Class-specific and noise-specific speech enhancement approaches

A Thesis

Submitted for the Degree of

Doctor of Philosophy

in the **Faculty of Engineering**

by

Nazreen P.M.

Electrical Engineering

Indian Institute of Science

Bangalore – 560 012 (INDIA)

July 2019

DEDICATED TO

My mother

Signature of the Author: ....

Nazreen P.M.
Dept. of Electrical Engineering
Indian Institute of Science, Bangalore

Signature of the Thesis Supervisor: ....

A.G. Ramakrishnan
Professor
Dept. of Electrical Engineering
Indian Institute of Science, Bangalore

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. A.G. Ramakrishnan for his valuable support and guidance. He was always there to inspire and motivate me throughout my life at IISc. He happens to be an excellent life coach as well, and no matter what my issues were, always stood by my side to motivate me. I would like to thank Dr. Prasanta Kumar Ghosh for all his valuable guidance and advise. I thank the faculty of Electrical and Electrical Communication Engineering for all the excellent courses offered.

I would like to thank my labmates Vijay Girish, Deepak, Anoop, Abhiram, Pathosh, Madhav and Ram Pandey for all those interesting and enlightening discussions. I believe campus life gets its completeness when you make good friends. I am glad that I have made some really good friends. I would like to thank my friends Harinarayanan, who was always there for me as a great guide and partner for discussions, Vijitha, Sivaram, Shafeek, Vandana, Arshed, Sriram, Anoop Thomas, Anjali and Reuben who helped and supported me on several occasions. I would always cherish the best times I had with my friends, the long walks through the campus and those refreshing coffee breaks at Prakruthi canteen. I would like to thank the staff at the office of department of Electrical Engineering who helped to process several academic related paperworks in time.

Finally I am grateful to my mother, my grandparents and my sister for believing in me.

Abstract

Speech enhancement algorithms process the noisy speech signal and try to estimate the clean speech in order to improve its quality or intelligibility. Restoring the required speech from the one corrupted by background noise is still a challenging problem in the field of speech processing. There are a variety of applications for speech enhancement algorithms such as communication, hearing aids, automatic speech recognition (ASR), speech coding, forensic applications and restoration of historic recordings.

In this thesis we propose, implement and analyze various speech sound class-specific and noise-specific enhancement approaches and frame-wise selection methods for class-specific and noise-specific models. For all the work presented in this thesis, we consider a single channel, additive noise framework. Experiments are performed with speech data from TIMIT corpus and noise samples from NOISEX-92 database. As a final, exploratory study, we have recorded traffic noise from 'CV Raman road' and conducted limited experiments on speech corrupted by this noise.

As the first experiment, we have analyzed the performance of our enhancement scheme, where we use various speech-sound class-specific dictionaries to enhance noisy speech. Speech signal is composed of several sounds which can be categorized in various ways, such as manner-of-articulation, place-of-articulation, or phonemes. Some of these classes, such as fricatives, might correlate well with certain noise types more than the other classes. Hence the atoms in a dictionary learned using these classes may represent noise power to varying degrees and consequently result in poor speech reconstruction. By removing the contribution from atoms of these classes that correlate well with noise, one could improve the enhancement performance. One way to achieve this is to learn different dictionaries for different classes and select a particular dictionary for a segment. We explore a class-specific enhancement approach, where we use a sparse coding dictionary based approach to learn dictionaries of various speech classes namely, manner of articulation, place of articulation and phonemes and noises factory2, m109, leopard, babble and volvo. We found that using class-specific dictionaries for enhancing each frame would result in better enhancement than using class-independent dictionaries for all the frames. Initially, a set of labels are obtained by recognizing the speech, enhanced using a class-independent dictionary. Using these approximate labels, the corresponding class-specific dictionaries are used to enhance each frame of the original noisy speech. We use dictionary learning method using approximate KSVD with LARC coding. We have evaluated the SSNR and PESQ measures of the proposed approach using ground truth phoneme labels and found only marginal improvements in most of the cases over class-independent enhancement. However, when we analyze the performance of our various class-specific approaches in terms of phoneme recognition, we obtain performances superior to the class-independent case, even when we use estimated (approximate) labels for enhancement. We have analyzed the performance using manner of articulation (MOA), place of articulation (POA) and phoneme-specific dictionaries. The phoneme-specific dictionary based enhancement outperforms the MOA and POA based schemes in most of the cases.

An error in the estimated class labels in the class-specific approach results in the selection of an erroneous dictionary for enhancement. The joint enhancement-decoding (JED) algorithm that we propose tries to overcome this issue by jointly optimizing the labels for all the frames and the decoding path to improve the phoneme recognition accuracy. The algorithm optimizes over multiple enhanced versions of each frame using different phoneme specific dictionaries and gives the maximum likelihood path of state sequences as well as the best (in the maximum likelihood sense) choice for the enhanced observation sequence as its output. The current noisy speech frame is enhanced by multiple (N) phoneme-specific dictionaries close to the approximate label of that frame. These N enhanced frames are then fed into the JED algorithm. The algorithm accepts these N observations and chooses the best for each frame such that the overall likelihood is maximized to obtain the final recognized labels. The Viterbi decoding algorithm used in speech recognition is integrated with the class label selection to develop the JED algorithm. Experiments are conducted by varying N from 1 to 5 based on the phoneme confusion matrix to find the best value of N that gives the maximum recognition performance for various noises and SNRs. Our experiments show that the recognition performance varies with the number of dictionaries, and in most of the cases, is the best when two or three dictionaries are employed.

We also propose a method of picking the best DNN model in the scenario where multiple noise-specific DNN models are available for enhancement, using the Monte Carlo (MC) dropout proposed by Gal and Ghahramani. MC dropout is a tool for modeling the uncertainty in a DNN, using dropout during inference stage. The conventional dropout of these multiple DNN models is replaced with MC dropout and a measure of the model uncertainty is used for the selection of DNN models. The trace of the covariance matrix (Var) of the output signal vectors, resulting from different MC dropout trials, is used as a measure of the model precision to select one out of multiple models for each frame, using this variance as a proxy for squared error. We find this method to be particularly useful for unseen noisy scenario, where the noise corrupting the test speech is different from those with which the available DNN models are trained. The method performs better than the approach of using a DNN classifier for the selection of noise-specific models for unseen noisy scenario. We observe some promising results in enhancement performance of the algorithm on speech corrupted with a mixture of multiple noises and for the case where random segments of speech are corrupted by different unseen noises. In another significant experiment, we evaluate the performance of our algorithm on real world, traffic noise recorded by us. Our algorithm gives performances superior to classifier-based noise-specific model selection scheme in this case as well. We also explore the use of MC dropout in improving the generalizability of a single DNN model for enhancement when the conventional dropout is replaced by MC dropout. We show that in the case of noisy speech corrupted with unseen noises, MC dropout models can give a better denoised output than conventional dropout models.

Notations

$\mathcal{A}$	active set of dictionary atoms
a	the inner product vector using Gram matrix and w for updating residual coherence
b	scalar used for computing the unit vector w in LARC
C	sparse coefficient matrix
c	total number of dictionaries in each class categories
c^*	selected class label
c_{con}	sparse coefficient vectors of concatenated dictionary D_s and Dx
c^I	sparse coefficient for I^{th} iteration of KSVD
c_x	sparse coefficient vector of noise dictionary
c_o^*	sparse coefficient solution of any general sparse coding problem
c_s	sparse coefficient vector of speech dictionary
D	dictionary matrix concatenating D_s and D_x
$D0$	composite dictionary of D_{ind} and D_x
$D1$	composite dictionary of $D_{c^*}^{MOA}$ and D_x
$D2$	composite dictionary of $D_{c^*}^{POA}$ and D_x
$D3$	composite dictionary of $D_{c^*}^{PHN}$ and D_x
D_s	overcomplete dictionary of speech
D_x	overcomplete dictionary of noise
$D^{(0)}$	initial dictionary for KSVD
D^I	dictionary for I^{th} iteration of KSVD
D_{ind}	class independent speech dictionary
D_c^{MOA}	manner of articulation-specific speech dictionary
D_c^{POA}	place of articulation-specific speech dictionary
D_c^{PHN}	phoneme-specific speech dictionary
$D_i^*;\ 1 \leq i \leq N$	N best dictionaries corresponding to the obtained class labels in best-N scheme
$D_{(:,\mathcal{A})}$, $G_{(:,\mathcal{A})}$	D and Gram matrices defined for set $\mathcal{A}$
E_l	KSVD estimation error for the N examples when l^{th} atom is removed

E_l^R	KSVD error involving samples using atom $D(:,l)$
E_{lg}	Mean square logarithmic error for DNN training
F	The total number of frames
G	Gram matrix for LARC
g	vector obtained using Gram matrix and sign vector for computing the unit vector w
g_1	updated elements $C(l,\mathcal{N})$ in approximate KSVD
h	updated atom $D(:,l)$ in approximate KSVD
j^*	index of atom most coherent to the residue in sparse coding
J	number of forward passes in MC dropout scheme
k	frequency index
L	number of dictionary atoms
l	each atom index of D
M	total number of noise-specific DNN models
N	number of best labels considered for enhancement in each frame for best-N scheme
$ob(\cdot\|q_j)$	observation probability given state q_j
$Q = q_1^*, q_2^* \ldots q_F^*$	state sequence in JED
$\mathbb{L}(X)$	vector space spanned by vectors x_1 and x_2
$\mathcal{N}$	location of the nonzero coefficients in the coding row C(l,n)
R	number of frequency bins
$S(\omega_k)$	STFT of the clean speech
S	magnitude STFT of clean speech
S_f	magnitude STFT of a frame of clean speech
$\hat{S}_k$, S_k	the estimated and reference spectral features, at frequency index k
$\hat{S_f}$	estimate of the STFT of a frame of enhanced speech
$\hat{s}$	estimate of enhanced speech in time domain
$s(m)$	m^{th} sample of the time domain clean speech
sgn	sign of $\mu_{\mathcal{A}}$
t	cardinality constraint for sparse coding problem with l_0 norm
t_1	cardinality constraint for LASSO
$tr(q_k \rightarrow q_j)$	transition probability from states q_k to q_j
U, V^T, Σ	orthonormal matrices and sigma matrix of SVD decomposition
u	equiangular vector in LARC
u_2	unit bisector for LARS
$X(\omega_k)$	STFT of the noise signal
X	magnitude STFT of noise

X_f	magnitude STFT of a frame of noise
$x(m)$	m^{th} sample of the time domain noise signal
x_1, x_2	basis vectors of $\mathbb{L}(X)$ space
$Y(\omega_k)$	STFT of the noisy speech
Y	magnitude STFT of noisy speech
Y_f	magnitude STFT of a frame of noisy speech
Y_{mat}	data matrix including N samples of Y_f
$y(m)$	m^{th} sample of the time domain noisy speech
$\hat{Y_f}$	estimate of the STFT of a frame of noisy speech
$\bar{y_2}$	projection of vector Y_f into $\mathbb{L}(X)$ space spanned by vectors x_1 and x_2 in LARS
z	LARC estimate of the vector Y_f input
$\phi(f, q_j)$	maximum likelihood of observing speech vectors θ_1 to θ_f being in state q_j at instant f
$\Psi(f, q_j)$	partial path with maximum likelihood in state q_j at time instant f
θ_f^i	enhanced observation at the f^{th} frame using class-specific dictionary with i^{th} label for JED
Θ	$= \{\theta_f^i; 1 \leq f \leq F, 1 \leq i \leq N\}$
σ	error constraint for sparse coding problem
γ	step size for LARC
$\hat{\gamma_1}$, $\hat{\gamma_2}$	step factor for LARS
$\mu_{\mathcal{A}}$	residual coherence corresponding to active set in LARC
$\hat{\mu_0}$, $\hat{\mu_1}$	vectors in $\mathbb{L}(X)$ space for LARS
μ_{coh}	residual coherence threshold of LARC
μ_j, μ_k	j^{th} and k^{th} element of residual coherence in LARC
$\mu^{(Y_f)}$, $\mu^{(z)}$	constant and variable part of the residual coherence in LARC

Publications from the Thesis

- P. M. Nazreen, A. G. Ramakrishnan, P. K. Ghosh, A class-specific speech enhancement for phoneme recognition: A dictionary learning approach, Proc. Interspeech (2016) 3728–3732.

- PM Nazreen, AG Ramakrishnan, and Prasanta Kumar Ghosh. A joint enhancement-decoding formulation for noise robust phoneme recognition. In 14th IEEE India Council International Conference (INDICON), pages 1–6. IEEE, 2017. 53

- PM Nazreen and AG Ramakrishnan. DNN based speech enhancement for unseen noises using Monte Carlo dropout. In 12th International Conference on Signal Processing and Communication Systems (ICSPCS), pages 1–6. IEEE, 2018.

- Nazreen P.M., A. G. Ramakrishnan, Improving Generalization of Monte Carlo Dropout Based DNN Ensemble Model for Speech Enhancement and Results on Real world, Traffic Noise; 16th IEEE India Council International Conference, 13-15 December 2019

  Preprints

- Nazreen P.M., A.G. Ramakrishnan, Monte Carlo dropout for low SNR, non-stationary, unseen noise reduction from speech; arXiv:1808.09432 [eess.AS]

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the unique features that differentiates humans from other species in the animal kingdom is their ability to produce properly articulated sounds defined as speech. The ability of humans to communicate through speech in various languages is truly astonishing. It is a very efficient means as well. It not only conveys linguistic information but also other important aspects like the mood of the speaker.

Unless the speaker is in a perfectly quiet environment like an anechoic chamber [1], the speech recorded is affected by the background noise present. As the noise increases, the information extracted from the recorded speech becomes less accurate. Hence the quality and intelligibility of speech is an important factor. The performance of speech processing systems used for communication or storage degrades due to these background noises, which results in information loss and hence, the removal of noise from speech is of great relevance.

Speech enhancement processes the noisy speech signal in order to improve its quality or intelligibility. Enhancement algorithms try to estimate the clean speech from the noisy recordings. Over the years, a vast number of methods have been developed to enhance speech [2]. But the complexity of the speech signals makes it rather difficult. Thus restoring the required speech from the one corrupted with background noise is still a challenging problem in the field of speech processing. The affected noise can be additive or convolutive. Most existing enhancement algorithms assume the noise to be additive. The additive noise problem, despite appearing to be rather simple, considerably reduces the quality and intelligibility of speech and is a challenging one even today. For the present work, we assume an additive noise framework.

Figure 1.1 shows a basic block diagram of speech enhancement framework for additive noise. The noise affecting the speech could be from several sources like traffic, vehicle engine, train, nearby speakers, party and factory. There are a variety of applications for speech enhancement algorithms. One such application is in the field of communication [3–5]. The invention of mobile phones for personal communication was a giant leap in the field. At the same time, reliable communication is a real challenge in today's world due to the surrounding noise. Speech enhancement is applied here to reduce this interfering noise for an efficient communication. In intercom systems such as the one used by pilots, aircraft crew, rescue personnel etc, where the noise level is quite high, efficient noise free communication is of utmost importance. A similar application is in communication over Internet such as via Skype, Google talk etc. Another

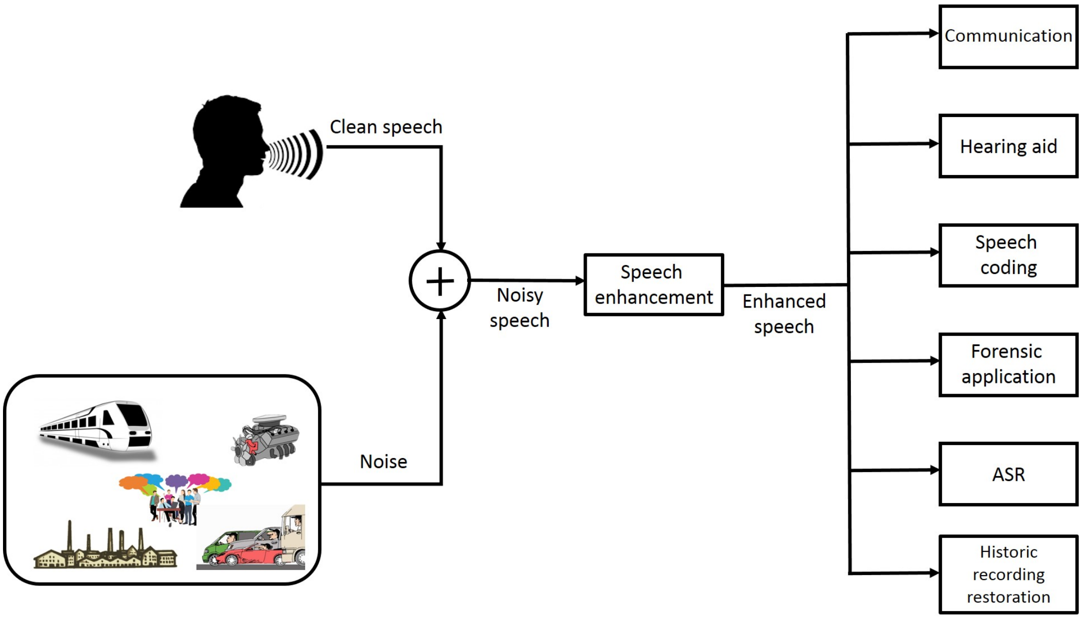


Figure 1.1: Enhancement framework for speech with additive noise

important application of enhancement algorithms is in hearing aids. A person with good hearing ability is able to understand the speech even in a noisy environment due to the redundancy of speech [6]. But for a person with hearing loss, most part of the speech is inaudible or distorted. Thus even a small background noise can heavily affect the intelligibility for such a person. A person with hearing loss requires a higher SNR range for better hearing compared to others. A hearing aid helps to achieve this by amplifying the speech. This might also result in the amplification of the background noise [7]. Hence speech enhancement algorithms are an integral part of hearing aids [6–8].

Speech recognition is a process of converting speech signal into a sequence of meaningful symbols such as words or phonemes [9]. The performance of an Automatic Speech Recognition (ASR) system degrades significantly in the presence of noise due to the mismatch between training and testing environments. Several techniques have been proposed to address this problem of which one of the most popular method is to employ a speech enhancement algorithm as the front-end processing [10, 11]. The enhancement algorithms employed for an ASR system has to ensure that the noise is reduced while ensuring that it is still suitable for machine recognition.

Speech enhancement algorithms also find other applications such as speech coding, forensic applications, restoration of historic recordings etc.

1.1 Enhancement of single channel speech with additive noise

Speech enhancement algorithms can be broadly classified into single channel and multi channel based on the number of input channels. For single channel enhancement only one input channel is available whereas multi channel case take the advantage of the availability of multiple signal inputs using microphone arrays. For the present work we have considered a single channel case with additive noise.

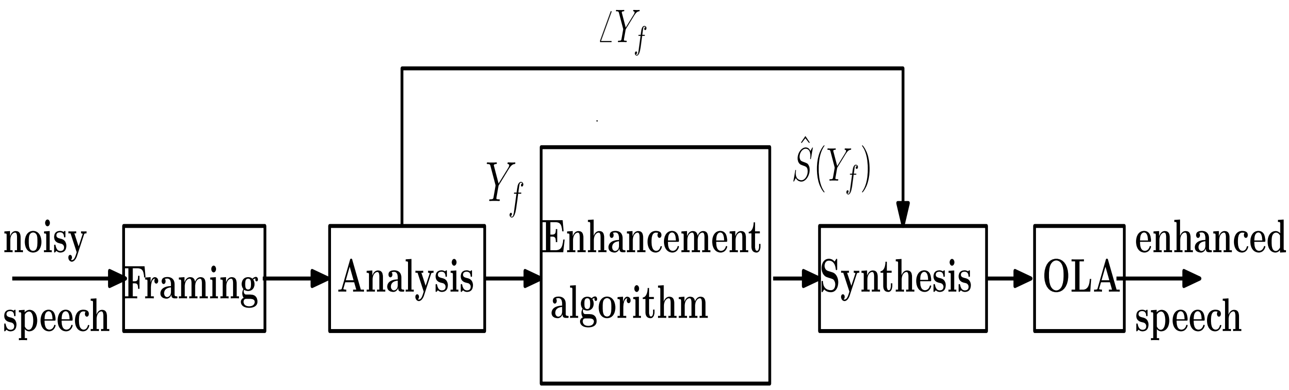


Figure 1.2: Single channel speech enhancement framework. The input noisy speech $y(m)$ is divided into frames. This is followed by an analysis stage to obtain the transform Y_f for a particular frame. The enhancement algorithm is applied on this transform which gives $\hat{S}(Y_f)$. A synthesis stage transforms the enhanced frames $\hat{S}(Y_f)$ back to time domain, usually using the noisy phase $\angle Y_f$, which is followed by an overlap add (OLA) stage to get the enhanced speech samples.

Figure 1.2 shows the general block diagram of a single channel speech enhancement framework under additive noise. Let $s(m)$ be the m^{th} sample of clean speech and $x(m)$ be the additive noise. The noisy speech is given as;

$$y(m) = s(m) + x(m) \tag{1.1}$$

The aim of single channel speech enhancement is to obtain an estimate of the clean speech $\hat{s}(m)$. The noisy speech is first divided into overlapping frames and then a transform such as discrete Fourier transform (DFT) is applied at the analysis stage to obtain $Y(\omega_k)$, where $\omega_k = \frac{2\pi k}{R}$, $k = 0, 1, 2...R-1$, R is the number of frequency bins and k is the index. It is a common practice to apply the enhancement algorithm on the magnitude term Y_k and combine it with the noisy phase $\angle Y(\omega_k)$. A synthesis stage is used to apply the inverse transform on this enhanced transform coefficients to convert it back to time domain. Finally overlap add method is applied to obtain the enhanced speech $\hat{s}(m)$.

1.2 Classification of speech enhancement algorithms

Speech enhancement algorithms can be generally classified as 1) spectral subtraction methods 2) Wiener Filtering 3) statistical model based algorithms 4) subspace methods 5) supervised learning approaches [2].

1.2.1 Spectral subtraction methods

Early work on enhancement using spectral subtraction was proposed in [12, 13]. Spectral subtraction methods assume that the noise affecting the speech is additive. The basic idea is to obtain an estimate of the clean speech spectrum by subtracting an estimate of the noise spectrum from the spectrum of noisy speech. The operation is usually performed in the magnitude spectral domain. Multiplying the noisy magnitude spectrum Y_k by a gain function H_k we get;

$$\hat{S}_k = H_k \times Y_k \tag{1.2}$$

where $\hat{S}_k$ is the enhanced speech magnitude spectrum at the k^{th} frequency index.

H_k is computed from the magnitude spectrum of the noisy speech Y and an estimate of the noise X_k as;

$$H_k = 1 - \frac{X_k}{Y_k} \tag{1.3}$$

The clean speech estimate is obtained by taking the inverse transform of $\hat{S}_k$ with the phase of the noisy speech $\angle Y(\omega_k)$. Thus this technique largely depends on the noise estimate, which is obtained from the non-speech sections and hence is highly dependent on the voice activity detector (VAD) [14] used. This in effect causes errors in the estimation of magnitude spectrum $\hat{S}_k$. Several techniques have been proposed to address this problem [2, 13, 15]. The noise estimation errors also produce a new randomly fluctuating type of noise referred to as musical noise. Many present day research in this field tries to address this problem as well [16–19].

1.2.2 Wiener filtering

Wiener filtering technique enhances the speech signal by minimising the mean squared error (MSE) [20]. In this case also, the enhanced magnitude spectrum $\hat{S}_k$ is obtained as;

$$\hat{S}_k = H_k \times Y_k \tag{1.4}$$

To obtain the gain function H_k, the mean square error (MSE) between the clean speech S_k and estimated speech signal $\hat{S}_k$ is minimized. Assuming additive noise, the gain function is computed as;

$$H_k = \frac{S_k^2}{Y_k^2} = 1 - \frac{X_k^2}{Y_k^2} \tag{1.5}$$

The clean speech estimate is obtained by taking the inverse transform of $\hat{S}_k$ using the phase of the noisy speech $\angle Y(\omega_k)$. Alternative methods of computing the gain function include an a-priori SNR based approach [21]. An iterative Weiner filter approach was proposed by Lim and Oppenheim [22]. Sreenivas and Kirnapure [23] developed a codebook constrained, iterative Wiener filter for enhancement. Several techniques have been proposed for the reduction of distortion introduced by this technique [24, 25]. An enhancement technique was used in [26] to deal with the musical noise introduced during Wiener filtering. Chen and Loizou [27] developed a new frequency-specific composite gain function for Wiener filtering.

1.2.3 Statistical model based algorithms

Statistical model based methods are defined as a statistical estimation problem with a well defined optimality criterion and statistical assumptions. These methods derive the noise suppression filter response based on a statistical model of the desired signal and noise. The parameter estimation based on maximum likelihood principle was applied to speech enhancement by McAulay and Malpass [28]. They performed a spectral decomposition of a frame of noisy speech and attenuated specific spectral lines depending on the proportion in which the measured speech plus noise power exceeded an estimate of the background noise power. Minimum mean square error (MMSE) method of speech enhancement uses mean square error for estimation. This method uses non-linear Bayesian estimation techniques. MMSE requires prior knowledge of the probability density functions (pdfs) of the speech and noise. Ephraim and

Malah proposed an MMSE estimator of the short time spectral amplitude (STSA) [29]. The MMSE-STSA method models speech and noise spectral components as statistically independent Gaussian random variables. Ephraim and Malah also proposed an estimator that minimizes the mean square error of the log spectra for enhacing noisy speech [30]. A power spectral density MMSE (PSD-MMSE) estimation was used in [31] for enhancement. Srinivasan *et.al.* proposed a codebook based Bayesian MMSE approach for speech enhancement in non stationary noise [32]. An analysis of the musical noise generated by MMSE-STSA estimator was done in [33]. Enhancement techniques have also been developed assuming a non-Gaussian pdf for MMSE estimation [34, 35]. Maximum a posteriori (MAP) methods estimate the parameters by maximizing the posterior pdf. Lotter and Vary [36] proposed a maximum a posteriori (MAP) estimator on spectral amplitude assuming a super-Gaussian prior. Loizou [37] proposed many Bayesian estimators in the magnitude spectrum using perceptually relevant distortion metrics as cost functions. Several techniques have also been developed [38, 39] to improve the estimation of a priori SNR. I. Cohen [38] proposed a non-causal estimator for the a priori SNR, and a corresponding non-causal speech enhancement algorithm. A two-step noise reduction (TSNR) approach was proposed in [39] to refine the estimation of the a priori SNR.

1.2.4 Subspace methods

The subspace methods for speech enhancement are derived using the principles of linear algebra. These methods assume that speech occupies a small subspace of the entire noisy speech space whereas noise occupies the entire space. In other words, the covariance matrix of the clean speech is rank deficient, while that of noise is full rank. Thus in this technique, the noisy speech signal is decomposed into two subspaces; the signal plus noise subspace and the noise subspace. The noise subspace is then removed from the signal plus noise subspace to estimate the clean speech signal. One approach in this method is to use the singular value decomposition (SVD) of time domain signals ordered in either Toeplitz or Hankel matrices [2]. Dendrinos *et. al.* [40] proposed an SVD based enhancement method by neglecting the eigenvectors corresponding to the smallest singular values, where the most of the noise information is contained and retaining the ones corresponding to the largest singular values where signal information is contained. There are several modifications of the SVD based method for speech enhancement [41–43]. Another similar approach is to use the eigenvalue decomposition (EVD) on the covariance matrix of the signal. Ephraim and Van Trees [44] proposed a Karhunen Loeve transform (KLT) based decomposition approach for speech enhancement. The main limitation of these methods is the assumption that the noise affecting speech is white. Several extensions have also been suggested [45–49] to enhance speech signal corrupted with colored noise .

1.2.5 Supervised learning approaches

Supervised learning approaches are different from the methods discussed above. The unsupervised methods discussed above are based on certain assumptions on the speech and noise signal and in general require the estimation of certain unknown quantities such as a priori SNR. Supervised approaches for enhancement, on the other hand, make use of representative data to learn the task. Certain models are considered for speech and noise signals and the parameters are learned from the set of training data available.

Ephraim developed a Bayesian estimation approach for enhancing speech signals where MMSE and MAP estimators were learned using hidden Markov models (HMM) [50]. Another HMM based enhancement method using MMSE principle for nonstationary noise was proposed in [51]. Kundu *et. al.* [52] proposed a GMM based approach for modeling the pdf of clean speech. Several codebook-driven approaches have also been developed [32, 53]. Dictionary learning approach for speech enhancement is also widely popular due to the effectiveness of the method. Non-negative matrix factorization (NMF) based methods [54–58] have been proposed where speech and noise dictionaries are learned from their respective training data using NMF. Thus the noisy speech spectrogram can be represented as a linear combination of basis vectors from clean speech and noise. Sparse coding based dictionary learning [59] is also widely used for speech enhancement, in which the general assumption is that the structured signals like speech can be sparsely represented as a linear combination of dictionary atoms. In some of the early works using artificial neural networks (ANN) [60–62], the authors have developed shallow networks to directly learn the mapping between a noisy speech frame and the clean speech frame. Recently, with the development of highly efficient computational resources and availability of huge amount of learning data, deep neural networks (DNN) have been widely used to learn the complex non-linear mapping between noisy and clean speech [63–68].

