# *ASR Under Noise*: Exploring Robustness for Sundanese and Javanese

**Salsabila Zahirah Pranida**[*,1] **Muhammad Cendekia Airlangga**[*,1] **Rifo Ahmad Genadi**[*,1]
**Shady Shehata**[2]
[1] MBZUAI [2] University of Waterloo
{salsabila.pranida, muhammad.airlangga, rifo.genadi}@mbzuai.ac.ae
[*] Equal contribution

## Abstract

We investigate the robustness of Whisper-based automatic speech recognition (ASR) models for two major Indonesian regional languages: Javanese and Sundanese. While recent work has demonstrated strong ASR performance under clean conditions, their effectiveness in noisy environments remains unclear. To address this, we experiment with multiple training strategies, including synthetic noise augmentation and SpecAugment, and evaluate performance across a range of signal-to-noise ratios (SNRs). Our results show that noise-aware training substantially improves robustness, particularly for larger Whisper models. A detailed error analysis further reveals language-specific challenges, highlighting avenues for future improvements. Code is available at https://github.com/rifoagenadi/robust_jvsu_asr.

## 1 Introduction

Automatic Speech Recognition (ASR) systems have made remarkable progress in recent years, especially for high-resource languages like English. While modern ASR handles diverse accents (Rao and Sak, 2017) and noise (Seltzer et al., 2013) in high-resource languages, it remains unreliable for low-resource ones.

Indonesia, with 284M people and over 700 languages, is among the world's most linguistically diverse countries (Badan Pusat Statistik, 2025; Eberhard et al., 2025; PetaBahasa, 2019; BPS, 2024). Yet, both remain underrepresented in ASR research and resources.

These languages exhibit high dialectal variation and are spoken daily in uncontrolled, noisy settings, which makes them difficult for standard ASR models, which are mostly trained on Indo-European data (Sani et al., 2012). Figure 1 right illustrates how background noise severely degrades transcription quality, even with advanced models like Whisper. This demonstrates the vulnerability of current ASR systems to real-world acoustic challenges.

Amid the growing use of large-scale speech-language models, Whisper has emerged as a strong multilingual ASR system (Radford et al., 2023). Unlike prior models such as wav2vec 2.0 and XLS-R, Whisper demonstrates superior robustness and generalization, particularly in noisy and low-resource scenarios (Pratama and Amrullah, 2024; Shah et al., 2024). These strengths make Whisper an ideal foundation for exploring ASR robustness in Javanese and Sundanese.

In this work, we present the first systematic study of ASR robustness to noise in these languages using over 60 hours of training data. Our key takeaways are: (1) evaluating Whisper models across clean and noisy test conditions; (2) exploring training strategies like `SpecAugment` and noise-aware fine-tuning; (3) analyzing language-specific transcription errors; and (4) releasing our training and evaluation pipeline for reproducibility. This is the first work to benchmark ASR robustness to noise in these languages systematically.

## 2 Related Works

**ASR for Sundanese and Javanese** The `NusaASR` benchmark (Cahyawijaya et al., 2023) evaluates ASR models on Javanese and Sundanese primarily in zero-shot settings. While prior work has fine-tuned large models like XLS-R and Whisper (Ariseputra et al., 2024; Pratama and Amrullah, 2024), these efforts often rely on limited data and lack reproducibility. Moreover, they rarely address robustness under noisy conditions. In contrast, our work provides a more comprehensive evaluation by fine-tuning Whisper across both languages.

**Noise Robustness** Ensuring ASR robustness in noisy environments is a well-recognized challenge (Shah et al., 2024; Feng et al., 2021; Likhomanenko et al., 2020). Prior work addresses this through data augmentation techniques such as synthetic noise injection and room impulse re-



*Proceedings of the 9th Widening NLP Workshop*, pages 87–99

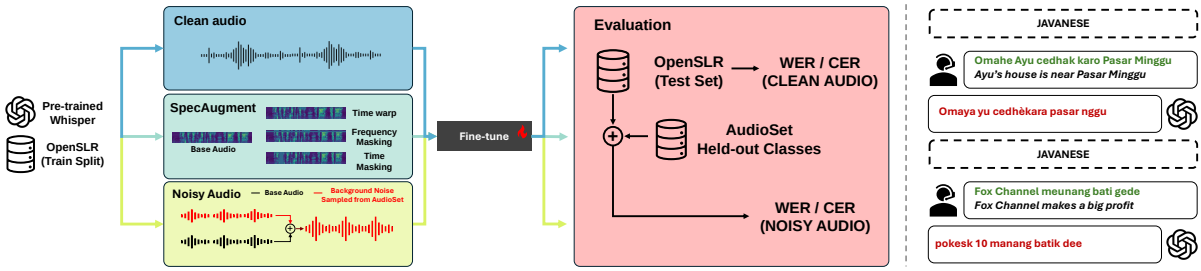



Figure 1: **(Left)** Training and evaluation pipeline for Whisper-based ASR models. Each fine-tuned model is evaluated on clean and noisy versions of the OpenSLR test set. **(Right)** Examples of noisy transcriptions in Javanese and Sundanese using Whisper. The top boxes show spoken utterances with noise; the bottom boxes show the corresponding ASR outputs, demonstrating significantly degraded quality under noisy conditions.

sponses. Among these, `SpecAugment` (Park et al., 2019) has gained popularity as a simple and effective method. Other approaches include noise-aware training (Orel and Varol, 2023) and denoising front-ends (Dissen et al., 2024). In our work, we independently evaluate `SpecAugment` and noise-aware fine-tuning, using noise samples from `AudioSet` (Gemmeke et al., 2017), as two distinct strategies to improve ASR robustness.

