applied sciences

MDPI

*Article*

# Spoken Language Identification System Using Convolutional Recurrent Neural Network

**Adal A. Alashban *, Mustafa A. Qamhan, Ali H. Meftah and Yousef A. Alotaibi**

Computer Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

* Correspondence: 439204375@student.ksu.edu.sa

**Abstract:** Following recent advancements in deep learning and artificial intelligence, spoken language identification applications are playing an increasingly significant role in our day-to-day lives, especially in the domain of multi-lingual speech recognition. In this article, we propose a spoken language identification system that depends on the sequence of feature vectors. The proposed system uses a hybrid Convolutional Recurrent Neural Network (CRNN), which combines a Convolutional Neural Network (CNN) with a Recurrent Neural Network (RNN) network, for spoken language identification on seven languages, including Arabic, chosen from subsets of the Mozilla Common Voice (MCV) corpus. The proposed system exploits the advantages of both CNN and RNN architectures to construct the CRNN architecture. At the feature extraction stage, it compares the Gammatone Cepstral Coefficient (GTCC) feature and Mel Frequency Cepstral Coefficient (MFCC) feature, as well as a combination of both. Finally, the speech signals were represented as frames and used as the input for the CRNN architecture. After conducting experiments, the results of the proposed system indicate higher performance with combined GTCC and MFCC features compared to GTCC or MFCC features used individually. The average accuracy of the proposed system was 92.81% in the best experiment for spoken language identification. Furthermore, the system can learn language-specific patterns in various filter size representations of speech files.

**Keywords:** Arabic; CNN; CRNN; classification; language identification; LSTM; Mozilla speech corpus; multi-lingual speech; RNN



**Citation:** Alashban, A.A.; Qamhan, M.A.; Meftah, A.H.; Alotaibi, Y.A. Spoken Language Identification System Using Convolutional Recurrent Neural Network. *Appl. Sci.* **2022**, *12*, 9181. https://doi.org/10.3390/app12189181

Academic Editors: Valentino Santucci and Paolo Mengoni

Received: 8 August 2022
Accepted: 9 September 2022
Published: 13 September 2022



## 1. Introduction

Several issues have impeded the development of Automatic Speech Recognition (ASR) systems. The most important of these is spoken Language Identification (LID) in multi-lingual ASR systems [1]. The purpose of spoken LID systems is to classify spoken language from a given audio sample. As such, in the absence of automatic language detection, speech utterances cannot be correctly processed, and grammatical rules cannot be applied [2]. Language has no geographical boundaries in the contemporary era. More than five thousand languages are spoken worldwide, and each has distinct properties at different levels, from acoustics to semantics [3–5]. Western countries have made significant progress in using applications based on spoken language recognition. However, these applications have not gained much traction in multi-lingual and multi-dialectal Arabic countries, including Saudi Arabia, due to the complexity of the native Arabic language [6,7]. Spoken LID systems seek to determine and classify the language that is spoken within a speech utterance [8]. The problem of identifying the spoken language of a given audio file is of considerable interest to a range of multi-lingual speech processing applications, including speech translation [9], multi-lingual speech recognition [10–12], spoken document retrieval [13], and defense and surveillance applications [14]. This article's main contribution lies in considering seven spoken language identification, including Arabic, using public audio speech corpus, and the joint use of the MFCC and GTCC under the CRNN framework. The contribution can be

summarized as follows: (1) We present a hybrid CRNN that combines a CNN's descriptive capabilities with an LSTM architecture's capacity to capture sequence data; (2) We compare the effect of fine-tuning the features; (3) We conduct comprehensive tests and experiments with the proposed architecture, demonstrating its applicability in a variety of contexts, as well as its extension to new languages; and (4) We use system errors to infer the degree of similarities in considered languages.

The article is structured as follows: in Section 2, we provide a literature review in the field of spoken LID systems; in Section 3, the chosen speech corpora are presented; Section 4 showcases the proposed system; in Section 5, we evaluate the proposed system using extensive experiments; in Section 6, we present and discuss the results; and finally, in Section 7, we present concluding remarks and recommendations for future research.

## 2. Literature Review

A comprehensive survey of the literature on state-of-the-art spoken LID, focusing especially on speech features and architectures, was conducted by Shen et al. [15]. Most of the architectures used spectral and acoustic features for spoken LID [16–18]. However, there are few reported attempts for spoken Arabic language identification among the set of languages. In 1999, Nofal et al. [19] proposed a spoken LID system for identifying just two languages: Arabic and English. Zazo et al. [20] presented an LSTM-Recurrent Neural Network (LSTM-RNN) architecture for the spoken LID system. The system outperformed a reference I-vector system by 26.00% on a subset of the National Institute of Standards and Technology (NIST) Language Recognition Evaluation (LRE) 2009 corpus for eight languages. Furthermore, the results demonstrated that an LSTM-RNN architecture could identify languages with an accuracy of 70.90%. Kumar [16] proposed Fourier Parameter (FP) features for the spoken LID system. The performance of the FP features was analyzed and compared with the legacy Mel Frequency Cepstral Coefficient (MFCC) features. They used the Indian Institute of Technology Kharagpur Multi-lingual Indian Language Speech Corpus (IITKGP-MLILSC) and the Oriental Language Recognition Speech Corpus (AP18-OR) to identify the spoken language. Finally, they developed three architectures with the extracted FP and MFCC features: Support Vector Machine (SVM), feed-forward Artificial Neural Network (ANN), and Deep Neural Network (DNN). The results demonstrated that the proposed FP features effectively recognize different spoken languages from speech signals. A performance improvement was also observed when combining FP and MFCC.

Draghici et al. [21] investigated the previous architectures for spoken LID systems, which utilize the CNN and CRNN architectures. Additionally, they used a set of seven languages. Despite the increasing complexity of the architectures, it was successful, achieving an accuracy of 71.00% for the CNN architecture and 83.00% for the CRNN architecture. Kim and Park [11] proposed a spoken LID system and investigated another feature that characterizes language-specific properties: speech rhythm. They evaluated two corpora published by different organizations. The first corpus is the Speech Information Technology and Industry Promotion Center (SiTEC). The second corpus is the Mozilla Common Voice (MCV) corpus. Despite the low computational complexity of the system, the results were either poorer than or similar to the performance of traditional approaches, with an error rate of up to 65.66%. Guha et al. [22] developed a hybrid Feature Selection (FS) algorithm using the versatile Harmony Search (HS) algorithm and the Naked Mole-Rat (NMR) algorithm for Human-Computer Interaction (HCI)-based applications by attempting to classify various languages from the three corpora: the Collection of Single Speaker Speech corpus (CSS10) for ten languages, the VoxForge corpus for six languages, and the Madras (IIT-M) speech corpus for ten languages. This study involved extracting their Mel spectrogram features and Relative Spectral Transform-Perceptual Linear Prediction (RASTA-PLP) features, and these features were used as input for five classifiers: SVM, k-Nearest Neighbor (k-NN), Multi-layer Perceptron (MLP), Naïve Bayes (NB), and RF. Accuracies of 99.89% on CSS10, 98.22% on VoxForge, and 99.75% on the IIT-Madras speech corpus databases were achieved using RF. Sangwan et al. [23] presented a system that used hybrid features and ANN

learning architectures for spoken LID of four languages. First, they used RASTA-PLP and MFCC features with RASTA-PLP features. The classifier selected for the spoken LID system was a Feed-Forward Back Propagation Neural Network (FFBPNN). The results indicate better performance with combined MFCC and RASTA-PLP features compared to the individual use of RASTA-PLP features. The proposed classification yielded an accuracy of 95.30%. Garain et al. [24] suggested a deep learning-based system called FuzzyGCP for spoken LID. This system integrates the classification principles of Deep Convolutional Neural Networks (DCNN), a Deep Dumb Multi-Layer Perceptron (DDMLP), and Semi-Supervised Generative Adversarial Networks (SSGAN). They tested the system in four corpora containing two Indic and two foreign language corpora. As a result, the system achieved an accuracy of 98.00% on the Multi-lingual Corpus of Sentence-Aligned Spoken Utterances (MaSS) corpus. Moreover, it achieved the worst performance of 67.00% on the VoxForge corpus. Shen et al. [25] proposed an RNN Transducer (RNN-T) architecture for the spoken LID system. The system exploits phonetically-aware acoustic features and explicit linguistic features for LID tasks. Experiments were conducted on eight languages from the Multi-lingual LibriSpeech (MLS) corpus. The results showed that the proposed system significantly improved the performance on spoken LID tasks, with relative improvements of 12.00% to 59.00% and 16.00% to 24.00% on in-domain and cross-domain corpora, respectively.

