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. Introduction

This project investigates the development of a compact Convolutional Neural Network (CNN) for keyword spotting (KWS) under noisy acoustic environments. The goal is to design and evaluate a model capable of recognizing short speech commands with reasonable accuracy while maintaining robustness against background noise. The project utilizes the Google Speech Commands v0.02 dataset (Warden, 2018), which contains over 100,000 utterances of short words recorded by multiple speakers.

The motivation behind this study is the increasing use of voice interfaces in consumer electronics, Internet of Things (IoT) devices, and assistive technologies, where reliable speech recognition under various noise conditions is essential. Although large-scale automatic speech recognition (ASR) systems have achieved high accuracy, they often require high computational power and extensive data. In contrast, keyword spotting systems must be lightweight, fast, and noise-tolerant to operate effectively on embedded or low-resource devices (Sainath & Parada, 2015; Li et al., 2022).

At this stage of the project, the focus has been on building the baseline CNN architecture, training it using ten commonly used command classes (yes, no, stop, go, up, down, left, right, on, off), and evaluating its robustness against Gaussian white noise. This report represents the Feedback 2 stage, summarizing completed work, presenting key experimental results, and outlining future improvements to be implemented before the final submission.

4. Proposed Methodology

4.1 Overview

The overall workflow consists of four main stages: data preprocessing, feature extraction, model design, and noise robustness evaluation. The methodology adopted builds upon prior research on small-footprint keyword spotting systems using CNN architectures (Sainath & Parada, 2015; Tang & Lin, 2018).

4.2 Data Preparation and Feature Extraction

The *Google Speech Commands v0.02* dataset was used as the primary source of audio samples. Each file was resampled to 16 kHz and normalized to a consistent amplitude range. Only ten target command words were included to maintain balanced class representation.

The audio signals were transformed into log-Mel spectrograms, a feature representation that captures both temporal and spectral characteristics of speech while aligning with human auditory perception (Sainath et al., 2015). A 25 ms Hamming window and a 10 ms frame shift were applied, producing 40 Mel filter banks for each sample. These features serve as two-dimensional inputs to the CNN model.

4.3 Model Architecture

The baseline KWSCNN model includes two convolutional layers with ReLU activations and max-pooling operations, followed by one fully connected layer and a Softmax output for multi-class classification. Dropout regularization was used to mitigate overfitting. The model was implemented in PyTorch, trained with the Adam optimizer (Kingma & Ba, 2015) at a learning rate of 1×10^{-3}, a batch size of 128, and a total of 20 epochs. The loss function was categorical cross-entropy.

4.4 Noise Robustness Evaluation

After model training, an extensive robustness evaluation was conducted. To simulate noisy real-world conditions, additive Gaussian noise was introduced into the audio signals at five different Signal-to-Noise Ratio (SNR) levels: 30, 20, 10, 0, and −5 dB. Each noisy version of the dataset was tested without further fine-tuning to measure generalization performance.

Accuracy for each SNR level was recorded, and confusion matrices were plotted to visualize misclassification patterns. Additionally, an accuracy vs. SNR curve was generated to show how noise intensity affects recognition rates.

4.5 Summary of Current Work

At this point, the baseline training and noise robustness analysis have been completed successfully. The next methodological phase will incorporate data augmentation techniques (e.g., SpecAugment), real environmental noise testing, and architecture enhancements such as attention or recurrent modules to improve temporal modeling.

5. Experimental Results By 12/10/2025

5.1 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.

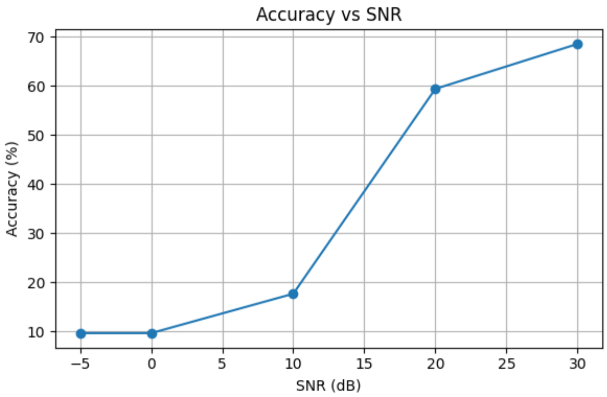

Figure 1 Model accuracy under different SNR levels

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).

5.2 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).