## 3 Experimental Setup

### 3.1 Linguistic Characteristics

**Javanese** Javanese has more than 80 million speakers (Eberhard et al., 2021) and is part of the Austronesian, Malayo Polynesian family (Cohn and Ravindranath, 2014). It is agglutinative with extensive affixation that produces many word forms and is commonly divided into Western, Central, and Eastern varieties, each with distinct phonology and vocabulary (Wedhawati et al., 2001). A notable feature is its speech levels, such as *ngoko* (informal) and *krama* (polite), which encode social hierarchy in interaction (Isodarus, 2020).

**Sundanese** Sundanese, spoken by about 30–40 million people in western Java (Eberhard et al., 2021), is part of the Austronesian, Malayo Polynesian family and shows agglutinative morphology with rich affixation. Major dialects include Bogor, Priangan, and Cirebon, which differ in vocabulary and pronunciation (Kurniawan, 2013). The language also encodes politeness through registers that guide lexical choice.

### 3.2 Dataset

**Data Overview** We use the OpenSLR Javanese and Sundanese corpora (Kjartansson et al., 2018), collected with support from Universitas Gadjah Mada in Yogyakarta and Universitas Pendidikan Indonesia in Bandung. The recordings are read speech from volunteers. These corpora are valuable but do not cover the full range of dialects or spontaneous use.

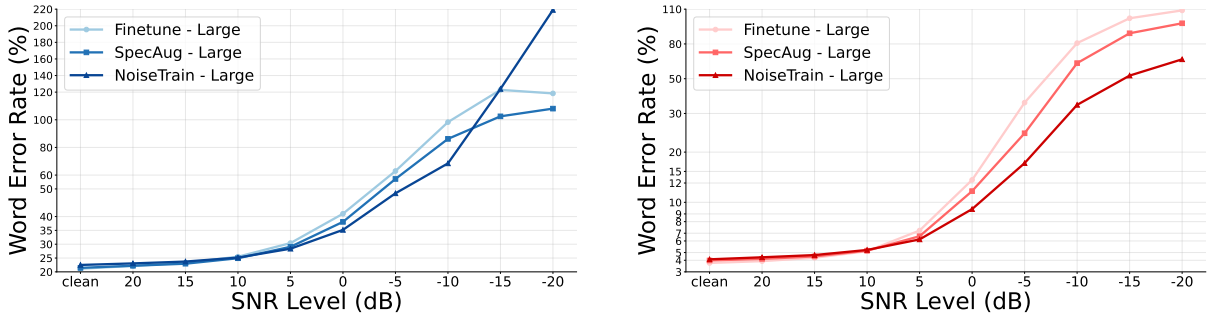
From the full releases (185k utterances / 296 hours for Javanese and 219k utterances / 333 hours for Sundanese), we selected 10 subsets for training and 6 for testing (Kjartansson et al., 2018). This gives about 60 hours of training data and 10 hours of test data per language, with train and test speakers kept separate (Table 1). The size is adequate for baseline ASR, but limited coverage should be considered when interpreting results. While we were unable to identify detailed dialectical or speaker variations from the original paper Kjartansson et al. (2018), we estimated the proportion of female and male speakers using a fine-tuned version of wav2vec (Baevski et al., 2020)*.

| Lang | Train | Test | #Speakers (F%) |
|---|---|---|---|
| JV | 37,439 | 6,276 | 758 (57%) |
| SU | 39,560 | 6,563 | 529 (57%) |

Table 1: Number of utterances and unique speakers for each language, with female speaker proportion.

**Synthetic Noise Data Generation** To simulate real-world conditions, we augment clean train-

*https://huggingface.co/prithivMLmods/Common-Voice-Gender-Detection




(a) Javanese

(b) Sundanese

Figure 2: WER performance of Large-v3 Whisper across different SNR levels for Javanese and Sundanese. Models trained with `NoiseTrain` consistently outperform others under low-SNR conditions. Higher SNR values indicate cleaner audio.

ing data with background noise at various Signal-to-Noise Ratio (SNR) levels, following prior work (Orel and Varol, 2023; Maas et al., 2012). The noise types reflect common environments like traffic and indoor chatter. Details on the noise selection, SNR values, and mixing procedure are provided in Appendix B.

### 3.3 Training Pipeline

We fine-tune four Whisper variants—`Tiny`, `Medium`, `Large-v3`, and `Large-v3-Turbo`—on Javanese and Sundanese ASR using OpenSLR. While these models support the languages, their zero-shot performance is poor due to limited training exposure. We explore three training strategies to improve robustness, as illustrated in Figure 1.

**Clean Fine-tuning** Models are trained on unmodified OpenSLR data as a baseline.

**Clean + SpecAugment** In this setup, we fine-tune the models by applying `SpecAugment` on clean data, a data augmentation method that applies time and frequency masking on input spectrograms. To tune augmentation hyperparameters, we use a 90/10 split of the training data for training and validation (see details in Appendix A).

**Fine-tune in Noisy Audio** We synthetically augment the training set by mixing clean OpenSLR utterances with background sounds from 24 classes in `AudioSet` (Gemmeke et al., 2017), at various SNR levels. Noise audio in the train splits is shuffled and mapped in a many-to-one manner to SNR values. It means that one SNR was used for different audio files, but the audio files did not repeat. The resulting noisy dataset is then used to fine-tune the Whisper models. This setup is referred to as `NoiseTrain`.

### 3.4 Evaluation Pipeline

Models are evaluated on both clean and synthetic noisy versions of the OpenSLR test set, as shown on the evaluation side of Figure 1, using word error rate (WER) as the main metric. Noisy test sets are created by mixing the clean utterances with background sounds from 8 held-out noise `AudioSet` classes[†].

