****

**Ministry of Higher Education**

**And Scientific Research**

**al-Farabi University College**

**Computer Engineering Department**

**Speech Denoising Using Deep Learning**

A project submitted to the Department of Computer

Engineering in partial fulfillment of

B.Sc. degree in Computer Engineering

By

**Huumam F. Talib Mun'im**

**Farooq Ali Muhi**

Supervised by

**A.L. Akram Jabar**

July, 2021

**Supervisor Certification**

I certify that the preparation of this project entitled (**Speech Denoising Using Deep Learning** ) was prepared by (**Huumam F. Talib Mun'im** and **Farooq Ali Muhi**) under my supervision at the Computer Engineering Department, al-Farabi University College in partial fulfillment of the requirements for the degree of B.Sc. in Computer Engineering.

Name:

Scientific Degree:

Date:

**Examination Committee Certification**

We certify that we have read this project entitled (**Speech denoising using deep learning**), and as an examining committee examined the student (**Huumam F. Talib Mun'im** and **Farooq Ali Muhi**) in its contents and in our opinion, it meets the standards of the B.Sc. in Computer Engineering.

Signature: Signature:

Name: Name:

Scientific Degree: Scientific Degree:

Date: Date:

(Member) (Member)

Signature: Signature:

Name: Name:

Scientific Degree: Scientific Degree:

Date: Date:

Head of Computer Engineering (Chairman)

**Acknowledgement**

Thanks to God

For all things and peace, be upon his prophet

We would like to express our deepest gratitude and thanks to our supervisor **A.L. Akram Jabar** for hisadvice, guidance, helpful suggestion and encouragement throughout the period that we have worked under her supervision.

**List of Contents**

|  |  |
| --- | --- |
| Contents | Page No. |
| ABSTRACT |  |
| CHAPTER ONE (Introduction) |  |
| 1.1 Introduction |  |
| CHAPTER TWO (Theoretical Review) |  |
| 2.1 Theoretical Matter |  |
| CHAPTER THREE |  |
| 3.1 Project's Main Code |  |
| 3.2 Project's Result |  |
| CHAPTER FOUR |  |
| 4.1 Conclusion |  |
| REFERENCES |  |
| APPENDIX |  |

|  |  |  |
| --- | --- | --- |
| List of Figures | Name | Page No. |
| CHAPTER ONE (Introduction) |  |  |
| Fig (2.1) |  |  |
| Fig (2.2) |  |  |
| Fig (3.1) |  |  |

|  |  |  |
| --- | --- | --- |
| List of Tables | Name | Page No. |
| CHAPTER ONE (Introduction) |  |  |
| Table (2.1) |  |  |

**Abstract**

We present an end-to-end deep learning approach to de-noising speech signals by processing the raw waveform directly. Given input audio containing speech corrupted by an additive background signal, the system aims to produce a processed signal that contains only the speech content. Recent approaches have shown promising results using various deep network architectures. In this paper, we propose to train a fully-convolutional context aggregation network using a deep feature loss, and that loss is based on comparing the internal feature activations in a different network, trained for acoustic environment detection and domestic audio tagging. Our approach outperforms the state-of-the-art in objective speech quality metrics and in large-scale perceptual experiments with human listeners. It also outperforms an identical network trained using traditional regression losses. The advantage of the new approach is particularly pronounced for the hardest data with the most intrusive background noise, for which de-noising is most needed and most challenging.

**Chapter One: Introduction**

**1.1 Introduction**

Speech de-noising (or enhancement) refers to the removal of background content from speech signals. Due to the ubiquity of this audio degradation, de-noising has a key role in improving human-to-human (e.g., hearing aids) and human-to-machine (e.g., automatic speech recognition) communications. A particularly challenging but common form of the problem is the under-determined case of single-channel speech de-noising, due to the complexity of speech processes and the unknown nature of the non-speech material. The complexity is further compounded by the nature of the data, since audio material contains a high density of data samples (e.g., 16,000 samples per second). Challenges also arise in mediated human-to human communication, as perception mechanisms can make small errors still noticeable by the average user. In this work, we present an end-to-end deep learning approach to speech denoising. Our approach trains a fully convolutional de-noising network using a deep feature loss. To compute the loss between two waveforms, we apply a pre-trained audio classification network to each waveform and compare the internal activation patterns induced in the network by the two signals. This compares a multitude of features at different scales in the two waveforms. We perform extensive experiments that compare the presented approach to recent state-of-the-art end-to-end deep learning techniques for de-noising. Our approach outperforms them in both objective speech quality metrics and large-scale perceptual experiments with human listeners, which indicate that our approach is more effective than the baselines. The advantages of the presented approach are particularly pronounced for the hardest, noisiest inputs, for which denoising is most challenging.

**Aim of projects**

Denoising aims to reproduce clean speech from noise-polluted signals, which is crucial for various applications, such as automatic speech recognition (ASR) and hearing aids.

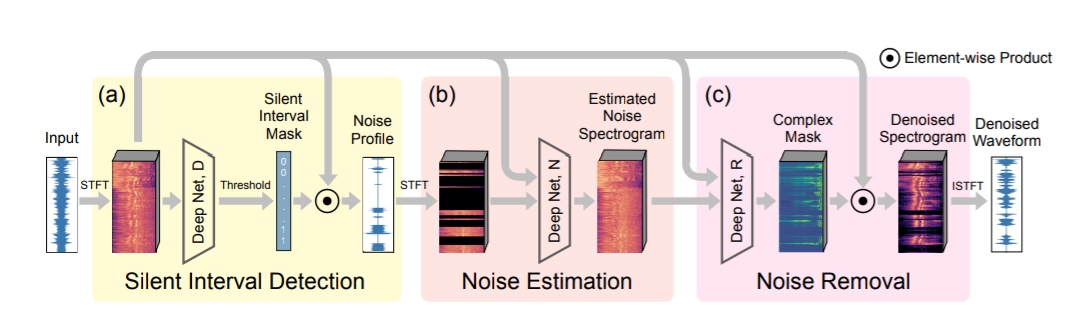
**Chapter Two**

**2.1 Introduction to Speech Denoising**

Speech denoising is a long-standing problem. Given an input noisy signal, we aim to filter out the undesired noise without degrading the signal of interest. You can imagine someone talking in a video conference while a piece of music is playing in the background. In this situation, a speech denoising system has the job of removing the background noise in order to improve the speech signal. Besides many other use cases, this application is especially important for video and audio conferences where noise can significantly decrease speech intelligibility.

