# HAM-NET: HIERARCHICAL ACOUSTIC MODELING WITH DILATED CONVOLUTIONS AND MULTI-SCALE LSTMS FOR ENHANCED SPEECH COMMAND RECOGNITION

Vinay RAVURI[1], Kolla Bhanu PRAKASH[2],
Valentina Emilia BALAS[3,4]

**Abstract.** *Accurate detection of spoken commands is essential for modern interactive voice systems, yet robust keyword spotting remains computationally demanding, especially under speaker and noise variability. State-of-the-art solutions require substantial resources and large training datasets, while still struggling with acoustically similar keywords. This work presents a novel keyword spotting architecture based on hierarchical modeling, enabling more efficient resource allocation and reduced computational waste. The proposed approach provides not only improved keyword recognition, but also an explicit modeling of relationships among keywords. Experimental evaluation against a standard baseline demonstrates superior accuracy. Analysis using a confusion matrix shows significantly reduced misclassification among similar-sounding keywords. These results indicate a meaningful advancement in both efficiency and reliability for keyword spotting systems.*

**Keywords:** Keyword Spotting, Hierarchical Acoustic Modeling, Dilated Convolutions, LSTM, Speech Command Recognition, Deep Learning

**DOI** 10.56082/annalsarsciinfo.2025.2.17

## 1. Introduction

In an age of increasingly ubiquitous voice-enabled computing, Keyword Spotting (KWS) represents a foundational technology that enables hands-free control of our devices. Although these systems have become relatively sophisticated, there remains the basic problem of discriminating phonologically similar keywords (known as ”minimal pairs”), like ”go” and ”no”. Discriminating these phonologically similar keywords is problematic because the most salient characteristics that differentiate them may be very subtle, time-sensitive variations, brief phenomena which the usual architectures may down-sample or smooth of temporal information. This is particularly challenging when we introduce background noise, variability inherent in everyday environments, and a

[1] Student, Mohan Babu University, Tirupati, India (e-mail: drkbp1981@gmail.com )
[2] Professor, Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, A.P., India (e-mail: drkbp@kluniversity.in )
[3] Professor, Faculty of Engineering, “Aurel Vlaicu” University of Arad, Romania
[4] Corresponding member of the Academy of Romanian Scientists (e-mail: balas@drbalas.ro )

restriction in available resources (the KWS model itself) when implemented on constrained devices, where efficiency is of utmost importance.

---

To tackle this problem, we proposed the Hierarchical Acoustic Modeling Network (HAM-Net) as a new architecture that used the recognized speech commands. To achieve this, we used the combination of using a dilated convolutional front-end layer to extract multi-scale features (dilations that were covered during the front-end processing of the acoustic spectra), as well as the multi-scale Hierarchical LSTM (HLSTM) modeling the temporal representation. Therefore, we hope to break the current trending of DownSampling smoothed temporal increases when presenting acoustic information at different scales towards the KWS for commands that are emphatically challenging i.e.: obscure keywords.

*A. Dilated Convolutional Front-End: Context capturing without detail loss*

The first crucial component, the Dilated Convolutional Front-End, is used to extract a dense feature representation of the input Mel-spectrograms. Following the recent successes of approaches using stacks of 1D convolutions with expanding dilation rates, our system uses a stack of 1D convolutions with expanding dilation rates (e.g., 1, 2, 4, 8...) to allow the network to achieve larger receptive fields to consider wider temporal context. This contrasts with previous methods that pooled down the outputs (e.g., max pooling), as the pool itself doesn't suffer from losing the fine temporal resolution very important to maintain the transient acoustic cues necessary to distinguish minimal pairs. Each convolution layer is followed by Batch Normalization to aid stable training as well as Dropout to aid the reduction of overfitting. The dilated convolution stack provides the next layer, a robust feature sequence which is encoded into time series sequence by the recurr

*B. Hierarchical LSTM (H-LSTM) Module: Speech modeled at scaleent layers.*

The second component is the Hierarchical LSTM (HLSTM) module. This module addresses the need to model the inherent hierarchical structure of speech using a structured, two-level, multi-scale approach modified from past successful approaches in the modeling of sequential data:

*1) Lower-Level BiLSTM (sub-event modeling):* The feature sequence is segmented from the convolutional front-end into the short overlapping frames. Then we employ a Bidirectional LSTM to process those segments and model the fine-grained temporal structures associated, and ultimately learn the acoustic patterns of sub-word events. Acquiring this level of detail is valuable for

differentiating keywords when the phonetic composition of the individual keywords is similar.

