Received 14 January 2024, accepted 7 March 2024, date of publication 11 March 2024, date of current version 20 March 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3376237*

SURVEY

# Arabic Speech Recognition: Advancement and Challenges

**ASHIFUR RAHMAN[1], MD. MOHSIN KABIR[2,3], M. F. MRIDHA[4], (Senior Member, IEEE), MOHAMMED ALATIYYAH[5], HAIFA F. ALHASSON[6], (Member, IEEE), AND SHUAA S. ALHARBI[6], (Member, IEEE)**

[1] RIoT Research Center, Independent University, Dhaka 1229, Bangladesh
[2] Superior Polytechnic School, University of Girona, 17004 Girona, Spain
[3] Faculty of Informatics, Eötvös Loránd University, 1117 Budapest, Hungary
[4] Department of Computer Science and Engineering, American International University-Bangladesh, Dhaka 1229, Bangladesh
[5] Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia
[6] Department of Information Technology, College of Computer, Qassim University, Buraydah 52571, Saudi Arabia

Corresponding author: M. F. Mridha (firoz.mridha@aiub.edu)

This work was supported by Prince Sattam bin Abdulaziz University under Project PSAU/2023/R/1445.

**ABSTRACT** Speech recognition is a captivating process that revolutionizes human-computer interactions, allowing us to interact and control machines through spoken commands. The foundation of speech recognition lies in understanding a given language's linguistic and textual characteristics. Although automatic speech recognition (ASR) systems flawlessly convert speech into text for various international languages, their implementation for Arabic remains inadequate. In this research, we diligently explore the current state of Arabic ASR systems and unveil the challenges encountered during their development. We categorize these challenges into two groups: those specific to the Arabic language and those more general. We propose strategies to overcome these obstacles and emphasize the need for ASR architectures tailored to the Arabic language's unique grammatical and phonetic structure. In addition, we provide a comprehensive and explicit description of various feature extraction methods, language models, and acoustic models utilized in the Arabic ASR system.

**INDEX TERMS** Arabic speech recognition, speech recognition, Arabic speech-to-text, ASR technology, voice recognition.

## I. INTRODUCTION

Without a doubt, speech is the most captivating and effective communication between individuals. Furthermore, it has proven to be an exceptional tool for interacting with machines. As a result, the study of speech recognition has transitioned from controlled laboratory experiments to practical and real-world applications. Consequently, speech recognition systems are now commonly encountered and embraced in our everyday use of various applications [1]. In today's world, our reliance on ASR (Automatic Speech Recognition) systems is ever-present, making it essential for these systems to deliver the utmost accuracy. Users expect a seamless experience using voice search features or automated calling functions based on ASR. Any inaccuracies, such as jumbled or incorrect words, can lead to frustrating interruptions. Therefore, creating a reliable ASR system necessitates thoroughly examining speech-to-text translation mechanisms, encompassing aspects like grammar and word-level comprehension.

Language dependency poses a significant challenge for speech recognition systems, which must be tailored to a particular language. This means a design optimized for recognizing English speech might not perform as accurately when processing other languages with different linguistic properties. The complexity of this issue is evident in languages like Arabic, which exhibit even more diverse structural and grammatical variations than English. Surprisingly, despite its importance, language dependency has not received extensive attention from researchers.

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

Most existing literature on Automatic Speech Recognition (ASR) focuses on aspects like algorithm selection, handling speech variation challenges, and exploring architectural improvements rather than delving into the intricacies of language dependency. Table 1 comprehensively compares various recent analyses carried out in ASR. In light of this, our research thoroughly examines the grammatical elements involved in speech recognition. We also explore the challenges faced by algorithms concerning grammar and phonetics.

This paper examines obstacles and possibilities inherent in developing an Arabic ASR system. The main focus of this research lies in its fundamental contributions, which encompass:

- We have thoroughly examined various Arabic Automatic Speech Recognition (ASR) systems, encompassing speech datasets and architectural approaches. As far as we know, no extensive survey has been conducted to explore the architectural and grammatical aspects of Arabic ASR systems.
- We analyze the difficulties while developing Arabic Automatic Speech Recognition (ASR) systems. Additionally, we delve into the intricacies of these challenges.
- We also provide a comprehensive and explicit description of various feature extraction methods, language models, and acoustic models for the Arabic ASR system.
- In conclusion, we outline potential future avenues to consider when designing architectures. Additionally, we suggest an ideal framework that holds promise in tackling the difficulties encountered by Arabic ASR systems.

The following parts of this document are structured in the following manner: In Section II, we familiarize ourselves with the existing generic architectures explored in the domain of Arabic ASR. Moving on to Section III, we discuss some commonly used dataset and their preprocessing methods for Arabic ASR systems. In Section IV, we investigate the Feature Extraction method, unraveling the process of transforming raw audio data into informative representations. Section V navigates through Language Modeling, elucidating the construction of models capturing spoken language patterns. In Section VI, we explore cutting-edge methodologies within the ASR domain. Moving to Section VII, Decoding and Recognition algorithms are dissected. Section VIII briefly describes the evaluation matrix for the Arabic ASR system. In Section IX, a comprehensive analysis of the challenges the Arabic ASR system poses is presented. Section X concisely summarizes the potential research scope and suggested guidelines for future developments in Arabic automated speech recognition systems. Finally, Section XI concludes the paper.

## II. RELATED WORKS

### A. PREVIOUS ARABIC ASR

The exploration of Arabic ASR systems initiation can be traced back to the latter part of the 2000s [2]. Numerous collaborative studies have been carried out to explore this area. The methods employed in previous research mainly revolved around machine learning or deep learning approaches. Within this segment, we will delve into the current state of Arabic ASR systems, examining their advancements and features.

#### 1) MACHINE LEARNING METHODS FOR ARABIC ASR

Over the past few years, there has been a notable interaction between machine learning (ML) and automatic speech recognition (ASR) circles, as evident from the inclusion of dedicated workshops and speech-processing sessions in ML-centric conferences.

Over the years, various machine-learning techniques have been utilized in creating ASR systems, particularly for Arabic speech recognition. ASR is a pivotal driver behind historically prevalent machine learning (ML) methods, including hidden Markov models, discriminative learning, structured sequential learning, adaptive learning, and Bayesian learning. Furthermore, machine learning in ASR enables large-scale practical testing of various techniques. It facilitates the exploration of new challenges arising due to speech's sequential and ever-changing characteristics. Consequently, there is a need to develop a robust machine capable of accurately distinguishing everyday human speech from other speakers.

The most commonly used toolkits for developing the Arabic ASR system were the Hidden Markov Model (HMM) toolkit, also known as HTK [10], and the Kaldi toolkit. The study highlighted the algorithms and techniques employed to model the acoustic-phonetic patterns of Arabic speech using HMMs and Kaldi toolkit [11]. In a correlated investigation conducted by Alotaibi and Hussain [12], they designed an ASR system focused on Arabic vowels using HMM. The ASR system was segmented into three distinct modules, each assigned specific functions. The initial training module was designed to create insights into speech and language, forming a foundational resource for the system's operation. The second module, the HMM model bank, stored and organized the knowledge acquired in the previous module. Lastly, the recognizer module was responsible for interpreting the meaning of voice inputs during the testing phase, utilizing the aforementioned HMM models. The HMM Toolkit (HTK), developed in 2002 [13], is a portable and versatile toolkit designed to create and influence HMM models. Its primary application is in speech recognition, but it can also be utilized for various other research tasks related to ASR. HTK offers extensive tools that facilitate HMM training, manipulation, and working with pronunciation dictionaries, n-gram models, finite-state language models, and speech recording and transcription [12]. This comprehensive toolkit is valuable for creating, experimenting, and deploying ASR systems and associated research pursuits.

In their study, EL-Mashad et al. [14] explored the recognition of Arabic speech speakers using SVM models.

**TABLE 1.** Some of the previous review papers on automated speech recognition system.

| Ref | Examined approaches for extracting features | Examined approaches to deep learning | Talk about existing ASR methods | Reviewed datasets | Discussed grammatical variation | Core Contribution |
|---|---|---|---|---|---|---|
| [3] | × | ✓ | ✓ | ✓ | × | This paper primarily aims to investigate the utilization of sophisticated machine learning methods, such as RNN, NN, DNN, CNN, and DAE, to effectively identify and understand spoken Arabic language. |
| [4] | × | × | ✓ | ✓ | × | The paper primarily highlights recent progress in the field of Arabic ASR system. It covers essential components of a typical ASR system and proposes novel directions for potential upcoming research. |
| [5] | × | ✓ | ✓ | × | × | This paper offers a concise overview of the current research on utilizing deep learning methods for recognizing Arabic speech. Additionally, it underscores available resources and software kits for developing Arabic ASR systems. Furthermore, the paper showcases a practical usage example of Arabic Automatic Speech Recognition (AASR). |
| [6] | ✓ | ✓ | ✓ | × | × | This paper's primary achievement involves conducting a comprehensive Systematic Literature Review (SLR) to spotlight research trends in Arabic ASR. The goal is to provide researchers with valuable insights into the most impactful studies spanning a decade, from 2011 to 2021. |
| [7] | × | ✓ | ✓ | × | × | The main focus of this paper is to analyze the latest methods and significant advancements in Arabic speech recognition, particularly highlighting the use of end-to-end DL approaches |
| [8] | × | ✓ | ✓ | × | ✓ | The paper reviews the challenges of the most advanced methods for developing ASR systems in Arabic. |
| [9] | ✓ | ✓ | ✓ | ✓ | × | The central focal point of the paper is to highlight the recent advancements in dialectal Arabic ASR systems, analyze and evaluate several studies, and discuss the challenges and techniques used in developing these systems. |
| Ours | ✓ | ✓ | ✓ | ✓ | ✓ | This paper provides a concise overview of Arabic speech recognition, focusing on architectural strategies that address grammatical properties. |

Specifically, they focused on recognizing connected Arabic digits (numbers) by leveraging neural networks. The numbers utilized in the recognition phase formed the input for the neural networks. To create a comprehensive dataset, they compiled a corpus of 1000 values, encompassing 10000 numbers uttered by 20 speakers with diverse characteristics, such as different genders, ages, and physical conditions, and recorded in a noisy environment. Every recorded measurement was converted into 10 unique numerical representations. The features of these numerical representations were then extracted using the Mel-Frequency Cepstral Coefficients (MFCC) technique. By utilizing the SVM approach, the system achieved a performance level of 94%. Taleb et al. [15] was inspired by the recognition that the existing standards impose limitations on the potential advancements achievable through HMMs in speech recognition. Researchers have been investigating novel modeling approaches that explicitly incorporate temporal dynamics to enhance resilience, especially in noisy environments. The EUIST FP6 HIWIRE research project partly supported this study. Initially, dynamic linear models (DLM), which capture spatial similarities, were proposed for their application in speech recognition.

Ali et al. [11] first introduce a comprehensive recipe and resources for training Arabic ASR systems using the

**TABLE 2. Research efforts in Arabic ASR with machine learning are presented in the table. The "matching scheme" refers to how patterns are matched by comparing speech to words or speech to phonemes. The "features" column specifies the proposed architecture's method of extracting features. The "Model" column explains the type of architecture employed. The 'dataset' column showcases the training data employed for model training, while the 'accuracy' column displays the corresponding test accuracy results. The notation (-) means the author did not mention it in the paper.**

| Ref | Domain | Matching scheme | Features | Model | Dataset | Accuracy |
|---|---|---|---|---|---|---|
| [10] | Speech to Phoneme | Phoneme | MFCC | HMMs | Self-made | — |
| [12] | Speech and Vowel Recognition | Phoneme | Formants | HMMs | Self-made | 91.6% |
| [14] | Digit Recognition | Word | MFCC | SVM | Self-made | 94% |
| [15] | Speech Recognition | Phoneme | MFCC | HMM | Self-made | — |
| [16] | Digit Recognition | Word | — | HMM | Egyptian Colloquial Arabic connected digits corpus | 99.34% |
| [17] | Speech Recognition | Word | — | MTL | OSACT, L-HSAB, and T-HSAB | 95.20% |
| [18] | Speech and Digit Recognition | Word | — | HMMs | HQC-1, CAC-1, and ADC | — |
| [19] | Speech Recognition | Word | MFCC | HMMs | KACST | 95.92% and 96.29% |
| [20] | Speech Recognition | Word | MFCC | HMMs | Self-made | 98.01% |
| [21] | Speech Recognition | Word | LPC coefficient | HMMs | Self-made | 95.0% |
| [22] | Digit Recognition | Word | DDMFCC | GMM | Self-made | 99.31% |
| [23] | Speech Recognition | Word | MFCC | GMM-HMM and DNN-HMM | Arabic Speech Corpus for Isolated Words (ASCIW) | 39.2% |
| [24] | Speech Recognition | Word | MFCC | CNN-LSTM | Self-made | 34.4% |

KALDI toolkit. It details developing a prototype news system, incorporating phoneme-based models and a QCRI lexicon for improved performance and reproducibility. The paper [23] conducts a comparative study on GMM-HMM and DNN-HMM architectures for Arabic ASR in noisy environments. Evaluating performance using hybrid models, researchers employ the CMU Sphinx and KALDI toolkit, emphasizing noise-resilient training and testing on the Arabic speech corpus. Alsayadi et al. [24] investigate the effectiveness of end-to-end deep learning for diacritical Arabic ASR, utilizing Mel-Frequency Cepstral Coefficients and log Mel-Scale Filter Bank energies. It surpasses traditional ASR by introducing novel methods such as CTC-based ASR, CNN-LSTM, and attention-based approaches, demonstrating notable enhancements in word error rates. The adoption of state-of-the-art frameworks, ESPnet [25] and Espresso [26], further elevates performance, particularly showcasing the superior efficacy of CNN-LSTM with an attention framework in Arabic speech recognition. The study also underscores recent advancements in conventional ASR through the Kaldi toolkit.

In a study by Elmahdy et al. [16], a dialectal Arabic speech recognition system was developed utilizing an innovative multilingual approach. This approach incorporates multiple acoustic models based on HMM. The training and testing stages incorporated a speech corpus from news broadcasts, encompassing modern standard Arabic and colloquial Egyptian Arabic. Notably, the system attained an impressive accuracy level of 99.34%. The paper [17], explores instances of offensive and hateful language on social media platforms within the Arab region. They devise a multi-task learning approach to enhance the precision of identifying such content. The model's performance surpasses existing models documented in the literature for three datasets. Hyassat and Zitar [18] present the inaugural Arabic ASR system using SPHINX-IV and offer an automated toolkit for generating a Pronunciation Dictionary for both the Holy Qur'an and the standard Arabic language. In this study, three distinct sets of data are created: the Holly Qura'an Corpus (HQC-1), the command and control corpus (CAC-1), and the Arabic digits corpus (ADC). The research tackles the limited exploration of the Arabic ASR system and the difficulties arising from the absence of diacritic Arabic text and Pronunciation Dictionary.

### 2) DEEP LEARNING METHODS FOR ARABIC ASR

Deep learning, a sub-field of machine learning, draws inspiration from the information-processing capabilities of the human brain [27]. It employs multiple layers of complex structures or non-linear transformations to effectively learn from unstructured or unlabeled data [28]. DL has significantly advanced in various domains, such as speech recognition, machine translation, and natural language processing (NLP). In recent years, it has rapidly evolved in NLP, image recognition, handwriting recognition, computer vision, and ASR technology [29]. Notably, recent progress in DL has played a vital role in enhancing the precision and effectiveness of ASR systems.

Deep learning has become a robust approach for effectively classifying data, particularly in ASR, following significant advancements in computational and machine learning algorithms [30]. Wazir and Chuah [31] researched applying deep learning to speech recognition. The study utilized a dataset containing 1040 samples of Arabic, with 840 and 200 samples for training and testing, respectively. Feature extraction involved using MFCC and LSTM techniques. Remarkably, the study achieved an impressive accuracy rate of 94%. According to AbdAlmisreb et al. [32], the performance of Deep Neural Networks (DNN) utilizing the Maxout activation function and the MFCC for feature extraction was examined. The researchers proposed a dropout function to enhance the efficiency of the DNN during training, and experimental results demonstrated significant performance gains compared to the sigmoid and ReLU activation functions. Significantly, the deep architecture using Maxout activation showed remarkable results, showcasing the lowest error rate compared to other deep neural networks. These encompassed the RBM, DBN, CNN, TFNN, and CAE.

Emami and Mangu [33] conducted an extensive investigation on the utilization of neural networks for the Arabic ASR system, employing dispersed word representation. The neural network model demonstrated robust generalization capabilities, effectively addressing the challenge of data sparseness. The study encompassed diverse configurations of neural probabilistic models, experimentation with n-gram order parameters, output vocabulary, normalization methods, model size, and other relevant parameters. The experimental evaluation focused on Arabic news broadcasts and conversational broadcasts. The optimized neural network model showcased notable improvements compared to the 4-gram baseline model, achieving absolute reductions of up to 0.8% and relative word error rate (WER) reductions of 3.8%. However, it was noted that changing these parameters had little effect on the model's overall performance. In 2019, Zerari et al. [34] proposed a framework for Arabic ASR utilizing a neural network with LSTM. Mel Frequency (MF) and Filter Banks (FB) coefficients were used to extract features. These coefficients were encoded as vectors of specific sizes. Subsequently, an MLP was employed to process these vectors. The study incorporated deep architectures such as GRU and recurrent LSTM for classification tasks. Two distinct datasets were used: spoken digit recognition and spoken TV commands. The experiments were conducted on both datasets, achieving an accuracy of 95%. Furthermore, the use of delta features resulted in an accuracy exceeding 96%. Algihab et al. [5] employed small Recurrent Neural Networks (RNNs) to develop a limited vocabulary speech recognizer. The recognizer focused on isolated words such as "hirra" (cur), "manzel" (house), "tariq" (road), "chajara" (tree), "zeina" (zeina), and "ghinaa" (singing). Each word was individually detected using a dedicated RNN. The training process consisted of two phases: consistent training and discriminative training. During consistent training, various utterances of the specific word were used for training. During discriminative training, various utterances containing different words were incorporated, not just limited to the specific designated word. The training dataset comprised recordings from four female speakers in an environment devoid of background noise. A male and a female speaker were employed during testing, with each individual in a pristine environment.

