifier has been used to classify the Arabic alphabet's *Makhraj* pronunciation. The system is trained with the recorded audio of the Arabic alphabet *Makhraj* pronunciation. For new input data, the system extracted the features and matched them with the trained data. Then, it was classified and analyzed using the SVM method with RBF (Radial Basis Function) kernel. The audio data used in the research is a collection of 28 Arabic alphabets' audio and 12 features coefficients were extracted to distinguish between *Makhraj* pronunciation of Arabic alphabet. Different waveforms analysis is used to present the audio data, using audio visualization, FFT (Fast Fourier Transform), and Mel waveform. The result showed that using audio visualization, all letters had a similar representation. On the other hand, using FFT and Mel waveform, each Arabic *Makhraj* pronunciation showed different representations, which can be used to distinguish between different Arabic alphabets. Another research study proposed a CNN feature-based model to detect mispronunciation in Arabic words [20]. The proposed system extracted features from different layers of the AlexNet network. Researchers collected Quranic verses words that cover all Arabic alphabet letters 30 times by speakers of different ages. The participators of the collected dataset were native/non-native Arabic speakers. After removing the noise from the dataset, it converted to a 2D spectrogram and was used to input the CNN model. Then, discriminative features were extracted from fully connected layers 6, 7, and 8 of AlexNet containing high dimensions features. This method showed a significant result compared with the CNN model. It achieved 85% on the complete Arabic dataset. The MDD (Mispronunciation Detection and Diagnosis) task was performed using the RNN (Recurrent Neural Network) [21] and CTC (Connectionist Temporal Classification) model. The model consisted of five parts, the input layer, which accepts the framewise acoustic features. Then is the convolution, which contains a total of four CNN layers and two max-pool layers. The third part is a bi-directional RNN that captures the temporal acoustic features. The fourth part is MLP layers (Time Distributed Dense layers), which ends with a soft-max layer for the classification output. The last part is the CTC output layer that generate the predicted phoneme sequence. The experiment results showed that the proposed approach significantly outperformed previous approaches. Some researchers used LDA (Linear Discriminant Analysis) to classify the data into the correct class and draw a decision region between the given classes [22]. The first step in this system is preprocessing speech signals, which helps prepare the data for the next processing. The preprocessing includes end-point detection, pre-emphasis, and normalization. The second step is features extraction and uses MFCC techniques to extract different coefficients order 12, 20, and 35. The main contribution of this research is to test a different number of MFCC coefficients with varying percentages of training and testing in LDA. The best performance achieved is 92% for the Arabic phoneme (*Taa*) when using 35 MFCC coefficients and 80% of training data. A recent work uses APDM (Acoustic and Phonetic Decoding Model) for recognizing vowels for naturally uttered MSA-based Gas (Genetic Algorithms) [23]. They use MFCC and the LPC (Linear Prediction Coding) techniques to obtain speech parameters from the speech signal. GA based on Manhattan distance decision rule is applied on several Algerian male and female speakers' recordings and classify phonemes with accuracy of 98.02%. The studies that used the audio features data require extra preprocessing efforts to identify and extract the features from the audio dataset for further processing. Therefore, these methods are costly in terms of processing performance.

2.3. Tools in Arabic Learning

Many previous studies have developed tools for recitation assessment; in this regard, the HAFSS system [24] took user recitation of some Holy Quran phrases (in Arabic) as an

input and then assessed the quality of the users' recitation. It provides feedback messages to help users know their pronunciation errors and improve their recitation. This system included a speech recognizer to detect errors in user's recitation. For each decision from the speech recognizer, there is a confidence score that is used to choose the suitable feedback. One of the main components in the system is the automatic generation of the pronunciation hypotheses model, which is used to generate pronunciation errors in user's recitation and detect the pronunciation patterns. The system has been tested in a school with two student groups, and the results showed improvement in students' recitation when using the HAFSS system. The performance of the students has been increased from 38% to 77% while getting lessons from a teacher and using the HAFSS system to practice recitation. In [25], a tool was developed that detects pronunciation errors in young Algerian students. The idea was to differentiate between young Algerian students who have difficulties in pronunciation from those who have standard pronunciation. Since the native language of Algerian is a delicate Arabic language, it is very difficult for Algerians to formulate the equivalent sound in standard Arabic. The researchers proposed a system based on a decision tree that provides a decision for the pronunciation of Arabic if it is correct. Moreover, the system provides feedback if the pronunciation contains articulations problems to enhance the pronunciation skills of the learners—the system trained with the acoustical model built on MFCC representation and HMM models. Three scores were used for the decision tree, first the GLL (global average log-likelihood). The second score was the TDS (Time Duration of the Speech), the total time to produce the sound, and the total number of phonemes of each pronounced word. All the scores are input to the decision tree to accept or reject the pronunciation sound. The system was trained on correct Arabic pronounced words and tested with eight young Algerian students who read 16 Arabic words to test the system. The result showed a 95.8% TPR (true positive rate) for good pronunciation and 88.4% for bad pronunciation. Their dataset was too specific and smaller in size. So, more data with further experiments would be needed to validate their approach. Arafa et al. [9] developed a system for teaching Arabic phonemes employing ASR (Automatic speech recognition) by detecting mispronunciation and giving feedback to the learner. In the experimental study, the authors recorded Arabic phonemes 10 times from 89 elementary school children, which resulted in 890 recordings for each Arabic phoneme. The previous studies supported the fact the automated tools are beneficial in support the CA learning process and improve the students' performance.

2.4. Audio Datasets in CA

From the previous research, we found that many studies have used a variety of datasets in developing CA audio processing systems. However, many of such studies are limited to Arabic alphabets and some basic words. Although, the challenging task in learning Arabic pronunciation is the correct use of Arabic short vowels. ASR is a way of automatically transcribing the speech into text [26]. One of the major challenges for ASR for Arabic is the predominance of non diacritized text material where diacritics [27] is the use of vowels (e.g., short vowels Fatha, Damma, and Khasra) for both acoustic and language modeling that can change the meaning of the sentence. However, the majority of the acoustic features for ASR are available without diacritized form [26]. It means that there is no vowel information, which results in the loss of variations in the pronunciation of the speech. In this scenario, it becomes difficult to train a reliable acoustic model without knowing short vowels. Authors report encouraging results for classifying Arabic phonemes, but they do not consider vowels in their study. The research on the Arabic speech for vowels focuses that the MSA [28] used the formants and consonant-vowels-consonant (CVC) utterances to identify vowel similarities and differences. The HMM technique was applied to classify vowels, and the resulting performance of phonetic features of vowels was analyzed. Al-Anzi and AbuZeina [28] highlighted that different researchers had considered different phonemes for the Arabic language; for instance, some take 34 phonemes (28 consonants and six vowels)

while others take 112 phonemes by considering four diacritics for each alphabet. So, indeed, Arabic short vowels are crucial in CA; however, the research is limited in this area.