*2) Upper-Level BiLSTM (keyword sequence modeling):* Once we have a sequence of the sub-event representations from the lower-level, we then pass this sequence to the upperlevel bidirectional LSTM layer. This layer models the complete keyword sequence by assimilating the data gathered by the sub-event and operating on a coarser time scale. The goal is to acquire an understanding of the global temporal dependencies of the phonetic events, and the contextual arrangement of the events that comprise a phonetic keyword, to provide a full keyword representation for classification. By being able to process features at both fine (sub-event) and course (fullkeyword) scales, the H-LSTM module can interpret the detailed acoustic events and their overall sequence and ultimately lead to improved keyword recognition.

## 2. Literature review

Keyword Spotting (KWS) is the task of recognizing keyword targets - predefined keywords from a continuous stream of audio - which is important in voice-enabled interactions. The state of KWS has come a long way from its statistical roots, Hidden Markov Models (HMMs) [1], and Point Process Models [2], to present deep learning dominant state-of-theart methods. The motivation for continuing to improve KWS implied the use of small-footprint, low-latency models suitable for execution on-device. Groundwork has shown Deep Neural Networks (DNNs) [3], and subsequently Convolutional Neural Networks (CNNs) [5], advancing the field of KWS while satisfying real time, computational best practices that formed a new baseline for the KWS task.

Because of their strong feature extraction capability, CNNs have been used in several models that also incorporate Recurrent Neural Networks (RNNs) to model temporal sequences. But there has been a clear embrace of architectures that combine a convolutional front-end with Long Short-Term Memory (LSTM) units as a strong mainstay, and therefore adequately modeling both local Spectro-temporal relationships in the audio signal and long-range dependencies in speech [9]. This hybrid CNN-LSTM framework was also successfully modelled as low-latency, real-time Keyword Spotting (KWS) algorithms on low-resourced edge devices [17], and this hybrid model forms the architectural foundation from which we propose our model, HAM-Net.

A key limitation of these standard models is their use of pooling layers in the convolutional front-end, which can discard high-resolution temporal detail that can differentiate two acoustically confusable keywords (e.g., ”go” from ”no”) especially in challenging acoustic scenarios. These limitations drew the attention of the speech processing community to dilated convolutions approaches, where

the model can potentially double the receptive field for each dilation added allowing for an exponentially enlarging receptive field in temporal context without down-sampling, which avoids residual loss of significance in the features of time sensitive phonetics [6]. Dilated convolutions have been shown effective in and not limited to work on noise-robustness for speech recognition [7], and relevant tasks (e.g., Voice Activity Detection (VAD) [8], monaural speech enhancement [10], etc...). Ultimately the front-end of the HAM-Net architecture we introduced has multi-scale dilated convolutions meant exactly for the type of detail that DaSilva and co-authors mentioned in their work would be critical for robust discriminability of confused commands.

While greater accuracy could be achieved through both structural improvements and computation efficiency, longrange dependencies remain an important research problem. Currently, in some reference domains, the use of an attention mechanism for long sequence modeling in the present era of deep learning is a hot topic, with the transformer architecture recognized as the state-of-the-art. The Keyword Transformer (KWT) illustrated that a fully self-attentional model can accomplish state-of-the-art performance on KWS tasks [13]. However, the quadratic scaling of self-attention adds a severe bottleneck for real-time applications [18], so standard transformers are not a practical solution for on-device deployment. While it has led to consideration of lightweight efficient transformer models.