1.3 Performance measures

Evaluation of a speech enhancement algorithm is needed to analyze the effectiveness of the algorithm by measuring the degradation of the estimated speech from the clean speech. This can be achieved by subjective (human listeners) or objective methods. Some of the objective evaluation scores used for the present work are given below [69],

1.3.1 Perceptual evaluation of speech quality (PESQ)

The perceptual evaluation of speech quality is one of the most widely used measure [70, 71]. It was mainly designed for voice quality testing under real network conditions such as voice over IP (VoIP) [2]. PESQ approximates the mean opinion score (MOS), which is a listening test procedure used for evaluating speech. MOS ranges from 1 to 5, where 5 represents the best quality. The PESQ algorithm is quite complex, since it tries to approximate MOS. The first stage is a preprocessing stage in which the clean and the enhanced speech are equalized to a standard listening level and time aligned. The signals are then filtered with a filter having an impulse response similar to that of a standard telephone handset. The signals are then divided into frames and Bark scale filter banks are applied on the power spectra to obtain the loudness spectra after frequency and gain equalization stages. The difference between the clean and estimated signal is then computed considering positive and negative differences differently. Positive difference indicates noise addition and negative difference indicates attenuation. The negative difference is not as easily perceived due to masking. Hence different weights are applied to the positive and negative differences, also termed as disturbance values. These disturbance values are then averaged over time and frequency. The final PESQ score is computed as a linear combination of the average disturbance values and ranges between 1 and 4.5 to approximate MOS.

1.3.2 Segmental signal to noise ratio (SSNR)

Segmental SNR is the average of frame-wise SNRs. To evaluate SSNR, the speech signal is first divided into F frames and the SNR of each frame is computed. The mean of SNRs of all the frames gives the final SSNR value.

$$SSNR = \frac{10}{F}\sum_{f=1}^{F} log_{10}\frac{\sum_{m=1}^{M} s(m,f)^2}{\sum_{m=1}^{M}(s(m,f)-\hat{s}(m,f))^2} \tag{1.6}$$

where $s(m,f)$ and $\hat{s}(m,f)$ indicate the m^{th} sample of clean and the estimated speech in frame f and M indicates the total number of samples in any frame. This method provides an accurate measure of SNR for speech enhancement methods, provided the original and estimated signals are aligned in time.

1.3.3 Squared error (SE) metric

The squared error metric is frequently used in signal processing and is defined as

$$SE = \sum_{k=1}^{R}(S_k - \hat{S}_k)^2 \tag{1.7}$$

where S_k and $\hat{S}_k$ indicate the spectra of clean and estimated speech , k indicates the frequency index and R is the total number of frequency bins.

1.3.4 Itakura-Saito (IS) distance measure

The Itakura-Saito distance proposed by Fumitada Itakura and Shuzo Saito in the 1960s [2] is a measure of the perceptual difference between an original power spectrum, $P(\omega)$ and an estimation, $\hat{P}(\omega)$ of that spectrum.

$$IS(P(\omega), \hat{P}(\omega)) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\frac{P(\omega)}{\hat{P}(\omega)} - log\frac{P(\omega)}{\hat{P}(\omega)} - 1\right) \tag{1.8}$$

In short time power spectral domain, IS distortion at the k^{th} frequency bin is defined as;

$$IS(S_k^2, \hat{S}_k^2) = \frac{S_k^2}{\hat{S}_k^2} - log\frac{S_k^2}{\hat{S}_k^2} - 1 \tag{1.9}$$

where S_k is the clean speech spectra and $\hat{S}_k$ is the estimated spectra at the k^{th} frequency bin. Itakura-Saito distance is not a symmetric measure and it gives more emphasis to spectral peaks than spectral valleys.

1.4 Contributions of the thesis

We analyze various speech-sound class-specific and noise-specific model-based enhancement algorithms and frame-wise selection methods of these models. Experiments are performed with speech data from TIMIT corpus and noise samples from NOISEX-92 database. In our final experiment, we have also recorded traffic noise from 'CV Raman road' and conducted limited experiments on speech corrupted by this noise. The proposed algorithms does not specifically distinguish between indoor or outdoor environments. The noises used includes both indoor noises such as factory, babble and volvo as well as outdoor noises such as m109, leopard (tank noises) and traffic noise. We also evaluate the performance on white and pink noises as well.

1.4.1 A class-specific speech enhancement and its application for phoneme recognition: a dictionary learning approach

In Chapter 2, we explore class-specific enhancement approach. We use a sparse coding dictionary based approach to learn dictionaries of various speech classes, namely manner of articulation, place of articulation and phonemes. We found that using class-specific dictionaries for enhancing each frame would result in better enhancement than using class-independent dictionaries for all the frames. Some of these classes might be correlated well with noise compared to the other and hence by removing the contribution of these classes from those frames, where it is not required, we could expect a better enhancement than using a single generic dictionary in all the frames. To select the appropriate class dictionary for a particular frame, we use approximate labels obtained from an ASR system, whose input is the speech enhanced using a class-independent dictionary.

We use dictionary learning method using approximate KSVD with LARC coding. In the analysis section, we have evaluated the SSNR and PESQ measures of the proposed approach using ground truth phoneme labels and found only marginal improvements in most of the cases over class-independent scheme. However, when we analyze the performance of our various class-specific approaches in terms of phoneme recognition, we obtain superior performances compared to class-independent case, even when we use estimated labels for enhancement.

1.4.1.1 Speech sound classes

A phoneme is defined as the unit of sound that differentiates one word from another in a particular language. The sounds produced in each language can be classified into groups based on the phonetic properties they share. In the present study, we use the classification of phonemes based on manner and place of articulation to learn our class-specific dictionaries. We also use phoneme-specific dictionaries. Manner of articulation classification is based on the ways in which different articulators such as lips, tongue and velum are positioned to produce different sounds [72]. Place of articulation class on the other hand is based on the point at which airstream can be modified to produce a different sound [73, 74].

1.4.1.2 Sparse coding and dictionary learning

A dictionary is a matrix $D \in \mathbb{R}^{R \times L}$, which is composed of a set of prototype vectors of the data matrix. Here, R is the dimension of the data vector and L is the number of columns or atoms in the dictionary. Usually the dictionaries are overcomplete, ie; $L > R$. The dictionary atoms are normalized to unit ℓ_2 norm. The idea of sparse coding is to sparsely represent any feature vector $Y_f \in \mathbb{R}^{R\times 1}$ as a linear combination of the dictionary atoms. Thus $Y_f = D \times c_o$ where $c_o \in \mathbb{R}^{L\times 1}$ is the sparse coefficient vector. To ensure that the coefficient vector c_o is sparse, usually a constraint is set on the ℓ_0 or ℓ_1 norm of c_o. Thus any sparse coding technique tries to find the solution of the optimization problem

$$\underset{c}{\operatorname{argmin}}\, dist(Y_f, D \times c_o) \tag{1.10}$$

subject to a sparsity constraint on ℓ_0 or ℓ_1 norm of c_o. Here $dist(Y_f, D \times c_o)$ is the distance measure between Y_f and $D \times c_o$. Some of the widely used sparse coding algorithms are matching pursuit [75], orthogonal matching pursuit (OMP) [76], focal underdetermined system solver (FOCUSS) [77], basis pursuit [78], Least angle regression (LARS) [79] and batch LARS with coherence criterion (LARC) [59].

The dictionary matrix D is learned form the data matrix such that the data vector Y_f can be represented as the linear combination of the dictionary atoms using the coefficient vector c_o. A probabilistic method for the construction of dictionaries was proposed in [80, 81]. Coates and Andrew Y. Ng [82] proposed a k-means based feature representation approach. Relationship between sparse coding and vector quantization is discussed in [83, 84]. A generalization of k-means based approach, KSVD was proposed in [85]. An approximate KSVD dictionary update step was proposed by Rubinstein *et.al.* in [86].

For the present work, we use the approximate KSVD based dictionary learning with LARC sparse coding [59, 85, 86], the details of which are given in Chapter 2.

1.4.2 A joint enhancement-decoding formulation for noise robust phoneme recognition

In Chapter 3, we propose an algorithm, which aims to overcome the selection of erroneous dictionaries due to the error in the estimated class labels in the class-specific approach for improving the recognition performance. The joint enhancement-decoding (JED) algorithm jointly optimizes these class labels and the final recognized phoneme labels. The current noisy speech frame is enhanced by multiple N phoneme-specific dictionaries close to the approximate label of that frame. These multiple enhanced frames are then fed into the JED algorithm. The algorithm accepts these N observations and chooses the best in each frame such that the overall likelihood is maximized to obtain the final recognized labels. The Viterbi decoding algorithm used in speech recognition is integrated with the class label selection to develop the

JED algorithm. Experiments are conducted by varying N from 1 to 5 to find the best value of N that gives the maximum recognition performance for various noises and SNRs.

1.4.3 Monte Carlo dropout for low SNR, non-stationary noise reduction from speech

In Chapter 4, we propose a method of picking the best DNN model in the scenario where multiple noise-specific DNN models are available for enhancement, using the Monte Carlo (MC) dropout proposed by Gal and Ghahramani [87]. MC dropout is a tool for modeling uncertainty in DNN, using dropout during inference stage. The conventional dropout of these multiple DNN models is replaced with MC dropout and a measure of the model uncertainty is used for the selection of DNN models. We find this method to be particularly useful for unseen noisy scenario, where the noise corrupting the test speech is different from that with which the available DNN models are trained. This method performs better than the method of using a DNN classifier for the selection, in the case of unseen noises. In order to compensate for the poor performance of the above algorithm in the case of seen noises compared to classifier-based selection scheme, we propose a threshold-based algorithm to switch between model uncertainty-based selection scheme and classifier-based model selection scheme. This algorithm is found to be useful for unseen noises at the same time giving comparable performance to that of classifier-based scheme for seen noises. Some promising results in enhancement performance of the algorithms on speech corrupted with a mixture of multiple noises and for a time varying scenario where different segments of speech are corrupted by different noises are given. The algorithms also give performances superior to classifier-based model selection scheme in a real world scenario where speech is corrupted with real world, traffic noise. We also explore the use of MC dropout in improving the generalizability of a single DNN model for enhancement, when the conventional dropout is replaced by MC dropout. We show that in the case of noisy speech corrupted with unseen noises, MC dropout models can give a better denoised output than conventional dropout models.

Chapter 2

A class-specific speech enhancement and its application for phoneme recognition: a dictionary learning approach

We study the advantage of class-specific dictionaries over class-independent dictionary for enhancement of noisy speech. We hypothesize that, using class-specific dictionaries would remove the noise more than a class-independent dictionary, thereby resulting in better phoneme recognition. Experiments are performed with speech data from TIMIT corpus and noise samples from NOISEX-92 database. Using KSVD, four types of dictionaries have been learned: class-independent, manner-of-articulation-class, place-of-articulation-class and 39 phoneme-class. Initially, a set of labels are obtained by recognizing the speech enhanced using a class-independent dictionary. Using these approximate labels, the corresponding class-specific dictionaries are used to enhance each frame of the original noisy speech, and this enhanced speech is then recognized. Compared to the results obtained using the class-independent dictionary, the 39 phoneme-class based dictionaries provide a relative phoneme recognition accuracy improvement of 5.5%, 3.7%, 2.4% and 2.2%, respectively for factory2, m109, leopard and babble noises, when averaged over 0, 5 and 10 dB SNRs.

2.1 Introduction

The aim of speech enhancement is to improve the quality of speech by attenuating the noise associated with it, without degrading the actual speech. The challenge in speech enhancement is mainly because of the non-stationary nature of the noise associated. For applications such as speech recognition, speaker recognition and hearing aids, speech enhancement is employed as a front end processing.

Recently, sparse coding techniques have gained popularity. A speech enhancement scheme based on sparse coding has been proposed by Sigg *et al.* [59], who show that it performs better than techniques like geometric spectral subtraction [19]. Several exemplar-based techniques [88, 89] have also been proposed in the past for robust speech recognition. In sparse coding, the basic assumption is that we can represent structured signals like speech as sparse linear combinations of prototype vectors or atoms.

The performance of various speech enhancement algorithms can be evaluated by certain objective measures [69]. In this chapter, we analyze the usefulness of our class-specific enhancement method for phoneme recognition. We observe that even though we did not find any significant improvement over class-independent method in terms of objective measures, our algorithms give superior performance for phoneme recognition [11]. Enhancing the speech signal as a front-end processing before it is fed into a recognizer is fairly popular because of the simplicity of the approach and also since it obviates the need to retrain the ASR systems for different types of noisy inputs and the same ASR trained on clean speech can be used.

2.1.1 Motivation

Speech signal is composed of several sounds which can be categorized in various ways, like manner-of-articulation (MOA) [72], place-of-articulation (POA) [73, 74] or phonemes (PHN). Some of these classes might correlate well with certain noise types more than the other classes. Hence the atoms in a dictionary learned using these classes may represent noise power to varying degrees and consequently result in poor speech reconstruction. By removing the contribution from atoms of the classes that correlate well with noise, one can improve the enhancement performance. One way to achieve this is to learn different dictionaries for different classes and intelligently select a particular dictionary for a segment. Raj *et al.* [90] propose a similar approach, where they use phoneme-dependent non-negative matrix factorization (NMF) for separation of music from speech. However, no quantitative analysis has been performed in that work. In this work, we extend their idea to sparse coding to analyze how, using class-specific dictionaries, the performance of an ASR system could be improved over that obtained using a dictionary learned in a class-independent manner. Wang et al [91] investigated the use of class-specific, ideal ratio mask estimation for speech enhancement using DNN. But the recognizer used as well as the mask estimator were trained using noisy speech. However, we consider a more realistic scenario where the noise level is not known a-priori and a recognizer trained on clean speech is used.

2.2 Enhancement using learned dictionary

We consider a single channel speech enhancement framework. This can be modeled as an additive mixture of clean speech and noise.

Under additive model, noisy speech can be represented as,

$$y(m) = s(m) + x(m) \tag{2.1}$$

where $y(m)$, $s(m)$ and $x(m)$ are the m^{th} samples of the time domain noisy speech, clean speech and noise signal, respectively.

Considering the short time Fourier transform (STFT),

$$Y(\omega_k) = S(\omega_k) + X(\omega_k) \tag{2.2}$$

where $\omega_k = \frac{2\pi k}{R}$, $k = 0, 1, 2...R - 1$, R is the number of frequency bins and k is the index.

Taking the magnitude STFT, the noisy speech can be approximated as

$$Y \approx S + X \in R^{R\times 1} \tag{2.3}$$

where S and X represent the spectra of the clean speech signal and the noise signal, respectively.

In order to recover clean speech from this noisy speech using dictionary learning approach, models of both clean speech and noise are learned, which result in a collection of prototype vectors called dictionaries.

Let $D_s \in R^{R\times L}$ and $D_x \in R^{R\times L}$, $L > R$, be the overcomplete dictionaries of L atoms each, learned using speech and noise data. Given an input noisy speech vector for a frame Y_f, the two dictionaries could be concatenated to form $D = [D_s\ D_x]$. The corresponding sparse coefficient vectors learned using this dictionary D can be represented as $c_{con} = [c_s\ c_x]$. Using D and c_{con} an estimate of the STFT of the noisy speech is given by

$$\hat{Y_f} = D \times c_{con} = D_s \times c_s + D_x \times c_x \tag{2.4}$$

The sparse coefficients c_{con} can be separated to c_s and c_x to estimate the enhanced speech $\hat{S}_f$;

$$\hat{S}_f = D_s \times c_s \ . \tag{2.5}$$

2.2.1 Sparse coding

The aim of sparse coding is to sparsely represent any given vector $Y_f \in \mathbb{R}^{R\times 1}$ as a linear combination of representative vectors obtained from dictionary $D \in \mathbb{R}^{R\times L}$ [59]. The dictionary considered is usually over-complete, i.e., $L > R$

For a given dictionary D and the spectrum Y_f of a given noisy speech frame, the sparse coefficients can be obtained by solving the formulation using a cardinality constraint;

$$c_o^* = \underset{c_o}{\operatorname{argmin}} \|Y_f - Dc_o\|_2; \quad \text{s.t.} \ \|c_o\|_0 \ \leq t; \ t \ll R \tag{2.6}$$

Using an error constraint, the problem can be formulated as;

$$c_o^* = \underset{c_o}{\operatorname{argmin}} \|c_o\|_0; \quad \text{s.t.} \ \|Y_f - Dc_o\|_2 \leq \sigma; \tag{2.7}$$

The above problem can be solved by various schemes like orthogonal matching pursuit [76], which is a greedy iterative approach which computes an approximate solution to the sparse coding problem (2.6) or (2.7).

Applying a convex relaxation of ℓ_0 norm to ℓ_1 norm, the problem (2.6) becomes

$$c_o^* = \underset{c_o}{\operatorname{argmin}} \|Y_f - Dc_o\|_2; \ \text{s.t.} \ \|c_o\|_1 \ \leq t_1; \ t_1 \ll R \tag{2.8}$$

This formulation is known as least absolute shrinkage and selection operator (LASSO) [92]. Since the objective function and the constraint in eq. (2.8) are convex the solution is unique and can be found using quadratic optimization techniques. Least angle regression (LARS) [79] is a very efficient iterative algorithm, which gives a solution very close to LASSO.

2.2.1.1 Least angle regression (LARS)

LARS involves an atom selection stage based on maximum correlation with the residue. The algorithm proceeds in an equiangular direction to the current set of atoms until a new atom, which is equally correlated with the current residue as the atoms in the active set, finds its way to the active set. LARS then proceeds in a direction equiangular to the new set of atoms and this process repeats [59, 79]. A cardinality or error based stopping criterion can be used.

Let $\bar{y_2}$ be the projection of Y_f into $\mathbb{L}(X)$ which is spanned by the bases x_1 and x_2. The algorithm begins at $\hat{\mu_0} = 0$ and proceeds in the direction of x_1, as $\bar{y_2} - \hat{\mu_0}$ has greater correlation with x_1 than x_2. LARS then estimates $\hat{\mu_1} = \hat{\mu_0} + \hat{\gamma_1} x_1$; where $\hat{\gamma_1}$ is chosen such that $\bar{y_2} - \hat{\mu_1}$ bisects the angle between x_1 and x_2. Finally the algorithm estimates $\hat{\mu_2} = \hat{\mu_1} + \hat{\gamma_2} u_2$; where u_2 is the unit bisector and $\hat{\mu_2} = \bar{y_2}$ for the case where the number of bases vectors is two.

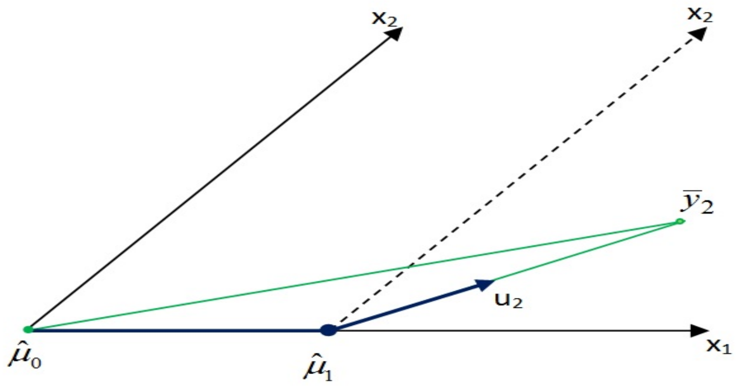


Figure 2.1: The LARS algorithm; x_1 and x_2 are the bases vectors: Initialize $\hat{\mu_0} = 0$ and proceed in the direction of x_1. $\hat{\mu_1} = \hat{\mu_0} + \hat{\gamma_1} x_1$; $\hat{\gamma_1}$ is chosen such that $\bar{y_2} - \hat{\mu_1}$ bisects the angle between x_1 and x_2. Estimate $\hat{\mu_2} = \hat{\mu_1} + \hat{\gamma_2} u_2$; where u_2 is the unit bisector and $\hat{\mu_2} = \bar{y_2}$.

2.2.1.2 Batch LARS with coherence criterion (LARC)

Sigg *et.al.* [59] proposed a two way modification to LARS called batch LARS with coherence criterion (LARC). For the present work we use the batch LARC algorithm for sparse coding. In LARC , the Gram matrix $G = D^T D$ is precomputed. The matrix inverse $G^{-1}_{\mathcal{A},\mathcal{A}}$ corresponding to the active set of atoms $\mathcal{A}$ is computed iteratively using an update scheme based on Cholesky factorization [86].

The algorithm also proposes a new stopping criterion based on a threshold μ_{coh} on the residual coherence. the steps involved in LARC are given in Algorithm 1.

2.2.2 Dictionary learning

A dictionary D is a matrix with each column consisting of atoms whose linear combination can be used to represent a given signal. Usually the dictionary is chosen to be over-complete. An ideal dictionary is considered as the one which can sparsely represent the given signal with minimum representation error. M. Aharon *et.al.* introduced a novel algorithm [85] for adapting dictionaries, which is a generalization of K-means algorithm.

Algorithm 1: Batch LARC

$\mathcal{A}$: The active set of atoms and any variable with subscript $\mathcal{A}$ indicates the one corresponding to this active set.

μ_{coh}: residual coherence threshold; G: Gram matrix; c_o: sparse coefficient solution; z: LARC approximation of the input vector; $\mu^{(Y_f)}$, $\mu^{(z)}$: constant and variable part of the residual coherence

Input : $Y_f \in \mathbb{R}^{R\times 1}$; $D \in \mathbb{R}^{R\times L}$; μ_{coh}; $G = D^T D$

Output: $c_o \in \mathbb{R}^{L\times 1}$

$c_o \leftarrow 0$; $\mathcal{A} \leftarrow \{\}$; $z \leftarrow 0$

$\mu^{(Y_f)} \leftarrow D^T Y_f$; $\mu^{(z)} \leftarrow 0$

while $|\mathcal{A}| < R$ **do**

- $\mu \leftarrow \mu^{(Y_f)} - \mu^{(z)}$
- $j^* \leftarrow \text{argmax}_j |\mu_j|, j \in \mathcal{A}^c$
- $\mathcal{A} \leftarrow \mathcal{A} \cup \{j^*\}$
- **if** $|\mu_{j^*}|/\|Y_f - z\|_2 < \mu_{coh}$ **then**
  - break
- $sgn \leftarrow \text{sign}(\mu_{\mathcal{A}})$
- $g \leftarrow G^{-1}_{(\mathcal{A},\mathcal{A})} sgn$
- $b \leftarrow (g^T sgn)^{-1/2}$
- $w \leftarrow bg$
- $u \leftarrow D_{(:,\mathcal{A})} w$
- $a \leftarrow G_{(:,\mathcal{A})} w$
- $\gamma \leftarrow \min^+_{q\in\mathcal{A}^c}[(|\mu_{j^*}| - \mu_q)/(b - a_q), (|\mu_{j^*}| + \mu_q)/(b + a_q)]$
- $z \leftarrow z + \gamma u$
- $c_{o\mathcal{A}} \leftarrow c_{o\mathcal{A}} + \gamma w$
- $\mu^{(z)} \leftarrow \mu^{(z)} + \gamma a$

2.2.2.1 KSVD based dictionary learning

KSVD is an iterative algorithm which tries to sparsely represent a given data matrix Y_{mat} $\in\mathbb{R}^{R\times N}$ in terms of the dictionary D $\in\mathbb{R}^{R\times L}$ and the sparse coding matrix C $\in\mathbb{R}^{L\times N}$. It involves both sparse coding and dictionary update stages. The algorithm tries to solve the problem,

$$\min_{D,C} \|Y_{mat} - DC\|_F^2 \tag{2.9}$$

subject to a sparsity constraint on C and $\forall l$; $\|D(:,l)||_2 \; = 1$

The algorithm involves the following steps;

Step1 : Dictionary initialization

Usually the dictionary $D^{(0)}$ is initialized either by choosing at random on a unit hypershere or by randomly sampling from the training data Y_{mat} itself. The atoms are then rescaled to unit length.

Step2 : Sparse coding

At each iteration I, the sparse coefficients are obtained by the LARC algorithm as;

$$c^I(:,l) = LARC(Y_{mat}(:,l), D^{I-1}, \mu_{coh}) \tag{2.10}$$

Step3 : Dictionary update

For each column in the dictionary $l = 1, 2, ...L$, the penalty term in (2.9) can be rewritten as,

$$\begin{aligned} \|Y_{mat} - DC\|_F^2 &= \|Y_{mat} - \sum_{j=1}^{L} D(:,j)C(j,:)\|_F^2 \\ &= \|\left(Y_{mat} - \sum_{j\neq l} D(:,j)C(j,:)\right) - D(:,l)C(l,:)\|_F^2 \\ &= \|E_l - D(:,l)C(l,:)\|_F^2 \end{aligned} \tag{2.11}$$

Thus DC is decomposed into the sum of L rank 1 matrices. For updating the l^{th} column, the rest of $L-1$ terms are assumed to be fixed. Thus E_l stands for the error for the N examples when l^{th} atom is removed.

In order to ensure that the updated $D(:,l)$ enforces the sparsity constraint, the error matrix E_l is restricted as E_l^R representing the error involving samples of Y_{mat} which only use atom

$D(:,l)$

Applying SVD decomposition to this matrix E_l^R we get;

$$E_l^R = U\Sigma V^T \tag{2.12}$$

The solution of $D(:,l)$ is defined as the first column of U while the first column of V multiplied by $\Sigma(1,1)$ gives the coefficient vector $C(l,:)$.

Steps 2 and 3 is repeated until convergence.

For the present work, we use the approximate KSVD dictionary update step proposed by Rubinstein *et.al.* [86] with reduced complexity (algorithm 2).

Algorithm 2: Approximate KSVD dictionary update

Input : $Y_{mat} \in \mathbb{R}^{R\times N}$; $D \in \mathbb{R}^{R\times L}$; $C \in \mathbb{R}^{L\times N}$

Output: Updated D

for $l \leftarrow L$ **do**

- $D(:,l) \leftarrow 0$
- $\mathcal{N} \leftarrow \{n | C(l,n) \neq 0, 1 \leq n \leq \mathcal{N}\}$
- $E_l^R \leftarrow Y_{mat}(:,\mathcal{N}) - DC(:,\mathcal{N})$
- $g_1 \leftarrow C^T(l,\mathcal{N})$
- $h \leftarrow E_l^R g_1$
- $h \leftarrow h/\|h\|_2$
- $g_1 \leftarrow (E_l^R)^T h$
- $D(:,l) \leftarrow h$
- $C(l,\mathcal{N}) \leftarrow g_1^T$

2.3 Speech enhancement with class-specific dictionaries

In this work we try to explore the enhancement performance of using class-specific dictionaries rather than a class-independent one. We explore the objective measures as well as the ASR performance of our algorithm. For our experiments, three different categories of dictionaries are considered. Let there be c dictionaries in each category. In the first category, separate dictionaries are learned based on manner-of-articulation (MOA) of speech where $c = 5$, denoted by $D_1^{MOA}, ...D_5^{MOA}$. In the second category, dictionaries are learned based on place-of-articulation (POA) of speech where $c = 14$, denoted by $D_1^{POA}, ...D_{14}^{POA}$. In the third case, separate dictionaries are learned for 39 different phonemes (PHN) [93, 94] with $c = 39$, denoted by $D_1^{PHN}, ...D_{39}^{PHN}$.

2.3.1 Experimental setup

All the experiments are conducted on the TIMIT [95] speech corpus consisting of 6300 sentences from 630 speakers with train and test sets containing 4620 and 1680 utterances, respectively. The sampling frequency is 16 kHz. The *sa* utterances are not used, since they are common to both training and testing sets. The ASR is trained on the entire clean TIMIT training data and its performance is obtained on the entire TIMIT test set after adding various noises. Thus training is done on clean speech and testing on noisy data. We use factory2, m109, leopard, babble and volvo noises from the NOISEX-92 [96] database after downsampling to 16 kHz, to synthesize noisy test speech signals. For the recognition experiments, HTK [97] is used. The size of analysis frame is chosen to be 30 ms with 10 ms frame shift. 39-dimensional mel frequency cepstral coefficients (MFCC) [98] are used, together with zeroth coefficient, delta and delta-delta coefficients. Cepstral mean normalization (CMN) is applied. A three-state monophone HMM model with diagonal covariance matrix is used for the recognizer. The number of Gaussian mixtures per state is set to 32, since increasing it further does not improve the recognition performance significantly. A bigram phoneme language model is used. For phoneme recognizer, the 61 phonemes in TIMIT are mapped to a reduced set of 39 phonemes [93, 94] and the results are reported on this reduced set.

