


Noise-Robust Automatic Speech Recognition: A Case Study for Communication Interference

Julio Cesar Duarte   [Military Institute of Engineering | duarte@ime.eb.br]

Sérgio Colcher  [Pontifical Catholic University of Rio de Janeiro | colcher@inf.puc-rio.br]

 Military Institute of Engineering, Praça Gen. Tibúrcio, 80, Urca, Rio de Janeiro — RJ, 22290–270, Brazil.

Received: 16 March 2024 • Accepted: 01 July 2024 • Published: 09 July 2024

Abstract: An Automatic Speech Recognition (ASR) System is a software tool that converts a speech audio waveform into its corresponding text transcription. ASR systems are usually built using Artificial Intelligence techniques, particularly Machine Learning algorithms like Deep Learning, to address the multi-faceted complexity and variability of human speech. This allows these systems to learn from extensive speech datasets, adapt to several languages and accents, and continuously improve their performance over time, making them each time more versatile and effective in their purpose of transcribing spoken language to text. Much in the same way, we argue that the noises commonly present in the different environments also need to be explicitly dealt with, and, when possible, modeled within specific datasets with proper training. Our motivation comes from the observation that noise removal techniques (commonly called *denoising*), are not always fully (and generically) efficient. For instance, noise degeneration due to communication interference, which is almost always present in radio transmissions, has peculiarities that a simple mathematical formulation cannot model. This work presents a modeling technique composed of an augmented dataset-building approach and a profile identifier that can be used to build ASRs for noisy environments that perform similarly to those used in noise-free environments. As a case study, we developed a specific ASR for the interference noise in radio transmissions with its specific dataset, while comparing our results with other state-of-the-art work. As a result, we report a Character Error Rate value of 0.3163 for the developed ASR under several different noise conditions.

Keywords: Automatic Speech Recognition Systems, Noise Robustness, Portuguese ASRs

1 Introduction

Speech recognition is the process of converting the human voice into a sequence of words and linguistic resources that provide an understanding of what is being said [Huang *et al.*, 2001]. When it is done through an automated process, typically with software embedded in a device, this tool is often called an *Automatic Speech Recognition (ASR)* System [Li *et al.*, 2014] that converts an audio waveform into a speech text transcription [Duarte and Colcher, 2021].

Due to their complexity and diversity caused by different languages and accents, such tools are usually built using Artificial Intelligence (AI) techniques that can be applied to construct an ASR efficiently and effectively, ensuring a good performance both in transcription quality and processing time. Currently, Machine Learning (ML) is the most well-known technique for building such recognizers, as it has the ability to learn from historical data—in this case, pairs of waveforms and their transcriptions. More specifically, Deep Learning (DL) techniques can be used not only to learn from the data but also to create expressive attributes from the raw waveform signal, replacing the work of a specialist in creating the essential attributes for the model.

In this context, Deep Speech [Hannun *et al.*, 2014] stands out as a robust end-to-end speech system framework for developing and evaluating ASRs. It uses a Recurrent Neural Network (RNN) and the Connectionist Temporal Classification (CTC) loss functions to learn from data, in conjunction

with a Language Model (LM), allowing the adjustment of several hyperparameters. Deep Speech has proven to have good results for the task, especially when considering the compromise of performance over training time, for the task on languages such as English and Mandarin [Amodei *et al.*, 2016].

As ASRs are increasingly becoming part of everyone's daily life [Duarte and Colcher, 2021], built into personal assistants and helping in the execution of common daily tasks, noise robustness is also becoming a natural requirement. Nevertheless, the construction of good ASRs is still a challenging task [Li *et al.*, 2014], since they are increasingly employed in environments with high distortions and different characteristics from those employed when recording the datasets used for their training.

Much in the same way that ASRs need to have different training for each employed language, the noises commonly present in the environments in which they are used also need to be mapped and, when possible, modeled within the datasets, since natural removal techniques, like denoising, face several challenges related to limited training sources, where obtaining both clean and noisy audio samples proves difficult, and the fact that real-world audio signals usually contain inseparable noises. Consequently, denoising may not exhibit comparable performance in real-world settings in the same way as in controlled experimental environments [Zhang and Li, 2023].

Furthermore, most of the current work considers the noise

in ASRs as being a simple Additive White Gaussian Noise (AWGN) [Carlson *et al.*, 2002], or uses collected noise samples from regular household appliances and urban noises [Yılmaz *et al.*, 2014; Prodeus and Kukharicheva, 2017; Shimada *et al.*, 2019; Maruf *et al.*, 2020] such as cars or loud chats. Also, some works reduce the complexity of ASRs by using datasets with simple commands, names, or digits [Meneses Santos, 2016; Pervaiz *et al.*, 2020]. These presumed simplifications frequently fall short of accurately capturing the true complexity of the noise environments in which the ASR will operate. This holds true despite the capability of neural networks, particularly those trained in a DL context, to extract hierarchical features from noisy data without necessarily relying on a priori knowledge of the noise model. For instance, noise degeneration due to communication interference, which is almost always present in radio transmissions, has peculiarities that a simple mathematical formulation cannot model [ITU, 1992]. For example, this is especially true in military communication environments such as high-frequency (HF) channels used in the Amazon rainforest. Also, the attempt to model noisy environments as reliably as possible has been addressed in various ways, as we see, for example, in published recommendations [ITU, 1992] for noise parameter setup.

In this sense, the Brazilian Army's priority 1.1 [Centro Tecnológico do Exército, 2020] is the Software Defined Radio Project [Exército Brasileiro, 2019], and radios developed under this project can benefit from a technology that can automatically generate transcripts of received audio across different platforms. This is even more useful if incoming message storage is a requirement.

The main objective of this article is to present the proposal of modeling techniques that can be used to build ASRs used in specific noise environments that perform similarly to those trained and evaluated in noise-free environments. As a case study, noise from interference in radio transmissions is applied to a set of transcriptions in Portuguese [Duarte and Colcher, 2021]. The choice for the Portuguese language is due to the scarcity of related articles in this language [Quintanilha *et al.*, 2020; Gris, 2021; Gris *et al.*, 2022]. However, the methodology proposed here is generic and can be applied to any language.

We evaluate the developed ASRs with a dataset as close as possible to the noisy environments in which they are meant to be used while comparing them with the state-of-the-art.

To summarize, the contributions of this work are threefold. First, we contribute to the development of noise-robust ASRs by showing results obtained in a noisy communication environment. The steps presented here can be reproduced in any noisy environment as long as real-world data sets or mathematical models are available; Second, we also contribute to the development of Portuguese ASRs by comparing our results with other works that used similar datasets and language models. This can be used in any other language as long as datasets with pairs of audio and transcriptions are provided; Finally, we contribute to the development of a profile identifier that determines the best ASR to be applied in an audio file where its noise characteristics are unknown. Such profile identifier performs similarly (in terms of the performance measures) to the ASRs when compared to a perfect

theoretical identifier.

In the remainder of this article, we will present the necessary steps to reproduce our results. First, in Section 2, we show related work, focusing on Portuguese ASRs as well as other works dealing with noisy environments. Section 3 presents the Deep Speech framework used in our experiments, while Section 4 presents the used dataset and proposed configuration for the development of noise-robust ASRs. We then conclude our evaluation in Section 5 with our experiments that show near state-of-the-art results for the task. Finally, concluding remarks and possible future work are presented in Section 6.

2 Related Work

This section frames our contribution in the context of existing research both on Portuguese and noise-robust ASRs, as these are the two main aspects for comparison with the results of our work.