Zada and Ullah [35] proposed a method in 2020 for Arabic language recognition by isolating digits using Convolutional Neural Networks (CNNs). They constructed a dataset comprising 50 utterances ranging from 0 to 9 for each digit. MFCC was employed for feature extraction, facilitating the digit isolation process. The CNN architecture consisted of four convolutional layers, ReLU activation, and max-pooling layers. The system underwent training and evaluation, achieving a benchmarked accuracy of 84.17%. TensorFlow [36] is a prominent framework widely utilized by developers, offering robust deep-learning capabilities. When integrated with other models, this library proves highly effective in speech recognition tasks. Notably, Alghamdi et al. [37] leveraged the power of TensorFlow to enhance the efficiency of the Forward-backward algorithm, specifically in English speech recognition. Deep learning implementation frameworks currently leverage the power of DNNs. Choubassi et al. [38] proposed a novel methodology for developing an Arabic-isolated ASR system using modular recurrent Elman neural networks (MRENN). The researchers reported promising findings, indicating that this innovative neural network approach exhibits competitiveness comparable to traditional HMM-based speech recognition methods. The study included a tabular representation of the achieved results, encompassing six speakers, with some recordings conducted in noisy backgrounds while others in clean environments. Notably, the accuracy of speaker recognition varied from around 85% to a perfect 100% for different individuals.

In this paper, Messaoudi et al. [39] propose a methodology for developing an end-to-end Tunisian dialect speech system based on deep learning. The "TunSpeech" dataset contains paired text-speech data for the Tunisian dialect. During the study, existing Modern Standard Arabic (MSA) speech data was combined with dialectal Tunisian data, which reduced Out-of-Vocabulary rates and improved perplexities. The Word Error Rate increased when synthetic dialectal data was extracted from text-to-speech.

Recently, Ameen et al. [40] to identify documented signals from the Servox Digital EL Electro-Larynx developed an autoencoder that combines LSTM and GRU models. There were three steps in the proposed framework: denoising, feature extraction, and Arabic speech recognition. The best autoencoder was constructed by combining LSTMs and GRUs. Using LSTM & GRU models, Mahmoudi and Bouami [41] proposes two classes of Arabic speech commands based on the Arabic Speech Commands Dataset. The model's training utilized a GPU with NVIDIA's CUDA to expedite the training process. Throughout the training phase, multiple experiments were carried out to assess the influence of different factors on the system's performance and identify the optimal parameters for the model. The outcomes of these experiments indicated that the proposed method achieved satisfactory levels of accuracy in training, validation, and testing.

A new transcribed corpus of Yamani Arabic, Jordanian Arabic, and multi-dialectal Arabic is presented in [42]. Several baseline sequence-to-sequence DNN models were also designed for end-to-end recognition of Arabic dialects. Additionally, Mozilla's DeepSpeech2 model was trained from scratch using our corpora. With a 59% WER and a 51% CER, the Bidirectional LSTM (Bi-LSTM) with Attention model performed inspiring results on the Yamani speech corpus. On the Jordanian speech corpus, the Bi-LSTM with attention performed 83% WER and 70% CER concerning the Jordanian speech corpus. Comparatively, the model was able to produce 53% WER and 39% CER on the multi-dialectal Yem-Jod-Arab speech corpus. In the Yamani corpus, DeepSpeech2 has achieved 31% better WER and 24% better CER than the baseline model; in the Jordanian corpus, 68 WER and 40 CER have been achieved. Finally, DeepSpeech2 provided better results, with 30% WER and 20% CER, when applied to a multi-dialect Arabic corpus.

As a versatile machine-learning paradigm, deep learning leverages the principle of compositionality to represent the surrounding world efficiently. It encompasses utilizing DNNs, which undergo proper training to uncover intricate representations, starting from simpler ones progressively. The application of this principle extends to various practical challenges, such as the recognition of human speech [43]. The current implementation of the deep learning approach involves the use of DNNs. These networks, belonging to the broader category of Artificial Neural Networks (ANNs), consist of multiple concealed layers between the input and output layers. Each of these layers captures more complex attributes, which are then refined by the following layers in the network [44]. A comprehensive benchmarking of transformer speech recognition (ASR), modular HMM-DNN speech recognition (ASR), and human speech recognition (HSR) is performed on Arabic languages and their dialects in [45]. Our investigation evaluates how linguists and native speakers without linguistic expertise perform on a recently gathered dataset integral to our research. The implementation of end-to-end Automatic Speech Recognition (ASR) has yielded new performance benchmarks, with WER of 12.5%, 27.5%, and 33.8% for MGB2, MGB3, and MGB5, respectively. The outcomes of our study suggest that human proficiency in the Arabic language remains significantly superior to machine performance, demonstrating an average absolute WER gap of 3.5%.

### B. RECENT ADVANCES IN ASR TECHNOLOGIES

The world witnessed the advent of the speech recognition system in 1920, which represented a groundbreaking achievement as the first-ever machine capable of understanding and deciphering spoken language [47]. Subsequently, advancements in speech recognition technology were propelled forward through the dedicated efforts of researchers worldwide. These individuals, intrigued by the potential of speech recognition systems, brought forth and embraced numerous cutting-edge techniques, continuously enhancing the accuracy of such systems. Among these techniques were pattern-matching strategies, including brute-force methods, phonetic segmentation, and hybrid systems, which found their initial applications in speech recognition. Nevertheless, significant advancements were observed after the introduction of HMM [48] in the late 1970s. HMM has gained widespread popularity in ASR systems due to its enhanced capabilities in analyzing complex patterns across extensive vocabularies [49], [50], making it a practical and viable choice for implementation [51].

Recently, advancements in Artificial Neural Network (ANN) architectures have led to notable enhancements in speech recognition systems based on neural networks. Prominent DNN architectures like CNNs [52] and Residual Networks [53] are being successfully integrated into Automatic Speech Recognition (ASR) systems, demonstrating their efficacy and superior performance. DNN-based structures have demonstrated superior effectiveness to alternative architectures employed in the Arabic ASR system [53]. Various well-known techniques such as Principal Component Analysis (PCA) [54], Independent Component Analysis (ICA) [55], Wavelet Analysis [56], and Linear Discriminant Analysis (LDA) [57] have been employed for deriving speech characteristics from acoustic waveform. Among these methods, PCA is commonly used to find patterns in input data. However, a limitation of PCA is its ability to recognize only linear relationships within the data. In contrast, a DL-based approach called AutoEncoder can capture the non-linear characteristics of the data. As a result, AutoEncoders have gained popularity for embedding the non-linear aspects of the data in current applications. In speaker recognition tasks, probabilistic LDA (PLDA) is commonly applied to identify characteristics from speech embeddings. Both LDA and PLDA have been extensively investigated for their effectiveness in this context [58], [59]. Various dedicated feature extraction systems, such as MFCC [60], [61], Cepstral Mean Subtraction [62], and RASTA filtering [63], [64], have been utilized for extracting features from the waveform. Among these, MFCC has been extensively studied in speech and speaker recognition. MFCC is combined with various CNN architectures, improving speech recognition framework accuracy. The key to its success lies in the mel-scales of the MFCC, as a low-scale version filters out unwanted features and dramatically emphasizes the speech's phonetic components [65].

An ASR system typically employs two main processing approaches commonly found in practice: a) Feature Extraction and b) Pattern Matching. Feature extraction involves analyzing the acoustic waveform of speech to extract relevant speech parameters with acoustic correlations [66]. On the other hand, pattern matching entails comparing the extracted speech features with the appropriate patterns stored in the system's database to determine the correct output [67]. We have two types of pattern matching in this context: speech-to-phoneme matching [68] and speech-to-word matching [69]. However, we have devised a hybrid approach capable of performing both tasks. In ASR architectures, the term 'hybrid' typically refers to systems that blend the HMM and MLP methods [70], [71]. This research establishes the notion of 'hybrid' as integrating speech-to-phoneme and speech-to-text methodologies. By skillfully combining and adjusting these two key approaches (feature extraction and pattern matching), we can notably enhance the system's performance. Additionally, supplementary elements, like word segmentation,phoneme-to-word conversion, and noise reduction, are commonly incorporated into ASR systems to optimize their usability and resilience further. Figure 1 illustrates the complete sequence of operations within an ASR system. Additionally, Figure 2 presents an overview of the processes carried out in a hybrid ASR system.

**TABLE 3.** Research efforts in Arabic ASR with DL are presented in the table. The "matching scheme" refers to how patterns are matched by comparing speech to words or speech to phonemes. The "features" column specifies the proposed architecture's method of extracting features. The "Model" column explains the type of architecture employed. The 'dataset' column showcases the training data employed for model training, while the 'accuracy' column displays the corresponding test accuracy results. The notation (-) means the author did not mention it in the paper.

| Ref | Domain | Matching scheme | Features | Model | Dataset | Accuracy |
|---|---|---|---|---|---|---|
| [31] | Digits Recognition | Word | MFCC | RNN+LSTM | Self-made | 94% |
| [32] | Phoneme Recognition | Phoneme | MFCC | DNN | Self-made | — |
| [33] | Speech Recognition | Word | MFCC | Neural Network | Arabic digit database | — |
| [34] | Speech and Digit Recognition | Word | MFCC and Filter Banks (FB) coefficients | Bidirectional LSTM+MLP | Arabic Gigaword corpus | 99.32% |
| [35] | Digit Recognition | Word | MFCC | CNN | Pashto digits | 84.17% |
| [38] | Speech Recognition | Word | Cepstral coefficients | MRENN | Self-made | 98.51% |
| [39] | Speech Recognition | Word | MFCC | RNN + KenLM | TunSpeech | — |
| [46] | Phoneme Recognition | Phoneme | MFCC | ANN | Self-made | 75.0% |
| [40] | Speech Recognition | Word | MFCC | Autoencoder | Self-made | 95.31% |
| [41] | Speech Recognition | Word | MFCC | LSTM + GRU | Arabic Speech Commands | – |
| [42] | Speech Recognition | Word | MFCC | DeepSpeech2 DNN+ Bi-LSTM | Self-made | – |

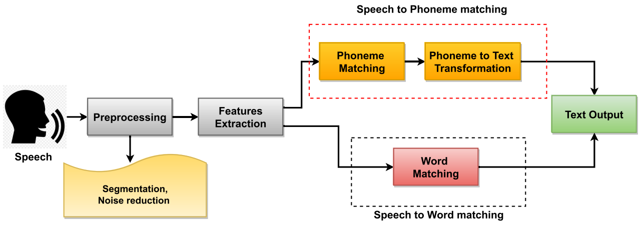


**FIGURE 1.** The diagram depicts the typical structure of hybrid ASR systems. Within the diagram, the red-dashed box symbolizes phoneme matching, while the black-dashed box represents word matching. These two crucial pattern-matching or classification methods are commonly employed in speech recognition models.

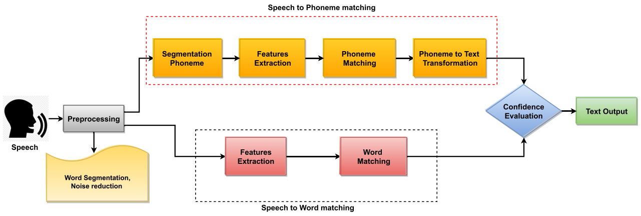


**FIGURE 2.** The diagram depicts the typical structure of hybrid ASR systems. Within the diagram, the red-dashed box symbolizes phoneme matching, while the black-dashed box represents word matching. These two crucial pattern-matching or classification methods are commonly employed in ASR architectures. The ultimate textual result is generated by assessing the confidence of the word-matching process.

In addition to conventional speech recognition approaches recent advancements in recurrent neural networks (RNN) have paved the way for a novel strategy known as end-to-end ASR [72]. An RNN-based architecture can simultaneously perform feature extraction and match speech to patterns holistically. The benefit of this approach lies in using a single loss function to train the complete network. The commonly employed loss function in such frameworks is Connectionist Temporal Classification (CTC) loss. However, a drawback of these methods is that they require a significant volume of data to attain precise outcomes [73]. Additionally, acquiring the best possible characteristics from the input flow also requires a significant investment of time. Figure 3 provides a graphical depiction of the fundamental structure of an end-to-end framework.

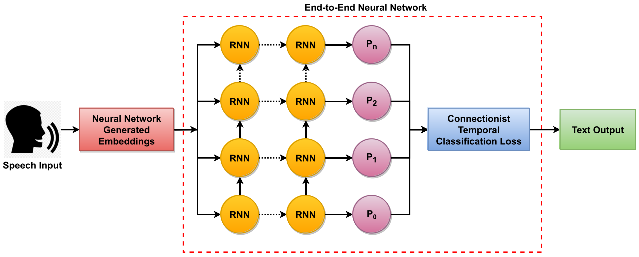

End-to-End Neural Network
Speech Input
Neural Network Generated Embeddings
RNN
$P_n$
$P_2$
$P_1$
$P_0$
Connectionist Temporal Classification Loss
Text Output

**FIGURE 3.** The provided diagram depicts a comprehensive end-to-end configuration for ASR systems. In this configuration, the neural network generates embeddings from input features, subsequently fed into a sequence of recurrent layers. These recurrent layers examine patterns by considering past and current input features, resulting in a conclusive output. The network is trained using the CTC loss function, effectively utilizing the backpropagation technique.

Specific adaptations of end-to-end architectures have demonstrated exceptional effectiveness in handling continuous speech and textual analysis. Notably, sequence-to-sequence (seq2seq) and attention-driven models have gained significant recognition. Seq2seq architectures involve an encoder and a decoder, consisting of numerous tiers of RNNs. The encoder generates valuable embeddings from the input information, guiding the decoder for precise prediction generation. Figure 4 demonstrates a typical situation within the seq2seq framework. In contrast, attention-based architectures [74] exhibit comparable performance to a seq2seq model [75]. More precisely, the attention mechanism is integrated with a seq2seq model, enabling it to leverage information from previous inputs and outputs, leading to a more advanced understanding of the network.

Extensive research has been devoted to exploring the integration of RNNs within end-to-end architectures. Consequently, two advanced techniques LSTM [76] and GRU [77] have emerged as a result of these investigations. Standard RNN-based models often suffer from the vanishing gradient problem, but LSTM and GRU networks overcome these issues. LSTM and GRU possess memory capabilities, making them more popular than general RNNs. GRU stands out for its efficiency as it requires fewer parameters than LSTM. However, LSTM has demonstrated superior performance in language modelling and speech recognition tasks [78].

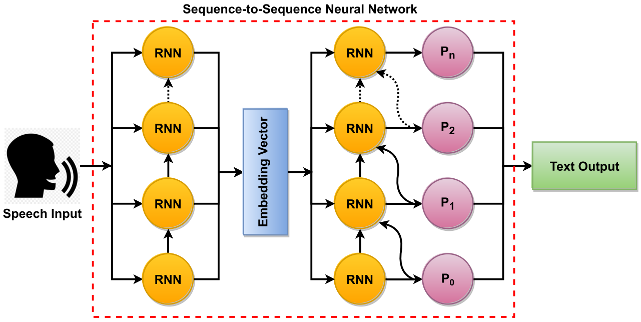


**FIGURE 4.** The diagram illustrates a typical speech recognition Seq2seq model. In this model, the encoder, which comprises a series of recurrent neural networks (RNNs), generates embedding vectors. These vectors are then passed to the decoder, another RNN, which produces the outcomes. Notably, the RNN in the decoder has access to previous predictions, potentially allowing subsequent predictions to be more precise.

Researchers in the ASR (Automatic Speech Recognition) field remain intrigued by recurrent architectures as they excel at deciphering intricate speech sequences.

## III. DATA COLLECTION AND PREPROCESSING

### A. DATABASES FOR ARABIC SPEECH RECOGNITION

Some progress has been made in developing Arabic speech recognition systems; however, there are still ample opportunities for further exploration. The main challenge lies in the scattered nature of the works, as the availability of suitable datasets for Arabic ASR is limited. Due to this scarcity, individual researchers have had to create their speech corpora, but unfortunately, these datasets have not been publicly shared. Consequently, it has been difficult to compare and verify the validity and standards of different databases and research efforts. As of now, our knowledge indicates the existence of nine available corpora for Arabic ASR systems. This comprises a speech dataset with real-number values, another dataset containing voice commands, and the remaining sets are composed of complete Arabic speech datasets. Table 4 provides an in-depth examination of these speech datasets.

To address the limited availability of Arabic speech datasets, a crucial step is the creation of extensive, openly accessible datasets of high quality. Such a dataset would serve diverse applications like speech-to-text processing, text-to-speech processing, speaker recognition, far-field speech recognition, and more [89]. When developing an Arabic speech dataset, careful attention should be given to the following scenarios:

- At present, speech datasets are designed to cater to particular scenarios, such as clean environments, telephony environments, broadcast settings (television/radio), meetings, distant surroundings, and real-world situations. Telephony, far-field, and in-the-wild environments are the most difficult among these. Consequently, cutting-edge speech recognition systems primarily focus on these challenging datasets.