The dictionaries are learned on the magnitude STFT computed using a frame size of 30 ms with 10 ms frame shift. A 512-point FFT is taken and we use only the first 257 points for learning the dictionary because of symmetry in the spectrum. We use approximate KSVD algorithm with LARC coding [59] for learning the dictionaries. The number of iterations for KSVD is set to 30. The dictionaries are speaker independent and each dictionary is trained to have 512 atoms. The class-independent dictionary is learned on a subset of 2×10^5 frames which are randomly sampled from the training data. For learning class-specific dictionaries, the training frames are classified into different classes, using the TIMIT labels. MOA, POA, as well as PHN specific dictionaries are learned from the spectra of corresponding training frames. For MOA class, vowels, diphthongs and semivowels are grouped together [72]. For POA class, consonants and vowels are classified as per [73] and [74], respectively. For PHN specific dictionary, we learn only 39 dictionaries based on the reduced phoneme set.

2.3.2 Preliminary experiments using ground truth phoneme labels

As a preliminary set of experiments, we try to analyze the enhancement performance when 39 phoneme (PHN) based dictionaries are used instead of a single class-independent dictionary. Figure 2.2 shows the basic block diagram of the approach. We use the ground truth labels obtained from TIMIT [95] for our experiment.

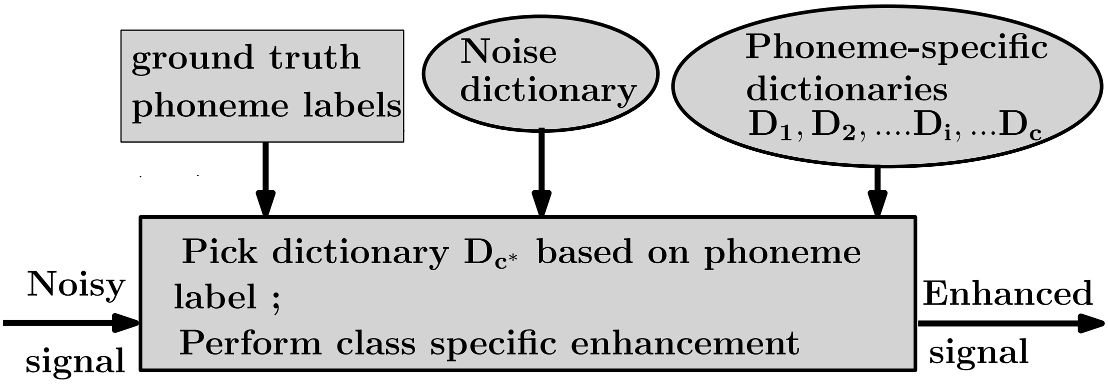


Figure 2.2: Speech enhancement with phoneme specific dictionaries using ground truth phoneme labels.

Let the magnitude spectrum of the fth frame of noisy speech input be $Y_f \in R^{R\times 1}$ and the ground truth phoneme label for that frame be c^* where $1 \leq c^* \leq 39$. Using the composite dictionary $D3 = [D_{c^*}^{PHN}\ D_x]$, the sparse coefficients and the clean speech are estimated as,

$$[c_s^{PHN}\ c_x^{PHN}] = LARC(Y_f, D3, \mu_{coh}) \tag{2.13}$$

$$\hat{S}_f^{PHN} = D_{c^*}^{PHN} \times c_s^{PHN} \tag{2.14}$$

where c_s^{PHN} is the sparse coefficient vector corresponding to $D_{c^*}^{PHN}$ and μ_{coh} is the threshold on residual coherence.

2.3.2.1 Results and discussion

Table 2.1 shows the performance of this approach in terms of PESQ and segmental SNR (SSNR) [69] for speech corrupted with noises factory2, m109, leopard, babble and volvo at 0, 5 and 10 dB SNRs. The results are averaged over 10 files randomly selected from TIMIT[95] test set. Only 10 files are considered as this is a preliminary experiment to evaluate the performance in terms of objective quality measures [37, 99]. The random selection ensures that there is no bias in the selected files.

It can be inferred from the Table that not much improvement is observed in terms of PESQ, for PHN-gnd compared to the method where a single class-independent (class-ind) dictionary is used for enhancement. In terms of SSNR, the method gives marginal improvement over class-ind scheme in most of the cases. For babble noise case, better improvement is observed for PHN-gnd over class-ind scheme in terms of SSNR.

An analysis of the phoneme recognition performance of the speech enhanced by PHN-gnd method for factory2, m109, leopard, babble and volvo noises for SNRs 0, 5 and 10 dB is shown in Figs. 2.3 (a-e). In this case, the accuracy is computed for the entire TIMIT [95] test set.

It can be inferred that even though not much improvement is observed in terms of the quality measures like PESQ and SSNR, in terms of phoneme recognition accuracy, PHN-gnd scheme gives really good improvement in performance over class-ind scheme. This is expected, as for many enhancement methods, improvements in terms of quality measures does not necessarily translate to improvements in terms of machine recognition and vice versa. This has been explored in [10], where they give a comparative evaluation of various enhancement approaches in terms of quality measures and ASR performance (in terms of phoneme correctness) and show that there might not be a one to one correspondence between the two.

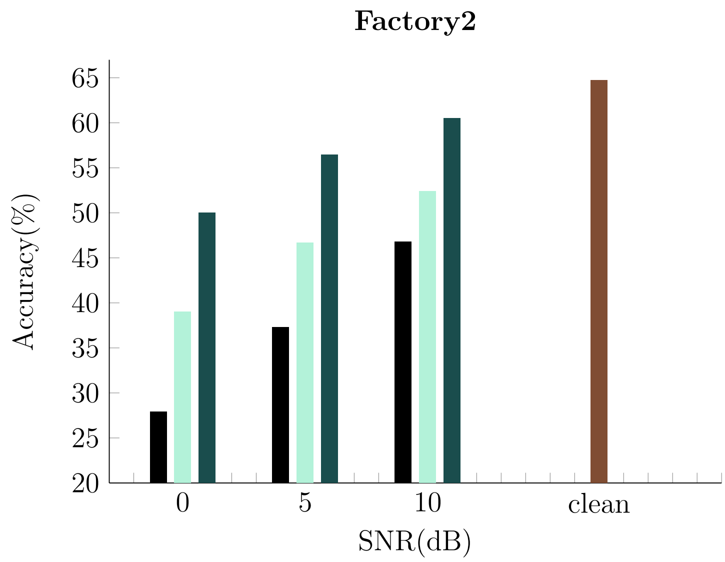
For factory2 noise case, the relative accuracy improvement (RAI) of PHN-gnd method over class-ind method for phoneme recognition are 28.0 %, 20.9 % and 15.4 %, respectively for 0, 5 and 10 dB SNRs. For m109 noise , the RAI values are 23.3 %, 17.0 % and 12.5 %. For leopard and babble noise cases, the RAI values are 15.2 %, 11.5 % , 9.5 % and 35.1 %, 29.8 % 22.6 %, respectively.

In the case of volvo noise, it is observed that after CMN, the recognition accuracy using noisy speech outperforms the class-independent case. Still PHN-gnd outperforms both noisy and clas-ind cases. For volvo noise, The RAI values of PHN-gnd over class-ind case are 10.8%, 9.3% and 7.8%. RAI values over noisy case are 7.8%, 5.2% and 4.6%.

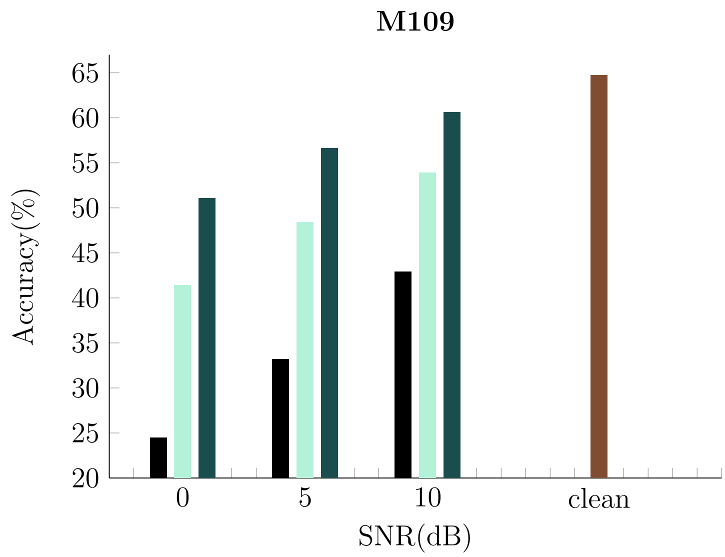
Figures 2.3 (a-e) show the potential application of class-specific enhancement scheme in phoneme recognition. From the preliminary experiments using ground truth labels obtained from TIMIT [95], we have inferred that, using class-specific dictionaries for enhancement rather than a class-independent dictionary improves the phoneme recognition performance though no improvement is observed in terms of measures like PESQ over class-independent enhancement scheme.

Table 2.1: Performance evaluation in terms of PESQ and Segmental SNR (SSNR) of enhancement using class-specific dictionaries using phoneme ground (PHN-gnd) truth labels for input SNRs of 0, 5 and 10 dB for five different noises. The results are averaged over 10 files randomly selected from TIMIT test set. Training set: entire TIMIT training set excluding *sa* utterances. Noise signals from NOISEX-92 database. class-ind: class-independent dictionary used.

	SNR (dB)	PESQ			SSNR		
		Noisy	class-ind	PHN-gnd	Noisy	class-ind	PHN-gnd
Factory2	0	1.2	1.7	1.7	-4.3	2.1	2.2
	5	1.5	2.1	2.1	-1.2	4.1	4.2
	10	1.8	2.5	2.6	2.1	6.1	6.2
Leopard	0	1.2	1.8	1.8	-4.4	3.9	4.0
	5	1.3	2.1	2.1	-1.4	5.5	5.6
	10	1.5	2.5	2.5	2.0	7.1	7.2
M109	0	1.1	1.7	1.7	-4.3	2.6	2.7
	5	1.4	2.1	2.1	-1.3	4.5	4.7
	10	1.7	2.6	2.6	2.1	6.4	6.6
Babble	0	1.1	1.3	1.3	-4.2	-0.8	-0.4
	5	1.3	1.5	1.6	-1.9	1.2	1.7
	10	1.6	1.9	2.0	2.2	3.3	3.8
Volvo	0	1.7	2.7	2.7	-3.7	7.5	7.8
	5	2.0	3.1	3.1	-0.6	9.0	9.2
	10	2.4	3.4	3.4	2.8	10.4	10.4



(a)



(b)

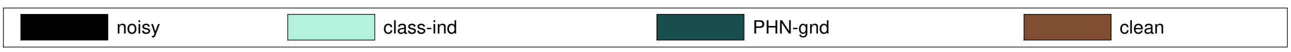
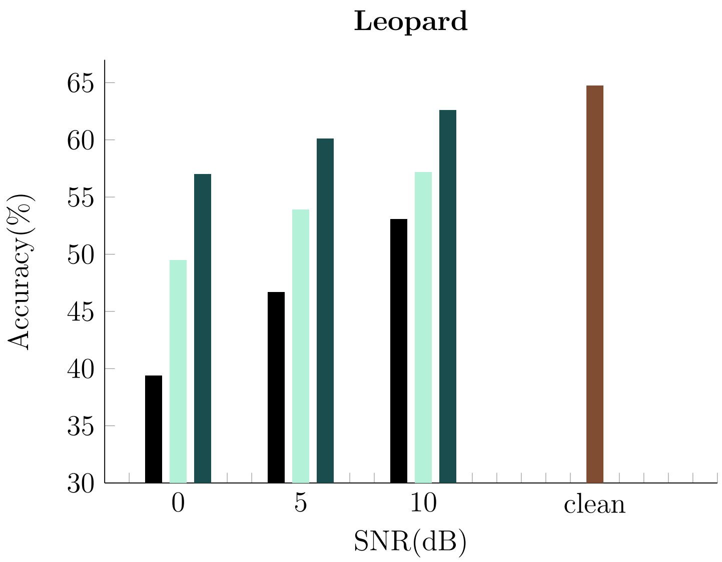
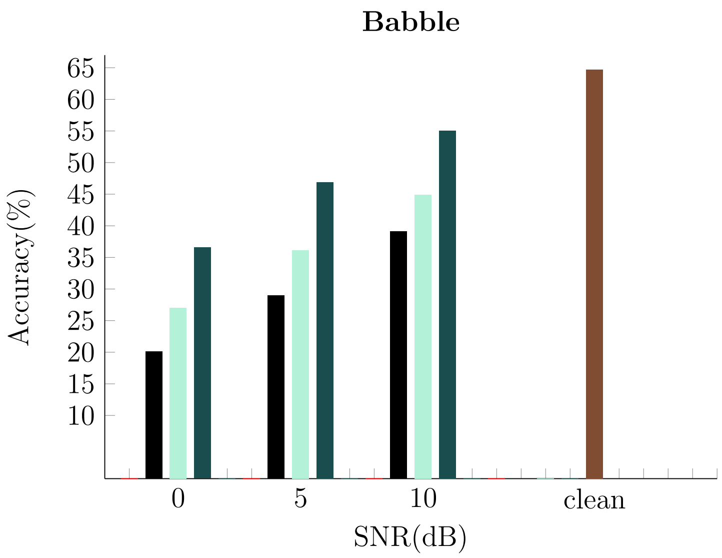


Figure 2.3: Comparison of phoneme recognition accuracies for (a) Factory2, (b) M109 noises for PHN-gnd enhancement over class-ind enhancement for 0, 5 and 10 dB input SNRs. Noisy indicates the recognition performance for the noisy input speech itself. The ASR is trained on the entire clean TIMIT training data and its performance is obtained on the entire TIMIT test set excluding *sa* utterances after adding various noises from NOISEX-92 database.



(c)



(d)

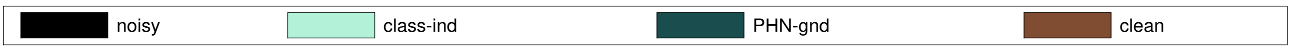
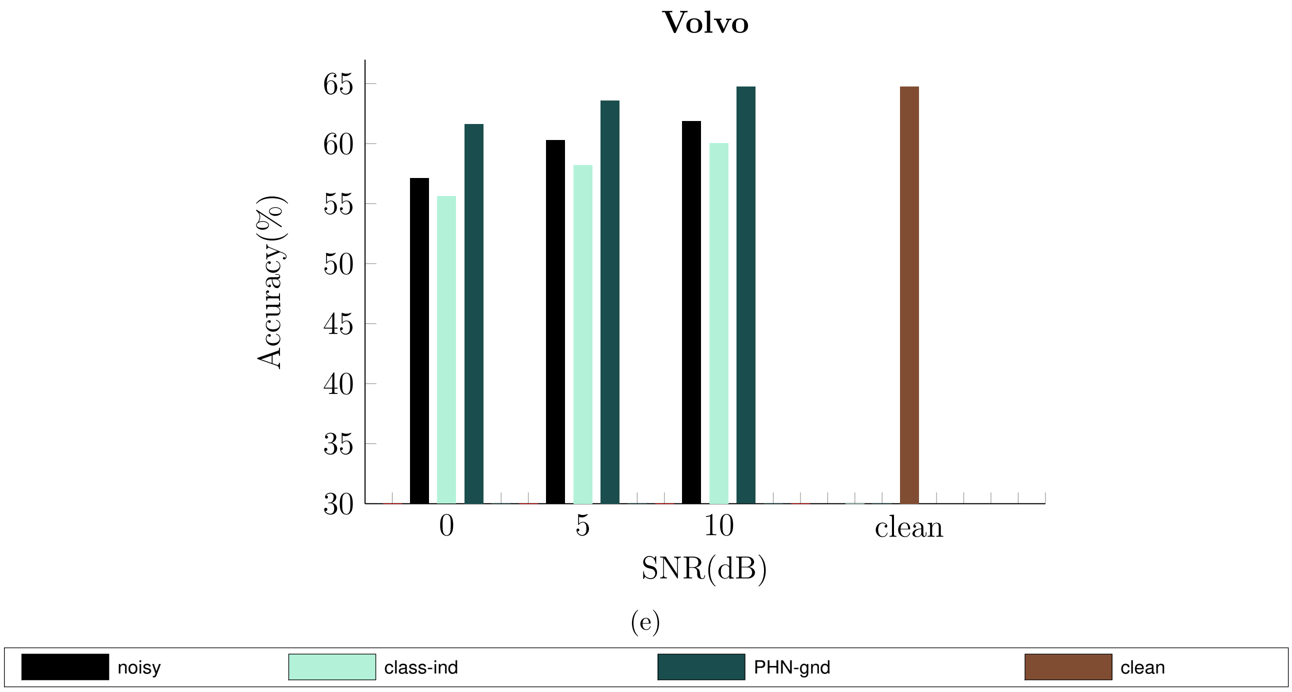


Figure 2.3: Comparison of phoneme recognition accuracies for (c) Leopard, (d) Babble noises for PHN-gnd enhancement over class-ind enhancement for 0, 5 and 10 dB input SNRs. Noisy indicates the recognition performance for the noisy input speech itself. The ASR is trained on the entire clean TIMIT training data and its performance is obtained on the entire TIMIT test set excluding *sa* utterances after adding various noises from NOISEX-92 database.



(e)

Figure 2.3: Comparison of phoneme recognition accuracies for (e) Volvo noise for PHN-gnd enhancement over class-ind enhancement for 0, 5 and 10 dB input SNRs. Noisy indicates the recognition performance for the noisy input speech itself. The ASR is trained on the entire clean TIMIT training data and its performance is obtained on the entire TIMIT test set excluding *sa* utterances after adding various noises from NOISEX-92 database.

2.3.3 Phoneme recognition performance on speech enhanced with class-specific dictionaries using estimated class labels

In this section, we estimate MOA, POA and PHN class labels from noisy speech and use it to perform class-specific enhancement and explore its usefulness in phoneme recognition. Figure 2.4 shows the block diagram summarizing the steps of class-specific dictionary based enhancement for phoneme recognition proposed in the present study. At first, the class label of each frame is obtained by recognizing the speech enhanced using the class-independent dictionary. Using this approximate label, the corresponding class-specific dictionary, which was learned from the training data, is used to enhance the input noisy speech in each frame, and this newly enhanced speech is recognized. The enhancement and recognition stages are explained in Algorithm 3.

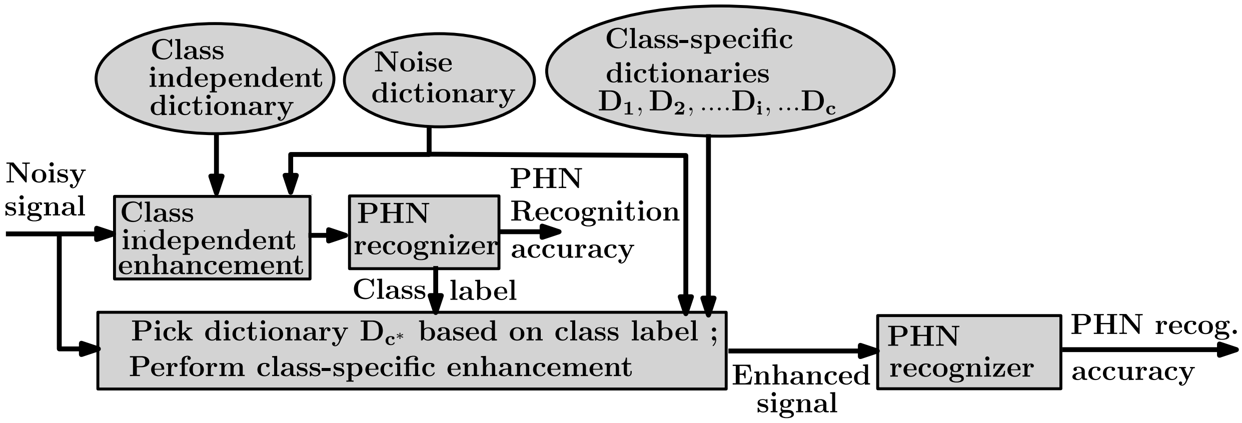


Figure 2.4: Phoneme recognition on speech enhanced with class-specific dictionaries

Algorithm 3

1. *Enhance the noisy data using a class-independent dictionary:*

   Let $Y_f \in \mathbb{R}^{R\times 1}$ *be the noisy speech spectrum of a frame.* $D_{ind} \in \mathbb{R}^{R\times L}$ *and* $D_x \in \mathbb{R}^{R\times L}$ *be the dictionaries for class-independent speech and the noise, respectively. Using the composite dictionary* $D0 = [D_{ind}\ D_x]$ *, the sparse coefficients of the noisy speech are obtained as*

$$[c_s^{ind}\ c_x] = LARC(Y_f, D0, \mu_{coh}) \tag{2.15}$$

   where μ_{coh} *is the threshold on residual coherence and* c_s^{ind} *represents the sparse coefficient vector corresponding to* D_{ind}*. Clean speech is estimated as*

$$\hat{S}_f = D_{ind} \times c_s^{ind} \tag{2.16}$$

2. *Get the phoneme labels using a phoneme recognizer on this enhanced speech. From the phoneme labels, obtain both the MOA and POA class labels of each frame.*

3. *Perform class-specific enhancement of the original noisy data using the dictionary corresponding to the obtained class label: Three different enhancements are carried out based on the MOA, POA and PHN labels of the frame obtained from step 2.*

   Method 1: *In this method, depending on the MOA class label the enhanced speech observation $\hat{S}_f$ is assigned to, the corresponding dictionary is chosen for enhancing the original noisy speech observation Y_f. Let the class label be c^*; $1 \leq c^* \leq 5$. Thus, the sparse coefficients and the clean speech estimate obtained using composite dictionary $D1 = [D_{c^*}^{MOA}\ D_x]$ are*

$$[c_s^{MOA}\ c_x^{MOA}] = LARC(Y_f, D1, \mu_{coh}) \tag{2.17}$$

$$\hat{S}_f^{MOA} = D_{c^*}^{MOA} \times c_s^{MOA} \tag{2.18}$$

   where c_s^{MOA} corresponds to $D_{c^*}^{MOA}$

   Method 2: *In this method, we use dictionaries based on POA, depending on the assigned label c^*; $1 \leq c^* \leq 14$, of $\hat{S}_f$. The sparse coefficients and the clean speech estimate obtained using the composite dictionary $D2 = [D_{c^*}^{POA}\ D_x]$ are*

$$[c_s^{POA}\ c_x^{POA}] = LARC(Y_f, D2, \mu_{coh}) \tag{2.19}$$

$$\hat{S}_f^{POA} = D_{c^*}^{POA} \times c_s^{POA} \tag{2.20}$$

   where c_s^{POA} corresponds to $D_{c^*}^{POA}$

   Method 3: *This method employs dictionaries based on the assigned PHN labels; $1 \leq c^* \leq 39$, of $\hat{S}_f$. Using the composite dictionary $D3 = [D_{c^*}^{PHN}\ D_x]$, the sparse coefficients and the clean speech are estimated as*

$$[c_s^{PHN}\ c_x^{PHN}] = LARC(Y_f, D3, \mu_{coh}) \tag{2.21}$$

$$\hat{S}_f^{PHN} = D_{c^*}^{PHN} \times c_s^{PHN} \tag{2.22}$$

   where c_s^{PHN} corresponds to $D_{c^*}^{PHN}$

4. Find the performance of the phoneme (PHN) recognizer on the enhanced speech in each case; (2.18)*,* (2.20) *and* (2.22)*.*

2.3.3.1 Results and discussion

Improvements in the phoneme recognition accuracies are compared across the three enhancement methods for 0, 5 and 10 dB SNRs. Figures 2.5 (a-e) show the phoneme recognition accuracies for factory2, m109, leopard, babble and volvo noises, respectively. We compare the recognition accuracies of our method with class-independent enhancement scheme and also with four other enhancement schemes available in the literature; multi-band spectral subtraction (MBSS) [18], non causal apriori SNR estimator (NC) [38], harmonic regeneration noise reduction (HRNR) [39] and geometric spectral subtraction (GA) [19]. Our method achieves superior performance over all the other methods. Results are reported for MOA, POA and PHN enhancement using estimated labels and MOA, POA and PHN enhancement using ground truth labels.

Figure 2.5 shows that enhancement using class-specific dictionaries outperforms the class-independent enhancement in terms of phoneme recognition accuracies. This is true not only when we use class labels from the ground truth but also from the recognition of speech enhanced using class-independent dictionary (referred to as approximate labels).

For phoneme recognition, PHN based enhancement using approximate labels gives a relative accuracy improvement (RAI) of 5.5%, 3.7%, 2.4% and 2.2%, respectively for factory2, m109, leopard and babble noise over class-independent enhancement method, when averaged over SNRs 0, 5 and 10 dB. MOA based enhancement gives average RAI of 2.7%, 2.3%, 1.6% and 2.2%, respectively. Similarly for POA based enhancement, the average RAIs are 4.3%, 2.5%, 1.8% and 2.1%, respectively.

The recognition accuracies obtained from the speech enhanced using ground truth labels (2.5), show that we get higher performance as the number of classes c increases. It is to be noted that c_s in Eq. (2.4) need not be zero even if the speech component in Y_f is zero. We refer this contribution of speech atoms for representing noise as noise confusion. We observe that, as we increase the number of classes and use only one class dictionary per frame the noise confusion reduces.

A small experiment was conducted with a total of 300 Factory2 noise frames to evaluate the noise energy contribution by the speech dictionary. Let X_f be the input noise frame with no

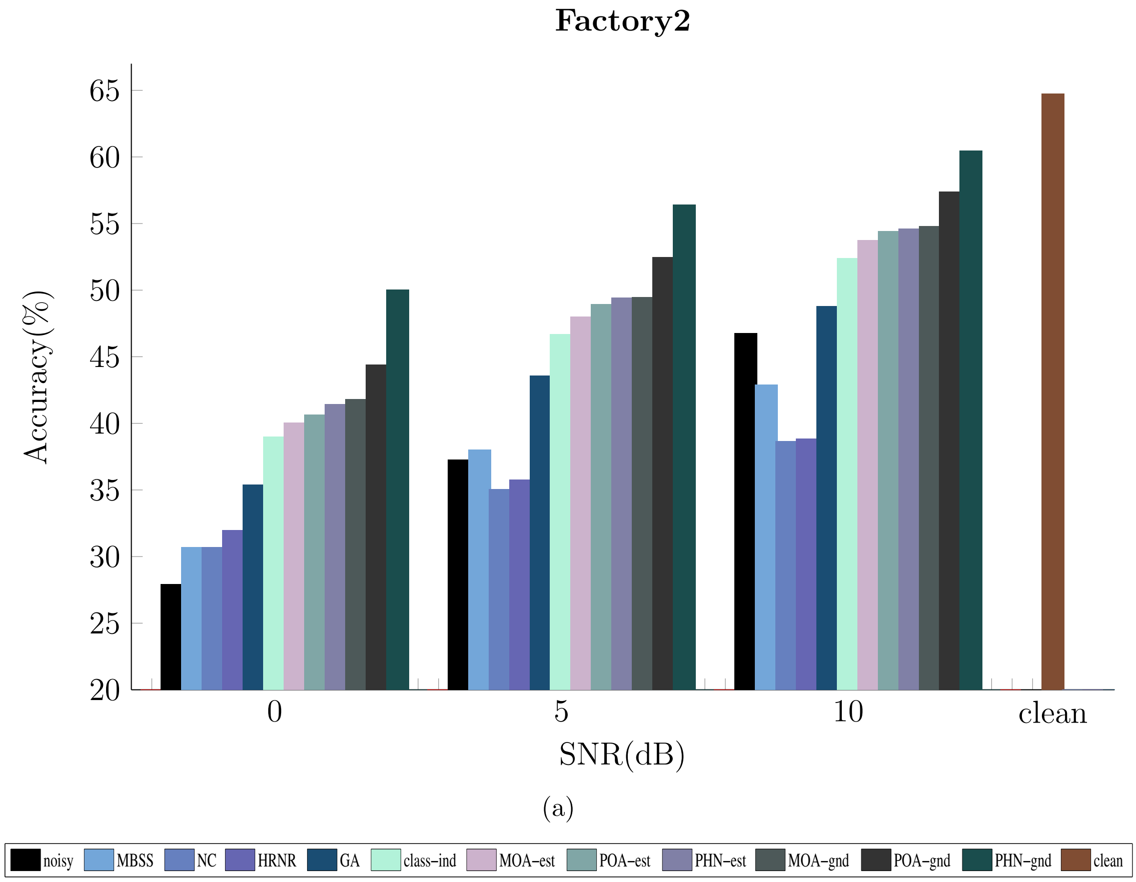


Figure 2.5: Comparison of phoneme recognition accuracies for (a) Factory2 noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class independent enhancement scheme and various class dependent enhancement schemes. In the legend, class-ind, MOA-est, POA-est, PHN-est, MOA-gnd, POA-gnd and PHN-gnd refer to class-independent case, MOA, POA, PHN enhancement using estimated labels , MOA, POA and PHN enhancement using ground truth labels, respectively. The ASR is trained on the entire clean TIMIT training data and its performance is obtained on the entire TIMIT test set excluding *sa* utterances after adding the noise from NOISEX-92 database.