## 4 Results and Analysis

### 4.1 Model Robustness

We evaluate Whisper models on Javanese and Sundanese under varying noise conditions. Figure 2 shows how WER changes across SNR conditions using the `Large-v3` model (see details in Appendix D), while Tables 2 and 3 report detailed results for all model variants and training strategies. Zero-shot performance is poor, with WERs exceeding 70–120 even on clean audio, confirming that adaptation is critical. We selected `SpecAugment` configuration **#9** as the best-performing setup (see Appendix A) and use it for all reported results. Both `NoiseTrain` and `SpecAugment` significantly improve robustness, especially under low-SNR conditions.

Models trained with `NoiseTrain` or `SpecAugment` consistently outperform clean-only models, especially under low-SNR conditions. For instance, in Javanese –SNR, `Medium` improves from 225.38 to 111.89 WER, and in Sundanese, from 199.09 to 56.15. Even larger models like `Large-v3` benefit, dropping from 79.91 to 41.37, showing the importance of noise-aware

[†]See Appendix E for the list of held-out noise classes.

training for real-world robustness. Running all experiments, including SpecAugment tuning, clean, and noise-aware fine-tuning, required over 240 GPU-hours.

We also the Large variant to be slightly better than Large-turbo. Whisper large-turbo is a fine-tuned of pruned whisper large. Thus, they are both the exact same model except the turbo variant have reduced number of decoding layers, from 32 to 4. The turbo model is optimized for faster inference with a minor degradation. Therefore, the result we have in Table 3 and Table 2 is expected since we fine-tune a larger number of parameters in the large variant.

| Model | Clean | Noisy | |
|---|---|---|---|
| | | +SNR | –SNR |
| **Tiny** | | | |
| Zero-shot | 128.56 | 170.65 | 205.89 |
| Clean | **60.42** | 77.60 | 133.53 |
| SpecAug + Clean | 60.99 | 78.41 | 133.59 |
| NoiseTrain | 65.09 | **76.10** | **106.51** |
| **Medium** | | | |
| Zero-shot | 92.08 | 105.33 | 152.42 |
| Clean | **25.40** | 33.85 | 225.38 |
| SpecAug + Clean | 25.45 | 32.79 | 140.05 |
| NoiseTrain | 26.87 | **32.41** | **111.89** |
| **Large-v3** | | | |
| Zero-shot | 74.62 | 82.66 | 148.12 |
| Clean | **21.14** | 28.47 | 100.76 |
| SpecAug + Clean | 21.45 | 27.45 | **88.48** |
| NoiseTrain | 22.50 | **27.10** | 114.95 |
| **Large-v3-Turbo** | | | |
| Zero-shot | 67.13 | 80.29 | 195.65 |
| Clean | 24.12 | 77.80 | **134.19** |
| SpecAug + Clean | **23.89** | 31.75 | 140.82 |
| NoiseTrain | 24.79 | **30.95** | 153.73 |

Table 2: WER on the Javanese test set across clean and noisy conditions. All models are fine-tuned on Javanese only. "+SNR" refers to high SNR and "–SNR" to low SNR. Zero-shot results are only evaluated on clean audio.

## 4.2 Error Analysis

We conduct error analysis on the best model, `Large-v3`, using two views. *First*, we use character error rate (CER) to quantify fine grained edits: extra spaces, vowel changes, consonant changes, and diacritics, which is appropriate for agglutinative languages where small affix or spacing differences can inflate word errors. *Second*, we use WER to summarize word insertions, deletions, and substitutions. Table 4 reports the CER-based error distribution for Javanese and Sundanese(see Appendix C).

| Model | Clean | Noisy | |
|---|---|---|---|
| | | +SNR | –SNR |
| **Tiny** | | | |
| Zero-shot | 116.79 | 194.18 | 360.48 |
| Clean | 40.37 | 68.50 | 413.56 |
| SpecAug + Clean | **40.19** | 61.64 | 274.32 |
| NoiseTrain | 43.82 | **58.89** | **201.79** |
| **Medium** | | | |
| Zero-shot | 83.20 | 93.06 | 282.98 |
| Clean | **4.03** | 8.43 | 199.09 |
| SpecAug + Clean | 4.09 | **7.84** | 165.36 |
| NoiseTrain | 5.46 | 8.59 | **56.15** |
| **Large-v3** | | | |
| Zero-shot | 78.90 | 83.62 | 171.76 |
| Clean | **3.72** | 6.60 | 79.91 |
| SpecAug + Clean | 3.98 | 6.24 | 67.59 |
| NoiseTrain | 4.10 | **5.88** | **41.37** |
| **Large-v3-Turbo** | | | |
| Zero-shot | 73.20 | 81.04 | 187.01 |
| Clean | **4.83** | 9.84 | 160.43 |
| SpecAug + Clean | 4.83 | 8.95 | 124.15 |
| NoiseTrain | 6.17 | **8.62** | **65.42** |

Table 3: WER on the Sundanese test set across clean and noisy conditions. All models are fine-tuned on Sundanese only. "+SNR" refers to high SNR and "–SNR" to low SNR. Zero-shot results are only evaluated on clean audio.

| Error Type | Cased | | Uncased | |
|---|---|---|---|---|
| | jav | sun | jav | sun |
| Additional Space | 900 | 338 | 918 | 351 |
| Consonant Mistake | 7702 | 2284 | 5815 | 1952 |
| Vowel Mistake | 3722 | 1214 | 3660 | 1236 |
| Diacritics Mistake | 1702 | 4 | 1680 | 4 |

Table 4: Distribution of different types of errors for Javanese (jav) and Sundanese (sun) language datasets.

**Additional Space** This error occurs when the model inserts or removes spaces incorrectly. In Javanese, examples include *dipunpanggihaken* becoming *dipun panggihaken*, or *adipati* split into *adi pati*. In Sundanese, errors often involve foreign names (e.g., *baekhyun* → *baek hyun*) or place names (e.g., *situ lengkong* → *situlengkong*). Common words like *minangka* were also occasionally split into *minang ka*.