Initially, Li *et al.* [2014] provide an in-depth survey of the theme of robust noise ASRs, comparing more than 50 works in the field in terms of domain processing (feature versus model), distortion modeling (implicit versus explicit), prior knowledge of the distortion, processing (deterministic versus uncertain) and training (joint versus disjoint). The authors point out the good results as well as the challenges in using a Deep Neural Network (DNN) for this type of system, since DNNs provide a strong normalization to heterogeneous data present in noisy audio in the form of new powerful features that can then be used by other techniques such as an Hidden Markov Model (HMM). We highlight, then, more recent works on the theme of noise-robust ASRs.

Yılmaz *et al.* [2014] propose the use of Noise Robust Exemplar Matching (N-REM) with the Active Noise Exemplar Selection (ANES) technique that extracts noise exemplars from noise-only training sequences. The authors used the Chime-2 and Aurora-2 datasets formed by utterances in English combined with several types of noise such as subway, car, restaurant, and street, among others, and obtained results of 93.5% accuracy (Signal-to-noise ratio (SNR) 9 dB) and Word Error Rates (WERs) of 4.9% and 5.6% (SNR 10 dB).

Conversely, Wang and Wang [2016] combine two DNNs with a speech separation front-end and an acoustic model to form a better network for ASR, while adjusting the weights for each module. Experiments that were conducted by adding reverberant noises such as speakers, electronic devices, footsteps, and laughter to the English Chime-2 dataset consisting of multiple utterances, achieved an WER of 10.63%.

Meneses Santos [2016] proposes the use of a hybrid model that uses both a CNNs and a HMM to build an ASR for a dataset that contains Portuguese utterances and digits. He used noises from different sources like chitchat, engines, and industry while obtaining accuracies ranging from 88.91% to 99.67%.

In two different works, Prodeus and Kukharicheva [2016; 2017] propose the use of training ASRs with noise samples such as grinders, computers, and trucks that use the Fully Matched Training (FMT) and Spectrum Matched Training

(SMT) techniques. Experiments performed on a simple Russian dataset showed 95% performance in terms of accuracy for an SNR of 10dB or more.

On the other hand, Wang *et al.* [2018] proposes the use of a 6-layer Context-dependent (CD)-DNN-HMM in order to train a dataset composed of English utterances. Those utterances were extracted from the WSJ0 corpus that was processed with reverberation, interference, and background noises, and achieved an WER of 6.56%.

Throat microphones are highlighted by Ribeiro [2019] as a way to enhance the performance of ASRs in an environment of multimedia noises such as music and acoustic video content. Experiments with Multilayer Perceptrons (MLP) and Self-Organizing Maps (SOM) were conducted on a Portuguese dataset containing simple command strings. The use of the proposed device presented a WER reduction of 19.6%.

Shimada *et al.* [2019] propose the use of online Minimum Variance Distortionless Response (MVDR) beamforming to initialize and update the parameters of Multichannel Nonnegative Matrix Factorization (MNMF) instead of DNN beamforming methods to build noise robust ASRs. Experiments conducted on the Chime-3 utterance dataset mixed with urban noises such as buses, cafeterias, pedestrian areas, and streets, showed a performance in terms of WER ranging from 13.88% to 16.16%.

The use of Mel Frequency Cepstral Coefficients (MFCC) and Relative Spectral Transform-Perceptual Linear Prediction (RASTA-PLP) features are investigated by Maruf *et al.* [2020] to train a CNN-based ASR. A Bangla dataset consisting of digits and utterances combined with urban noises was evaluated, providing a reported accuracy of 93.18%.

Pervaiz *et al.* [2020] analyze the use of data augmentation techniques in conjunction with Deep Feed-Forward Networks, Long Term Memory Networks (LSTMs) and CNNs. Their proposed methodology was applied to an Asian-accented English data set called Speech Command Dataset, which includes several public utterances combined with background noise files such as running taps, dishwashing, and white and pink noises. The authors report the best result of 88.2% in terms of Top-One error.

A Gated Recurrent Fusion (GRF) method used in conjunction with a joint training framework to dynamically combine features that can remove noise signals and also learn raw fine structures to alleviate speech distortion is proposed by Fan *et al.* [2021]. The method is applied to a speech corpus called AISHELL-1 that consists of Mandarin utterances, combined with several noises from the Nonspeech Sounds noise database, such as traffic, animals, claps, and showers, among other noises. The reported results presented a performance of 10.04% in terms of Character Error Rate (CER) reduction.

In order to finish the literature review, we now present works whose focus is on the construction of ASRs for the Portuguese language with only limited applicability to use in noisy environments.

Quintanilha *et al.* [2020] propose the use of DNNs inside the Deepspeech framework by adding a newly trained 15-gram language model. The authors presented experiments with the LibriSpeech and BRSD (v1 and v2) datasets, showing results in terms of CER (10.49%) and WER (24.45%).

Spread across two different works, Gris *et al.* [2021; 2022] proposed a fine-tuned version of Wav2vec 2.0 XLSR-53 for the development of a Portuguese ASR, using only open available audios. Several different Portuguese datasets were used for the training of this ASR, such as CETUC, LAPSMB 1.4, VoxForge, Multilingual LibriSpeech, and Common Voice Dataset. An WER of 11.95% was reported as the best result for this task.

Candido Junior *et al.* [2023] introduced CORAA (Corpus of Annotated Audios) ASR, a publicly available dataset designed for ASR in both Brazilian and European Portuguese, aiming to fill the gap in datasets containing spontaneous speech. Additionally, they presented results for a public ASR model also based on Wav2Vec 2.0 XLSR-53, fine-tuned using CORAA ASR. This model achieved a WER of 24.18% and 20.08% on CORAA ASR and Common Voice datasets, respectively, along with a CER of 11.02% and 6.34%.

Finally, Scart *et al.* [2022] proposed the use of a simplified training version of wav2vec, which only fine-tunes a pre-trained model. The data set used was derived from the Common Voice Portuguese subset and built by simulating, through software, the characteristics of a narrowband FM transmitter and receiver, in addition to a noisy communication channel. This proposed methodology presented results of a relative reduction of 51.7% in terms of CER when using a value of SNR of 0 dB.

Although a direct comparison between the works presented is challenging to achieve, due to the use of different datasets, performance metrics, and validation strategies, Table 1 provides a summary of the main characteristics of these works. It can be noticed that few works fully consider all the aspects necessary for the construction of an ASR, but rather focus on or are limited to basic commands and utterances. Moreover, a significant portion of these works predominantly concentrate on the English language, and even those that incorporate the Portuguese language tend to overlook real noisy data, relying on generic data augmentation techniques instead. Furthermore, even considering different types of noise, only one of them considers the analysis of the influence of interference noise on radio communications, which is the case study of this work.

3 The Deep Speech framework

Deep Speech [Hannun *et al.*, 2014] is an end-to-end open-source ASR framework that uses Google TensorFlow [Abadi *et al.*, 2015] to implement the deep neural networks used for the character-based classification process.