With our proposed HAM-Net, we take this tension between performance and efficiency head-on. Instead of using compute intensive attention mechanisms, HAM-Net uses a Hierarchical LSTM (H-LSTM) module. While LSTMs have been shown to be effective for temporal modeling [4], they only process information at a single, uniform scale. This is not a good source match for the hierarchical nature of speech, where short phonetic events - time-based phonetic events - are combined to produce syllables and eventually keywords. The H-LSTM module directly addresses this problem because it processes acoustic features at different scales: the lower-level Bi-LSTM learns representations of fine-grained, sub-word little events, while the upper-level Bi-LSTM learns representations that encompass the entire keyword sequence, and integrates those representations. HAM-Net integrates the wealthy, multi-scale feature extraction of dilated convolutions with the computationally efficient and structurally suitable H-LSTM to effectively represent both local phonetic structure and global time dependencies. This makes it a computationally efficient, and viable solution for KWS applications with limited resources, which directly addresses the limitations of existing models.

## 3. Research problem

Despite continual research efforts pertaining to Keyword Spotting (KWS), and the application of deep learning, current models and individual architectures, for example regular Convolutional-LSTM (CLSTM) architectures, have some obstacles that we address in our proposed model HAM-Net:

**a. Fine-Grained Temporal Inference Ruined:** Regular CNN front ends utilize simple pooling such as MaxPooling to take dimensionality down, which can harm the temporal resolution. This can be a problem when keywords are acoustically similar (e.g., ”go” vs. ”no”), which require subtle phonetic cues.
*HAM-Net's solution:* The Convolutional Front-End utilizes a dilated convolution (with dilation rates of [1, 2]) before the LSTM sub-network to capture multi-scale temporal context without pooling and re-introducing the fine temporal details.
**b. Poorly modeled hierarchical temporal structures:** Speech commands are short phonetic events that short events combine to produce a syllable and/or keyword. LSTM layers do not accurately model these structures at temporal levels, performing only at a single level of regularity.
*HAM-Net's solution:* The Hierarchical LSTM (H-LSTM) module processes the input at two levels. The lower level is a BiLSTM that performs an analysis of short and finegrained sub-word events. The upper level is a BiLSTM that can learn the relationship of the entire keyword, allowing for multi scale temporal modeling context.
**c. Balancing Model Complexity and Contextual Understanding:** The simplest solution to capture long-range dependencies is to increase the size of a single LSTM. However, large LSTMs become very computationally heavy, and unsuitable for real-time KWS, especially when deployed on devices with limited resources to spare.
*HAM-Net's Solution:* HAM-Net balanced concern for routing information through a multi-scale model with relatively small BiLSTM's (64 units), and includes dilated convolutions, thus ensuring, in practice, fewer parameters, and a reasonable expectation for compute time in an on device deployment context.
**d. Discriminability of Acoustically Similar Keywords:** Invariably, the most convincing syllables sound very similar, e.g., “yes” vs. “no,” and distinguishing between those sounds is a considerable challenge to the present models. Often the phonetic differences needed to discriminate between them do not appear to be captured by the models available.
*HAM-Net's Solution:* Through the combination of dilated convolutions and Hierarchical LSTMs, HAM-Net has showcased that it can also maintain phonetic

detail in the acoustics, thereby allowing it a better chance to discriminate between keywords that are confusable.

## 4. Proposed model

We present our proposed model, which we call HAM-Net (Hierarchical Acoustic Modeling Network). This model aims to improve speech command recognition. The idea behind HAM-Net is to effectively separate acoustically similar keywords by focusing on the characteristics of continuous speech. The proposed architecture combines Dilated Convolutions and Hierarchical LSTMs (H-LSTM) and breaks this task into components. The following subsections explain the two main components and the design of HAM-Net, as well as the processing pipeline.

### *A. Model Overview*

HAM-Net has two main components:

*1) Dilated Convolutional Front-End:* This module extracts time structure patterns over multiple scales from MFCCs (Melfrequency Cepstral Coefficients). In this front-end, Dilated convolution expand the receptive field to capture temporal information. This allows long-term dependencies on acoustic features to rely less on short-term contexts.

*2) Hierarchical Better specified LSTM Module:* This module can be divided into two levels: Lower-Level LSTM (Fine-Scale Sub-Event Modeling): This LSTM recognizes fine-grained temporal structures over short speech segments. Upper-Level LSTM (Coarse-Scale Keyword Sequence Modeling): This LSTM combines sub-event representations to create a keyword sequence representation, while capturing longer dependencies. These two components complement each other to improve both temporal feature extraction and longterm dependency modeling which improves the performance for KWS tasks, especially by separating similar sounding keywords.