Spoken LID systems have been widely investigated using multiple architectures in terms of feature extraction and classifier learning. However, they are limited to most human languages [26,27]. Additionally, while many researchers have successfully used deep learning techniques to solve spoken LID problems, as presented in [22–24], the central problem remains that the Arabic language has only received limited attention in the spoken LID research community among the set of languages [19,28,29]. Therefore, this article seeks to improve the performance of spoken LID tasks for six spoken languages from industrial countries in conjunction with Arabic [30]. These are seven languages of the ten most in-demand foreign languages across the globe [31]. A summary of the literature on spoken LID systems is presented in Table 1.

**Table 1.** Review of previous literature on spoken language identification with results.

| Reference | Year | Features | Classifiers | Corpus | Classes | Languages | Results (%) |
|---|---|---|---|---|---|---|---|
| Zazo et al. [20] | 2016 | MFCC-SDC | LSTM-RNN | NIST LRE | 8 | Dari, English, French, Chinese, Pashto, Russian, Spanish, and Urdu | Accuracy = 70.90 |
| Bartz et al. [2] | 2017 | Spectrogram | CRNN | European Parliament Statements and News Channels on YouTube | 6 | English, French, German, Chinese, Russian, and Spanish | Accuracy = 91.00 |
| Ma and Yu. [27] | 2018 | DNN-BN | LSTM | AP17-OLR | 10 | Tibetan, Japanese, Kazakh, Korean, Indonesian, Mandarin, Cantonese, Vietnamese, Uyghur, and Russian | ER = 50.00 |
| Kumar. [16] | 2019 | FP-MFCC | ANN | IITKGP-MLILSC and AP18-OLR | 10 | Russian, Vietnamese, Indonesian, Cantonese, Japanese, Kazakh, Korean, Tibetan, Uyghur, and Mandarin | Accuracy = 70.80 |
| Kim and Park [11] | 2020 | Rhythm | R-Vector with I-Vector | SiTEC<br>Mozilla | 2<br>3 | English and Korean<br>Chinese, English, and Spanish | ER = 4.73<br>ER = 47.38 |

**Table 1.** *Cont.*

| Reference | Year | Features | Classifiers | Corpus | Classes | Languages | Results (%) |
|---|---|---|---|---|---|---|---|
| Guha et al. [22] | 2020 | Mel spectrogram-RASTA-PLP | RF | CSS10 | 10 | French, Chinese, German, Dutch, Spanish, Greek, Finnish, Japanese, Russian, and Hungarian | Accuracy = 99.89 |
| | | | | VoxForge | 6 | English, French, German, Italian, Russian, and Spanish | Accuracy = 98.22 |
| | | | | IIT-Madras | 10 | Assamese, English, Bangla, Gujarati, Tamil, Hindi, Telugu, Kannada, Marathi, and Malayalam | Accuracy = 99.75 |
| Draghici et al. [21] | 2020 | Mel spectrogram | CRNN | EU Repo | 6 | English, French, German, Greek, Italian, and Spanish | Accuracy = 83.00 |
| Sisodia et al. [32] | 2020 | MFCC-DFCC | Extra Trees | VoxForge | 5 | Dutch, English, French, German, and Portuguese | Accuracy = 85.71 |
| Garain et al. [24] | 2021 | MFCC-Spectral Bandwidth-Spectral Contrast- Spectral Roll-Off- Spectral Flatness-Spectral Centroid-Polynomial-Tonnetz | FuzzyGCP | MaSS | 8 | Basque, English, Finnish, French, Hungarian, Romanian, Russian, and Spanish | Accuracy = 98.00 |
| | | | | VoxForge | 5 | French, German, Italian, Portuguese, and Spanish | Accuracy = 67.00 |
| Sangwan et al. [23] | 2021 | MFCC-RASTA-PLP | FFBPNN | New Corpus | 4 | English, Hindi, Malayalam, and Tamil | Accuracy = 95.30 |
| Singh et al. [33] | 2021 | Log-Mel spectrograms | CNN | Mozilla | 4 | Estonian, Tamil, Turkish, and Mandarin | Accuracy = 80.21 |
| Shen et al. [25] | 2022 | Acoustic-Linguistic | RNN-T | MLS | 8 | Dutch, English, French, German, Polish, Portuguese, Italian, and Spanish | ER = 5.44 |

## 3. Selected Speech Corpus

The corpus selected for this article is from the Mozilla Common Voice (MCV) corpora. MCV is a multi-lingual corpus of speech intended for speech technology research and development. It is designed for ASR purposes and is also well known for being useful in other domains; e.g., LID and gender classification [34]. Each input in the corpora consists of an individual MP3 file and a corresponding text file [35]. The most recent release includes 87 languages, but the publishers periodically add more voices and languages. Over 200,000 male and female speakers have participated, resulting in 18,243 h of collected audio.

To achieve scale and sustainability, the MCV project uses crowdsourcing for data collection and validation [36]. As an example use case for MCV, we present our spoken LID experiments using subsets of MCV for seven target languages: Arabic, German, English, Spanish, French, Russian, and Chinese. These are the first experiments undertaken on most of these languages for spoken LID. The aim was to compare the spoken LID performance and outcomes of Arabic with other languages. Table 2 presents the characteristics of the MCV corpus. Table 3 summarizes the experimental corpus.

**Table 2.** Mozilla corpus characteristics.

| Parameter | Value |
|---|---|
| Sampling Rate | 48 kHz |
| Date | 19 January 2022 |
| Validated Hours | 14,122 |
| Recorded Hours | 18,243 |
| Languages | 87 |

**Table 3.** Selected Mozilla corpus description.

| Language | Number of Speech Files | Training/Validation 80%, Testing 20% | | |
|---|---|---|---|---|
| | | Training (90%) from 80% | Validation (10%) from 80% | Testing (20%) |
| Arabic | 2000 | 1440 | Arabic | 2000 |
| German | 2000 | 1440 | German | 2000 |
| English | 2000 | 1440 | English | 2000 |
| Spanish | 2000 | 1440 | Spanish | 2000 |
| French | 2000 | 1440 | French | 2000 |
| Russian | 2000 | 1440 | Russian | 2000 |
| Chinese | 2000 | 1440 | Chinese | 2000 |
| Total | 14,000 | 10,080 | Total | 14,000 |

## 4. Proposed Spoken Language Identification System

This section outlines the motivation for the article and describes the proposed spoken LID system presented in Figure 1.

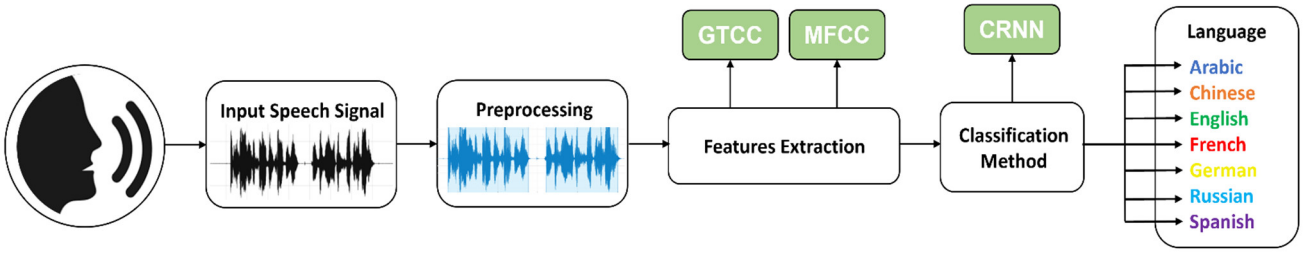


**Figure 1.** Spoken LID system.

### *4.1. Motivation*

The motivation for this article is to present a novel spoken LID system for seven languages, including Arabic, that investigates and analyzes spoken LID problems with the most cutting-edge systems available.