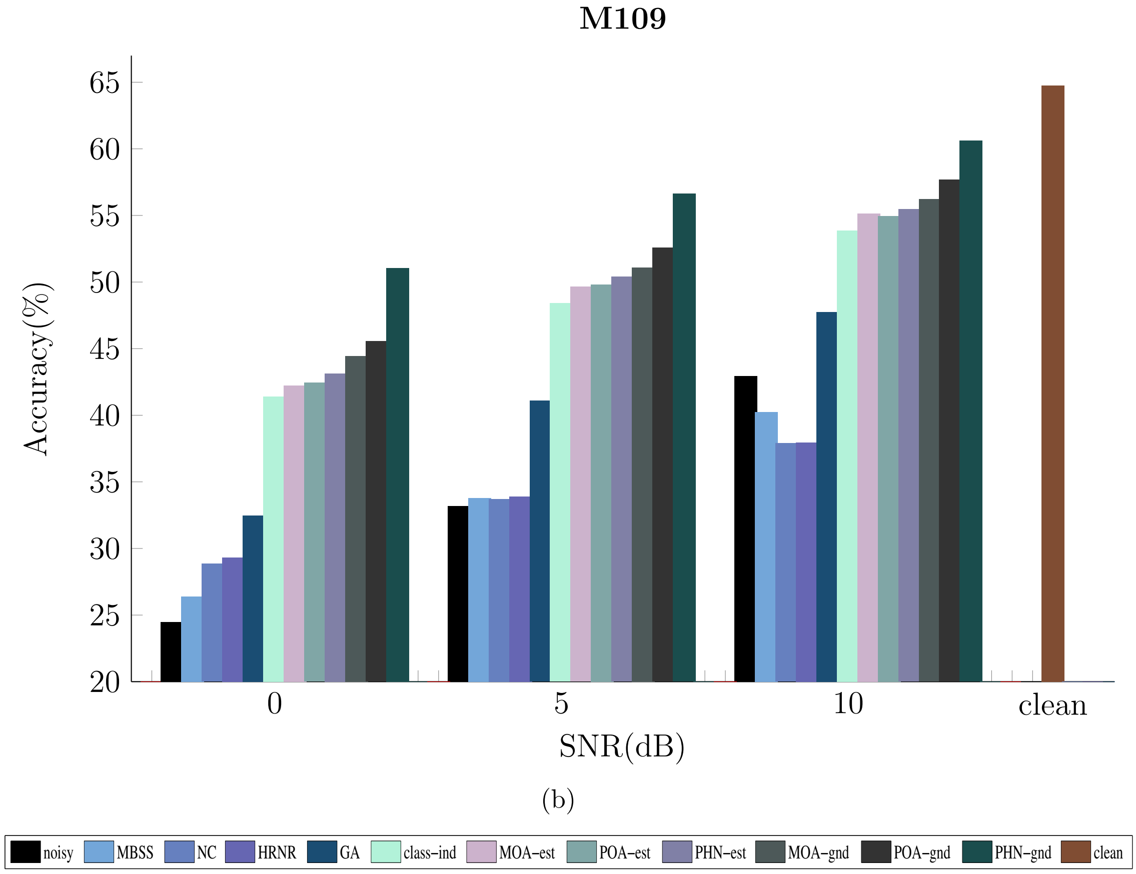
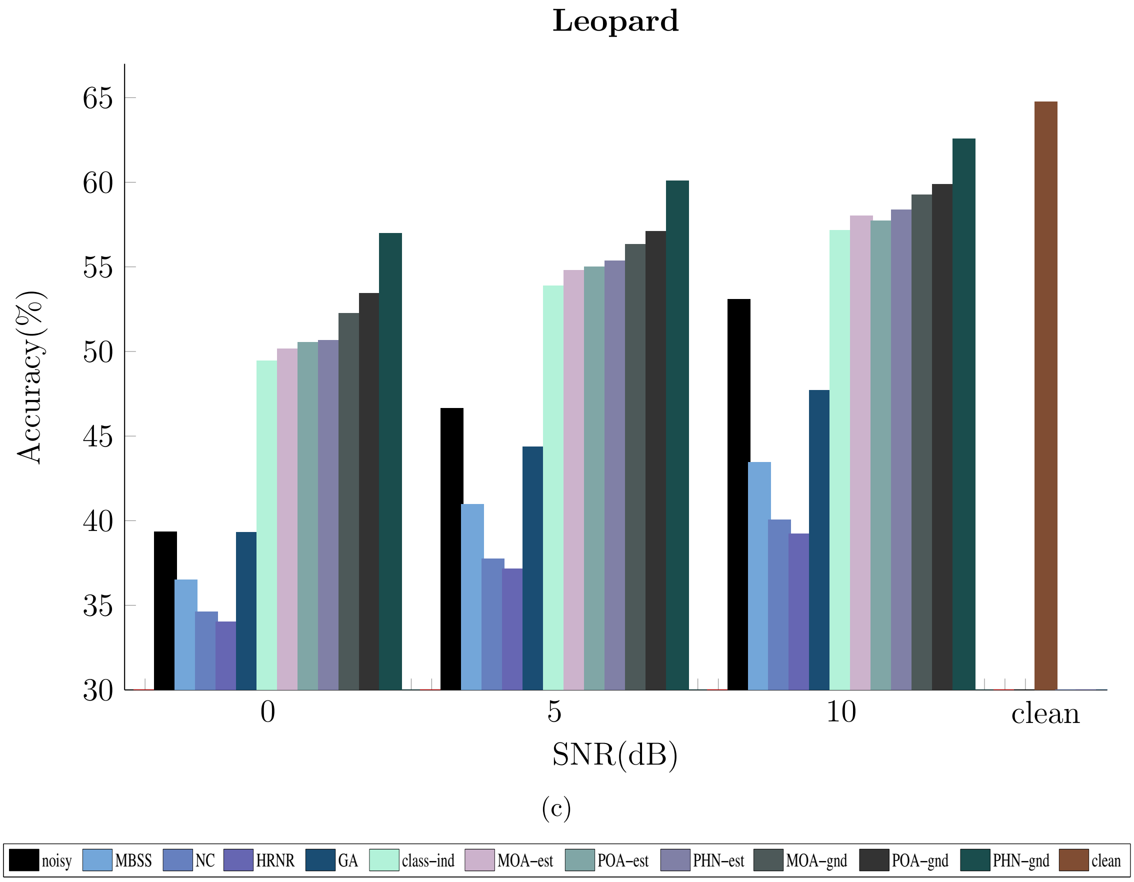


Figure 2.5: Comparison of phoneme recognition accuracies for (b)M109 noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class independent enhancement scheme and various class dependent enhancement schemes. In the legend, class-ind, MOA-est, POA-est, PHN-est, MOA-gnd, POA-gnd and PHN-gnd refer to class-independent case, MOA, POA, PHN enhancement using estimated labels , MOA, POA and PHN enhancement using ground truth labels, respectively. The ASR is trained on the entire clean TIMIT training data and its performance is obtained on the entire TIMIT test set excluding *sa* utterances after adding the noise from NOISEX-92 database.



(c)

Figure 2.5: Comparison of phoneme recognition accuracies for (c) Leopard noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class independent enhancement scheme and various class dependent enhancement schemes. In the legend, class-ind, MOA-est, POA-est, PHN-est, MOA-gnd, POA-gnd and PHN-gnd refer to class-independent case, MOA, POA, PHN enhancement using estimated labels , MOA, POA and PHN enhancement using ground truth labels, respectively. The ASR is trained on the entire clean TIMIT training data and its performance is obtained on the entire TIMIT test set excluding *sa* utterances after adding the noise from NOISEX-92 database.

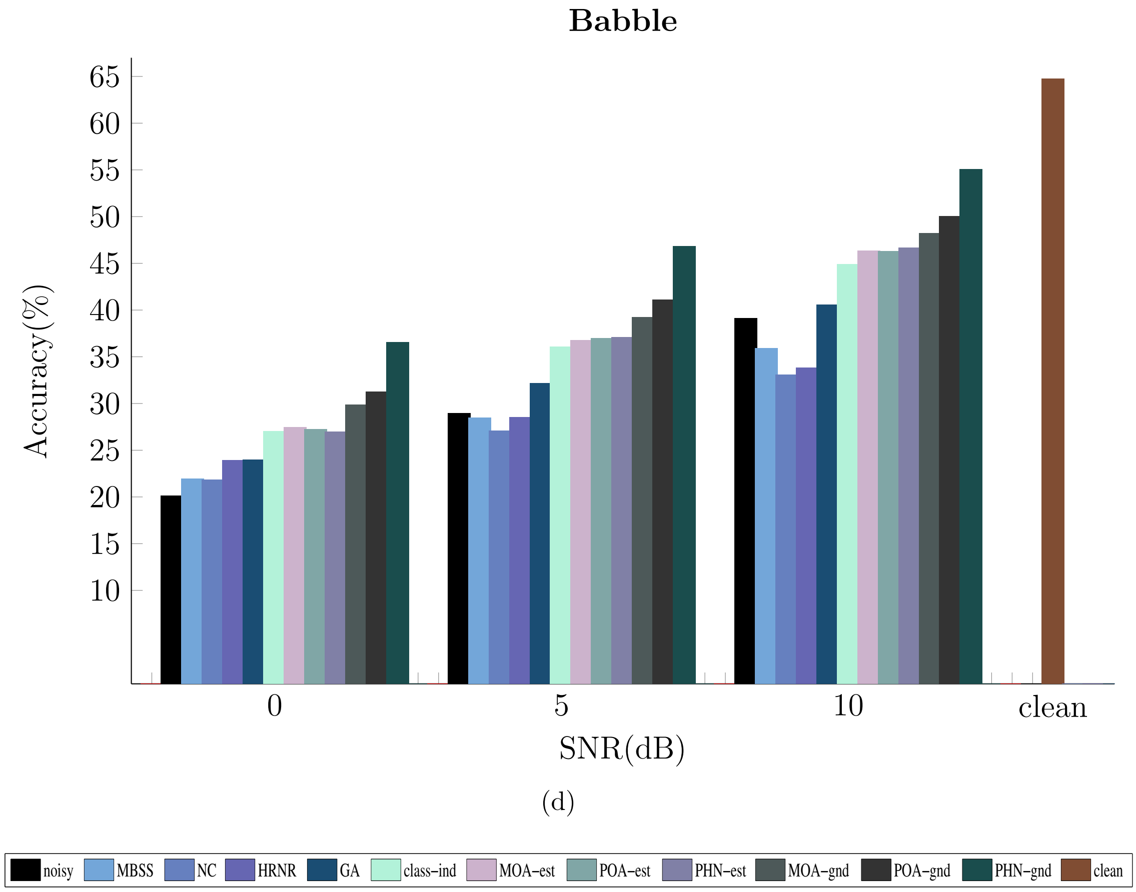


Figure 2.5: Comparison of phoneme recognition accuracies for (d) Babble noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class independent enhancement scheme and various class dependent enhancement schemes. In the legend, class-ind, MOA-est, POA-est, PHN-est, MOA-gnd, POA-gnd and PHN-gnd refer to class-independent case, MOA, POA, PHN enhancement using estimated labels , MOA, POA and PHN enhancement using ground truth labels, respectively. The ASR is trained on the entire clean TIMIT training data and its performance is obtained on the entire TIMIT test set, excluding *sa* utterances, after adding the noise from NOISEX-92 database.

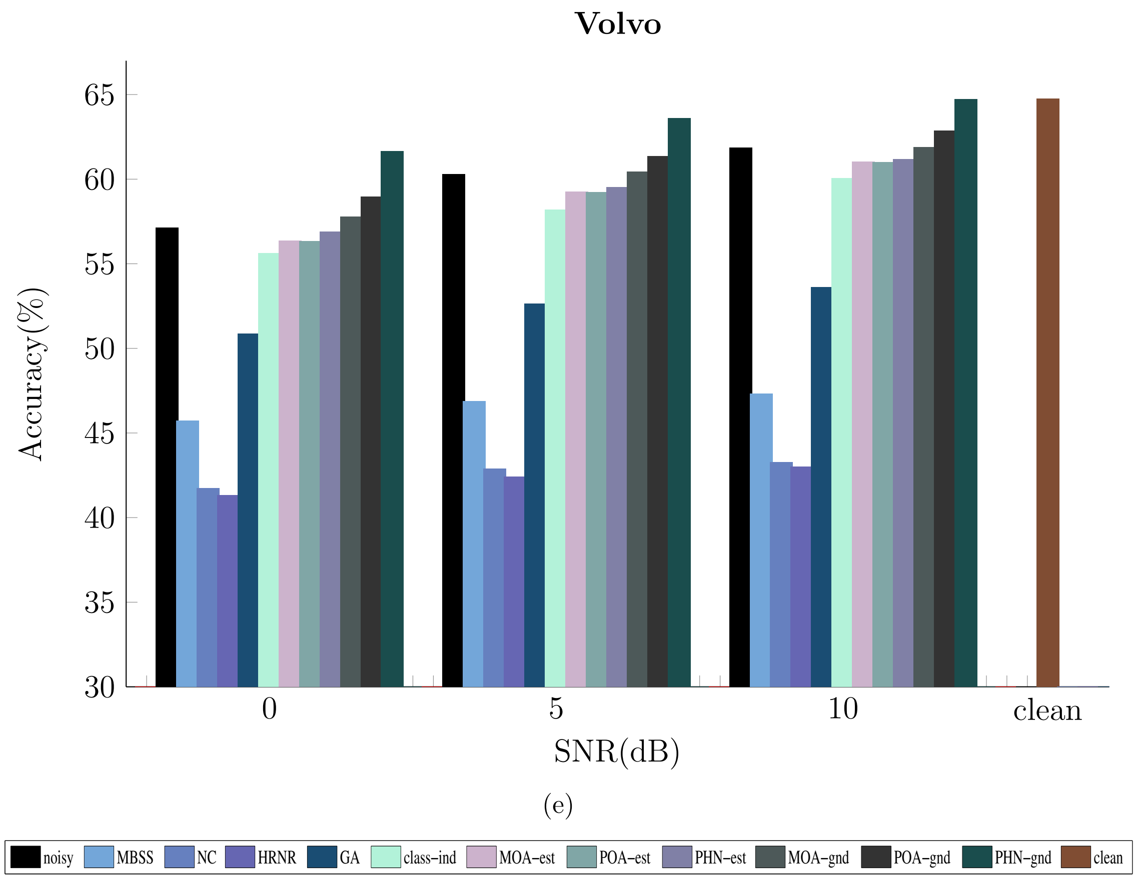


Figure 2.5: Comparison of phoneme recognition accuracies for (e) Volvo noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class independent enhancement scheme and various class dependent enhancement schemes. In the legend, class-ind, MOA-est, POA-est, PHN-est, MOA-gnd, POA-gnd and PHN-gnd refer to class-independent case, MOA, POA, PHN enhancement using estimated labels , MOA, POA and PHN enhancement using ground truth labels, respectively. The ASR is trained on the entire clean TIMIT training data and its performance is obtained on the entire TIMIT test set, excluding *sa* utterances, after adding the noise from NOISEX-92 database.

speech content. The class independent dictionary D_{ind} and noise dictionary are concatenated to form the composite dictionary $D0 = [D_{ind}\ D_x]$

The sparse coefficients of the noise input are obtained as

$$[c_s^{ind}\ c_x] = LARC(X_f, D0, \mu_{coh}) \tag{2.23}$$

where c_s^{ind} and c_x indicates the noise contribution by the speech dictionary D_{ind} and the noise dictionary D_x, respectively.

The experiment was repeated using the 39 phoneme based dictionaries. In this case, the composite dictionary is $D3 = [D_{c^*}^{PHN}\ D_x]$, $1 \leq c^* \leq 39$ and the corresponding sparse coefficients are

$$[c_s^{PHN}\ c_x^{PHN}] = LARC(X_f, D3, \mu_{coh}) \tag{2.24}$$

In our experiment, we found that the fraction of the energy of the coefficients for class-independent dictionary $\frac{energy(c_s^{ind})}{energy(c_x)}$ is 0.025. However, the fraction $\frac{energy(c_s^{PHN})}{energy(c_x^{PHN})}$ is only 0.0184 (averaged over all the phoneme classes), when the phoneme-specific dictionaries are used in place of a class-independent dictionary.

When we use approximate labels, the performance improvement also depends on the accuracy of ASR, which usually goes down as the number of classes increases. Hence to achieve the best recognition performance, one needs to choose an optimal number of classes by trading off ASR accuracy and noise confusion. It is observed that, PHN based enhancement outperforms MOA and POA based enhancements in most cases. This indicates that the approximate PHN labels obtained from the ASR are good enough to get a performance better than that from the MOA and POA labels.

For babble noise, at 0 dB SNR, no significant improvement is observed when we use the approximate labels. This could be due to the very low recognition accuracy we obtain after the enhancement using class-independent dictionary resulting in a poor choice of dictionary for most frames.

In the case of volvo noise, it is observed that after CMN, the recognition accuracy using noisy speech outperforms the class-independent and class-dependent schemes. For phoneme recognition, our PHN based enhancement using approximate labels shows an average relative degradation of -0.8% over the noisy performance. However, it is to be noted that the results for class-dependent scheme are still better than the class independent scheme. For phoneme recognition, the average RAIs over the class-independent scheme are 2.2%, 1.6% and 1.6% for PHN, MOA and POA based enhancements using approximate labels, respectively.

For volvo noise, we also did an experiment using the labels obtained after recognizing the

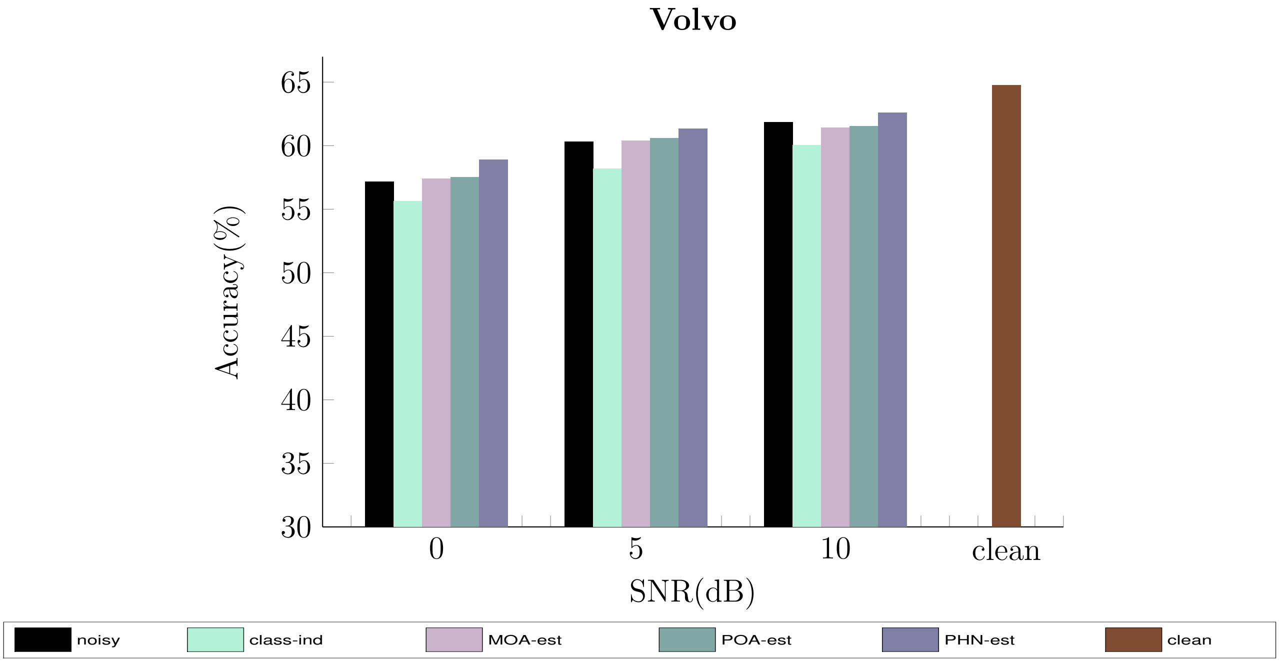


Figure 2.6: Comparison of phoneme recognition accuracies for volvo noise at 0, 5, and 10 dB SNRs after MOA, POA and PHN enhancement using approximate noisy labels.

noisy speech itself. Figure 2.6 shows the performance improvement we obtain for speech corrupted with volvo noise for MOA, POA and PHN enhancement using approximate noisy labels over both class-independent and noisy cases. It can be inferred that PHN based enhancement using approximate labels gives an average RAI of 2.1% over the noisy case whereas POA based enhancement gives an average RAI of 0.7 % over noisy case when averaged over 0, 5 and 10 dB SNRs. For MOA we do not get any improvement when averaged over the 3 SNRs. Thus it could be inferred that, for phoneme recognition of speech corrupted by highly band-limited, predictable and strictly stationary noises like volvo, most enhancement techniques distorts the speech, causing a performance degradation and hence noisy speech after CMN itself could give a better recognition performance.

Figure 2.7 shows the log magnitude spectral plots of a few exemplary frames which are correctly recognized after PHN based enhancement using ground truth labels (PHN-gnd) but wrongly recognized after class independent enhancement, one each for the noises factory2, m109, leopard, babble and volvo at 0 dB SNR. The phoneme label of the frame is mentioned in the figures. The corresponding Itakura-Saito (IS) distortion measures (computed in the power spectral domain) with the clean spectrum is also shown in each figure. From the spectral plots it can be inferred that the spectrum recovered after PHN-gnd enhancement matches more closely with the clean speech spectrum than that after the class-independent enhancement.

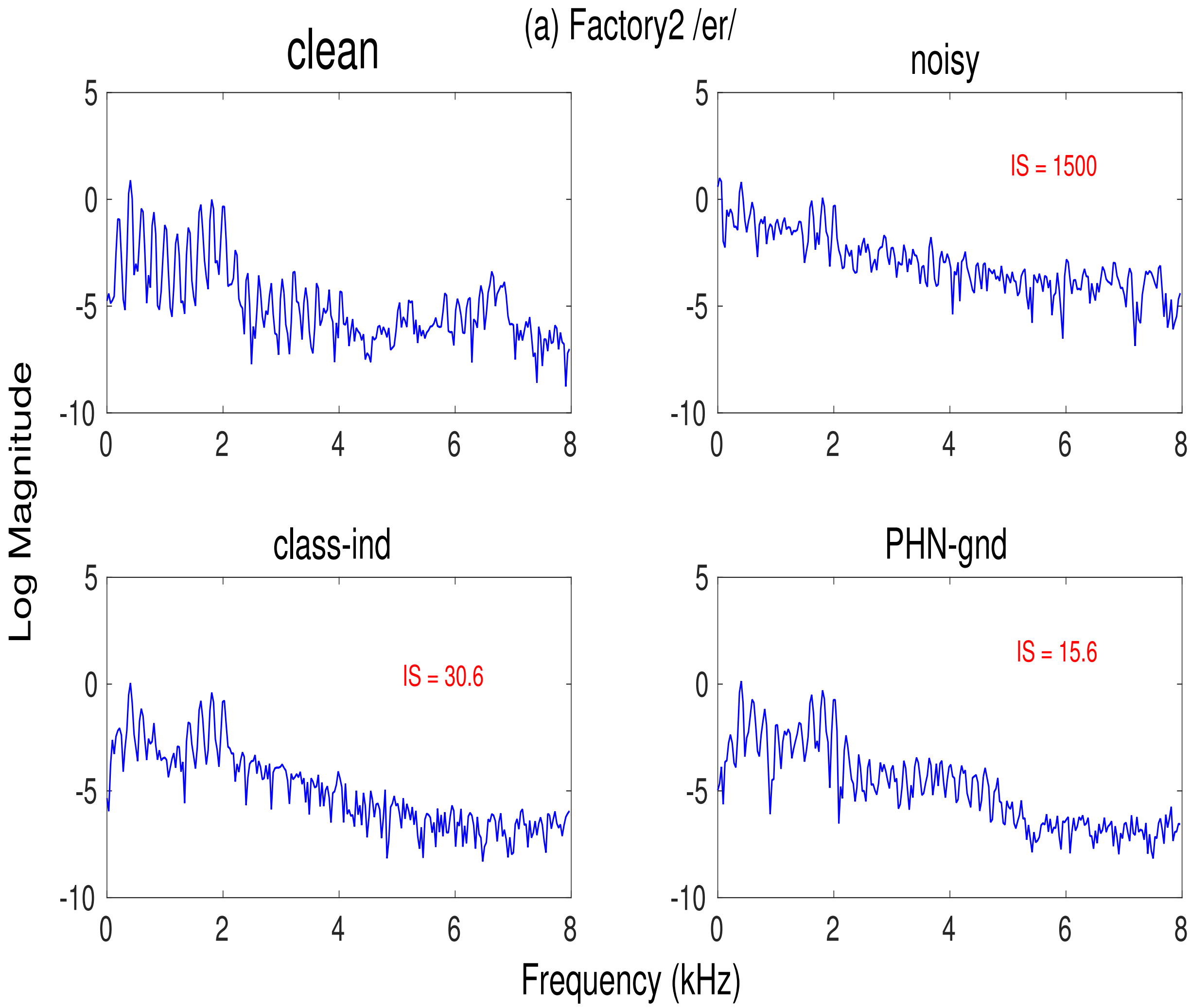


Figure 2.7: Log magnitude spectra of an example frame correctly recognized after PHN based enhancement using ground truth labels (PHN-gnd) but misclassified after class-independent (class-ind) enhancement, for (a) speech with factory2 noise at 0 dB SNR. Corresponding clean and noisy speech spectra is also shown. IS: Itakura-Saito distance computed in the power spectral domain.

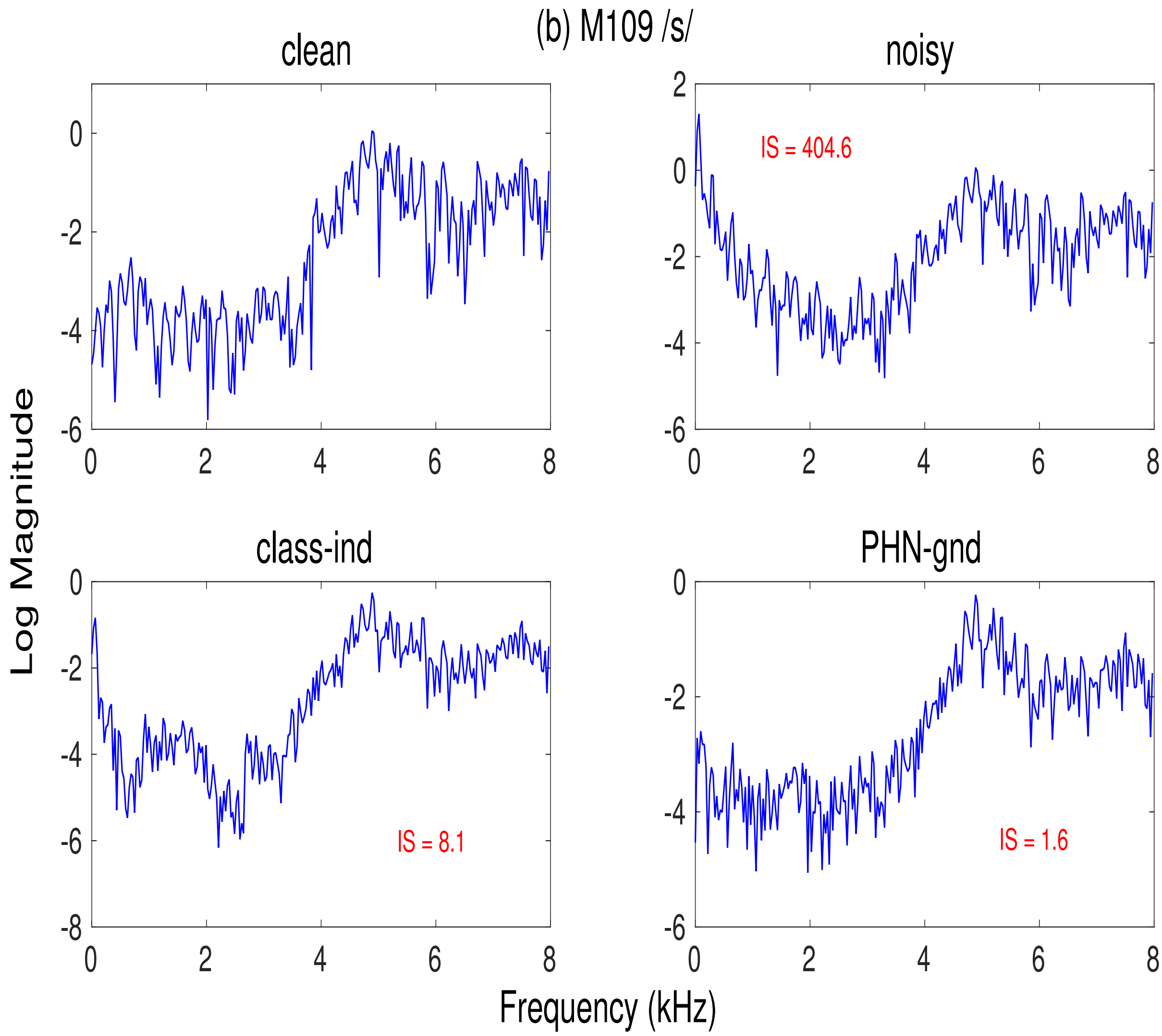


Figure 2.7: Log magnitude spectra of an example frame correctly recognized after PHN based enhancement using ground truth labels (PHN-gnd) but misclassified after class-independent (class-ind) enhancement, for (b) speech with m109 noise at 0 dB SNR. Corresponding clean and noisy speech spectra is also shown. IS: Itakura-Saito distance computed in the power spectral domain.