**Vowel Mistakes** Vowel-related errors often arise from subtle phonetic variations and orthographic influences. In Sundanese, confusion among the three *e*-like vowels—e (as in lebak), è (bèbèk), and eu (teuas)—frequently leads to transcription mistakes, such as heulang being rendered as helang. Foreign names are also problematic when pronounced with

local phonology, e.g., Taylor pronounced as Tayler /[ˈtaj.ler]/. In Javanese, vowel shifts and reductions are common, with examples like permata becoming permato or terus shortened to trus, reflecting dialectal or colloquial speech that ASR models struggle to handle. Additionally, Dutch-influenced spellings, such as oe for /u/—, can cause errors like Doel being transcribed as Dul.

**Consonant Mistake** These were far more common in Javanese, probably because it has more complex consonant sounds, including digraphs like *dh*, *ng*, *ny*, and *th*, which are sometimes simplified or misheard. Some Javanese examples include *cetha* becoming *ceto*, *baut* as *baud*, *djoni* as *jani*, *aktris* as *apris*, and *putuku* written as *puduku*. In Sundanese, consonant errors were less frequent, but often appeared in borrowed or foreign words. For instance, some speakers pronounce *f* or *v* as *p*, resulting in words like *felton* → *pelton*, *pevita* → *fevita*, or *shidqia* → *shidgya*.

**Diacritics Mistake** Diacritic-related errors were mainly happen in Javanese. Javanese uses diacritics more extensively, especially marks like *é* and *è*, which affect pronunciation and meaning. These are known as *sandhangan swara*. We found examples like *dhèwèké* written as *dhaweke*, *radén* as *radenma*, *warnané* as *warnane*, and *saliyané* as *saliyane*. Additionally, we would like to note that data from OpenSLR in Sundanese does not include diacritics, even though diacritics are supposed to be used in Sundanese to differentiate e and *è* (pronounced differently). Due to the absence of diacritics in the Sundanese transcript, we only observed a few minor cases, involving only the name *Beyoncé*, which was predicted without the accent as *Beyonce*, since the models are fine-tuned without any diacritics.

## 5 Limitations

This study has three main limitations. First, the OpenSLR corpora were only from limited regions, which may not reflect spontaneous or dialectal variation in Javanese and Sundanese. Second, the noisy conditions are synthetic and cannot fully capture real-world environments such as conversational overlap or varied recording devices. Third, our experiments focus only on Whisper-based models with a small set of fine-tuning strategies. These factors constrain the generalizability of the findings but also motivate directions for improvement.

## 6 Conclusion

We evaluated Whisper-based ASR models on Javanese and Sundanese under noisy conditions. While clean audio performance was strong, WER degraded by 2–3× in low-SNR scenarios without noise-aware training. Both `SpecAugment` and synthetic noise improved robustness, with `NoiseTrain` consistently outperforming other methods on average across models and languages. Error analysis showed Sundanese struggled with vowel confusion and name errors, while Javanese had more digraph and consonant issues, resulting in higher WER. Future work includes dialect-aware fine-tuning and speech enhancement for better real-world robustness.

## References


Panji Arisaputra, Alif Tri Handoyo, and Amalia Zahra. 2024. Xls-r deep learning model for multilingual asr on low-resource languages: Indonesian, javanese, and sundanese. *arXiv preprint arXiv:2401.06832*.

Badan Pusat Statistik. 2025. *Statistik Indonesia 2025*, 1 edition. Badan Pusat Statistik (BPS), Jakarta, Indonesia. Nomor Katalog: 1101001, Nomor Publikasi: 03200.25004. Tanggal Rilis: 28 Februari 2025.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations.

Indonesian BPS. 2024. Profil suku dan keragaman bahasa daerah, hasil long form sensus penduduk 2020. https://www.bps.go.id/.

Samuel Cahyawijaya, Holy Lovenia, Alham Fikri Aji, Genta Winata, Bryan Wilie, Fajri Koto, Rahmad Mahendra, Christian Wibisono, Ade Romadhony, Karissa Vincentio, Jennifer Santoso, David Moeljadi, Cahya Wirawan, Frederikus Hudi, Muhammad Satrio Wicaksono, Ivan Parmonangan, Ika Alfina, Ilham Firdausi Putra, Samsul Rahmadani, Yulianti Oenang, Ali Septiandri, James Jaya, Kaustubh Dhole, Arie Suryani, Rifki Afina Putri, Dan Su, Keith Stevens, Made Nindyatama Nityasya, Muhammad Adilazuarda, Ryan Hadiwijaya, Ryandito Diandaru, Tiezheng Yu, Vito Ghifari, Wenliang Dai, Yan Xu, Dyah Damapuspita, Haryo Wibowo, Cuk Tho, Ichwanul Karo Karo, Tirana Fatyanosa, Ziwei Ji, Graham Neubig, Timothy Baldwin, Sebastian Ruder, Pascale Fung, Herry Sujaini, Sakriani Sakti, and Ayu Purwarianti. 2023. NusaCrowd: Open source initiative for Indonesian NLP resources. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13745–13818, Toronto, Canada. Association for Computational Linguistics.

Abigail C Cohn and Maya Ravindranath. 2014. Local languages in indonesia: Language maintenance or language shift. *Linguistik Indonesia*, 32(2):131–148.

Yehoshua Dissen, Shiry Yonash, Israel Cohen, and Joseph Keshet. 2024. Enhanced asr robustness to packet loss with a front-end adaptation network. In *Proc. Interspeech 2024*, pages 5008–5012.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2021. *Ethnologue: Languages of the World*, 24 edition. SIL International, Dallas, Texas.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2025. *Ethnologue: Languages of the World*, twenty-eighth edition. SIL International, Dallas, Texas. Online version.