It has been emphasized that autonomous Arabic speech recognition poses challenges due to the vast lexical forms of vowels, which are semantically associated with a word in a sentence [2,29]. Thus, the correct classification of Arabic alphabets pronunciation with vowels is another important research challenge. None of the above research work has considered the correct pronunciation of the CA alphabet with vowels. Furthermore, the research is also limited in preprocessing performance due to features identification and extraction tasks. Therefore, it is required to improve the existing methods and algorithms for Arabic speech classification.

2.5. Summary of the Literature

The survey of the literature shows that the existing work in Arabic pronunciation can be categorized as mispronunciation detection [9,13], sometimes with a focus on non-native speakers [17,18], speech error detection [15], correct pronunciation detection [10], and detecting the similarities and differences between pronunciation of vowels and consonants for MSA [30]. As such, there have been a number of approaches from articulatory models [15] and transfer learning [10] to CNN [13] and acoustic models [31]. To the best of our knowledge, we could not identify any work that can detect the correct pronunciation of Arabic alphabets vowels in CA. One possible reason of not undertaking this work previously maybe because it makes a total of 84 cases to be distinguished from one another. This task is not only challenging in terms of data availability and retrieval but also in terms of model development. To carry out such a challenging task will remove many barriers in the correct utterance of CA words, which can benefit millions of people.

In this research work, we take up this challenging task and propose a deep convolutional neural network algorithm for the classification of Arabic alphabets with vowels. Our research is different from the previous works in terms of the dataset, features engineering, proposed architecture, and evaluation performance.

3. Materials and Methods

After exploring the existing literature, we have identified limited contributions in short Arabic vowels classification; because there was no existing dataset available, we began by collecting the Arabic short vowels audio data. We used a convenience sampling technique whereby the acquaintance, students, and faculty members were identified through personal contacts and social media. The data was collected in the form of audio clips from each participant.

3.1. Data Collection

One way to collect the audio dataset was to use an online audio recording tool like Phonic.ai (http://www.phonic.ai, accessed on 10 November 2021) website [32]. The website provides a service for collecting the participants' responses via sound recording. The website first collects user information through a survey followed by the instructions to record their audio. Then, demographic data such as age and gender are collected from the participants. The users also specify if they are native or non-native Arabic speakers. The survey is followed by a permission agreement to participate in this study; the participants must agree to continue and complete the survey. Then a sample of sound records for the Arabic alphabets with short vowels is played for the participants to listen and understand how their voice should be recorded. Finally, the recording page shows a video containing all the Arabic alphabets with a sequence of short vowels arranged in order and displaying them one by one with a gap of three to five seconds. The video lasts for two minutes and forty-nine seconds. Moreover, some participants shared their sound recordings through WhatsApp messenger application after reading and understanding the instructions.

The total number of the received audio was 85 individual recordings from 42 males to 43 females. There were 81 native Arabic speakers, and four non-native speakers. The

received audio files were saved using a naming scheme consisting of the speaker identification code, their gender, native or non-native specification, the age range of the participant, the source of data collection, and the date of the recording. The dataset had 6229 records belonging to 84 classes. It contained an average of 74 examples per class. The data set was imbalanced in terms of the number of examples per class.

Table 1 describes the summary of statistics for the obtained recordings. The dataset is available in the form of audio clips on Kaggle (https://www.kaggle.com/amnaasif/arabic-short-vowels-audio-dataset, accessed on 10 November 2021) for researchers and interested stakeholders.

Table 1. Summary of data collection from participants.

Gender	Status	Number of Records	Age Distribution
Male	Native Non-native	40 2	7–50+ years old
Female	Native Non-native	41 2	7–40 years old

3.2. Data Preprocessing

After data collection, we applied the following preprocessing steps to the audio files.

Noise Reduction: The received audio recordings contained various noise and background sounds, as they were collected in different environments. The Audacity (https://www.audacityteam.org, accessed on 10 November 2021) software was used to identify the noise in our dataset and helped in reducing it.

Data Segmentation, Resampling, and Silence Truncation**:** Each recorded audio by a participant contained all the pronunciations, which were to be segmented into separate pronunciations for each vowel. This was done using audio segmentation based on the silence between each segment. We made all the segments in equal time duration of one-second padding the shorter one with silence. Since the received recordings were collected from different sources, the audio sampling rate varies from one sample to another; so, the audio recordings were resampled to a 16 kHz sample rate. Finally, we tested the segments of audio files to ensure intelligibility and clarity of pronunciation by listening to each segment and removing any sounds that were not clear or had the incorrect pronunciation of an alphabet.

Data Labeling: To label each recorded instance belonging to a class, each short vowel is coded with a unique number for further processing. Table 2 presents the short vowels and their class labels.

3.3. Spectrogram Conversion

Instead of using audio signals as input, we converted each input instance into its equivalent waveform and then to a spectrogram of 32 $\times$ 32 pixels. "Spectrograms are 2D images representing sequences of spectra with time along one axis, frequency along with the other, and brightness or color representing the strength of a frequency component at each time frame" [33]. The advantage of spectrograms over audio signals is that they retain more information than the hand-crafted features for audio analysis, they are of lower dimension than the raw audio and have found their role in neural networks [33].

Table 2. Representation of Arabic Short Vowels (ASV) with Class Label (CL) in IPA and native script along.

Arabic short vowel	أَ	إِ	أُ	بَ	بِ	بُ	تَ	تِ	تُ	ثَ	ثِ	ثُ	جَ	جِ
Class label	1	2	3	4	5	6	7	8	9	10	11	12	13	14
IPA symbol	aa	ai	au	Ba	bi	bu	Ta	Ti	Tu	θ_a	θ_i	θ_u	dʒa	dʒi
Arabic short vowel	جُ	حَ	حِ	حُ	خَ	خِ	خُ	دَ	دِ	دُ	ذَ	ذِ	ذُ	رَ
Class label	15	16	17	18	19	20	21	22	23	24	25	26	27	28
IPA symbol	dʒu	ħa	ħi	Ħu	xa	xi	Xu	Da	Di	du	ða	Đi	ðu	ra
y Arabic short vowel	رِ	رُ	زَ	زِ	زُ	سَ	سِ	سُ	شَ	شِ	شُ	صَ	صِ	صُ
Class label	29	30	31	32	33	34	35	36	37	38	39	40	41	42
IPA symbol	ri	ru	za	Zi	zu	sa	Si	Su	ʃa	ʃi	ʃu	$s^ʕa$	$s^ʕi$	$s^ʕu$
Arabic short vowel	ضَ	ضِ	ضُ	طَ	طِ	طُ	ظَ	ظِ	ظُ	عَ	عِ	عُ	غَ	غِ
Class label	43	44	45	46	47	48	49	50	51	52	53	54	55	56
IPA symbol	$d^ʕa$	$d^ʕi$	$d^ʕu$	$t^ʕa$	$t^ʕi$	$t^ʕu$	$ð^ʕa$	$ð^ʕi$	$ð^ʕu$	ʕa	ʕi	ʕu	ʁa	ʁi
Arabic short vowel	غُ	فَ	فِ	فُ	قَ	قِ	قُ	كَ	كِ	كُ	لَ	لِ	لُ	مَ
Class label	57	58	59	60	61	62	63	64	65	66	67	68	69	70
IPA symbol	ʁu	fa	fi	Fu	qa	Qi	Qu	Ka	Ki	ku	la	Li	lu	ma
Arabic short vowel	مِ	مُ	نَ	نِ	نُ	هـ	هـ	هـ	وَ	وِ	وُ	يَ	يِ	يُ
Class label	71	72	73	74	75	76	77	78	79	80	81	82	83	84
IPA symbol	mi	Mu	na	Ni	nu	Ha	Hi	Hu	Wa	wi	wu	Ja	ji	ju