Classical solutions for speech denoising usually use generative modeling. The idea is to use statistical methods like Gaussian Mixtures, to build a model of the noise of interest. Then, we can use it to recover the source (clean) audio from the input noisy signal. However, recent development has shown that in situations where data is plenty available, deep learning often outperforms such solutions.

In this article, we tackle the problem of speech denoising using Convolutional Neural Networks (CNNs). Given a noisy input signal, we aim to build a statistical model that can extract the clean signal (the source) and return it to the user. Here, we focus on source separation of regular speech signals from 10 different types of noise often found in an urban street environment.

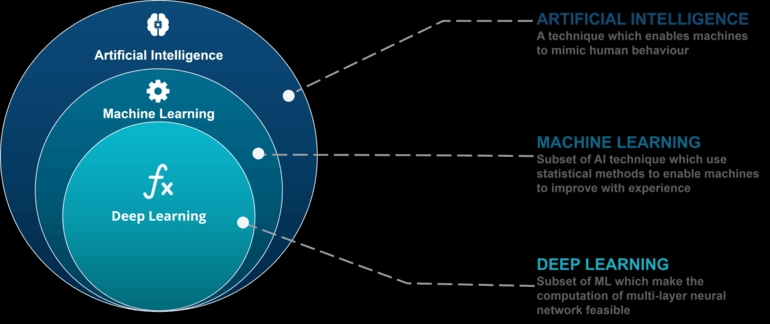
**2.1.1 Speech denoising**

**Fig (2-1): Speech Denoising**

Speech denoising [53] is a fundamental problem studied over several decades. Spectral subtraction [7, 98, 6, 72, 79] estimates the clean signal spectrum by subtracting an estimate of the noise spectrum from the noisy speech spectrum. This classic method was followed by spectrogram factorization methods [84]. Wiener filtering [80, 40] derives the enhanced signal by optimizing the Mean-square error. Other methods exploit pauses in speech, forming segments of low acoustic Energy where noise statistics can be more accurately measured [13, 57, 86, 15, 75, 10, 11]. Statistical Model-based methods [14, 34] and subspace algorithms [12, 16] are also studied. Our model has three components: (a) one that detects silent intervals over time, and outputs a noise profile observed from detected silent intervals. (b) another that estimates the full noise profile, and © yet another that cleans up the input signal Applying neural networks to audio denoising dates back to the 80s [88, 69]. With increased computing Power, deep neural networks are often used [104, 106, 105, 47]. Long short-term memory networks (LSTMs) [35] are able to preserve temporal context information of the audio signal [52], leading to Strong results [56, 90, 100]. Leveraging generative adversarial networks (GANs) [33], methods such as [70, 71] have adopted GANs into the audio field and have also achieved strong performance audio signal processing methods operate on either the raw waveform or the spectrogram by Short-Time Fourier Transform (STFT). Some work directly on waveform [23, 68, 59, 55], and others use Wave net [91] for speech denoising [74, 76, 30]. Many other methods such as [54, 94, 61, 99, 46, 107, 9] work on audio signal’s spectrogram, which contains both magnitude and phase information. There are works discussing how to use the spectrogram to its best potential [93, 67], while one of the disadvantages is that the inverse STFT needs to be applied. Meanwhile, there also exist Works [51, 29, 28, 95, 19, 101, 60] investigating how to overcome artifacts from time aliasing. Speech denoising has also been studied in conjunction with computer vision due to the relations Between speech and facial features [8]. Methods such as [31, 26, 3, 36, 32] utilize different network Structures to enhance the audio signal to the best of their ability. Adeel et al. [1] even utilize lip-reading to filter out the background noise of a speech.

**2.1.2 Deep learning**

is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.[1][2][3] Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.[4][5][6][7] Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogue.[8][9][10] The adjective "deep" in deep learning refers to the use of multiple layers in the network. Early work showed that a linear perceptron cannot be a universal classifier, but that a network with a nonpolynomial activation function with one hidden layer of unbounded width can. Deep learning is a modern variation which is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed connectionist models, for the sake of efficiency, trainability and understandability, whence the "structured" part.