Lingyun Feng, Jianwei Yu, Deng Cai, Songxiang Liu, Haitao Zheng, and Yan Wang. 2021. Asr-glue: A new multi-task benchmark for asr-robust natural language understanding. *ArXiv*, abs/2108.13048.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA.

Praptomo Baryadi Isodarus. 2020. Penggunaan tingkat tutur bahasa jawa sebagai representasi relasi kekuasaan. *Sintesis*, 14(1):1–29.

Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha. 2018. Crowd-sourced speech corpora for javanese, sundanese, sinhala, nepali, and bangladeshi bengali. In *SLTU*, pages 52–55.

Eri Kurniawan. 2013. *Sundanese complementation*. The University of Iowa.

Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Paden Tomasello, Jacob Kahn, Gilad Avidov, Ronan Collobert, and Gabriel Synnaeve. 2020. Rethinking evaluation in asr: Are our models robust enough? In *Interspeech*.

Andrew L Maas, Quoc V Le, Tyler M O'neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y Ng. 2012. Recurrent neural networks for noise reduction in robust asr. In *Interspeech*, volume 2012, pages 22–25.

Daniil Orel and Huseyin Atakan Varol. 2023. Noise-robust automatic speech recognition for industrial and urban environments. In *IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–6. IEEE.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *Proc. Interspeech 2019*, pages 2613–2617.

PetaBahasa. 2019. Peta Bahasa. `https://petabahasa.kemdikbud.go.id`.

Riefkyanov Surya Adia Pratama and Agit Amrullah. 2024. Analysis of whisper automatic speech recognition performance on low resource language. *Jurnal Pilar Nusa Mandiri*, 20(1):1–8.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Kanishka Rao and Haşim Sak. 2017. Multi-accent speech recognition with hierarchical grapheme based models. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4815–4819.

Auliya Sani, Sakriani Sakti, Graham Neubig, Tomoki Toda, Adi Mulyanto, and Satoshi Nakamura. 2012.

Towards language preservation: Preliminary collection and vowel analysis of indonesian ethnic speech data. In *2012 International Conference on Speech Database and Assessments*, pages 118–122.

Michael L. Seltzer, Dong Yu, and Yongqiang Wang. 2013. An investigation of deep neural networks for noise robust speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7398–7402.

Muhammad A. Shah, David Solans Noguero, Mikko A. Heikkila, Bhiksha Raj, and Nicolas Kourtellis. 2024. Speech robust bench: A robustness benchmark for speech recognition.

Wedhawati Wedhawati, Marsono Marsono, Edi Setiyanto, Dirgo Sabariyanto, Syamsul Arifin, Sumadi Sumadi, Restu Sukesti, Herawati Herawati, Sri Nardiati, Laginem Laginem, et al. 2001. *Tata bahasa Jawa mutakhir*. Pusat Bahasa Departemen Pendidikan Nasional.

## A Experimental Configuration

To find the best SpecAugment setup for our training, we ran a series of controlled experiments using different time and frequency masking combinations. Table 5 lists the configurations we tested, each with different masking probabilities, lengths, and minimum number of masks applied to the time and frequency dimensions of the input spectrograms.

We started with individual masking strategies and then explored balanced and mixed configurations. These ranged from light to aggressive settings to see how much augmentation the model could benefit from before performance started to drop. Based on the validation WER, the best-performing configuration was then used to retrain the final model on the whole training set.

| Exp | Description | Time Prob | Time Len | Time Min | Freq Prob | Freq Len | Freq Min |
|---|---|---|---|---|---|---|---|
| 0 | Baseline (no SpecAugment) | 0.00 | 0 | 0 | 0.00 | 0 | 0 |
| 1 | Light Time Masking Only | 0.05 | 10 | 2 | 0.00 | 0 | 0 |
| 2 | Medium Time Masking Only | 0.10 | 15 | 2 | 0.00 | 0 | 0 |
| 3 | Heavy Time Masking Only | 0.20 | 20 | 3 | 0.00 | 0 | 0 |
| 4 | Light Frequency Masking Only | 0.00 | 0 | 0 | 0.05 | 10 | 1 |
| 5 | Medium Frequency Masking Only | 0.00 | 0 | 0 | 0.10 | 15 | 2 |
| 6 | Balanced Light (Time + Freq) | 0.05 | 10 | 2 | 0.05 | 10 | 1 |
| 7 | Balanced Medium (Time + Freq) | 0.10 | 12 | 2 | 0.10 | 12 | 2 |
| 8 | Time-Heavy Mix | 0.15 | 15 | 3 | 0.05 | 8 | 1 |
| 9 | Frequency-Heavy Mix | 0.05 | 8 | 1 | 0.15 | 15 | 3 |
| 10 | Aggressive (Heavy Time + Freq) | 0.20 | 20 | 3 | 0.15 | 18 | 3 |

Table 5: SpecAugment configurations used in each experiment. Values represent the masking probabilities, lengths, and minimum number of time and frequency dimensions masks.

## B Synthetic Noise Generation

To simulate real-world conditions, we create a set of noisy training data by mixing clean speech from the OpenSLR dataset with different types of background noise. We follow the general approach of Orel and Varol (2023) and use samples from `AudioSet` as our noise source. The noise types we picked were meant to reflect various environments in which people often speak, such as traffic, crowds, or indoor chatter, listed in Appendix E.

In our experiments, we use the following Signal-to-Noise Ratio (SNR) values: `-20, -15, -10, -5, 0, 5, 10, 15, 20, clean`, where `clean` refers to the original audio without any added noise. Negative SNR values mean more noise relative to the speech, whereas positive values are closer to clean conditions. We specifically chose these values, similar to prior work (Maas et al., 2012), since they cover the full spectrum of acoustic conditions from severe noise corruption to optimal listening environments.