### *4.2. Preprocessing*

Data preprocessing is critical for the success of deep learning systems [37,38]. In this article, data processing was done by detecting the boundaries of speech. In particular, speech signals were segmented and silent parts removed depending on speech boundaries using R2020b-MATLAB's detect speech function [39]. This was done to prevent the silent parts from influencing the classification task. Figure 2 shows plots of the detected speech boundaries.

### *4.3. Selected Features*

To reduce signal redundancy and improve accuracy in the spoken LID system, we propose combining MFCC and GFCC features [40].

#### 4.3.1. MFCC

Mel Frequency Cepstral Coefficient (MFCC) is the most popular feature extraction method in speech processing [41] and the best feature extraction technique [42]. The process for elaborating a character vector of MFCC is given as follows.

First, a pre-emphasis filter is applied to the signal. It is then divided into frames, with the windowing function subsequently applied to the frames. When obtaining the Discrete Fourier Transform (DFT) of each frame, the amplitude of the spectrum is used, and this information is passed to a Mel domain through the Filter Bank (Mel scale); this scale is a simulated human listener. After the logarithm of the signal is obtained, the Discrete

Cosine Transform (DCT) is applied from the obtained vector. In turn, the number of desired coefficients per frame is taken [43].

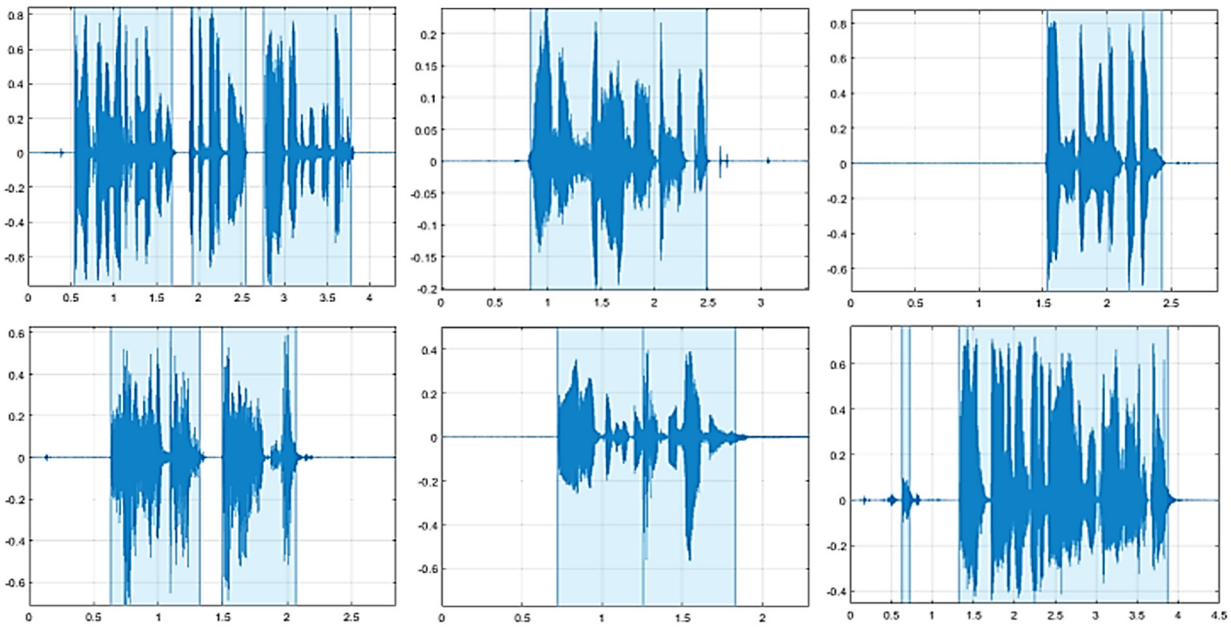

**Figure 2.** Data preprocessing step.

#### 4.3.2. GTCC

Gammatone Cepstral Coefficient (GTCC) is often referred to as Frequency Cepstrum Coefficient (GFCC) or Gammatone Cepstrum Coefficient (GTCC). The Gammatone filter is inspired by biological life, in which the representation of the human auditory filter response in the cochlea is very similar to the magnitude response of a Gammatone filter [44]. The method of extracting GTCC is similar to MFCC, with two main differences. The first is the frequency scale used, where GTCC provides greater resolution at low frequencies than MFCC. This is because it is based on the equivalent rectangular bandwidth scale, while MFCC is based on Mel-scale. The GTCC uses the cubic root, whereas the MFCC employs the log.

Due to its high accuracy and low complexity, MFCC is frequently used for sound processing systems. However, it does not have high resistance to noise. Recent investigations have revealed that the properties of GTCC, by contrast, are that it is particularly resistant to noise and acoustic change [45]. With these considerations in mind, the primary goal is to combine the characteristics of GTCC and MFCC to improve the overall testing accuracy of the system.

### *4.4. Proposed CRNN Architecture*

CNN has been used widely in various application areas, including image classification and speech recognition. In these domains, A CNN has achieved state-of-the-art levels of performance. The convolutional layer in a CNN facilitates feature extraction, which is performed using convolutional processing and filtering. A sliding window (filter) is applied to the inputs to perform this filtering operation. By calculating the convolution product between the window and the considered input portion, the applied filter will represent the characteristic extracted by the network, with its parameters estimated during network training. The objective of subsampling layers is to reduce the dimensionality of the characteristics that result from the convolution without significant loss of information. Standard methods are max-pooling and average-pooling.

In the RNN family, the LSTM network is a unique subclass [46]. This network has been shown to yield remarkable results for learning long-time dependencies. Recursion

is the critical feature of recurrent networks. In particular, this network creates loops in the network diagram that can preserve information. As a result, they can essentially "remember" previous states and use that knowledge to determine what will happen next. This property makes them ideal for working with time series data. LSTM units are used for the construction of recurrent networks. Each LSTM unit is a cell with four gates: the input gate, the external input gate, the forget gate, and the output gate. The most important aspect of the cell is its internal state, which enables it to store and preserve values over time.

This article proposes a CRNN architecture that combines CNN with LSTM, as demonstrated above. CNN is an excellent network for feature extractions, while the LSTM has proved its ability to identify the language in sequence-to-sequence series. Figure 3 shows the architecture of the proposed CRNN.

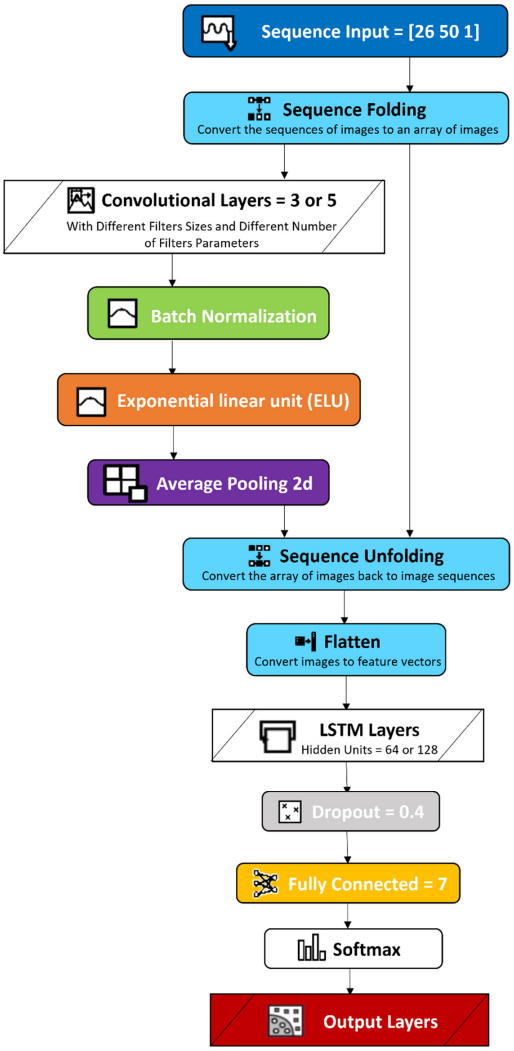


**Figure 3.** CRNN architecture.

The proposed CRNN architecture, as shown in Figure 3, contains CNN layers for feature extraction on the input data, as well as LSTM to support long-term temporal dynamics. The novelty of the proposed CRNN architecture is in using three layers to convert the type of input data depending on the networks used in CRNN: a sequence folding layer, a sequence unfolding layer, and a flatten layer.