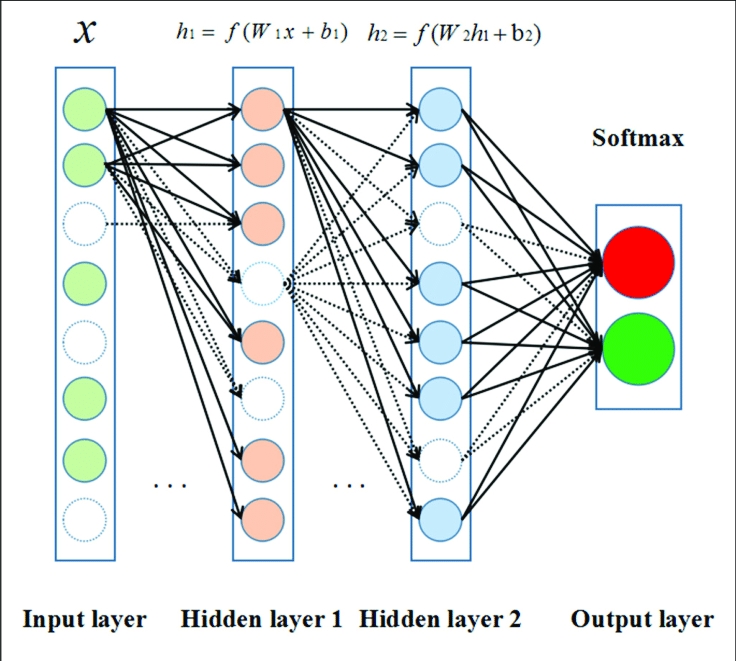
**Fig (2-2): Deep Learning**

**Artificial neural networks (ANNs)**

or connectionist systems are computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve their ability) to do tasks by considering examples, generally without task-specific programming. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the analytic results to identify cats in other images. They have found most use in applications difficult to express with a traditional computer algorithm using rule-based programming. An ANN is based on a collection of connected units called artificial neurons, (analogous to biological neurons in a biological brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have state, generally represented by real numbers, typically between 0 and 1. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream. Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input), to the last (output) layer, possibly after traversing the layers multiple times. The original goal of the neural network approach was to solve problems in the same way that a human brain would. Over time, attention focused on matching specific mental abilities, leading to deviations from biology such as backpropagation, or passing information in the reverse direction and adjusting the network to reflect that information. Neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis. As of 2017, neural networks typically have a few thousand to a few million units and millions of connections. Despite this number being several orders of magnitude less than the number of neurons on a human brain, these networks can perform many tasks at a level beyond that of humans (e.g., recognizing faces, playing "Go"[111]

**A deep neural network (DNN)**

**Fig (2-3): Neural Network**



is an artificial neural network (ANN) with multiple layers between the input and output layers.[13][2] There are different types of neural networks but they always consist of the same components: neurons, synapses, weights, biases, and functions.[112] These components functioning similar to the human brains and can be trained like any other ML algorithm.[citation needed] For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence the name "deep" networks. DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives.[113] The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network.[13] For instance, it was proved that sparse multivariate polynomials are exponentially easier to approximate with DNNs than with shallow networks.[114] Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures, unless they have been evaluated on the same data sets. DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights", to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network did not accurately recognize a particular pattern, an algorithm would adjust the weights.[115] That way the algorithm can make certain parameters more influential, until it determines the correct mathematical manipulation to fully process the data. Recurrent neural networks (RNNs), in which data can flow in any direction, are used for applications such as language modeling.[116][117][118][119][120] Long short-term memory is particularly effective for this use.[61][121] Convolutional deep neural networks (CNNs) are used in computer vision.[122] CNNs also have been applied to acoustic modeling for automatic speech recognition (ASR).[78]

**Deep Learning Architecture**

Our Deep Convolutional Neural Network (DCNN) is largely based on the work done by A fully convolutional neural network for speech enhancement. Here, the authors propose the Cascaded Redundant Convolutional Encoder-Decoder Network (CR-CED). The model is based on symmetric encoder-decoder architectures. Both components contain repeated blocks of Convolution, ReLU and Batch Normalization. In total, the network contains 16 of such blocks, which add up to 33K parameters. Also, there are skip connections between some of the encoder and decoder blocks. Here the feature vectors from both components are combined through addition. Very much like ResNets, the skip connections speed up convergence and reduces the vanishing of gradients. Another important characteristic of the CR-CED network is that convolution is only done in 1 dimension. More specifically, given an input spectrum of shape (129 x 8), convolution is only performed in the frequency axis (i.e the first one). This ensures that the frequency axis remains constant during forwarding propagation. The combination of a small number of training parameters and model architecture makes this model a lightweight option for fast execution on mobile or edge devices. Once the network produces an output estimate, we optimize (minimize) the mean squared difference (MSE) between the output and the target (clean audio) signals. A. De-noising Network Let x be an audio signal corresponding to speech ß that is corrupted by an additive background signal n so that x = ß + n. Our goal is to find a de-noising operator g such that g(x) ≈ ß. We use a fully-convolutional network architecture based on context aggregation networks [26]. The output signal is synthesized sample by sample as we slide the network along the input. Context aggregation networks have been previously used in the Wave Net architecture for speech synthesis [27]. Our architecture is simpler than Wave Net – no skip connections across layers, no conditioning, no gated activations – while our loss function is more advanced, as described in Section II-B.

a) Context aggregation:

Our network consists of 16 convolutional layers. The first and last layers (the degraded input signal and the enhanced output signal, respectively are 1-dimensional tensors of dimensionality N ×1. The number of samples N in the input signal varies and is not given in advance. The signal sampling frequency fs is assumed to be 16 kHz. Each intermediate layer is a 2-dimensional tensor of dimensionality N ×W, where W is the number of feature maps in each layer. (We set W = 64.) The content of each intermediate layer is computed from the previous layer via a dilated convolution with 3 × 1 convolutional kernels [26] followed by an adaptive normalization (see below) and a pointwise nonlinear leaky rectified linear unit (LReLU) [28] max(0.2x, x). Because of the normalization, no bias term is used for the intermediate layers. We zero-pad all layers so that their “effective” length is constant at N. Our network is then trained to handle the beginning and end of audio files even when speech content is near the sequence edges. The dilation operator aggregates long-range contextual information without changing sampling frequency across layers [26], [27]. Here, we increase the dilation factor exponentially with depth from 2 0 for the 1st intermediate layer to 2 12 for the 13th one. We do not use dilation for the 14th and last one. For the output layer, we use a linear transformation (1 × 1 convolution plus bias with no normalization and no nonlinearity) to synthesize the sample of the output signal. The receptive field of the pipeline is 2 14 + 1 samples, i.e., about 1 s of audio for fs = 16 kHz. We thus expect the system to capture context on the time scales of spoken words. A similar network architecture was shown to be advantageous in terms of compactness and runtime for image processing [29].

b) Adaptive normalization:

The adaptive normalization operator used in our network matches the one proposed in [29] and improves performance and training speed. It adaptively combines batch normalization and identity mapping of the input x as the weighted sum αkx+βkBN(x) (where αk, βk ∈ R are scalar weights for the k-th layer and BN is the batch normalization operator [30]). The weights α, β are. learned by backpropagation as network parameters B. Feature loss in our experiments, simple training losses (e.g., L 1) led to noticeably degraded output quality at lower signal-to-noise ratios (SNRs). The network seemed to improperly process low energy speech information of perceptual importance. Instead, we train the denoising network using a deep feature loss that penalizes differences in the internal activations of a pretrained deep network that is applied to the signals being compared. By the nature of layered networks, feature activations at different depths in the loss network correspond to different time scales in the signal. Penalizing differences in these activations thus compares many features at different audio scales. In computer vision, there are standard classification networks such as VGG-19 [31], pretrained on standard clas sification datasets such as ImageNet [32]. Such standard classification networks do not exist in the audio processing field yet, so we design and train our own feature loss network.