The choice of Deep Speech as the tool for ASR generation in this work is due to its ease of conducting experiments and feasibility of execution on low-cost computing platforms. This allows for the evaluation of the proposed methodology in a highly reproducible environment that can be extended to other applications, while allowing a wide range of setups and scenarios. Naturally, other modern approaches to ASR generation, such as Whisper [Radford *et al.*, 2023] based on transformers, could also be utilized, albeit it would require extensive more computational resources. Nevertheless, the experiments conducted here enable the evaluation of the pro-

Table 1. Related work summary for noise-robust and Portuguese-based ASRs

Work	Techniques	Datasets	Noise types	Language	Main result
Yılmaz <i>et al.</i> [2014]	N-REM, ANES	Chime-2, Aurora-2	mainly street noises	English	Acc. - 93.5%
Wang and Wang [2016]	DNN	Chime-2	in-house reverberant noises	English	WER - 10.63%
Menêses Santos [2016]	CNN, HMM	Utterances and digits	chitchat, engines, and industry noises	Portuguese	Acc. - 88.91% to 99.67%
Prodeus and Kukharicheva [2016; 2017]	FMT, SMT	Names and numbers	urban noises	Russian	Acc. - 95%
Wang <i>et al.</i> [2018]	DNN	Utterances	reverberation, interference, and background noises	English	WER - 6.56%
Ribeiro [2019]	MLP, SOM	Simple commands	multimedia noises	Portuguese	WER reduction - 19.6%
Shimada <i>et al.</i> [2019]	MVDR, MNMF	Chime-3	urban noises	English	\overline{WER} - 13.88% to 16.16%
Maruf <i>et al.</i> [2020]	CNN	Utterances and digits	urban noises	Bangla	Acc. - 93.18%
Pervaiz <i>et al.</i> [2020]	DNN, LSTM, CNN	Speech Command	running taps, dish-washing, and white and pink noises	English	Top-One error - 88.2%
Fan <i>et al.</i> [2021]	GRF	AISHELL-1	traffic, animals, claps, and shower noises	Mandarin	CER reduction - 10.04%
Quintanilha <i>et al.</i> [2020]	DNN	LibriSpeech, BRSD	clean speech	Portuguese	CER - 10.49% WER - 24.45%
Gris [2021]	DNN	mainly Common Voice	clean speech	Portuguese	WER - 11.95%
Candido Junior <i>et al.</i> [2023]	DNN	CORAA, Common Voice	background noise, spontaneous speech	Portuguese	CER - 11.02% 6.34% WER - 24.18% 20.08%
Scart <i>et al.</i> [2022]	DNN, CNN	Common Voice	narrowband FM channel	Portuguese	CER reduction - 51.7%
This work	DNN	Common Voice	communication interference noises	Portuguese	\overline{CER} - 31.64% (SNR>0) CER - 23.00% (SNR30)

posed methodology, highlighting its contribution.

Deep Speech allows not only the use of pre-trained ASR models but also their construction from scratch, simply feeding the tool with a dataset of audio files and respective transcriptions in the target language, as well as setting a diverse set of hyperparameters for the neural networks used, such as the number of neurons per layer, dropout and learning rates, among others.

Figure 1 shows the basic architecture of Deep Speech, which consists of an RNN architecture with five hidden layers. The first three layers, like the fifth, are composed of non-recurring neurons and use a clipped rectified-linear (ReLU) activation function. The outputs from the previous layers are used as the inputs to the next ones. The fourth layer is a recurring LSTM that includes a set of hidden units with forward recurrence. Finally, the output layer uses a softmax function that outputs the probabilities for each character considered in the alphabet.

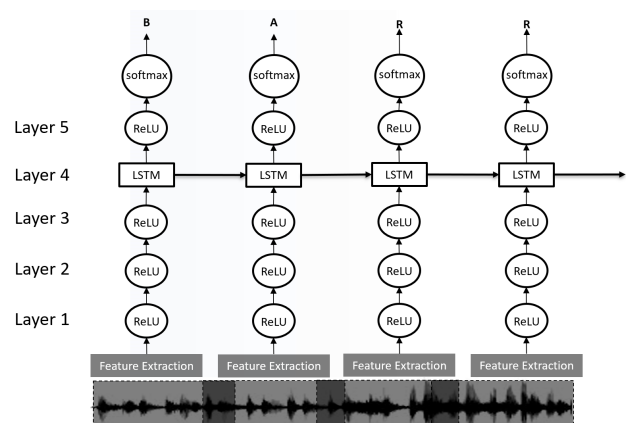


Figure 1. Deep Speech basic architecture. Source: Duarte and Colcher [2021]

In order to train their RNNs, Deep Speech uses the Connectionist Temporal Classification (CTC) [Graves *et al.*, 2006;

Hannun, 2017] loss function that transforms the outputs into conditional probabilities over all alphabet sequences. Those probabilities can then be used to predict the most probable labels for a given sequence.

The main idea of CTC is to provide a free alignment between the input and output sequences. CTC works by summing over the probability of all possible alignments among all possible outputs of an input. With this in mind, the transcription *BAR*, for instance, also depicted in Figure 1, can be recognized by multiple outputs such as *BARR*, *BAAAR*, and *BAR* itself. To recover the output sequence, any sequence of the same characters from the alphabet is replaced by that character, which results in two problems. First, the input can have silence streaks without a character for the output. Also, multiple characters in a row can appear in a transcript, such as in the word *passes*.

To solve these problems, CTC introduces a dummy blank token (ϵ) for the alphabet that can represent the absence of a character or sequences of the same character. Whenever the token ϵ is generated, the sequences of the same characters are not merged and, in the end, the token is simply removed from the output. With this simple approach, the word *passes*, for instance, can be produced by the *pasεses* output, without loss of representation.

In addition to the classification process that uses DNNs, Deep Speech allows the use of an “external scorer” that makes corrections in the transcriptions after the neural network recognition process. This external scorer is composed of a prefix tree data structure that contains all possible vocabulary words and a language model [Mozilla Corporation, 2020].

A Language Model (LM), applied as a post-processing step, assigns probabilities for word sequences present in a training corpus [Jurafsky and Martin, 2021]. The idea behind this is that spoken words correlate, meaning that the next possible words in a sentence have probabilities based on their general appearance in the language. For instance, after the definite article *the*, there cannot be a verb, so the probability of any verb after the word *the* is null.¹

Using language models helps fix small mistakes made by the recognizer, which would deteriorate performance in terms of word recognition. Deep Speech supports the usage of the KenLM Language Model Implementation [Heafield, 2011] and two hyperparameters to control the effects of the language model, α and β . α controls the weight of the language model about the output of the neural network. A value of zero, for example, disables its usage. Conversely, β controls the weight of word insertion, which can be useful, especially if the recognizer misses some small words. A model optimizer script provided by Deep Speech can automatically determine these values.

4 Designing a noise-robust automatic speech recognition

This section presents the setup necessary to reproduce the experiments reported in Section 5 and is divided into two

parts. The first one contains the details about the dataset used and its subsets used in the training and evaluation phases of the ASRs developed in this work. The second part describes the selected hyperparameters for configuring Deep Speech.

4.1 Dataset

In all our experiments we use the Common Voice Dataset [Ardila *et al.*, 2020], more specifically, the Portuguese subset of the noisy version developed by Duarte and Colcher [2021]. The reason for choosing such a dataset is that, currently, Common Voice is a benchmark for comparing ASRs. Particularly, its noisy version fulfills the necessary requirements for this work, which are: Portuguese language; and different noises from radio communication interference.

The noisy dataset has four distinct subsets, and each subset has noises with the following Signal-to-noise ratio (SNR) relations: $\{-30, -20, -10, -5, 0, 5, 10, 20, 30\}$, following all recommendations provided by ITU [1992]. SNR measures the degree of noise contamination in the signal [Carlson *et al.*, 2002]. A small value, usually negative, indicates a high degree of degeneration due to noise. Thus, even reporting statistics on all listed SNRs, the discussion of results will be limited to their positive values, since for negative SNRs, even the human ear has extreme difficulty in understanding what is being said.