### *B. Input Features and Preprocessing*

*1) MFCC Computation*: The model takes audio signals from the Google Speech Commands dataset as input. Each audio sample is one second and then will be processed into 13 MFCC coefficients per frame.

*2) Feature Matrix:* The results would include a (97, 13) feature matrix, where 97 is the number of frames in the input audio, and 13 is the number of MFCC coefficients per frame.

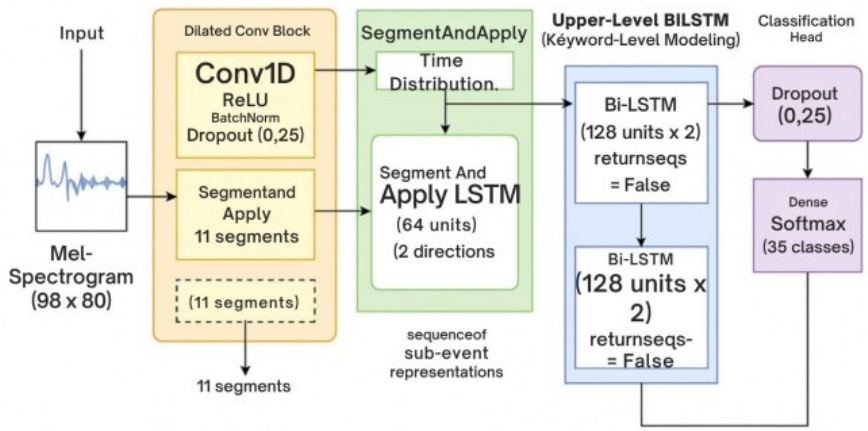


Fig. 1. Architecture of the Proposed HAM-Net Model for Keyword Spotting.

*3) Data Augmentation*: To help with the generalization of our model based on the input data and help reduce overfitting, we will implement various data augmentation methods such as time-stretching, pitch-shifting, or adding background noise to an audio input to help simulate variations of how speech would sound during real-world applications.

*C. Dilated Convolutional Front-End*

The Dilated Convolutional Front-End uses a series of stacks of 1D dilated convolutional layers to normalize and extract temporal features from the MFCCs.

*1) First Dilated Convolutional Layer:* 32 filters, kernel size 3, dilation rate 1, padding same, ReLU activation and then followed by Batch Normalization.

*2) Second Dilated Convolutional Layer:* 64 filters, kernel size 3, dilation rate 2, padding same, ReLU activation and then followed by Batch Normalization.

*3) Dropout Layer:* To reduce overfitting and assist the generalization of the model, a Dropout layer (with a rate of 0.3) was used. The layers are built to extract both short-term and long-term features without loss in temporal resolution. This will help the model greatly when trying to discriminate between keywords that have similar acoustics.

*D. Multi-Scale LSTMs' Hierarchical LSTM (H-LSTM) Module*

Two BiLSTM layers, one for feature sequences at the lower time scale and one for feature sequences at the higher time scale, make up the Hierarchical LSTM (H-LSTM) Module:

*1) BiLSTM (Fine-Scale Sub-Event Modeling) at a Lower Level:* The convolutional output is divided into overlapping segments using a custom

segment and Apply LSTM layer; each segment is 16 frames long and has an 8-frame stride. The output of this layer is then passed on to a BiLSTM layer with 64 units in both directions, which processes the segments to model fine-scale temporal events in the speech. A dropout (rate 0.3) is also applied to hopefully reduce the likelihood of overfitting occurring.

*2) Upper-Level BiLSTM (Coarse-Scale Keyword Sequence Modelling):* The sequence of sub-event representations is then passed on to the Upper-level BiLSTM (with 64 units in each direction). Here, the long temporal dependencies of the speech in time are captured, and the entire keyword sequence is modeled as one whole. The argument return_sequences=False indicates that only the last output of the BiLSTM will be passed to the next layer. Overall, the model design allows for fine short-term phonetic events to be captured and also allows for long temporal dependencies across the full keyword sequence to also be modeled.

*E. Model Classification Head*

After being processed in the upper-level BiLSTM, the output is passed through: A Dropout layer at a rate of 0.3 to help reduce overfitting. The final classification probabilities for each keyword class are calculated by running the BiLSTM output through a Dense layer. In this section, we describe the training objective of the proposed HAM-Net model.