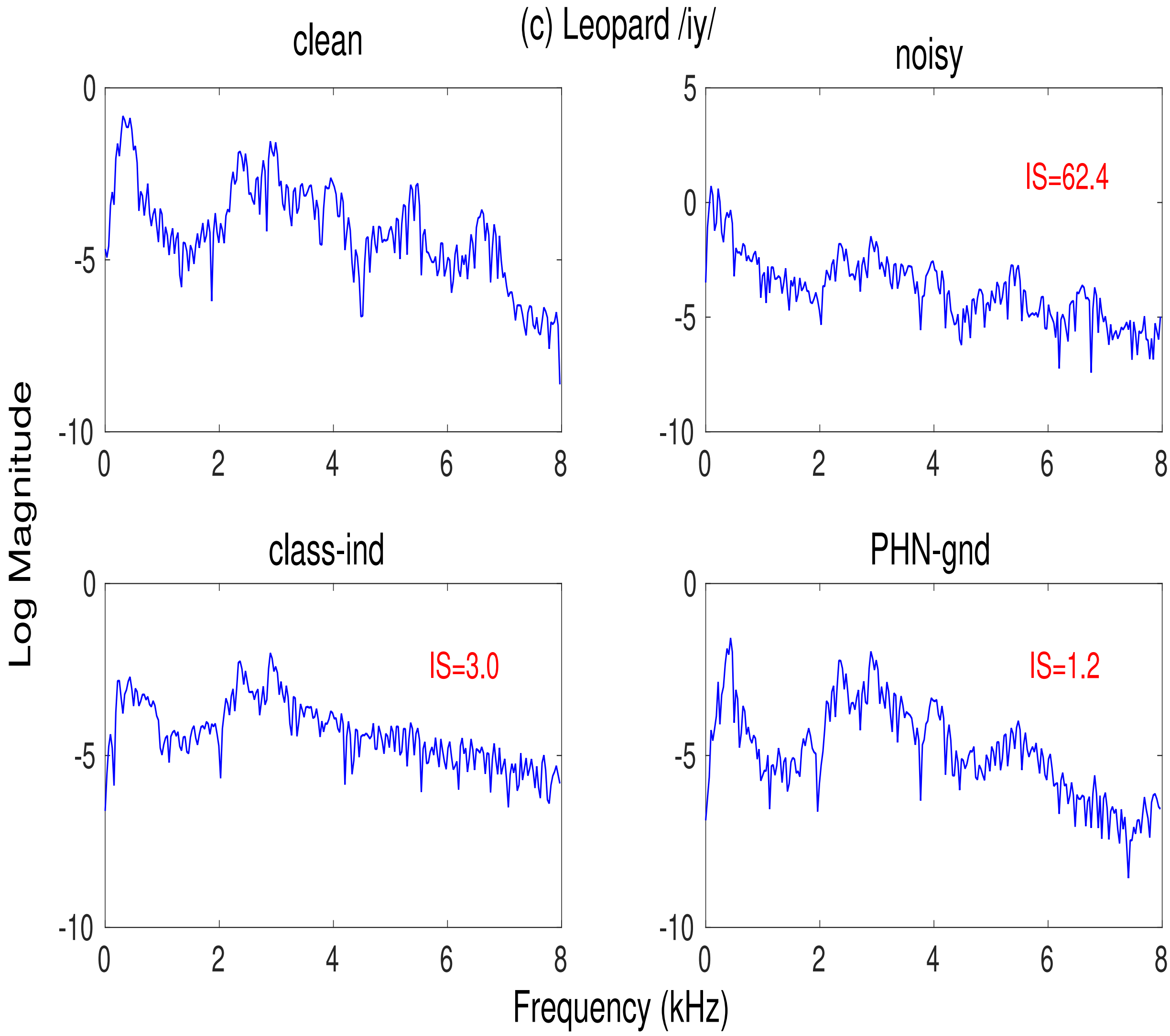


Figure 2.7: Log magnitude spectra of an example frame correctly recognized after PHN based enhancement using ground truth labels (PHN-gnd) but misclassified after class-independent (class-ind) enhancement, for (c) speech with leopard noise at 0 dB SNR. Corresponding clean and noisy speech spectra is also shown. IS: Itakura-Saito distance computed in the power spectral domain.

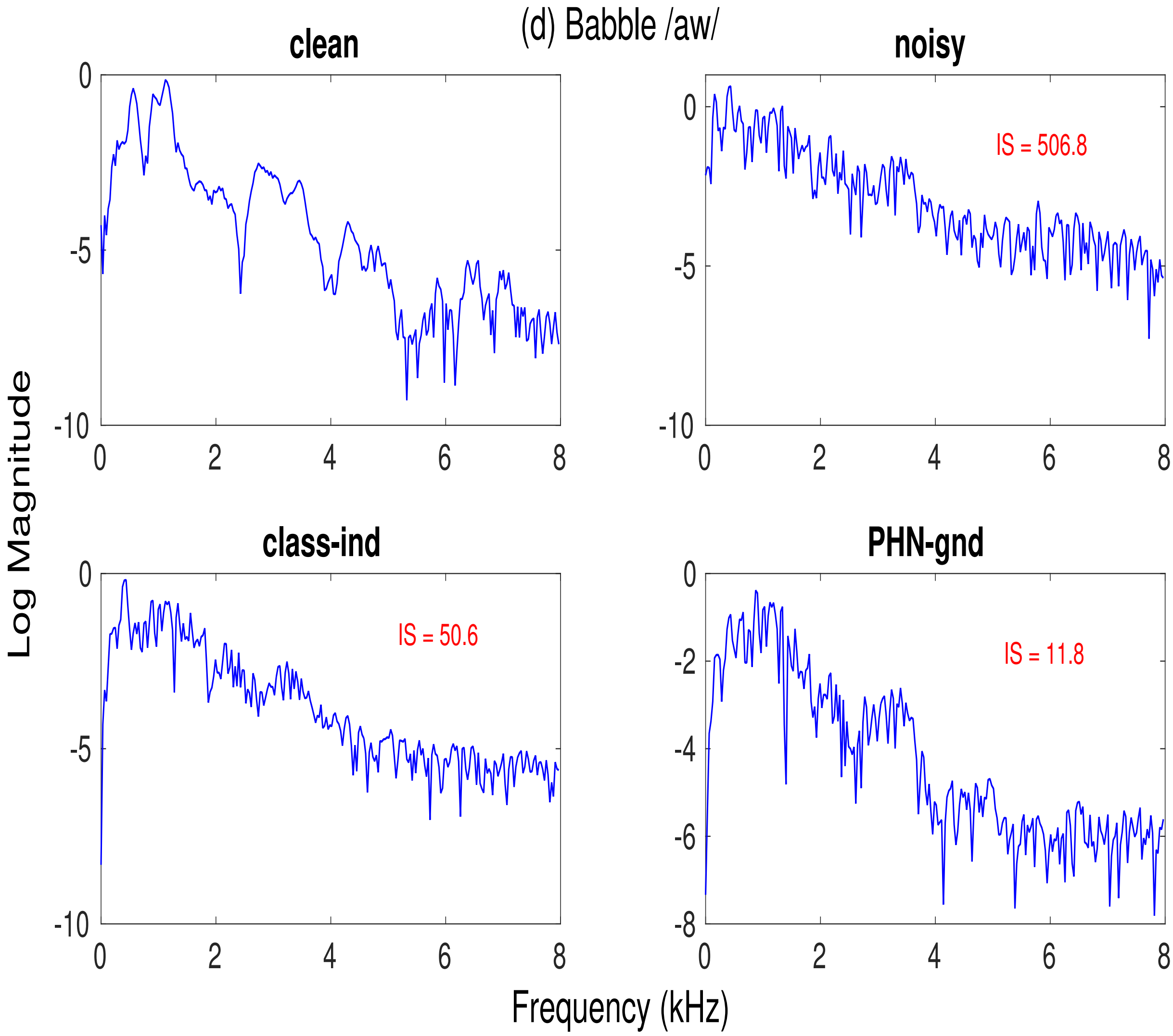


Figure 2.7: Log magnitude spectra of an example frame correctly recognized after PHN based enhancement using ground truth labels (PHN-gnd) but misclassified after class-independent (class-ind) enhancement, for (d) speech with babble noise at 0 dB SNR. Corresponding clean and noisy speech spectra is also shown. IS: Itakura-Saito distance computed in the power spectral domain.

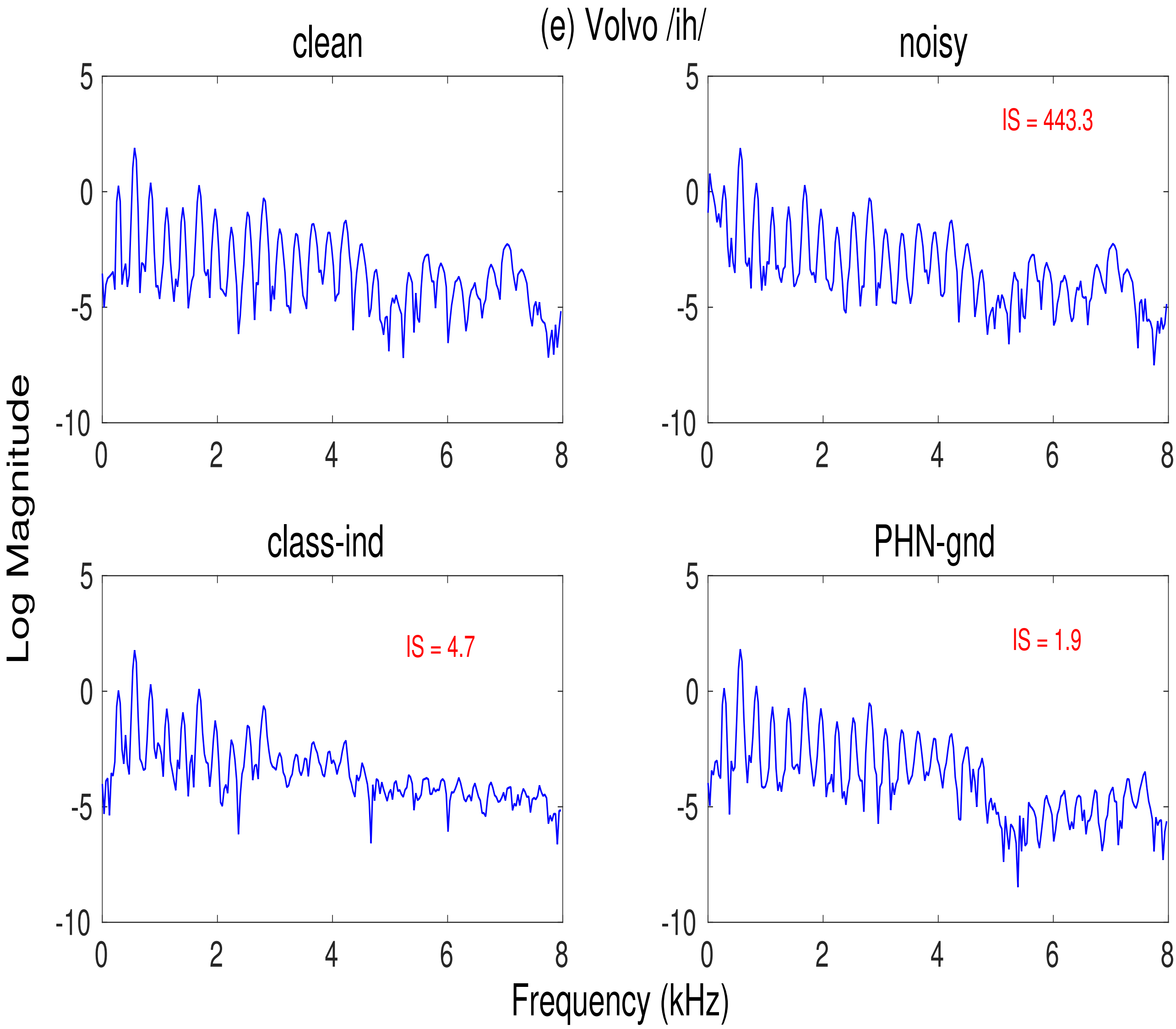


Figure 2.7: Log magnitude spectra of an example frame correctly recognized after PHN based enhancement using ground truth labels (PHN-gnd) but misclassified after class-independent (class-ind) enhancement, for (e) speech with volvo noise at 0 dB SNR. Corresponding clean and noisy speech spectra is also shown. IS: Itakura-Saito distance computed in the power spectral domain.

2.3.4 Phoneme recognition performance of multi-stage class-specific enhancement-recognition scheme

Instead of performing enhancement followed by recognition only once, one can think of a multi-stage enhancement-recognition scheme, where class-specific enhancement is performed in each stage and the required class labels are taken from the recognition outputs of the previous stage. Figure 2.8 shows the block diagram of a two stage scheme. In this case, the enhanced output of the class-specific enhancement scheme is fed to the phoneme recognizer and the label obtained from this recognizer is used to perform a second stage class-specific enhancement. The phoneme recognition performance of this enhanced output is then analyzed.

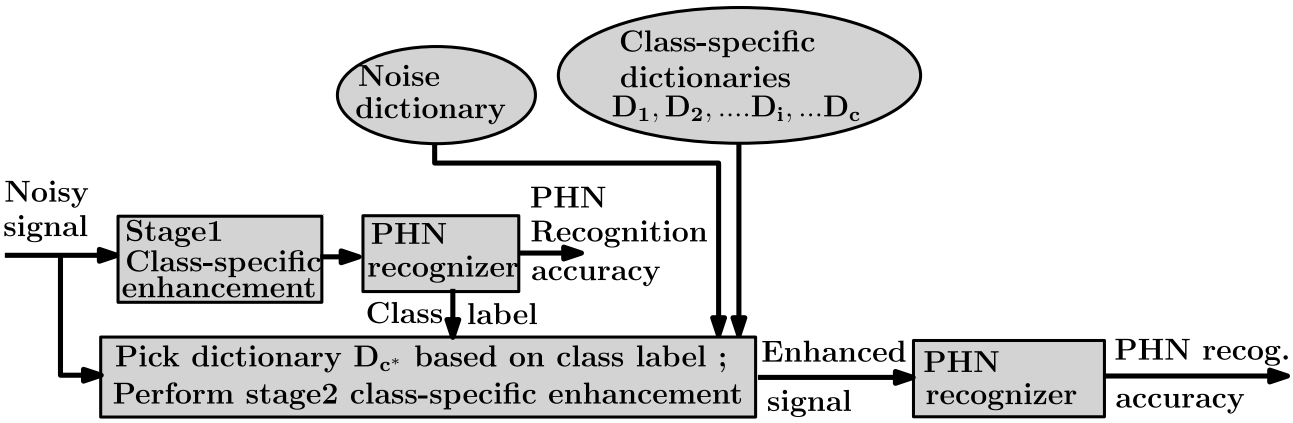
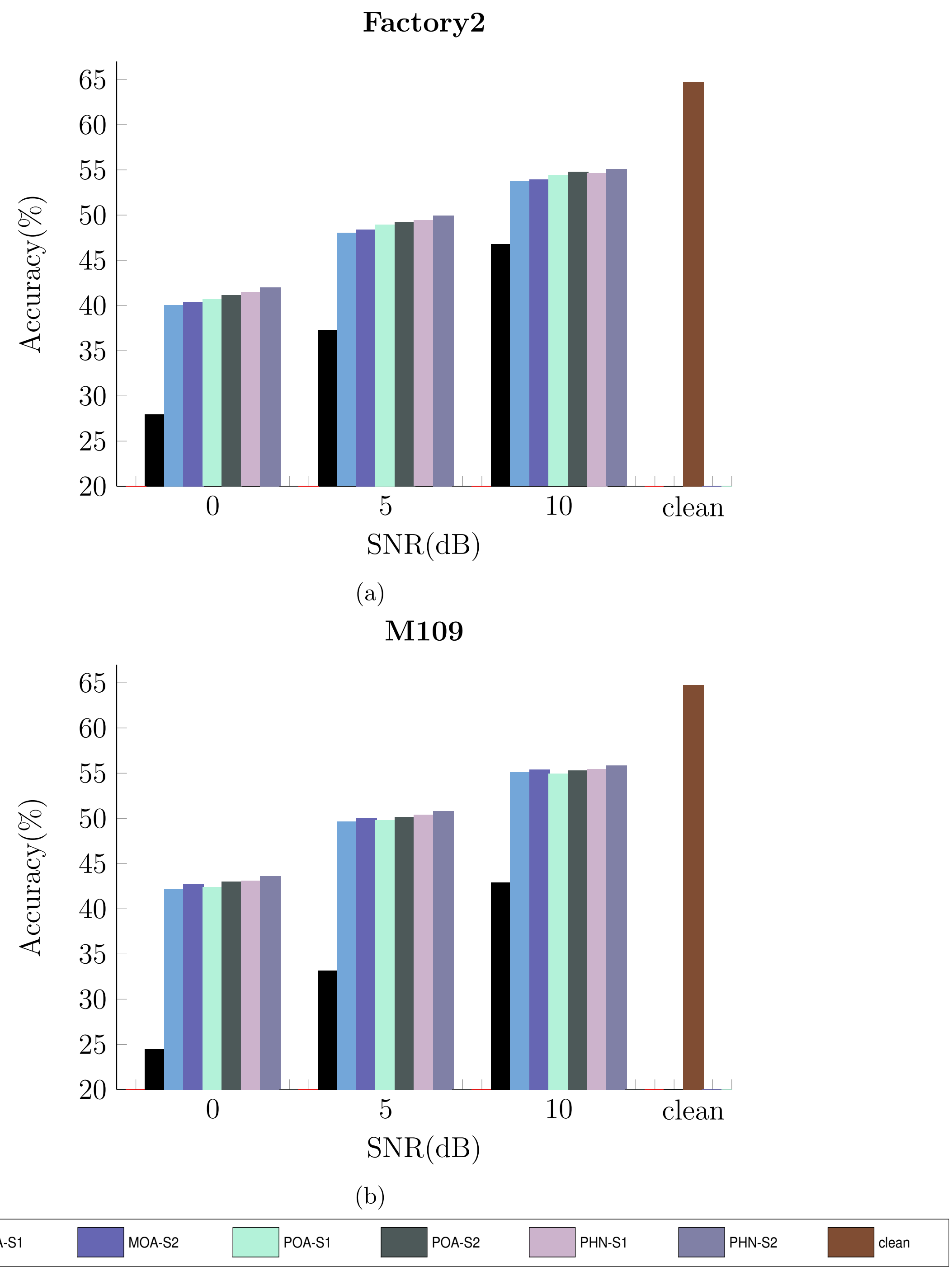


Figure 2.8: Two stage class-specific enhancement-recognition scheme

2.3.4.1 Results and discussion

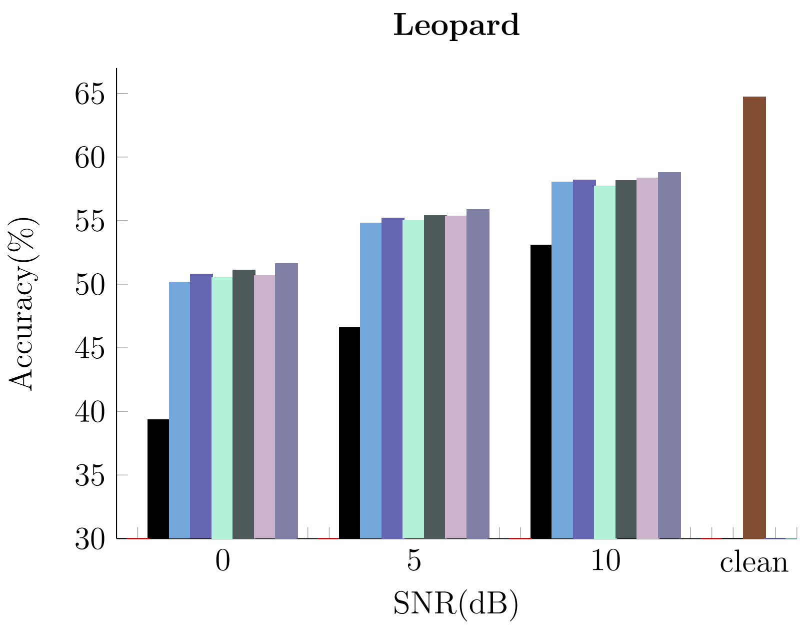
Figure 2.9 shows the phoneme recognition performance of a two stage class-specific enhancement-recognition scheme. The performance improvement over the single stage scheme is analyzed. The results show that, the two stage scheme, with PHN enhancement using estimated labels from stage 1 gives a relative accuracy improvement of 1.0%, 0.8%, 1.0% and 0.7% (averaged over 0, 5, and 10 dB SNRs) for factory2, m109, leopard and babble noises, respectively, for phoneme recognition over the single stage scheme. For MOA enhancement, the two stage scheme gives average RAI of 0.6%, 0.8%, 0.7% and 1.4% for factory2, m109, leopard and babble noises, respectively. For POA enhancement, the average RAI values are 0.8%, 0.8%, 0.8% and 1.5%, respectively, for factory2, m109, leopard and babble noises. It can be observed that the two stage scheme gives only marginal improvement over single stage scheme.



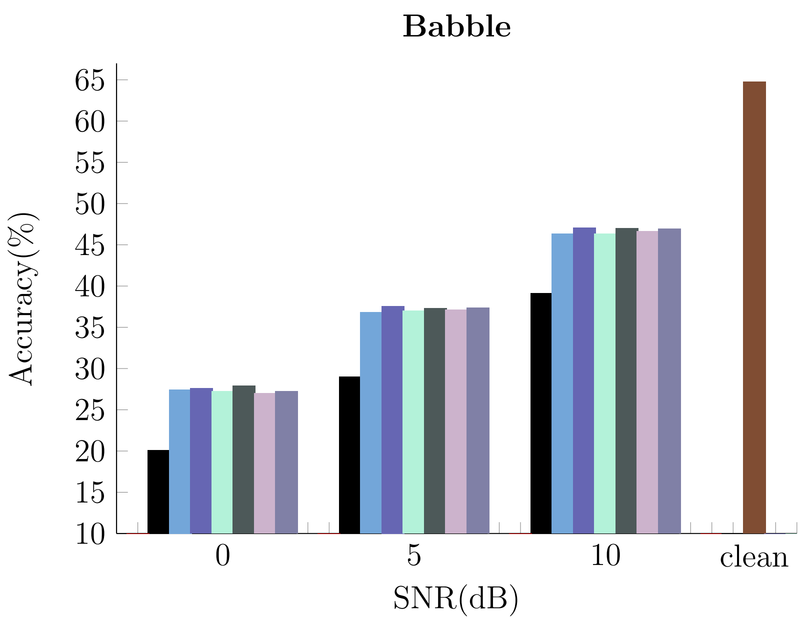
(a)

(b)

Figure 2.9: Comparison of phoneme recognition accuracies of a two stage class-specific enhancement-recognition scheme over single stage scheme. In the legend, S1 indicates single stage scheme and S2 indicates two-stage scheme. The ASR is trained on the entire clean TIMIT training data and its performance is obtained on the entire TIMIT test set, excluding *sa* utterances, after adding the noise from NOISEX-92 database.



(c)



(d)

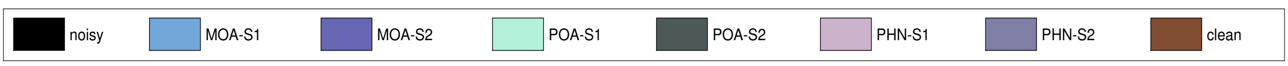


Figure 2.9: Comparison of phoneme recognition accuracies of a two stage class-specific enhancement-recognition scheme over single stage scheme. In the legend, S1 indicates single stage scheme and S2 indicates two-stage scheme. The ASR is trained on the entire clean TIMIT training data and its performance is obtained on the entire TIMIT test set, excluding *sa* utterances, after adding the noise from NOISEX-92 database.

2.3.5 Manner and place of articulation recognition performance on speech enhanced with class-specific dictionaries using estimated class labels

We also analyze the performance improvements for MOA and POA recognizers for the single-stage scheme. Figure 2.10 shows the block diagram summarizing the steps. The class-specific enhanced outputs are fed into MOA and POA recognizers and the results are compared to MOA and POA recognizer outputs after class-independent enhancement.

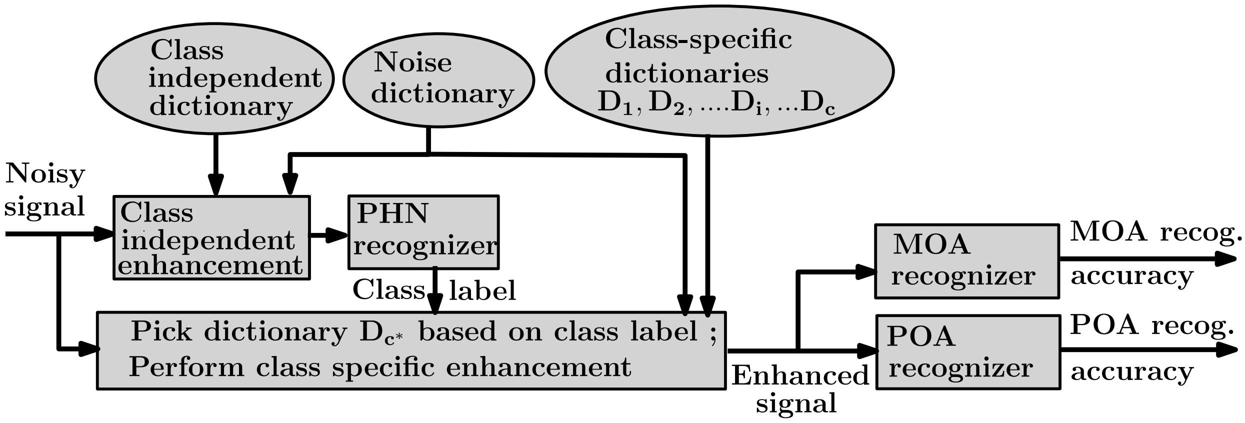


Figure 2.10: MOA and POA recognition on speech enhanced with class-specific dictionaries

2.3.5.1 Results and discussion

MOA recognition results

Figure 2.11 shows the performance of MOA recognizer for various class-specific enhancement schemes. In the case of MOA recognition, PHN based enhancement gives average relative recognition accuracy improvements of 2.4%, 2.1%, 2.3% and 2.8% for factory2, m109, leopard and babble noises, respectively, over class-independent enhancement, while MOA based enhancement gives improvements of 1.3%, 1.1%, 1.7% and 2.1%. For POA based enhancement, the average RAIs are 1.7%, 0.8%, 1.3% and 2.4%, respectively.

POA recognition results

The POA recognition performance of various class-specific enhancement schemes are shown in Fig. 2.12. For POA recognition, PHN based enhancement gives average RAIs of 2.9%, 2.1%, 2.4% and 0.7%. MOA based enhancement achieves improvements of 2.3%, 1.8%, 1.9% and 0.9%. For POA based enhancement, we get improvements of 3.3%, 2.2%, 2.7% and 0.8%.

