Advances in Artificial Intelligence and Machine Learning; Research 5 (3) 4074-4093 Received 4-05-2025; Accepted 24-07-2025; Published 31-07-2025

A Deep Learning Framework for Arabic Continuous Speech Keyword Spotting in Low-Resource Settings Using Isolated-Word Keyword Spotting and Posterior Probability Functions

Osama Deeb osama.deeb@hiast.edu.sy
Department of Telecommunications, Higher Institute for Applied Sciences and Technology
Damascus, Syria

Assef Jafar assef.jafar@hiast.edu.sy
Department of Informatics, Higher Institute for Applied Sciences and Technology
Damascus, Syria

Oumayma Al Dakkak oumayma.dakkak@hiast.edu.sy
Department of Telecommunications, Higher Institute for Applied Sciences and Technology
Damascus, Syria

Corresponding Author: Osama Deeb

Abstract

Continuous Speech Keyword Spotting (CSKWS) presents a challenging paradigm shift from isolated-word Keyword Spotting (KWS), focusing on discovering the occurrences of predefined keywords within continuous speech streams. This paper addresses the prevalent issue of data scarcity in CSKWS for low-resource languages by introducing the innovative Posterior Probability Function approach (PPF-CSKWS). Utilizing unsupervised method, this approach leverages a few-shot KWS system to derive posterior probability estimates of keyword occurrences as a discrete-time functions. Contrary to the data-intensive training procedures typically associated with CSKWS system development, this method requires only 15 isolated audio samples per keyword, significantly reducing the data bottleneck. The generated posterior probability functions provide crucial temporal information, facilitating both keyword identification and localization. This characteristic allows for the reformulation of the CSKWS problem as a detection task, with evaluation metrics such as mean Average Precision (mAP) and Maximum Term Weighted Value (MTWV) being applicable. To evaluate the proposed method, a dedicated Arabic speech corpus was constructed. Experimental results demonstrated the achievement of mAP = 0.613 and MTWV = 0.641. This performance meets or exceeds that of established supervised techniques requiring significantly larger quantities of labelled training data, thereby demonstrating the potential of PPF-CSKWS in data scarcity scenarios.

Keywords: Arabic, Continuous speech keyword spotting, Deep learning, Low-Resource languages, Posterior probability functions.

Citation: Deeb, et al. A Deep Learning Framework for Arabic Continuous Speech Keyword Spotting in Low-Resource Settings Using Isolated-Word Keyword Spotting and Posterior Probability Functions. **Advances in Artificial Intelligence and Machine Learning**. 2025;5(3):229.

1. INTRODUCTION

Keyword Spotting (KWS) systems are designed to detect the occurrence of predefined words within a speech context, representing a continuously evolving challenge that has acquired significant attention in speech processing. With the rapid proliferation of smart devices and Internet of Things (IoT) over the past decade, and the increasing demand for seamless human-machine interaction, this area of research has gained substantial prominence [1]. KWS systems are integral to voice-activated technologies, with growing industrial applications including intelligent vehicle cabins, smart device control in smart cities and virtual assistant activation with phrases such as "Hey Siri" and "Ok Google" becoming familiar nowadays to activate virtual assistants on smart phones [2, 3]. Beyond device activation, KWS holds critical applications in voice search, audio classification, and automatic audio labelling. Moreover, KWS plays a vital role in monitoring multimedia content to identify instances of hate speech, racism, and potentially criminal activities in real-time voice communications, underscoring its broader societal and security implications [4, 5].

The evolution of KWS systems began with the recognition of a limited set of predefined isolated words, a paradigm later referred to as "closed-vocabulary KWS". Subsequently, the demand emerged for systems that allow users to customize and modify the set of target keywords, giving rise to "open-vocabulary or user-defined KWS". As technology advanced, the necessity for devices to comprehend people's phrases and interact with them led to the emergence of "Continuous Speech Keyword Spotting (CSKWS)", marking a significant evolution in this field. Each of these paradigms presents its own challenges, however, processing continuous speech introduces additional complexities, such as modelling non-speech segments, mitigating interference from adjacent words, and addressing the high similarity between target words and fractions of other words. These factors collectively affect the performance of the system by increasing false alarm rates [5, 6].

The initial efforts to tackle the KWS problem involved the application of Hidden Markov Models (HMM) [7–9]. Later, Deep Neural Networks (DNN) were adopted [10–13]. The provision of training data is a cornerstone in addressing such problems, with the complexity of this task varying significantly based on the context. When the target keywords are fixed and pertain to high-resource languages such as English, the process is relatively straightforward. This scenario is exemplified by systems designed to capture voice commands for device control, where datasets like Google Speech Commands [14] are readily available and applicable. However, the situation becomes considerably more complex when the target keywords are custom and subject to frequent changes, particularly in low-resource languages [15].

In this research, a CSKWS model tailored for Arabic was proposed, with a particular focus on addressing the data scarcity challenges inherent to Arabic. In this context, data scarcity refers to the scarcity or difficulty in obtaining annotated training data, which typically consists of speech samples (short phrases) containing target keywords paired with transcriptions. While Arabic audio clips are abundant, the lack of accompanying transcriptions renders them unsuitable for traditional supervised learning approaches. Building on an established few-shot closed-vocabulary KWS system [16] that attained 99.7% accuracy with only 15 samples per keyword, the present work adapts this architecture for continuous speech detection through unsupervised methodology. This approach eliminates the need for additional training data or significant architectural modifications. By treating the problem as a detection task, this model not only identifies the occurrence of keywords but also localizes

their positions and durations within continuous speech, thereby enhancing its utility in real-world applications. The principal contributions of this work can be summarized as follows:

- A novel approach that leverages a few-shot closed-vocabulary KWS system to provide a CSKWS solution without requiring additional computational resources or training data.
- Development and empirical validation of an unsupervised algorithm that utilizes contextual representations of speech to precisely identify temporal keyword positions within continuous speech.
- Demonstration of effective CSKWS model trained with limited isolated-word samples (=15 per keyword) rather than extensive sentence-level data, establishing a practical framework for low-resource language applications.
- Creation and deployment of a real-world evaluation dataset for Arabic CSKWS, derived from the Multi-Genre Broadcast (MGB-2) corpus and specifically designed for detection task assessment.
- Reformulation of CSKWS as a detection problem rather than a classification task, with comprehensive performance evaluation using Term-Weighted Value (TWV) and mean Average Precision (mAP) metrics.

The remainder of this paper is organized as follows: Section 2 reviews related work in keyword spotting, with particular emphasis on continuous speech approaches. Section 3 details the foundational materials, including the baseline few-shot closed-vocabulary KWS system and the Arabic corpus used to construct evaluation dataset. Section 4 provides a comprehensive description of the proposed methodology. Section 5 outlines the evaluation metrics and evaluation dataset. Section 6 presents the results and and comparison with existing approaches. Finally, Section 7 concludes the paper with key findings and implications.

2. RELATED WORK

Initial efforts to address the KWS problem relied on Hidden Markov Models (HMMs), where a separate HMM was constructed for each keyword, along with an additional HMM for non-keyword segments. The observation probabilities in these models were typically modelled using Gaussian Mixture Models (GMMs) [4, 7, 8]. These approaches employed sequence matching algorithms such as Viterbi, which, while effective, were computationally intensive, prompting the need for more efficient methods [17, 18]. Subsequently, various types of neural networks were introduced, demonstrating significant improvements in accuracy and computational efficiency, thereby enabling their deployment on resource-constrained smart devices [10–13, 19]. In closed-vocabulary KWS, supervised learning was employed, utilizing labelled audio samples [10–13]. For open-vocabulary KWS, text-to-speech matching techniques were developed, allowing users to define keyword sets in textual form. These models typically incorporated a text encoder and an audio encoder to extract embeddings from text and audio inputs, respectively, with specialized loss functions ensuring the convergence of these embeddings in latent spaces. Such approaches heavily relied on datasets comprising audio phrases and their corresponding transcriptions to train both the audio and text encoders [6, 19–22].

