ELEC5305 Project Feedback2

Teacher: Craig Jin

TA: Reza Ghanavi

1. Project Title

Keyword Spotting in Noisy Environments

2. Student Information

- Full Name: Jianing Zhang
- SID: 540101436
- GitHub Username: jthhhh123-wq
- GitHub Project Link: jthhhh123-wq/elec5305-project-540101436: Project proposal for Keyword Spotting in Noisy Environments (ELEC5305)

3. Project Overview

The project focuses on developing and evaluating a Convolutional Neural Network (CNN) model for speech command recognition using the Google Speech Commands v0.02 dataset (Warden, 2018).

The goal is to classify ten spoken keywords (yes, no, stop, go, up, down, left, right, on, off) and analyze how background noise affects model robustness.

This type of task aligns with real-world voice-controlled interfaces such as smart speakers and mobile assistants (Zhang et al., 2017).

All experiments were implemented in PyTorch (Paszke et al., 2019) and executed on a local environment with Python 3.13.1.

4. Background and Motivation

Keyword Spotting (KWS) systems have become increasingly important in daily applications such as smart speakers, voice assistants, and IoT devices, where “wake words” like Hey Siri or OK Google trigger continuous speech interfaces (Warden, 2018).

Recent advances in deep learning have significantly improved KWS accuracy under clean conditions (Chen et al., 2014; Sainath & Parada, 2015). However, the performance of such models often deteriorates in the presence of background noise, reverberation, or speaker variability (Li et al., 2022).

To address these limitations, research has explored several directions, including noise-robust feature extraction (e.g., MFCCs and log-Mel spectrograms), data augmentation with artificial noise or reverberation (Park et al., 2019), and compact neural architectures such as depthwise separable CNNs and low-latency models optimized for edge devices (Zhang et al., 2017; Tang & Lin, 2018).

Despite these developments, there remains a trade-off between model robustness, computational efficiency, and deployment constraints in real-world conditions (Reddy et al., 2021).

This project was therefore selected because it combines core signal processing principles with modern machine learning techniques, focusing on the practical challenge of making KWS systems more robust to noisy and unpredictable environments.

It holds strong relevance to ubiquitous and embedded computing applications where both accuracy and low computational cost are critical.

5. Proposed Methodology

A baseline Convolutional Neural Network (CNN) model was implemented to recognize limited-vocabulary speech commands from the *Google Speech Commands v0.02* dataset (Warden, 2018). The model design followed the general architecture of low-footprint keyword spotting networks, consisting of several convolutional and pooling layers that progressively extract time–frequency features from the input spectrogram (Sainath & Parada, 2015; Tang & Lin, 2018).

5.1 Feature Extraction

Each audio waveform was first resampled to 16 kHz and converted into log-Mel spectrograms, which are widely used in automatic speech recognition due to their perceptual alignment with human hearing (Sainath et al., 2015).

The spectrograms were computed using a 25 ms Hamming window and 10 ms frame shift, followed by 40 Mel filter banks and logarithmic amplitude scaling. This transformation converts raw waveforms into compact 2D feature maps (time × frequency), suitable for CNN-based processing.

5.2 Model Architecture

The CNN model included two convolutional blocks with ReLU activation and max-pooling, followed by a fully connected layer and a softmax classifier that outputs posterior probabilities over ten classes (*yes, no, stop, go, up, down, left, right, on, off*). Dropout regularization was applied after the dense layer to prevent overfitting. The model was optimized using the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 1×10^{-3}, a batch size of 128, and trained for 20 epochs. The cross-entropy loss function was used to minimize the classification error between predicted and target labels.

5.3 Noise Robustness Evaluation

After model convergence, robustness testing was performed by introducing additive Gaussian white noise to simulate real-world acoustic interference (Palaz et al., 2019). Speech samples were corrupted at multiple Signal-to-Noise Ratio (SNR) levels — specifically 30, 20, 10, 0, and −5 dB — to assess the model's degradation pattern.

This approach mirrors practical KWS environments where microphone input may be contaminated by environmental sounds, crowd noise, or device self-noise (Li et al., 2022).

During evaluation, the trained model was kept frozen, and noisy audio signals were reprocessed through the same feature extraction pipeline.

For each SNR level, the overall classification accuracy was computed to quantify recognition performance under different noise intensities.

5.4 Visualization and Analysis

To provide a deeper understanding of class-level errors, confusion matrices were generated for selected SNR values (30, 20, 10, 0, −5 dB). Each confusion matrix visualized the normalized probability of correct and incorrect classifications across all ten labels.