The four different subsets contain different ways to add noise to the original base. The first subset uses a simple and generic Additive White Gaussian Noise (AWGN). Conversely, the second subset contemplates the addition of noises collected directly from HF receivers in the form of files that can be merged into the original base. The third subset implements, via software, complex mathematical models for the representation of such noises. Such models incorporate the simulation of parameters reported in official recommendations [ITU, 1992]. Finally, the fourth subset contains files generated through a dedicated device that performs the complete simulation of a customized HF channel using several parameters also present in recommendations [Duarte and Colcher, 2021].

Figure 2 shows an example of the same file subjected to different forms of noise (SNR = 10) in the dataset. It is interesting to note that, although the files are very similar visually, the built ASRs lose performance when submitted to different subsets than those for which they were trained.

The fourth subset implements the most effective way to generate simulated noise through an HF channel. For this reason, this subset is used in the training phase of our SNRs. Regardless of that, to be able to compare the four strategies implemented in the dataset, we report the results in all subsets.

4.2 Deep Speech Setup

Since Deep Speech implements the development of ASRs through deep neural networks, the choice of hyperparameters plays an important role in its training and consequent performance. Thus, a poor choice of hyperparameters can invalidate the comparison of ASRs, even when the objective

¹There are special cases where noun verbs can follow a definite article.

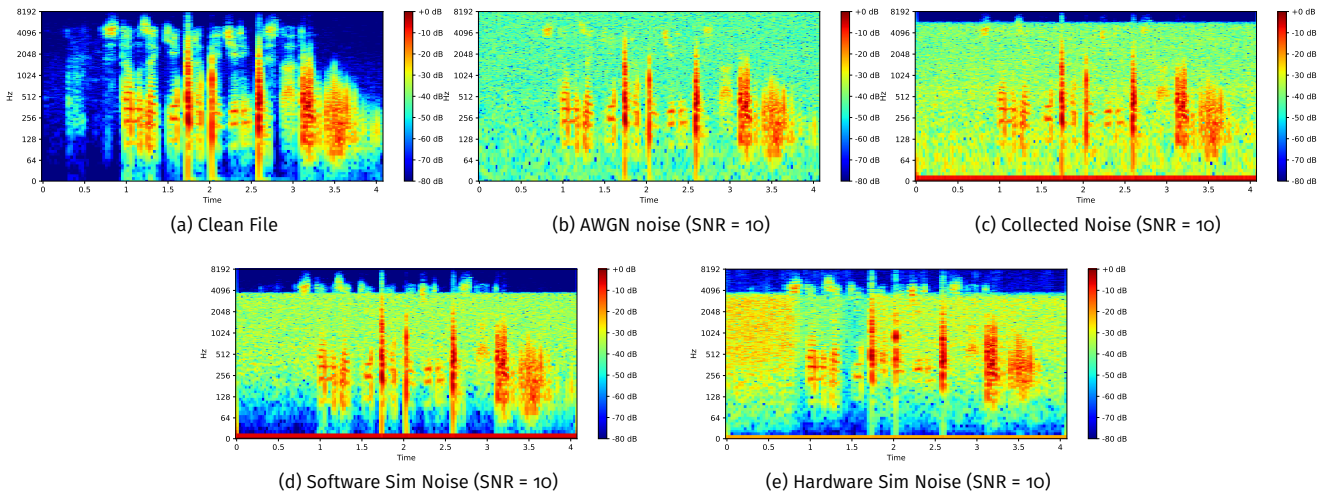


Figure 2. Spectrogram plots of the same file applied to all noise strategies

is mostly to verify the impact of using noisy datasets in their training.

Conversely, choosing the “best” hyperparameter set can be a costly task, especially with very large datasets. Keeping that in mind, we followed some recommendations also used for developing ASRs for others languages [Duarte and Colcher, 2021]. All hyperparameters were set to the default values, except the ones provided in Table 2. Also, the batch training size was set to 16. This was the maximum value at which the experiments could be run on our computing infrastructure, due to memory limitations available to each Graphics Processing Unit (GPU).

Parameter	Value
Epochs	100
Neurons per Layer	2048
Feature Extraction Audio Window Length	32 ms
Learning Rate	0.0001
Dropout Rate	0.40

Table 2. Deep Speech hyperparameter settings

In terms of using the language model, the same principle cannot be applied. Language models are language dependent and their best parameters (α and β) should be chosen according to the data sets and alphabets used for training the ASR. Following this idea, a straightforward language model was trained using the KenLM tool [Heafield, 2011] compatible with Deep Speech. Furthermore, Deep Speech’s `lm_optimize` script was used to find the best values for α and β . The language model generation was performed using the validated subset of the dataset, and, of course, better models can be created using larger text Portuguese corpora, but for our tests, the derived language model has already greatly improved the performance of the ASRs in terms of WER.

4.3 ASR Profile Identifier

Noise environments exhibit significant variation in terms of type and conditions, posing a challenge for a singular ASR to effectively handle every aspect. By training multiple ASRs under different setups, it is possible to employ a strategy to

selectively choose the most appropriate one. In essence, this approach allows the implementation of a strategy for determining the optimal ASR to use at any given time.

Here, as we may have ASRs trained in different circumstances, in our case trained with different SNR values, we could choose the most suitable ASR for each processed audio. In our case, we train a MLP neural network called Profile Identifier (PI), whose main objective is to choose, among the trained ASRs, which one was trained under the most similar noisy or noise-free environment.

To train the PI, we extract the MFCC features from the training dataset, dividing all files into audio slices. These MFCC features are then used as inputs to a MLP training process.

5 Experiments and Results

We present here the analysis of the results of the experiments carried out in order to show our methodology for building noise-robust ASRs. The materials needed to reproduce the experiments are listed in the “Availability of data and materials” of this article, making available the source codes for Deep Speech [Hannun *et al.*, 2014], Common Voice [Ardila *et al.*, 2020] and its noisy-version [Duarte and Colcher, 2021].

Basically, we want to provide evidence of three main aspects, which are: dealing with noisy audio data in the training process, applying an LM after the classification process, and using a profile identifier to determine the better ASR to be applied for each instance. As already stated in Section 4, we will only consider the fourth subset of the full dataset, limited to its positive values of SNR for training, in order to ensure a fair evaluation. Conversely, all subsets will be reported in the evaluation.

We begin with the premise that random white noise cannot accurately represent a real-world noisy environment. This is supported by our initial experiments [Duarte and Colcher, 2021] and is consistent with the findings of all related works on noise in ASRs presented in Section 2.

First, we evaluate the performance results in terms of CER

for the trained ASR using only noiseless audio, as shown in Figure 3. This is similar to the initial evaluation conducted by Duarte and Colcher [2021] and it will serve us as a baseline result. We can see that all subsets show the same pattern, where the results improve with higher values of SNR. The best results are always for the clean (noise-free) subset.