a) Feature loss network:

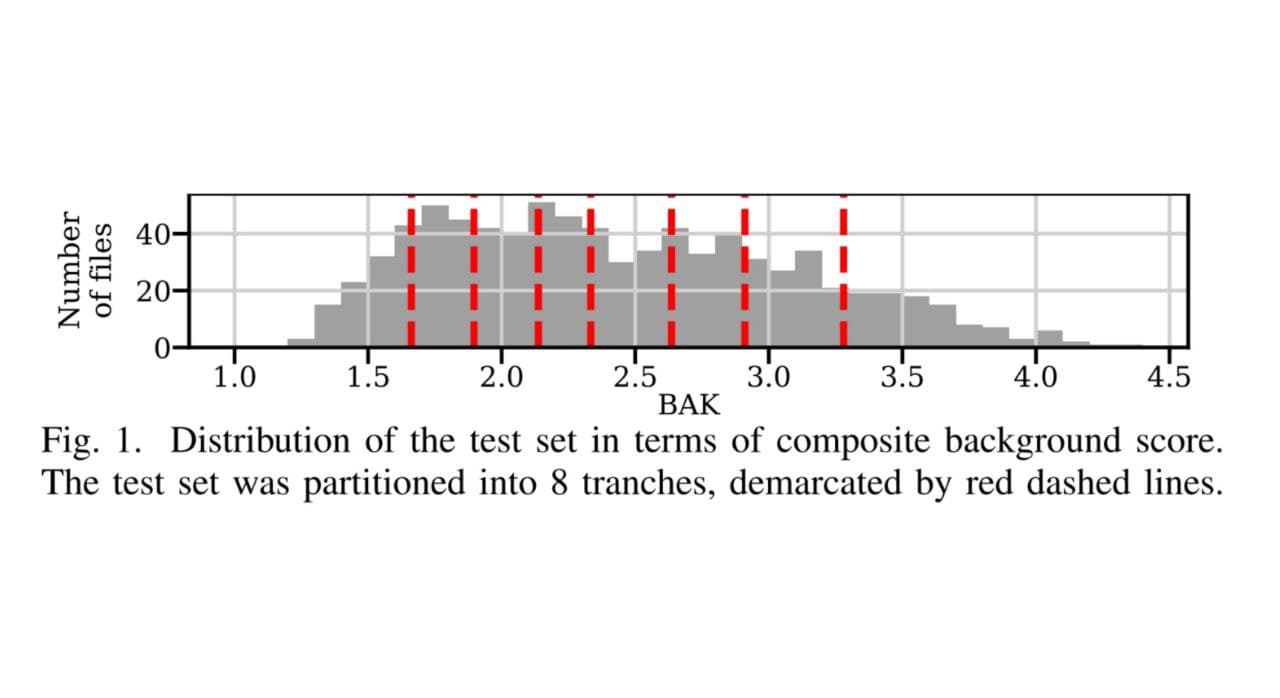
We design a simple audio classification network inspired by the VGG architecture in computer vision [31], since it is known as a particularly effective feature loss architecture [25]. The network consists of 15 convolutional layers with 3×1 kernels, batch normalization, LReLU units, and zero padding. Each layer is decimated by 2, halving the length of the subsequent layer compared to the preceding one. The number of channels is doubled every 5 layers, with 32 channels in the first intermediate layer. Each channel in the last feature layer is average-pooled to yield the output feature vector. The receptive field is 2 15 − 1 samples. We train the network using backpropagation by feeding its output vector as features to one or more logistic classifiers with a cross-entropy loss for one or more classification tasks.

b) Denoising loss function:

Let Φ m be the m-th feature layer of the feature loss network, with layers at different depths corresponding to features with various time resolutions. The feature loss function is defined as a weighted L 1 loss on the difference between the feature activations induced in different layers of the network by the clean reference signal ß and the output g(x) of the denoising network being trained:

where θ are the parameters of the denoising network. The weights λm are set to balance the contribution of each layer to the loss. They are set to the inverse of the relative values of kΦ m(ß) − Φ m (g (x; θ)) k1 after 10 training epochs. (For these first 10 epochs, the weights are set to 1.)

A. Baselines As baselines, we use a Wiener filtering pipeline with a priori noise SNR estimation (as implemented in [39]), and two recent state-of-the-art methods that use deep networks to perform end-to-end denoising directly on the raw waveform: the Speech Enhancement Generative Adversarial Network (SEGAN) [21] and a WaveNet-based network [20]. This last one is designed around minor modifications to the architecture in [27]. It uses stacked context aggregation modules with gated activation units, skip connections, and a conditioning mechanism. The modifications include training with a regression loss (L 1 on the raw waveform) rather than a classification loss. The number of layers is larger than in our network (30), while the receptive field is smaller (3 · 2 11 samples), capturing contextual information on more limited time scales. The network architecture is also distinctly more complex than ours. For both deep learning baselines, we use the code and models published by their respective authors. These models are optimized by their authors on the exact same training dataset, allowing fair comparison.