4. CNN Model for Arabic Short Vowels Classification

Inspired by existing approaches to developing CNN architecture, we started with an initial architecture consisting of five convolutional and two pooling layers. With our constructed dataset and the first CNN model, we achieved a training accuracy of 84.27% and validation accuracy of 40.15%. Based on previous approaches [34,35], and the low validation accuracy, it was evident that the small dataset could not give high classification accuracy. Because obtaining more user data was not an option, for achieving good accuracy, we developed a two-phase approach as shown in Figure 1. In the first phase, our focus was on improving the validation results as much as the initial training results by generating more training and validation data. Data augmentation is a useful technique to improve the performance of model and expand limited datasets to take advantage of deep learning models. The validation error must continue to decrease with the training error to develop a useful deep learning model. However, data augmentation is not only a technique to improve accuracy by avoiding the overfitting. Many other alternative solutions are model fine-tuning and hyperparameter tuning to achieve higher accuracy by avoiding the overfitting [36]. Therefore, once our desired validation accuracy of 85% could be achieved using a bigger dataset, in the second phase our focus was to improve the deep learning model to achieve a test accuracy of 95% on the test data. As can be seen in the figure, model development is an iterative process in each of these phases. In phase 2, after each cycle of training, validation, and testing, the results are analyzed, and the model is modified by fine-tuning it. At the same time, the hyperparameters are optimized to achieve the desired accuracy above 95%.

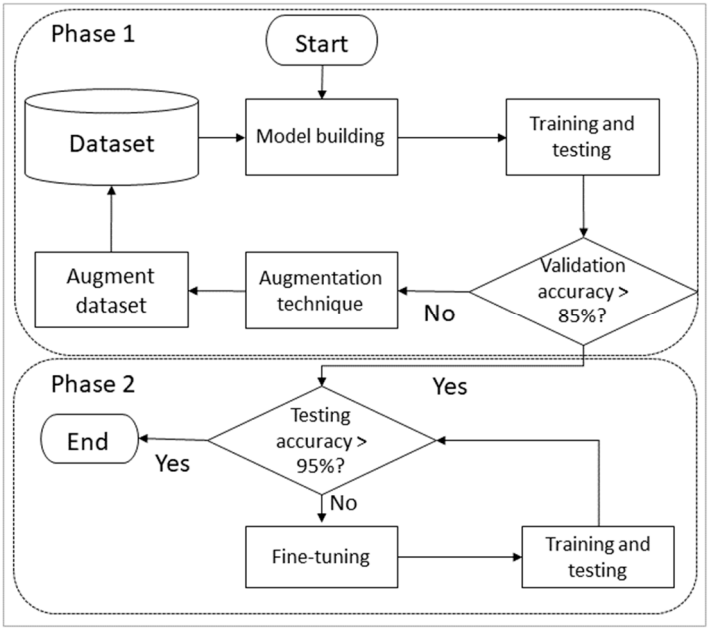


Figure 1. A two-phase approach of developing a CNN model for classifying Arabic short vowels.

4.1. Data Augmentation

Data augmentation is a technique for increasing the number of records in the dataset by slightly modifying the copies of original data for producing synthetic data using statistical methods and algorithms. Many data augmentation techniques exist in the literature [37], so we applied a few techniques one-by-one. Each time an augmentation technique was applied, it resulted in the gradual increase of the audio instances in the dataset; we also kept evaluating the validation accuracy of the base model, which was also improving. We used the augmentation techniques of noise injection, time-shifting, and changing the audio speed using time stretch, as shown in Figure 2. Other audio data augmentation techniques, such as pitch shift, were not ideal for our dataset because it required randomly changing the pitch, thus, distorting the pronunciation. The techniques are described as follows.

Noise injection: This technique introduces white noise in the audio dataset [38] as a ratio between the signal and noise. This method is appropriate for our model as we can assume that given environmental variations, user input is not noise-free in most circumstances. We applied random noise augmentation by adding two types of noise values X to audio files using the NumPy library in Python, with X = 0.005, and X = 0.0005. Figure 2b illustrates the noise injection results of an audio file, where noise is injected at the rate of X = 0.005.

Time-shifting: The method of time-shifting is applied to shift the audio forward or backward [38]. We used the roll method in Python's Numpy library to shift the start of an audio file S milliseconds. If D_a is audio data, and $D_a = [x_1, x_2, \ldots., x_n]$, by applying Silent $S_t = [S_1, S_2, \ldots., S_m]$, it becomes $D_{shifted} = [S_1, S_2, \ldots., S_m, x_1, x_2, \ldots., x_{n-m}]$, Where $S_t \leq D_s$. Figure 2c presents the result of the time-shifting function; the audio file is shifted forward for 2 milliseconds at the beginning of the graph by replacing it with silence.

Changing speed: This technique allows adjusting the speed of audio signal S by a certain rate R as $S = S \div R$. We used the values of R = [1.25, 1.4, 1.5, 1.6]. Figure 2d illustrates the result of changing the audio speed where R = 1.5. It is investigated in [39] that short vowels can be varied on shorter and longer duration.

After applying the data augmentation techniques, we ended up with a total of 49,829 audio files. With the augmented dataset, we achieved a validation accuracy of more than 85%.