To address the challenge of data scarcity in low-resource languages, numerous methodologies have shifted towards transfer learning, wherein models are initially trained on high-resource languages such as English and subsequently fine-tuned with limited target language datasets [13]. An alternative strategy to mitigate data scarcity employs unsupervised methodologies, including Dynamic Time Warping (DTW), to analyze pattern similarities in acoustic feature representations associated with keywords and corresponding waveform signals [23].

For CSKWS, researchers have investigated various methodologies. Early methodologies focused on leveraging Automatic Speech Recognition (ASR) systems, followed by keyword identification through text-based search algorithms [18]. However, this approach inherits the inherent limitations of conventional ASR systems and text-based search methods, including error propagation from transcription inaccuracies and semantic ambiguities. Furthermore, such frameworks necessitate the development of robust ASR architectures and extensive labelled training datasets—requirements that are prohibitively resource-intensive for low-resource linguistic environments— rendering such approaches suboptimal in scenarios with limited data availability or computational constraints. Seth et al., [24] proposed a prototypical metric transfer learning approach for CSKWS in scenarios with limited training data. Speech phrases were synthesized by randomly extracting speech phrases from publicly available audio and inserting keywords at varying positions. During the training phase, a prototype for each keyword was generated using acoustic embeddings extracted from a DeepSpeech model [25]. The prototypical loss function was utilized to minimize the distance between sentences containing the keyword and its corresponding prototype while simultaneously maximizing the distances to prototypes of other keywords. Zhao et al., [5] proposed an anchor-free detector for CSKWS, drawing inspiration from anchor-based object detectors commonly used in Computer Vision (CV). They framed CSKWS as a detection problem, enabling the prediction of the center location of a keyword in continuous speech and the regression of the keyword's length. Their approach utilized an up-convolutional residual network, specifically Res34, as the backbone model, and three trainable prediction heads were employed to determine keyword occurrence, length, and precise word positions. The model was tailored for English and was trained and evaluated on LibriSpeech [26]. Xi et al., [19] introduced a novel Contrastive Learning with Audio Discrimination (CLAD) approach to learn keyword representations with both audio-text matching and audio-audio discrimination capabilities for user-defined CSKWS. They argued that incorporating audio-audio discrimination aids in addressing challenges associated with continuous speech, such as co-articulation and streaming word segmentation, which can produce similar audio patterns for distinct texts, potentially leading to false alarms. To assess their models, they constructed an English dataset derived from the LibriSpeech dataset. Kim et al., [27] proposed an on-device query-by-example keyword spotting system. The model operates in two stages: query enrolment and testing. During the query enrolment phase, a small-footprint ASR system is utilized to extract phonetic posteriors, which are then used to construct a finite-state transducer (FST). In the testing phase, the FST is employed to compute a log-likelihood score for the input audio. The model was evaluated using two English keywords, with the data sourced from publicly available datasets.

Based on the results discussed in this review, it can be concluded that isolated-word KWS systems, in both open and closed scenarios, have achieved satisfactory performance levels. However, CSKWS still facing unresolved challenges, particularly in situations involving data scarcity, making it an active area of research. While the majority of existing studies on CSKWS have focused on English, very few studies have addressed low-resource languages, primarily due to the difficulties in obtaining sufficient training data. Our proposed approach to address data scarcity involves upgrading an

isolated-word KWS system, which is trainable with limited data, into a CSKWS system without requiring additional training data. This represents a significant step toward enabling CSKWS for low-resource languages.

3. MATERIALS

This section describes in brief the few-shot isolated-word KWS which will be used to accomplish the CSKWS task, and the MGB-2 corpus, which will be used to construct the CSKWS evaluation dataset.

3.1 Few-Shot Isolated-Word KWS Model (KWS-base)

A transformer-based isolated-word KWS model trainable with limited number of samples was previously presented in [16]. This model comprises two primary components: the contextual representation extractor and the matching and classification module, as illustrated in FIGURE 1. The contextual representation extractor is based on a pre-trained HuBERT[1]-Large model. HuBERT [28] divides the input waveform *w* into frames of 20-millisecond and generates contextual representation for each frame outputting a tensor with [batch_size, sequence_length, hidden_size] dimensions. Given the use of HuBERT-Large and the processing of one waveform at a time in the inference mode, the output tensor assumes the shape [1, sequence_length, 1024], where "sequence_length" dynamically varies depending on the duration of the input waveform.

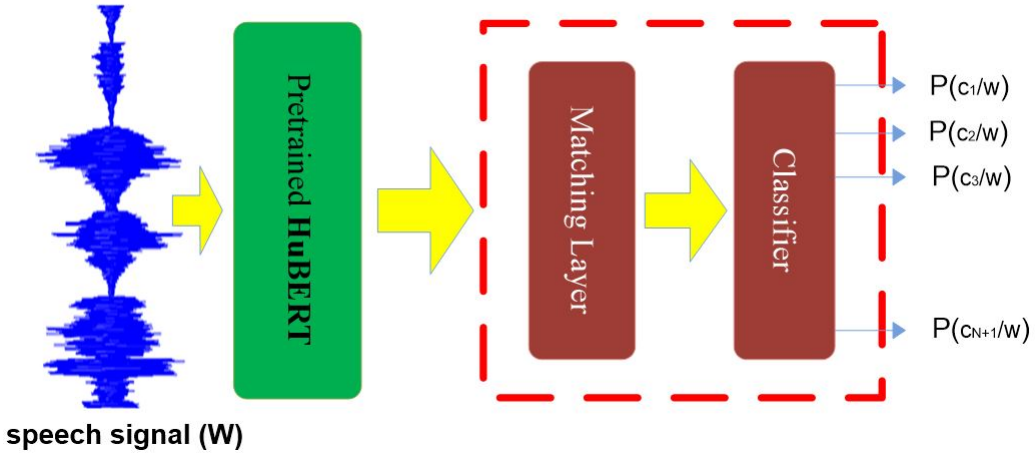

Figure 1: Isolated-word KWS architecture [16]

The matching layer computes the average of the tensor across the sequence_length dimension, resulting in a condensed tensor of shape [1, 1, 1024], which is subsequently fed into the classifier. The classifier is designed to categorize the output tensors into N+1 classes representing *N* keywords, and an additional global class, denoted as "unknown" encompassing all non-keywords. A softmax layer at the final stage of the classifier generates the posterior probabilities $P(c_i/w) : i \in [1 : N+1]$,

[1] HuBERT: Hidden Unit Bidirectional Encoder Representations from Transformers.

which quantify the likelihood of the tensor belonging to the class c_i. The final classification decision is made based on these posteriors following the formula provided in Equation 1.

$$ChosenClass = \underset{c_i}{Argmax}\left(P\left(c_i/w\right)\right) \quad (1)$$

This model was implemented and evaluated it on Arabic. It has demonstrated the capability to achieve 99.7% accuracy while being trained with only 15 audio samples per keyword, rendering it highly suitable for scenarios characterized by data scarcity. For the current work, a version of this model was constructed to detect eight predefined Arabic keywords: "إلغاء: cancel", "إغلاق: close", "ثمانية: eight", "يسار: left", "تسعة: nine", "خيارات: options" "توقف: stop" and "أعلى: up" (with their English equivalents provided for clarity). The model was trained with a dataset consisting 120 positive audio samples (15 samples per keyword) and 300 negative samples representing non-keyword instances. This trained version of the model was used in all the following experiments and will be referred to as "**KWS-base**".