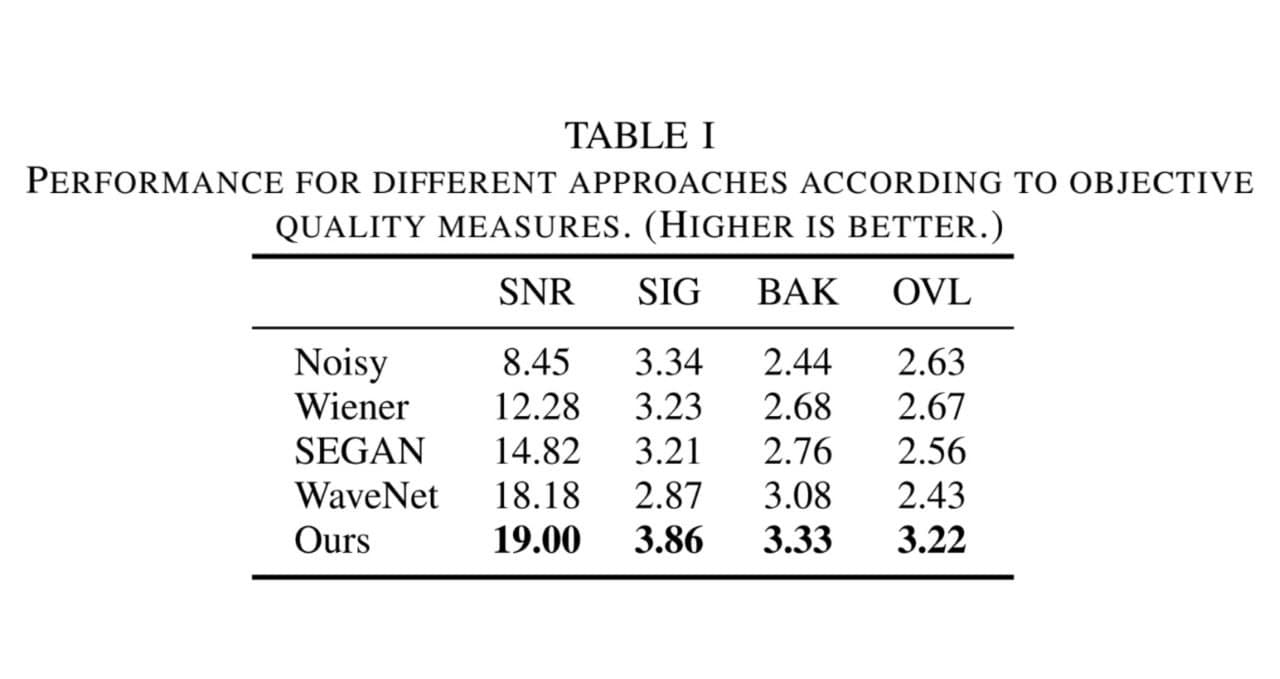


**Fig (2-4): Distribution of the test set**

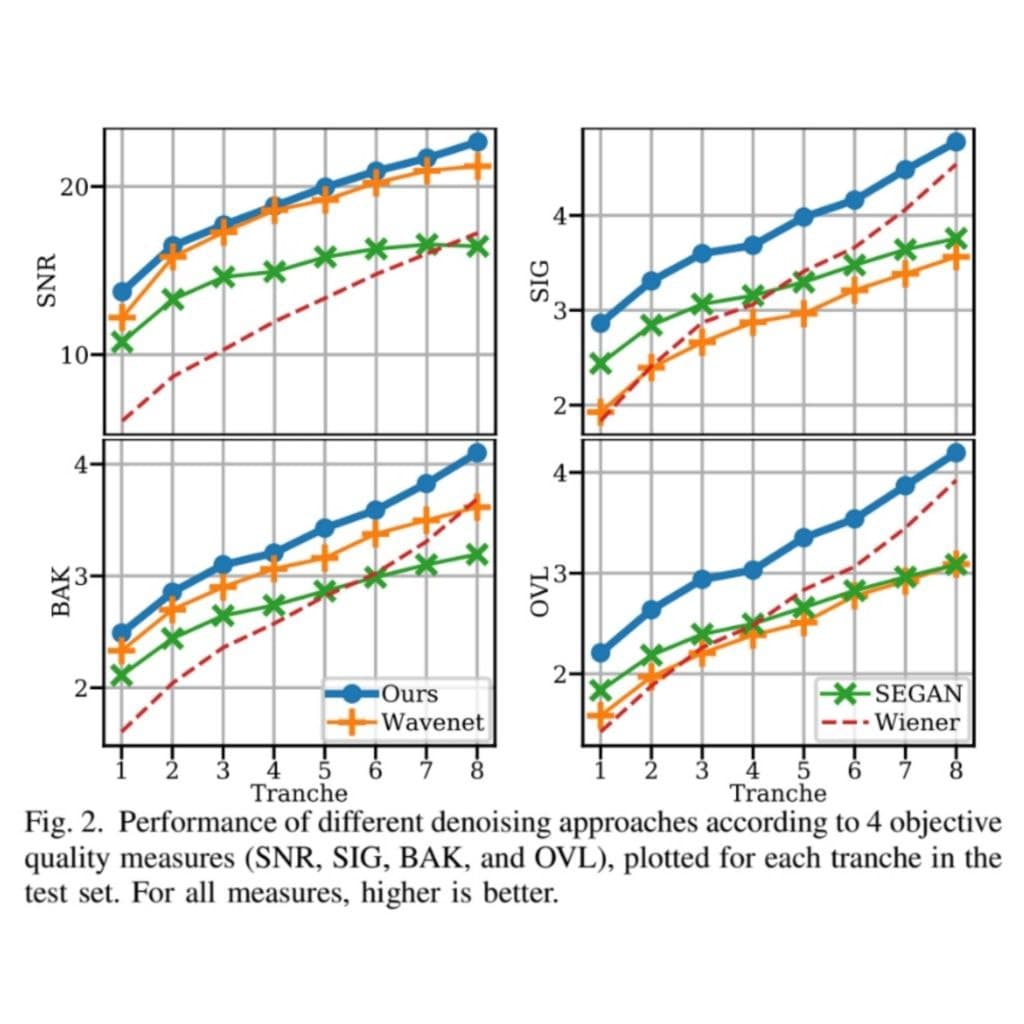
C. Quantitative measures

a) Objective quality metrics:

To evaluate each system, we compare its output to the ground-truth speech signal (i.e., the clean speech alone). The common metrics to measure speech quality given ground-truth are compared in [1]. We use here the composite scores from [39] that were found to be best correlated with human listener ratings. These consist of the overall (OVL), the signal (SIG), and the background (BAK) scores, each on a scale from 1.0 to 5.0, and corresponding respectively to the measure of overall signal quality, the measure of quality when considering speech signal degradation alone, and the measure of quality when considering background signal intrusiveness alone [40]. We also report the SNR [41], as a raw measure of the relative energies of the residual background and the speech in a given signal, quantified in decibel (dB). We use the implementations in [1]. For all metrics, higher scores denote better performance. The test dataset is divided into 4 mixing SNR subgroups (see Section IV-B). We argue that the dataset should be rather considered as a continuous distribution of degradation, since SNR correlates poorly with human perception of the degradation level [1]. The continuum of degradation levels is better represented in the distribution of the background intrusiveness BAK score. (The SIG score is less informative since the undistorted speech signal is added.) To evaluate performance as a function of input degradation magnitude, we partition the test set into 8 tranches of equal size, corresponding to the 8 octiles of the BAK score distribution as shown in Figure 1, with tranches representing a different denoising difficulty.