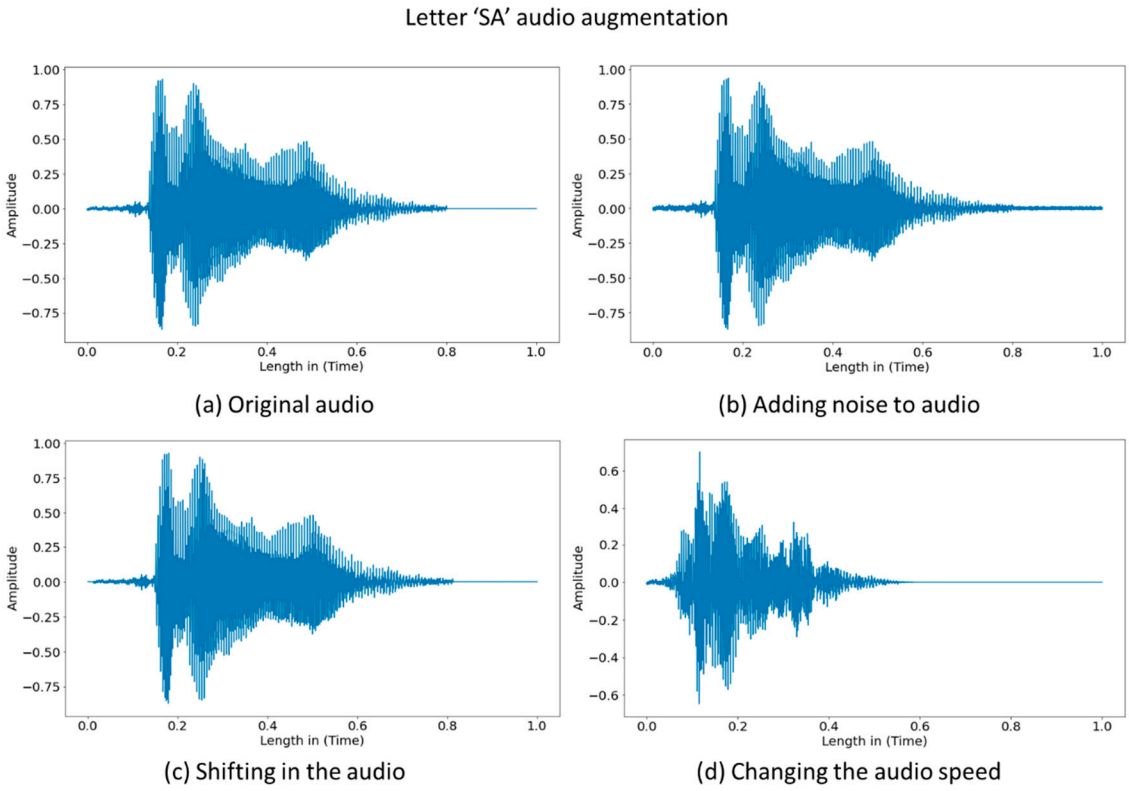


Figure 2. The results of different data augmentation techniques on an audio file: (**a**) the original audio data, (**b**) the audio signal after adding noise to original file, (**c**) the audio file after time-shifting, and (**d**) the audio file after increasing the speed.

4.2. Fine-Tuning the CNN Architecture

We started with a sequential CNN architecture with eight processing layers to develop our baseline model. The architecture is made up of convolutional layers such that each layer applies a set of convolution filters to an input followed by a non-linear activation function [40]. The convolutional layers are defined with a kernel of size 3 and the number of filters set to 32, 64, and 128 in different layers. A resizing layer is used to downsample each input to speed up the model training process. A normalization layer normalizes each pixel in the image using mean and standard deviation values. The other important layer of the CNN architecture is the pooling layer [41], and its function is to progressively reduce the input to decrease the number of parameters and increase the network performance. There are different types of pooling operations: max pooling, average pooling, global max pooling, and global average pooling used for downsampling of the input samples. The CNN can also have additional layers for optimization and improved performance. The batch normalization layer [42] allows normalizing the input of each layer, as the problem of internal covariate shift occurs due to constantly changing the distribution of activation, and each layer needs to learn to adapt to a new distribution. Similarly, the dropout layer [43] is used to reduce the model overfitting by randomly dropping out some percentage of the layer output. The flatten layer is used to convert the two-dimensional data into one-dimension for final classification by the dense layer.

Each layer has its associated activation function [44], whose task is to define how the weighted sum of input is transformed to the output from the nodes in a layer of the network. There are various activation functions, and the most popular ones are the ReLU (rectified linear unit), Sigmoid, and SoftMax. The ReLU function does not allow the activation of all the neurons simultaneously; when the output of the linear transformation is zero, the output neurons get deactivated. SoftMax function restricts the output values in the range 0 to 1, which are treated as probabilities of a particular class and usually used in the

last layer of the neural network. Mathematically, the SoftMax function σ is applied to a vector of inputs, $\vec{z}$, where each component of z is converted into corresponding probability according to its weight, e^{z_i} is standard exponential function for input vector, k stands for the number of classes and e^{z_j} presents standard exponential function for output vector shown in Equation (1) as follows:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad (1)$$

The dataset was divided into a ratio of 80% for the training set and 10% for validation and testing sets each. The batch size was set to 32. With the CNN architecture having eight layers, we achieved a testing accuracy of 80%. An improvement strategy [35] was to update the baseline sequential model by adding more convolutional, max pooling, and dense layers and updating the network to 12 layers. To find the optimal CNN model, we ran our model by applying a random search for the best parameters, which allows finding the optimal values of filter, kernel, and learning rate to achieve maximum accuracy. We found that the accuracy can be increased by increasing the filter size. We tested many CNN model modifications in a trial-and-error [35,45] manner and kept increasing the testing and validation accuracy in small steps. That allowed us to improve the testing accuracy to 90.0%.

4.3. Hyperparameters Tuning

In deep learning neural networks, the function of optimizers [46] is to reduce the losses to achieve the most accurate results possible. We evaluated different optimizers: Adam, Nadam, RMSprop, and SGD, and found Adam [47] to achieve the best performance in the proposed network. Adam stands for adaptive moment estimation that adaptively estimates the first and second-order moments. It updates the network weights iteratively based on the training data. Specifically, Adam uses the update vector $\hat{v}_t$ and the past gradient $\hat{m}_t$ differently than the previous algorithms, shown in Equation (2):

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \quad (2)$$

Here θ_t represents the weights and bias parameters, η stands for the learning rate or step-size, $\hat{m}$ and $\hat{v}$ represent the first and second moment vectors and by parameterizing them by t we get their moving averages over time. Adam combines the advantages of AdaGrad and RMSProp algorithms [26] and compared to other optimizers, it requires less memory, is computationally efficient, and is suited for problems with large data or many parameters.

We utilized the learning rate scheduler to dynamically adjust the learning rate to achieve the desired accuracy. Initially, the learning rate was set to 0.001, and after 80 Epochs, the learning rate was reduced by 10% every Epoch. We continuously reduced the learning rate because it helped in achieving optimal weight update by gradually maintaining the training loss and avoiding its oscillating over training epochs [48]. We utilized the early stopping hyperparameter for stopping the model prematurely if the loss was not decreasing anymore. We also used the Tensorboard tool to get training details for analysis and further improvements to the model through the generated time-series graphs, histograms, and distributions [49]. We added 2 batch normalization and 1 dropout layer to this model. These steps are explained in Algorithm 1, where, labeled audio dataset is input to the model that output as training loss, accuracy, and predicted labels. The audio data was first converted to a labeled waveform dataset. Then the transformed to labeled spectrogram dataset. The spectrogram dataset was resized to 32×32 pixels and input to our optimized sequential DLNN model. The model was executed by passing the callback functions of learning rate schedular (lr_scheduler), early stopping (callback_Early_stopping), best model checkpoint (model_checkpoint_callback), and tensorboard (tensorboard_callback).