To generate the noisy samples, we use the following formula:

$$\texttt{noisy\_audio} = \texttt{original\_audio} + \alpha \cdot \texttt{noise}$$

The scaling factor $\alpha$ controls how much noise is added and is calculated based on the target SNR using:

$$\alpha = \sqrt{10^{-\frac{\text{SNR}}{10}} \cdot \frac{\|\texttt{original\_audio}\|_2^2}{\|\texttt{noise}\|_2^2}}$$

## C Error Analysis

We analyzed the outputs of all Whisper models to understand the kinds of errors made in Javanese and Sundanese. To focus on more meaningful mistakes, we ignored casing differences.

### C.1 Character-level error analysis (CER)

We analyze CER to capture small edits common in agglutinative morphology, grouping aligned character edits into four types: extra spaces, vowel errors, consonant errors, and diacritic errors. Table 6 reports

counts by model and language: Javanese is dominated by consonant and diacritic changes, whereas Sundanese shows relatively more vowel and consonant changes; lowercasing the text (uncased CER) consistently reduces total character edits by about 7–18% across models, indicating that many mismatches are orthographic rather than full lexical substitutions. For computation, we normalize reference and hypothesis to NFC, collapse repeated whitespace, apply casefolding for uncased scoring, and compute $\mathrm{CER} = \frac{S+D+I}{N}$, where $S$, $D$, and $I$ are minimal character substitutions, deletions, and insertions from the alignment and $N$ is the number of reference characters; error types are assigned from aligned edits: whitespace $\rightarrow$ space; $\{a, i, u, e, o\} \rightarrow$ vowel; base–diacritic pairs (e.g., $e$ vs. é) $\rightarrow$ diacritics; remaining letters $\rightarrow$ consonant.

| **Error Type** | **Tiny** | | **Medium** | | **Large-v3** | | **Large-v3-turbo** | |
|---|---|---|---|---|---|---|---|---|
| | **jav** | **sun** | **jav** | **sun** | **jav** | **sun** | **jav** | **sun** |
| **Cased** | | | | | | | | |
| Additional space | 6249 | 6110 | 1278 | 419 | 900 | 338 | 1039 | 391 |
| Consonant mistake | 32881 | 30614 | 9611 | 2552 | 7702 | 2284 | 8632 | 2810 |
| Vowel mistake | 15744 | 14417 | 4742 | 1494 | 3722 | 1214 | 4168 | 1563 |
| Diacritics mistake | 3343 | 8 | 1797 | 0 | 1702 | 4 | 1799 | 1 |
| **Uncased** | | | | | | | | |
| Additional Space | 6355 | 6402 | 1308 | 439 | 918 | 351 | 1057 | 391 |
| Consonant mistake | 26693 | 21061 | 7157 | 2135 | 5815 | 1952 | 6392 | 2408 |
| Vowel mistake | 15650 | 14606 | 4661 | 1416 | 3660 | 1236 | 4119 | 1578 |
| Diacritics mistake | 3300 | 7 | 1759 | 0 | 1680 | 4 | 1793 | 1 |
| **Reduction (%)** | 10.68 | 17.65 | 14.56 | 8.19 | 13.88 | 7.45 | 14.52 | 7.48 |

Table 6: Character-level error type counts for Javanese (jav) and Sundanese (sun) across model sizes under cased and uncased evaluation; the bottom row shows the relative CER reduction (%) from cased to uncased per column.

### C.2 Word-level error analysis (WER)

We decompose word errors into insertions (I), deletions (D), and substitutions (S) under cased and uncased scoring, Table 7 reports per-language counts across model sizes, and the bottom row gives the relative reduction in total word edits when lowercasing is applied. For computation, we normalize reference and hypothesis to NFC, collapse repeated whitespace, apply casefolding for uncased scoring, tokenize by whitespace, and obtain minimal word-level alignments to count $I$, $D$, and $S$; word error rate is then $\mathrm{WER} = \frac{S+D+I}{N_{\text{ref words}}}$.

| **Error Type** | **Tiny** | | **Medium** | | **Large-v3** | | **Large-v3-turbo** | |
|---|---|---|---|---|---|---|---|---|
| | **jav** | **sun** | **jav** | **sun** | **jav** | **sun** | **jav** | **sun** |
| **Cased** | | | | | | | | |
| Insertion | 1541 | 1551 | 472 | 105 | 344 | 63 | 376 | 104 |
| Deletion | 2587 | 2615 | 592 | 224 | 414 | 227 | 526 | 191 |
| Substitution | 22178 | 17563 | 9995 | 1842 | 8445 | 1713 | 9600 | 2305 |
| **Uncased** | | | | | | | | |
| Insertion | 1546 | 1562 | 472 | 105 | 345 | 63 | 377 | 105 |
| Deletion | 2592 | 2625 | 592 | 224 | 415 | 227 | 527 | 192 |
| Substitution | 20535 | 15339 | 8715 | 1767 | 7386 | 1640 | 8359 | 2208 |
| **Reduction (%)** | 6.21 | 10.13 | 11.57 | 3.45 | 11.49 | 3.64 | 11.80 | 3.65 |

Table 7: Word-level error type counts (WER components) for Javanese (jav) and Sundanese (sun) across model sizes under cased and uncased evaluation. The bottom row shows the relative reduction (%) in total word edits per column.