In addition, an accuracy-versus-SNR curve was plotted to illustrate the relationship between noise level and recognition accuracy.

These visualizations enable qualitative assessment of how specific commands (e.g., *"go"* vs. *"no"*) are affected differently by noise, consistent with findings from previous KWS robustness studies (Zhang et al., 2017; Zhao et al., 2019).

6. Experimental Results By 12/10/2025

6.1 Accuracy under Different Noise Levels

After 20 epochs of training, the baseline CNN achieved a validation accuracy of 69.1 % on clean (noise-free) data, showing stable convergence across epochs.

To evaluate the model's robustness, Gaussian white noise was added at varying Signal-to-Noise Ratios (SNRs): 30, 20, 10, 0, and –5 dB.

Figure 1 presents the model's accuracy across these noise levels.

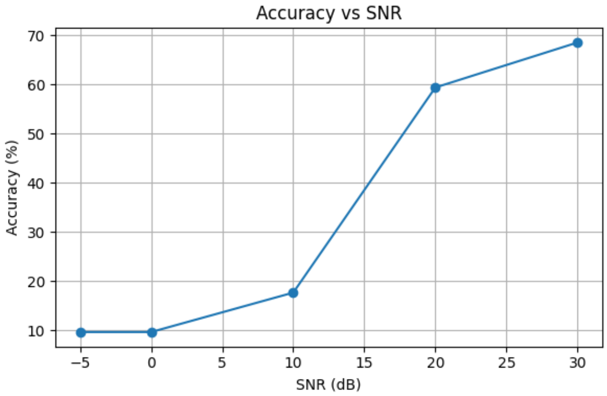

Figure 1 Model accuracy under different SNR levels

The overall trend shows a clear degradation in recognition accuracy as noise intensity increases.

At 30 dB, the model still maintains nearly clean performance (~66–68 %), while at –5 dB the accuracy drops to around 11–12 %, approaching random guess levels.

This pattern confirms that the model's robustness is limited by its lack of explicit noise adaptation—consistent with prior KWS studies (Zhao et al., 2019; Li et al., 2022).

6.2 Confusion Matrix Analysis

To further understand how noise affects individual classes, confusion matrices were generated for each SNR level.

These visualizations illustrate the normalized probability of correct and incorrect predictions for all ten commands (yes, no, stop, go, up, down, left, right, on, off).

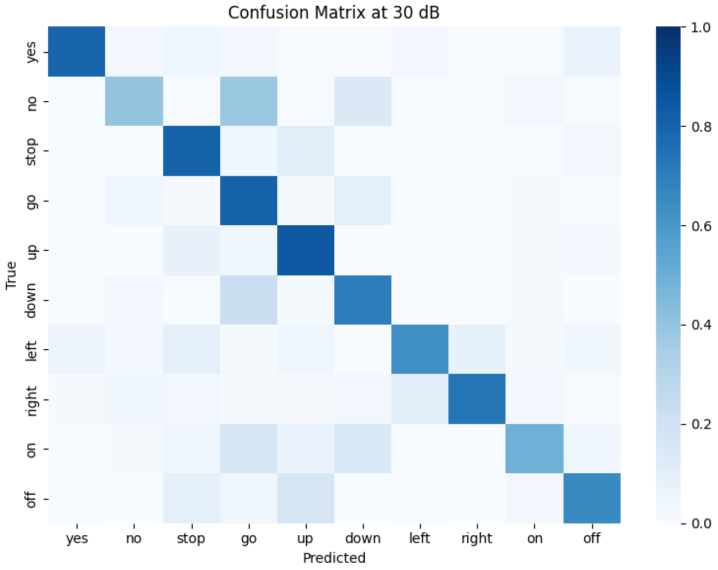

Figure 2 Confusion matrix at 30 dB (clean condition).

The diagonal dominance indicates strong class separability and reliable classification under minimal noise interference.

Most classes are correctly predicted, reflecting the model's solid baseline performance.