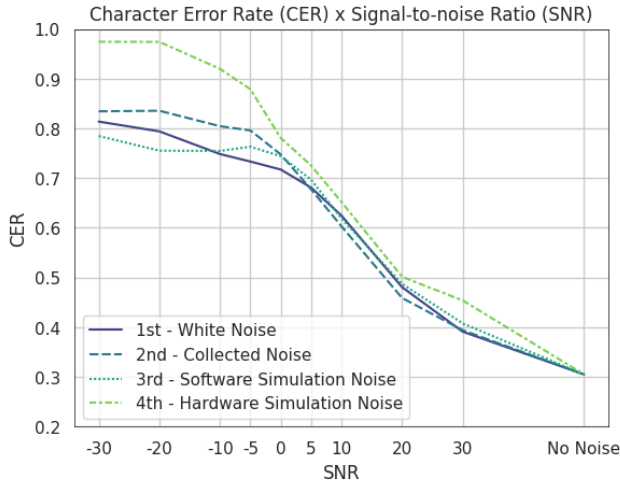


Figure 3. CER results for the noise-free trained ASR

Continuing, in Figure 4, we present the same evaluation, but now using ASRs trained with noise variant audios. Figures 4.a, 4.b, 4.c, and 4.d show, respectively, the results in the test dataset for the ASRs trained with SNR values of 30, 20, 10, and 5.

As expected, better results are achieved for the fourth test subset. However, it is also possible to observe a pattern where results improve for SNR values close to those used in training. This fact confirms the importance of performing the ASR training in channel conditions similar to those where it will be used, represented here by the SNR value. For instance, the CER for the 4th subset and SNR30 is 0.4538 for the noise-free trained ASR, while it improves to 0.3265 for the ASR trained with the SNR30 audio variants. This behavior is also present in ASRs that were trained with the other audio variants.

Next, we present results where we mixed different audio variants with their noiseless versions. The idea here is to provide some sort of *data augmentation* by embedding noisy audio in the training dataset. We incorporated the noisy audio in stages, from the least noisy (SNR30) to the most noisy (SNR5). Figures 5.a, 5.b, 5.c, and 5.d show, respectively, the results of incremental addition on the noise-free training dataset of noisy audio with SNR values of 30, 20, 10 and 5.

When comparing Figures 3 and 5, we can notice some interesting behaviors. First, we see the anticipated result that, the noise-free test subset shows superior performance when training with a dataset composed of clean and noisy (SNR30) audio than when training only with clean audio. As it can be seen in the second column of Table 3 (CER (Clean)), the results in terms of CER with the ASR trained only with noise-free audio is 0.3049, while the ASR trained with noise-free and noisy audio (SNR30) has a performance of 0.2886. On the other hand, augmenting the training dataset further does not improve the results. This shows that the method can work

just like a traditional data augmentation technique, where the augmented dataset gives better results than the original one.

Table 3. Results for the data augmented sets

	CER (Clean)	\overline{CER} (SNR>0)
Clean	0.3049	0.5275
SNR30+Clean	0.2886	0.4608
SNR20+SNR30+Clean	0.2936	0.4058
SNR10+SNR20+SNR30+Clean	0.2981	0.3709
Full Training Set	0.3306	0.3902

Second, when comparing all generated SNRs by further augmenting the training dataset, we see better results in terms of average CERs across all test subsets with a positive SNR value. We show these values on the third column of Table 3, where we can see the improvement behavior in terms of the average CER up to the second-to-last augmented set. This indicates that augmenting with very noisy audio can degrade performance.

In the next experiment, we want to evaluate the impact of adding a LM to the output of the generated ASRs. This is extremely important since ASRs make many single character mistakes which impact little on CER but a lot on WER. For instance, a single character error in a ten-length word penalizes 10% in terms of CER, while completely nullifying (100% error) that word, in terms of WER. A good LM can slightly improve CER results while greatly improving WER.

Table 4 presents the results of each individual ASR, while training and evaluating with the same SNR values, with and without the LM. For all trained ASRs, the LM is the same, as already stated in Section 4.

Table 4. Impact of the Language Model in each individual ASR

	CER		WER	
	w/o LM	with LM	w/o LM	with LM
SNR05	0.5169	0.4492	0.9173	0.6879
SNR10	0.4478	0.3777	0.8595	0.5927
SNR20	0.3741	0.2801	0.7869	0.4492
SNR30	0.3265	0.2300	0.7318	0.3773
CLEAN	0.3049	0.1936	0.7158	0.3167

As we can see in Table 4, both CER and WER take advantage of using the LM. CER shows an average percent improvement of 8.79% while WER presents a huge 31.75% improvement on average.

Finally, we report the results of our experiments using the PI strategy that determines the best ASR to use at any given time. After some preliminary tests to find the optimal hyperparameter setup for the MLP, we discovered that the setup outlined in Table 5 exhibited the most favorable performance-to-training time ratio. The trained classifier can then be used to determine the SNR value of any audio, among our fixed list: CLEAN, SNR30, SNR20, SNR10, and SNR5.

Figure 6 shows the results obtained by the PI on the test dataset. The primary idea is to illustrate the specific instances where the Profile Identifier (PI) tends to make more errors when attempting to identify the SNR noise values utilized in the training dataset. As we can see in this figure, the PI has great overall results, an accuracy of 0.74, but its performance