## D Experimental Result

We report WER across SNR levels in Tables 8 and 9 and visualize the trends in Fig. 3. The tables cover four Whisper variants (`Tiny`, `Medium`, `Large-v3`, `Large-v3-Turbo`), each trained with **Clean**,

**SpecAug+Clean**, and **NoiseTrain**. Figure 3 shows `Tiny`, `Medium`, and `Large-v3-Turbo` for both languages, and Figure 2 presents the `Large-v3` curves. **As expected, WER increases as SNR decreases, and smaller models degrade more. Noise aware training reduces this drop, especially at low SNR.**

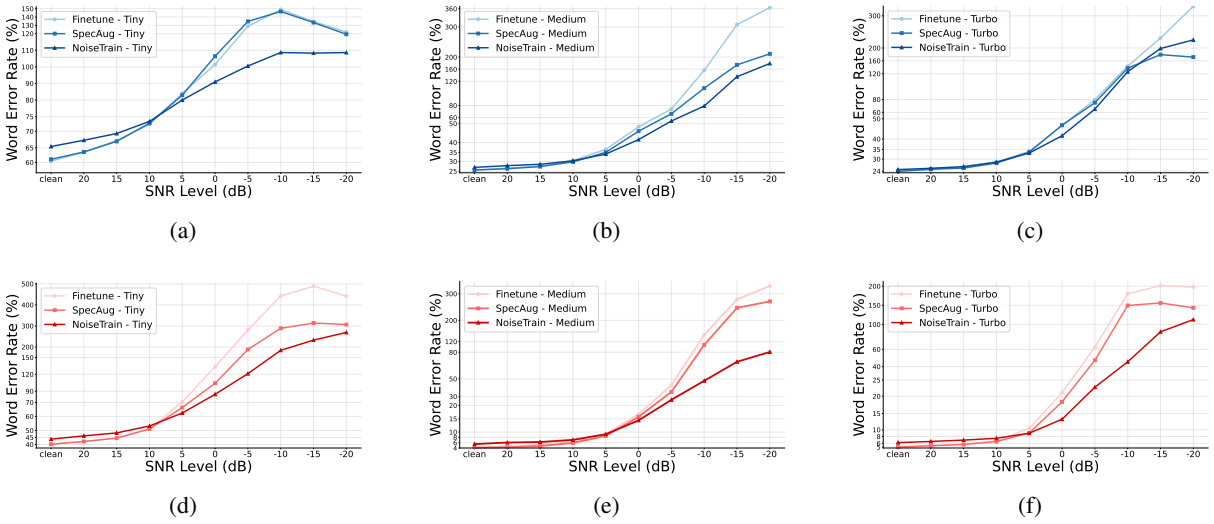



Figure 3: WER performance of Whisper variants across different SNR levels for Javanese and Sundanese: (a) Tiny - Javanese, (b) Medium - Javanese, (c) Large-v3-Turbo - Javanese, (d) Tiny - Sundanese, (e) Medium - Sundanese, (f) Large-v3-Turbo - Sundanese.

| **Tiny** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | Clean |
| Clean | 121.82 | 134.56 | 148.93 | 128.82 | 101.47 | 84.04 | 72.20 | 67.04 | 63.23 | 60.43 |
| SpecAug + Clean | 119.60 | 133.37 | 146.66 | 134.74 | 106.40 | 82.98 | 72.57 | 66.73 | 63.35 | 60.99 |
| NoiseTrain | 108.62 | 108.27 | 108.63 | 100.52 | 90.94 | 80.02 | 73.16 | 69.26 | 67.10 | 65.09 |
| **Medium** | | | | | | | | | | |
| **Model** | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | Clean |
| Clean | 363.54 | 307.64 | 156.21 | 74.14 | 48.35 | 36.37 | 30.47 | 27.55 | 26.50 | 25.40 |
| SpecAug + Clean | 210.88 | 174.37 | 108.62 | 66.32 | 46.00 | 34.79 | 29.76 | 27.26 | 26.14 | 25.45 |
| NoiseTrain | 178.90 | 135.18 | 79.08 | 54.38 | 41.56 | 33.90 | 30.32 | 28.50 | 27.76 | 26.87 |
| **Large-v3** | | | | | | | | | | |
| **Model** | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | Clean |
| Clean | 119.07 | 122.72 | 98.30 | 62.93 | 41.02 | 30.40 | 25.48 | 23.27 | 22.16 | 21.14 |
| SpecAug + Clean | 108.07 | 102.49 | 86.18 | 57.19 | 38.08 | 29.06 | 24.93 | 22.95 | 22.21 | 21.45 |
| NoiseTrain | 219.07 | 123.84 | 68.47 | 48.41 | 35.18 | 28.37 | 25.16 | 23.73 | 23.06 | 22.50 |
| **Large-v3-Turbo** | | | | | | | | | | |
| **Model** | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | Clean |
| Clean | 146.75 | 147.43 | 137.47 | 105.12 | 89.64 | 79.23 | 75.50 | 72.72 | 71.89 | 24.12 |
| SpecAug + Clean | 171.69 | 179.26 | 137.23 | 75.10 | 46.84 | 33.47 | 28.01 | 25.51 | 24.93 | 23.89 |
| NoiseTrain | 225.05 | 198.41 | 126.32 | 65.12 | 41.59 | 32.99 | 28.48 | 26.25 | 25.42 | 24.79 |

Table 8: WER across SNR levels for Javanese

## E Noise Classes from AudioSet

We provide a list in Table 10 of environmental and synthetic noise classes used during training and evaluation, sourced from AudioSet. These include a variety of real-world and synthetic sound events, some of which were used as held-out classes for testing generalization. Held-out classes are marked with a superscript $^*$.