The sequence folding layer converts the sequences of images to an array of images [47] for CNN to be able to extract spatial features from the input array. The sequence unfolding layer converts this array of images back to image sequences [48], and the flatten layer converts image sequences to feature vectors [49] for input to the LSTM layer. Finally, to

carry out the final prediction, the results are provided as input, fully connected layers, and output layers in the prediction block [50].

### 4.5. Layers Description

(1) The Sequence Input Layer for 2-D image sequence input contains the vector of three elements. Depending on the used parameters, this article includes 26 dimensions: 13 for GTCC and 13 for MFCC; there are 50 feature vectors per sequence (Because the features that were extracted from segments have different feature vectors depending on speech file length, we need the framing to buffer the feature vectors into fixed sequences of size 50 frames with 47 overlaps); hence the input size is [26 50 1], where 26, 50, and 1 correspond to the height, width, and the number of channels of the image, respectively.
(2) To use Convolutional Layers to extract features (that is, to apply convolutional operations to each speech frame independently), we use a Sequence Folding Layer followed by convolutional layers.
(3) The Batch Normalization Layer follows the Convolution Layer, where Batch Normalization is responsible for the convergence of learned representations. Then the ELU Layer.
(4) Average Pooling 2 d = $1 \times 1$ with stride [10 10] and padding [0 0 0 0]. A 2-D average pooling layer performs downsampling by dividing the input into rectangular pooling regions and computing the average values of each region.
(5) We use a Sequence Unfolding Layer and a Flatten Layer to restore the sequence structure and reshape the output to vector sequences.
(6) We include the LSTM Layers to classify the resulting vector sequences.
(7) The Dropout Layer, with a dropout possibility of 40%, always follows every LSTM layer.
(8) The Fully Connected Layer contains seven neurons.
(9) The Softmax Layer applies a softmax function to the input.
(10) The Classification Output Layer acts as an output layer for the proposed system.

## 5. Experiments

This section provides detail regarding performance metrics and the different proposed experiments.

### 5.1. Performance Evaluation

Two options are presented in this article for evaluating the system's performance. The first is the evaluation adopted on the sequence level, and the second is the evaluation based on the file level. Each speech file contained different numbers of sequences. Therefore, to compute the complete speech file accuracy, as in [51], the accuracy of each sequence was calculated. Choosing the right metrics is crucial for accurately measuring the performance of trained architectures using a testing corpus. It is always desirable to ensure the exactness of the system by computing various metrics. This article used four performance metrics: accuracy (A), precision (P), recall (R), and F-measure (F1) [26] to determine the effectiveness of the CRNN architecture for the spoken LID problem. For all metrics, TP represents all true positives, TN denotes all true negatives, FP shows all false positives, and FN represents all false negatives [32].

#### 5.1.1. Accuracy (A)

As shown in Equation (1), accuracy is the ratio of truly predicted levels to total levels.

$$A = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (1)$$

#### 5.1.2. Precision (P)

As shown in Equation (2), precision is the fraction of truly positive predicted levels to overall positive predicted levels. A low precision value indicates a high false-positive rate and vice versa.

$$P = \frac{TP}{TP + FP} \times 100 \tag{2}$$

#### 5.1.3. Recall (R)

As shown in Equation (3), recall (also known as sensitivity) is the fraction of truly positive projected levels to overall positive actual levels. A low recall value suggests a high false-negative rate and vice versa.

$$R = \frac{TP}{TP + FN} \times 100 \tag{3}$$

#### 5.1.4. F-Measure (F1)

As shown in Equation (4), a weighted average of recall and precision is known as the F-measure.

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{4}$$

The variables $P$ and $R$ represent precision and recall in this equation.

### *5.2. Experimental Setup*

The proposed system contains adjustable parameters, including filter size, the number of filters, the number of layers, and the number of hidden units (for LSTM), which may be tuned optimally to reduce the prediction error. In contrast, these differences in parameters achieved complexity in the proposed CRNN architecture. We performed 12 experiments with different parameters to study what combination of variables would achieve the highest accuracy for the system. Table 4 provides a snapshot of the configurations and results of a selection of the experiments undertaken. GTCC and MFCC were generated with MATLAB for each audio file in the database using Hamming windows with a window length of 30 msec and a window step of 20 msec. This resulted in [26 50 1] matrices, which were fed into the architecture as inputs. An NVIDIA GeForce RTX 2080 Ti graphics processing unit with 11 GB RAM was employed for training, with a batch size of 128 samples for 15 epochs and a learning rate of 0.0009, using Adam's adaptive gradient descent method as the optimizer.

**Table 4.** Snapshot of selected experiment configurations and results.

| # Experiment | # Convolution Layers | Parameters | # Hidden Units in LSTM Layer | Training Accuracy (%) | Validation Accuracy (%) | Testing Accuracy (%) | Time Taken |
|---|---|---|---|---|---|---|---|
| 1 | 5 | Filter Size = 5<br># Filters = 12 | 128 | 99.74 | 94.46 | 92.21 | 162 min 40 s |
| 2 | 5 | Filter Size = 10<br># Filters = 32 | 128 | 97.99 | 93.12 | 92.00 | 251 min 37 s |
| 3 | 3 | Filter Size1 = 10<br># Filters1 = 32<br>Filter Size2 = 5<br># Filters2 = 12<br>Filter Size3 = 15<br># Filters3 = 64 | 128 | 99.42 | 93.83 | 92.14 | 170 min 4 s |
| 4 | 5 | Filter Size1 = 4<br># Filters1 = 32<br>Filter Size2 = 6<br># Filters2 = 12<br>Filter Size3 = 8 | 128 | 98.94 | 94.29 | 92.57 | 236 min 41 s |

**Table 4.** *Cont.*

| # Experiment | # Convolution Layers | Parameters | # Hidden Units in LSTM Layer | Training Accuracy (%) | Validation Accuracy (%) | Testing Accuracy (%) | Time Taken |
|---|---|---|---|---|---|---|---|
| 4 | 5 | # Filters3 = 64<br>Filter Size4 = 10<br># Filters4 = 32<br>Filter Size5 = 12<br># Filters5 = 12 | 128 | 98.94 | 94.29 | 92.57 | 236 min 41 s |
| 5 | 5 | Filter Size = 10<br># Filters = 32 | 64 | 98.69 | 93.57 | 92.71 | 253 min 12 s |
| 6 | 5 | Filter Size1 = 5<br># Filters1 = 4<br>Filter Size2 = 6<br># Filters2 = 8<br>Filter Size3 = 7<br># Filters3 = 16<br>Filter Size4 = 8<br># Filters4 = 32<br>Filter Size5 = 7<br># Filters5 = 64 | 128 | 98.90 | 92.05 | 91.42 | 196 min 39 s |
| 7 | 3 | Filter Size1 = 12<br># Filters1 = 16<br>Filter Size2 = 8<br># Filters2 = 24<br>Filter Size3 = 5<br># Filters3 = 32 | 128 | 98.44 | 90.98 | 90.18 | 142 min 14 s |
| 8 | 5 | Filter Size = 5<br># Filters = 32 | 64 | 97.54 | 92.5 | 90.89 | 195 min 33 s |
| 9 | 3 | Filter Size1 = 10<br># Filters1 = 32<br>Filter Size2 = 10<br># Filters2 = 64<br>Filter Size3 = 10<br># Filters3 = 128 | 128 | 97.92 | 92.86 | 92.14 | 355 min 34 s |
| 10 | 5 | Filter Size = 10<br># Filters = 32 | 32 | 98.57 | 93.03 | 92.32 | 862 min 51 s |
| 11 | 5 | Filter Size1 = 10<br># Filters1 = 32<br>Filter Size2 = 5<br># Filters2 = 12<br>Filter Size3 = 15<br># Filters3 = 64<br>Filter Size4 = 5<br># Filters4 = 12<br>Filter Size5 = 10<br># Filters5 = 32 | 128 | 98.96 | 94.01 | 93.00 | 221 min 29 s |
| 12 | 5 | Filter Size = 5<br># Filters = 32 | 128 | 97.78 | 91.42 | 92.57 | 195 min 2 s |

In contrast, the # symbol means number, and min and s abbreviations mean minute and second.