Together, hyperparameter tuning and model fine-tuning helped in achieving the testing accuracy of 95%.

Algorithm 1: Steps of classification of Arabic short vowels on a fully optimized and fine-tuned neural network

```
Input
        aDataset = Audio dataset
        labels = Labels of classes
        train_files, val_files, test_files = split(aDataset(80,10,10))
Output
        Accuracy = Model accuracy
        Loss = Model learning loss
        y_pred = Predicted labels
Algorithm
Begin
        waveform_ds = Map waveform and labels from aDataset
        spectrogram_ds = Map spectrogram and labels from waveform_ds

        Function preprocess_dataset(files)
        output_ds = Map waveform_ds from files_ds of files
        output_ds = Map spectrogram_ds of output_ds
        Return output_ds
        Endfunction
        train_ds = preprocess_dataset from train_files
                val_ds = preprocess_dataset from val_files
        test_ds = preprocess_dataset from test_files

        input_shape = Shape of spectrogram in spectrogram_ds
        norm_layer = Normalization in preprocessing

        model = Sequential(input_shape, Resizing(32, 32), norm_layer, layers)
        trainNetwork train_ds, val_ds, callbacks =
        [lr_scheduler,callback_Early_stopping, model_checkpoint_callback,
        tensorboard_callback]

        load weights of best_model
        model train accuracy = Evaluate(train_ds)
        model val accuracy = Evaluate(val_ds)
        model test_accuracy = Evaluate(test_ds)
        Loss_graph metrics['loss'], metrics['val_loss']
        Accuracy_graph metrics['Accuracy'], metrics['val_ Accuracy']
End
```

Figure 3 shows the resulting architecture after fine-tuning and applying hyperparameter tuning to the baseline architecture. As can be seen, hyperparameters have been added in the form of batch normalization and dropout layers in addition to the optimizer selection and learning weight. FC1 and FC2 represent the two fully connected layers for classification followed by the SoftMax activation function that assigns a probability to each of the output classes.

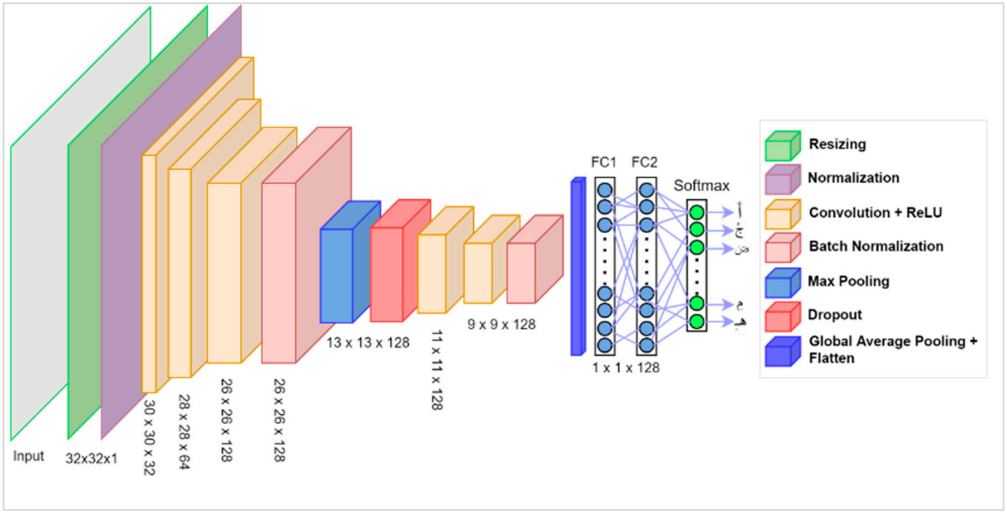


Figure 3. The architecture of the optimized CNN model.

4.4. Model Execution

The code is executed on a GPU-based desktop system with hardware configuration of Intel (R) Core (TM) i9 CPU @3.70, NVIDIA GeForce RTX 3070, and 64 GB RAM. The model is developed and run using the TensorFlow (https://www.tensorflow.org/, accessed on 10 November 2021) platform, mainly the Keras (https://keras.io/, accessed on 10 November 2021) API. The maximum number of epochs was set to 150. It took on average 91 s to complete one Epoch. We used the Jupyter notebook in conda (https://docs.conda.io/en/latest/, accessed on 10 November 2021) environment management system with miniconda installer for running the CNN model.

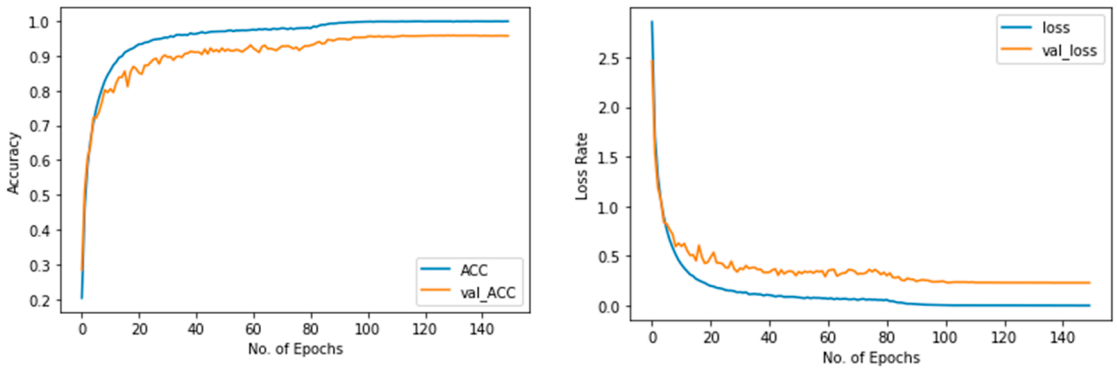
5. Results

In this study, we designed and executed four different experiments using combinations of two datasets and two model settings: (1) the original dataset trained on a sequential baseline CNN model, (2) the original dataset trained on an optimized and fine-tuned model, (3) the augmented dataset trained on the baseline CNN model, and (4) the augmented dataset trained on the optimized, fine-tuned model. Table 3 shows the comparative results (training, validation, and testing) of the four experiments. In the first experiment, the validation and testing accuracy gave very low results on the original dataset. The reason is that 6229 audio files are too few for 84 classes, which are on average 74 instances per class. Therefore, when performing validation and testing, the sample size becomes very small. However, after data augmentation, model optimization, and fine-tuning, the validation and testing accuracy increased to 95.87% and 95.77%, respectively. This implies that data augmentation and fine-tuning proved useful in model improvement.

Table 3. Model execution results on different parameters.