**Table (2-1): Performance**



**Fig (2-5): Performance**

b) Results:

Table I reports these metrics for our approach and the baselines, evaluated over the test set. Our method outperforms all the baselines according to all measures by a comfortable margin. The plots in Figure 2 further show that our network yields the best quality for all levels of background intrusiveness separated in tranches, with a particularly significant margin according to perceptually-motivated composite measures. Table II shows the benefit of using a feature loss compared to training the same denoising network, by the same procedure on the same data, using an L 1 or an L 2 loss. Training with a feature loss outperforms networks trained with other losses. In particular, while an L 1 loss achieves a similar SNR score as our feature loss, the feature loss shows definite improvement for the BAK and OVL metrics. It also scores well for the SIG metric, especially in the noisier tranches, demonstrating the ability to capture meaningful features when important cues are hidden in the noise.

**Chapter Three:** **Experimental Results**

**3.1 Best Deep Learning platforms**

Good data science and machine-learning platform should offer data scientists the building blocks for creating a solution to a data science problem. It should also provide these experts with an environment where they can incorporate the solutions into products and business processes. The platform needs to provide data scientists with all the support they need when carrying out data and analytics tasks. These tasks encompass visualization, interactive

exploration, deployment, performance engineering data preparation and data access.

There are a lot of machine-learning platforms include: Alteryx Analytics, H2O.ai, KNIME Analytics Platform, RapidMiner, SAS, TIBCO Software, Domino Data Science Platform, and MathWorks’ MATLAB and Simulink. We used the latter in our experiment for many reasons.

**3.2 Why MATLAB**

MATLAB programming platform has numerous advantages over other techniques or languages. The fundamental structure has a basic data element in a matrix. A simple integer is recognized as a matrix of one row and one column. Different mathematical methods that work on arrays or matrices are built into the MATLAB environment. For instance, cross-products, dot-products, determinants, inverse matrices. Vectorized operations such as adding two arrays together need only one command, instead of a for or while loop. The graphical output is optimized for communication. Users can plot their data very simply, and then modify colors, sizes, scales, etc. by handling the graphical interactive tools. MATLAB’s functionality can be considerably expanded by the addition of toolboxes. These are sets of specific functions that provided more specialized functionality. These features make the programming language very effective for implementing deep learning.

**3.3 Experiment steps**

**3.3.1 Experiment’s Main Code**

Note: all the text written after % is a comment not part of the code.

%Code Start

clc %Clears all the text from the Command Window

clear all %To clear all variables from the current workspace

close all % Closes all open MATLAB figure windows

% Speech signal sampled at 8 kHz.

% Reads data from the file, and returns sampled data, cleanAudio, and a sample rate for that data, fs.

[cleanAudio,fs] = audioread('SpeechDFT-16-8-mono-5secs.wav');

% Convert matrix of signal data to sound

sound(cleanAudio,fs)

%Put the noise audio inside noise variable

noise = audioread('WashingMachine-16-8-mono-1000secs.mp3');

% Extract a noise segment from a random location in the noise file

ind = randi(numel(noise) - numel(cleanAudio) + 1, 1, 1);

noiseSegment = noise(ind:ind + numel(cleanAudio) - 1);

noiseSegment= noiseSegment';

speechPower = sum(cleanAudio.^2);

noisePower = sum(noiseSegment.^2);

noisyAudio = cleanAudio + sqrt(speechPower/noisePower) \* noiseSegment;

sound(noisyAudio,fs)

t = (1/fs) \* (0:numel(cleanAudio)-1);

subplot(2,1,1)

plot(t,cleanAudio)

%Set the title of up output

title('Clean Audio')

grid on

subplot(2,1,2)

plot(t,noisyAudio)

%Set the title of down output

title('Noisy Audio')

xlabel('Time (s)')

grid on

%Code End

**3.3.1 Implementation of Experiment**

1. write the code above into the MATLAB platform, As shown in the figure (3-1) below.

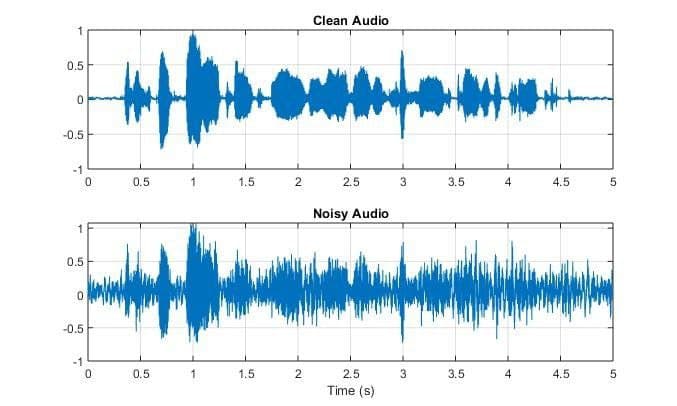
And run the code to get the result.

Graphical user interface, text, application, Word

Description automatically generated

**Fig (3-1): Write code in MATLAB**

2. Choose two audio files, one with noise and the other without and the output signal will be Noise-free sound wave. As shown in the figure (3-2) below.