3.2 The Multi-Genre Broadcast (MGB-2) Corpus

The MGB-2 dataset is an Arabic audio corpus developed by the Qatar Computing Research Institute (QCRI) [29]. It comprises over 1,200 hours of speech audio accompanied with transcriptions extracted using an Arabic ASR system developed by QCRI with no timing information. The audio clips, recorded directly from TV broadcasts without post-processing, may contain ambient noise, background music, and overlapping speech from multiple speakers. While these characteristics enhance the dataset's alignment with real-world conditions, they also introduce significant challenges for models evaluated with it. For the evaluation of the proposed CSKWS model, a subset of this corpus was curated by selecting short sentences containing the target keywords along with their corresponding audio, forming the basis of our evaluation dataset.

4. METHODS

This section details our methodological framework for addressing the CSKWS challenge, including posterior probability functions extraction using KWS-base, decision-making strategy and the proposed CSKWS system architecture.

4.1 CSKWS Approach: Building Posterior Probability Functions

When a speech segment of length L, extracted from position k in an utterance, is processed by the KWS-base system, the softmax layer generates the posterior probabilities $P\left(c_i/w\right)$ for each class at position k. By sliding the window with a stride ($strd$) across the utterance, a discrete-time function representing the posterior probability of class c_i at position k is obtained, as formally defined in Equation 2.

$$f_{c_i}\left(k\right) = P\left(c_i/w_k\right) \quad @k^{th}window \quad (2)$$

By selecting an appropriate window length and stride, the entire sentence can be systematically scanned, generating the posterior probability function for each class. At each position k, these

functions provide an estimate of the likelihood that the corresponding segment of speech belongs to a class c_i. For a keyword kw_m (length L_m) occurring in sentence S, setting window length $L = L_m$ and stride $strd = L_m/10$ results in 19 overlapping windows intersecting kw_m. The posterior probability function f_{c_m} demonstrates: (1) monotonic increase during initial overlap, peaking at complete keyword containment; (2) symmetric decay during window departure; and (3) baseline return upon full disengagement. FIGURE 2 illustrates this ideal behavior, while FIGURE 3 shows empirical results from applying the KWS-base model to a phrase beginning with the keyword "options".

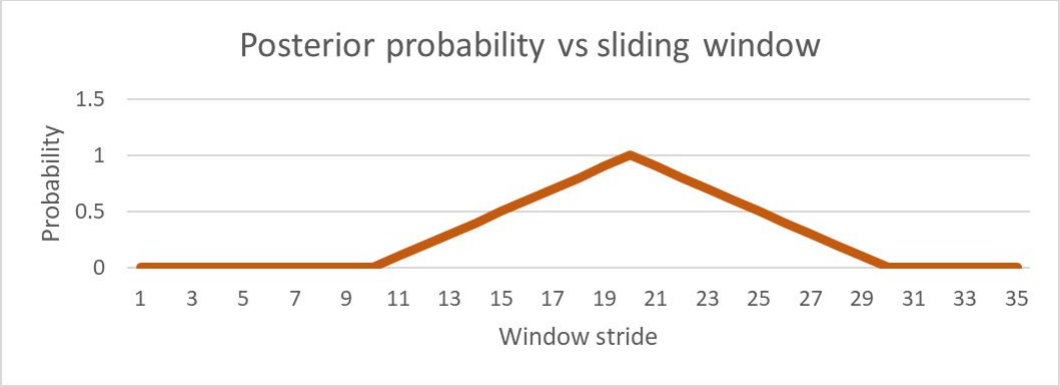

Figure 2: An ideal representation of the posterior probability function changes while sliding a window of length $L = 1$ seconds on a sentence containing a keyword of length $L_m = 1$ seconds by a 0.1 seconds stride.

FIGURE 3 demonstrates that the global class posterior probability (grey "unknown" curve) dominates most temporal regions, reflecting absence of target keywords. During keyword occurrences, however, class-specific probability function (brown "options", green "eight") exhibit characteristic peaks. This discriminative behavior confirms the functions' utility for both keyword detection and temporal localization when processed with an optimal decision strategy.

4.2 Decision-Making Strategy

Reliable keyword detection requires multi-window analysis, as single-window evaluation risks false positives from spurious maxima (FIGURE 3). Moreover, true keyword kw_m occurrences exhibit two key characteristics: (1) sustained elevation of f_{c_m} across consecutive windows, and (2) significant divergence of f_{c_m} from alternative $f_{c_{i,\ i\neq m}}$. Conversely, non-keyword regions show dominance of the global class posterior probability function f_{c_g}. These observations motivate our detection criterion: keyword kw_m is confirmed when f_{c_m} exceeds $P_{threshold}$ for at least $N_{threshold}$ consecutive windows, as formalized in Equation 3.

$$kw_m\ appeared = \begin{cases} true : if\ f_{c_m} \geq P_{threshold}\ for\ N_{threshold}\ successive\ windows \\ false : otherwise \end{cases} \tag{3}$$

FIGURE 4 shows comparative analysis of keyword detection performance before and after applying the decision-making strategy, using a test phrase containing the keyword "options". The colored

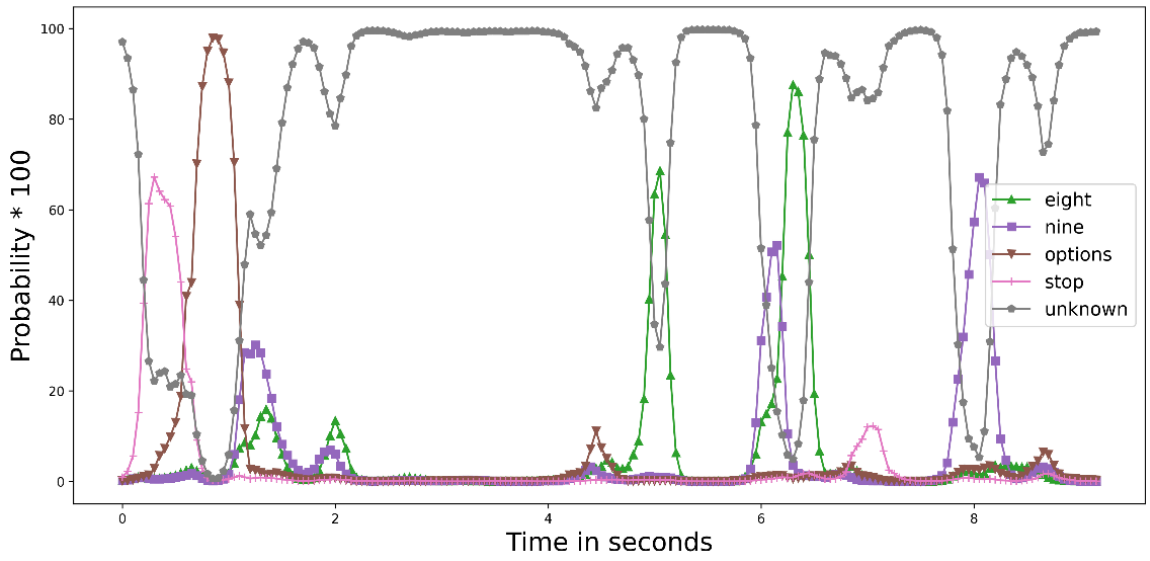

Figure 3: Posterior probability functions for 4 different Arabic keywords (eight, nine, options, stops) and the global class (denoted as "unknown") extracted from continuous waveform containing the word "options" at its beginning

panels at the top of the figure gives an indication of the predicted intervals for keywords occurrences with/without applying the proposed decision-making strategy.