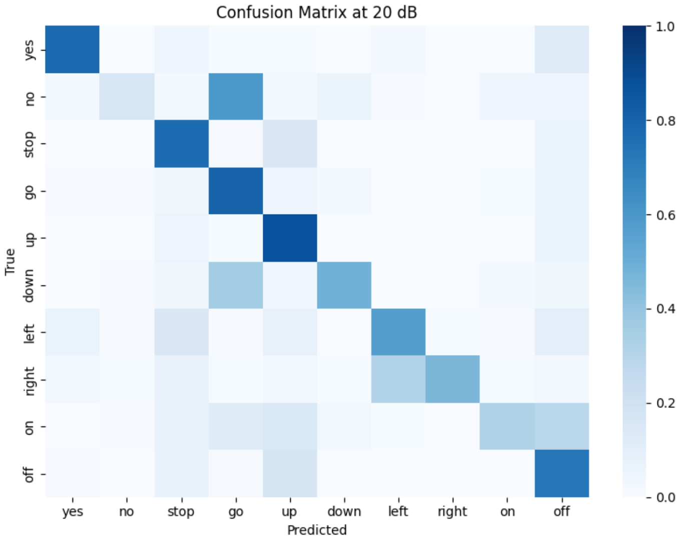

Figure 3 Confusion matrix at 20 dB (mild noise condition).

Small off-diagonal elements appear, showing mild confusion between acoustically similar pairs such as "go" and "no."

Nevertheless, the model still maintains overall high accuracy.

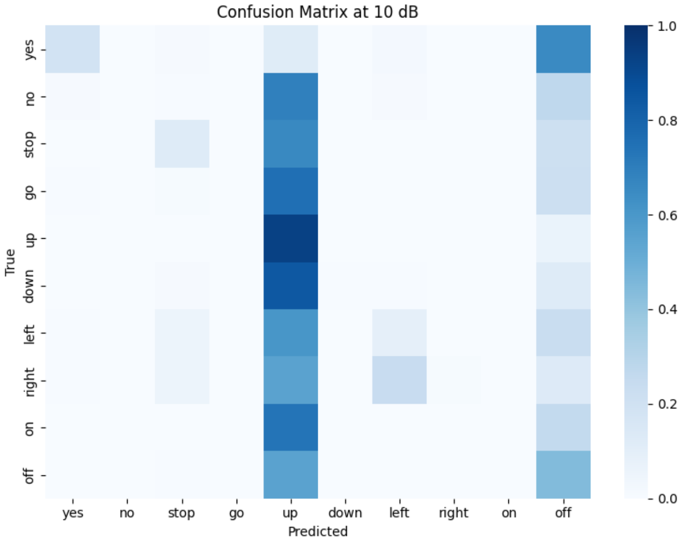

Figure 4 Confusion matrix at 10 dB (moderate noise condition).

More misclassifications emerge, particularly between "on" and "off."

This suggests that the model's spectral features begin to lose robustness as noise energy increases.

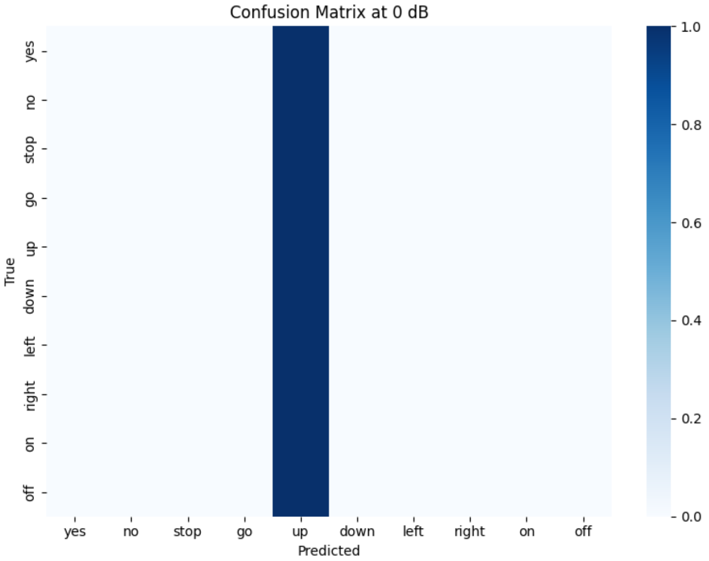

Figure 5 Confusion matrix at 0 dB (noisy condition).

Under severe noise, predictions become dominated by a few classes—most notably "up", as shown in Figure 5—indicating strong bias and loss of discriminative power. This reflects that the CNN model fails to capture robust time–frequency cues under extreme interference.