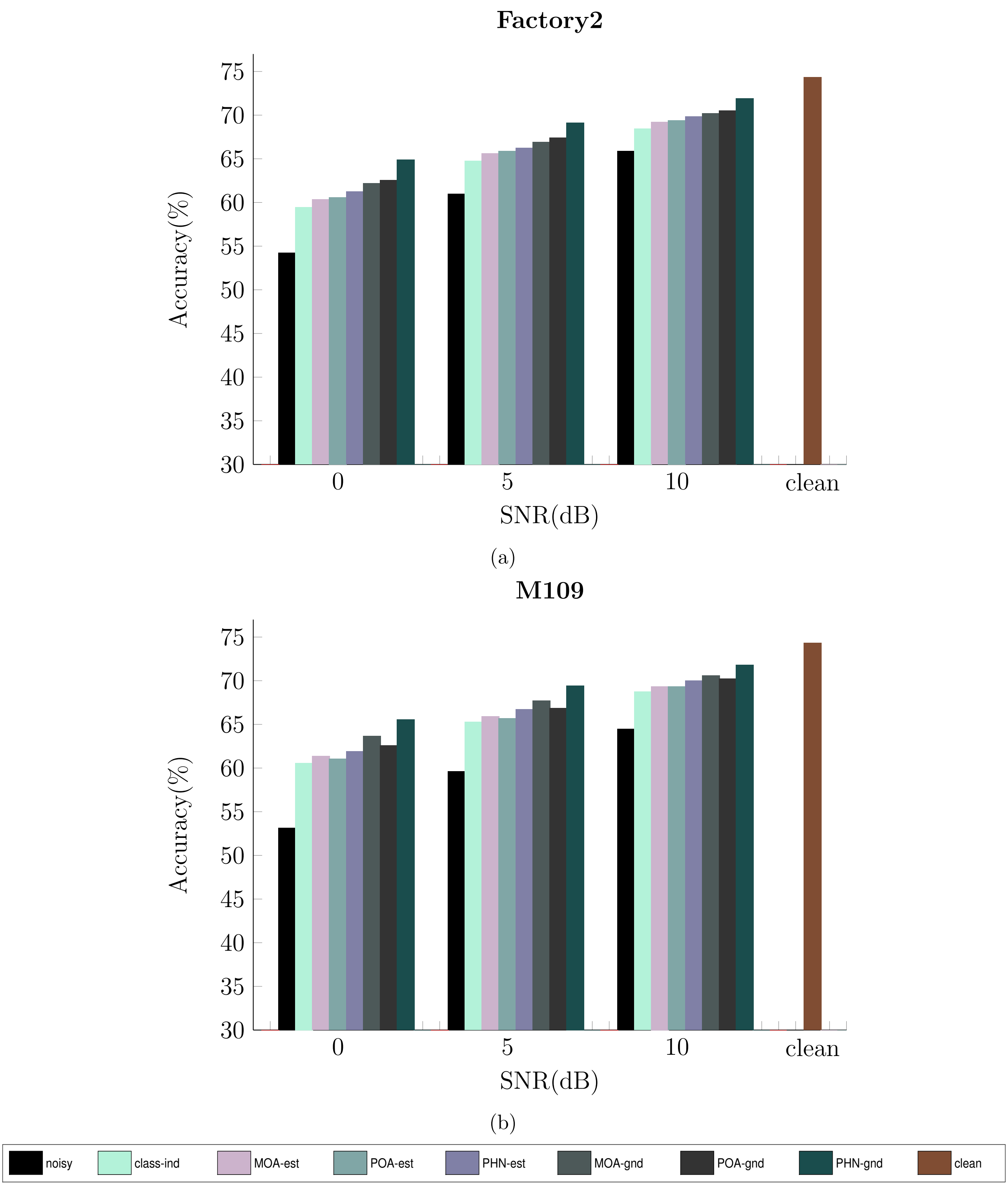
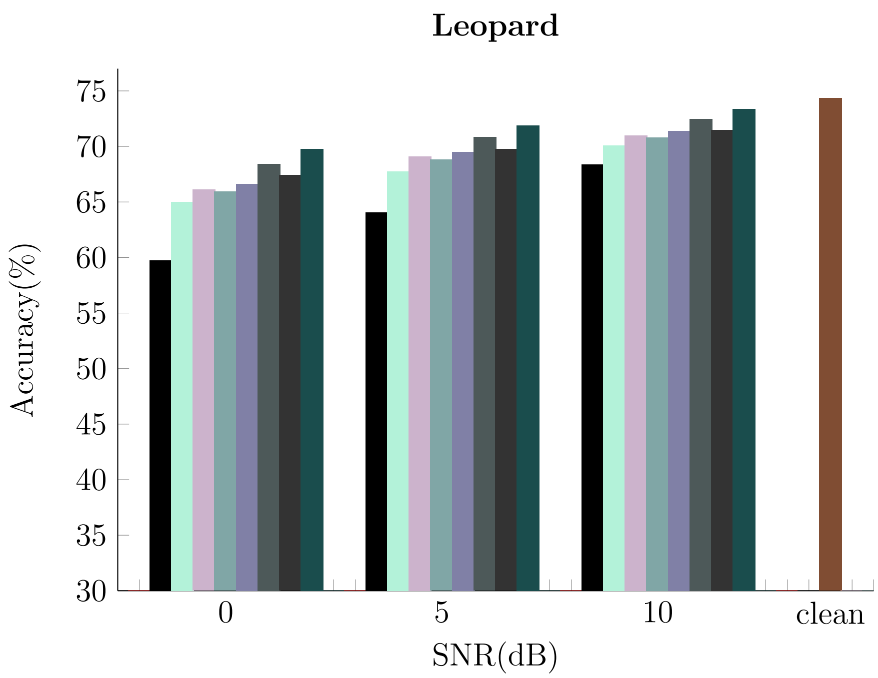
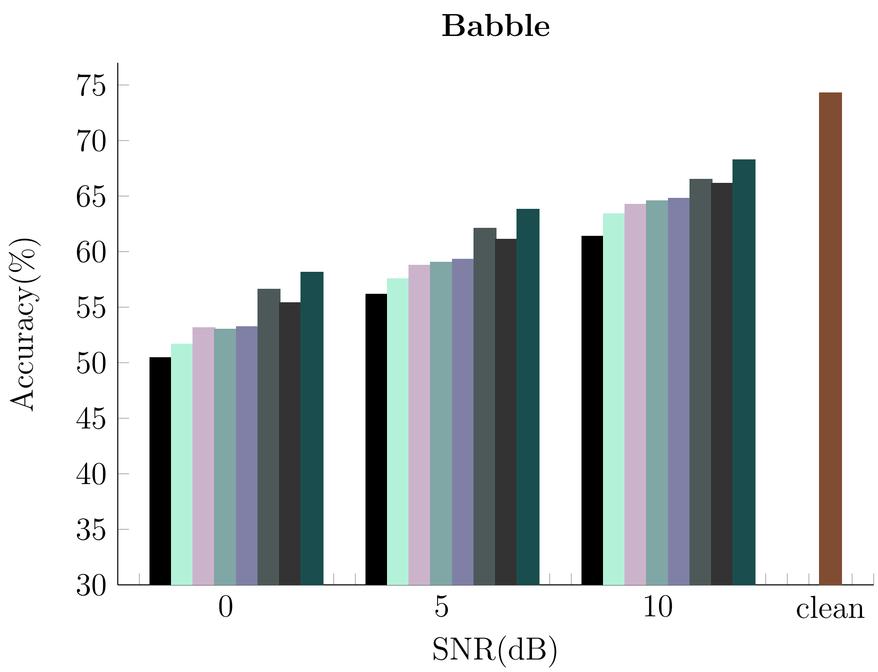


Figure 2.11: Comparison of manner of articulation recognition performance on speech enhanced with class-specific dictionaries using estimated class labels with the one with class-independent enhancement for (a) Factory2 and (b) M109 noises.



(c)



(d)

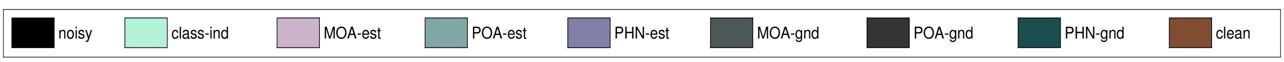


Figure 2.11: Comparison of manner of articulation recognition performance on speech enhanced with class-specific dictionaries using estimated class labels with the one with class-independent enhancement for (c) Leopard and (d) Babble noises.

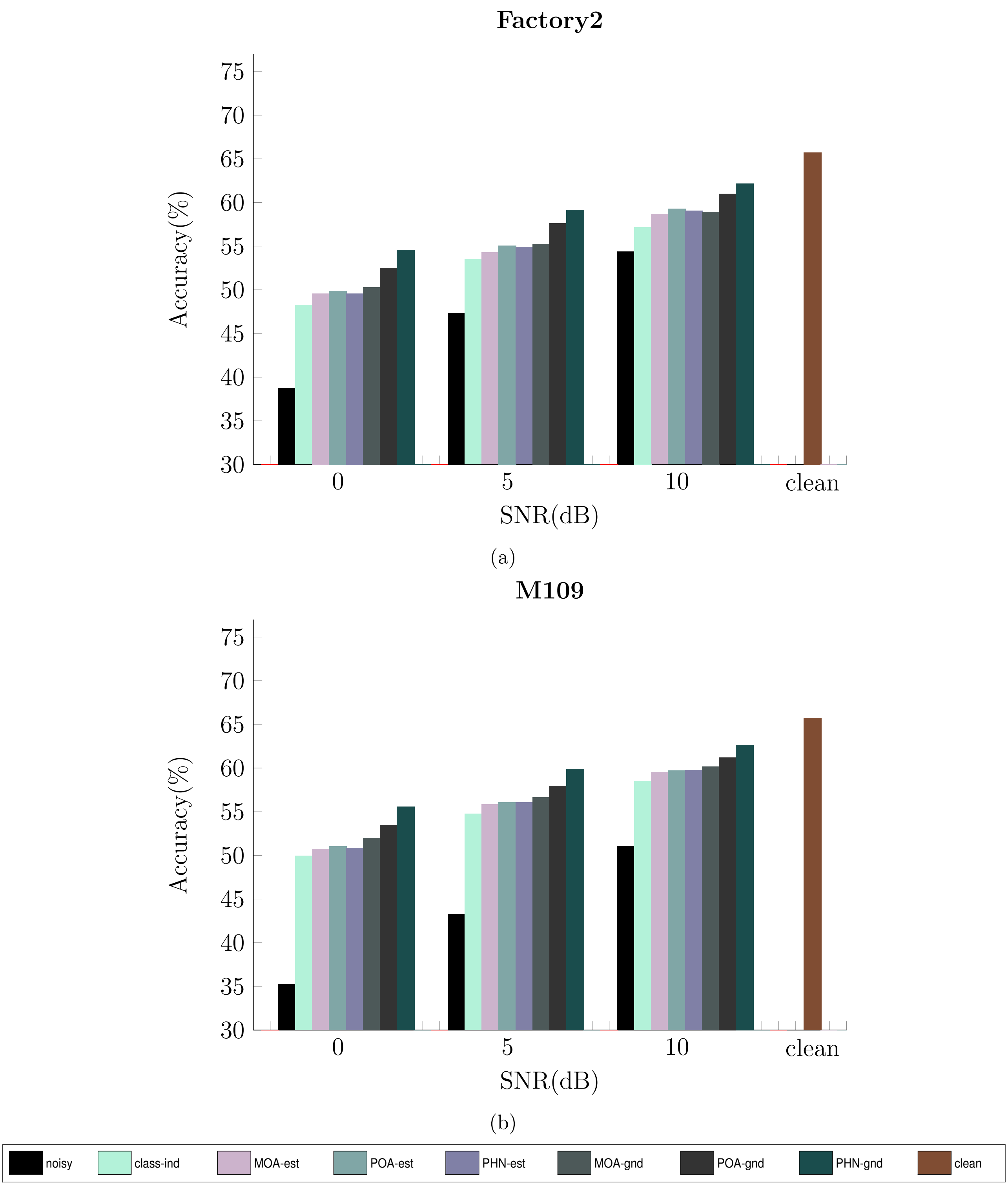
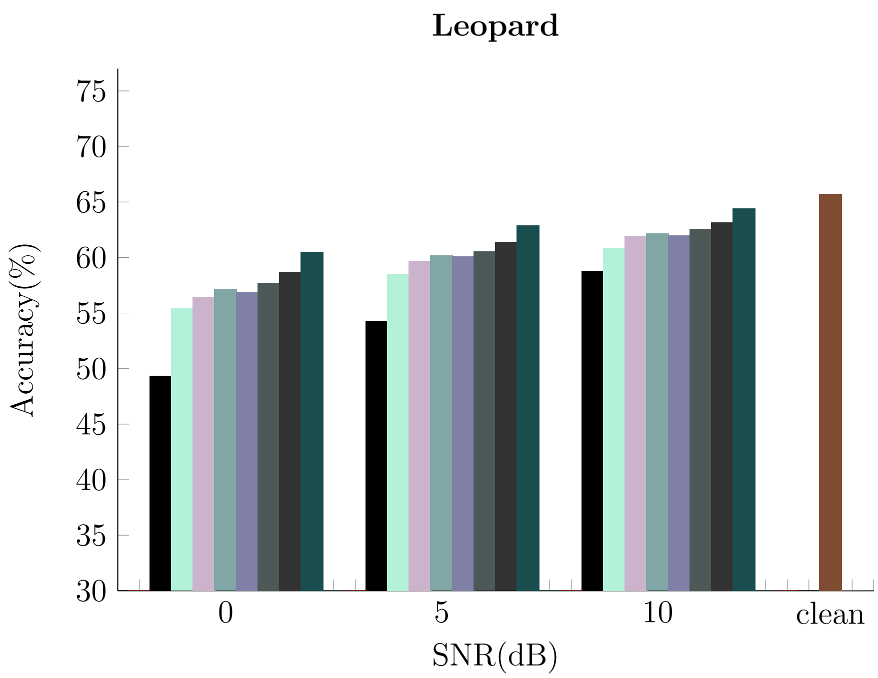
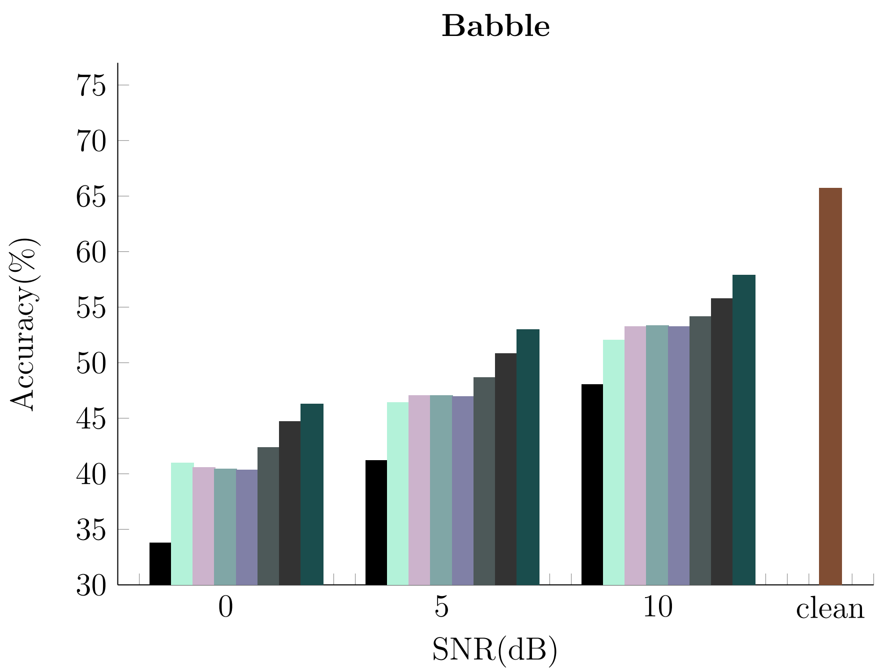


Figure 2.12: Comparison of place of articulation recognition performance on speech enhanced with class-specific dictionaries using estimated class labels with the one with class-independent enhancement for (a) Factory2 and (b) M109 noises.



(c)



(d)

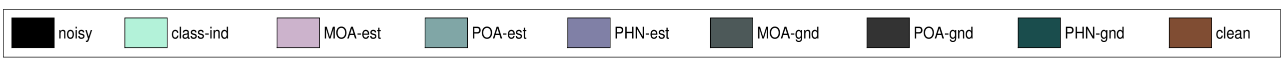


Figure 2.12: Comparison of place of articulation recognition performance on speech enhanced with class-specific dictionaries using estimated class labels with the one with class-independent enhancement for (c) Leopard and (d) Babble noises.

2.4 Conclusions

We have analyzed how the performance of the enhancement schemes vary when we use class-specific dictionaries for enhancement rather than a class-independent dictionary in terms of objective quality measures like PESQ, SSNR as well as in terms of recognition accuracy. The experiments are carried out in a speaker independent scenario. With the ground truth class labels, there is significant improvement in recognition accuracy for class-specific enhancement over the class-independent scheme but not much improvement was found in terms of objective quality measures. When ground truth labels are used, the 39-PHN based enhancement gives average RAI in phoneme recognition of 21.5%, 17.6%, 12.1%, 29.2% and 9.3% for factory2, m109, leopard, babble and volvo noises, respectively, over class-independent enhancement.

The 39-PHN based enhancement outperforms the MOA and POA based schemes in most of the cases. Using the approximate labels obtained from the ASR gives better recognition accuracy than the class-independent enhancement, although it is lower than that using the ground truth labels. Future work could employ a DNN framework with other features like multistream features [100] for recognition and examine the benefit of class-specific enhancement, since it has been shown to perform significantly better than GMM-HMM framework with MFCC features. Also we would like to examine the usefulness of our algorithms on a more realistic scenario involving real world speech mixed with noise.

Chapter 3

A joint enhancement-decoding formulation for noise robust phoneme recognition

We consider a dictionary based speech enhancement in the context of automatic recognition of noisy speech. Speech in each analysis frame is denoised as a front-end processing using a class-specific (e.g. phoneme) dictionary selected based on the estimated class label. However, when the estimated label is erroneous, a wrong class model is chosen for many frames. We propose a joint enhancement-decoding (JED) algorithm to overcome this issue by jointly optimizing for labels of all the frames and the decoding path. The algorithm optimizes over multiple enhanced versions of each frame using different phoneme-specific dictionaries and gives the maximum likelihood path of state sequences as well as the best (in the maximum likelihood sense) choice of the enhanced observation sequence as its output. The number of phoneme-specific dictionaries used for enhancement in an analysis frame is varied from 1 to 5 based on the phoneme confusion matrix and the recognition results are reported for each case. Experiments with TIMIT corpus and five different noises at 0, 5 and 10 dB SNRs show that the recognition performance varies with the number of dictionaries, and in most of the cases, is the best when two or three dictionaries are employed.

3.1 Introduction

In the past decade, there has been tremendous improvements in the field of automatic speech recognition (ASR). Despite these, the performance of an ASR system degrades significantly in the presence of noise due to the mismatch between the training and test environments, for example, when training is done on clean speech and testing is performed on noisy speech. The presence of noise distorts the spectrum of speech and hence degrades the performance.

Several techniques have been proposed to address this problem, and improve the recognition performance in noisy environments. One such approach is to employ model adaptation schemes, like parallel model combination [101] and HMM adaptation [102–104]. Another approach is to analyze the existing features and enhance them to make them more noise-robust, like cepstral mean subtraction [105], RASTA filtering [106] and vector Taylor series [107]. A third approach is to enhance the speech as a front-end processing, using methods such as spectral subtraction [108] or Wiener filtering [109] before it is fed into a recognizer. This obviates the need to retrain the ASR systems for different types of noisy inputs since the same ASR trained on clean speech can be used. A comparative study [10] has also been reported on the performance of ASR systems with various enhancement approaches. In Chapter 2, we explored the enhancement performance of using class-specific dictionaries and found it to be particularly useful for phoneme recognition in noisy environment [11].

3.1.1 Motivation

All the class specific enhancement schemes mentioned so far [11, 90, 91] depend on the estimated class label for each frame, which may be erroneous. This leads to the selection of wrong class model for the enhancement in the respective frames. To overcome this, we propose a joint enhancement-decoding (JED) algorithm that jointly optimizes these class labels and the final recognized speech labels [110]. By this approach, we aim to find the best possible frame-wise model for enhancement and the recognition labels together for an input speech signal in a single optimization framework. We develop this algorithm by integrating the class label estimation into the Viterbi decoder [111] typically used for speech recognition. We implement the same using the HTK toolkit. The proposed algorithm accepts multiple enhanced observations and chooses the best in each frame such that the overall likelihood is maximized. Multiple observations are obtained by enhancing every noisy frame using multiple class-specific dictionaries. The best sequence of observations is chosen to maximize the likelihood. Thus we do not separately choose a class label and consequently the class model for frame-wise enhancement as in [11] [90] or [91].

We analyze the performance of our algorithm on TIMIT database. We use the confusion matrix obtained from the recognition output of clean speech to select the pool of dictionaries. This results in an improvement in the performance in most of the cases over the enhancement using a class-independent dictionary. It is to be noted that when the number of dictionaries is set to 1, the algorithm becomes the same as class-specific enhancement explained in chapter 2 [11].

3.2 Class - specific enhancement combined with joint enhancement - decoding algorithm for phoneme recognition

Figure 3.1 shows a generic class-specific enhancement-recognition framework. This scheme depends on the estimated frame labels of noisy speech. Once the class labels of the frames are obtained, the corresponding class-specific dictionaries are used for enhancement. This estimate of class labels is often erroneous, which results in the selection of wrong class dictionary resulting in poor enhancement of the respective frames and subsequent performance reduction in phoneme recognition.

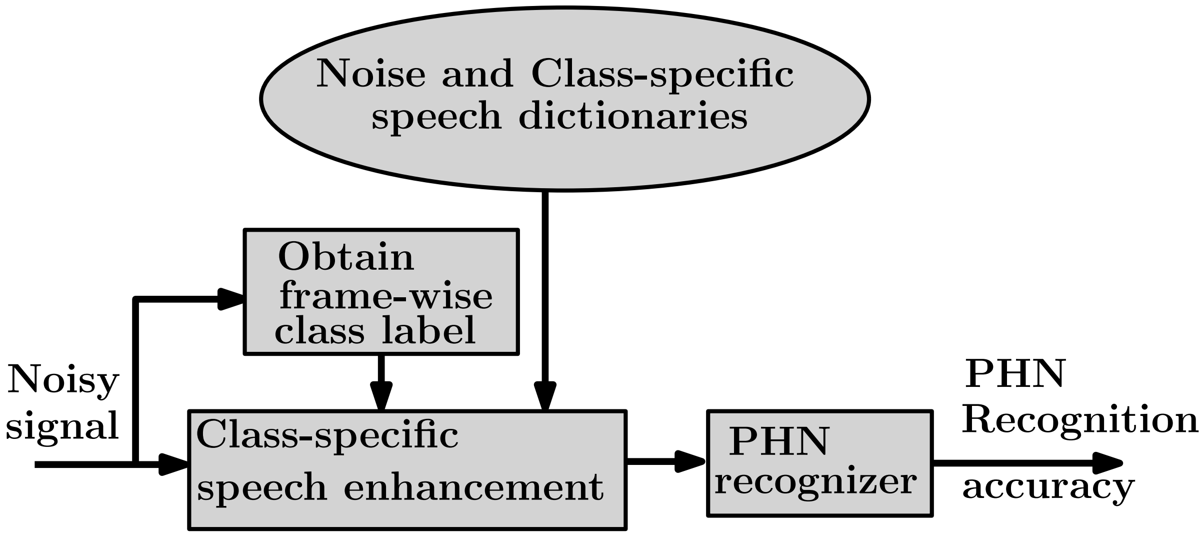


Figure 3.1: Class-specific enhancement framework

We propose a joint enhancement-decoding (JED) formulation to compensate for this error and to improve the phoneme recognition accuracy. The block diagram of class-specific enhancement combined with the proposed JED algorithm is shown in Fig. 3.2. We use multiple class-specific dictionaries for enhancing a single frame and these different denoised versions of a frame are fed into the JED algorithm. The algorithm accepts multiple enhanced observations for each frame and selects the best observation for each frame as well as the best state

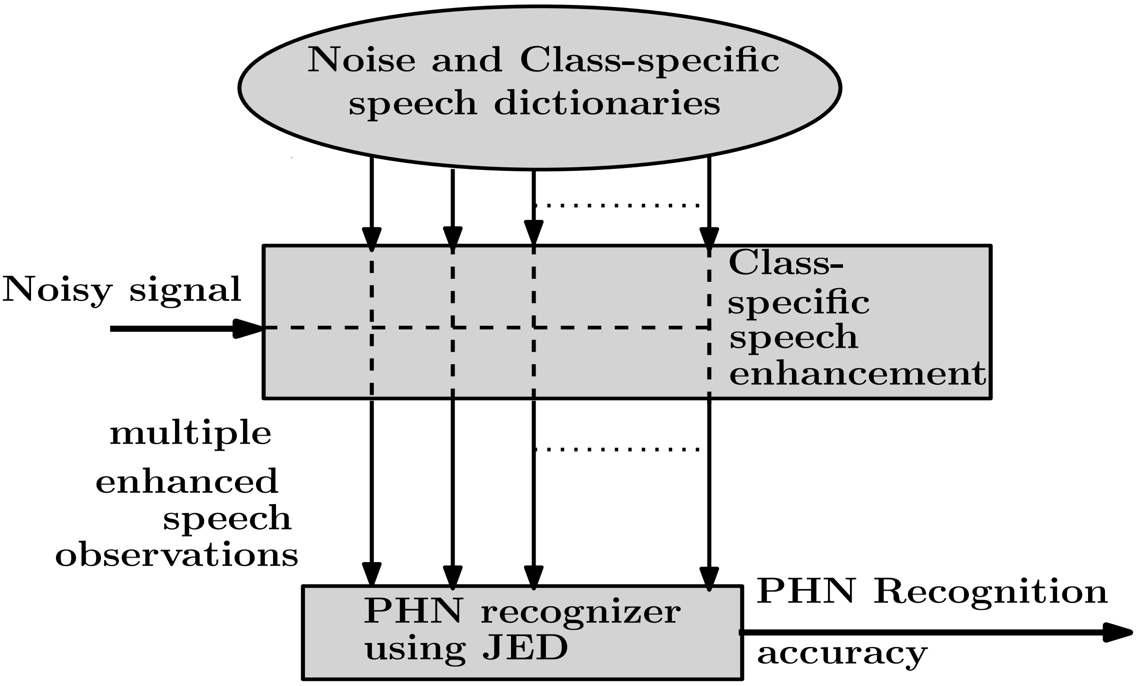


Figure 3.2: Class-specific enhancement framework using JED

sequence that maximizes the likelihood of the chosen observations. We use sparse coding based dictionary learning approach for obtaining the enhanced speech observations. This approach involves learning of speech and noise dictionaries as well as a sparse coding stage for learning the coefficients. The sparse coding and dictionary learning parts are explained in detail in Chapter 2.

3.2.1 Speech enhancement using dictionary learning

For this work we use KSVD based dictionary learning as mentioned in Chapter 2 [85]. Batch least angle regression with coherence criterion (LARC) [59] is used for sparse coding for which a residual coherence threshold is applied as a stopping criterion.

Let $s(m)$ and $x(m)$ be the m^{th} samples of the clean speech and noise signal corrupting the speech. Considering additive noise model, the m^{th} sample of the noisy speech, $y(m)$ is given as;

$$y(m) = s(m) + x(m). \tag{3.1}$$

The short time Fourier transform (STFT) of the above is given as;

$$Y(\omega_k) = S(\omega_k) + X(\omega_k) \tag{3.2}$$

Here $\omega_k = \frac{2\pi k}{R}$, $k = 0, 1, 2...R-1$, R is the number of frequency bins and k is the frequency index.

For learning the dictionary, we consider only the magnitude spectra. The phase of the noisy speech is retained to be used for reconstruction of the estimated speech signal.

Considering only the magnitude spectra, we can write,

$$Y \approx S + X \in \mathbb{R}^{R\times 1} \tag{3.3}$$

where S and X represent the magnitude spectra of the clean speech and the noise, respectively.

Using the learned overcomplete dictionaries D_s and $D_x \in \mathbb{R}^{R\times L}$, $L > R$, learned from the speech and noise training set and their corresponding sparse coefficient vectors c_s and c_x, an estimate of the magnitude STFT of noisy speech for a frame f is given as,

$$\hat{Y}_f = D_s \times c_s + D_x \times c_x \tag{3.4}$$

The enhanced speech is estimated as;

$$\hat{S}_f = D_s \times c_s \tag{3.5}$$

3.2.2 Viterbi algorithm

The Viterbi algorithm is a dynamical programming algorithm that is used to find the most likely sequence of labels or hidden states, given a set of observations $\Theta = \{\theta_f; 1 \leq f \leq F\}$, where F denotes the total number of frames, in hidden Markov model (HMM) based recognition systems. The algorithm is given in Algorithm 4.

3.2.3 JED algorithm

The JED algorithm accepts multiple enhanced observations for each frame and finds a path that maximizes the likelihood of the chosen observations. Thus we could use multiple class-specific dictionaries for enhancing a single frame and these frames can be fed into the phoneme recognizer using JED to obtain the maximum likelihood path.