## 6. Results and Discussion

### *6.1. Feature Comparison Results*

In this article, a comparison of the accuracy of GTCC and MFCC was undertaken for spoken LID in the proposed system. The accuracy of GTCC compared with MFCC was evaluated in Experiment 1 of Table 4 for all files using a considered MCV corpus with the execution of five runs. Table 5 shows the experimental results for GTCC and MFCC, as well as combined features.

**Table 5.** Comparison of GTCC and MFCC for spoken LID in the proposed system.

| Features: | GTCC | | MFCC | | GTCC-MFCC | |
|---|---|---|---|---|---|---|
| Per Files (%) Experiment 1 | Testing Accuracy (%) | Time Taken | Testing Accuracy (%) | Time Taken | Testing Accuracy (%) | Time Taken |
| Run # 1 | 89.07 | 142 min 49 s | 89.21 | 137 min 48 s | 92.21 | 162 min 40 s |
| Run # 2 | 90.00 | 145 min 31 s | 89.93 | 139 min 2 s | 91.79 | 161 min 36 s |
| Run # 3 | 88.68 | 145 min 43 s | 90.11 | 139 min 52 s | 91.25 | 163 min 11 s |
| Run # 4 | 89.21 | 145 min 54 s | 90.82 | 140 min 52 s | 91.86 | 164 min 57 s |
| Run # 5 | 88.64 | 147 min 3 s | 90.57 | 141 min 15 s | 91.68 | 153 min 46 s |
| Average | 89.12 | 145 min 36 s | 90.13 | 138 min 34 s | 91.76 | 160 min 38 s |

The highest average testing accuracy was achieved when concatenating GTCC and MFCC as system features. The average precision of GTCC for spoken LID in Arabic was 94.88%, Chinese 94.20%, English 91.30%, French 90.58%, German 74.84%, Russian 92.98%, and Spanish 89.86%. In contrast, the average precision of MFCC for spoken LID in Arabic was 94.82%, Chinese 94.88%, English 89.76%, French 92.14%, German 76.78%, Russian 95.32%, and Spanish 91.72%. The results indicate that GTCC outperformed MFCC in Arabic and English spoken LID. However, the most effective results were obtained when GTCC was combined with MFCC. These results are illustrated in Figure 4. However, Table 5 shows that consumption time is a major issue in spoken LID systems when using more than one feature, especially in the real-time setting.

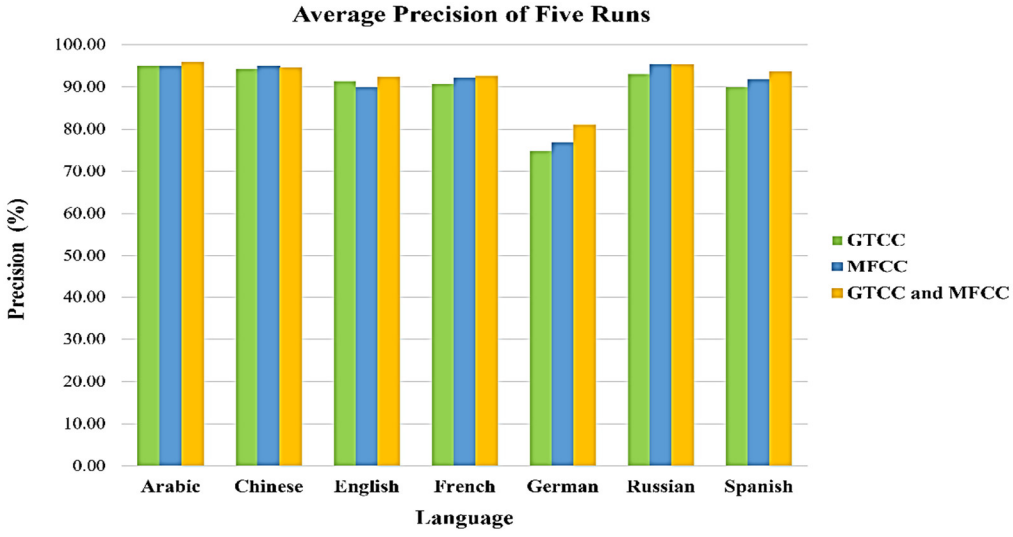


**Figure 4.** The average precision of each feature.

The average precision when combining GTCC with MFCC in Experiment 1 for spoken LID in Arabic is 95.78%, Chinese 94.47%, English 92.35%, French 92.49%, German 80.95%, Russian 95.25%, and Spanish 93.49%. Depending on the testing accuracies and experiment consumption time, the systems in Experiments 1 and 11 achieved the highest accuracy and the lowest execution time, as shown in Table 4. The proposed architecture's accuracy in

Experiment 1 reached 92.21% in around 163 min, while in Experiment 11 it reached 93.00% in 221 min. Each experiment was performed for five runs per sequence and per whole file.

### 6.2. Spoken Language Identification Results

Table 6 summarizes the results for Experiments 1 and 11, as listed in Table 4. Table 6 shows that the overall testing accuracy per file was better than per sequence, suggesting a level of similarity between the languages in the file sequences. The average per-file accuracy of five runs was 91.76% in Experiment 1 and 92.81% in Experiment 11. The highest accuracy per sequence was 83.73% in Experiment 1 and 84.08% in Experiment 11, while the worst accuracy was 82.18% in Experiment 1 and 83.58% in Experiment 11. On the other hand, the highest per-file accuracy was 92.21% in Experiment 1 and 93.11% in Experiment 11, while the lowest accuracy was 91.25% in Experiment 1 and 92.46% in Experiment 11. As shown in Table 6, the standard deviation is low, especially in Experiment 11, which means that the accuracies of all runs are too close to the average accuracy. In addition, the system is yielding more stability regarding classification robustness.

**Table 6.** Summary results for selected experiments.

| **Testing Accuracy** | **Per Sequences (%)** | | **Per Files (%)** | |
|---|---|---|---|---|
| **GTCC and MFCC** | **Experiment 1** | **Experiment 11** | **Experiment 1** | **Experiment 11** |
| Run # 1 | 83.73 | 83.77 | 92.21 | 93.00 |
| Run # 2 | 82.31 | 83.61 | 91.79 | 92.61 |
| Run # 3 | 82.18 | 83.93 | 91.25 | 92.86 |
| Run # 4 | 82.28 | 84.08 | 91.86 | 93.11 |
| Run # 5 | 82.44 | 83.58 | 91.68 | 92.46 |
| **Average** | 82.59 | **83.79** | 91.76 | **92.81** |
| **Standard Deviation** | 0.64 | **0.21** | 0.35 | **0.27** |
| **The Best Accuracy** | 83.73 | **84.08** | 92.21 | **93.11** |
| **The Worst Accuracy** | 82.18 | **83.58** | 91.25 | **92.46** |

The choice between accuracy or time is a well-known dilemma, and many studies have dealt with the discussion of the interpretability vs. performance trade-off. It is not always true that more complex models produce more accurate results. However, this can be incorrect when the given data is structured and has meaningful features. The statement "models that are more complicated are more accurate" can be valid in cases when the function being approximated is complex, the given data is widely distributed among suitable values for each variable, and the given data is adequate to build a complex model. The trade-off between interpretability and performance is evident in this circumstance [52,53].

In our work, the Experiment 11 results were favorable compared to the Experiment 1 results regarding the proposed system's accuracy, while Experiment 1 outperformed Experiment 11 regarding the speed of execution. That means Experiment 1 is around 0.9% less accurate $((93 - 92.21)/92.21)$ but 27% faster $((221 - 162)/221)$ than Experiment 11. Figure 5 illustrates the language-identified accuracy results for the two experiments per sequence and file. The best language-identified accuracy was the Russian language, followed by Arabic. The worst language-identified accuracy was Spanish.

In particular, two conventional architectures also revealed difficulties in discriminating between English and Spanish due to their similarity [11]. The Spanish language is also similar to the German language [54]. In contrast, the French language is similar to Spanish [21]. German and English are more likely to be confused, but English has a slightly stronger bias toward French [2]. All in all, the learned representations of the architecture are pretty distinctive for each language. For more details, Tables 7 and 8 present the confusion matrices of the average five runs per file for Experiments 1 and 11. In addition, the average accuracy, precision, recall, and F-measure results are presented.