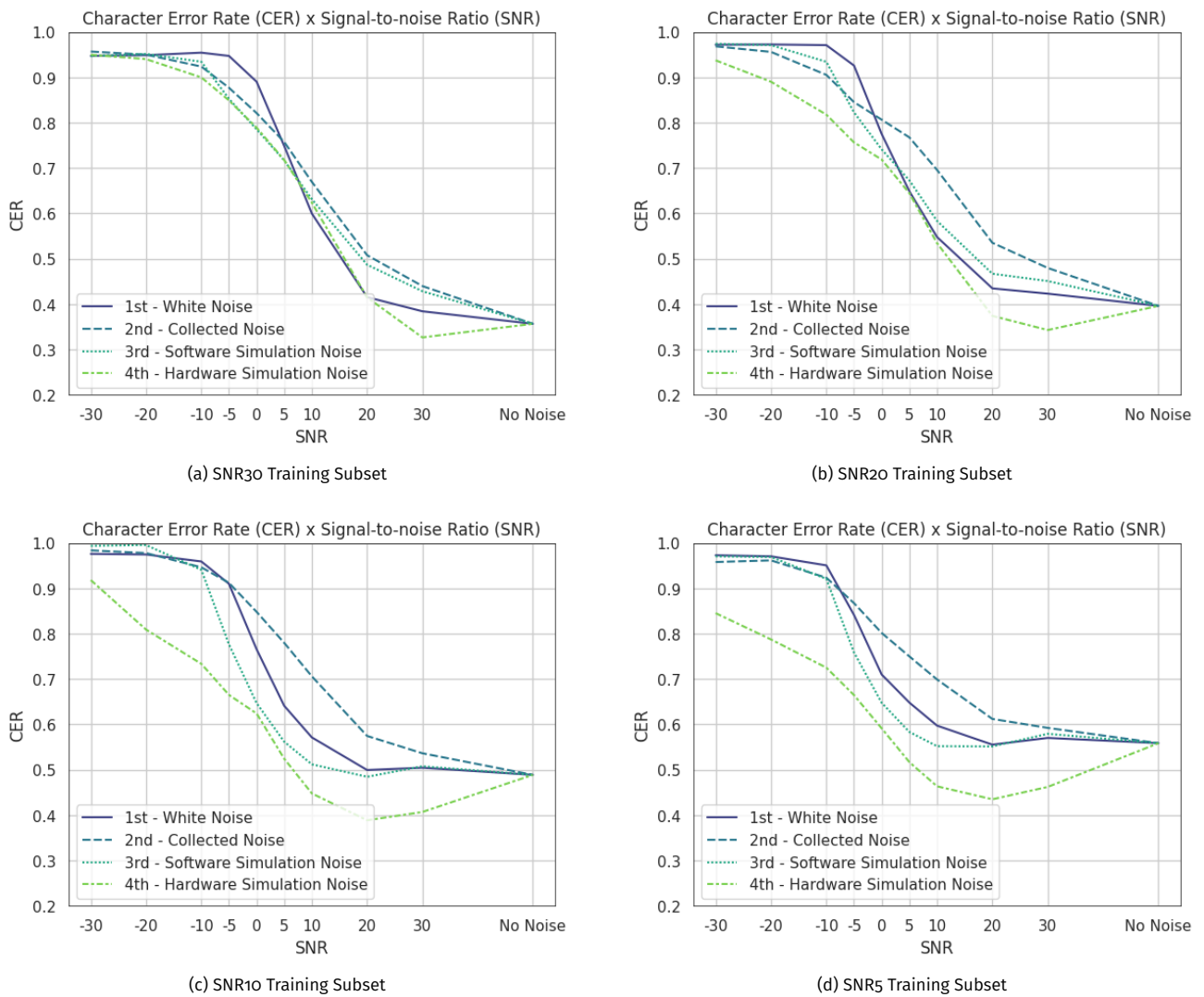


Figure 4. CER results for each trained ASR with noise variant audio

Parameter	Value
Neurons	Number of MFCC Features
Iterations	30
Audio slices	10
Alpha Value	0.0001
Solver	L-BFGS
Learning Rate	0.1

Table 5. Hyperparameter setup for the PI training process

decreases as the audio gets noisy. For instance, the CLEAN class has a f1-score of 0.99, while SNR10 has a f1-score of 0.43. Anyway, the most common mistakes are made in adjacent classes (SNR10 as SNR05 and SNR20 as SNR10), which still helps by assigning a good ASR for transcription recognition.

Using the proposed strategy, we can build an ASR which is a two-step transcription system: first, we determine the “best” ASR to use and then apply it to the input audio. Figure 7 shows the results of this PI-derived ASR in terms of WER (bar plots) and CER (line plots), both with or without the LM.

Figure 7 compares the result of this strategy to the ones that use an individual ASR trained with only one kind of noisy

audio data (one SNR value). We also compare it to a theoretical “perfect” PI that knows the answer of which SNR value was used to generate the noisy data. We denote this theoretical PI as the LowerBound for our strategy. Keep in mind that the results shown in this figure are not the same as those shown in Table 4, as that table shows results over the individual noisy datasets, while Figure 7 shows averages across all noisy datasets.

As we can see, the provided strategy improves the results on all individually trained ASRs, while presenting competitive results over the theoretical LowerBound strategy. This indicates that even making small mistakes in terms of the SNR value of the audio file, the chosen ASR can provide a good transcription. For example, the averages CER and WER (with the LM) for the PI are, respectively, 0.3163 and 0.5067, while for the “SNR20”-trained ASR (best individually) are 0.3982 and 0.5886. On the other hand, the best theoretical results, which are, respectively, 0.3061 and 0.4848, provide only a small improvement.

As a learning experience from the experiments carried out with the proposed trained ASRs, we can initially notice that training with noisy data is a good strategy, mainly, but not

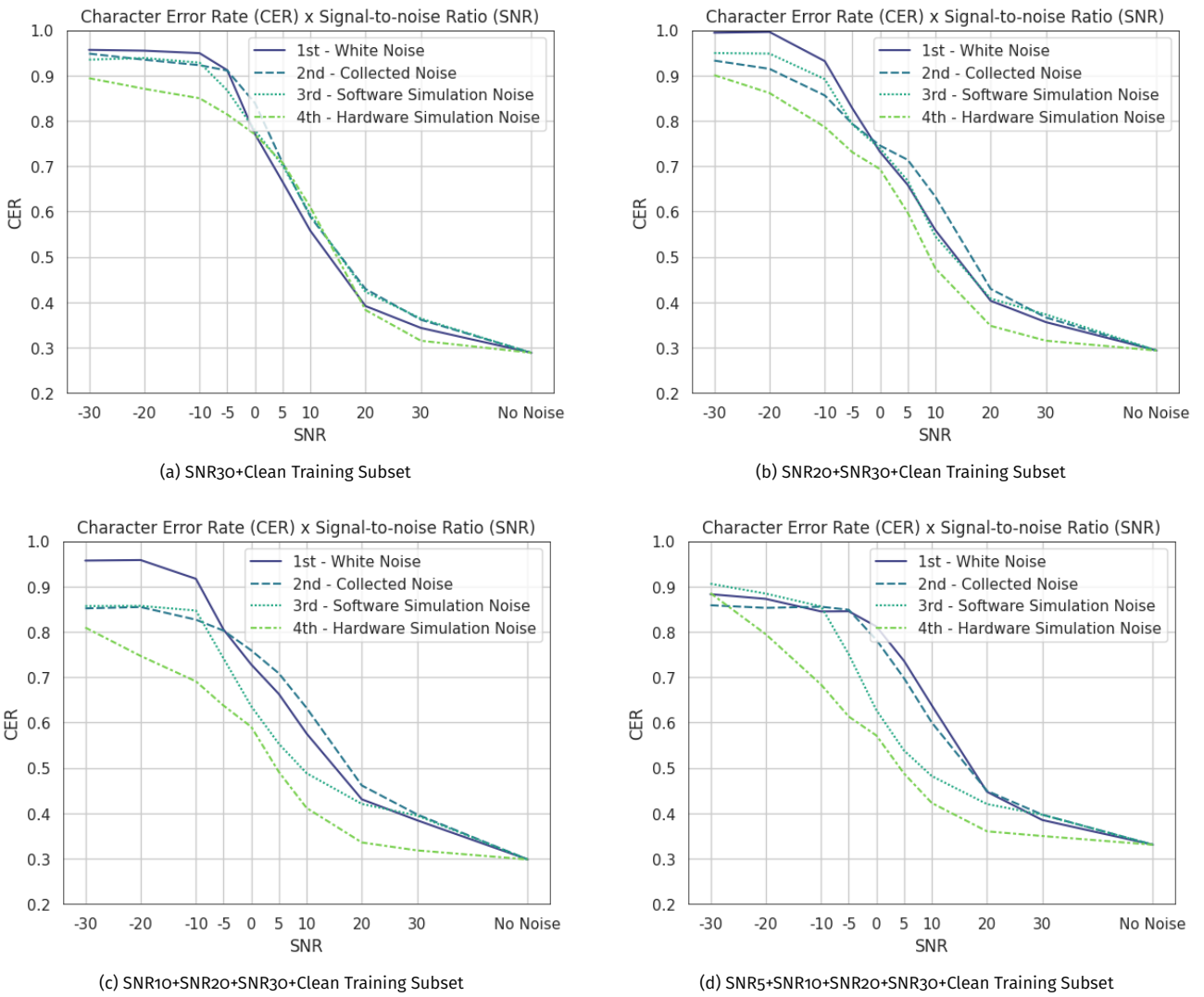


Figure 5. CER Results for the ASRs trained with the noisy augmented sets

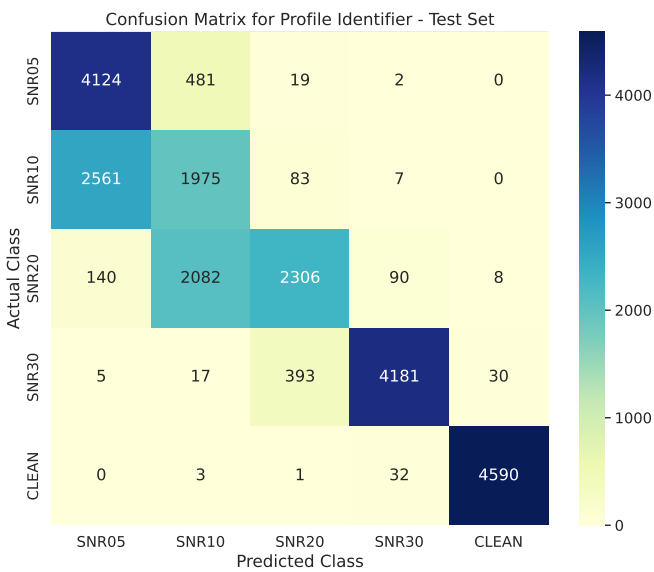


Figure 6. Confusion matrix results for the Profile Identifier

limited to, when the noise profile present in audio is known. If the noise profile is unknown, even ASRs trained under similar conditions can provide good results, either using a data