The optimal values for $P_{threshold}$ and $N_{threshold}$ will be determined empirically through evaluation curves to ensure robust and accurate detection (Section 6.1). Additionally, k_{met} is defined as the window index at which the condition was first satisfied, and k_{brk} as the window index at which the condition was broken, hence, the start and end time of the keyword can be calculated as follows:

$$\begin{aligned} start\ time &= (k_{met} - N_{threshold}) * stride \\ end\ time &= k_{brk} * stride \end{aligned} \tag{4}$$

Implementing the decision-making strategy will eliminate spurious detections, thereby reduce the false alarm rate while maintain the true positive detection rate, ultimately enhancing the overall performance of the model.

4.3 The Proposed CSKWS Model

The CSKWS task was framed as a detection problem. Given a predefined set of keywords $Q = \{kw_1, kw_2, kw_3, \ldots\ldots, kw_N\}$ and an input continuous speech signal S, the model is designed to detect the occurrence of any keyword from the set and accurately localize it. In low-resource languages, where data scarcity is a significant challenge, an unsupervised algorithm is a suitable choice for performing CSKWS. The KWS-base architecture is proposed for extracting posterior

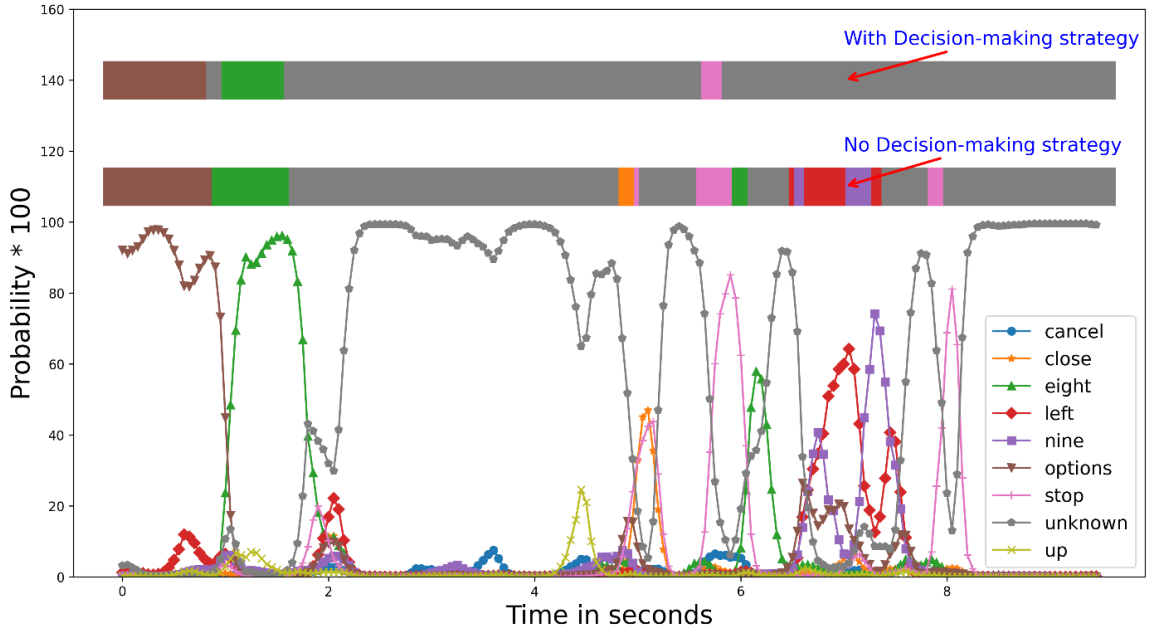

Figure 4: Effect of applying the Decision-making strategy at ($P_{threshold}$ = 73, $N_{threshold}$ = 3) on a system designed to detect list of 8 keywords in addition to the global class (denoted as "unknown"). Brown color represents a true positive detection of the keyword "options" and its position and duration, while green and pink colors represent false alarms of keywords "eight" and "stop" respectively. Gray color represents all words outside the list.

probability functions using a sliding window, as explained in Section 4.1, followed by applying the decision-making strategy described in Section 4.2 to accomplish the detection task.

The HuBERT model utilized in the isolated-word KWS system is characterized by a large parameter count (316 million parameters), which inherently leads to a relatively high inference time. When employing the sliding window approach, multiple overlapped segments of the waveform are processed repeatedly by the HuBERT model, resulting in a multiplicative increase in inference time that scales with the window size and stride. However, reducing the stride is crucial to ensure that no critical segments of the speech signal are missed between successive windows. In other word, there is a trade-off between the stride length and the model's accuracy; while increasing the stride will improve the inference speed, it will increase the miss rate and hence reduce the model's accuracy (this behavior has been demonstrated in practice, increasing the stride from 40ms to 80ms reduced inference time by 48% but decreased accuracy by 14%). To mitigate this challenge, an alternative strategy is proposed: instead of applying the sliding window directly to the raw waveform, it is applied to the contextual representations of the waveform generated by the HuBERT. FIGURE 5 illustrates the proposed methodology. Empirical evaluations demonstrate that this approach achieves about 20-fold reduction in inference time for identical audio files processed on the same hardware platform, while maintaining near-optimal detection accuracy with minimal computational overhead.

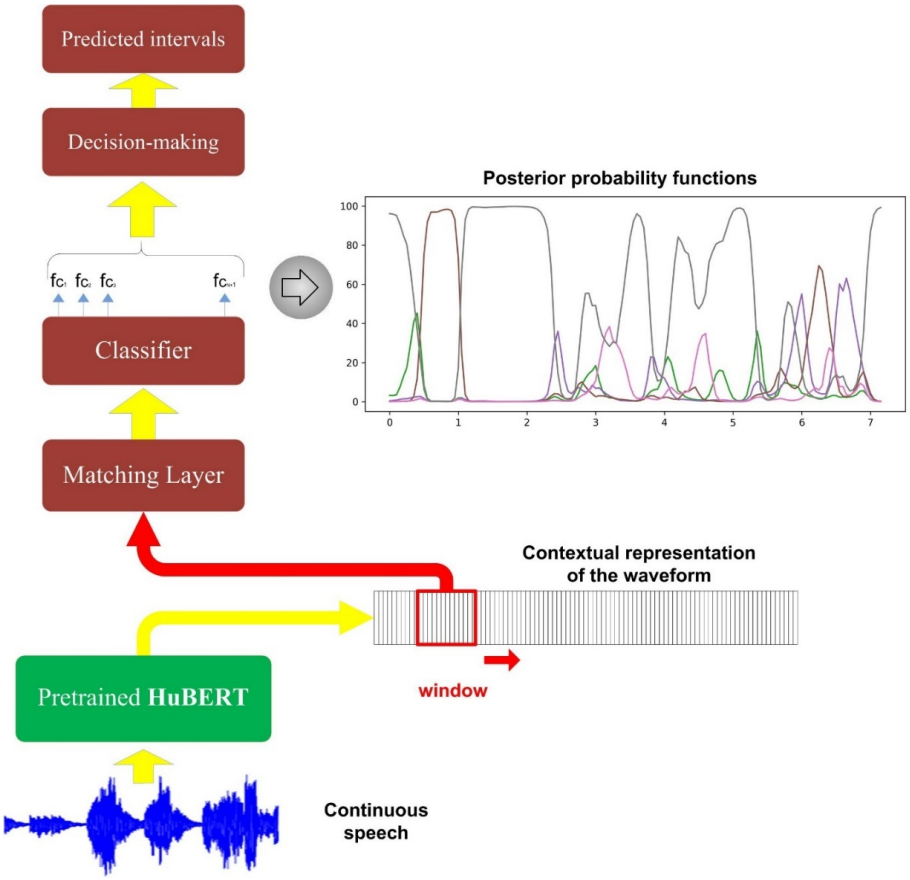

Figure 5: Architecture of the proposed model "PPF-CSKWS": applying sliding window on contextual representation of the waveform to extract posterior probability functions and Decision-making strategy.