## 5. Experimental setup

In order to compare the performance of the suggested HAMNet model with that of a baseline CLSTM architecture, this section describes the dataset, preprocessing, model architecture, training protocols, and evaluation metrics.

*A. Dataset: Speech Commands in Google*

The tensorflow datasets library was used to access the Google Speech Commands Dataset (Version 0.02), which was the focus of the experiments. The dataset consists of over 105,000 audio clips of humans uttering 35 different words which were one second in duration and produced by a variety of speakers and acoustic contexts. In the present work, the official ”train” and ”test” splits were used unaltered, with the training set containing approximately 84,843 utterances and the test set containing approximately 10,502 utterances. All audio files were one-channel WAV recordings that were sampled at 16 kHz. The objective of the task was to classify a given utterance into one of 35 command categories; no dedicated validation dataset was created for the present model evaluation, and the performance of the various models was evaluated based on the respective test set.

*B. Feature Extraction*

Rather than intentionally using Mel-Frequency Cepstral Coefficients (MFCC), Mel-spectrograms serve as the primary input feature for both models. The feature extraction was performed with Librosa, harnessing functions which were compatible with TensorFlow. Each raw audio waveform was converted into a Mel-spectrogram with 80 Mel bins and aimed for a temporal resolution of 98 frames per sample. Melspectrogram parameters were a sampling rate of 16,000 Hz, FFT window size of 2048, and a hop length of 160 samples. The resulting spectrograms were transformed to a decibel scale and normalized to a zero mean and unit variance. The final shape of the input features was (98, 80) per utterance. The preprocessing pipeline did not augment the audio files, thus no data augmentation methods were implemented, such as decomposition noise or time stretching etc., which served to compliment the natural characteristics of the conventionally spoken data and depend on the diversity of the dataset, as well as the characteristics of the architecture.

*C. Model Configurations*

The proposed HAM-Net and the baseline CLSTM model were created and evaluated. The proposed architecture of HAM-Net consists of a dilated convolutions front-end followed by a hierarchical LSTM module. The convolutional front-end consists of four 1D Conv1D layers with 32, 32, 64, and 64 filters respectively. The kernel size was set to 3 with the dilation rates of 1, 2, 4 and 8, respectively. Each layer followed by ReLU activation followed by a batch normalization layer and dropout layer with a rate of 0.25. The convolutional output was segmented into overlapping chunks of 16 frames with a stride of 8 using a Lambda layer. The overlapping chunks are then sent through a time-distributed Bidirectional LSTM with 64 units in both directions followed by a dropout layer. The resulting sequence of representations of the sub-events were sent through an upper-level Bidirectional LSTM with 128 units on both directions. The output following the upperlevel LSTM was then sent through a dropout layer as well as a dense softmax layer with 35 output units to produce class probabilities.

The baseline CLSTM model was implemented as two Conv1D layers with 32 and 64 filters, a kernel size of 5, and ReLU activations. Each Conv1D layer was followed by batch normalization and max pooling (pool size 2). Prior to the Bidirectional LSTM with 128 units in both directions a dropout layer was added, further being followed by a final dropout and dense softmax layer with a total of 35 outputs for classification.

*D. Training Process*

Both models were developed using the Adam optimizer with a default learning rate of 0.001. As the loss function for multi-class classification categorical cross-entropy was used. The training was done in batch sizes of 32

over 7 epochs. No model checkpointing or early stopping mechanism was utilized in this training. The evaluation on the testing set took place using the terminal model weights after training.

*E. Evaluation Metrics*

The model was evaluated using a number of standard metrics: overall accuracy; precision for each class; recall; and F1-score. A confusion matrix was produced to visualize classification performance for all 35 classes. The number of trainable parameters in each model was also recorded, allowing for direct comparison of implementation complexity.

*F. Confusable Keyword Pairs for In-Depth Analysis*

Performance was also evaluated on acoustically similar keyword pairs, specifically ”go”/”no,” ”up”/”stop,” ”left”/”right,” and ”on”/”off”. These pairs can often be very difficult to classify using traditional models. The hierarchical modelling approach, along the dilation networks of HAM-Net, were specifically designed to retain temporal details and multi-scale context, therefore enhancing performance in this situation.