Exp. No.	Experiment Settings	Accuracy		
		Training	Validation	Testing
1	The original dataset on baseline CNN model (n = 6229)	84.27%	40.15%	36.0%
2	The original dataset on fine-tuned model (n = 6229)	99.74%	59.33%	56.91%
3	The augmented dataset using baseline CNN model (n = 49,829)	98.93%	90.63%	90.0%
4	Model optimization, hyperparameter tuning, and augmented dataset (n = 49,829)	99.9%	95.87%	95.77%

Figure 4a,b show the accuracy and loss of the optimized and fine-tuned CNN model. We can observe that the training and validation losses improve continuously. The optimized and fine-tuned CNN model has achieved the best training accuracy of 99.857% and loss of 0.0073, and the validation accuracy is 95.87% and loss of 0.2329.



(**a**) Training and Validation Accuracy (**b**) Training and Validation Loss

Figure 4. (**a**) training and validation accuracy and (**b**) training and validation loss of the optimized CNN model on the augmented dataset.

Figure 5 shows the details of the misclassification by the classifier. Eleven of the 84 classes have 0% error rate. Only 20 classes have four or more mispredictions. The Figure shows that class 10 (phoneme (ثَ, θ_a)) and class 15 (phoneme (جُ, dʒu)) have been misidentified a maximum of eight times. From the analysis of the confusion matrix, we observe that the topmost misclassified classes are those not spoken in CA pronunciation by the native local people. For example, class 10 (phoneme (ثَ, θ_a)) is often pronounced as 7 (phoneme (تَ, ta)) in local Arabic dialect.

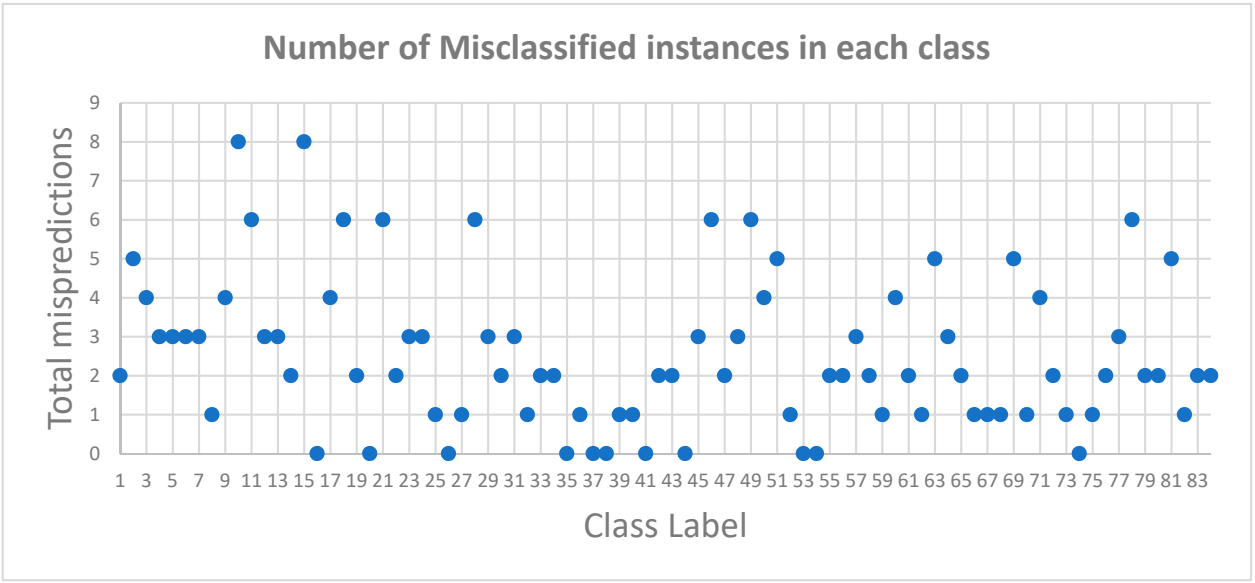


Figure 5. Scatter graph showing classes misclassified as other classes.

6. Discussion

This research addresses the recognition of short vowel phonemes in the Arabic language. The recognition task falls into the classification of classical Arabic words, a dialect of the Arabic language that is very old but is understood and applied to this day by millions of

people around the world. Our work is one step ahead of the previous efforts in classifying a total of 84 Arabic short vowels. The authors in [10] achieved an accuracy of 97.88% using CNN for Arabic alphabets (28 classes) recognition without considering the vowels. We believe that the authors could achieve highly accurate results because without considering the vowel phonemes the chances of interclass misclassification are reduced significantly. Alotaibi and Hussain [30] have considered formants and consonant-vowels-consonant (CVC) utterances for identifying similarities and differences of vowels for MSA, but to the best of our knowledge, there is no work for CA vowels recognition despite their similarities with the MSA [12].

This research started with data collection followed by its augmentation through various techniques, which is lacking in the previous studies. In [9,10,20,50], the authors have contributed to the basic Arabic alphabets audio data collection, and they mostly performed manual feature extraction. In [51], the authors collected the audio dataset of Arabic words. However, Almisreb et al. [39] investigated Arabic short vowels' properties that helped us understand their characteristics and duration variations. We were inspired by this approach and applied data augmentation to our collected audio files. Our model supports the automatic classification of Arabic short vowels on 84 classes using deep learning neural networks instead of manually identifying the audio features as done in the studies [10,23,26]. The data augmentation helped us to achieve accuracy above 95%, as in the previous research the authors have used data augmentation on 28 classes of Arabic alphabet dataset, and improved DCNN model's accuracy from 65.89% to 95.95%, Alexnet's model accuracy from 78.03% to 98.41%, and BLSTM's model accuracy from 53.18% to 87.90%[10]. In the previous studies, the authors have identified different audio features for the classification of Arabic alphabets [10,18] into 28 classes, and Arabic words [2,7,25,29] using the CNN model. The studies in [13,17,19,20,26] identified appropriate features of Arabic audio for applying machine learning techniques [17,23,50] for classification purposes.

Mispronunciation detection of Arabic is a significant parameter in a Computer Assisted Language Learning (CALL) system [51]. This is mainly a problem for non-native speakers, and approaches like [52] try to detect confusing Arabic pronunciation of similar-sounding letters for non-native speakers. However, this problem even exists in the Arabic-speaking world due to the prevalence of different regional dialects in the various parts of the world.

Our proposed approach can be utilized for developing CNN models in a similar domain for learning support systems. This approach helps construct CNN models from scratch and improves them by applying various techniques of data augmentation, fine-tuning neural networks, and hyperparameters tuning. Similar methods are used in other domains [34] for CNN models improvements. Given that the model's performance improved with synthetic data, there are chances of achieving high accuracy if more real data can be retrieved. Given our experience in this research, we believe that it will be helpful for the researchers to save their time and make these processes simple by reducing the complexity of audio data preprocessing by bypassing features identification steps.