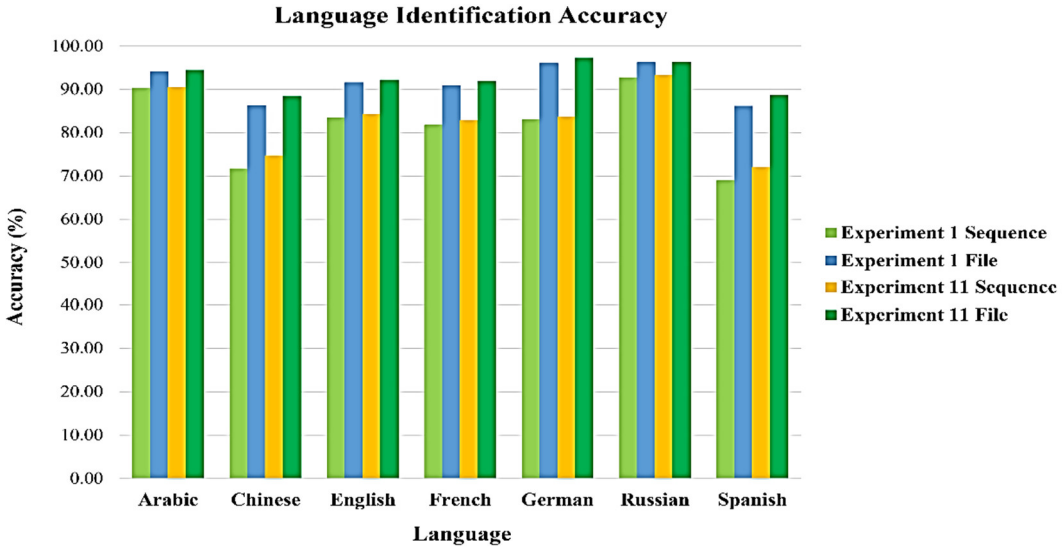


**Figure 5.** Language identification accuracies.

**Table 7.** The average five-run confusion matrices per file in Experiment 1.

| | | **Predicted** | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **Arabic** | **Chinese** | **English** | **French** | **German** | **Russian** | **Spanish** |
| **Actual** | Arabic | **377** | 5 | 3 | 4 | 8 | 2 | 1 |
| | Chinese | 4.6 | **345** | 14 | 9.8 | 15.6 | 5 | 6 |
| | English | 4 | 4.2 | **366.8** | 2 | 16 | 1 | 6 |
| | French | 2 | 3 | 1 | **364.8** | 21.2 | 3 | 5 |
| | German | 2 | 1 | 2.2 | 3 | **385.8** | 2 | 4 |
| | Russaian | 3 | 1 | 1.6 | 3 | 4 | **385.4** | 2 |
| | Spanish | 1 | 6 | 8.6 | 7.8 | 26 | 6.2 | **344.4** |
| **P(%)** | | 95.78 | 94.47 | 92.35 | 92.49 | 80.95 | 95.25 | 93.49 |
| **R(%)** | | 94.25 | 86.25 | 91.70 | 91.20 | 96.45 | 96.35 | 86.10 |
| **F1(%)** | | 95.01 | 90.17 | 92.02 | 91.84 | 88.02 | 95.80 | 89.64 |
| **Accuracy(%)** | | | | | **91.76** | | | |

**Table 8.** The average five-run confusion matrices per file in Experiment 11.

| | | **Predicted** | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **Arabic** | **Chinese** | **English** | **French** | **German** | **Russian** | **Spanish** |
| **Actual** | Arabic | **378.2** | 4 | 3.6 | 4 | 7 | 1 | 2.2 |
| | Chinese | 4.4 | **353.4** | 10.6 | 8.8 | 14.4 | 4.4 | 4 |
| | English | 3.8 | 3.2 | **369** | 2 | 16 | 1 | 5 |
| | French | 3 | 1.6 | 2 | **367.8** | 21.8 | 1.2 | 2.6 |
| | German | 1.8 | 0 | 1 | 2 | **390.2** | 1 | 4 |
| | Russian | 3.4 | 1 | 0 | 4.2 | 3 | **385.4** | 3 |
| | Spanish | 0 | 1.8 | 7 | 8.2 | 24.2 | 4.2 | **354.6** |
| **P(%)** | | 95.84 | 96.82 | 93.85 | 92.64 | 81.87 | 96.79 | 94.46 |
| **R(%)** | | 94.55 | 88.35 | 92.25 | 91.95 | 97.55 | 96.35 | 88.65 |
| **F1(%)** | | 95.19 | 92.39 | 93.04 | 92.29 | 89.02 | 96.57 | 91.46 |
| **Accuracy(%)** | | | | | **92.81** | | | |

### *6.3. Discussion*

From the Experiment 1 results in Table 7, five Arabic test files were identified as Chinese and eight as German. Fourteen Chinese test files were identified as English, nine as French, fifteen as German, five as Russian, and six as Spanish. Sixteen English test files

were identified as German and six as Spanish. Moreover, twenty-one French test files were identified as German and five as Spanish. In addition, six Spanish test files were identified as Chinese, eight as English, seven as French, twenty-six as German, and six as Russian. This experiment confuses Chinese and Spanish, English and Spanish, and Spanish and French. As mentioned previously, these are similar languages. Additionally, based on the results of the confusion matrices in Table 7, we can conclude that the German language is confused with Arabic, Chinese, English, French, and Spanish, as in the predicted German column. This implies a considerable similarity between German and these languages. We can also conclude that German is the only language that serves as a source of confusion for Arabic, as in the actual Arabic row. However, Arabic causes minimal confusion for all other languages, as in the predicted Arabic column.

From the Experiment 11 results in Table 8, seven Arabic test files were identified as German. Ten Chinese test files were identified as English, eight as French, and fourteen as German. Sixteen English test files were identified as German and five as Spanish. Twenty-one French test files were identified as German. Seven Spanish test files were identified as English, eight as French, and twenty-four as German. As the results indicate, this experiment still confuses English and Spanish. In contrast, the system was effective at distinguishing between Spanish and Chinese, as well as between French and Spanish. We can draw the same conclusions from Experiment 11 as from Experiment 1. In particular, German was the main source of confusion for all six other languages, as seen in Table 8. In addition, Arabic was not a source of confusion for any of the other six languages.

From Tables 7 and 8, German achieved the lowest P values (80.95% and 81.87% in Experiments 1 and 11, respectively) and the highest R values (96.45% and 97.55%, respectively), as well as the lowest F1 values: (88.02% and 89.02%, respectively). The impact of the German language is clear from the confusion matrices in these tables, where most spoken LID error files of any language, and Spanish in particular, were linked to German. Interpreting the relationships between these languages requires further linguistic research. Nevertheless, Tables 7 and 8 indicate that the proposed system is ideal for identifying Arabic spoken language and distinguishing it from the other six languages. Moreover, the system can discriminate among similar languages with the best accuracy. Table 9 presents the studies that used CRNN, CNN, LSTM architectures, or Mozilla corpus to compare with our proposed system.

**Table 9.** Comparison of proposed CRNN language identification architecture with other studies using different corpora with varying numbers of classes.

| Study and Ref. | Features | Classifiers | Corpus | Classes | Languages | Results (%) |
|---|---|---|---|---|---|---|
| Bartz et al. [2] | Spectrogram | CRNN | European Parliament Statements and News Channels on YouTube | 6 | English, French, German, Chinese, Russian, and Spanish | Accuracy = 91.00 |
| Kim and Park. [11] | Rhythm | R-vector with SVM | SiTEC | 2 | English and Korean | ER = 2.26 |
| | | | Mozilla | 3 | Chinese, English, and Spanish | ER = 53.27 |
| Zazo et al. [20] | MFCC-SDC | LSTM-RNN | NIST LRE | 8 | Dari, English, French, Chinese, Pashto, Russian, Spanish, and Urdu | Accuracy = 70.90 |
| Ma and Yu [27] | DNN-BN | LSTM | AP17-OLR | 10 | Tibetan, Japanese, Kazakh, Korean, Indonesian, Mandarin, Cantonese, Vietnamese, Uyghur, and Russian | ER = 50.00 |
| Draghici et al. [21] | Mel-spectrograms | CRNN | EU Repo | 6 | English, French, German, Greek, Italian, and Spanish | Accuracy = 83.00 |
| Singh et al. [33] | Log-Mel spectrograms | CNN | Mozilla | 4 | Estonian, Tamil, Turkish, and Mandarin | Accuracy = 80.21 |
| **Proposed System** | **GTCC-MFCC** | **CRNN** | **Mozilla** | **7** | **Arabic, German, English, Spanish, French, Russian, and Chinese** | **Accuracy = 92.81** |

As shown in Table 9, we clarified the features, corpus, languages, the proposed classifiers used in each study, and the number of classes to make a general comparison of the available studies in the spoken LID field. Compared to these studies, we found that our proposed system applied to languages of industrial countries, including Arabic, obtained the best results.