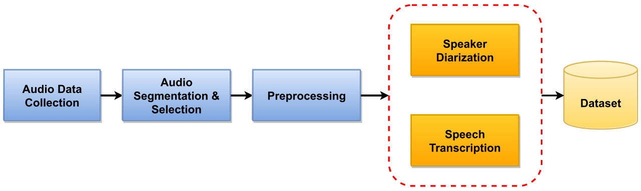


**FIGURE 5.** The picture demonstrates the overall process of creating a speech dataset for the Arabic language.

- An Arabic ASR dataset must include precise transcriptions of the spoken content. Additionally, it can incorporate supplementary details such as speaker characteristics (gender/emotion) and environmental context.
- To create a challenging and realistic speech dataset, it is essential to incorporate a wide range of features. These diverse aspects may include variations in input devices, dialects, age groups, environments, noise levels, and even speech disabilities.
- Many renowned datasets are organized into distinct clean and noisy subsets [90]. This segregation proves beneficial for researchers as it enables them to develop Arabic speech recognition prototypes tailored to specific scenarios.
- An Arabic speech dataset needs to focus on capturing Arabic-specific characteristics, including collecting speech samples from various dialects, gathering speech data for crucial and similar words, and giving special attention to addressing letter and utterance similarities.
- A comprehensive Arabic speech dataset should encompass an extensive vocabulary and effectively capture the diverse variations and statistics within the dataset.

Developing an Arabic speech dataset tailored for deep learning models poses significant challenges due to the extensive data required for training such architectures. Figure 5 outlines the critical stages involved in the data collection process. Gathering speech datasets could involve crowd-sourcing techniques or targeting specific populations for selection. Large datasets are frequently obtained through crowd-sourcing. When dealing with an Arabic speech dataset, performing additional statistical analysis to ensure a balanced representation across various domains is crucial. Pruning and carefully selecting data may also be necessary. Preprocessing the speech dataset is essential, involving tasks like optional noise cancellation, sound normalization, and reducing silent intervals. Additionally, creating speaker diarization and speech transcription requires a manual or semi-automated approach. Finally, the entire process needs thorough validation to produce a high-quality Arabic speech dataset.

### B. DATA CLEANING AND ANNOTATION

Data cleaning and annotation are vital steps in preparing the collected speech data for the training and evaluation of the Arabic ASR system [91]. These processes involve ensuring data quality, removing noise, and providing

**TABLE 4.** The table provides valuable information about the currently accessible dataset suitable for Arabic speech recognition.

| Ref | Summary |
|---|---|
| Alghmadi et al. [79] | The KACST Arabic Phonetics Database (KAPD) is a comprehensive database of Arabic sounds, containing more than 46,000 files. The dataset comprises information from nine trials involving eight native Arabic speakers. It encompasses multiple aspects like airflow, air pressure, oral contact, nasality, perception, and images of facial angles. Also included are stroboscopic images of the glottis, pharyngeal cavity, and velo-pharyngeal port. The dataset, known as KAPD, is unprocessed and holds potential for research and advancement in domains like speech therapy, synthesis, recognition, perception, and modeling. |
| Bhai et al. [21] | The dataset consists of recordings of 50 speakers uttering the ten Arabic digits three times each for training. The evaluation dataset consists of two categories: the first includes 30 speakers in the training phase, and the second comprises 10 new speakers. The authors used this dataset to develop and test their Arabic speech recognition system. |
| Elmisery et al. [80] | This dataset comprises isolated Arabic digits recorded by a male speaker. Each of the 10 digits is repeated 20 times. It includes 200 utterances for constructing a codebook. The speech data is sampled at 16KHz with 16-bit resolution and transformed into frames using defined parameters. |
| Amrouche et al. [81] | This corpus comprises 1800 spoken digits by 60 speakers (30 males, 30 females). Additionally, 1000 digits were included for testing, pronounced by 50 different speakers (25 males, 25 females). The dataset facilitates research in speaker-dependent and speaker-independent digit recognition tasks. |
| Bourouba et al. [82] | The dataset includes 92 Arabic speakers (46 men and 46 women), each enunciating 10 isolated words. Recordings are at 11.025 kHz with a 16-bit size. Learning utilizes 10/46 of the dataset, while testing employs 36/46. |
| Essa et al. [83] | This dataset contains 600 utterances, encompassing 10 speakers, 10 words, and 6 repetitions. It's divided into 300 training and 300 testing utterances, facilitating research in speech analysis. |
| Satori et al. [84] | This dataset is derived from 10 Arabic digits, spoken by 60 Moroccan speakers (35 males, 25 females). Each speaker pronounced all digits five times. |
| Qatab et al. [20] | This dataset comprises 3650 speech recordings featuring 13 distinct speakers. Training incorporates 3000 files, while testing encompasses 650. |
| Azmi et al. [85] | The dataset consists of recordings from 59 Egyptian men, with 33 speakers for training and 26 for testing. These speakers recited 16 proverbial sentences each. |
| Kolobov et al. [86] | MediaSpeech, a dataset designed for testing ASR systems, comprises short speech segments from YouTube videos. The dataset consists of four languages (French, Arabic, Turkish, and Spanish), each lasting 10 hours. This dataset is publicly available on the OpenSLR website. |
| Ubuntu et al. [87] | This dataset comprises the entire Quran comprising 6236 ayat and 114 suras. It includes multiple audio files featuring various reciters, where each file corresponds to a verse (ayat) from a surah. The dataset is publicly accessible on both the OpenSLR and Kaggle websites. |
| OpenSLR [88] | The Tunisian MSA corpus, collected in 2003 near Tunis, originally served to train acoustic models for pronunciation in Arabic language learning. Divided into recited and prompted speech subcorpora, it includes contributions from 118 informants, resulting in 11.2 hours of speech. Additionally, a small 2017 corpus, used for testing, features speech from 4 speakers, including 3 male Libyans and 1 female from Tunisia. |

accurate transcriptions to create a reliable and effective ASR dataset [92]. The following point outlines the procedures and methodologies for data cleaning and annotation:

- **Data Preprocessing:** The collected speech data will undergo thorough preprocessing to enhance its quality and prepare it for further analysis. Preprocessing steps may include noise removal, background normalization, and audio segmentation to isolate individual utterances.
- **Noise Removal:** Noise interference in the audio recordings can negatively impact ASR performance. Various methods for reducing noise, like spectral subtraction or Wiener filtering, will be implemented to minimise background noise and improve the clarity of the speech signal.
- **Audio Segmentation:** The audio recordings will be segmented into individual utterances, ensuring that each

segment contains only one complete speech instance. Properly segmenting the data is crucial for accurate alignment with corresponding transcriptions during training.

- **Speaker Identification:** To keep track of individual speakers' contributions, speaker identification will allow the ASR system to handle multi-speaker scenarios and variations in speech patterns.
- **Data Annotation:** Each segmented utterance will be accurately transcribed to create the ground truth text for the ASR system. The annotation process involves associating the text transcriptions with the corresponding audio segments, forming a labeled dataset for supervised training.

By meticulously cleaning and annotating the collected data, this study aims to produce a high-quality dataset that serves as the foundation for training and evaluating the ASR system. The accuracy and reliability of the dataset will significantly influence the ASR system's performance, making data cleaning and annotation critical steps in achieving accurate and efficient Arabic speech recognition.

## IV. ACOUSTIC FEATURE EXTRACTION

Acoustic feature extraction is a vital step in Arabic ASR systems, as it involves converting the raw speech signal into a set of relevant and compact features that can effectively represent the linguistic content of the speech [66], [93]. Various acoustic feature extraction methods have been explored for Arabic ASR, and some of the frequently employed methods include:

### A. MEL FREQUENCY CEPSTRAL COEFFICIENTS (MFCC)

MFCC is a widely used acoustic feature extraction technique in ASR systems [94]. It captures the essential characteristics of speech signals by representing them in the cepstral domain, effectively transforming the raw audio data into a compact feature space [95]. The MFCC computation involves several steps:

#### 1) PRE-EMPHASIS

The initial action involves employing a pre-emphasis filter to accentuate the higher frequencies within the speech signal, thereby equalizing spectral energy distribution. The pre-emphasis filter is defined as follows:

$$y[n] = x[n] - \alpha \cdot x[n-1] \tag{1}$$

Here, $y[n]$ is the pre-emphasized speech signal at time index n. $x[n]$ is the original speech signal at time index n and $\alpha$ is the pre-emphasis coefficient (usually set to 0.97), which controls the amount of emphasis.

#### 2) FRAME BLOCKING

The speech signal that has undergone pre-emphasis is segmented into short frames, usually lasting around 20-30 milliseconds, with overlapping. Each frame is chosen to be an appropriate size to capture the stationary characteristics of speech within a short time window.

#### 3) WINDOWING

Each frame undergoes a process where a window function, like the Hamming window, is employed. This helps minimize the spreading of spectral information at the edges of the frame and maintains a smooth progression. The windowed frame is given by:

$$w[n] = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{N-1}\right) \tag{2}$$

Here, $w[n]$ is the windowed frame and $N$ is the number of samples in the frame.

#### 4) FAST FOURIER TRANSFORM (FFT)

The DFT (Discrete Fourier Transform) calculates individual windowed frames to transform the signal from the time domain to the frequency domain. The DFT is commonly computed efficiently using the FFT algorithm. The equation for the FFT is,

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi kn/N} \tag{3}$$

Here, $X[k]$ is the frequency-domain representation of the sequence at frequency bin $k$. $x[n]$ is the time-domain sample at index $n$. $j$ is the imaginary unit ($j^2 = -1$) and $N$ is the total number of samples in the sequence.

#### 5) MEL FILTERBANK

The output of the FFT is run through a set of Mel filters. These filters are triangular-shaped and spaced evenly on the Mel scale, a perceptually relevant frequency scale. The output of each filter is the sum of the magnitudes of the FFT spectrum weighted by the filter's triangular shape. The equation for the Mel Filterbank is as follows:

For each filter $k$ in the Mel Filterbank:

- Define the center frequency $f_k$ of the $k$-th triangular filter in the Mel scale:

$$f_k = \frac{700 \cdot (m_k + 1)}{f_{\max}} \tag{4}$$

Here, $m_k$ is the index of the Mel filter, with $m_k = 0, 1, 2, \ldots, k-1$. $k$ is the number of filters in the filterbank and $f_{\max}$ is the maximum frequency of the spectrum, typically half of the sampling rate.

- Compute the frequencies at which the filter starts and ends:

$$f_{\text{start},k} = f_{k-1} \tag{5}$$

$$f_{\text{end},k} = f_{k+1} \tag{6}$$

- Transform the start and end frequencies back to the linear scale:

$$f_{\text{start},k} = \frac{f_{\text{start},k} \cdot f_{\max}}{700} \tag{7}$$

$$f_{\text{end},k} = \frac{f_{\text{end},k} \cdot f_{\max}}{700} \tag{8}$$

- Set the values of the triangular filter as follows:

$$H_k(n) = \begin{cases} 0, & \text{if } f_n < f_{\text{start},k}, \\ \dfrac{f_n - f_{\text{start},k}}{f_k - f_{\text{start},k}}, & \text{if } f_{\text{start},k} \le f_n < f_k, \\ \dfrac{f_{\text{end},k} - f_n}{f_{\text{end},k} - f_k}, & \text{if } f_k \le f_n < f_{\text{end},k}, \\ 0, & \text{if } f_n \ge f_{\text{end},k} \end{cases} \tag{9}$$

Here, $f_n$ is the frequency corresponding to the $n$-th bin of the FFT spectrum.

#### 6) LOGARITHM

The filterbank outputs are subjected to a logarithmic transformation, which converts their magnitudes into a scale that closely mirrors how humans perceive loudness. In mathematical notation, it is defined as:

$$y = \log_b(x) \tag{10}$$

#### 7) DISCRETE COSINE TRANSFORM (DCT)

Finally, the DCT is applied to the logarithmically scaled filterbank outputs to obtain the MFCC coefficients. The DCT decorates the filterbank outputs and decreases the dimensionality of the feature vector. The equation for the DCT is as follows:

$$c[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x[n] \cdot \cos\left(\frac{\pi k}{N}\left(n + \frac{1}{2}\right)\right)$$
$$\text{for} \quad k = 0, 1, \ldots, N-1 \tag{11}$$

Here, $c[k]$ is the $k$-th DCT coefficient. $x[n]$ is the $n$-th Mel filterbank output and $N$ is the number of Mel filterbank outputs.

The resulting MFCC coefficients illustrate the distinctive features of the speech signal in a compact representation, making them suitable for ASR systems. These coefficients are commonly used as input features for various ASR models, including HMMs, DNNs, and Transformer-based models.

Adding voiced formants and pitch features with MFCC, we can effectively address the expressive nature of Arabic. The work referenced in [96] demonstrates the significance of this combination in Arabic ASR, enhancing the system's ability to capture emotional content and nuances in speech. By incorporating pitch features, we capture variations indicating emphasis, excitement, or stress in spoken words, essential for accurate transcription in emotionally expressive languages like Arabic. This hybrid approach enriches the feature set, contributing to improved performance and recognizing the subtleties inherent in the language.

### B. PERCEPTUAL LINEAR PREDICTION (PLP)

PLP is one of the most used acoustic feature extraction methods in speech processing, known for its ability to capture the perceptually significant aspects of the speech signal [97], [98]. It aims to model the human auditory system's characteristics to improve ASR system performance, particularly in noisy environments. The PLP feature extraction process involves several steps, as described below:

#### 1) PRE-EMPHASIS

The first step is pre-emphasis, which emphasizes the higher frequencies in the speech signal to balance the spectral components. It helps to counteract the attenuation of high-frequency components during the speech production process. The pre-emphasis operation is represented in the equation 1.

#### 2) FRAMING AND WINDOWING

After applying pre-emphasis to the speech signal, it is segmented into frames of a consistent duration, usually around 20-30 milliseconds, and with some degree of overlap. Before conducting Fourier analysis, a windowing function such as Hamming or Hanning window is applied to each frame. This helps reduce spectral leakage for more accurate results. The equation of windowing is given in equation 2.

#### 3) POWER SPECTRUM ESTIMATION

Next, the power distribution across frequency components in each segmented portion is computed using the FFT. The magnitude squared of the FFT coefficients provides the power spectral density (PSD) of the signal, denoted as P(k) for the $k^{th}$ frequency bin. The Power Spectrum Estimation equation is given by:

$$P(k) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi kn/N}\right|^2 \tag{12}$$

Here, $P(k)$ is the estimated power spectral density at frequency index $k$. $N$ is the length of the signal (number of samples). $x[n]$ is the signal sample at time index $n$. $e^{-j2\pi kn/N}$ is the complex exponential term used in the DFT/FFT computation and $|\cdot|$ denotes the absolute value, and the square of the absolute value represents the power.

#### 4) MEL-FREQUENCY WRAPPING

The power spectrum $P(k)$ is then transformed into the Mel-frequency domain, mimicking the non-linear human perception of speech frequencies. This is achieved using triangular Mel filters, represented as $H_m(k)$, where m refers to the $m^{th}$ Mel filter. The Mel-frequency wrapping equation can be expressed as follows:

$$M(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \tag{13}$$

In this equation, $M(f)$ represents the Mel-frequency corresponding to the linear frequency $f$.

#### 5) NON-LINEAR COMPRESSION

A logarithmic operation is performed on the Mel-filtered power spectrum to account for the logarithmic nature of human perception. The logarithmic compression function is represented as $L_m(k)$, emphasizes smaller spectral details and attenuates large ones. The non-linear compression function is represented as:

$$L_m(k) = \log\left(1 + \beta \cdot |H_m(k)|^2\right) \tag{14}$$

Here, $L_m(k)$ is the compressed value of the $K$-th Mel-filtered energy in the $m$-th filter. $H_m(k)$ represents the magnitude of the $K$-th frequency component in the Mel-filtered power spectrum, and $\beta$ is a compression parameter that controls the strength of compression. Higher values of $\beta$ lead to stronger compression.

#### 6) CEPSTRAL COEFFICIENTS

The DCT is applied to the logarithmic compressed Mel-filtered energies to obtain the PLP cepstral coefficients, also known as PLP features. The DCT operation can be represented as:

$$C_p(m) = \sum_{k=1}^{N} \log(L_m(k)) \cdot \cos\left[\frac{\pi m}{N}(k - \frac{1}{2})\right] \tag{15}$$

Here, $C_p(m)$ represents the $m$-th Cepstral Coefficient. $L_m(k)$ is the compressed value of the $K$-th Mel-filtered energy in the $m$-th filter, as obtained from the non-linear compression. $N$ is the total number of Mel filters and $k$ ranges from 1 to $N$.

### *C. FILTER BANK ENERGIES (FBANK)*

Filter Bank Energies (FBANK) is a commonly used technique for extracting acoustic features in Automated Speech Recognition (ASR) systems [99]. The process entails breaking down the speech signal into various frequency ranges and measuring the energy within each range to capture the essence of the speech content. The process of calculating FBANK features can be outlined as follows:

#### 1) PRE-EMPHASIS

A pre-emphasis filter is applied to the raw speech signal to enhance higher-frequency components and reduce low-frequency noise. The pre-emphasis operation is defined in equation 1:

#### 2) FRAMING

The speech signal that has undergone pre-emphasis is segmented into short frames, usually lasting around 20-30 milliseconds, with overlapping. Overlapping frames are often used to better capture temporal information. Let's denote the frame length as $N$ samples.