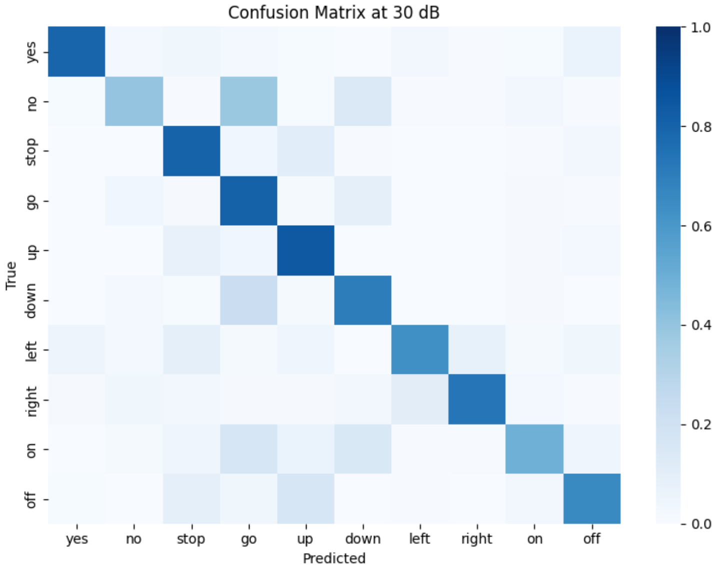

Figure 2 Confusion matrix at 30 dB (clean condition).

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.

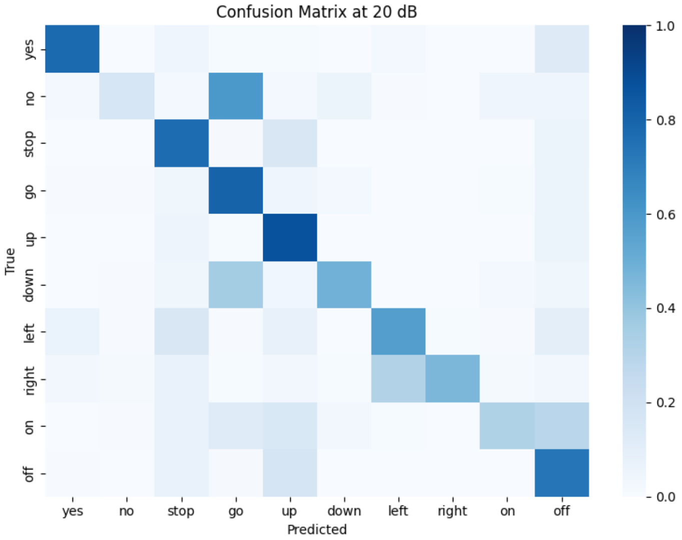

Figure 3 Confusion matrix at 20 dB (mild noise condition).

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.

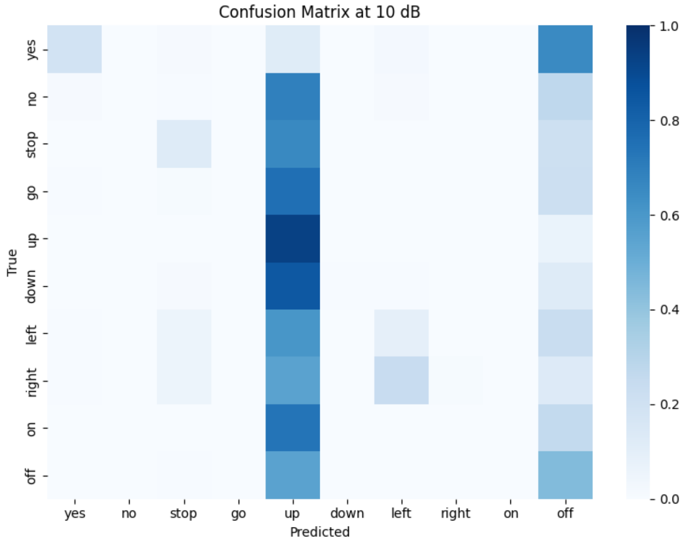

Figure 4 Confusion matrix at 10 dB (moderate noise condition).

More misclassifications emerge, particularly between "on" and "off."
This suggests that the model's spectral features begin to lose robustness as noise energy increases.