## 7. Conclusions

In this article, we proposed a spoken LID system using a CRNN architecture that works on combined GTCC and MFCC features of speech signals. At the feature extraction stage, the GTCC and MFCC features were compared, as well as a combination of both. The average precision of GTCC for spoken LID in Arabic was 94.88%, Chinese 94.20%, English 91.30%, French 90.58%, German 74.84%, Russian 92.98%, and Spanish 89.86%. In Experiment 1, by contrast, the average precision of MFCC for spoken LID in Arabic was 94.82%, Chinese 94.88%, English 89.76%, French 92.14%, German 76.78%, Russian 95.32%, and Spanish 91.72%. The results indicate that GTCC outperformed MFCC in Arabic and English spoken LID. We performed extensive experiments with different parameters and reported a state-of-the-art result in that the system in Experiment 11 achieved the highest overall average accuracy of 92.81% for identifying the seven spoken languages considered in the selected corpus.

The average precision when combining GTCC and MFCC in Experiment 11 for spoken LID in Arabic was 95.84%, Chinese 96.82%, English 93.85%, French 92.64%, German 81.87%, Russian 96.79%, and Spanish 94.46%. Furthermore, our experiments indicate that the proposed system gives the best results for identifying Arabic and Russian, discriminates among similar languages, and is extensible to new languages. Additionally, this article provides a good reference for people interested in developing Arabic-related ASR systems.

In future research, we can extend the number of languages and compare our CRNN architecture with other variations of deep learning architectures. Many other speech corpora can also be used for evaluation. The immediate benefit of an effective multi-lingual spoken LID system is that it can be developed based on the system's output for different purposes, including multi-lingual automatic translation.

**Author Contributions:** Conceptualization, A.A.A., M.A.Q., A.H.M. and Y.A.A.; methodology, A.A.A. and M.A.Q.; software, A.A.A. and M.A.Q.; validation, A.A.A., M.A.Q., A.H.M. and Y.A.A.; formal analysis, A.A.A., M.A.Q., A.H.M. and Y.A.A.; investigation, A.A.A. and Y.A.A.; resources, A.A.A. and M.A.Q.; data curation, A.A.A.; writing—original draft preparation, A.A.A., M.A.Q. and A.H.M.; writing—review and editing, A.A.A. and Y.A.A.; visualization, A.A.A.; supervision, Y.A.A.; project administration, A.A.A.; funding acquisition, Y.A.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially supported by the Researchers Supporting Project number (RSP-2021/322), King Saud University, Riyadh, Saudi Arabia.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to thank Researchers Supporting Project at King Saud University.

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

## References

1. Lounnas, K.; Satori, H.; Hamidi, M.; Teffahi, H.; Abbas, M.; Lichouri, M. CLIASR: A Combined Automatic Speech Recognition and Language Identification System. In Proceedings of the 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 16–19 April 2020; pp. 1–5. [CrossRef]
2. Bartz, C.; Herold, T.; Yang, H.; Meinel, C. Language Identification Using Deep Convolutional Recurrent Neural Networks. *arXiv* **2017**. [CrossRef]
3. Fromkin, V.; Rodman, R.; Hyams, N.M. *An Introduction to Language*, 10th ed.; Wadsworth/Cengage Learning: Boston, MA, USA, 2014.
4. *The World's Major Languages*; Routledge Handbooks Online: London, UK, 2008. [CrossRef]
5. Crystal, D. *The Cambridge Encyclopedia of Language*, 3rd ed.; Cambridge University Press: Cambridge, NY, USA, 2010.

6. Shaalan, K.; Siddiqui, S.; Alkhatib, M.; Monem, A.A. Challenges in Arabic Natural Language Processing. In *Systems Computational Linguistics, Speech and Image Processing for Arabic Language*; World Scientific: Singapore, 2018; pp. 59–83. [CrossRef]
7. Alotaibi, Y.A.; Muhammad, G. Study on pharyngeal and uvular consonants in foreign accented Arabic for ASR. *Comput. Speech Lang.* **2010**, *24*, 219–231. [CrossRef]
8. Spoken Language Recognition: From Fundamentals to Practice. *IEEE J. Mag. IEEE Xplore* **2013**, *101*, 1136–1159. Available online: https://ieeexplore.ieee.org/document/6451097 (accessed on 26 February 2022).
9. Waibel, A.; Geutner, P.; Tomokiyo, L.; Schultz, T.; Woszczyna, M. Multilinguality in speech and spoken language systems. *Proc. IEEE* **2000**, *88*, 1297–1313. [CrossRef]
10. Schultz, T.; Waibel, A. Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Commun.* **2001**, *35*, 31–51. [CrossRef]
11. Kim, H.; Park, J.-S. Automatic Language Identification Using Speech Rhythm Features for Multi-Lingual Speech Recognition. *Appl. Sci.* **2020**, *10*, 2225. [CrossRef]
12. Liu, D.; Xu, J.; Zhang, P.; Yan, Y. A unified system for multilingual speech recognition and language identification. *Speech Commun.* **2020**, *127*, 17–28. [CrossRef]
13. Chelba, C.; Hazen, T.; Saraclar, M. Retrieval and browsing of spoken content. *IEEE Signal Process. Mag.* **2008**, *25*, 39–49. [CrossRef]
14. Walker, K.; Strassel, S. The RATS radio traffic collection system. In *Odyssey Speaker and Language Recognition Workshop*; ISCA: Cape Town, South Africa, 2012.
15. Shen, P.; Lu, X.; Li, S.; Kawai, H. Knowledge Distillation-Based Representation Learning for Short-Utterance Spoken Language Identification. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 2674–2683. [CrossRef]
16. Srinivas, N.S.S.; Sugan, N.; Kar, N.; Kumar, L.S.; Nath, M.K.; Kanhe, A. Recognition of Spoken Languages from Acoustic Speech Signals Using Fourier Parameters. *Circuits Syst. Signal Process.* **2019**, *38*, 5018–5067. [CrossRef]
17. He, K.; Xu, W.; Yan, Y. Multi-Level Cross-Lingual Transfer Learning With Language Shared and Specific Knowledge for Spoken Language Understanding. *IEEE Access* **2020**, *8*, 29407–29416. [CrossRef]
18. Padi, B.; Mohan, A.; Ganapathy, S. Towards Relevance and Sequence Modeling in Language Recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 1223–1232. [CrossRef]
19. Nofal, M.; Abdel-Reheem, E.; El Henawy, H. Arabic/English automatic spoken language identification. In Proceedings of the 1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 1999). Conference Proceedings (Cat. No.99CH36368), Victoria, BC, Canada, 22–24 August 1999; pp. 400–403. [CrossRef]
20. Zazo, R.; Lozano-Diez, A.; Gonzalez-Dominguez, J.; Toledano, D.T.; Gonzalez-Rodriguez, J. Language Identification in Short Utterances Using Long Short-Term Memory (LSTM) Recurrent Neural Networks. *PLoS ONE* **2016**, *11*, e0146917. [CrossRef]
21. Draghici, A.; Abeßer, J.; Lukashevich, H. A study on spoken language identification using deep neural networks. In Proceedings of the 15th International Conference on Audio Mostly, New York, NY, USA, 15–17 September 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 253–256. Available online: https://doi.org/10.1145/3411109.3411123 (accessed on 17 June 2022).
22. Guha, S.; Das, A.; Singh, P.K.; Ahmadian, A.; Senu, N.; Sarkar, R. Hybrid Feature Selection Method Based on Harmony Search and Naked Mole-Rat Algorithms for Spoken Language Identification From Audio Signals. *IEEE Access* **2020**, *8*, 182868–182887. [CrossRef]
23. Sangwan, P.; Deshwal, D.; Dahiya, N. Performance of a language identification system using hybrid features and ANN learning algorithms. *Appl. Acoust.* **2021**, *175*, 107815. [CrossRef]
24. Garain, A.; Singh, P.K.; Sarkar, R. FuzzyGCP: A deep learning architecture for automatic spoken language identification from speech signals. *Expert Syst. Appl.* **2021**, *168*, 114416. [CrossRef]
25. Shen, P.; Lu, X.; Kawai, H. Transducer-based language embedding for spoken language identification. *arXiv* **2022**, arXiv:2204.03888.
26. Das, A.; Guha, S.; Singh, P.K.; Ahmadian, A.; Senu, N.; Sarkar, R. A Hybrid Meta-Heuristic Feature Selection Method for Identification of Indian Spoken Languages From Audio Signals. *IEEE Access* **2020**, *8*, 181432–181449. [CrossRef]
27. Ma, Z.; Yu, H. Language Identification with Deep Bottleneck Features. *arXiv* **2020**, arXiv:1809.08909. Available online: http://arxiv.org/abs/1809.08909 (accessed on 23 February 2022).
28. Alshutayri, A.; Albarhamtoshy, H. Arabic Spoken Language Identification System (ASLIS): A Proposed System to Identifying Modern Standard Arabic (MSA) and Egyptian Dialect. In Proceedings of the Informatics Engineering and Information Science Conference, Kuala Lumpur, Malaysia, 12–14 November 2011; Springer: Berlin/Heidelberg, Germany; pp. 375–385. [CrossRef]
29. Mohammed, E.M.; Sayed, M.S.; Moselhy, A.M.; Abdelnaiem, A.A. LPC and MFCC Performance Evaluation with Artificial Neural Network for Spoken Language Identification. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2013**, *6*, 55.
30. Pimentel, I. The Top 10 Languages in Higher Demand for Business. Available online: https://blog.acolad.com/the-top-10-languages-in-higher-demand-for-business (accessed on 21 August 2022).
31. "10 Foreign Languages in Demand across the Globe". Education World, 19 November 2018. Available online: https://www.educationworld.in/foreign-languages-in-demand-across-the-globe/ (accessed on 21 August 2022).
32. Sisodia, D.S.; Nikhil, S.; Kiran, G.S.; Sathvik, P. Ensemble Learners for Identification of Spoken Languages using Mel Frequency Cepstral Coefficients. In Proceedings of the 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 28–29 February 2020; pp. 1–5. [CrossRef]