| Tiny | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | Clean |
| Clean | 441.91 | 489.04 | 442.65 | 280.62 | 133.43 | 70.98 | 51.23 | 44.68 | 42.19 | 40.37 |
| SpecAug + Clean | 306.70 | 313.82 | 288.70 | 188.06 | 104.14 | 66.37 | 51.04 | 44.50 | 42.13 | 40.19 |
| NoiseTrain | 269.09 | 232.61 | 184.20 | 121.24 | 84.40 | 62.44 | 53.18 | 48.26 | 46.15 | 43.82 |
| **Medium** | | | | | | | | | | |
| **Model** | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | Clean |
| Clean | 329.53 | 278.64 | 145.19 | 43.01 | 16.69 | 9.23 | 6.33 | 5.23 | 4.66 | 4.03 |
| SpecAug + Clean | 271.12 | 247.21 | 107.82 | 35.27 | 15.73 | 8.55 | 5.91 | 4.74 | 4.29 | 4.09 |
| NoiseTrain | 81.13 | 69.27 | 47.81 | 26.37 | 14.39 | 9.19 | 7.06 | 6.25 | 6.06 | 5.46 |
| **Large-v3** | | | | | | | | | | |
| **Model** | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | Clean |
| Clean | 107.27 | 100.49 | 79.05 | 32.81 | 12.80 | 7.08 | 4.94 | 4.26 | 3.91 | 3.72 |
| SpecAug + Clean | 96.16 | 87.53 | 61.77 | 24.88 | 11.16 | 6.49 | 4.99 | | 4.40 | 4.14 |
| NoiseTrain | 65.07 | 51.02 | 32.23 | 17.15 | 9.30 | 6.16 | 5.07 | 4.54 | 4.31 | 4.10 |
| **Large-v3-Turbo** | | | | | | | | | | |
| **Model** | -20 | -15 | -10 | -5 | 0 | 5 | 10 | 15 | 20 | Clean |
| Clean | 197.83 | 201.34 | 180.01 | 62.52 | 21.10 | 10.45 | 6.64 | 5.70 | 5.33 | 4.83 |
| SpecAug + Clean | 143.48 | 156.14 | 149.58 | 47.39 | 18.35 | 9.06 | 6.51 | 5.60 | 5.24 | 4.83 |
| NoiseTrain | 112.65 | 80.81 | 45.48 | 22.72 | 13.16 | 8.99 | 7.47 | 6.93 | 6.53 | 6.17 |

Table 9: WER across SNR levels for Sundanese

| Class Name | Description | Count |
|---|---|---|
| Siren | The sound of a loud noise-making device used to provide warnings to people nearby. A siren typically consists of a single pitch that changes either smoothly or abruptly on timescales around one second. | 2188 |
| Car passing by | The sound of a motorized vehicle as it passes by a listener close to the vehicle's path. The sound may include engine and tire noise and will typically involve a clear build-up and/or decay of intensity as the vehicle approaches and retreats, as well as possible Doppler shift. | 1010 |
| Clatter | An irregular rattling noise, often produced by rapid movement, consisting of a cluster of transient sounds. | 772 |
| White noise | A random, unstructured sound in which the value at any moment provides no information about the value at any other moment. White noise has equal energy in all frequency bands. | 738 |
| Crackle | An irregular sequence of sharp sounds, as from sudden vaporization of liquids trapped in a burning solid, or from a collection of snapping noises. | 662 |

Continued on next page

**Table 10 – continued from previous page**

| Class Name | Description | Count |
|---|---|---|
| Wind noise (microphone) | The noise produced when a strong air current passes over a microphone, causing large amplitude local turbulence, normally recorded as mechanical clipping as the microphone element exceeds its limits of linearity. | 548 |
| Environmental noise* | The combined sounds of transport, industrial, and recreational activities. | 322 |
| Pink noise* | Unstructured noise whose energy decreases with frequency such that equal amounts of energy are distributed in logarithmic bands of frequency, typically octaves. | 283 |
| Boom* | A deep prolonged loud noise. | 283 |
| Firecracker | The sound of a small explosive device primarily designed to produce a large amount of noise, especially in the form of a loud bang. | 279 |
| Microwave oven | Sounds made by a kitchen appliance that heats food by exposing it to microwave radiation, including the noise of the fan, rotation mechanism, and microwave source, as well as the alert sound used to indicate that cooking is complete. | 250 |
| Traffic noise, roadway noise | The combined sounds of many motor vehicles traveling on roads. | 196 |
| Air horn, truck horn | The sound of a pneumatic device mounted on large vehicles designed to create an extremely loud noise for signalling purposes. | 161 |
| Hubbub, speech noise, speech babble | Loud, disordered, unintelligible speech noise from many sources. | 146 |
| Static | A crackling or hissing noise caused by electrical interference. | 101 |
| Inside, public space* | Sounds that appear to have been recorded in a public space such as store, restaurant, or travel terminus, often characterized by both reverberation and continuous background noise. | 98 |
| Rumble | A loud, low-pitched, dull, continuous noise. | 90 |
| Grunt* | A short low gruff noise, resembling the sound made by animals such as pigs. Specifically refers to humans. | 73 |
| Stomach rumble* | A rumbling, growling or gurgling noise produced by movement of the contents of the gastro-intestinal tract. | 64 |
| Noise | A sound that has no perceptible structure and that typically interferes with the perception of more interesting or important sounds. | 58 |

Continued on next page

**Table 10 – continued from previous page**

| Class Name | Description | Count |
|---|---|---|
| Knock | A sharp noise of a rigid surface being struck, usually without damage and deliberately, most often with the knuckles of the hand. | 54 |
| Clang* | A loud, resonant, discordant noise, as of a large and partly hollow metal structure being struck. | 49 |
| Bang | A brief and loud noise. | 38 |
| Squeak* | A short, high-pitched noise without a sharp attack. | 27 |
| Creak | A high-pitched noise with a perceptible variation in pitch as a result of pressure being shifted or applied on a surface, most commonly on wood. | 16 |

Table 10: Descriptions and counts of noise classes used from AudioSet. Held-out classes are marked with *.