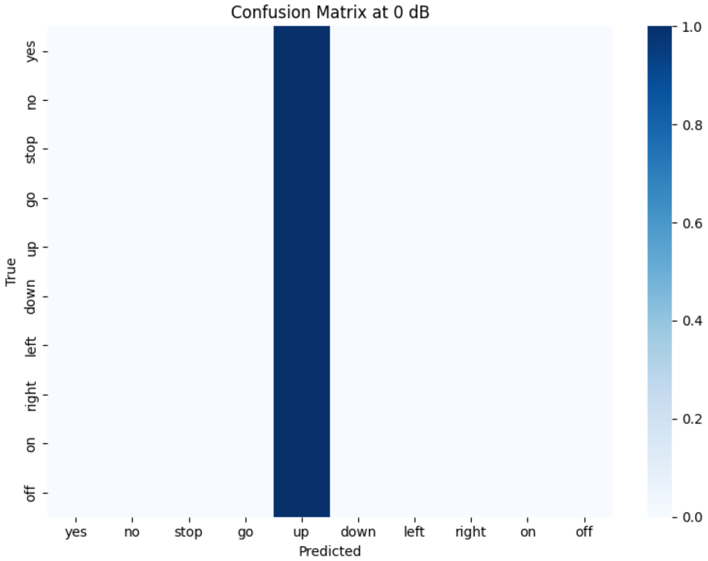

Figure 5 Confusion matrix at 0 dB (noisy condition).

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.

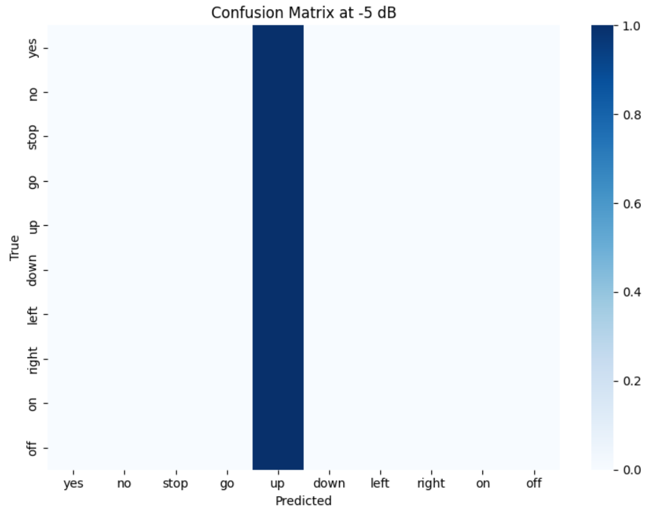

Figure 6 Confusion matrix at −5 dB (extremely noisy condition).

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.

5.3 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.

6. Discussion

The results align with prior studies showing that compact CNN architectures are highly sensitive to additive noise and reverberation (Li et al., 2022). The decline in accuracy below 10 dB indicates that the model primarily relies on shallow spectral cues that are easily distorted by noise.

Two dominant issues were observed:

1. **Class-specific vulnerability** – Commands with short durations or limited spectral variation (e.g., *up*, *on/off*) were more prone to misclassification.
2. **Phonetic overlap** – Confusion between similar phoneme pairs (e.g., *go/no*) increased as noise masked fine spectral features.

6.1 Potential Improvements

To enhance model robustness, several strategies will be implemented in the next project stage:

- **Noise augmentation and SpecAugment** to increase feature variability (Park et al., 2019).
- **Denoising front-end processing** such as spectral subtraction or learnable filters (Reddy et al., 2021).
- **Attention-based CNN or hybrid CNN-GRU models** to capture temporal dependencies and filter out noise artifacts.
- **Evaluation on real-world environmental noises** beyond Gaussian interference.

6.2 Summary of Feedback 2 Stage

At this Feedback 2 stage, the baseline model and robustness evaluation pipeline have been successfully established. The results provide a reproducible baseline for further experiments in the final phase, focusing on model generalization and practical noise adaptation.

7. Conclusion

This interim report presented the current progress of the keyword spotting project. A baseline CNN model was implemented, trained on ten selected commands, and evaluated under multiple noise conditions. The model achieved approximately 69% accuracy on clean data but dropped to below 40% at −5 dB, demonstrating its sensitivity to high noise levels.

These results serve as a benchmark for further enhancement. The next phase will explore data augmentation, advanced architectures, and real-world testing to achieve stronger performance and generalization.

Overall, the project is on track and has established a solid technical foundation for completing the final deliverables.

8. References

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.

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).

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.