33. Singh, G.; Sharma, S.; Kumar, V.; Kaur, M.; Baz, M.; Masud, M. Spoken Language Identification Using Deep Learning. *Comput. Intell. Neurosci.* **2021**. [CrossRef]
34. Alashban, A.A.; Alotaibi, Y.A. Speaker Gender Classification in Mono-Language and Cross-Language Using BLSTM Network. In Proceedings of the 2021 44th International Conference on Telecommunications and Signal Processing (TSP), Brno, Czech Republic, 26–28 July 2021; pp. 66–71. [CrossRef]
35. Mozilla Common Voice. Available online: https://commonvoice.mozilla.org/ (accessed on 27 February 2022).
36. Ardila, R.; Branson, M.; Davis, K.; Henretty, M.; Kohler, M.; Meyer, J.; Morais, R.; Saunders, L.; Tyers, F.M.; Weber, G. Common Voice: A Massively-Multilingual Speech Corpus. *arXiv* **2020**, arXiv:1912.06670. Available online: http://arxiv.org/abs/1912.06670 (accessed on 27 December 2021).
37. Automatic Speech Recognition: A Deep Learning Approach—PDF Drive. Available online: http://www.pdfdrive.com/automatic-speech-recognition-a-deep-learning-approach-e177783075.html (accessed on 30 March 2022).
38. Alashban, A.A.; Alotaibi, Y.A. Language Effect on Speaker Gender Classification Using Deep Learning. In Proceedings of the 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), Vijayawada, India, 12–14 February 2022; pp. 1–6. [CrossRef]
39. Detect Boundaries of Speech in Audio Signal—MATLAB detectSpeech—MathWorks Switzerland. Available online: https://ch.mathworks.com/help/audio/ref/detectspeech.html (accessed on 17 August 2022).
40. Journal, I. Extracting Mfcc and Gtcc Features for Emotion Recognition from Audio Speech Signals. Available online: https://www.academia.edu/8088548/EXTRACTING_MFCC_AND_GTCC_FEATURES_FOR_EMOTION_RECOGNITION_FROM_AUDIO_SPEECH_SIGNALS (accessed on 31 March 2022).
41. Kotsakis, R.; Matsiola, M.; Kalliris, G.; Dimoulas, C. Investigation of Spoken-Language Detection and Classification in Broadcasted Audio Content. *Information* **2020**, *11*, 211. [CrossRef]
42. Dua, S.; Kumar, S.S.; Albagory, Y.; Ramalingam, R.; Dumka, A.; Singh, R.; Rashid, M.; Gehlot, A.; Alshamrani, S.S.; AlGhamdi, A.S. Developing a Speech Recognition System for Recognizing Tonal Speech Signals Using a Convolutional Neural Network. *Appl. Sci.* **2022**, *12*, 6223. [CrossRef]
43. Nisar, S.; Shahzad, I.; Khan, M.A.; Tariq, M. Pashto spoken digits recognition using spectral and prosodic based feature extraction. In Proceedings of the 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), Doha, Qatar, 4–6 February 2017; pp. 74–78. [CrossRef]
44. Liu, G.K. Evaluating Gammatone Frequency Cepstral Coefficients with Neural Networks for Emotion Recognition from Speech. *arXiv* **2018**, arXiv:1806.09010.
45. Liu, J.-M.; You, M.; Li, G.-Z.; Wang, Z.; Xu, X.; Qiu, Z.; Xie, W.; An, C.; Chen, S. Cough signal recognition with Gammatone Cepstral Coefficients. In Proceedings of the 2013 IEEE China Summit and International Conference on Signal and Information Processing, Beijing, China, 6–10 July 2013; pp. 160–164. [CrossRef]
46. Alcaraz, J.C.; Moghaddamnia, S.; Peissig, J. Efficiency of deep neural networks for joint angle modeling in digital gait assessment. *EURASIP J. Adv. Signal Process* **2021**, *2021*, 10. [CrossRef]
47. Sequence Folding Layer—MATLAB—MathWorks Switzerland. Available online: https://ch.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.sequencefoldinglayer.html#mw_e600a552-2ab0-48a8-b1d9-ae672b821805 (accessed on 18 August 2022).
48. Sequence Unfolding Layer—MATLAB—MathWorks Switzerland. Available online: https://ch.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.sequenceunfoldinglayer.html?searchHighlight=unfolding%20layer&s_tid=srchtitle_unfolding%20layer_1 (accessed on 18 August 2022).
49. Flatten Layer—MATLAB—MathWorks Switzerland. Available online: https://ch.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.flattenlayer.html?searchHighlight=flatten%20layer&s_tid=srchtitle_flatten%20layer_1 (accessed on 18 August 2022).
50. Time Series Forecasting Using Hybrid CNN—RNN. Available online: https://ch.mathworks.com/matlabcentral/fileexchange/91360-time-series-forecasting-using-hybrid-cnn-rnn (accessed on 30 March 2022).
51. Qamhan, M.A.; Altaheri, H.; Meftah, A.H.; Muhammad, G.; Alotaibi, Y.A. Digital Audio Forensics: Microphone and Environment Classification Using Deep Learning. *IEEE Access* **2021**, *9*, 62719–62733. [CrossRef]
52. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
53. Saeed, W.; Omlin, C. Explainable AI (XAI): A Systematic Meta-Survey of Current Challenges and Future Opportunities. *arXiv* **2021**. [CrossRef]
54. Božinović, N.; Perić, B. The role of typology and formal similarity in third language acquisition (German and Spanish). *Stran-Jez.* **2021**, *50*, 9–30. [CrossRef]