Due to a shortage of participants for audio collection we could only get maximum 85 audio records per class and total 6229 audio files. This dataset is smaller for the requirements of the CNN model training and validation process. However, the previous studies [10,20] on CA audios the authors utilized between 2k–4k audio files to investigate DLNN models. Thus, data augmentation helped us in obtaining sufficiently large dataset. Furthermore, in this paper, we have focused on data collection in single geographic region (95%) from Saudi Arabia, which has a native Arabic speaking population. It would be of interest to evaluate this approach on data from non-native speakers as well as natives from other Arab countries.

In the near future, we intended to improve our work by obtaining data from other Arabic-speaking groups to see the generalization of our approach in a wider context. We planned to expand our current work to participants from different nationalities as well as non-native speakers and age groups to explore their pronunciation similarity and differences from native speakers, areas of improvements to facilitate development of Arabic

learning tools and applications. The analysis will be made on duration, variability, and overlapping attributes among CA learners. In addition, we also aim to quantify the standard duration of pronunciation of both short and long vowels in the classical Arabic language.

7. Conclusions

This article introduced a CNN architecture for the classification of Arabic short vowel alphabets. Using data augmentation techniques and hyperparameters tuning, we achieved a significant boost in our testing accuracy of 95.77% from a baseline model. Compared to previous approaches for Arabic alphabet classification, which classify only 28 basic alphabets, the current task was more challenging as it involved some similar sounding phonemes from 84 classes. The current work can be considered a significant leap in achieving highly accurate detection of mispronunciation of Arabic short vowels, which is considered an important step in learning classical Arabic. This contribution is beneficial for all interested stakeholders in CA to assist them in developing applications concerning Arabic pronunciation and learning recitation of the Holy Quran. Consequently, the CA learner will be benefitted for practicing Arabic short vowels using any tool based on our proposed model in unavailability of their real teacher. Furthermore, the comprehensive process of developing DLNN reported in this paper will help the developers and researchers to build learning tools by following the similar steps.

Author Contributions: Conceptualization, A.A. (Amna Asif), H.M. and H.F.A.; methodology, A.A. (Amna Asif) and H.M.; validation, A.A. (Amna Asif) and H.M.; formal analysis, A.A. (Amna Asif) and H.M.; investigation, A.A. (Amna Asif), H.M. and F.A.; resources, H.F.A.; data curation, F.A.; writing—original draft preparation, A.A. (Amna Asif), H.M., F.A. and H.F.A.; writing—review and editing, A.A. (Amna Asif), H.M.; visualization, F.A.; supervision, H.F.A.; project administration, A.A. (Abdulaziz Alhumam); funding acquisition, A.A. (Amna Asif) and A.A. (Abdulaziz Alhumam). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported through the Annual Funding track by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Project No. AN000292].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The collected dataset is available on (https://www.kaggle.com/amnaasif/arabic-short-vowels-audio-dataset, accessed on 10 November 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Julian, G. What are the most spoken languages in the world. *Retrieved May* **2020**, *31*, 2020.
2. Ali, A.; Chowdhury, S.; Afify, M.; El-Hajj, W.; Hajj, H.; Abbas, M.; Hussein, A.; Ghneim, N.; Abushariah, M.; Alqudah, A. Connecting Arabs: Bridging the gap in dialectal speech recognition. *Commun. ACM* **2021**, *64*, 124–129. [CrossRef]
3. Twaddell, W.F. On defining the phoneme. *Language* **1935**, *11*, 5–62. [CrossRef]
4. Ibrahim, A.B.; Seddiq, Y.M.; Meftah, A.H.; Alghamdi, M.; Selouani, S.-A.; Qamhan, M.A.; Alotaibi, Y.A.; Alshebeili, S.A. Optimizing arabic speech distinctive phonetic features and phoneme recognition using genetic algorithm. *IEEE Access* **2020**, *8*, 200395–200411. [CrossRef]
5. Witt, S.M. Automatic error detection in pronunciation training: Where we are and where we need to go. In Proceedings of the International Symposium on Automatic Detection on Errors in Pronunciation Training, Stockholm, Sweden, 6–8 June 2012.
6. Huang, H.; Xu, H.; Hu, Y.; Zhou, G. A transfer learning approach to goodness of pronunciation based automatic mispronunciation detection. *J. Acoust. Soc. Am.* **2017**, *142*, 3165–3177. [CrossRef] [PubMed]
7. Al-Marri, M.; Raafat, H.; Abdallah, M.; Abdou, S.; Rashwan, M. Computer Aided Qur'an Pronunciation using DNN. *J. Intell. Fuzzy Syst.* **2018**, *34*, 3257–3271. [CrossRef]
8. Ibrahim, N.J.; Idris, M.Y.I.; Yusoff, M.Z.M.; Anuar, A. The problems, issues and future challenges of automatic speech recognition for quranic verse recitation: A review. *Al-Bayan J. Qur'an Hadith Stud.* **2015**, *13*, 168–196. [CrossRef]
9. Arafa, M.N.M.; Elbarougy, R.; Ewees, A.A.; Behery, G. A Dataset for Speech Recognition to Support Arabic Phoneme Pronunciation. *Int. J. Image Graph. Signal Process.* **2018**, *10*, 31–38. [CrossRef]