**Fig (3-2): Result**

**Chapter Four: Future Work and Conclusion**

**4.1 Conclusion**

We presented an end-to-end speech denoising pipeline that uses a fully-convolutional network, using a deep feature loss network pretrained on several relevant audio classification tasks for training. This approach allows the denoising system to capture speech structure at various scales and achieve better denoising performance without added complexity in the system itself or expert knowledge in the loss design. Experiments demonstrate that our approach significantly outperforms recent state-of-the-art baselines according to objective speech quality measures as well as large-scale perceptual experiments with human listeners. In particular, the presented approach is shown to perform much better in the noisiest conditions where speech denoising is most challenging. Our paper validates the combined use of convolutional context aggregation networks and feature losses to achieve state-of-the-art performance.

**4.2 Future Work**

It is possible to improve the project in the future with many ideas like: instead of merging a noise-free audio wave with noise audio and then denoising it.

Corrupted audio waves are used, such as old or damaged recordings, and the noise is removed from them to make them clear and understandable.

**References**

[1] P. C. Loizou, Speech Enhancement: Theory and Practice, 2nd ed. CRC Press, 2013.

[2] M. Bosi and R. E. Goldberg, Introduction to Digital Audio Coding and Standards. Springer, 2002.

[3] P. Smaragdis, C. Fevotte, G. J. Mysore, N. Mohammadiha, and M. Hoffman, “Static and dynamic source separation using nonnegative factorizations: A unied view,” IEEE Signal Processing Magazine, vol. 31, no. 3, 2014.

[4] Y. Wang and D. Wang, “Cocktail party processing via structured prediction,” in Neural Information Processing Systems (NIPS), 2012.

[5] X. Lu, Y. Tsao, S. Matsuda, , and C. Hori, “Speech enhancement based on deep denoising autoencoder,” in Interspeech, 2013.

[6] A. Narayanan and D. Wang, “Ideal ratio mask estimation using deep neural networks for robust speech recognition,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013.

[7] F. Weninger, J. R. Hershey, J. L. Roux, and B. Schuller, “Discriminatively trained recurrent neural networks for single-channel speech separation,” in IEEE Global Conference on Signal and Information Processing, 2014.

[8] Y. Xu, J. Du, L.-R. Dai, , and C.-H. Lee, “A regression approach to speech enhancement based on deep neural networks,” IEEE/ACM Transactions on Audio, Speech and Language Processing, vol. 23, no. 1, 2015.

[9] A. Kumar and D. Florencio, “Speech enhancement in multiple-noise conditions using deep neural networks,” arXiv:1605.02427, 2016.

[10] X.-L. Zhang and D. Wang, “A deep ensemble learning method for monaural speech separation,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 5, 2016.

[11] J. Chen and D. Wang, “Long short-term memory for speaker generalization in supervised speech separation,” Journal of the Acoustical Society of America, vol. 141, no. 6, 2017.

[12] J. L. Roux and E. Vincent, “Consistent Wiener ltering for audio source separation,” IEEE Signal Processing Letters, vol. 20, no. 3, 2013.

[13] F. G. Germain, G. J. Mysore, and T. Fujioka, “Equalization matching of speech recordings in real-world environments,” in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2016.

[14] T. Gerkmann, M. Krawczyk-Becker, and J. L. Roux, “Phase processing for single-channel speech enhancement: History and recent advances,” IEEE Signal Processing Magazine, vol. 32, no. 2, 2015.

[15] Y. Wang and D. Wang, “A deep neural network for time-domain signal reconstruction,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015.

[16] H. Erdogan, J. R. Hershey, S. Watanabe, and J. L. Roux, “Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015.

[17] D. S. Williamson and D. Wang, “Time-frequency masking in the complex domain for speech dereverberation and denoising,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 7, 2017.

[18] J. A. Moorer, “A note on the implementation of audio processing by short-term Fourier transform,” in IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2017.

[19] S.-W. Fu, Y. Tsao, X. Lu, and H. Kawai, “Raw waveform-based speech enhancement by fully convolutional networks,” arXiv:1703.02205, 2017.

[20] D. Rethage, J. Pons, and X. Serra, “A WaveNet for speech denoising,” in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2018.

[21] S. Pascual, A. Bonafonte, and J. Serra, “SEGAN: Speech enhancement generative adversarial network,” in Interspeech, 2017.

[22] K. Qian, Y. Zhang, S. Chang, X. Yang, D. Florencio, and M. HasegawaJohnson, “Speech enhancement using Bayesian WaveNet,” in Interspeech, 2017.

[23] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for realtime style transfer and super-resolution,” in European Conference on Computer Vision (ECCV), 2016.

[24] Q. Chen and V. Koltun, “Photographic image synthesis with cascaded renement networks,” in International Conference on Computer Vision (ICCV), 2017.

[25] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in Computer Vision and Pattern Recognition (CVPR), 2018.

[26] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in International Conference on Learning Representations (ICLR), 2016.

[27] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” arXiv:1609.03499, 2016.

[28] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectier nonlinearities improve neural network acoustic models,” in ICML Workshop on Deep Learning for Audio, Speech, and Language Processing, 2013.

[29] Q. Chen, J. Xu, and V. Koltun, “Fast image processing with fullyconvolutional networks,” in International Conference on Computer Vision (ICCV), 2017.

[30] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in International Conference on Machine Learning (ICML), 2015.

[31] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in International Conference on Learning Representations (ICLR), 2015.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, “ImageNet large scale visual recognition challenge,” International Journal on Computer Vision (IJCV), vol. 115, no. 3, 2015.

[33] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, “Detection and classication of acoustic scenes and events: Outcome of the DCASE 2016 challenge,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 2, 2018.

[34] A. Mesaros, T. Heittola, and T. Virtanen, “TUT database for acoustic scene classication and sound event detection,” in European Signal Processing Conference (EUSIPCO), 2016.

[35] P. Foster, S. Sigtia, S. Krstulovic, J. Barker, and M. D. Plumbley, “CHiMe-Home: A dataset for sound source recognition in a domestic environment,” in IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015.

[36] X. Glorot and Y. Bengio, “Understanding the dif culty of training deep feedforward neural networks,” in International Conference on Articial Intelligence and Statistics (AISTATS), 2010.