#### 3) WINDOWING

All frames are multiplied with a window function, such as the Hamming or Hanning window, to decrease spectral leakage and minimize artifacts at the frame boundaries. Windowing is represented in equation 2.

#### 4) DISCRETE FOURIER TRANSFORM (DFT)

Next, the DFT transforms the signal from the time domain to the frequency domain. The magnitude spectrum $X[k]$ of the DFT is calculated as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi kn}{N}} \cdot w[n] \tag{16}$$

Here, $X[k]$ is the magnitude spectrum at frequency index $k$. $x[n]$ is the time-domain signal at time index $n$. $j$ is the imaginary unit and $N$ is the frame length in samples

#### 5) MEL FILTER BANK

FBANK features are obtained by passing the magnitude spectrum $X[k]$ through a set of Mel filters. The Mel filter bank is designed to approximate the non-linear human perception of pitch. The filter bank typically consists of triangular filters, and the filter outputs are computed as follows:

$$H_m[k] = \sum_{f=f_l}^{f_h} M_m(f) \cdot X[f] \tag{17}$$

Here, $H_m[k]$ is the output of the $m^{th}$ Mel filter at frequency index $k$. $M_m[f]$ is the Value of $m^{th}$ triangular Mel filter centered at frequency $f$. $f_l$ is the Lower frequency bound of the filter and $f_h$ is the Upper frequency bound of the filter

#### 6) LOGARITHMIC COMPRESSION

To mimic the logarithmic nature of human hearing, the filter bank energies are usually subjected to logarithmic compression:

$$\text{FBANK}[m] = \log(H_m[k]), \tag{18}$$

Here, FBANK$[m]$ is the FBANK coefficient for the $m^{th}$ filter.

The resulting FBANK feature vector, composed of the filter bank energies, is then used as input to the ASR system for further processing and recognition.

### *D. GAMMATONE FREQUENCY CEPSTRAL COEFFICIENTS (GFCC)*

The GFCC is a set of acoustic features commonly used in ASR systems [100], particularly for analyzing speech signals in the frequency domain. The GFCC aims to mimic the human auditory system's processing mechanism, which is sensitive to the various frequency bands in sound signals.

#### 1) GAMMATONE FILTERBANK

The Gammatone filterbank emulates the filtering traits of the human cochlea, breaking down speech signals into multiple frequency segments. For each filter, the output is obtained by convolving the input speech signal $x(t)$ with the corresponding Gammatone filter $g_i(t)$:

$$g_i(t) = \sum_{k=1}^{K} a_k \cdot t^{k-1} \cdot \cos(2\pi f_c t) \cdot \exp(-2\pi bt), \tag{19}$$

Here, $g_i(t)$ is the output of the $i^{th}$ Gammatone filter, $a_k$ and $b$ are parameters determining the filter shape, $f_c$ is the center frequency of the filter, and $K$ is the order of the Gammatone filter.

#### 2) SHORT-TIME FOURIER TRANSFORM (STFT)

After passing the speech signal through the Gammatone filter bank, the STFT is applied to represent the filtered signal in the time-frequency domain. The STFT magnitude spectrum is denoted as $X_i(k, \omega)$, where $k$ represents the time frame index and $\omega$ represents the frequency index. The equation for computing the STFT can be expressed as:

$$X(k, \omega) = \int_{-\infty}^{\infty} x(t) \cdot w(t - kT) \cdot e^{-j\omega t}\, dt, \tag{20}$$

Here, $T$ is the length of the analysis window. $w(t)$ is the window function used to reduce spectral leakage, typically a tapering window like the Hamming or Hann window. $k$ is the time frame index, representing the position of the analysis window in the signal. $\omega$ is the frequency index, representing the frequency component at a specific point in time, and $j$ is the imaginary unit.

#### 3) LOG AMPLITUDE

The log amplitude $L_i(k, \omega)$ of each Gammatone filter output is computed to mimic the human auditory system's logarithmic perception of loudness:

$$L_i(k, \omega) = \log(|X_i(k, \omega)|^2 + \epsilon), \tag{21}$$

where $\epsilon$ is a small constant added to avoid taking the logarithm of zero.

#### 4) DISCRETE COSINE TRANSFORM (DCT)

Finally, the DCT is applied to the log amplitude spectrum $L_i(k, \omega)$ to obtain the GFCC coefficients. The DCT reduces the dimensionality of the log amplitude spectrum while preserving the most relevant information:

$$\mathrm{GFCC}(i, n) = \sum_{k=0}^{K-1} w_n(k) \cdot L_i(k, \omega), \tag{22}$$

where GFCC$(i, n)$ is the $i^{th}$ filter's $n^{th}$ GFCC coefficient, and $w_n(k)$ represents the DCT basis function.

Using the Gammatone Frequency Cepstral Coefficients, ASR systems can effectively capture critical frequency-related information from speech signals, improving speech recognition performance, especially in noisy environments or competing background sounds.

## V. LANGUAGE MODELING

Language modeling stands as a foundational idea in the realm of NLP, holding significant importance in various tasks that involve working with language. At its core, language modeling consists of building a statistical model that aims to predict the likelihood of sequences of words or characters occurring in a given language. The main objective of language modeling is to predict the likelihood distribution of a sequence of words or characters within a sentence. By understanding the likelihood of different word sequences, language models can generate new text, predict the next word in a sentence, evaluate the grammaticality of a sentence, and even assess a paragraph's coherence.

Different methods exist for language modeling, but the ones most frequently utilized are:

### A. N-GRAM LANGUAGE MODELS

$N$-gram language models are a type of statistical language model that estimates the probability of a word or a sequence of words based on the occurrence frequencies of $N$-grams ($N$ consecutive words) in a given text corpus [101]. These models operate under the idea that the preceding $N-1$ words solely influence the likelihood of a word appearing. This concept is referred to as the Markov assumption.

The main idea behind $N$-gram language models is to estimate the conditional probability $P(w_i|w_{i-1}, w_{i-2}, \ldots, w_{i-N+1})$, which represents the likelihood of word $w_i$ given the previous N-1 words $w_{i-1}, w_{i-2}, \ldots, w_{i-N+1}$ [102].

The probability of an $N$-gram is calculated using the relative frequency of its occurrence in the training data. The formula for estimating the $N$-gram probability is given as follows:

$$P(w_i|w_{i-1}, w_{i-2}, \ldots, w_{i-N+1}) = \frac{C(w_{i-N+1}, w_{i-N+2}, \ldots, w_i)}{C(w_{i-N+1}, w_{i-N+2}, \ldots, w_{i-1})}, \tag{23}$$

Here, $C(w_{i-N+1}, w_{i-N+2}, \ldots, w_i)$ is the count of the $N$-gram sequence $(w_{i-N+1}, w_{i-N+2}, \ldots., w_i)$ in the training corpus. And $C(w_{i-N+1}, w_{i-N+2}, \ldots.., w_{i-1})$ is the count of the $(N-1)$-gram sequence $(w_{i-N+1}, w_{i-N+2}, \ldots, w_{i-1})$ in the training corpus.

To handle cases where certain $N$-grams have not been seen in the training data, smoothing techniques like add-$k$ smoothing (Laplace smoothing) or backoff techniques are commonly used. These techniques assign a small probability to unseen $N$-grams to ensure that no $N$-gram has a zero probability. The choice of $N$ in $N$-gram language models affects the trade-off between capturing local context (e.g., unigrams for basic word prediction) and considering longer-range dependencies (e.g., trigrams for capturing some phrase-level information).

$N$-gram language models have been widely used in various NLP tasks, especially in the early days of NLP, when computational resources were limited. However, with the advent of neural network-based language models, such as the Transformer-based models, $N$-grams have been surpassed

**TABLE 5.** Advantages and disadvantages of acoustic feature extraction methods for arabic ASR system.

| Feature Extraction Method | Advantages | Disadvantages |
|---|---|---|
| MFCC | • Widely used and well-established method in Arabic ASR.<br>• Effectively captures the frequency characteristics of Arabic speech.<br>• Robust against variations in speaker and environment in Arabic ASR.<br>• Suitable for tasks with limited Arabic training data. | • May not capture complex temporal variations in Arabic speech.<br>• Relies on manual tuning of parameters for optimal performance.<br>• Limited ability to model non-linear relationships in Arabic speech.<br>• Requires careful consideration of window size and frame shift for Arabic ASR. |
| PLP | • Mimics the human auditory system, enhancing robustness in Arabic ASR.<br>• Captures both spectral and temporal features of Arabic speech.<br>• Effective in handling Arabic speech in noisy environments.<br>• Less sensitive to background noise compared to some other methods in Arabic ASR. | • Computationally more complex compared to MFCC for Arabic ASR.<br>• Requires careful tuning of parameters for optimal performance.<br>• Limited availability of open-source PLP implementations for Arabic ASR.<br>• May not outperform MFCC in certain Arabic ASR tasks. |
| FBANK | • Simple and computationally efficient method for Arabic ASR.<br>• Captures the overall spectral characteristics of Arabic speech.<br>• Less sensitive to variations in speaker and environment in Arabic ASR.<br>• Suitable for Arabic ASR tasks with limited training data. | • May not capture detailed frequency information in Arabic speech.<br>• Limited ability to model non-linear relationships in Arabic speech.<br>• Prone to issues with spectral aliasing in Arabic ASR.<br>• Performance highly depends on the number of filter banks chosen for Arabic ASR. |
| GFCC | • Mimics the human auditory system, enhancing robustness in Arabic ASR.<br>• Effective in capturing fine-grained frequency information in Arabic speech.<br>• Robust against variations in speaker and environment in Arabic ASR.<br>• Suitable for Arabic ASR tasks with complex frequency patterns. | • Computationally more complex compared to MFCC and FBANK for Arabic ASR.<br>• Limited availability of open-source GFCC implementations for Arabic ASR.<br>• May not significantly outperform other methods in all Arabic ASR scenarios.<br>• Requires careful consideration of parameters for optimal performance. |

mainly in terms of performance and capability to capture long-range dependencies. Nonetheless, $N$-grams still serve as a foundational concept in language modeling and continue to find applications in specific scenarios.

### B. NEURAL NETWORK LANGUAGE MODELS (NNLM)

NNLM is a language model that uses neural networks to estimate the conditional probability distribution of a sequence of words in a sentence or document [103]. The NNLM uses the context of preceding words in a sequence to anticipate the likelihood of the next word. The general architecture of an NNLM involves embedding the phrase in a continuous vector space and using NN layers to learn the relationships between the words in the context [104]. Here, we present all the steps for a basic NNLM:

- **Word Embedding:** Each word in the vocabulary is represented by a fixed-size dense vector (embedding). Let's denote the word embeddings as $e(w)$, where $w$ represents the word.
- **Context Formation:** Given a sequence of words, the context of a specific word $w_t$ (target word) is formed by considering the preceding $n-1$ words as the context for predicting the target word. The context is represented as $C(w_t)$.
- **Neural Network Architecture:** The context $C(w_t)$ is passed through one or more neural network layers to capture the relationships between the words in the context. These layers can be fully connected (dense), recurrent neural network (RNN), convolutional, or transformer-based layers. Let's denote this neural network layer as $f(C(w_t))$ for simplicity.
- **Softmax Layer:** The output of the neural network layer is then passed through a softmax function to convert the logits (raw scores) into probabilities. The softmax function takes the form:

$$P(w_t|C(w_t)) = \frac{\exp(f(C(w_t))[w_t])}{\sum_{w_i} \exp(f(C(w_t))[w_i])} \tag{24}$$

where $P(w_t|C(w_t))$ is the probability of word $w_t$ given the context $C(w_t)$, $f(C(w_t))[w_t]$ is the score (logit) assigned to word $w_t$ by the neural network. The sum in the denominator is taken over all words in the vocabulary.

- **Loss Function:** The training of the model involves reducing the cross-entropy loss. This loss quantifies the disparity between the predicted probabilities assigned to words and the real probabilities represented by one-hot encoding for the target word. The equation for Cross-Entropy Loss:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log\left(P(w_i \mid w_{i-1}, w_{i-2}, \ldots \right. \\ \left. \ldots, w_{i-n+1}; \theta)\right) \tag{25}$$

  Here, $L(\theta)$ is the Cross-Entropy Loss. $N$ is the number of training examples in the dataset. $w_i$ represents the $i$-th word in the sequence and $P(w_i \mid w_{i-1}, w_{i-2}, \ldots, w_{i-n+1}; \theta)$ is the predicted probability of word $w_i$ given the context words $w_{i-1}, w_{i-2}, \ldots, w_{i-n+1}$, parameterized by $\theta$.
- **Training:** The model is trained using backpropagation and optimization algorithms, such as SGD or Adam, to adjust the neural network's weights and minimize the loss function.

### C. TRANSFORMER-BASED LANGUAGE MODELS

Transformer-based language models have revolutionized natural language processing tasks, including automated speech recognition [105]. The core innovation of the Transformer model lies in its self-attention mechanism, which allows it to capture long-range dependencies between words in a sequence without the need for recurrent connections. The model consists of an encoder-decoder architecture, but we typically use only the encoder part for language modeling since we do not need the decoding aspect for this task.

#### 1) SELF-ATTENTION MECHANISM

The self-attention mechanism empowers the model to assess the significance of individual words within a sequence while creating representations for each word [106]. The attention score between a word at position $i$ and a word at position $j$ are calculated using three learned matrices: Query ($Q$), Key ($K$), and Value ($V$). The attention mechanism can be represented as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{26}$$

Here, $d_k$ represents the dimension of the Key and Query matrices. $Q$ is the matrix representing queries (word representations to be attended). $K$ is the matrix representing keys (word representations used to calculate attention scores), and $V$ is the matrix representing values (word representations used in the final weighted sum).

#### 2) MULTI-HEAD ATTENTION

To capture different types of information and dependencies, the Transformer model uses multiple self-attention heads [107]. Each head has its set of learned $Q$, $K$, and $V$ matrices. The final attention output is generated by combining and transforming the results of several attention heads. This multi-head attention process can be described as follows in mathematical terms:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W_O \tag{27}$$

Here, $h$ is the number of attention heads, $\text{head}_i = \text{Attention}(QW_{Qi}, KW_{Ki}, VW_{Vi})$ represents the $i$-th attention head, and $W_O$ is the output linear transformation matrix.

#### 3) POSITIONAL ENCODING

Since Transformers don't have an inherent sense of word order like recurrent models, positional encoding injects information about word positions into the model. One common approach is to add positional encodings to the word embeddings. The positional encoding is defined as follows:

$$\text{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{28}$$

$$\text{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{29}$$

Here, $pos$ is the position of the word in the sequence, $i$ is the index of the dimension in the word embedding, and $d_{\text{model}}$ is the dimension of the word embeddings and the positional encodings.

These equations enable the Transformer-based language models to efficiently process and generate representations for language sequences, making them highly effective for a wide range of natural language processing tasks, including speech recognition.

## VI. ACOUSTIC MODELING

Acoustic modeling plays a vital role in the functionality of Arabic ASR systems. It involves building statistical models that capture the relationship between acoustic features extracted from speech signals and corresponding phonetic or subword units. The goal is to accurately map acoustic patterns to linguistic units, enabling the ASR system to transcribe speech into text effectively. In this context, we'll explore several frequently employed methods for creating acoustic models in Arabic ASR systems.

### A. HIDDEN MARKOV MODELS (HMM)

HMM is a statistical model widely used in ASR systems, including those for Arabic speech recognition [19], [108]. It is based on the Markov property, which assumes that the future state depends only on the current state and not on the sequence of states leading up to it.

#### 1) MODEL COMPONENTS

- **Hidden States ($h$):** The hidden states represent the underlying linguistic units, such as phonemes or subword units, which are not directly observable.

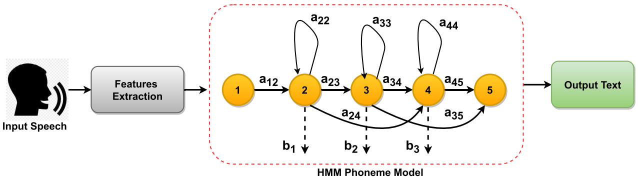


**FIGURE 6.** Hidden markov model (HMM) for Arabic ASR system.

- **Observations ($o$):** The observations are the acoustic features extracted from the speech signal, such as MFCC coefficients or filter banks, which can be observed.
- **State Transition Probabilities ($A$):** $A$ is a matrix containing the probabilities of transitioning from one hidden state to another.
- **Emission Probabilities ($B$):** $B$ is a matrix representing the probabilities of observing specific acoustic features given a hidden state.
- **Initial State Probabilities ($\pi$):** $\pi$ is a vector containing the starting probabilities from each hidden state.

### 2) MODEL EQUATION

The HMM [109] can be mathematically represented as follows:

**Initialization:**

$$\pi = [\pi_1, \pi_2, \ldots, \pi_n] \tag{30}$$

where $\pi_i$ is the initial probability of being in state $i$, and $n$ is the total number of hidden states.

**State Transition Probability Matrix:**

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \tag{31}$$

where $a_{ij}$ represents the probability of transitioning from state $i$ to state $j$.

**Emission Probability Matrix:**

$$B = \begin{bmatrix} b_1(o_1) & b_1(o_2) & \cdots & b_1(o_k) \\ b_2(o_1) & b_2(o_2) & \cdots & b_2(o_k) \\ \vdots & \vdots & \ddots & \vdots \\ b_n(o_1) & b_n(o_2) & \cdots & b_n(o_k) \end{bmatrix} \tag{32}$$

where $b_i(o_j)$ represents the probability of observing acoustic feature vector $o_j$ given the hidden state $i$.