The pretrained HuBERT model generates the contextual representation of the entire waveform, comprising F vectors corresponding to the audio signal frames. A window of V vectors is then extracted and passed to the matching layer, followed by the trained classifier, which outputs the posterior probabilities. During the design of the KWS model, simplicity was prioritized in the classifier's architecture, enabling it to be effectively trained with a limited number of samples [16]. This simplicity, resulted in significant reduction in inference time which allowed for minimizing the stride while sliding the window over the contextual representations, thereby producing more accurate posterior probability functions. These functions are subsequently utilized to identify intervals of keyword occurrences through the predefined decision-making strategy.

During inference, for a specific waveform, the proposed CSKWS model will output the Predicted Intervals ($Inv_{predict}$), where each predicted interval is characterized by three parameters: $start\_t_p$, end_t_p and $class_p$; representing the predicted start time, predicted end time and the predicted class of the detected keyword respectively. The decision-making strategy in Section 4.2 defined by Equations 3 and 4 guarantees no overlapping between intervals in $Inv_{predict}$. FIGURE 6 illustrate model output for a waveform containing two keywords; showing predicted intervals (green) and

ground-truth intervals (red). A valid detection is registered when: (1) predicted and ground-truth intervals of the same keyword intersect, and (2) intersection duration exceeds threshold λ. Non-intersecting predictions are classified as false alarm or miss.

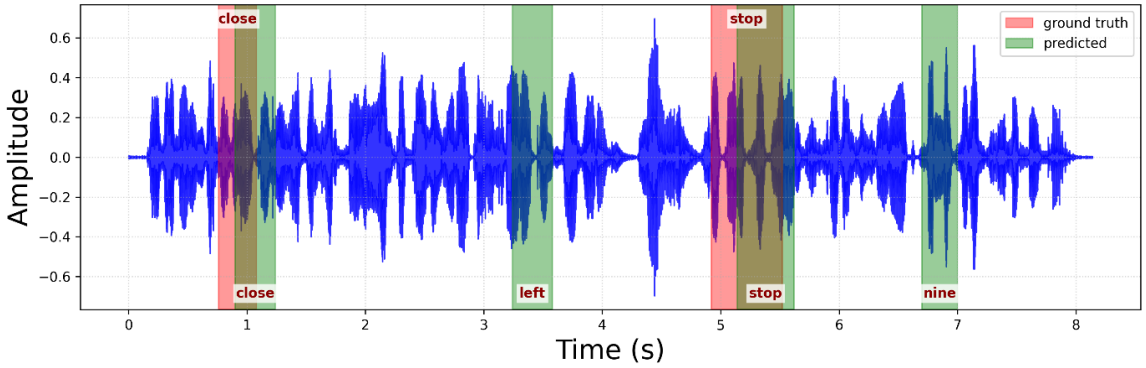

Figure 6: Example of a waveform containing two keywords: “close” and “stop”, with their ground truth intervals (in red) and the predicted intervals (in green). The overlapped regions represent the potential true detection (close, stop), while non-overlapped regions represent a false alarm (left, nine)

5. PERFORMANCE EVALUATION

This section presents the evaluation metrics employed to assess the model from a detection perspective, along with the construction methodology of the evaluation dataset.

5.1 Evaluation Metrics

The proposed architecture was evaluated using both standard metrics (Precision, Recall, F1-score) and advanced measures: Maximum Term Weighted Value (MTWV) [30] and mean Average Precision (mAP) [31]. Both metrics are limited to a maximum value of 1, with higher values indicating better performance, and are computed from hit/miss/false alarm rates (Equation 5). Here, “term” denotes a single keyword in the predefined set and refers to the detection threshold.

$$P_{hit}\left(term,\right) = \frac{N_{correct}\left(term,\right)}{N_{true}\left(term\right)}, \quad P_{miss} = 1 - P_{hit}, \quad P_{fa} = \frac{N_{fa}\left(term,\right)}{N_{NT}\left(term\right)}$$

where:

$N_{correct}$: represents the number of correct detections of a term by the system.

N_{true}: represents the total number of occurrences of a term in the corpus.

N_{fa}: is the number of detections of a term considered as true by the system while they are not.

N_{NT}: (number of non-targets) represents the number of opportunities for incorrect detection of term in the corpus.

In alignment with the definition of N_{NT}, it was calculated using Equation (5):

$$N_{NT}\,(term) = \frac{total\ time\ of\ the\ corpus}{mean\,(term\ durration)} - N_{true}\,(term) \tag{5}$$

For evaluation, audio clips containing target keywords (with ground-truth intervals (Inv_{ground_truth}), characterized by: $start\_t_{gt}$, end_t_{gt} and $class_{gt}$) were processed by the model to generate predicted intervals. For a predicted interval I_p and a ground-truth interval I_{gt} of the same term/class, the Intersection over Union (IoU) is defined as:

$$IoU_{I_p,I_{gt}}\,(term/class) = \frac{I_p \cap I_{gt}}{I_p \cup I_{gt}} \tag{6}$$

Based on the definition of Intervals and IoU in Equation 7, the hit, false alarm and miss rules can be defined as follows:

- H**it (True Positive: TP)**: for a given predicted interval $I_p \in Inv_{predict}$, a ground truth interval $I_{gt} \in Inv_{ground\_truth}$ where $class_p = class_{gt}$ and $IoU_{I_p,I_{gt}}\,(term/class) \geq$ was found.
- **False Alarm (False Positive: FA/FP)**: if there is a predicted interval $I_p \in Inv_{predict}$ and no ground truth interval $I_{gt} \in Inv_{ground\_truth}$ whose $class_p = class_{gt}$ and $IoU_{I_p,I_{gt}}\,(term/class) \geq$ was found.
- **Miss(False Negative: FN)**: if there is a ground truth interval $I_{gt} \in Inv_{ground\_truth}$ and no predicted interval $I_p \in Inv_{predict}$ where $class_p = class_{gt}$ and $IoU_{I_p,I_{gt}}\,(term/class) \geq$ was found.

5.1.1 The Term Weighted Value (TWV)

Term Weighted Value (TWV) measures system utility from a user perspective, decreasing with missed detections or false alarms. It is defined as:

$$TWV = 1 - \underset{term}{average}\left\{P_{miss} + \beta P_{fa}\right\} \tag{7}$$

Where:

$$\beta = \frac{c}{v}\left(P_{term}^{-1} - 1\right) \tag{8}$$

The Maximum TWV represents the TWV at the value of that yields the highest TWV. In the experimental results, c/v was set to 0.1, indicating that the value lost due to one miss is ten times the value lost due to one false alarm. For the evaluation dataset, P_{term} was calculated as the average of the occurrence probability of all keywords, thus, $P_{term} = 5.4 \times 10^{-3}$ and then β =18.4.