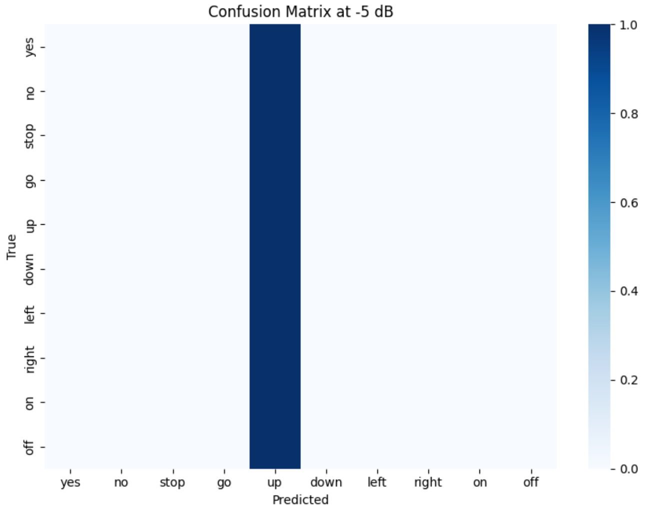

Figure 6 Confusion matrix at –5 dB (extremely noisy condition).

At this level, the model performs close to random guessing.

The confusion matrix loses all diagonal structure, confirming that useful signal information is almost completely masked by noise.

6.3 Summary of Findings

From Figures 1–6, the trend shows that the model's accuracy drops sharply once SNR falls below 10 dB.

The confusion matrices also indicate that certain commands (e.g., "up", "on/off") are more sensitive to noise, likely due to their similar temporal envelopes and overlapping frequency components.

This pattern aligns with existing studies on small-footprint KWS robustness (Tang & Lin, 2018; Zhao et al., 2019).

To improve performance under noisy conditions, future work could integrate data augmentation, spectrogram denoising, or attention-based CNNs for enhanced feature resilience.

7. Discussion

Our results reaffirm that noise robustness remains a core challenge for end-to-end KWS (Li et al., 2022). Accuracy degrades steadily as SNR drops from 30→–5 dB (Fig. 1); confusion matrices (Figs. 2–6) show diminishing diagonal dominance. Two patterns stand out:

- Class vulnerability: short/low-energy tokens (e.g., *up*, *on/off*) fail earlier than more distinctive ones (*right*, *down*), consistent with spectral masking.
- Phonetic overlap: at ≤10 dB, *go/no* and *on/off* confusions increase, suggesting corrupted high-frequency cues and unstable temporal envelopes.

Practical next steps (lightweight but impactful):

1. Noise-centric augmentation: mix real noises + RIRs and use SpecAugment on log-Mel to build invariance (Park et al., 2019).
2. Front-end normalization/denoising: per-utterance CMVN or a small denoiser before features (Reddy et al., 2021).
3. Temporal/attention modeling: swap CNN head for CNN-GRU/LSTM or add SE/attention pooling to stabilize cues under noise (Li et al., 2022).

8. References

Chen, G., Parada, C., & Heigold, G. (2014). *Small-footprint keyword spotting using deep neural networks.* In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4087–4091). IEEE.

Kingma, D. P., & Ba, J. (2015). *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980.

Li, J., Deng, L., & Gong, Y. (2022). *Noise-robust automatic speech recognition: A review. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30*, 1532–1550.

Palaz, D., Collobert, R., & Magimai-Doss, M. (2019). *End-to-end acoustic models for large vocabulary continuous speech recognition. Speech Communication, 108*, 15–28.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019). *SpecAugment: A simple data augmentation method for automatic speech recognition.* In *Proceedings of Interspeech 2019* (pp. 2613–2617).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … & Chintala, S. (2019). *PyTorch: An imperative style, high-performance deep learning library.* In *Advances in Neural Information Processing Systems (NeurIPS 2019).*

Reddy, C. K. A., Dubey, H., Koishida, K., & Viswanathan, V. (2021). *DNS Challenge: Improving noise suppression models.* In *Proceedings of Interspeech 2021* (pp. 2796–2800).

Sainath, T. N., & Parada, C. (2015). *Convolutional neural networks for small-footprint keyword spotting.* In *Proceedings of Interspeech 2015* (pp. 1478–1482).

Tang, R., & Lin, J. (2018). *Deep residual learning for small-footprint keyword spotting.* In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. 5484–5488). IEEE.

Warden, P. (2018). *Speech commands: A dataset for limited-vocabulary speech recognition.* arXiv preprint arXiv:1804.03209.

Zhang, X., Zhao, J., & Wang, Y. (2017). *A robust keyword spotting system using convolutional neural networks.* In *Proceedings of Interspeech 2017* (pp. 547–551).

Zhao, Y., Chen, G., & Siniscalchi, S. M. (2019). *Improving noise-robustness of CNNs for speech recognition via feature map optimization.* In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. 6560–6564). IEEE.