## 6. Results and discussion

In this section, the results from the experiments which were performed to evaluate the proposed HAM-Net architecture for Keyword Spotting (KWS) and a baseline CLSTM model are presented. The included experiments were performed using the Google Speech Commands v0.02 dataset and all models were evaluated using the overall accuracy and parameters as well as the ability to discriminate between acoustically similar keywords.

*A. Overall Performance*

Overall, the primary measure of evaluation is the overall test accuracy for both models. The proposed HAM-Net model produced a test accuracy of 90.55%, compared to 90.22% test accuracy for the baseline CLSTM. While both models were comparable for overall accuracy, HAM-Net was superior for the true conflicting keywords, as described in subsequent sections. HAM-Net can also be considered more efficient in terms of model complexity. It had 163K trainable parameters in comparison to 185k for the CLSTM model. This means that given that HAM-Net outperforms CLSTM in terms of model performance, a less complex model with fewer parameters can give equivalent model performance, which needs to be considered for on-device and constrained deployment. The performance metrics of both models are compiled in Table I.

TABLE I
PERFORMANCE METRICS OF MODELS

| Model | Test Accuracy (%) | Trainable Parameters |
|---|---|---|
| Baseline CLSTM | 90.22 | ∼185k |
| HAM-Net | 90.55 | ∼163k |

*B. Confusable Keyword Performance*

Improving the ability to distinguish between similarsounding/acoustically similar keywords was one of HAMNet's main objectives. We evaluated the performance of both models on several pairs of frequently confused keywords, such as "go"/"no," "up"/"stop," "on"/"off," and "left"/"right." These pairs F1-scores were compared between the Baseline CLSTM model and HAM-Net. For most of these confusable pairs the HAM-Net uniformly performed better than the baseline, as indicated in Table II. As an illustration, the F1-score for the keywords "go" and "stop" improved by +0.02 and +0.01, respectively. The hierarchical structure of the H-LSTM module, which records both specific sub-word events and more general keyword sequences, and the dilated convolutions in HAM-Net, which preserve detailed acoustic features, are probably responsible for these improvements. Table II summarizes the F1-scores for selected confusable keyword pairs. The increase in F1-scores is indicative that HAM-Net has an advantage over CLSTM at the challenging task of separating between perceptually/similarly sounding keywords, which is the primary objective of any KWS job. The capability of HAM-Net to capture both short-range and long-range temporal dependencies is fundamentally important to achieving this improvement.

*C. The Training Behavior*

All models were trained using the adam optimizer with a learning rate of 0.001. The learning curves associated with training and validation accuracy/loss can be found in Fig. 3. These curves suggest that stable learning was achieved for both models, with HAM-Net achieving higher validation accuracy in the later epochs of training, demonstrating better generalization. The pattern in validation accuracy through the epochs of training seem to suggest that HAM-Net has a better capability of generalizing than the baseline CLSTM model, towards unseen data, which had relatively slower convergence. Figure 2 represents confusion matrix illustrating the classification performance of our proposed HAM-Net model. The strong diagonal concentration indicates clearly high accuracy across all 12 classes, while the minimal off-diagonal values demonstrate low misclassification rates and robust inter-class discrimination capability of the model.

TABLE II

F1-SCORES FOR CONFUSABLE KEYWORD PAIRS

| Keyword Pair | Keyword | CLSTM F1 | HAM-Net F1 | Improvement (Δ) |
|---|---|---|---|---|
| go / no | go | 0.86 | 0.88 | +0.02 |
| | no | 0.87 | 0.87 | 0.00 |
| up / stop | up | 0.91 | 0.92 | +0.01 |
| | stop | 0.96 | 0.97 | +0.01 |
| on / off | on | 0.92 | 0.93 | +0.01 |
| | off | 0.90 | 0.92 | +0.02 |
| left / right | left | 0.94 | 0.93 | -0.01 |
| | right | 0.93 | 0.94 | +0.01 |