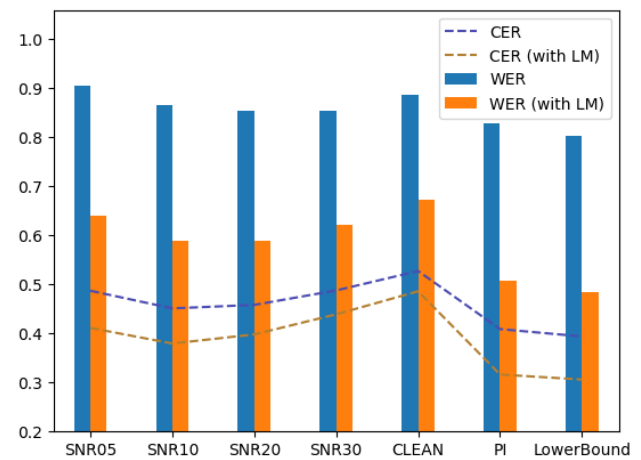


Figure 7. CER and WER results for the individuals ASRs, the Profile Identifier, and the LowerBound, across all noisy datasets

augmentation strategy to build the training dataset or a hybrid strategy where members of an ASR committee can be selected individually. Also, the use of an LM is very important, as ASRs generated with such techniques can lead to simple common errors that can be easily corrected, providing

better results in terms of WER.

All experiments presented here were performed with the same type of noise, originated from communication interference, and specialized based on its SNR, the most important attribute in this model. We may extrapolate that similar performance and behavior can be obtained by generated ASRs for other noise types or parameter variations in the modeling, as long as it is provided a way to represent such noise within the training dataset, either by collected examples, mathematical modeling, or dedicated systems for their generation. In this way, the more reliable this representation, the better the performance of the generated ASRs when applied to the same conditions in which the audios used in the training were obtained, while the ASRs generated with only noise-free audio, generally, perform worse.

6 Conclusion

The development of ASRs has become increasingly important due to their widespread use in several devices. Still, their performance often suffers in noisy environments where generic data augmentation techniques are ineffective, highlighting the need to integrate real noisy data into training for improved performance. The main aim of this work was to present a methodology for training ASRs in noise-specific environments, using representations of target noise obtained through mathematical modeling or noisy samples. Multiple approaches were presented for constructing these ASRs, employing traditional DNN training via the Deep Speech tool, along with techniques like data augmentation, hybrid models, and ensemble methods. Experimental results showed that the proposed training strategies improve ASR performance, with the hybrid ASR, incorporating noise characteristics, improving WER by 18.70% compared to ASRs trained solely on noise-free audio.

The contributions of the present work outpace the development of an ASR for the Portuguese language robust to telecommunications noise. In addition to the development and experimentation of an ASR prototype that deals with characteristic noises, our contribution lies in providing substantiating evidence endorsing the inclusion of noisy audio during the training phase, through traditional training of DNNs, ensemble methods that select the best ASR to be used for each audio input, and models trained using data augmentation techniques targeting at the characteristics present in the audio. Such experiments show that the training of ASRs can highly benefit from these data, in addition to the use of language models that allow the correction of transcription errors. All evaluations showed better results than the considered baseline Duarte and Colcher [2021], which was specifically designed to showcase an experiment intended to evaluate the feasibility of using the proposed dataset.

Despite the contributions made in this work, certain limitations should be acknowledged. Firstly, the utilization of a dataset in a single language restricts the scope of performance comparison. Expanding to multiple languages could provide a more comprehensive understanding of the proposed methodology's efficacy across linguistic variations. Furthermore, the hardware infrastructure imposed limita-

tions on conducting experiments with more powerful ASR development frameworks, and the access to enhanced computational resources would enable the use of more complex ASR models, improving the performance of the proposed methodology. Additionally, while the study predominantly focuses on modern DL approaches for denoising, the incorporation of traditional techniques for denoising was not explored. Integrating classical denoising approaches could offer valuable insights and contribute to a better evaluation of the proposed methodology.

As future work, moving forward, we intend to experiment with other noisy environments by changing the distortion parameters and creating even more specialized ASRs. Another noisy environment of interest is the Industry, where equipment and tools can create a hostile noisy environment for ASRs, particularly when using helmets and communication headsets. We also plan to improve the developed ASRs, incorporating more complex models into the training, as well as better language models. For example, the training methodology proposed here can be incorporated into the work of Quintanilha *et al.* [2020] or Gris *et al.* 2021; 2022, who also addressed the Portuguese language, generating ASRs with even better performance than those reported here. Additionally, recent models such as Whisper [Radford *et al.*, 2023] demonstrated significant robustness to noise, making it worthwhile to compare their performance for the specific noise context proposed here. Since employing an ensemble of ASR models may be computationally intensive, future research might focus on models that enhance audio quality before feeding it into the ASR, thereby improving the performance of any model used. Furthermore, the idea behind the profile identifier can be expanded to address other audio challenges, such as multiple speakers or different accents. Finally, we intend to test our methodology in another language, more precisely English, as there are many datasets available, to determine if the same kind of improvements are valid for other languages.

Declarations

Funding

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-22-1-0475. In addition, this work was partially supported by national funds through FINEP, Financiadora de Estudos e Projetos, and FAPEB, Fundação de Apoio à Pesquisa, Desenvolvimento e Inovação do Exército Brasileiro, under project “Sistema de Sistemas de Comando e Controle” with reference n° 2904/20 under contract n° 01.20.0272.00.

Authors' Contributions

JD: Conceptualization, Methodology, Investigation, Software, Validation, Writing — original draft, Writing – review & editing. SC: Methodology, Writing — original draft, Writing – review & editing. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The datasets analyzed during the current study and the source codes needed to reproduce the experiments are available at the following URLs: <https://commonvoice.mozilla.org/en/datasets>, <https://github.com/mozilla/DeepSpeech> and <https://github.com/duartejulio/noisy-asr/>.