### 3) MODEL VISUALIZATION

In Figure 6, the circles represent the hidden states $(1, 2, \ldots, n)$, and the arrows between them represent the state transition probabilities ($A$). By incorporating the probabilities of state transitions and emission probabilities, HMMs can effectively model the relationship between acoustic features and linguistic units in speech, making them a valuable tool for Arabic speech recognition and other ASR tasks.

HMM is one of the oldest and most powerful models for speech recognition tasks [110]. The paper [19] describes creating and assessing a natural Arabic ASR system that works for different speakers without needing specific training. This system employs HMMs and Sphinx tools, leading to impressive accuracy in recognizing words from comparable speakers and sentences. The paper [20] details the process of building an Arabic ASR engine using the HMM Toolkit. This engine effectively identifies both uninterrupted speech and individual words, achieving exceptional accuracy rates: 90.62% for sentence correction, 98.01% for word correction, and 97.99% for overall word accuracy. Bahi and Sellami [21] propose a system that integrates Hidden Markov Models with vector quantization for recognizing Arabic isolated words. The system transforms the word into a symbolic sequence and compares it to reference Markov models to recognize the word. The proposed system can handle the variability of the speech signal.

## B. GAUSSIAN MIXTURE MODELS (GMM)

Arabic speech recognition using Gaussian Mixture Models (GMM) involves modeling the acoustic features of speech using a combination of Gaussian distributions. The GMM is widely used in ASR systems for its ability to approximate complex probability distributions effectively [111], [112]. Each Gaussian component in the GMM represents a specific speech sound or phonetic unit.

The equation for the GMM [113] can be represented as follows:

Let's assume we have a set of acoustic feature vectors for a given speech signal:

$$X = x_1, x_2, \ldots, x_i, \ldots, x_n \tag{33}$$

Here, '$n$' is the total number of frames, and each '$x_i$' is a D-dimensional acoustic feature vector.

The GMM is represented as a weighted sum of $K$ Gaussian components:

$$P(X|\theta) = \sum_i w_i \cdot N(x_i|\mu_i, \Sigma_i) \tag{34}$$

Here, $P(X|\theta)$ is the likelihood of the observed acoustic feature vectors $X$ given the model parameters $\theta$. $w_i$ is the weight of the $i$-th Gaussian component, representing the probability of choosing that component. It satisfies $\sum_i w_i = 1$. And $N(x_i|\mu_i, \Sigma_i)$ is the multivariate Gaussian distribution representing the $i$-th Gaussian component, with mean vector $\mu_i$ and covariance matrix $\Sigma_i$.

The mathematical expression describing the probability density function (PDF) of the multivariate Gaussian distribution can be formulated as follows:

$$N(x_i|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{D}{2}} \cdot |\Sigma_i|^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2}(x_i - \mu_i)^T \Sigma_i^{-1}(x_i - \mu_i)\right) \tag{35}$$

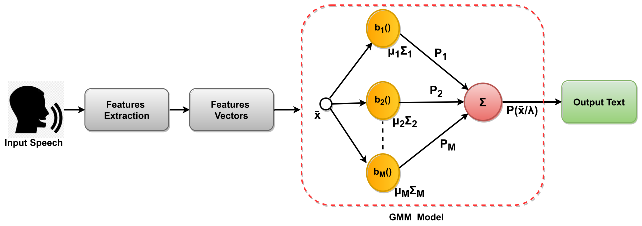


**FIGURE 7.** Gaussian mixture models (GMM) for automated arabic speech recognition (ASR).

Here, $D$ is the dimensionality of the acoustic feature vectors. $|\Sigma_i|$ is the determinant of the covariance matrix $\Sigma_i$. $(x_i - \mu_i)^T$ represents the transpose of the difference between the feature vector $x_i$ and the mean vector $\mu_i$ and $\Sigma_i^{-1}$ is the inverse of the covariance matrix $\Sigma_i$.

The model parameters (weights, means, and covariance matrices) are calculated using the Expectation-Maximization (EM) algorithm during training. Once the GMM is trained, it can be used in the ASR system to calculate the likelihood of acoustic feature vectors given the model, and this likelihood is used for speech recognition and decoding. Figure 7 represents a basic GMM for the Arabic ASR system.

The GMM finds extensive application within the realm of speech recognition [114]. The paper [22] presents a method for recognizing spoken Arabic digits using a Gaussian mixture model (GMM) classifier and Delta-Delta Mel-frequency cepstral coefficients (DDMFCC) for feature extraction. The experimental results show a 99.31% correct digit recognition rate, which is better than previous work on spoken Arabic digit speech recognition. Huang and Hasegawa-Johnson [115] propose a cross-dialectal GMM training scheme for Arabic ASR, which transfers knowledge between Modern Standard Arabic (MSA) and regional dialects and between different dialects to improve phone classification tasks. The results of the experiments demonstrate that training GMM models across different dialects brings notable benefits, particularly when a small quantity of MSA data is moved.

### *C. CONVOLUTIONAL NEURAL NETWORKS (CNN)*

CNNs are widely used in Arabic speech recognition to capture local patterns and features in speech data [116], [117]. The paper [118] focuses on Arabic ASR using MFSC and GFCC with their first and second-order derivatives. Using CNN facilitates feature learning and classification, enhancing Arabic ASR performance. The highest achieved accuracy when employing CNN in conjunction with GFCC is 99.77%. Amari et al. [119] presents a new model for Arabic ASR using deep CNNs. The proposed model is tested on the Arabic Isolated Words Corpus (ASD) database and achieves promising results. The study compares two models based on CNN and LSTM, and the deep CNN model performs better. The mathematical representation of a CNN [120] can be summarized as follows:

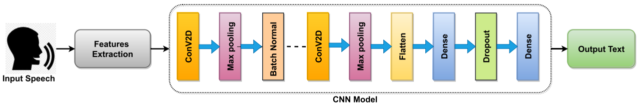


**FIGURE 8.** CNNs architecture for Arabic ASR system.

Let's consider an input speech signal represented as a time-domain waveform or a spectrogram, denoted as $x$. A CNN's structure includes various types of layers, such as convolutional, pooling, and fully connected layers.

#### 1) CONVOLUTIONAL LAYER

In the convolutional layer, we use a set of learnable filters (kernels) to convolve over the input speech signal $x$. This operation is mathematically represented as:

$$z_i = (x * w_i) + b_i \tag{36}$$

Here, $z_i$ is the output feature map for the $i$-th filter. $x$ is the input speech signal. $w_i$ is the learnable weight (filter) for the $i$-th filter. $*$ denotes the convolution operation and $b_i$ is the bias term for the $i$-th filter.

The convolution operation essentially slides the filter over the input signal, and element-wise multiplies the filter values with the corresponding input signal values. It sums them up to produce the output feature map $z_i$.

#### 2) ACTIVATION FUNCTION

After the convolution operation, an activation function (commonly ReLU) is applied to each element of the resulting feature map to introduce non-linearity:

$$a_i = \mathrm{ReLU}(z_i) \tag{37}$$

Here, $a_i$ is the output after applying the activation function to the $i$-th feature map.

#### 3) POOLING LAYER

We downsample the feature maps in the pooling layer to reduce the spatial dimensions and computational complexity. An often-used pooling technique is max-pooling, where the highest value within each local area is chosen.

$$p_i = \mathrm{max\_pool}(a_i) \tag{38}$$

Here, $p_i$ is the downsampled feature map (pooled output) for the $i$-th feature map.

#### 4) FULLY CONNECTED LAYER

Following multiple convolutional and pooling layers, the resulting feature maps undergo flattening and are then inputted into fully connected layers for prediction generation. The fully connected layers use weights and biases to map the extracted features to the final output classes (phonemes, words, or sentences). Figure 8 represent a basic architecture of CNNs for the Arabic ASR system.

### D. RECURRENT NEURAL NETWORKS (RNN)

RNNs represent a neural network design frequently employed for handling sequential information, like speech signals [121], [122]. They prove highly suitable for tasks like speech recognition as they excel in retaining concealed states that encapsulate time-related patterns present in the data [123]. In the context of Arabic ASR, an RNN can be utilized to convert the input speech signal into a sequence of phonemes or characters, which can then be further processed to recognize the spoken words [24], [38], [124]. The mathematical equation for the basic operation of a simple RNN can be defined as follows:

$$h_t = \sigma(W_{hh} \cdot h_{t-1} + W_{hx} \cdot x_t + b_h) \tag{39}$$
$$y_t = \sigma(W_y \cdot h_t + b_y) \tag{40}$$

Here, $h_t$ is the hidden state at time step $t$, representing the network's memory of past information. $x_t$ is the input at time step $t$, which can be a vector representation of the speech signal or its extracted features. $W_{hh}$ and $W_{hx}$ are weight matrices for the recurrent and input connections. $b_h$ is the bias vector for the hidden state. $W_y$ is the weight matrix for the output connection. $b_y$ is the bias vector for the output, and $\sigma$ denotes the activation function, such as the sigmoid function or hyperbolic tangent (tanh), used to introduce non-linearity.

The RNN processes the input speech signal frame by frame ($x_t$) and updates its hidden state ($h_t$) at each time step. The final output ($y_t$) can be used for various purposes, like phoneme or word recognition. However, traditional RNNs suffer from vanishing gradient problems when dealing with long sequences. To address this issue, variants like LSTM [76] and GRU [125] have been introduced to improve the capability of modeling long-term dependencies in speech data. These variants modify the basic RNN equation to incorporate gating mechanisms that regulate the flow of information and gradients, making them more effective for speech recognition tasks. Figure 3 gives a very detailed architecture of RNN for speech recognition.

### E. TRANSFORMER-BASED ACOUSTIC MODELS

Arabic speech recognition using Transformer-based Acoustic Models involves employing the Transformer architecture for acoustic modeling tasks [126], [127]. The Transformer model, initially proposed for NLP tasks, has also shown promising results in speech recognition [128]. It leverages self-attention mechanisms to capture long-range dependencies in the input sequence and has achieved state-of-the-art performance in various tasks. The architecture of the Transformer-based Acoustic Model can be represented as follows:

#### 1) INPUT SEQUENCE

Let's assume we have an input audio sequence represented as a sequence of acoustic feature vectors:

$$X = \{x_1, x_2, x_3, \ldots, x_T\} \tag{41}$$

#### 2) ACOUSTIC ENCODER

The acoustic encoder in the Transformer-based Acoustic Model processes the input sequence and transforms it into a sequence of high-level representations. This is achieved through self-attention layers and feed-forward neural networks.

#### 3) SELF-ATTENTION MECHANISM

The self-attention mechanism enables the model to assess the significance of each input element (acoustic feature vector) concerning all other elements in the sequence [106]. It computes a weighted sum of the input sequence, considering the relationships between different elements. The equation of the Self-Attention Mechanism is given in eq.26.

#### 4) MULTI-HEAD ATTENTION

The Transformer-based Acoustic Model usually employs multiple self-attention heads to capture different dependencies in the input sequence [107]. A detailed equation of Multi-Head Attention is given in eq. 27.

#### 5) FEED-FORWARD NEURAL NETWORKS (FFNN)

After the self-attention layers, the output is passed through a feed-forward neural network [129], which applies non-linear transformations to the representations:

$$\text{FFNN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{42}$$

Here, ReLU is an activation function. $W_1$, $b_1$, $W_2$, and $b_2$ are learnable parameters.

#### 6) LAYER NORMALIZATION AND RESIDUAL CONNECTIONS

Layer normalization and residual connections stabilize training and facilitate the flow of gradients during backpropagation.

#### 7) OUTPUT LAYER

The output layer of the Transformer-based Acoustic Model is typically a linear layer followed by a softmax activation, used for predicting the probabilities of different output units (phonemes, characters, or subword units) at each time step. It's important to note that the specifics of the Transformer-based Acoustic Model may vary depending on the implementation and the specific ASR task. During training, the model is trained to minimize a suitable loss function (e.g., cross-entropy loss) to learn the appropriate acoustic representations for accurate speech recognition.

Acoustic models are the backbone of speech recognition systems, shaping their performance and capabilities. These models offer distinct advantages that contribute to accurate and robust speech recognition. From Hidden Markov Models' adaptability to DNNs' capacity for automatic feature learning, each model type brings its strengths to the table, catering to a wide range of recognition needs. Table 6 presents the advantages and disadvantages of all the acoustic models discussed.

**TABLE 6.** Advantages and disadvantages of acoustic models for arabic ASR system.

| Acoustic Model | Advantages | Disadvantages |
|---|---|---|
| Hidden Markov Models (HMM) | • Well-established and understood modeling framework.<br>• Can be adapted to different Arabic speakers and environments.<br>• Can handle various Arabic acoustic features.<br>• Suitable for Arabic ASR tasks with limited training data. | • Relies heavily on manual feature engineering.<br>• Might struggle with capturing complex Arabic temporal patterns.<br>• Prone to overfitting, particularly for complex Arabic ASR tasks.<br>• Not as effective for modeling long-range dependencies in Arabic speech. |
| Gaussian Mixture Models (GMM) | • Simplicity in implementation and training for Arabic ASR.<br>• Suitable for modeling multi-modal distributions in Arabic speech.<br>• Can be effective for certain Arabic speech-related tasks.<br>• Performs well for some Arabic phoneme-level recognition. | • Struggles to capture intricate Arabic patterns in data.<br>• Limited ability to model long-range dependencies in Arabic speech.<br>• May require careful fine-tuning for optimal performance in Arabic ASR.<br>• Not as competitive compared to more modern approaches for Arabic ASR. |
| Convolutional Neural Networks (CNN) | • Effective at capturing local patterns and spatial information in Arabic speech.<br>• Suitable for extracting features from Arabic spectrogram-like data.<br>• Robust to variations in Arabic speaker and environment.<br>• Can leverage pre-trained CNNs for transfer learning in Arabic ASR. | • May struggle to capture long-range dependencies in Arabic speech.<br>• Requires careful tuning of hyperparameters for Arabic ASR.<br>• Can be computationally demanding, particularly with deeper architectures in Arabic ASR.<br>• The performance level relies significantly on the excellence and amount of Arabic training data available. |
| Recurrent Neural Networks (RNN) | • Proficient at modeling sequential dependencies in Arabic speech data.<br>• Capable of handling variable-length sequences in Arabic ASR.<br>• Effective for Arabic phoneme-level recognition and speech synthesis.<br>• Can incorporate historical context for better predictions in Arabic ASR. | • Prone to vanishing/exploding gradient problems during training in Arabic ASR.<br>• The training process can be slow due to its sequential nature in Arabic ASR.<br>• Limited in capturing very long-range dependencies in Arabic speech.<br>• May require specialized architectures (e.g., LSTM, GRU) to address RNN issues for Arabic ASR. |
| Transformer-based Models | • Exceptional at capturing long-range dependencies and context in Arabic speech.<br>• Highly parallelizable training, reducing training time for Arabic ASR.<br>• State-of-the-art performance on various NLP tasks for Arabic ASR.<br>• Adapt well to various Arabic data domains with minimal modifications. | • Can require substantial computational resources, especially for larger models in Arabic ASR.<br>• Complexity might hinder the interpretability of the model in Arabic ASR.<br>• Requires careful hyperparameter tuning for Arabic ASR.<br>• Limited by the availability of Arabic training data for optimal performance. |

The fundamental Bayesian equation for finding the optimal word sequence in an Arabic ASR system involves the combination of the feature extractor, acoustic model, and language model. The equation is often expressed using Bayes' theorem [130]:

$$P(W|X) = \frac{P(X|W) \cdot P(W)}{P(X)} \tag{43}$$

where:

- $P(W|X)$ is the posterior probability of the word sequence $W$ given the observed acoustic features $X$.
- $P(X|W)$ is the likelihood, representing the probability of observing the acoustic features $X$ given the word sequence $W$ (modeled by the acoustic model).
- $P(W)$ is the prior probability of the word sequence $W$, typically based on the language model.
- $P(X)$ is the probability of the observed acoustic features $X$ and acts as a normalization factor.

The goal is to find the word sequence $W$ that maximizes the posterior probability $P(W|X)$, which essentially identifies the most likely sequence of words given the observed acoustic features. This Bayesian framework helps integrate information from both the acoustic and language models to enhance the ASR system's accuracy. Figure 9 provides a pictorial description of finding the optimal word sequence using a feature extractor, acoustic, and language model.

## VII. DECODING AND RECOGNITION

This focuses on converting acoustic input, the spoken speech signal in Arabic, into a meaningful and accurate textual representation. This section addresses the complex task of deciphering spoken words and transcribing them

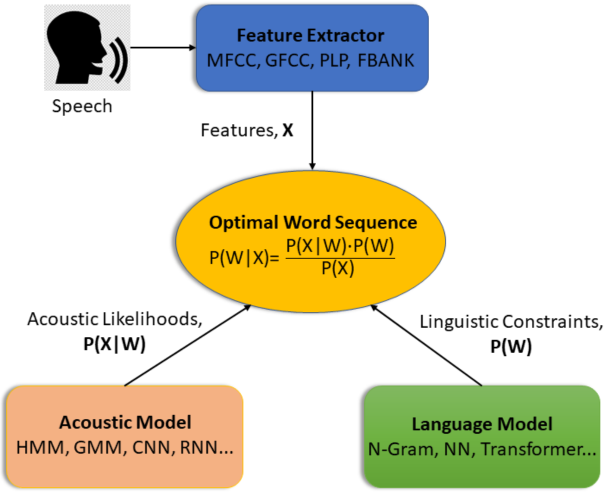


**FIGURE 9.** Arabic ASR system to find optimal word sequence using a feature extractor, acoustic model, and language model.