[37] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in International Conference on Learning Representations (ICLR), 2015.

[38] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, “Investigating RNN-based speech enhancement methods for noise-robust textto-speech,” in ISCA Speech Synthesis Workshop, 2016.

[39] Y. Hu and P. C. Loizou, “Subjective comparison of speech enhancement algorithms,” in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2006.

[40] ITU-T, “Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm,” ITU-T Recommendation P.835, Tech. Rep., 2003.

[41] S. R. Quackenbush, T. P. Barnwell, and M. A. Clements, Objective Measures of Speech Quality. Prentice Hall, 1988.

**Appendix**

This appendix provides additional details on the denoising and feature loss network architectures presented in Section II.

**A. Denoising Network**

a) Layer structure: We denote the 16 (consecutive) network layers by , . . . , . and are 1-dimensional tensors of dimensionality N×1 and correspond to the degraded input signal and the enhanced output signal, respectively. The number of samples N is not given in advance. Each intermediate layer ∈ { , . . . ,} is a 2-dimensional tensor of dimensionality N×W, where W is the width of (i.e., the number of feature maps in) each layer. For k = 1, . . . , 14, the content of each intermediate layer is computed from the previous layer via the operation

where is the -th feature map of layer , j is the j-th feature map of layer , is a learned 3×1 convolutional kernel, is the adaptive normalization operator and Ψ is a pointwise nonlinearity. Because of the presence of adaptive normalization, no bias term is used for these layers. The operator is a dilated convolution [26], i.e.,

The dilation factor for the k-th layer is set at = for k ∈ {1, . . . 13}. Between layer and , we do not use dilation (i.e., = 1). For the output layer , we use a linear transformation (1×1 convolution with no nonlinearity) in order to synthesize the sample of the output signal so that:

where b is a learned bias term. The receptive field of the network is + 1 = 16385 samples. b) Nonlinear units: For the pointwise nonlinearity Ψ, we use the leaky rectified linear unit (LReLU) [28]:

c) Adaptive normalization: corresponds to the adaptive normalization operation described in Section II-A. For k ∈ {1, . . . 13}, the operator adaptively combines batch normalization and identity mapping as

where , βk ∈ ℝ are learned scalar weights and BN is the batch normalization operator [30].

d) Zero padding: Our algorithm uses zero-padding at each layer so that the “effective” length of each layer tensor is constant and identical to N.

e) Training loss: The network is trained through back propagation using our deep feature loss as described in Section II-B (see in particular Equation 1). The feature loss classification network is further detailed in the next section.

**B. Feature Loss Network**

a) Feature layer structure: As mentioned in Section II-B, the network is inspired by the VGG architecture from computer vision. We denote its 15 (consecutive) layers by , . . . , . The first layer is a 1-dimensional tensor of dimensionality N × 1 and corresponds to the input signal. The number of samples N is not given in advance. Each intermediate layer ∈ { , . . . , } is a 2-dimensional tensor of dimensionality ×, where is the width of each layer, set to = 32×2 (i.e., the number of features is doubled every 5 layers).

The content of each intermediate layer is computed from the previous layer through the following operation:

where is the -th feature map of layer prior to the decimation operation, is the j-th feature map of layer , is a learned 3×1 convolutional kernel, BN is the batch normalization operator, and Ψ is the same pointwise linearity as in Equation 5. Because of the presence of batch normalization, no bias term is used for these layers. This is followed by the decimation operation

following which the length of the subsequent layer is half the length of the preceding one. The receptive field of the network is − 1 = 32767 samples. The network is zero-padded as necessary for each layer so that and have the same “effective” length.

b) Classification layer: To perform the -th classification task of interest, we first average-pool each channel in the last feature layer to yield an output feature vector of dimensionality 1×. This vector is fed to a linear layer to form a logit vector of dimensionality 1× (with the number of classes associated with the -th task) such that

Where is a learned scalar weight and is a learned bias term. We finally get the output classification vector of the network through the operation

where ∆ is the logistic nonlinearity associated with the type of multi-label classification for the -th task (i.e., vector softmax nonlinearity if the task asks for a unique label for each audio file, pointwise sigmoid if the task allows for any number of labels for each audio file). is of dimension 1× and its elements are in the range [0, 1].

c) Training loss: Training is done through back propagation using a cross-entropy loss between the vector associated with the current file (for task ) and its corresponding ground truth classification vector (i.e., the vector of dimension 1× in which the c-th element is 1 if the c-th classification label is associated with the file, 0 otherwise).

**ملخص بحث المشروع باللغة العربية**

نستخدم نهج التعلم العميق الشامل وهو نهج يستخدم نظريات تحاكي الخلايا العصبية للإنسان لتقليل الضوضاء من إشارات الصوت عن طريق معالجة شكل الموجة الخام مباشرةً من خلال إدخال صوت يحتوي على كلام تالف بإشارة ضوضاء مضافة، يهدف النظام إلى إنتاج إشارة معالجة تحتوي فقط على محتوى الكلام النقي، ويزداد تعقيد عمليات الكلام والطبيعة غير المعروفة للمواد غير الكلامية بسبب طبيعة البيانات حيث تحتوي المواد الصوتية على كثافة عالية من عينات البيانات و كمثال على ذلك فإن إشارة صوت يمكن أن تحمل 16,000 عينة في الثانية.

يتم تقليل الضوضاء باستخدام شبكة فقدان ميزة عميقة تم تدريبها مسبقاً على العديد من مهام التصنيف الصوتي ذات الصلة بذلك التدريب، يسمح هذا النهج لنظام تقليل الضوضاء بالتقاط بنية الكلام على مستويات مختلفة وتحقيق أداء أفضل لتقليل الضوضاء دون تعقيد إضافي في النظام نفسه، و توضح التجارب أن نهجنا يتفوق بشكل كبير على أحدث خطوط الأساس الحديثة وفقاً لمقاييس جودة الكلام الموضوعية بعد تجريبها على شريحة واسعة النطاق من المستمعين من البشر.