Let the enhanced observation at the f^{th} frame using class-specific dictionary with i^{th} label be denoted by θ_f^i. If F denotes the total number of frames and N denotes the number of best labels considered for enhancement in each frame, $\Theta = \{\theta_f^i; 1 \leq f \leq F, 1 \leq i \leq N\}$. The JED algorithm optimizes the observation sequence $\theta_f^{i^*(f)}, 1 \leq f \leq F$ as well as the state sequence

Algorithm 4: Viterbi Algorithm

$\Theta = \{\theta_f; 1 \leq f \leq F\}$: observation sequence
$ob(\cdot|q_j)$:observation probability given state q_j
$tr(q_k \rightarrow q_j)$:transition probability from q_k to q_j
for *each state* q! =*Starting state* **do**
  $\phi(1, q) = 1$
for $f \leftarrow$ *1 to* F **do**
  for *each state* q_j **do**
    $\phi(f, q_j) \leftarrow \max_k \phi(f-1, q_k) ob(\theta_f|q_j) tr(q_k \rightarrow q_j)$
    $\Psi(f, q_j) = \text{argmax}_k\, \phi(f-1, q_k) ob(\theta_f|q_j) tr(q_k \rightarrow q_j)$
$P(\Theta, \mathbf{Q}) = \max_j \phi(F, q_j)$
Backtrack

$q_1^*, q_2^*, \ldots, q_F^*$ to maximize the likelihood of the observation as follows :

$$\{\theta_f^{i^*(f)}, q_f^*, 1 \leq f \leq F\} = \underset{\theta_f^i, q_f}{\text{argmax}}\, P(q_1, q_2, \ldots q_F|\Theta) \tag{3.6}$$

Assuming each enhanced observations per frame to be equally likely;

$$\begin{aligned}\{\theta_f^{i^*(f)}, q_f^*, 1 \leq f \leq F\} &= \underset{\theta_f^i, q_f}{\text{argmax}}\, P(\Theta|q_1, q_2, \ldots q_F) P(q_1, q_2, \ldots q_F) \\ &= \underset{q_f}{\text{argmax}} \left\{ \max_{\theta_f^i} P(\Theta|q_1, q_2, \ldots q_F) \right\} P(q_1, q_2, \ldots q_F)\end{aligned} \tag{3.7}$$

Assuming independence among observations, given the state sequence, we write

$$\begin{aligned}\{q_1^*, q_2^*, \ldots q_F^*\} &= \underset{q_f}{\text{argmax}} \left\{ \prod_{f=1}^{F} \max_{1 \leq i \leq N} P(\theta_f^i|q_f) \right\} P(q_1, q_2, \ldots q_F) \\ &= \underset{q_f}{\text{argmax}} \left\{ \prod_{f=1}^{F} P(\theta_f^{i^*(f)}|q_f) \right\} P(q_1, q_2, \ldots q_F)\end{aligned} \tag{3.8}$$

where $i^*(f) = \text{argmax}_{1 \leq i \leq N}\, P(\theta_f^i|q_f)$.

Thus the JED algorithm could be considered as the modified version of Viterbi decoding algorithm, which incorporates N observations instead of a single observation per time instant.

The steps of the algorithm are given in Algorithm 5.

Algorithm 5: Joint Enhancement-Decoding Algorithm

$\Theta = \{\theta_f^i; 1 \leq f \leq F, 1 \leq i \leq N\}$: observation sequence
$ob(\cdot|q_j)$:observation probability given state q_j
$tr(q_k \rightarrow q_j)$:transition probability from q_k to q_j
for *each state* q ! =*Starting state* **do**
  $\phi(1, q) = 1$
for $f \leftarrow$ *1 to* F **do**
  for *each state* q_j **do**
    $\theta_f^{i^*(f)} = \text{argmax}_{\theta_f^i}\, ob(\theta_f^i|q_j)$
  for *each state* q_j **do**
    $\phi(f, q_j) \leftarrow \max_k \phi(f-1, q_k) ob(\theta_f^{i^*(f)}|q_j) tr(q_k \rightarrow q_j)$
    $\Psi(f, q_j) = \text{argmax}_k\, \phi(f-1, q_k) ob(\theta_f^{i^*(f)}|q_j) tr(q_k \rightarrow q_j)$
$P(\Theta, \mathbf{Q}) = \max_j \phi(F, q_j)$
Backtrack

3.2.4 Best-N class-specific dictionary based enhancement-recognition using JED

JED algorithm does not require the knowledge of frame labels to do the class-specific enhancement and decoding. In fact, all the phoneme dictionaries can be used for the enhancement of each frame and these multiple enhanced observations can be given as the input. The algorithm then jointly optimizes the class label in each frame and the decoding path. However, we observed that the use of all the phoneme dictionaries for frame-wise enhancement is not only computationally expensive, but also results in a performance drop. Hence we choose a subset of labels such that the chance of actual label belonging to this subset is high.

We use the confusion matrix obtained by running the recognizer on a subset of the clean TIMIT test sentences to choose this subset of labels. Selecting a small set of labels with high likelihood from the matrix for a given recognized label, ensures that the likelihood of the actual label being in this set is high. Hence we use this set of labels in class specific enhancement for enhancing each frame of a noisy test speech.

The block diagram summarizing the steps of the best-N class-specific dictionary based enhancement for phoneme recognition using the proposed JED algorithm is shown in Fig. 3.3. At first, noisy speech is enhanced using a class-independent dictionary and the phoneme label of each frame is estimated by recognizing this enhanced speech. The confusion matrix of ASR output for clean speech is used to obtain the next $N-1$ best labels. These labels are then used to obtain N enhanced observations for each frame using the respective dictionaries. The enhanced observations are then fed into the JED algorithm, which gives the decoded output by maximizing the likelihood.

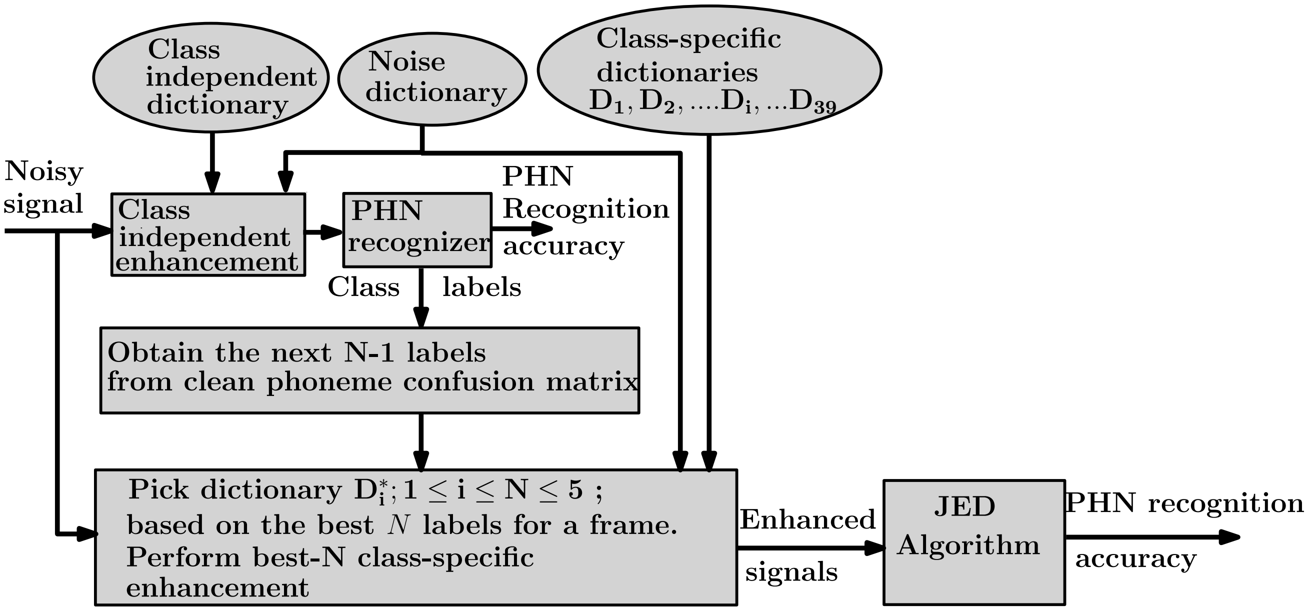


Figure 3.3: Phoneme recognition of noisy speech using best-N class-specific dictionaries using JED

The enhancement and recognition stages are explained in Algorithm 6.

Algorithm 6

1. *Enhance noisy data using class-independent dictionary:*

   Let $Y_f \in \mathbb{R}^{R\times 1}$ *be the noisy speech spectrum of a frame.* $D_{ind} \in \mathbb{R}^{R\times L}$ *and* $D_x \in \mathbb{R}^{R\times L}$ *be the dictionaries for class-independent speech and the noise, respectively. Using the composite dictionary* $D = [D_{ind}\ D_x]$ *, the sparse coefficients of the noisy speech are obtained as*

$$\begin{bmatrix} c_s^{ind} \\ c_x \end{bmatrix} = LARC(Y_f, D, \mu_{coh}) \tag{3.9}$$

   where μ_{coh} *is the threshold on mutual coherence and* c_s^{ind} *represents the sparse coefficient vector corresponding to* D_{ind}.

   Clean speech is estimated as

$$\hat{S}_f = D_{ind} \times c_s^{ind} \tag{3.10}$$

2. *Find the phoneme labels using a phoneme recognizer on this enhanced speech.*

3. *Based on the class label of a frame, obtain the next best $N-1$ labels using the confusion matrix of the recognition performance on a small subset of clean speech data.*

4. *Using the obtained N labels, perform a best-N-label class-specific enhancement of the original noisy data using the dictionaries corresponding to these class labels.*

   Let the N-best dictionaries corresponding to the obtained class label be D_i^*; $1 \leq i \leq N$. Enhance the original noisy speech observation Y_f separately using each of these N-best dictionaries. The sparse coefficients obtained using composite dictionary $D_i = [D_i^* \ D_x]$ are

$$\begin{bmatrix} c_s^{*(i)} \\ c_x^{*(i)} \end{bmatrix} = LARC(Y_f, D_i, \mu_{coh}) \tag{3.11}$$

   The clean speech estimate is given as;

$$\hat{S}_f^{*(i)} = D_i^* \times c_s^{*(i)} \tag{3.12}$$

   where $c_s^{*(i)}$ corresponds to D_i^*

5. *Input these N enhanced speech estimates for each frame to the JED algorithm and evaluate the recognition performance.*

3.3 Experiments and results

Experimental setup is similar to the one explained in chapter 2 sec. 2.3.1, except for the recognition setup which is modified slightly for the computation of the confusion matrix. The recognition setup is given below;

3.3.0.1 Recognition setup

The ASR is trained on the entire TIMIT clean training data. The TIMIT test set is randomly divided into two equal sets. One of them is used to obtain the clean speech confusion matrix after recognition. The second test set is used to compare the recognition accuracy. The results are reported on the reduced set of 39 phonemes. The source code of the Viterbi decoding algorithm for recognition in the HTK toolkit [97] is modified to implement our JED algorithm. We use 39-dimensional mel frequency cepstral coefficients [98] for recognition with 0-th coefficient, delta and delta-delta coefficients. Cepstral mean normalization is applied. The analysis frame is chosen to be 30 ms with 10 ms frame shift. For recognition, we use a three state monophone HMM model with diagonal covariance matrix. For each state, the number of Gaussian mixtures is set to 32, since increasing it further did not result in any significant improvement in recognition performance. A bigram phoneme language model is used.

3.3.1 Best-N class-specific dictionary based enhancement-recognition scheme using JED with monogram confusion matrix

Section 3.2.4 explains the steps involved in the proposed method. As explained, after we obtain the label of a frame, the next best $N-1$ labels are obtained using a confusion matrix obtained by recognizing a subset of the test data. In this section, we show the results of the case when the considered confusion matrix is mono-gram. Each entry in the matrix can be interpreted as the likelihood of row label being the actual label given that column label is the obtained label.

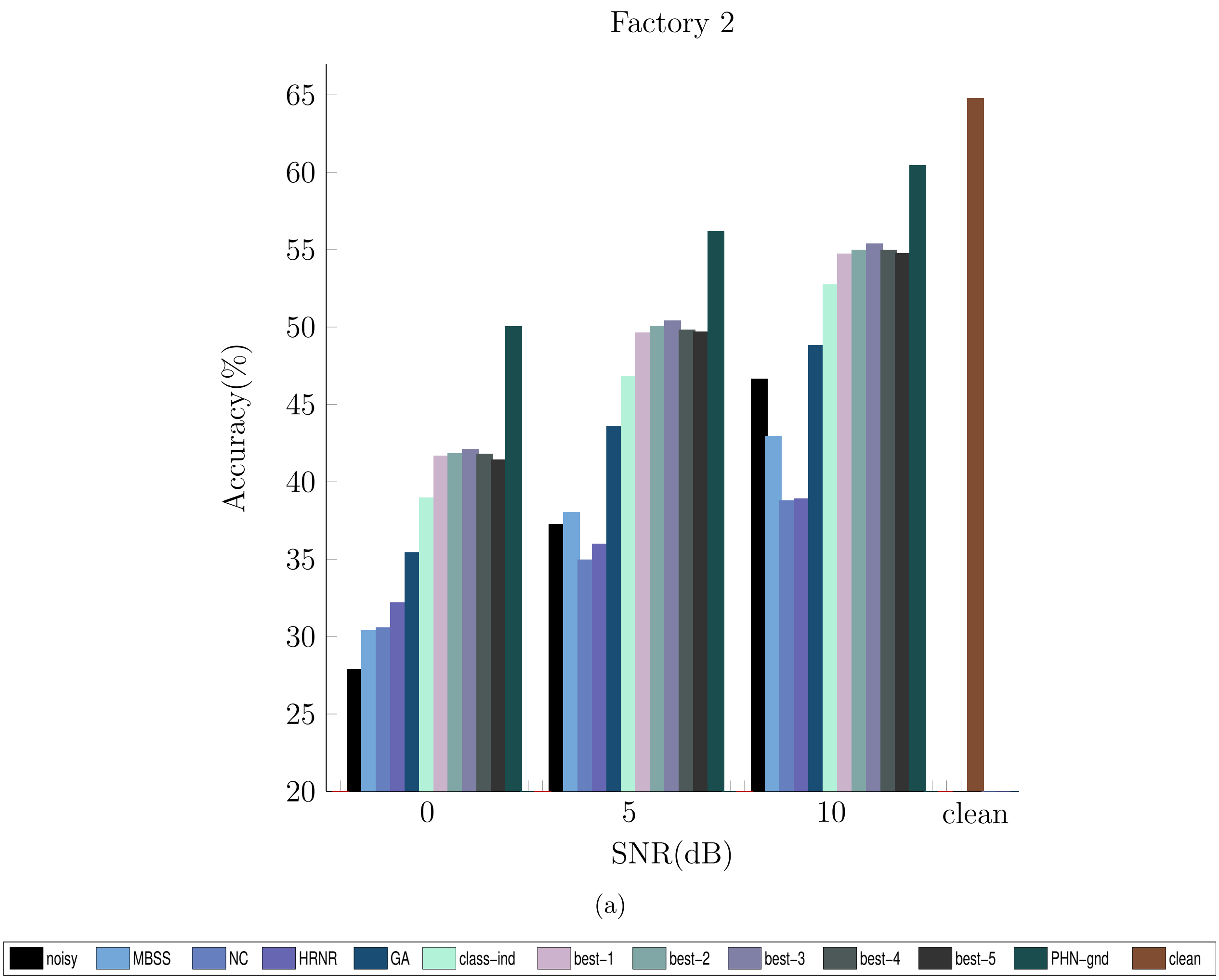
Figures 3.4 (a-e) show the improvements in the phoneme recognition accuracies for the proposed best-N class-specific enhancement-recognition scheme using JED with mono-gram confusion matrix for N varying from 1 through 5. The phoneme recognition accuracies for speech corrupted with noises (a) factory 2, (b) m109, (c) leopard, (d) babble and (e) volvo are shown. We compare the recognition accuracies of the proposed method with that of class-independent enhancement scheme and also with four other enhancement schemes: multi-band spectral subtraction (MBSS) [18], non-causal apriori SNR estimator (NC) [38], harmonic regeneration noise reduction (HRNR) [39] and geometric spectral subtraction (GA) [19].

It can be observed from figure 3.4 that the best-N enhancement scheme yields performance superior to all the other schemes for all noise types.

For speech corrupted with factory2 noise, best-N enhancement scheme gives an average relative accuracy improvement (RAI) of 5.6%, 6.2%, 6.9%, 6.0% and 5.4%, respectively, for values of $N = 1$ to 5, over class-independent enhancement scheme, when averaged over SNRs 0, 5, and 10 dB. For M109 noise, the average RAI values are 3.9%, 4.3%, 5.4%, 4.9% and 4.8%, respectively. The average RAI values for speech corrupted with leopard noise are 2.3%, 3.6%, 3.9%, 3.2% and 3.2%.

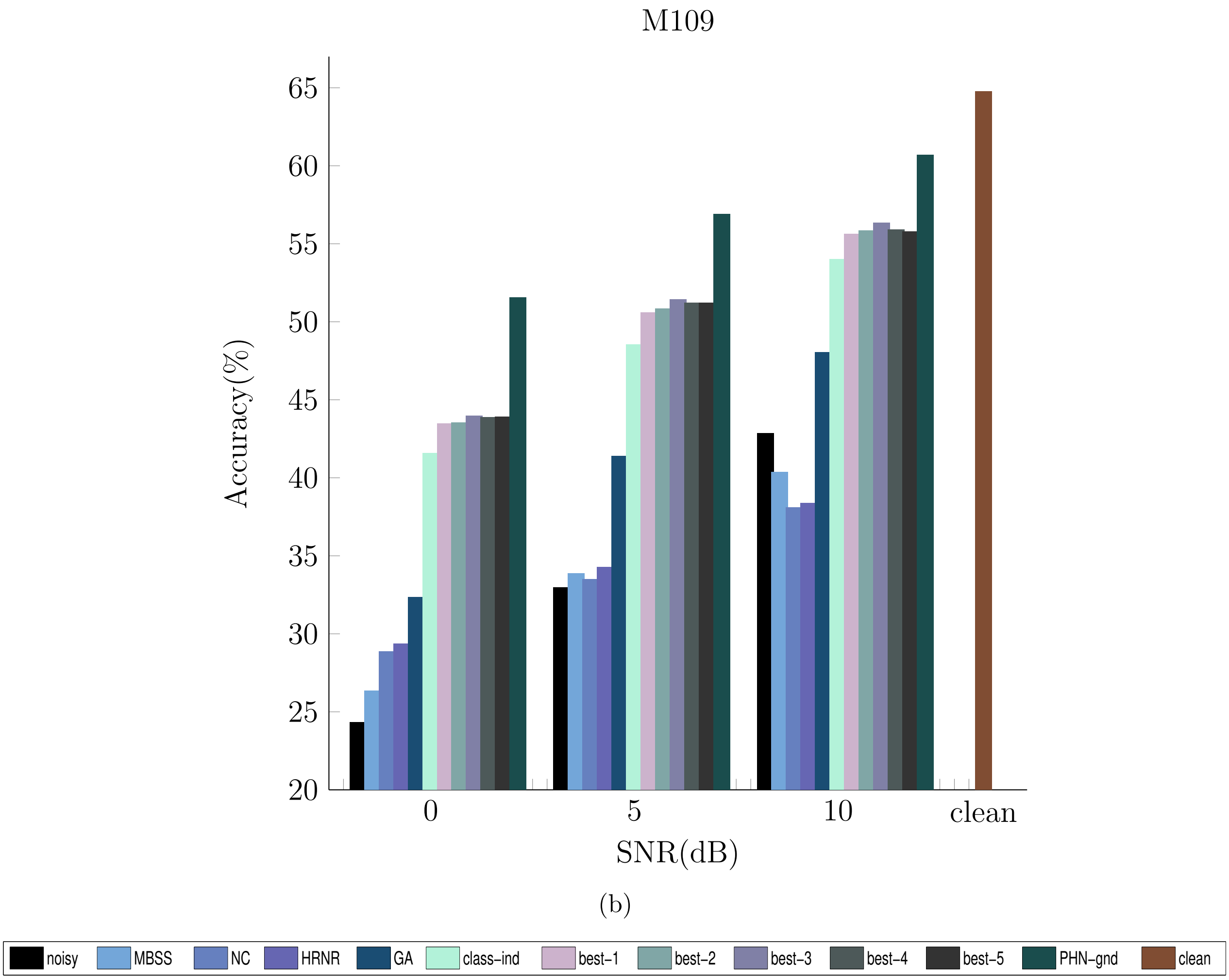
In the case of speech corrupted with babble noise, the proposed scheme gives superior performance only when $N = 1$. The average RAI values over class-independent scheme are 2.3%, -1.0%, 0.8%, -1.1% and -1.3% for values of $N = 1$ to 5.

For the case of volvo noise, it is observed that after cepstral mean normalization, the recognition accuracy using noisy speech outperforms the class-independent and class-dependent



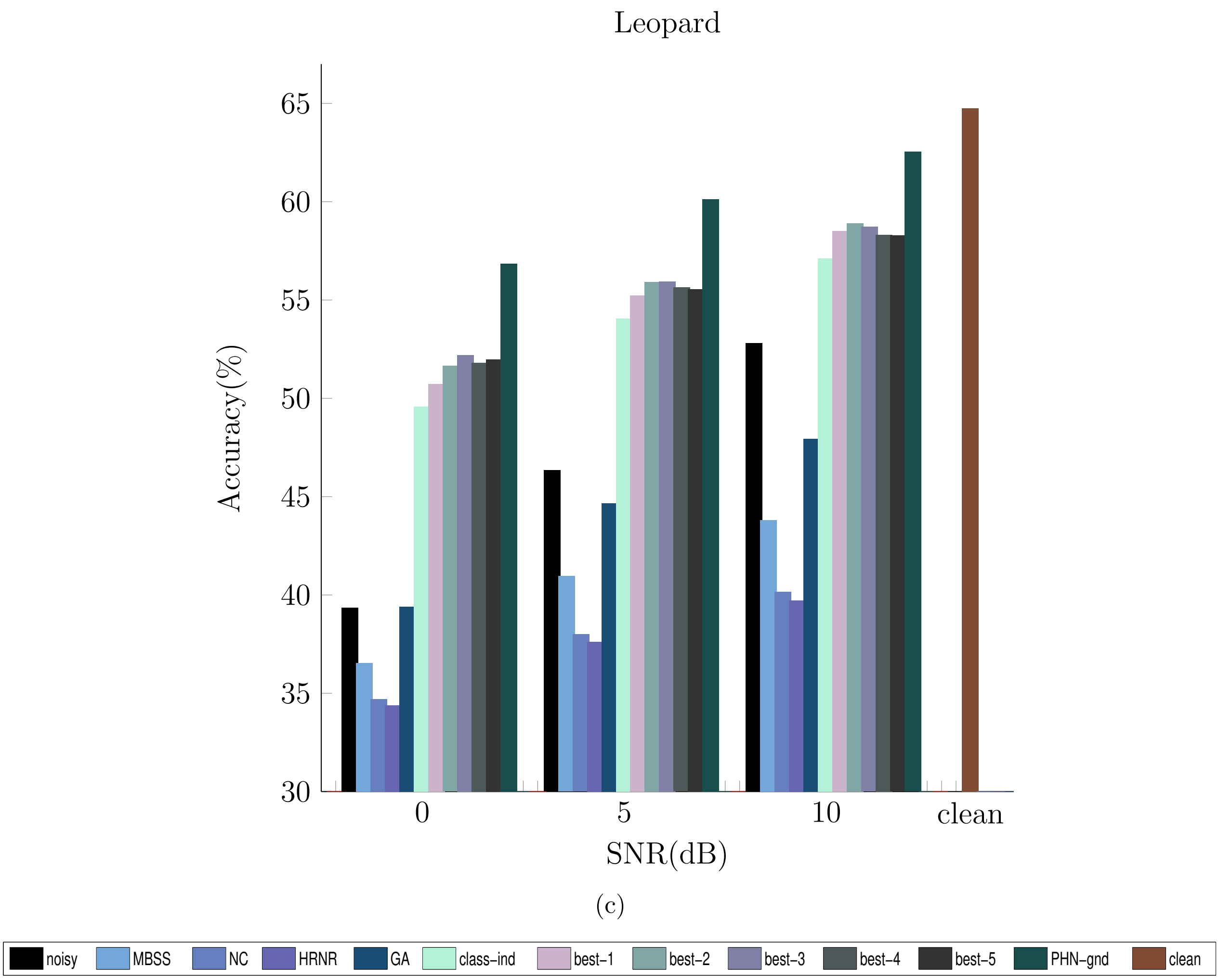
(a)

Figure 3.4: Performance of JED in terms of phoneme recognition accuracies for (a) Factory2 noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with mono-gram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.



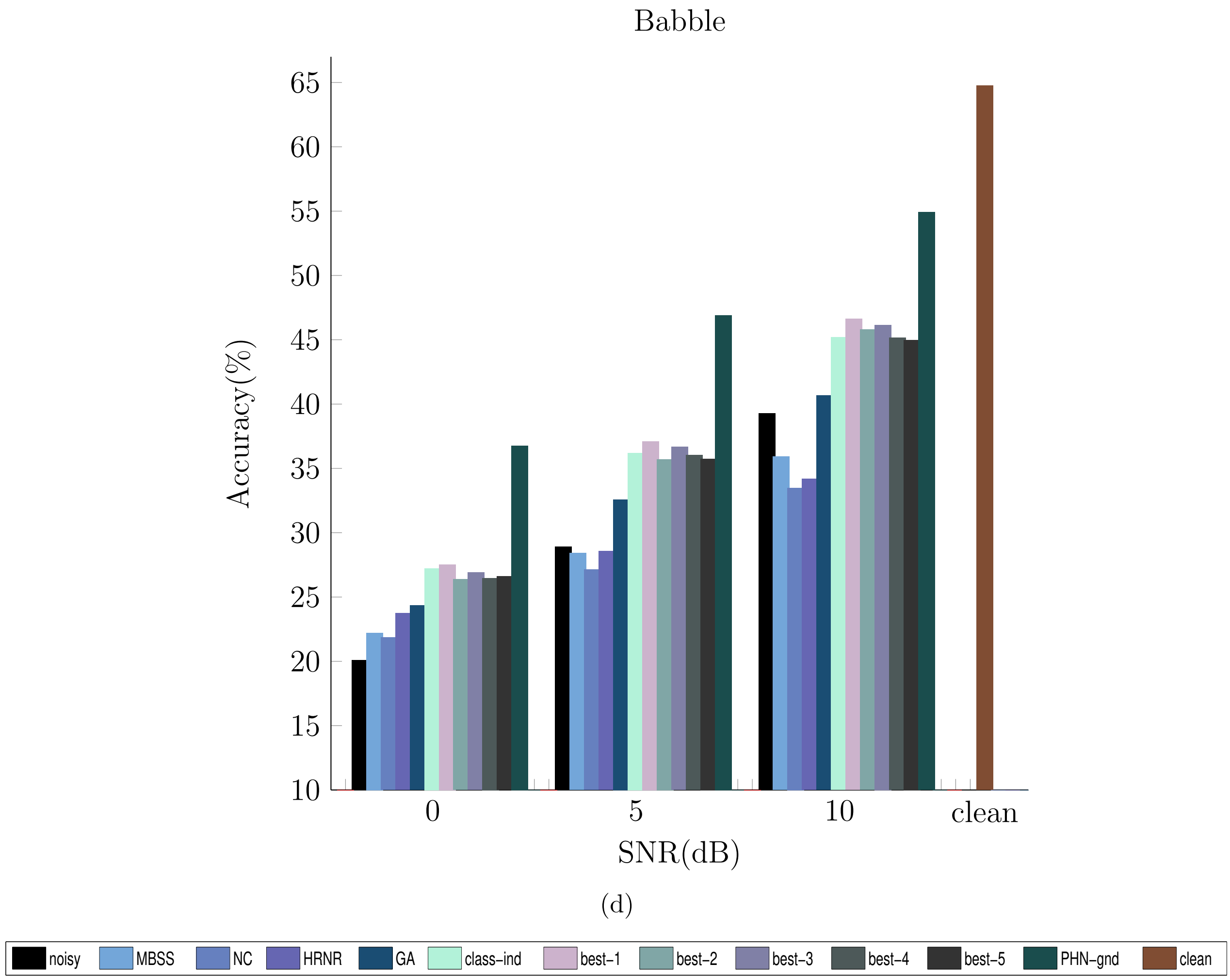
(b)

Figure 3.4: Performance of JED in terms of phoneme recognition accuracies for (b) M109 noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with mono-gram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.



(c)

Figure 3.4: Performance of JED in terms of phoneme recognition accuracies for (c) Leopard noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with mono-gram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.



(d)

Figure 3.4: Performance of JED in terms of phoneme recognition accuracies for (d) Babble noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with mono-gram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.