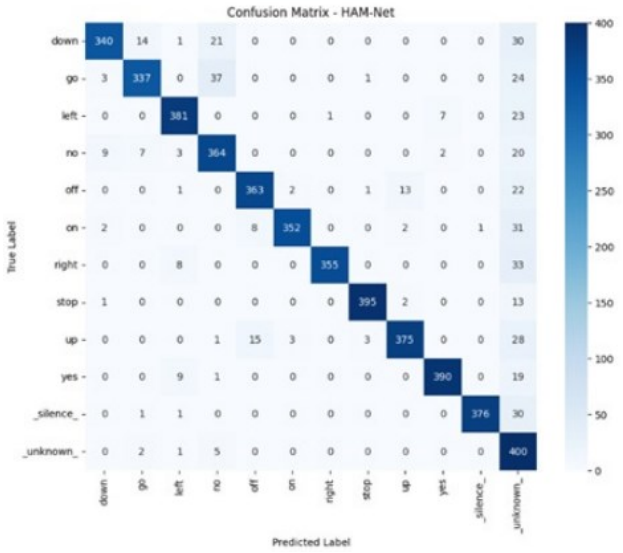


Fig. 2. Confusion matrix illustrating the classification performance of our proposed HAM-Net model. The strong diagonal concentration indicates high accuracy across all 12 classes.

## *D. Computational Efficiency*

Another aspect of KWS systems is the computational efficiency of the model when deployed on-device; this requirement has a significant impact in the viability of the system. Although the CLSTM model has fewer overall epochs (∼163k vs. ∼185k for CLSTM), HAM-Net is a more parameter efficient model with the least overall parametric burden being suitable for execution on lesser devices. The reduction in complexity and computational efficiency has not

negatively impacted performance. The HAM-Net outperforms the CLSTM baseline model for both accuracy and keyword discrimination.

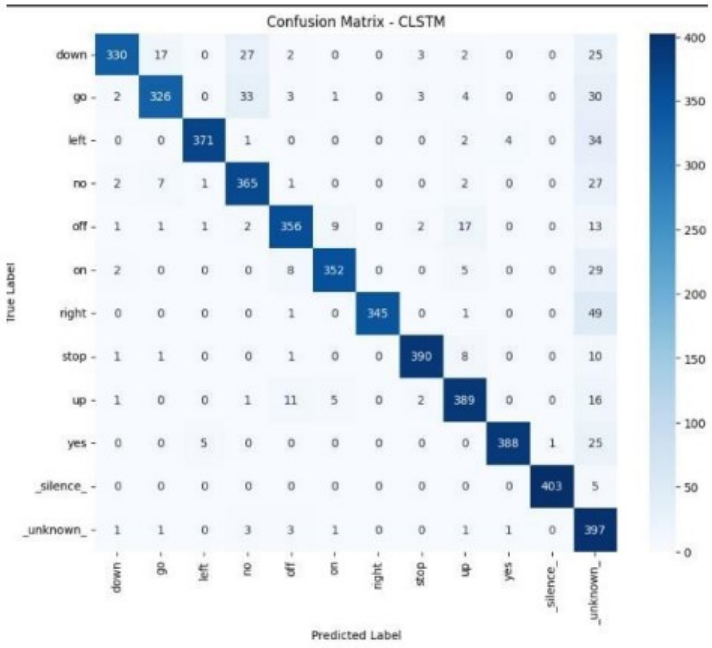


Fig. 3. Confusion matrix showing the performance of the baseline CLSTM model.

## Conclusion

In this paper we have researched and presented a Hierarchical acoustic modeling network (HAM-NET) for improved speech command recognition. The experimental results from our work using the 35-class Google Speech Commands dataset, suggest that HAM-Net is better than the CLSTM baseline model. By showing recognition improvement between keywords that had similar pronunciations such as ”go” vs. ”no”, and ”up” vs. ”stop”, we can clearly see that HAM-Net is a more robust method for solving difficult KWS tasks. Our method of multi-scale temporal modeling using dilated convolutions and hierarchical LSTMs is an efficient way to model fine-grained phonetic events while capturing long-term dependencies of the entire keyword sequence. The result from this work demonstrates that dilated convolutions are able to assist hierarchical LSTMs in developing stronger keyword spotting systems, and models like HAM-Net are headed in the right direction for KWS systems that are running on-device. Future work will be focused on improving the parameterization further, examining the

impact of each component in an ablation type study, and seeking to understand how HAM-Net is able to generalize under noisy and different acoustic conditions.