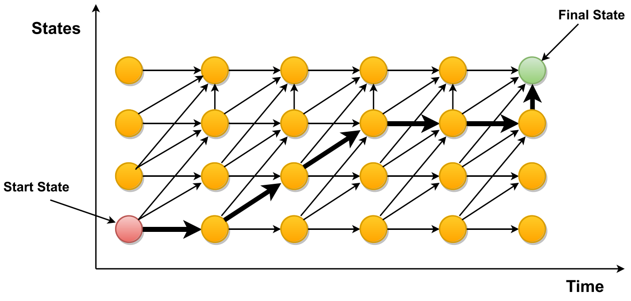


**FIGURE 10.** This figure illustrates the procedure of the Viterbi search.

into written text. Here's a breakdown of what this section entails:

### A. VITERBI ALGORITHM

The Viterbi Algorithm is a dynamic programming algorithm that plays a crucial role in various fields, including speech recognition, digital communications, and bioinformatics [131]. Named after its inventor Andrew Viterbi, this algorithm is fundamental in Hidden Markov Models (HMMs), where it's used for sequence estimation or decoding. The Viterbi search is like following a path through a map of connected HMM states. At each step and for each moment, it keeps track of the best score for the path so far. This method goes step by step through time, dealing with all the states at each time before moving to the next time [8]. Imagine it moving through a chart: one side shows the different states, and the other shows the progression of time. The basic procedure of the algorithm is made clearer by looking at Figure 10.

Even when dealing with a medium-sized set of words, conducting a complete search is impractical. The Viterbi beam search is a widely used and straightforward method to expedite the search process. However, solely relying on a beam is not always enough, leading to two commonly adopted alternatives for addressing this search challenge.

- Utilize basic acoustic and language models to create a list of several options or a structure called a lattice or N-best list. Then, employ more complex acoustic and/or language models to reevaluate and refine the smaller options to identify the most accurate sequence of words. This strategy is known as the multi-pass method.
- Create a complete set of options to explore, then simplify it using specific techniques to make it more manageable. After that, a specialized search method will be applied to find the most suitable sequence of words. This streamlined process is known as the single-pass approach.

### B. BEAM SEARCH DECODING

In Arabic ASR, the Beam Search decoding algorithm is a crucial technique for efficiently identifying the most likely sequence of words based on the provided audio features [132]. Beam Search enhances the decoding process by maintaining a limited number of possible hypotheses, referred to as the "beam width." This significantly reduces computational complexity while ensuring accurate transcriptions. The Beam Search algorithm can be described with the following equation:

At each time step $t$, the Beam Search algorithm maintains a set of $K$ hypotheses (word sequences) denoted as $H_t$, where $K$ is the beam width.

$$H_t = \{h_1, h_2, \ldots, h_K\} \tag{44}$$

Each hypothesis $h_k$ consists of a sequence of words up to time step $t$: $h_k = [w_1, w_2, \ldots, w_t]$.

The Beam Search algorithm operates as follows:

- **Initialization:** At time step $t = 1$, the algorithm starts with $K$ initial hypotheses, each consisting of a single word:

$$H_1 = \{[w_1^1], [w_1^2], \ldots, [w_1^K]\} \tag{45}$$

- **Expansion:** For each hypothesis $h_k$ in $H_t$, the algorithm generates $K$ new hypotheses by considering all possible next words based on acoustic and language model scores. The hypotheses with the highest combined scores are retained:

$$H_{t+1} = \text{Top-K}([h_k, w_{t+1}] \text{ for all } h_k \text{ in } H_t) \tag{46}$$

- **Pruning:** After generating the new hypotheses, the set is pruned to retain only the top-$K$ hypotheses with the highest combined scores.
- **Repeat:** Steps 2 and 3 are repeated for each subsequent time step until the entire input audio sequence is processed.
- **Final Selection:** Once the decoding process is completed, the hypothesis with the highest cumulative score among all time steps is selected as the final transcription.

In this algorithm, the combined score of a hypothesis $h_k$ at time step $t$ is calculated as the sum of its acoustic score (reflecting the fit between the acoustic features and the predicted phonemes) and its language model score (reflecting the linguistic likelihood of the word sequence up to time step $t$).

The Beam Search algorithm balances exploration and exploitation by retaining a limited number of hypotheses with the highest scores, effectively navigating the search space of possible word sequences. This strategy maintains a balance between precision and computational speed, rendering it appropriate for tasks that require real-time processing, such as speech recognition.

### C. WEIGHTED FINITE STATE TRANSDUCERS (WFST)

The WFST decoding algorithm for Arabic speech recognition involves utilizing a WFST to represent the relationship between input speech features and output word sequences [133]. This can be achieved through a composition process with a language model and an acoustic model. The final decoding is based on finding the best path through the WFST.

Here we provide a simplified representation of the decoding algorithm in equation form:

1) **Acoustic Model Score:**

$$\text{AcousticModelScore}(\mathbf{X}, \mathbf{O}) = \sum_{t=1}^{T} \log P(o_t|x_t) \quad (47)$$

where $\mathbf{X}$ represents the input feature sequence, $\mathbf{O}$ represents the output word sequence, $x_t$ is the acoustic feature at time $t$, and $o_t$ is the corresponding output symbol. $P(o_t|x_t)$ is the acoustic model probability.

2) **Language Model Score:**

$$\text{LanguageModelScore}(\mathbf{O}) = \log P(\mathbf{O}) \quad (48)$$

where $\mathbf{O}$ is the output word sequence. $P(\mathbf{O})$ is the language model probability.

**WFST Composition:**

$$\text{WFST} = \text{Compose}(\text{AcousticModel}, \text{LanguageModel}) \quad (49)$$

This represents the composition of the acoustic and language models into a WFST.

3) **Decoding:**

$$\text{BestPath} = \text{ShortestPath}(\text{WFST}, \text{AcousticModelScore} + \text{LanguageModelScore}) \quad (50)$$

The BestPath is determined by finding the shortest path through the composed WFST based on the combined scores of the acoustic and language models.

## VIII. EVALUATION METRICS

To evaluate the effectiveness and accuracy of the Arabic ASR system, various evaluation matrices are utilized [134]. These matrices help to measure the system's effectiveness in converting spoken Arabic utterances into text. The following are some of the main evaluation metrics that are used to assess the ASR system:

### A. WORD ERROR RATE (WER)

The WER is a widely adopted measure that quantifies the disparity between the transcribed output and the reference, representing the true intended text [135]. It's computed by tallying the insertions, deletions, and substitutions needed to convert the transcribed text into the reference text [136]. WER is defined as:

$$WER = \frac{S + D + I}{N} \quad (51)$$

where:

- $S$ = Number of substitution errors (words that were incorrectly recognized),
- $D$ = Number of deletion errors (words that were missed in recognition),
- $I$ = Number of insertion errors (extra words that were incorrectly added during recognition),
- $N$ = Total number of words present in the reference transcript.

### B. CHARACTER ERROR RATE (CER)

The CER quantifies the difference between the identified output and the reference transcript at the character level. It is beneficial when assessing the accuracy of the ASR system in languages with complex scripts, such as Arabic [137]. CER is calculated as follows:

$$CER = \frac{S_c + D_c + I_c}{N_c} \quad (52)$$

where:

- $S_c$ = Number of substitution errors (characters that were incorrectly recognized),
- $D_c$ = Number of deletion errors (characters that were missed in recognition),
- $I_c$ = Number of insertion errors (extra characters that were incorrectly added during recognition),
- $N_c$ = Total number of characters in the reference transcription.

### C. SENTENCE ERROR RATE (SER)

The SER evaluates the ASR system's ability to transcribe entire sentences correctly. It measures the percentage of sentences that are inaccurately recognized [138]. SER is defined as:

$$SER = \frac{N_{\text{incorrect sentences}}}{N_{\text{total sentences}}} \times 100 \quad (53)$$

where:

- $N_{\text{incorrect sentences}}$ = Number of sentences with recognition errors,
- $N_{\text{total sentences}}$ = Total number of sentences in the evaluation dataset.

These evaluation metrics provide a comprehensive analysis of the ASR system's performance, allowing for a thorough

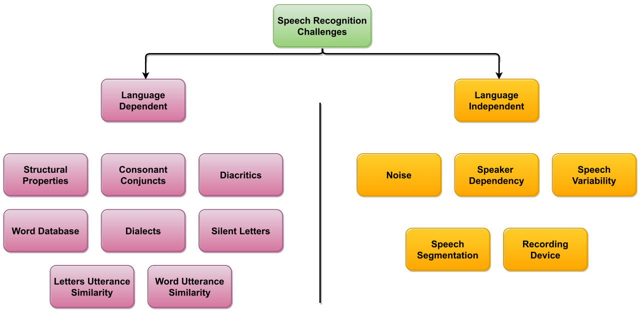


**FIGURE 11.** Some of the general difficulties faced by speech recognition systems.

assessment of its strengths and areas for improvement. The lower the values of WER, CER, and SER, the better the accuracy of the ASR system.

## IX. CHALLENGES ON ARABIC ASR

Speech recognition faces various challenges, which can be categorized into two main sections: a) challenges that are specific to the language being spoken (language-dependent), and b) challenges that are common across languages (language-independent). It is crucial for the design of speech recognition systems to consider these obstacles, and overcoming them would lead to improved performance for an Arabic ASR system. Language-independent challenges in speech processing include dealing with various factors such as background noise, dependence on individual speakers, variations in speech patterns, segmenting speech accurately, and accounting for differences in recording devices used. On the other hand, language-dependent challenges encompass specific linguistic features like structural properties, complex consonant conjuncts, diacritics, the availability of comprehensive word databases, diverse dialects, the presence of silent letters, and the similarity between word and letter utterances. Figure 11 visually represents how the obstacles depend on each other.

Addressing the language-dependent challenge of Arabic ASR will require linguistic expertise in morphological analysis, phonetic analysis, and morphophonemic interactions of Arabic. Here, we briefly describe each linguistic expertise one by one.

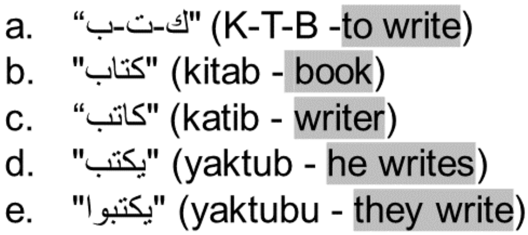
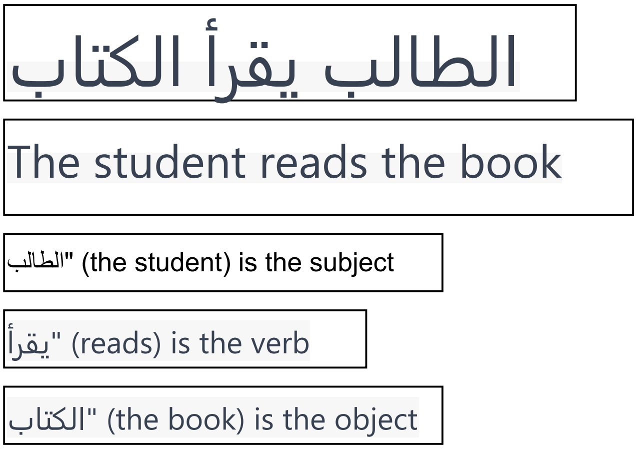

a. “ك-ت-ب” (K-T-B -to write)
b. "كتاب" (kitab - book)
c. “كاتب” (katib - writer)
d. "يكتب" (yaktub - he writes)
e. "يكتبوا" (yaktubu - they write)

**FIGURE 12.** Some Arabic words are provided here to illustrate the language-dependent challenges in Arabic ASR.

- **Morphological Analysis:** Morphological analysis is a linguistic process that involves the study of the structure and formation of words, focusing on how morphemes, the smallest units of meaning, combine to create complex word forms. In Arabic, a language renowned for its rich morphological system, morphological analysis is particularly crucial. Arabic words typically share a root consisting of consonants, and morphological analysis in Arabic involves identifying these roots and understanding how they change the addition of prefixes, suffixes, and vowels. For example, consider the root of Figure 12(a) word (K-T-B), meaning “to write.” The morphological variations include Figure 12(b) (kitab - book), 12(c) (katib - writer), and 12(d) (yaktub - he writes), illustrating how the root adapts to convey different meanings and grammatical forms. Morphological analysis is fundamental for creating accurate lexicons and understanding the derivational and inflectional processes that shape Arabic words. Linguists leverage this analysis to ensure precision in natural language processing tasks, including Automatic Speech Recognition (ASR), where a profound grasp of morphology aids in accurately transcribing spoken Arabic.
- **phonetic analysis:** Phonetic analysis is a linguistic discipline focused on the study and representation of speech sounds within a language. In ASR, phonetic analysis is crucial in accurately transcribing spoken words. This involves breaking down speech into discrete phonetic units, such as consonants and vowels, and understanding their articulatory and acoustic characteristics. For example, in English, the word “cat” can be phonetically analyzed as /kæt/, where /k/ represents the voiceless velar plosive sound, /æ/ denotes the short vowel in “cat,” and /t/ indicates the voiceless alveolar plosive. Phonetic analysis involves identifying and categorizing these sounds based on their distinctive features, such as place and manner of articulation. In more complex languages like Arabic, phonetic analysis extends to diverse consonants, vowels, and nuances. For instance, the Arabic word in Figure 12(b) (kitab - book) can be phonetically analyzed as /kıtæb/, where /k/ is the voiceless velar plosive, /I/ represents the short vowel, and /t/ and /b/ signify voiceless alveolar plosive and voiced bilabial plosive, respectively. The phonetic analysis is fundamental for developing accurate ASR systems, aiding in creating phonetic transcriptions that bridge spoken language and machine-understandable representations.
- **Morpho-phonemic Interactions:** Morpho-phonemic interactions in Arabic involve the intricate relationship between morphology and phonetics. This linguistic concept explores how morphological changes in words impact their pronunciation. For instance, root consonants undergo morphological modifications in the Arabic verb system to indicate tense, person, and number. The verb in Figure 12(d) (yaktub - he writes)



الطالب يقرأ الكتاب

The student reads the book

الطالب" (the student) is the subject

يقرأ" (reads) is the verb

الكتاب" (the book) is the object

**FIGURE 13.** A simple Arabic sentence.

shares the root Figure 12(a) (K-T-B) with variations like Figure 12(e) (yaktubu - they write). Understanding these morpho-phonemic interactions is crucial for accurate representation in ASR systems, ensuring that the system recognizes spoken words with consideration for both their morphological and phonetic dimensions.

- **Diacritization:** Diacritization in Arabic involves adding diacritical marks to written text to indicate short vowels, enhancing phonetic clarity. Discretization interactions are crucial in distinguishing words with similar consonantal roots but different vowel patterns. For instance, differentiating between Figure 12(b) (kitab - book) and 12(d) (katab - he writes) relies on diacritics to signify the vowels. Discretization is pivotal in linguistic analysis, aiding manual transcription and automatic processes like Automatic Speech Recognition (ASR). Its accurate application ensures proper pronunciation and understanding of Arabic words, particularly when written vowels are absent or ambiguous.

The sentence shown in Figure 13, has several language-dependent challenges for Arabic speech recognition:

- Phonetic Variation: Different dialects might pronounce the Arabic word "yaqra" (reads) differently, and a speech recognition system needs to handle these variations to transcribe the spoken words accurately.
- Diglossia: This sentence is in Modern Standard Arabic (MSA), but in spoken language, the pronunciation might differ significantly based on regional dialects. Recognizing both MSA and dialectal variations is a challenge.
- Vowel System: Arabic vowels, like "y" (pronounced as 'ee' or 'i' sound) in "yaqra," can be challenging to distinguish, especially when they're pronounced less distinctly in casual speech.
- Lack of Punctuation and Capitalization: The Arabic script doesn't use spaces to separate words, and there is no capitalization, making it difficult to segment the sentence into individual words.

The researchers thoroughly understand the obstacles faced in language-independent speech recognition and have developed advanced techniques to overcome these challenges effectively. Moreover, they have made considerable strides in illustrating the language-independent issues that arise in both speech recognition methods [139], [140] and feature extraction processes [141], [142]. In the subsequent sections, we will highlight the language-dependent challenges specific to an Arabic speech recognition system and propose potential solutions. However, before delving into that, we will concisely explain the language-independent challenges, which can be found in Table 7.

## X. FUTURE RESEARCH DIRECTION ON ARABIC ASR

In this segment, we outline the primary obstacles an Arabic ASR system faces to enhance the effectiveness of current approaches. Additionally, we put forward an architectural solution to address these challenges. After an extensive examination of Section IX, three crucial language-specific difficulties have been identified:

- Words can be filtered out based on their grammatical relationship with previous words. Additionally, understanding the literal dependencies between similar words with similar patterns can lead to selecting the correct word more easily. This reduces the search space for the right word. However, successfully extracting both grammatical and literal dependencies requires a robust memory-based architecture.
- Grammatical and preceding character dependencies involve understanding the correct combinations of vowel diacritics, consonant diacritics, and graphemes within a word. All languages possess specific grammatical patterns that enable accurate prediction of the appropriate graphemes from a given set. Revealing these patterns necessitates the assistance of a generator that operates based on memory, facilitating the extraction process.