5.1.2 The mean Average Precision (mAP)

The mean Average Precision (mAP) measures the effectiveness of a system in retrieving and ranking relevant results. As previously mentioned, the system outputs predicted intervals $Inv_{predict}$. To

compute the mAP, the outcomes of the system should be ranked. Therefore, the predicted intervals are scored using the peak of the posterior probability function within each interval:

$$score_{intertval}\left(term\right) = Max\left(f_{c_{term}}\left(i\right) : i \in interval\right) \tag{9}$$

The calculation of Average Precision (AP) relies on Precision at the k^{th} rank ($P@k$), which measures precision at a specific rank k in the ranked list of results, focusing on how many relevant items appear in the top k result:

$$P@k = \frac{Number\ of\ relevent\ results\left(TP\right)\ \in\ the\ top\ k\ results}{k} \tag{10}$$

Following the computation of detection scores for predicted intervals using Equation 9 and their subsequent ranking in descending order, AP is derived as the mean of precision values at the ranks where relevant results appear. The formal expression is mathematically given by:

$$AP = \frac{1}{N_{relevant}} \sum_{k=1}^{n} P@k.rel\left(k\right) \tag{11}$$

Where:

$N_{relevant}$: Total number of relevant items in the corpus.

n: Total number of discovered results by the system (TP + FA).

$rel\left(k\right)$: Indicates whether the item at rank k is relevant (1 if relevant, 0 otherwise).

For a set of keywords $Q = \{kw_1, kw_2, kw_3, \ldots\ldots, kw_N\}$, the mAP is calculated as the mean of average precisions of each keyword in Q as given in Equation 12:

$$mAP = \frac{1}{Q} \sum_{\forall term \in Q} AP_{term} \tag{12}$$

5.1.3 Conventional Metrics

When KWS is treated as a classification problem, metrics like Precision, Recall and F1-score are usually used to evaluate the model. They are calculated based on TP, FP and FN using the following formulas:

$$Precision = \frac{TP}{TP + FA} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{15}$$

These metrics assume equal FP/FN costs and suit simple, balanced cases. Advanced alternatives address nuanced needs: mAP assesses ranking quality (critical for exhaustive retrieval), while TWV enables cost-sensitive optimization via β. Although these metrics are more informative, they require greater computation.

5.2 Evaluation Dataset

To evaluate the performance of the proposed method as a detection task on Arabic, a specialized dataset was constructed using the MGB-2 corpus. The dataset creation process involved a text-based search to locate sentences in MGB-2 containing the target keywords, followed by the extraction of corresponding audio segments from it. Since MGB-2 does not provide temporal information at word level, a word-level forced-alignment system was utilized to determine the temporal locations of each keyword within the sentences. This process was further validated through manual verification to ensure the accuracy of the keyword positions. As a result, the evaluation dataset comprised of audio clips, each containing one or more keywords from the predefined set, and their corresponding ground-truth intervals Inv_{ground_truth}. The key characteristics of this dataset are summarized in TABLE 1.

Table 1: Evaluation dataset characteristics.

Language	Total duration (seconds)	Number of keywords	Number of occurrences For all keywords	Number of audio clips
Arabic	6479	8	619	552

Although limited in size, the current dataset provides adequate evaluation capacity for assessing both keyword detection accuracy and temporal localization performance. Future enhancements should focus on: (1) lexical expansion through additional keywords, and (2) increased utterance diversity per keyword. Such enrichment would improve statistical reliability and potentially enable direct application in training CSKWS systems.

6. EXPERIMENTS AND RESULTS

This section presents the experimental results of the proposed CSKWS model. First, key model variables are identified for performance optimization, followed by evaluation results based on the adopted metrics.

6.1 Choosing Model Variables

As discussed in Section 4.2, the decision-making strategy relies primarily on two thresholds: $N_{threshold}$ and $P_{threshold}$. Prior to evaluating the model's performance, it was necessary to determine the optimal values for these thresholds. This was accomplished using the Detection Error Tradeoff (DET) curve, which plots the false negative (miss) rate (P_{miss}) against the false alarm rate (P_{fa}) for various values of ($N_{threshold}, P_{threshold}$). To do that, statistical sampling of the evaluation dataset was conducted, selecting 178 audio clips containing 200 total keyword occurrences (25 occurrences per keyword). Using the formulas in Equations 5 and 6, a comprehensive parameter sweep was conducted to evaluate system performance across the complete threshold space. The successive threshold ($N_{threshold}$) was varied across integer values [2–6], while the probability threshold ($P_{threshold}$) was evaluated at 11 discrete levels spanning from 65% to 90% (specifically:

65, 67, 70, 73, 75, 77, 80, 83, 85, 87, 90). For each of the 55 possible combinations, both P_{miss} and P_{fa} were computed. The complete performance characteristics derived from this exhaustive search are presented in FIGURE 7. Based on the DET curve, it was observed that the thresholds (3,87) yield the optimal performance, achieving both low false alarm rates and low miss rates. Consequently, $N_{threshold} = 3$ and $P_{threshold} = 87\%$ were chosen to be used in all subsequent experiments.

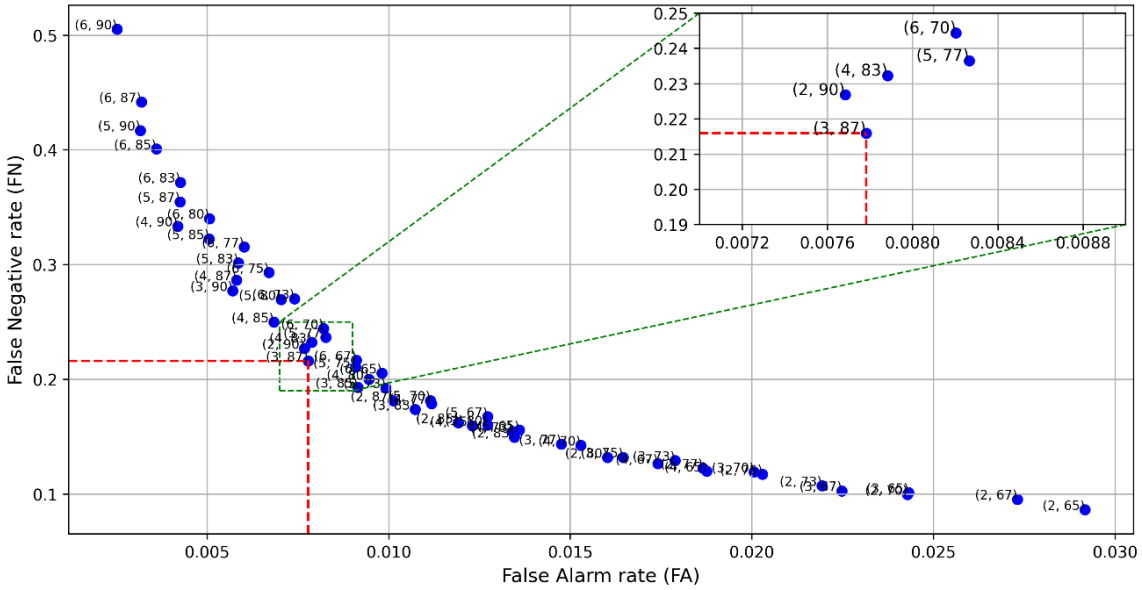

Figure 7: Detection Error Tradeoff (DET) curve showing False Negative (FN) rate versus False Alarm (FA) rate across threshold combinations ($N_{threshold}, P_{threshold}$). The inset highlights the optimal operating region (top-right) achieving balanced error minimization.

6.2 Selecting Sliding-Window Size

In continuous speech, word overlap can significantly degrade the performance of a KWS-base system primarily trained to identify isolated words. This degradation arises because features from adjacent words can affect the classifier's ability to recognize the target keyword. To investigate this phenomenon, various window lengths were tested and their impact on system performance was systematically analyzed using the mAP metric on the same subset of audio clips used in Section 6.1, FIGURE 8 illustrates the results.

The average duration of the terms in the evaluation dataset is 440 milliseconds. Analysis revealed that reducing the sliding window length to a specific extent enhances model performance, with optimal performance achieved at $L = 260$ milliseconds. This improvement can be attributed to the fact that a window size shorter than the average keyword length effectively eliminates the overlapping and potentially confusing segments at the beginning and ending of the word, thereby enhancing detection accuracy. Based on this finding, the window size was fixed at $L = 260$ milliseconds for all subsequent experiments.