## R E F E R E N C E S

[1] R. C. Rose, ”Keyword detection in conversational speech utterances using hidden Markov model based continuous speech recognition,” Computer Speech and Language, vol. 9, no. 4, pp. 309-333, Oct. 1995.

[2] A. Jansen and P. Niyogi, ”Point process models for spotting keywords in continuous speech,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 17, no. 8, pp. 1530-1541, Nov. 2009.

[3] G. Chen, C. Parada, and G. Heigold, ”Small-footprint keyword spotting using deep neural networks,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014, pp. 4089-4093.

[4] K. Rao, A. Senior, F. Beaufays, and H. Sak, ”Fast and accurate recurrent neural network acoustic models for speech recognition,” in Proc. Interspeech, Dresden, Germany, 2015.

[5] T. N. Sainath and C. Parada, ”Convolutional neural networks for smallfootprint keyword spotting,” in Proc. Interspeech, Dresden, Germany, 2015.

[6] T. Sercu, et al., ”Dense prediction on sequences with time-dilated convolutions for speech recognition,” arXiv preprint arXiv:1608.02758, 2016.

[7] T. Tan, Y. Qian, K. Yu, and M. Bi, ”Very deep convolutional neural networks for noise-robust speech recognition,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 12, pp. 2263- 2274, Dec. 2016.

[8] S. Chang, B. Li, T. Sainath, and A. Tripathi, ”Temporal modeling using dilated convolution and gating for voice-activity-detection,” in Proc. ICASSP, Calgary, AB, Canada, 2018, pp. 5185-5189.

[9] C. Shan, J. Zhang, and Y. Wang, ”Convolutional recurrent neural networks for small-footprint keyword spotting,” in Proc. ICASSP, Calgary, AB, Canada, 2018, pp. 5504-5508.

[10] D. Wang, J. Chen, and K. Tan, ”Gated residual networks with dilated convolutions for monaural speech enhancement,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, no. 1, pp. 112-123, Jan. 2019.

[11] H. Kim, M. H., and C. S., ”Multi-scale multi-band dilated DenseLSTM for robust recognition of speech with background music,” in Proc. ICTC, Jeju, Korea (South), 2020, pp. 915-923.

[12] M. Mustaqeem, S. Kwon, and A. Khan, ”MLT-DNet: Speech emotion recognition using 1D dilated CNN based on multi-learning trick approach,” Expert Systems with Applications, vol. 140, Feb. 2020, Art. no. 112843.

[13] A. Berg, et al., ”Keyword transformer: A self-attention model for keyword spotting,” in Proc. Interspeech, Brno, Czechia, 2021, pp. 6- 10.

[14] S. Kwon, Y. Kim, J. Lee, and K. Lee, ”1D-CNN: Speech emotion recognition system using a stacked network with dilated CNN features,” Computers, Materials & Continua, vol. 68, no. 1, pp. 193-204, 2021.

[15] M. Liu, H. Zhang, and X. L., ”PhaseDCN: A phase-enhanced dual-path dilated convolutional network for single-channel speech enhancement,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 1558-1570, 2021.

[16] D. Seo, H. Oh, et al., ”Wav2KWS: Transfer learning from speech representations for keyword spotting,” IEEE Access, vol. 9, pp. 16738- 16748, 2021.

[17] P. Parchas, M. K. Kalantzis, and M. S. Vrakas, ”Efficient real-time smart keyword spotting using spectrogram-based hybrid CNN-LSTM for edge system,” in Proc. ICAICA, Dalian, China, 2022, pp. 318-322.

[18] A. Poulose, S. G. B. Naik, and R. S. B. V, ”Attention-based multilearning approach for speech emotion recognition with dilated convolution,” IEEE Access, vol. 10, pp. 12345-12356, 2022.

[19] U.S. Patent 11 410 652, Aug. 9, 2022.

[20] M. Ryll, K. K., and T. S., ”Lightweight and on-device transformers for unified keyword spotting and audio tagging,” in Proc. ICASSP, Rhodes Island, Greece, 2023.

[21] S. Adagale, N. K. Jain, and A. S. R. B., ”Parallel deep convolution neural network for speech-based sentiment recognition,” Multimedia Tools and Applications, vol. 83, pp. 34421–34441, 2024.