References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., *et al.* (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. <http://tensorflow.org/>. Accessed: 09 July 2024.
- Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., *et al.* (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *ICML'16*, pages 173–182, New York, New York, USA. PMLR. DOI: <https://dl.acm.org/doi/10.5555/3045390.3045410>.
- Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., and Weber, G. (2020). Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association. DOI: <https://doi.org/10.48550/arXiv.1912.06670>.
- Candido Junior, A., Casanova, E., Soares, A., de Oliveira, F. S., Oliveira, L., Junior, R. C., da Silva, D. P. P., Fayet, F. G., Carlotto, B. B., Gris, L. R. S., and Aluísio, S. M. (2023). CORAA ASR: a large corpus of spontaneous and prepared speech manually validated for speech recognition in brazilian portuguese. *Language Resources and Evaluation*, 57:1139–1171. DOI: <https://doi.org/10.1007/s10579-022-09621-4>.
- Carlson, A. B., Crilly, P. B., and Rutledge, J. C. (2002). *Communication systems: An introduction to signals and noise in electrical communication*. Boston: McGraw-Hill, 4th edition. DOI: No DOI available.
- Centro Tecnológico do Exército (2020). Rádio definido por software de defesa (RDS-DEFESA). <http://www.ctex.eb.mil.br/projetos-em-andamento/84-radio-definido-por-software-rds>. Accessed: 09 July 2024.
- Duarte, J. C. and Colcher, S. (2021). Building a noisy audio dataset to evaluate machine learning approaches for automatic speech recognition systems. <https://doi.org/10.48550/arXiv.2110.01425>.
- Exército Brasileiro (2019). Plano Estratégico do Exército Brasileiro 2020-2023. http://www.sgex.eb.mil.br/sg8/006_outras_publicacoes/04_planos/port_n_1968_cmde_eb_03dez2019.html. Accessed: 09 July 2024.
- Fan, C., Yi, J., Tao, J., Tian, Z., Liu, B., and Wen, Z. (2021). Gated recurrent fusion with joint training framework for robust end-to-end speech recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:198–209. DOI: <https://doi.org/10.1109/TASLP.2020.3039600>.
- Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, page 369–376, New York, NY, USA. Association for Computing Machinery. DOI: <https://doi.org/10.1145/1143844.1143891>.
- Gris, L. R. S. (2021). *Reconhecimento de voz utilizando Wav2Vec 2.0 para o português brasileiro*. Bachelor's thesis, Federal University of Technology - Paraná.
- Gris, L. R. S., Casanova, E., de Oliveira, F. S., da Silva Soares, A., and Candido Junior, A. (2022). Brazilian portuguese speech recognition using wav2vec 2.0. In Pinheiro, V., Gamallo, P., Amaro, R., Scarton, C., Batista, F., Silva, D., Magro, C., and Pinto, H., editors, *Computational Processing of the Portuguese Language*, pages 333–343, Cham. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-030-98305-5_31.
- Hannun, A. (2017). Sequence modeling with CTC. <https://distill.pub/2017/ctc>. Accessed: 09 July 2024.
- Hannun, A. Y., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., and Ng, A. Y. (2014). Deep speech: Scaling up end-to-end speech recognition. *CoRR*, abs/1412.5567. DOI: <https://doi.org/10.48550/arXiv.1412.5567>.
- Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics. DOI: <https://dl.acm.org/doi/10.5555/2132960.2132986>.
- Huang, X., Acero, A., Hon, H.-W., and Reddy, R. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, USA, 1st edition. DOI: <https://dl.acm.org/doi/book/10.5555/560905>.
- ITU, I. T. U. (1978-1982-1992). *Recommendation F.520-2. Use of High Frequency Ionospheric Channel Simulators*, volume III. Recommendations and Reports of the CCIR, Genova. DOI: No DOI available.
- Jurafsky, D. and Martin, J. (2021). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall, 3 edition. DOI: <https://dl.acm.org/doi/book/10.5555/555733>.
- Li, J., Deng, L., Gong, Y., and Haeb-Umbach, R. (2014). An overview of noise-robust automatic speech recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 22(4):745–777. DOI: <https://doi.org/10.1109/TASLP.2014.2304637>.

- Maruf, M. R., Faruque, M. O., Mahmood, S., Nelima, N. N., Muhtasim, M. G., and Pervez, M. A. (2020). Effects of noise on RASTA-PLP and MFCC based bangla ASR using CNN. In *2020 IEEE Region 10 Symposium (TENSYPMP)*, pages 1564–1567. DOI: <https://doi.org/10.1109/TENSYPMP50017.2020.9231034>.
- Menêses Santos, R. (2016). *Uma abordagem hibrida CNN-HMM para reconhecimento de fala tolerante a ruídos de ambiente*. Dissertation, Universidade Federal de Sergipe.
- Mozilla Corporation (2020). Welcome to DeepSpeech’s documentation! <https://deepspeech.readthedocs.io/en/r0.9/>. Accessed: 09 July 2024.
- Pervaiz, A., Hussain, F., Israr, H., Tahir, M. A., Raja, F. R., Baloch, N. K., Ishmanov, F., and Zikria, Y. B. (2020). Incorporating noise robustness in speech command recognition by noise augmentation of training data. *Sensors*, 20(8). DOI: <https://doi.org/10.3390/s20082326>.
- Prodeus, A. and Kukharicheva, K. (2016). Training of automatic speech recognition system on noised speech. In *2016 4th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC)*, pages 221–223. DOI: <https://doi.org/10.1109/MSNMC.2016.7783147>.
- Prodeus, A. and Kukharicheva, K. (2017). Automatic speech recognition performance for training on noised speech. In *2017 2nd International Conference on Advanced Information and Communication Technologies (AICT)*, pages 71–74. DOI: <https://doi.org/10.1109/AICT.2017.8020068>.
- Quintanilha, I. M., Netto, S. L., and Biscainho, L. P. (2020). An open-source end-to-end asr system for brazilian portuguese using dnns built from newly assembled corpora. *Journal of Communication and Information Systems*, 35(1):230–242. DOI: <https://doi.org/10.14209/jcis.2020.25>.
- Radford, A., Kim, J. W., Xu, T., Brockman, G., Mcleavey, C., and Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR. DOI: <https://dl.acm.org/doi/10.5555/3618408.3619590>.
- Ribeiro, F. C. (2019). *Reconhecimento de comandos de voz em português brasileiro em ambientes ruidosos usando laringofone*. PhD thesis, Universidade Federal do Ceará.
- Scart, L. G., Vassallo, R. F., and Samatelo, J. L. A. (2022). Aplicação de um modelo neural para reconhecimento de fala em Áudios com características de comunicação via rádio. In *Anais do CBA 2022: XXIV Congresso Brasileiro de Automática*. DOI: No DOI available.
- Shimada, K., Bando, Y., Mimura, M., Itoyama, K., Yoshii, K., and Kawahara, T. (2019). Unsupervised speech enhancement based on multichannel nmf-informed beamforming for noise-robust automatic speech recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 27(5):960–971. DOI: <https://ieeexplore.ieee.org/document/8673623>.
- Wang, Q., Wang, S., Ge, F., Han, C. W., Lee, J., Guo, L., and Lee, C.-H. (2018). Two-stage enhancement of noisy and reverberant microphone array speech for automatic speech recognition systems trained with only clean speech. In *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 21–25. DOI: <https://doi.org/10.1109/ISCSLP.2018.8706595>.
- Wang, Z.-Q. and Wang, D. (2016). A joint training framework for robust automatic speech recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 24(4):796–806. DOI: <https://doi.org/10.1109/TASLP.2016.2528171>.
- Yilmaz, E., Gemmeke, J. F., and Van Hamme, H. (2014). Noise robust exemplar matching using sparse representations of speech. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 22(8):1306–1319. DOI: <https://doi.org/10.1109/TASLP.2014.2329188>.
- Zhang, Y. and Li, J. (2023). Birdsoundsdenoising: Deep visual audio denoising for bird sounds. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2247–2256. DOI: <https://doi.org/10.1109/WACV56688.2023.00228>.