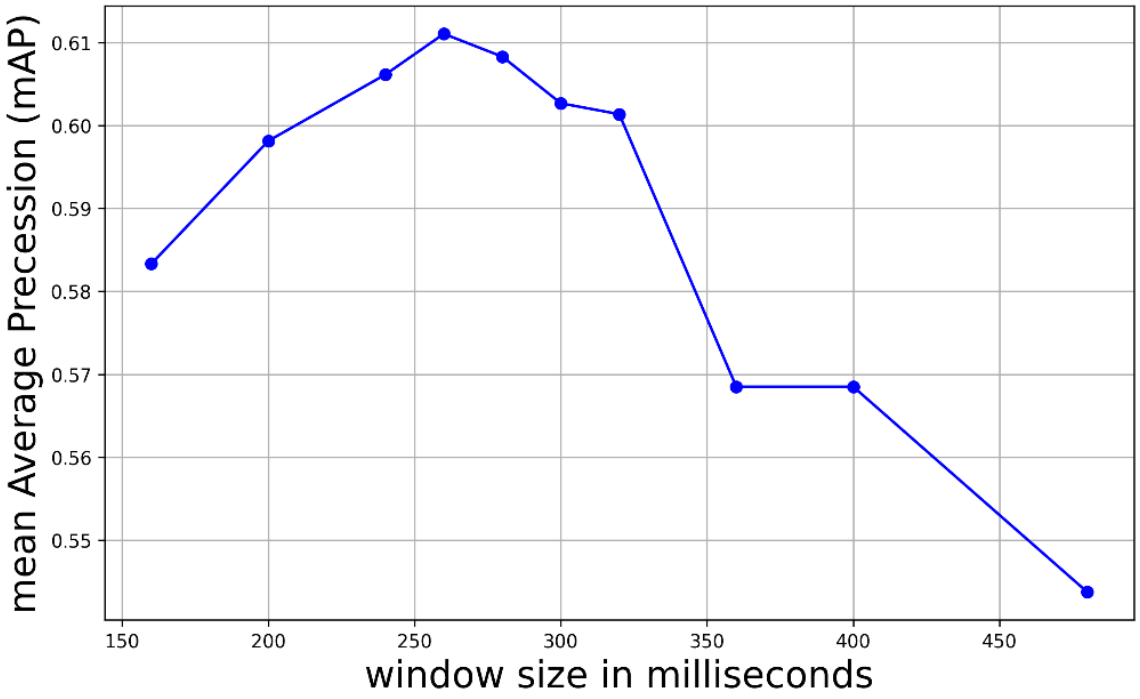

Figure 8: Window size optimization curve demonstrating maximal mAP at L = 260 milliseconds.

6.3 Results and Discussion

Following the identification of the model's optimal variables, its performance was evaluated on the evaluation dataset using the MTWV and mAP metrics in addition to conventional metrics like Precision, Recall and F1-Score. The proposed CSKWS resulted the predicted intervals ($Inv_{predict}$) based on the adopted decision-making strategy using Equations 3 and 4. Using the ground-truth intervals (Inv_{ground_truth}) accompanied with the evaluation dataset, the TP, FN and FP were determined using the rule defined in Section 5.1, then probabilities were calculated using Equations 5 and 6. MTWV was calculated using Equation 8, and mAP was calculated using Equations 10, 11 and 12. Conventional metrics were calculated using Equations 13, 14 and 15 utilizing the optimized parameters and thresholds derived from our earlier analysis in Section 6.1. TABLE 2 summarizes the results for different values of λ, which corresponds to the Intersection over Union (IoU) threshold. It is worth noting that the computation of mAP does not necessitate the specification of $P_{threshold}$, as it depends on ranking the results according to the probability of each interval, as outlined in Section 5.1.2.

Increasing λ leads to more predicted intervals failing to satisfy the hit rule, resulting in degradation in true positives and increment in false alarms. This ultimately causes performance degradation according to the adopted metrics as expressed in TABLE 2. The relatively high mAP value compared to Precision indicates that most true positive instances discovered by the system are ranked at the top, suggesting they have higher associated probabilities. This demonstrates that our model

Table 2: Performance evaluation of the PPF-CSKWS model across varying Intersection-over-Union (IoU) thresholds (λ = 0.1-0.75).

IoU threshold λ%	Number of occurrences	TP	FP	FN	Precision	Recall	F1	MTWV	mAP
10	**619**	**491**	**874**	**128**	**0.36**	**0.80**	**0.50**	**0.641**	**0.613**
25	619	474	891	145	0.35	0.77	0.48	0.614	0.600
40	619	391	974	228	0.29	0.63	0.39	0.459	0.494
50	619	309	1056	310	0.23	0.50	0.30	0.311	0.332
60	619	176	1189	443	0.13	0.28	0.18	0.080	0.178
75	619	65	1300	554	0.05	0.11	0.07	-0.110	0.027

effectively assigns higher probabilities when keywords genuinely appear in spoken phrases, which means higher ability to discover real occurrences of keywords.

Analysis of conventional metrics (Precision, Recall, F1-score) reveals suboptimal performance for the selected model variables, as evidenced by the observed high Recall (sensitivity) but low Precision (positive predictive value). To identify an optimal operating point for these metrics, the methodology from Section 6.1 was replicated to determine thresholds that balance precision and recall. This optimization yielded balanced performance (Precision = 0.52, Recall = 0.62, F1 = 0.56) at ($N_{threshold}$= 2, $P_{threshold}$= 95), though with an expected trade-off manifesting as reduced MTWV and mAP scores. These metrics are appropriate when framing the task as a classification problem, where only keyword occurrence (without temporal localization) requires detection. In such cases, hit/miss/false alarm criteria are defined without temporal constraints, necessitating different evaluation dataset construction. However, formulating continuous speech keyword spotting as a detection problem - evaluated through mAP and MTWV - more accurately reflects real-world performance requirements and provides superior model assessment.

These results demonstrate that the model successfully achieves the CSKWS task under the specified conditions, delivering acceptable mAP and MTWV rates using minimal number of training samples. The performance either matches or exceeds that of previous works [6, 24], while some other models demonstrate superior performance [5], comparative results are presented in TABLE 3. Evaluation results at λ = 10 were adopted as the model results based on operational requirements, where any non-zero temporal overlap between predicted and ground-truth intervals constitutes a valid detection event.

A direct comparison of our results with prior works may not be entirely appropriate due to fundamental differences in objectives. While our primary focus has been achieving CSKWS without requiring additional training data, most existing studies have prioritized performance optimization under data abundance conditions, enabling the use of supervised methods that typically yield superior results. Furthermore, the majority of these studies employed standardized datasets for both training and evaluation, whereas our evaluation dataset comprised real-world recordings containing natural acoustic variability that could potentially degrade performance metrics. These methodological differences emphasize the distinct challenges addressed by our work in low-resource scenarios.

Table 3: Performance comparison between the proposed method and baseline systems.