10. Ziafat, N.; Ahmad, H.F.; Fatima, I.; Zia, M.; Alhumam, A.; Rajpoot, K. Correct Pronunciation Detection of the Arabic Alphabet Using Deep Learning. *Appl. Sci.* **2021**, *11*, 2508. [CrossRef]
11. Czerepinski, K. *Tajweed Rules of the Qur'an: Part 1*; Dar Al Khair: Riyadh, Saudi Arabia, 2005.
12. Alghamdi, M.M. A spectrographic analysis of Arabic vowels: A cross-dialect study. *J. King Saud Univ.* **1998**, *10*, 3–24.
13. Nazir, F.; Majeed, M.N.; Ghazanfar, M.A.; Maqsood, M. Mispronunciation detection using deep convolutional neural network features and transfer learning-based model for Arabic phonemes. *IEEE Access* **2019**, *7*, 52589–52608. [CrossRef]
14. Nassif, A.B.; Shahin, I.; Attili, I.; Azzeh, M.; Shaalan, K. Speech recognition using deep neural networks: A systematic review. *IEEE Access* **2019**, *7*, 19143–19165. [CrossRef]
15. Duan, R.; Kawahara, T.; Dantsuji, M.; Nanjo, H. Efficient learning of articulatory models based on multi-label training and label correction for pronunciation learning. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 6239–6243.
16. Necibi, K.; Bahi, H. An arabic mispronunciation detection system by means of automatic speech recognition technology. In Proceedings of the 13th International Arab Conference on Information Technoloy Proceedings, Zarqa, Jordan, 10–13 December 2012; pp. 303–308.
17. Al Hindi, A.; Alsulaiman, M.; Muhammad, G.; Al-Kahtani, S. Automatic pronunciation error detection of nonnative Arabic Speech. In Proceedings of the 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), Doha, Qatar, 10–13 November 2014; pp. 190–197.
18. Khan, A.F.A.; Mourad, O.; Mannan, A.M.K.B.; Dahan, H.B.A.M.; Abushariah, M.A. Automatic Arabic pronunciation scoring for computer aided language learning. In Proceedings of the 2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA), Sharjah, United Arab Emirates, 12–14 February 2013; pp. 1–6.
19. Marlina, L.; Wardoyo, C.; Sanjaya, W.M.; Anggraeni, D.; Dewi, S.F.; Roziqin, A.; Maryanti, S. Makhraj recognition of Hijaiyah letter for children based on Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machines (SVM) method. In Proceedings of the 2018 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 6–7 March 2018; pp. 935–940.
20. Akhtar, S.; Hussain, F.; Raja, F.R.; Ehatisham-ul-haq, M.; Baloch, N.K.; Ishmanov, F.; Zikria, Y.B. Improving mispronunciation detection of arabic words for non-native learners using deep convolutional neural network features. *Electronics* **2020**, *9*, 963. [CrossRef]
21. Leung, W.-K.; Liu, X.; Meng, H. CNN-RNN-CTC based end-to-end mispronunciation detection and diagnosis. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 8132–8136.
22. Zainon, N.Z.; Ahmad, Z.; Romli, M.; Yaacob, S. Speech quality based on Arabic pronunciation using MFCC and LDA: Investigating the emphatic consonants. In Proceedings of the 2012 IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia, 23–25 November 2012; pp. 398–403.
23. Aissiou, M. A genetic model for acoustic and phonetic decoding of standard Arabic vowels in continuous speech. *Int. J. Speech Technol.* **2020**, *23*, 425–434. [CrossRef]
24. Abdou, S.M.; Rashwan, M. A Computer Aided Pronunciation Learning system for teaching the holy quran Recitation rules. In Proceedings of the 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), Doha, Qatar, 10–13 November 2014; pp. 543–550.
25. Necibi, K.; Frihia, H.; Bahi, H. On the use of decision trees for arabic pronunciation assessment. In Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication, Batna, Algeria, 23–25 November 2015; pp. 1–6.
26. Abdelhamid, A.A.; Alsayadi, H.A.; Hegazy, I.; Fayed, Z.T. End-to-End Arabic Speech Recognition: A Review. In Proceedings of the 19th Conference of Language Engineering (ESOLEC'19), Alexandria, Egypt, 26–30 September 2020.
27. Fadel, A.; Tuffaha, I.; Al-Ayyoub, M. Arabic text diacritization using deep neural networks. In Proceedings of the 2019 2nd International Conference on computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 1–3 May 2019; pp. 1–7.
28. Al-Anzi, F.S.; AbuZeina, D. Synopsis on Arabic speech recognition. *Ain Shams Eng. J.* **2021**, *13*, 9. [CrossRef]
29. Lamel, L.; Messaoudi, A.; Gauvain, J.-L. Automatic speech-to-text transcription in Arabic. *TALIP* **2009**, *8*, 1–18. [CrossRef]
30. Alotaibi, Y.A.; Hussain, A. Comparative analysis of Arabic vowels using formants and an automatic speech recognition system. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2010**, *3*, 11–22.
31. Yu, D.; Li, J. Recent progresses in deep learning based acoustic models. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 396–409. [CrossRef]
32. Alqadheeb, F.; Asif, A.; Ahmad, H.F. Correct Pronunciation Detection for Classical Arabic Phonemes Using Deep Learning. In Proceedings of the 2021 International Conference of Women in Data Science at Taif University (WiDSTaif), Taif, Saudi Arabia, 30–31 March 2021; pp. 1–6.
33. Wyse, L. Audio Spectrogram Representations for Processing with Convolutional Neural Networks. In Proceedings of the First International Conference on Deep Learning and Music, Anchorage, AK, USA, 17–18 May 2017; pp. 37–41.
34. Mukhtar, H.; Qaisar, S.M.; Zaguia, A. Deep Convolutional Neural Network Regularization for Alcoholism Detection Using EEG Signals. *Sensors* **2021**, *21*, 5456. [CrossRef]

35. Tajbakhsh, N.; Shin, J.Y.; Gurudu, S.R.; Hurst, R.T.; Kendall, C.B.; Gotway, M.B.; Liang, J. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Trans. Med. Imaging* **2016**, *35*, 1299–1312. [CrossRef]
36. Shorten, C.; M. Khoshgoftaar, T. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
37. Wei, S.; Zou, S.; Liao, F. A Comparison on Data Augmentation Methods Based on Deep Learning for Audio Classification. *J. Phys. Conf. Ser.* **2020**, *1453*, 012085. [CrossRef]
38. Nanni, L.; Maguolo, G.; Paci, M. Data augmentation approaches for improving animal audio classification. *Ecol. Inform.* **2020**, *57*, 101084. [CrossRef]
39. Abd Almisreb, A.; Abidin, A.F.; Tahir, N.M. An acoustic investigation of Arabic vowels pronounced by Malay speakers. *J. King Saud Univ. -Comput. Inf. Sci.* **2016**, *28*, 148–156. [CrossRef]
40. Traore, B.B.; Kamsu-Foguem, B.; Tangara, F. Deep convolution neural network for image recognition. *Ecol. Inform.* **2018**, *48*, 257–268. [CrossRef]
41. Sun, M.; Song, Z.; Jiang, X.; Pan, J.; Pang, Y. Learning pooling for convolutional neural network. *Neurocomputing* **2017**, *224*, 96–104. [CrossRef]
42. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 448–456.
43. Baldi, P.; Sadowski, P.J. Understanding dropout. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2814–2822.
44. Sharma, S.; Sharma, S. Activation functions in neural networks. *Towards Data Sci.* **2017**, *6*, 310–316. [CrossRef]
45. Young, H.P. Learning by trial and error. *Games Econ. Behav.* **2009**, *65*, 626–643. [CrossRef]
46. Zhang, Z. Improved adam optimizer for deep neural networks. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–2.
47. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
48. Brownlee, J. How to Configure the Learning Rate When Training Deep Learning Neural Networks. Available online: https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/ (accessed on 10 November 2021).
49. Google. TensorBoard: TensorFlow's Visualization Toolkit. Available online: https://www.tensorflow.org/tensorboard (accessed on 19 August 2021).
50. Lee, A.; Zhang, Y.; Glass, J. Mispronunciation detection via dynamic time warping on deep belief network-based posteriorgrams. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 8227–8231.
51. Maqsood, M.; Habib, H.A.; Nawaz, T. An efficientmis pronunciation detection system using discriminative acoustic phonetic features for arabic consonants. *Int. Arab J. Inf. Technol.* **2019**, *16*, 242–250.
52. Maqsood, M.; Habib, H.; Anwar, S.; Ghazanfar, M.; Nawaz, T. A comparative study of classifier based mispronunciation detection system for confusing Arabic phoneme pairs. *Nucleus* **2017**, *54*, 114–120.