Current studies on Arabic ASR systems overlook the connection between grammatical complexities and accurate word predictions. As a result, the mentioned issues present promising avenues for future research in Arabic ASR. Moreover, we aim to enhance the future scope of Arabic ASR by introducing a novel theoretical architecture. We present in Figure 14 a carefully devised architecture that we consider the most optimal after extensive research efforts. To our knowledge, the proposed framework remains untapped and unexplored in any preceding research endeavors. Furthermore, the suggested framework integrates an innovative blend of recurrent and hybrid components, introducing a new vantage point to the current research landscape. Thus, we subsequently outline the key attributes of the proposed ASR system.

- Grammatical dependencies between words play a crucial role in identifying the most appropriate literary expressions by applying specific rules. These rules can be reinforced and interconnected through short-term memory. The system can efficiently grasp grammatical

**TABLE 7.** An overview of obstacles not dependent on any particular language.

| Challenge | Description |
|---|---|
| Noise | Speech are often accompanied by environmental sounds, and the presence of noise can interfere with speech recognition by altering its features and leading to inaccurate word outputs. Hence, the crucial preprocessing step involves reducing or eliminating noise to improve overall performance. |
| Speaker reliance | Speaker Dependency in an ASR (Automatic Speech Recognition) system refers to the system's ability to recognize and target specific speakers. When an ASR system is customized for a specific person, it's termed a speaker-dependent ASR system. Otherwise, it belongs to the group of speaker-independent ASR systems. Nowadays, most ASR systems are designed to be speaker-independent. |
| Speech Variability | Speech variability refers to the variations in utterances influenced by factors such as human emotions, environmental conditions, and age. Developing appropriate ASR (Automatic Speech Recognition) architectures trained on diverse speech datasets can effectively address this issue. |
| Recording gadget | The selection of recording equipment determines the audio format used by the ASR system. The incoming audio might range from single-channel (mono) to dual-channel, stereo, or intensity stereo. Each input format possesses its own benefits and drawbacks, presenting distinct hurdles depending on the context. |
| Speech Segmentation | Speech segmentation can be categorized into two key categories: word segmentation and phoneme segmentation. These divisions play a vital role in achieving precise continuous speech recognition. Segmentation errors can result in misalignments with speech patterns, causing misunderstandings. However, it's important to highlight that some contemporary end-to-end ASR systems have moved beyond the need for traditional speech segmentation techniques. |

relationships by leveraging this short-term memory, especially when exposed to extensive speech data during training.

- By utilizing a blend of short-term memory and a speech character generator, it becomes possible to ascertain characters' grammatical structure and preceding dependency. Prominent systems in this domain rely on short-term memory to investigate character-level prediction dependency [75].
- All languages, including Arabic, have words with irregular letter patterns. To address this issue, one can memorize specific fixed words. Consequently, phoneme-to-word or speech-to-word matching would be the most effective approach. However, current architectures primarily employ end-to-end schemes, generating characters and relying solely on information from preceding characters [143], [144]. As a result, they often neglect the nuances of irregular word structures.
- At present, the existing implementations primarily focus on character recognition methods [75]. However, we put forth a hybrid strategy that merges word and character-matching techniques to tackle the issue of generating words that are irregular or not found in the dictionary. Through this approach, our recommended system adeptly seeks optimal word matches. In cases where an exact word match is not found, the model can intelligently extract characters from the speech to provide meaningful output.

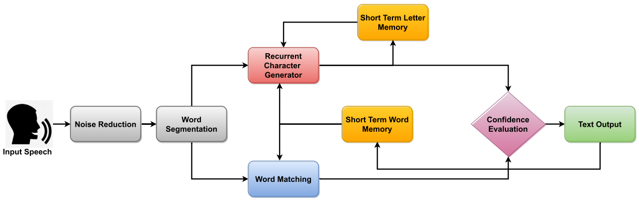


**FIGURE 14.** An ideal Arabic ASR system can be designed with a suggested architecture incorporating short-term memory. The system can comprehend words, characters, and their intricate linguistic associations by integrating these memory elements. Techniques for word matching can be harnessed to detect words harboring diverse or unpronounced letters. Furthermore, a confidence assessment procedure can ascertain the model's certainty regarding the presence of a spoken word in its existing speech-to-word lexicon. In instances of uncertainty, the model can derive recurring characters from the speech, enhancing its recognition prowess.

Suppose the proposed architectural pattern is trained using speech corpora that exhibit appropriate variations in speech and grammar. In that case, it can potentially address the overall challenges outlined in the paper effectively.

## XI. CONCLUSION

This survey initiates by exploring the ongoing research efforts in the field of Arabic Automatic Speech Recognition (ASR) systems, encompassing speech databases and recognition techniques. Subsequently, various challenges existing within the domain of Arabic ASR are thoroughly investigated. We have discussed the differences in structure and linguistic aspects among languages that researchers working on ASR (Automatic Speech Recognition) systems should focus on. We have thoroughly explored the foundational grammatical principles and proposed potential solutions for addressing these challenges. While some difficulties are shared across various languages, we have specifically emphasized the unique challenges and opportunities that arise when dealing with Arabic. We have thoroughly examined the latest implementations of Arabic ASR systems and discovered they are imperfect. Our extensive investigation led us to believe that our refined research could enhance Arabic-specific and universal ASR systems. By doing so, we hope to provide valuable guidance to researchers, helping them address the precise challenges that must be overcome.

## REFERENCES

[1] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[2] I. Hamed, P. Denisov, C.-Y. Li, M. Elmahdy, S. Abdennadher, and N. T. Vu, "Investigations on speech recognition systems for low-resource dialectal Arabic–English code-switching speech," *Comput. Speech Lang.*, vol. 72, Mar. 2022, Art. no. 101278.

[3] H. Qasim and H. A. Abdulbaqi, "Arabic speech recognition using deep learning methods: Literature review," in *Proc. AIP Conf.*, vol. 2398, 2022, Art. no. 050029.

[4] F. Al-Anzi and D. AbuZeina, "Literature survey of Arabic speech recognition," in *Proc. Int. Conf. Comput. Sci. Eng. (ICCSE)*, Mar. 2018, pp. 1–6.

[5] W. Algihab, N. Alawwad, A. Aldawish, and S. AlHumoud, "Arabic speech recognition with deep learning: A review," in *Social Computing and Social Media. Design, Human Behavior and Analytic*, Orlando, FL, USA. Germany: Springer, 2019, pp. 15–31.

[6] A. Dhouib, A. Othman, O. El Ghoul, M. K. Khribi, and A. Al Sinani, "Arabic automatic speech recognition: A systematic literature review," *Appl. Sci.*, vol. 12, no. 17, p. 8898, Sep. 2022.

[7] A. A. Abdelhamid, H. A. Alsayadi, I. Hegazy, and Z. T. Fayed, "End-to-end Arabic speech recognition: A review," in *Proc. 19th Conf. Lang. Eng.*, Alexandria, Egypt, 2020, pp. 26–30.

[8] S. M. Abdou and A. M. Moussa, "Arabic speech recognition: Challenges and state of the art," in *Computational Linguistics, Speech and Image Processing for Arabic Language*, 2019, pp. 1–27.

[9] H. A. Alsayadi, A. A. Abdelhamid, I. Hegazy, B. Alotaibi, and Z. T. Fayed, "Deep investigation of the recent advances in dialectal Arabic speech recognition," *IEEE Access*, vol. 10, pp. 57063–57079, 2022.

[10] M. Elghonemy, M. Fikri, M. Hashish, and E. Talkhan, "Speaker independent isolated Arabic word recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2008, pp. 697–700.

[11] A. Ali, Y. Zhang, P. Cardinal, N. Dahak, S. Vogel, and J. Glass, "A complete KALDI recipe for building Arabic speech recognition systems," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2014, pp. 525–529.

[12] Y. A. Alotaibi and A. Hussain, "Comparative analysis of Arabic vowels using formants and an automatic speech recognition system," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 3, no. 2, pp. 11–22, 2010.

[13] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, and D. Povey, "The htk book," *Cambridge Univ. Eng. Dept.*, vol. 3, no. 175, p. 12, 2002.

[14] S. Y. El-Mashad, M. I. Sharway, and H. H. Zayed, "Speaker independent Arabic speech recognition using support vector machine," Eszterhazy Karoly Foiskola Líceum Kiadó, Eötvös Loránd Univ., Tech. Rep., 2017.

[15] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

[16] M. Elmahdy, R. Gruhn, W. Minker, and S. Abdennadher, "Modern standard Arabic based multilingual approach for dialectal Arabic speech recognition," in *Proc. 8th Int. Symp. Natural Lang. Process.*, Oct. 2009, pp. 169–174.

[17] W. Aldjanabi, A. Dahou, M. A. A. Al-Qaness, M. A. Elaziz, A. M. Helmi, and R. Damasevicius, "Arabic offensive and hate speech detection using a cross-corpora multi-task learning model," *Informatics*, vol. 8, no. 4, p. 69, Oct. 2021.

[18] H. Hyassat and R. A. Zitar, "Arabic speech recognition using SPHINX engine," *Int. J. Speech Technol.*, vol. 9, nos. 3–4, pp. 133–150, Dec. 2006.

[19] M. A. M. Abushariah, R. N. Ainon, R. Zainuddin, M. Elshafei, and O. O. Khalifa, "Natural speaker-independent Arabic speech recognition system based on hidden Markov models using sphinx tools," in *Proc. Int. Conf. Comput. Commun. Eng. (ICCCE)*, May 2010, pp. 1–6.

[20] B. A. Q. Al-Qatab and R. N. Ainon, "Arabic speech recognition using hidden Markov model Toolkit(HTK)," in *Proc. Int. Symp. Inf. Technol.*, vol. 2, Jun. 2010, pp. 557–562.

[21] H. Bahi and M. Sellami, "Combination of vector quantization and hidden Markov models for Arabic speech recognition," in *Proc. ACS/IEEE Int. Conf. Comput. Syst. Appl.*, Jun. 2001, pp. 96–100.

[22] N. Hammami, M. Bedda, and N. Farah, "Spoken Arabic digits recognition using MFCC based on GMM," in *Proc. IEEE Conf. Sustain. Utilization Develop. Eng. Technol. (STUDENT)*, Oct. 2012, pp. 160–163.

[23] A. Ouisaadane and S. Safi, "A comparative study for Arabic speech recognition system in noisy environments," *Int. J. Speech Technol.*, vol. 24, no. 3, pp. 761–770, Sep. 2021.

[24] H. A. Alsayadi, A. A. Abdelhamid, I. Hegazy, and Z. T. Fayed, "Arabic speech recognition using end-to-end deep learning," *IET Signal Process.*, vol. 15, no. 8, pp. 521–534, Oct. 2021.

[25] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," 2018, *arXiv:1804.00015*.

[26] Y. Wang, T. Chen, H. Xu, S. Ding, H. Lv, Y. Shao, N. Peng, L. Xie, S. Watanabe, and S. Khudanpur, "Espresso: A fast end-to-end neural speech recognition toolkit," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 136–143.

[27] P. Ongsulee, "Artificial intelligence, machine learning and deep learning," in *Proc. 15th Int. Conf. ICT Knowl. Eng.*, Nov. 2017, pp. 1–6.

[28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 7553.

[29] X. Wang, Y. Zhao, and F. Pourpanah, "Recent advances in deep learning," *Int. J. Mach. Learn. Cybern.*, vol. 11, pp. 747–750, Jan. 2020.

[30] A. Mehrish, N. Majumder, R. Bharadwaj, R. Mihalcea, and S. Poria, "A review of deep learning techniques for speech processing," *Inf. Fusion*, vol. 99, Nov. 2023, Art. no. 101869.

[31] A. S. M. B. Wazir and J. H. Chuah, "Spoken Arabic digits recognition using deep learning," in *Proc. IEEE Int. Conf. Automatic Control Intell. Syst. (ICACIS)*, Jun. 2019, pp. 339–344.

[32] A. AbdAlmisreb, A. F. Abidin, and N. M. Tahir, "Maxout based deep neural networks for Arabic phonemes recognition," in *Proc. IEEE 11th Int. Colloq. Signal Process. Appl. (CSPA)*, Mar. 2015, pp. 192–197.

[33] A. Emami and L. Mangu, "Empirical study of neural network language models for Arabic speech recognition," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Dec. 2007, pp. 147–152.

[34] N. Zerari, S. Abdelhamid, H. Bouzgou, and C. Raymond, "Bidirectional deep architecture for Arabic speech recognition," *Open Comput. Sci.*, vol. 9, no. 1, pp. 92–102, Jan. 2019.

[35] B. Zada and R. Ullah, "Pashto isolated digits recognition using deep convolutional neural network," *Heliyon*, vol. 6, no. 2, Feb. 2020, Art. no. e03372.

[36] M. Abadi, "TensorFlow: Learning functions at scale," in *Proc. 21st ACM SIGPLAN Int. Conf. Funct. Program.*, Sep. 2016, p. 1.

[37] M. Alghamdi, M. Elshafei, and H. Al-Muhtaseb, "Arabic broadcast news transcription system," *Int. J. Speech Technol.*, vol. 10, no. 4, pp. 183–195, Dec. 2007.

[38] M. El Choubassi, H. El Khoury, C. J. Alagha, J. Skaf, and M. Al-Alaoui, "Arabic speech recognition using recurrent neural networks," in *Proc. 3rd IEEE Int. Symp. Signal Process. Inf. Technol.*, Dec. 2003, pp. 543–547.

[39] A. Messaoudi, H. Haddad, C. Fourati, M. B. Hmida, A. B. E. Mabrouk, and M. Graiet, "Tunisian dialectal end-to-end speech recognition based on DeepSpeech," *Proc. Comput. Sci.*, vol. 189, pp. 183–190, Jan. 2021.

[40] Z. J. M. Ameen and A. Abdulrahman Kadhim, "Deep learning methods for Arabic autoencoder speech recognition system for electro-larynx device," *Adv. Hum.-Comput. Interact.*, vol. 2023, pp. 1–11, Feb. 2023.

[41] O. Mahmoudi and M. F. Bouami, "Arabic speech commands recognition with LSTM & GRU models using CUDA toolkit implementation," in *Proc. 3rd Int. Conf. Innov. Res. Appl. Sci., Eng. Technol. (IRASET)*, May 2023, pp. 1–4.

[42] S. Nasr, R. Duwairi, and M. Quwaider, "End-to-end speech recognition for Arabic dialects," *Arabian J. Sci. Eng.*, vol. 48, no. 8, pp. 10617–10633, Aug. 2023.

[43] D. Yu and L. Deng, *Automatic Speech Recognition*, vol. 1. Berlin, Germany: Springer, 2016.

[44] N. S. Bostrom, *Paths, Dangers, Strategies*. Moscow, Russia: MIF Publishing House, 2016.

[45] A. Hussein, S. Watanabe, and A. Ali, "Arabic speech recognition by end-to-end, modular systems and human," *Comput. Speech Lang.*, vol. 71, Jan. 2022, Art. no. 101272.

[46] Z. J. M. Ameen and A. A. Kadhim, "Machine learning for Arabic phonemes recognition using electrolarynx speech," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 13, no. 1, p. 400, Feb. 2023.

[47] D. R. Reddy, "Speech recognition by machine: A review," *Proc. IEEE*, vol. 64, no. 4, pp. 501–531, 1976.

[48] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. ASSPM-3, no. 1, pp. 4–16, Jan. 1986.

[49] J. M. Tebelskis, *Speech Recognition Using Neural Networks*. USA: Carnegie Mellon Univ., 1995.

[50] M. Gales and S. Young, "The application of hidden Markov models in speech recognition," *Found. Trends Signal Process.*, vol. 1, no. 3, pp. 195–304, 2008.

[51] C. Rashmi, "Review of algorithms and applications in speech recognition system," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 4, pp. 5258–5262, 2014.

[52] M. A. Haque, A. Verma, J. S. R. Alex, and N. Venkatesan, "Experimental evaluation of CNN architecture for speech recognition," in *Proc. 1st Int. Conf. Sustain. Technol. Comput. Intell.* Germany: Springer, 2020, pp. 507–514.

[53] T. Zoughi, M. M. Homayounpour, and M. Deypir, "Adaptive windows multiple deep residual networks for speech recognition," *Exp. Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112840.

[54] T. Takiguchi and Y. Ariki, "PCA-based speech enhancement for distorted speech recognition," *J. Multimedia*, vol. 2, no. 5, pp. 13–18, Sep. 2007.

[55] O.-W. Kwon and T.-W. Lee, "Phoneme recognition using ICA-based feature extraction and transformation," *Signal Process.*, vol. 84, no. 6, pp. 1005–1019, Jun. 2004.

[56] M. Ziółko, R. Samborski, J. Gałka, and B. Ziółko, "Wavelet-Fourier analysis for speaker recognition," in *Proc. 17th Nat. Conf. Appl. Math. Biol. Med.*, vol. 134, 2011, p. 129.

[57] R. Haeb-Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition," in *Proc. ICASSP*, vol. 92, 1992, pp. 13–16.

[58] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[59] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 4052–4056.

[60] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of MFCC," *J. Comput. Sci. Technol.*, vol. 16, no. 6, pp. 582–589, Nov. 2001.

[61] C. Ittichaichareon, S. Suksri, and T. Yingthawornsuk, "Speech recognition using MFCC," in *Proc. Int. Conf. Comput. Graph., Simul. Model.*, vol. 9, 2012, pp. 135–138.

[62] M. Westphal, "The use of cepstral means in conversational speech recognition," in *Proc. EUROSPEECH*, 1997.

[63] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 578–589, Oct. 1994.

[64] H. Hermansky and P. Fousek, "Multi-resolution RASTA filtering for TANDEM-based ASR," Info-Sci., Switzerland, Tech. Rep., 2005.