Model	Method	Evaluation Metrics	Values
DeepSpeech prototypical metric **[24]**	Supervised	Precision/Recall/F1	0.55/0.488/0.51
CLAD **[6]**	Supervised	Recall	0.691
AF-KWS **[5]**	Supervised	mAP	0.86
PPF-CSKWS @ λ = 10 %[a]	**Unsupervised**	**Precision/Recall/ F1/mAP/MTWV**	**0.36/0.80/0.50/0.613/0.641**
PPF-CSKWS @ λ = 10 %[b]	**Unsupervised**	**Precision/Recall/ F1/mAP/MTWV**	**0.52/0.62/0.56/0.611/0.632**

[a] *thresholds are optimized for MTWV.*
[b] *thresholds are optimized for Precision, Recall and F1-score.*

7. CONCLUSION AND FUTURE WORK

This study presented a CSKWS model specifically designed for low-resource language scenarios characterized by data scarcity. To minimize training data requirements, an unsupervised algorithm was developed that leverages an isolated-word KWS model—trainable with minimal audio samples—and extends its functionality to continuous speech through a sliding window approach. The model was evaluated using real-world Arabic dataset containing various types of acoustic interference. Experimental results demonstrated competitive performance compared to existing state-of-the-art English-language systems. By framing the task as a detection problem, this approach not only identifies keyword occurrences but also precisely determines their temporal locations and durations within continuous speech.

While the proposed model achieves competitive performance, its reliance on the HuBERT architecture introduces latency challenges that may hinder real-time deployment. In addition, the large number of parameters may not suit low-resource edge devices. Increasing the stride of the sliding window will contribute to speed up the model but at the expense of accuracy. To address these challenges, we plan to: (1) develop a distilled version of HuBERT to reduce model's parameters and inference time while preserving accuracy and, (2) test the optimized model on embedded hardware platforms which will enable its deployment on edge devices.

References

[1] Weinan D, Jiang Y, Liu Y, Chen J, Sun X, et al. Contrastive Augmentation: An Unsupervised Learning Approach for Keyword Spotting in Speech Technology. 2024. ArXiv preprint: https://arxiv.org/pdf/2409.00356

[2] Hu S, Liu H, Xu L, Wang J, Wang Y, et al. End-to-end speech keyword spotting system. In: International Conference on Cyber-Physical Social Intelligence (ICCSI). IEEE. 2023.

[3] Zheng Y, Shi X, Sathyanarayana A, Shokouhi N, Hansen JH. In-vehicle speech recognition and tutorial keywords spotting for novice drivers performance evaluation. In: IEEE Intelligent

Vehicles Symposium. IEEE. 2015:168-173.

[4] Shokri A, Tabibian S, Akbari A, Nasersharif B, Kabudian J. A robust keyword spotting system for Persian conversational telephone speech using feature and score normalization and ARMA filter. In: IEEE GCC Conference and Exhibition Dubai. IEEE. 2011:497-500.

[5] Zhao Z, Tang C, Yao C, Luo C. An anchor-free detector for continuous speech keyword spotting. Interspeech. 2022. ArXiv preprint: https://arxiv.org/pdf/2208.04622.

[6] Xi Y, Yang B, Li H, Guo J, Yu K. Contrastive learning with audio discrimination for customizable keyword spotting in continuous speech. In:IEEE international conference on acoustics speech and signal processing (ICASSP). IEEE. 2024:11666-11670.

[7] Szöke I, Petr S, Pavel M, Lukáš B, Martin K, et al. Phoneme based acoustics keyword spotting in informal continuous speech. In: International Conference on Text Speech and Dialogue. Springer. 2005;3658:302-309.

[8] Tejedor J, Wang D, Frankel J, King S, Colás J. A comparison of grapheme and phoneme-based units for Spanish spoken term detection. Speech Commun. 2008;50:980-991.

[9] Tabibian S, Shokri A, Akbari A, Nasersharif B. Performance evaluation for an HMM-based keyword spotter and a Large-margin based one in noisy environments. Procedia Comput Sci. 2011;3:1018-1022.

[10] Chen G, Parada C, Heigold G. Small-footprint keyword spotting using deep neural networks. In:IEEE international conference on acoustics speech and signal processing (ICASSP). Florence. IEEE. 2014:4087-4091.

[11] Sainath TN, Parada C. Convolutional neural networks for small-footprint keyword spotting. In: 16th Annual Conference of the International Speech Communication Association INTERSPEECH. ISCA. 2015.

[12] Arık SÖ, Kliegl M, Child R, Hestness J, Gibiansky A, et al. Convolutional Recurrent Neural Networks for Small-Footprint Keyword Spotting. 2017. ArXiv preprint: https://arxiv.org/pdf/1703.05390

[13] Seo D, Oh HS, Jung Y. Wav2KWS: transfer learning from speech representations for keyword spotting. IEEE Access. 2021;9:80682-80691.

[14] Warden P. Speech commands: A dataset for limited-vocabulary speech recognition. 2018. ArXiv preprint: https://arxiv.org/pdf/1804.03209

[15] Ghandoura A, Hjabo F, Al Dakkak O. Building and benchmarking an Arabic Speech Commands dataset for small-footprint keyword spotting. Eng Appl Artif Intell. 2021;102:104267.

[16] `https://www.researchgate.net/publication/394064554`

[17] Tabibian S, Akbari A, Nasersharif B. Discriminative keyword spotting using triphones information and N-best search. Inf Sci. 2018;423:157-171.

[18] Li S, Li G, Han J, Zhi T. Overview of speech keyword recognition technology. J Phys Conf S. 2021;1827:012013.

[19] Xi Y, Tan T, Zhang W, Yang B, Yu K. Text adaptive detection for customizable keyword spotting. In: IEEE international conference on acoustics speech and signal processing (ICASSP). Singapore. 2022:6652-6656.

[20] Vuppala AK. Open vocabulary keyword spotting through transfer learning from speech synthesis. 2024. ArXiv preprint: https://arxiv.org/pdf/2404.03914

[21] Jung Y, Lee S, Yang JY, Roh J, Han CW, et al. Relational Proxy Loss for Audio-Text based Keyword Spotting. In: Island K editor. Interspeech. ISCA. 2024:327-331.

[22] Zhu J, Yang C, Samir F, Islam J. The taste of IPA: towards open-vocabulary keyword spotting and forced alignment in any language. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. ACL. 2024:750-772.

[23] Sudhakar P, Sreenivasa R, Pabitra M. Query-by-example spoken term detection for zeroresource languages using heuristic search. ACM Trans Asian Low Resour Lang Inf Process. 2023.

[24] Seth H, Kumar P, Srivastava MM. Prototypical metric transfer learning for continuous speech keyword spotting with limited training data. InInternational Workshop on Soft Computing Models in Industrial and Environmental Applications. Cham: Springer International Publishing. 2019;273-280.

[25] Hannun A, Case C, Casper J, Catanzaro B, Diamos G, et al. Deep speech: Scaling up end-to-end speech recognition. 2014. ArXiv preprint: https://arxiv.org/pdf/1412.5567

[26] Panayotov V, Chen G, Povey D, Khudanpur S. Librispeech: an asr corpus based on public domain audio books. In: IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE. 2015:5206-5210..

[27] Kim B, Lee M, Lee J, Kim Y, Hwang K. Query-by-example on-device keyword spotting. In: IEEE automatic speech recognition and understanding workshop (ASRU). IEEE. 2019:532-538.

[28] Hsu WN, Bolte B, Tsai YH, Lakhotia K, Salakhutdinov R, et al. Hubert: self-supervised speech representation learning by masked prediction of hidden units. IEEE ACM Trans Aud Speech Lang Process. 2021;29:3451-3460.

[29] Ali A, Bell P, Glass J, Messaoui Y, Mubarak H, et al. The MGB-2 challenge: Arabic multi-dialect broadcast media recognition. In: IEEE Spoken Language Technology Workshop (SLT). IEEE. 2016.

[30] Fiscus JG, Ajot J, Garofolo JS, Doddingtion G. Results of the 2006 spoken term detection evaluation. In: Proceedings of the SIGIR. 2007;7:51-57.

[31] Zhang E, Zhang Y. Average precision. Encyclopedia of database systems. 2009;2009:192-193.