[65] M. K. I. Molla and K. Hirose, "On the effectiveness of MFCCs and their statistical distribution properties in speaker identification," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Feb. 2004, pp. 136–141.

[66] N. Dave, "Feature extraction methods LPC, PLP and MFCC in speech recognition," *Int. J. Advance Res. Eng. Technol.*, vol. 1, no. 6, pp. 1–4, 2013.

[67] S. K. Gaikwad, B. W. Gawali, and P. Yannawar, "A review on speech recognition technique," *Int. J. Comput. Appl.*, vol. 10, no. 3, pp. 16–24, Nov. 2010.

[68] J. J. Bird, E. Wanner, A. Ekart, and D. R. Faria, "Phoneme aware speech recognition through evolutionary optimisation," in *Proc. Genetic Evol. Comput. Conf. Companion*, 2019, pp. 362–363.

[69] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for English conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4759–4763.

[70] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2013, pp. 273–278.

[71] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, vol. 247. Berlin, Germany: Springer, 1994.

[72] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[73] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," 2014, *arXiv:1412.5567*.

[74] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[75] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5884–5888.

[76] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[77] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[78] K. Irie, Z. Tüske, T. Alkhouli, R. Schluter, and H. Ney, "LSTM, GRU, highway and a bit of attention: An empirical overview for language modeling in speech recognition," in *Proc. Interspeech*, Sep. 2016, pp. 3519–3523.

[79] M. Alghmadi, "Kacst Arabic phonetic database," in *Proc. 15th Int. Congr. Phonetics Sci., Barcelona*, 2003, pp. 3109–3112.

[80] F. A. Elmisery, A. H. Khalil, A. E. Salama, and H. F. Hammed, "A FPGA-based HMM for a discrete Arabic speech recognition system," in *Proc. 12th IEEE Int. Conf. Fuzzy Syst.*, Dec. 2003, pp. 322–325.

[81] A. Amrouche and J. M. Rouvaen, "Arabic isolated word recognition using general regression neural network," in *Proc. 46th Midwest Symp. Circuits Syst.*, vol. 2, 2003, pp. 689–692.

[82] H. Bourouba, R. Djemili, M. Bedda, and C. Snani, "New hybrid system (supervised classifier/HMM) for isolated Arabic speech recognition," in *Proc. 2nd Int. Conf. Inf. Commun. Technol.*, 2006, pp. 1264–1269.

[83] E. M. Essa, A. S. Tolba, and S. Elmougy, "A comparison of combined classifier architectures for Arabic speech recognition," in *Proc. Int. Conf. Comput. Eng. Syst.*, Nov. 2008, pp. 149–153.

[84] H. Satori, H. Hiyassat, M. Haiti, and N. Chenfour, "Investigation Arabic speech recognition using CMU sphinx system," *Int. Arab J. Inf. Technol. (IAJIT)*, vol. 6, no. 2, pp. 186–190, 2009.

[85] M. Azmi, H. Tolba, S. Mahdy, and M. Fashal, "Syllable-based automatic Arabic speech recognition in noisy-telephone channel," *WSEAS Trans. Signal Process.*, vol. 4, no. 4, pp. 211–220, 2008.

[86] R. Kolobov, O. Okhapkina, A. P. Olga Omelchishina, R. Bedyakin, V. Moshkin, D. Menshikov, and N. Mikhaylovskiy, "Mediaspeech: Multilanguage ASR benchmark and dataset," 2021, *arXiv:2103.16193*.

[87] B-Ubuntu, *Quran Ayat Speech to Text*, openSLR, Czech Republic, 2022.

[88] openSLR, *Tunisian Modern Standard Arabic*, Czech Republic, 2017.

[89] A. Q. Ohi, M. F. Mridha, M. A. Hamid, and M. M. Monowar, "Deep speaker recognition: Process, progress, and challenges," *IEEE Access*, vol. 9, pp. 89619–89643, 2021.

[90] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

[91] A. P. Kaur, A. Singh, R. Sachdeva, and V. Kukreja, "Automatic speech recognition systems: A survey of discriminative techniques," *Multimedia Tools Appl.*, vol. 82, no. 9, pp. 13307–13339, Apr. 2023.

[92] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun.*, vol. 12, no. 3, pp. 247–251, Jul. 1993.

[93] V. Z. Kepuska and H. A. Elharati, "Robust speech recognition system using conventional and hybrid features of MFCC, LPCC, PLP, RASTA-PLP and hidden Markov model classifier in noisy conditions," *J. Comput. Commun.*, vol. 3, no. 6, pp. 1–9, 2015.

[94] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. ISMIR*, vol. 270, Plymouth, MA, USA, 2000, p. 11.

[95] J. Martinez, H. Perez, E. Escamilla, and M. M. Suzuki, "Speaker recognition using Mel frequency cepstral coefficients (MFCC) and vector quantization (VQ) techniques," in *Proc. 22nd Int. Conf. Electr. Commun. Comput.*, 2012, pp. 248–251.

[96] M. O. M. Khelifa, Y. M. Elhadj, Y. Abdellah, and M. Belkasmi, "Constructing accurate and robust HMM/GMM models for an Arabic speech recognition system," *Int. J. Speech Technol.*, vol. 20, no. 4, pp. 937–949, Dec. 2017.

[97] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Amer.*, vol. 87, no. 4, pp. 1738–1752, Apr. 1990.

[98] F. Hönig, G. Stemmer, C. Hacker, and F. Brugnara, "Revising perceptual linear prediction (PLP)," in *Proc. Interspeech*, Sep. 2005, pp. 2997–3000.

[99] H. B. Sailor and H. A. Patil, "Filterbank learning using convolutional restricted Boltzmann machine for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 5895–5899.

[100] G. K. Liu, "Evaluating gammatone frequency cepstral coefficients with neural networks for emotion recognition from speech," 2018, *arXiv:1806.09010*.

[101] P. F. Brown, V. J. D. Pietra, P. V. Desouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," *Comput. linguistics*, vol. 18, no. 4, pp. 467–480, 1992.

[102] A. Pauls and D. Klein, "Faster and smaller n-gram language models," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Human Lang. Technol.*, 2011, pp. 258–267.

[103] Y. Bengio, "Neural net language models," *Scholarpedia*, vol. 3, no. 1, p. 3881, 2008.

[104] K. Jing and J. Xu, "A survey on neural network language models," 2019, *arXiv:1906.03591*.

[105] O. Zheng, M. Abdel-Aty, D. Wang, Z. Wang, and S. Ding, "ChatGPT is on the horizon: Could a large language model be suitable for intelligent traffic safety research and applications?" 2023, *arXiv:2303.05382*.

[106] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018, *arXiv:1803.02155*.

[107] M. India, P. Safari, and J. Hernando, "Self multi-head attention for speaker recognition," 2019, *arXiv:1906.09890*.

[108] S. R. Eddy, "Hidden Markov models," *Current Opinion Struct. Biol.*, vol. 6, no. 6, pp. 361–365, 1996.

[109] S. R. Eddy, "What is a hidden Markov model?" *Nature Biotechnol.*, vol. 22, no. 10, pp. 1315–1316, Oct. 2004.

[110] J. Picone, "Continuous speech recognition using hidden Markov models," *IEEE ASSP Mag.*, vol. 7, no. 3, pp. 26–41, Jul. 1990.

[111] M. Afify, O. Siohan, and R. Sarikaya, "Gaussian mixture language models for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2007, p. 29.

[112] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. K. Goel, M. Karafiát, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas, "Subspace Gaussian mixture models for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 4330–4333.

[113] D. A. Reynolds, "Gaussian mixture models," *Encyclopedia Biometrics*, vol. 741, nos. 659–663, Jul. 2009.

[114] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiat, and A. Rastrow, "The subspace Gaussian mixture model—A structured model for speech recognition," *Comput. Speech Lang.*, vol. 25, no. 2, pp. 404–439, 2011.

[115] P.-S. Huang and M. Hasegawa-Johnson, "Cross-dialectal data transferring for Gaussian mixture model training in Arabic speech recognition," *Constraints*, vol. 1, pp. 1–4, Mar. 2012.

[116] S. Bhatia, A. Devi, R. I. Alsuwailem, and A. Mashat, "Convolutional neural network based real time Arabic speech recognition to Arabic Braille for hearing and visually impaired," *Frontiers Public Health*, vol. 10, May 2022, Art. no. 898355.

[117] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19143–19165, 2019.

[118] E. Abdelmaksoud, A. Hassen, N. Hassan, and M. Hesham, "Convolutional neural network for Arabic speech recognition," *Egyptian J. Lang. Eng.*, vol. 8, no. 1, pp. 27–38, Apr. 2021.

[119] R. Amari, Z. Noubigh, S. Zrigui, D. Berchech, H. Nicolas, and M. Zrigui, "Deep convolutional neural network for Arabic speech recognition," in *Proc. Int. Conf. Comput. Collective Intell.* Germany: Springer, 2022, pp. 120–134.

[120] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.

[121] T. Donkers, B. Loepp, and J. Ziegler, "Sequential user-based recurrent neural network recommendations," in *Proc. 11th ACM Conf. Recommender Syst.*, 2017, pp. 152–160.

[122] A. M. Ahmad, S. Ismail, and D. F. Samaon, "Recurrent neural network with backpropagation through time for speech recognition," in *Proc. IEEE Int. Symp. Commun. Inf. Technol.*, Oct. 2004, pp. 98–102.

[123] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU," *J. Artif. Intell. Soft Comput. Res.*, vol. 9, no. 4, pp. 235–245, Oct. 2019.

[124] A. Ahmed, Y. Hifny, K. Shaalan, and S. Toral, "End-to-end lexicon free Arabic speech recognition using recurrent neural networks," in *Computational Linguistics, Speech and Image Processing for Arabic Language*. Singapore: World Scientific, 2019, pp. 231–248.

[125] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 1597–1600.

[126] M. Sameer, A. Talib, A. Hussein, and H. Husni, "Arabic speech recognition based on encoder–decoder architecture of transformer," *J. Techn.*, vol. 5, no. 1, pp. 176–183, 2023.

[127] S. Latif, A. Zaidi, H. Cuayahuitl, F. Shamshad, M. Shoukat, and J. Qadir, "Transformers in speech processing: A survey," 2023, *arXiv:2303.11607*.

[128] M. Hadwan, H. A. Alsayadi, and S. Al-Hagree, "An end-to-end transformer-based automatic speech recognition for Qur'an reciters," *Comput., Mater. Continua*, vol. 74, no. 2, pp. 3471–3487, 2023.

[129] M. H. Sazli, "A brief review of feed-forward neural networks," *Commun. Faculty Sci. Univ. Ankara Ser. A2–A3 Phys. Sci. Eng.*, vol. 50, no. 1, pp. 3471–3487, 2006.

[130] K.-R. Koch and K.-R. Koch, "Bayes theorem," in *Bayesian Inference With Geodetic Applications*. Turkey: Ankara Univ., 1990, pp. 4–8.

[131] N. Seshadri and C.-E.-W. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. Commun.*, vol. 42, no. 234, pp. 313–323, Feb./Apr. 1994.

[132] A. Al Harere and K. Al Jallad, "Quran recitation recognition using end-to-end deep learning," 2023, *arXiv:2305.07034*.

[133] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 69–88, Jan. 2002.

[134] B. Dorr, M. Snover, and N. Madnani, "Part 5: Machine translation evaluation," Handb. Nat. Lang. Process. Mach. Transl. DARPA Glob. Auton. Lang. Exploit, The Netherlands, Tech. Rep., 801, 2011.

[135] A. Ahmed, Y. Hifny, K. Shaalan, and S. Toral, "Lexicon free Arabic speech recognition recipe," in *Proc. Int. Conf. Advanced Intell. Syst. Inform.* USA: Springer, 2017, pp. 147–159.

[136] A. Ali and S. Renals, "Word error rate estimation for speech recognition: E-WER," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 20–24.

[137] I. Bazzi, R. Schwartz, and J. Makhoul, "An omnifont open-vocabulary OCR system for English and Arabic," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 6, pp. 495–504, Jun. 1999.

[138] A. Stolcke, Y. Konig, and M. Weintraub, "Explicit word error minimization in n-best list rescoring," in *Proc. 5th Eur. Conf. Speech Commun. Technol. (Eurospeech )*, Sep. 1997.

[139] A. Y. Vadwala, K. A. Suthar, Y. A. Karmakar, and N. Pandya, "Survey paper on different speech recognition algorithm: Challenges and techniques," *Int. J. Comput. Appl.*, vol. 175, no. 1, pp. 31–36, Oct. 2017.

[140] P. Sahu, M. Dua, and A. Kumar, "Challenges and issues in adopting speech recognition," in *Proc. Speech Lang. Process. Human-Machine Commun.*, 2018, pp. 209–215.

[141] S. Nivetha, "A survey on speech feature extraction and classification techniques," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Feb. 2020, pp. 48–53.

[142] A. Singh, V. Kadyan, M. Kumar, and N. Bassan, "ASRoIL: A comprehensive survey for automatic speech recognition of Indian languages," *Artif. Intell. Rev.*, vol. 53, no. 5, pp. 3673–3704, Jun. 2020.

[143] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 12449–12460.

[144] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 2, pp. 92–102, Apr. 2018.

**ASHIFUR RAHMAN** was born in Jashore, Bangladesh. He received the B.Sc. degree in computer science and engineering (CSE) from Bangladesh University of Business and Technology (BUBT), Bangladesh, in 2022. He is currently a Research Assistant with the Center for Industrial Automation, Robotics & IoT (RIoT) Research Centre, Independent University, Bangladesh. He is always willing to learn new things with full enthusiasm and passion. He is particularly interested in machine learning, deep learning, pattern recognition, biomedical imaging, and computer vision. He has hands-on experience in Python, Keras, TensorFlow, Pandas, Matplotlib, Seaborn, Sklearn, and SciPy. He is researching histopathology image-based automated disease detection systems. His enthusiasm for his field has driven him to seek further opportunities in academia. He is looking for M.Sc. or Ph.D. positions in the exciting fields of machine learning, deep learning, or biomedical imaging.

**MD. MOHSIN KABIR** received the Bachelor of Science degree in CSE from Bangladesh University of Business and Technology (BUBT), in 2021. He is currently pursuing the joint master's degree in intelligent field robotics systems with the University of Girona, Spain, and Eötvös Loránd University, Hungary. He holds a position as a Lecturer with the Department of CSE, BUBT (Study-Leave). He was a Research Assistant with BUBT and a Researcher with the Advanced Machine Intelligence Research Laboratory. Also, he has contributed to several prominent research laboratories around the globe, including the Database System Laboratory, The University of Aizu, Japan. With an extensive research background, he has authored more than twenty articles in high-impact journals, such as *Heliyon*, *Computers & Security*, *Journal of Agriculture and Food Research*, IEEE Access, *Sensors*, *Cognitive Computation and Systems*, *International Journal of Information Management Data Insights*, and *Mathematics*. In addition, he has contributed to the scientific community by publishing more than ten conference papers and a few book chapters in books related to machine learning and AI. His primary research interests include artificial intelligence, machine learning, deep learning, computer vision, the IoT, and robotics.

**M. F. MRIDHA** (Senior Member, IEEE) received the Ph.D. degree in AI/ML from Jahangirnagar University, in 2017. He is currently an Associate Professor with the Department of Computer Science, American International University-Bangladesh (AIUB). Before that, he was an Associate Professor and the Chairperson of the Department of CSE, Bangladesh University of Business and Technology. He also worked as the CSE Department Faculty Member with the University of Asia Pacific and the Graduate Head, from 2012 to 2019. For more than ten years, he has been with the bachelor's and master's students as a supervisor of their thesis work. His research experience, within both academia and industry, results in more than 120 journal and conference publications. His research work contributed to the reputed journals of *Scientific Reports* (Nature), *Knowledge-Based Systems*, *Artificial Intelligence Review*, IEEE Access, *Sensors*, *Cancers*, and *Applied Sciences*. His research interests include artificial intelligence (AI), machine learning, deep learning, natural language processing (NLP), and big data analysis. He has served as a program committee member for several international conferences/workshops. He served as an Associate Editor for several journals, including *PLOS One*. He has served as a Reviewer for reputed journals and IEEE conferences, such as HONET, ICIEV, ICCIT, *IJCCI*, ICAEE, ICCAIE, ICSIPA, SCORED, ISIEA, APACE, ICOS, ISCAIE, BEIAC, ISWTA, IC3e, *Coast*, icIVPR, ICSCT, 3ICT, and DATA21.

**MOHAMMED ALATIYYAH** is currently an Assistant Professor in computer science with the Computer Science Department, Prince Sattam bin Abdulaziz University, Saudi Arabia. His research interests include recommender systems and computer vision, such as group recommender systems, travel recommender systems, and drone vision.

**HAIFA F. ALHASSON** (Member, IEEE) received the B.Sc. degree in computer science from Qassim University, Saudi Arabia, the M.Sc. degree in computer science from King Saud University, Saudi Arabia, and the Ph.D. degree in computer science from Durham University, U.K.

She is currently an Assistant Professor with the Computer College, Qassim University. Her interdisciplinary research focuses on image processing and machine learning. In particular, her research aims to understand better machine learning in object detection and recognition required for variable tasks.

**SHUAA S. ALHARBI** (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science from Qassim University, Saudi Arabia, and the Ph.D. degree in computer science from Durham University, U.K.

She is currently an Assistant Professor with the Computer College, Qassim University. Her interdisciplinary research focuses on machine learning and image processing in biology and medical domains. In particular, she is interested in using deep learning to analyze medical images and improve the accuracy of disease diagnosis, which is a rapidly growing area of interest.