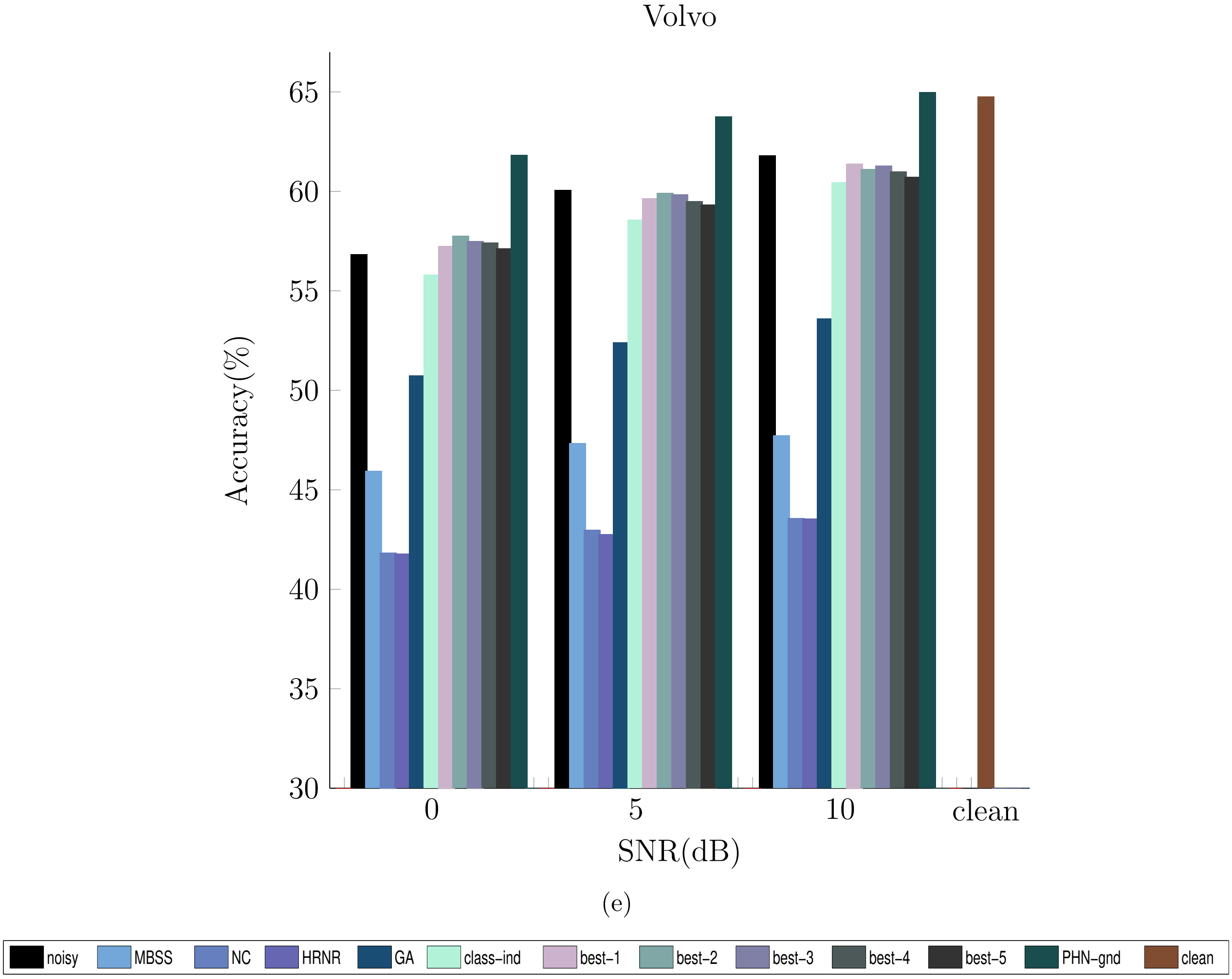


Figure 3.4: Performance of JED in terms of phoneme recognition accuracies for (e) Volvo noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with mono-gram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.

schemes in most cases. Thus the proposed scheme shows average RAI values of -0.2%, 0.1%, -0.02%, -0.4% and -0.8% respectively, for N varying from 1 to 5, over the performance of noisy speech. However, it is to be noted that the accuracies from the proposed scheme are still better than those of the class independent scheme. For phoneme recognition, the average RAIs over class-independent scheme are 2.0%, 2.3%, 2.2%, 1.8% and 1.4%, respectively, for N varying from 1 to 5.

From figure 3.4 it is observed that the recognition performance varies as N varies from 1 to 5. The best performance is obtained with two or three dictionaries in most of the cases. The motivation for using multiple enhanced observations based on best-N class-specific dictionaries is the fact that the set of labels employed for enhancing a frame has more chance of having the correct label when N=5 than when N=1.

Table 3.1: Percentage of frames for which none of the estimated N labels include the ground truth labels. The three columns for each noise correspond to N=1, N=3 and N=5. When N=5 on the average, the correct label percentage increases by about 20%

SNR	Factory2			M109			Leopard			Babble			Volvo		
(dB)	$N=1$	$N=3$	$N=5$	$N=1$	$N=3$	$N=5$	$N=1$	$N=3$	$N=5$	$N=1$	$N=3$	$N=5$	$N=1$	$N=3$	$N=5$
0	52	38	30	49	36	28	42	29	24	69	53	43	37	25	20
5	45	31	25	43	30	23	39	26	21	57	42	33	35	24	19
10	40	27	21	38	26	21	37	25	20	47	34	27	34	23	18

This is illustrated in Table 3.1, where we show the percentage of frames in the entire test set where the estimated labels do not include the ground truth class label for N=1, N=3 and N=5 for different noise and SNR conditions. It is clear that the percentage of such frames reduces when N=5 compared to when N=1.

The JED algorithm accepts multiple inputs per time instant and maximizes the overall likelihood of the output utterance. To evaluate this, we obtain the log likelihood values of a few utterances from the test set for the best-N class-specific schemes for factory2 noise at 0 dB SNR as shown in Table 3.2.The likelihood increases monotonically from N = 1 to 5. However, as observed from figure 3.4 (a-e), this does not always translate to a monotonic increase in recognition accuracy.

Table 3.2: Log likelihood values of a few utterances from TIMIT test set for best-N class-dependent schemes (best-N) for N varying from 1 to 5 for factory2 noise at 0 dB SNR.

	mnjm0/sx410	fpas0/sx404	mtaa0/sx115	fcal1/sx143
best-1	-21820	-21494	-23795	-19125
best-2	-20853	-20885	-22755	-18644
best-3	-20299	-20380	-22139	-18187
best-4	-20078	-20040	-21985	-18121
best-5	-19909	-19781	-21876	-18074

3.3.2 Best-N class-specific dictionary based enhancement-recognition scheme using JED with bigram confusion matrix

Instead of using a mono-gram confusion matrix to select the best-N labels to enhance a particular frame as described in section 3.3.1, we can consider a bigram or trigram confusion matrix to explore the effect of dependence of phonemes. The block diagram of the scheme is shown in figure 3.5.

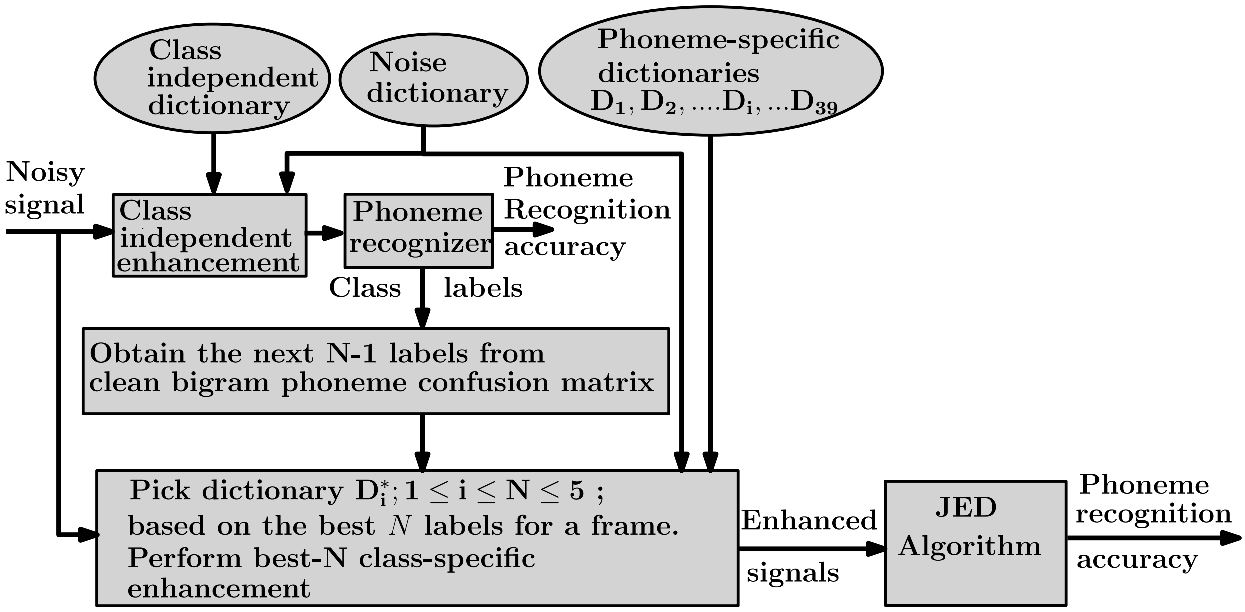


Figure 3.5: Best-N class-specific dictionary based enhancement-recognition scheme using JED with bigram confusion matrix

As shown in figure 3.5, all the steps involved are the same as those explained in section 3.2.4 except the confusion matrix used. In the bigram case, the best-N phonemes for each frame are selected based on a bigram confusion matrix. This matrix is populated by computing the occurrence of each phoneme for a given combination of estimated phonemes at the current and previous time instants.

3.3.2.1 Results and discussion

Figure 3.6 (a-e) shows the improvements in the phoneme recognition accuracies for the best-N class-specific enhancement-recognition scheme using JED with bigram confusion matrix for N varying from 1 through 5. The phoneme recognition accuracies for speech corrupted with noises (a) factory 2, (b) m109, (c) leopard, (d) babble and (e) volvo are shown. The figure also shows the recognition accuracies for the other five enhancement schemes class-independent, MBSS [18], NC [38], HRNR [39] and GA [19] for comparison.

The average RAI values for this scheme using bigram confusion matrix over class-independent scheme for N varying from 2 to 5 in the case of speech corrupted with factory 2 noise are, 6.2%, 6.6%, 6.4% and 6.2%. For m109 noise, the values are 4.6%, 4.7%, 4.9% and 4.6%. The values for speech corrupted with leopard noise are 3.5%, 4.1%, 4.3% and 4.0%.

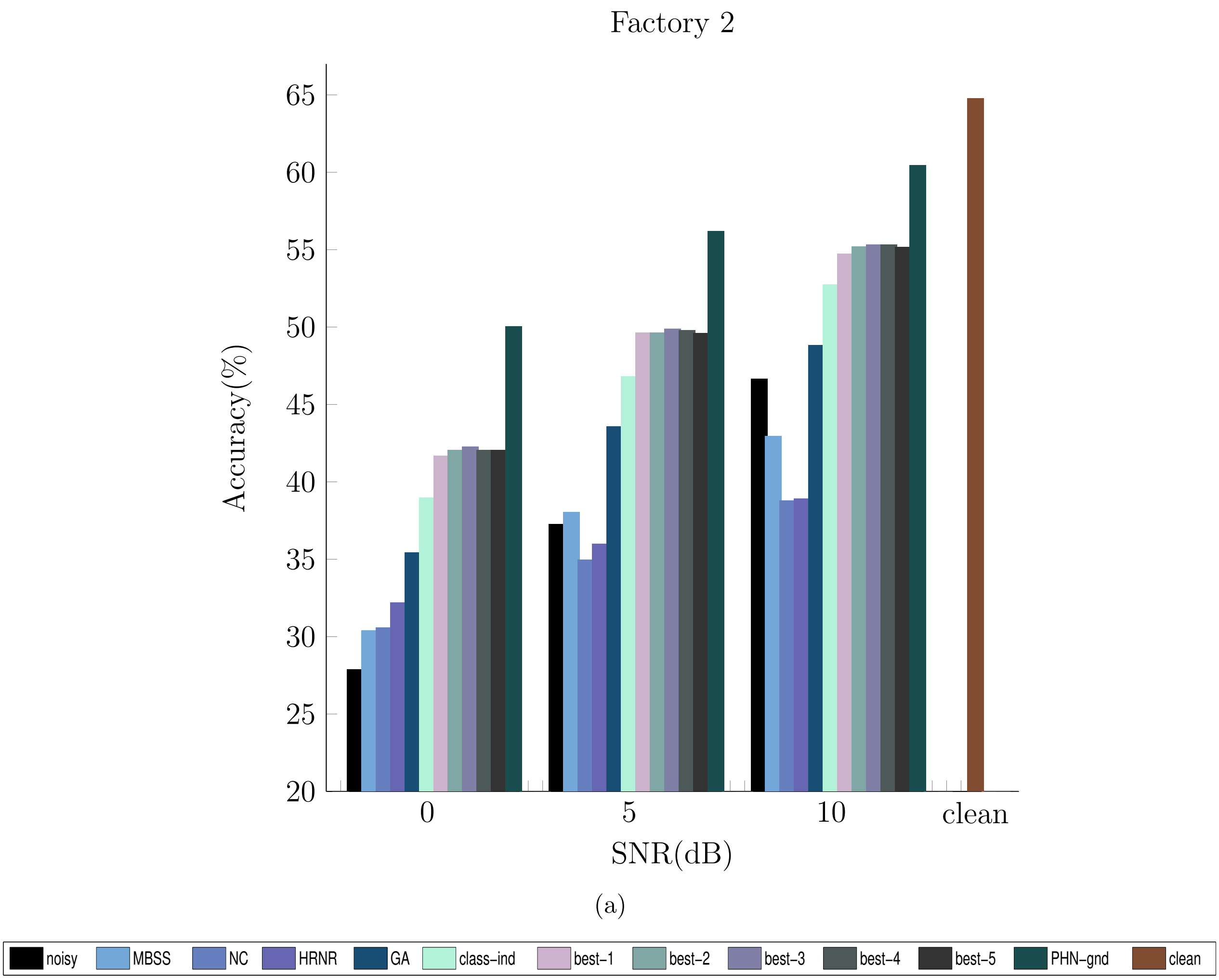
For babble noise, just like the previous cases of using mono-gram confusion matrix, the improvement is not much for $N = 2$ to 5. The average RAI values in this case are 0.2%, 1.2%, 0.6% and 0.9% .

In the case of volvo noise, just like the case of using monogram matrix, the recognition accuracy using noisy speech outperforms the class-independent and class-dependent schemes in most cases after CMN. The average RAI values of the proposed scheme over the performance for noisy speech for $N = 2$ to 5 are 0.4%, 0.3%, -0.3% and -0.4%. But just like the previous scheme, here also we get better accuracies than those of the class independent scheme. For phoneme recognition, the average RAIs over class-independent scheme are 2.7%, 2.5%, 1.8% and 1.8% respectively, for N varying from 2 to 5.

3.3.3 Best-N class-specific dictionary based enhancement-recognition scheme using JED with trigram confusion matrix

In this scheme, we use a trigram confusion matrix to select the best-N labels to enhance a particular frame. The block diagram of the scheme is shown in figure 3.7.

In the trigram case, the best-N phonemes for each frame are selected based on a trigram confusion matrix. The trigram confusion matrix uses the current, previous and next frames for computing the frequency of occurrence of a phoneme.



(a)

Figure 3.6: Performance of JED in terms of phoneme recognition accuracies for (a) Factory2 noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme, and best-N class-specific enhancement (best-N) schemes with bigram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.

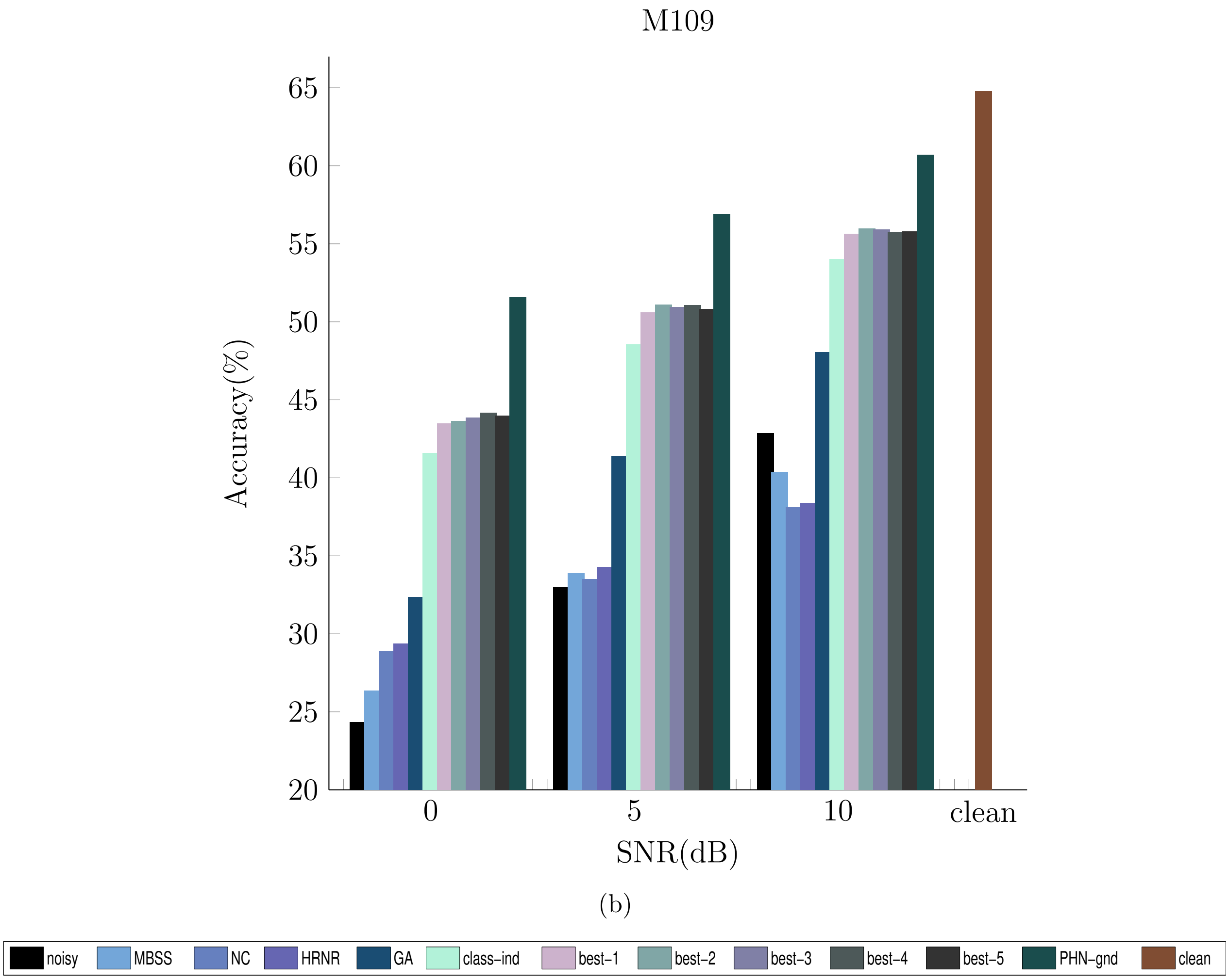
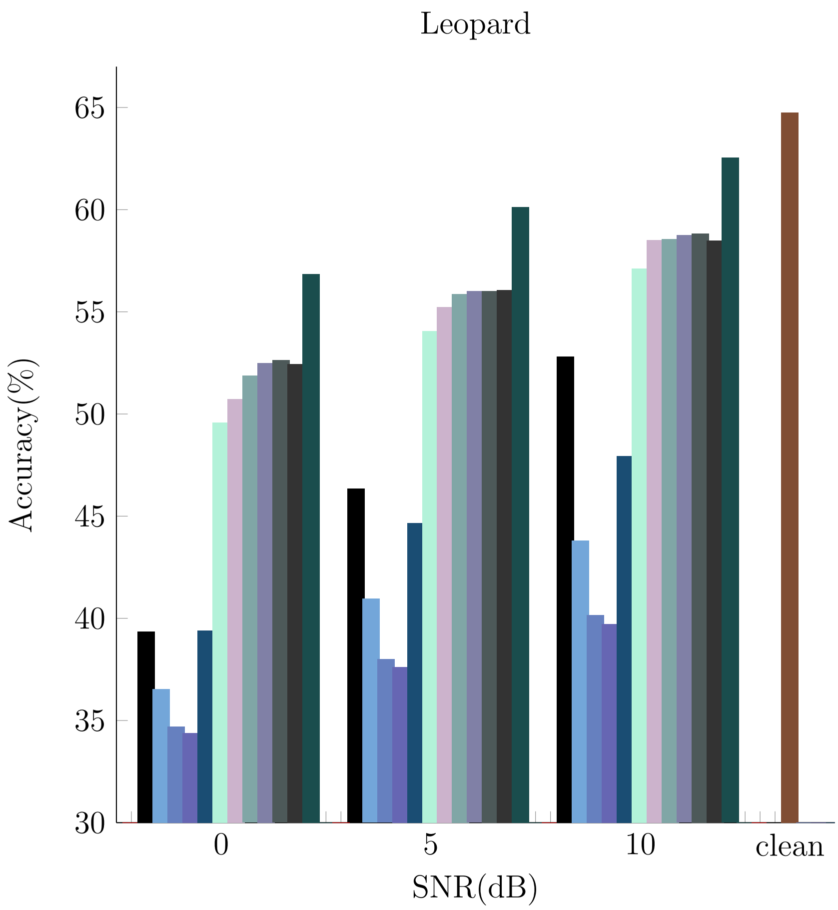


Figure 3.6: Performance of JED in terms of phoneme recognition accuracies for (b) M109 noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with bigram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.



(c)

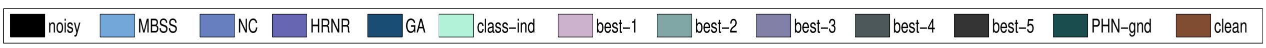


Figure 3.6: Performance of JED in terms of phoneme recognition accuracies for (c) Leopard noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with bigram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.

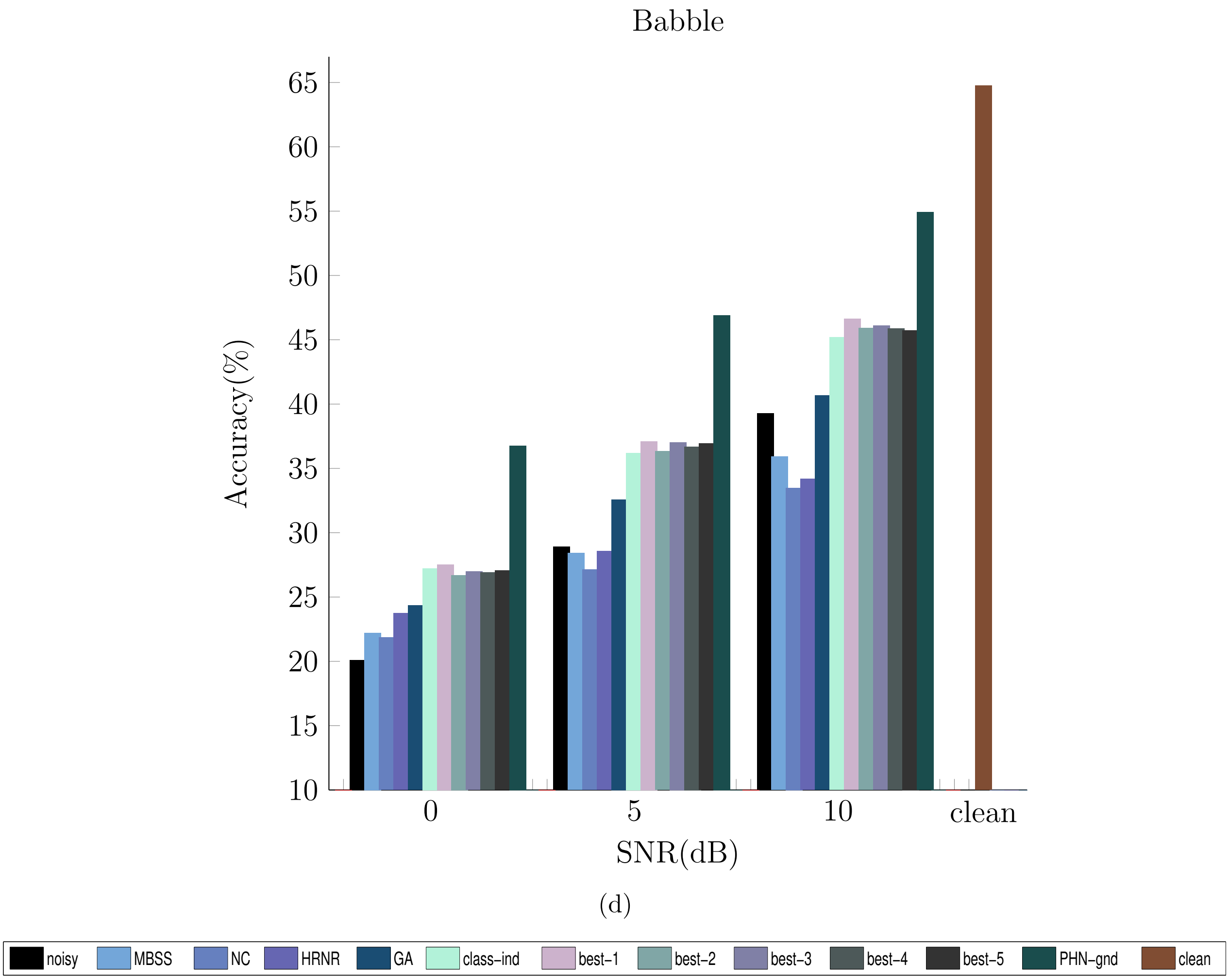
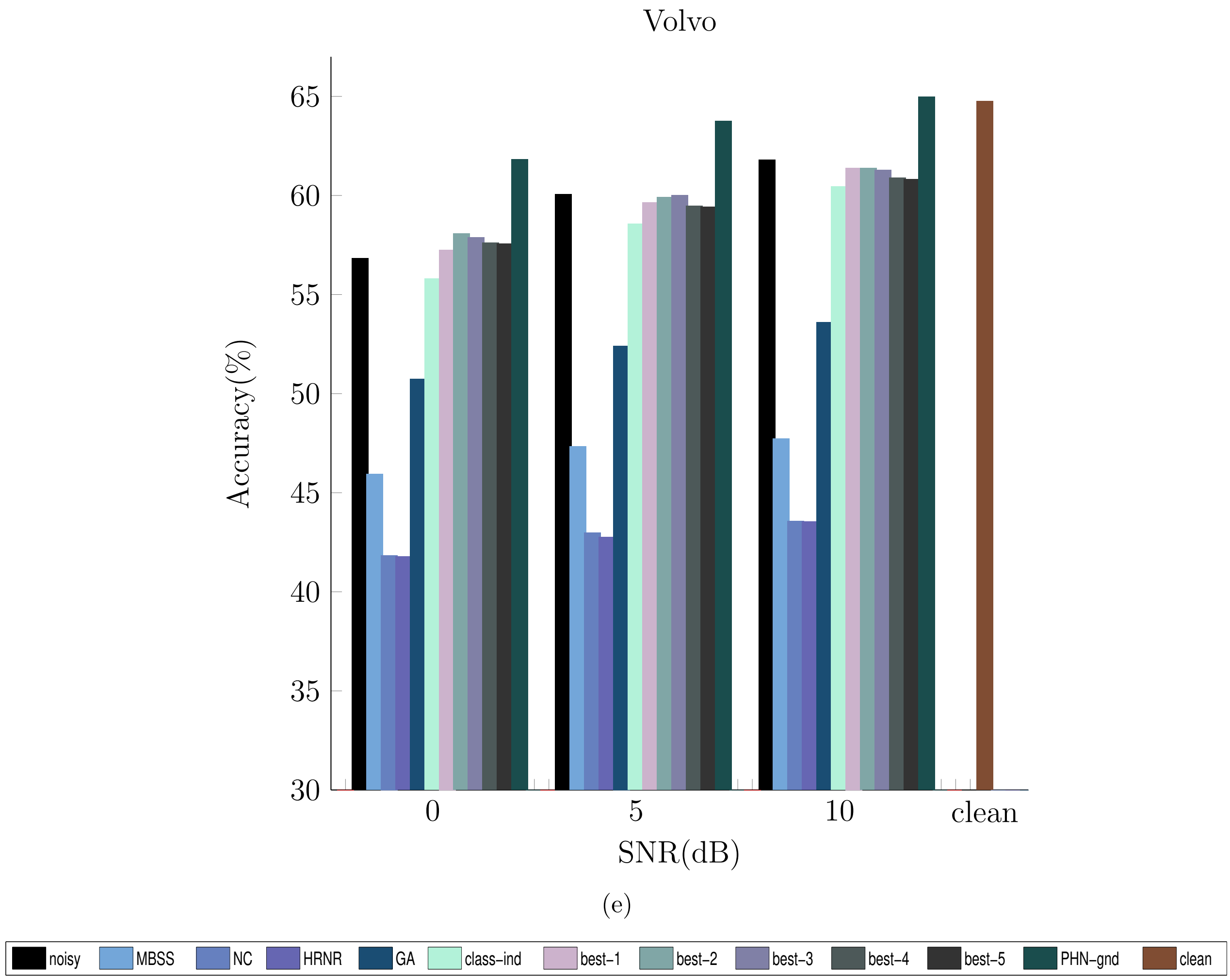


Figure 3.6: Performance of JED in terms of phoneme recognition accuracies for (d) Babble noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with bigram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.



(e)

Figure 3.6: Performance of JED in terms of phoneme recognition accuracies for (e) Volvo noise. In each figure, results are given for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with bigram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.

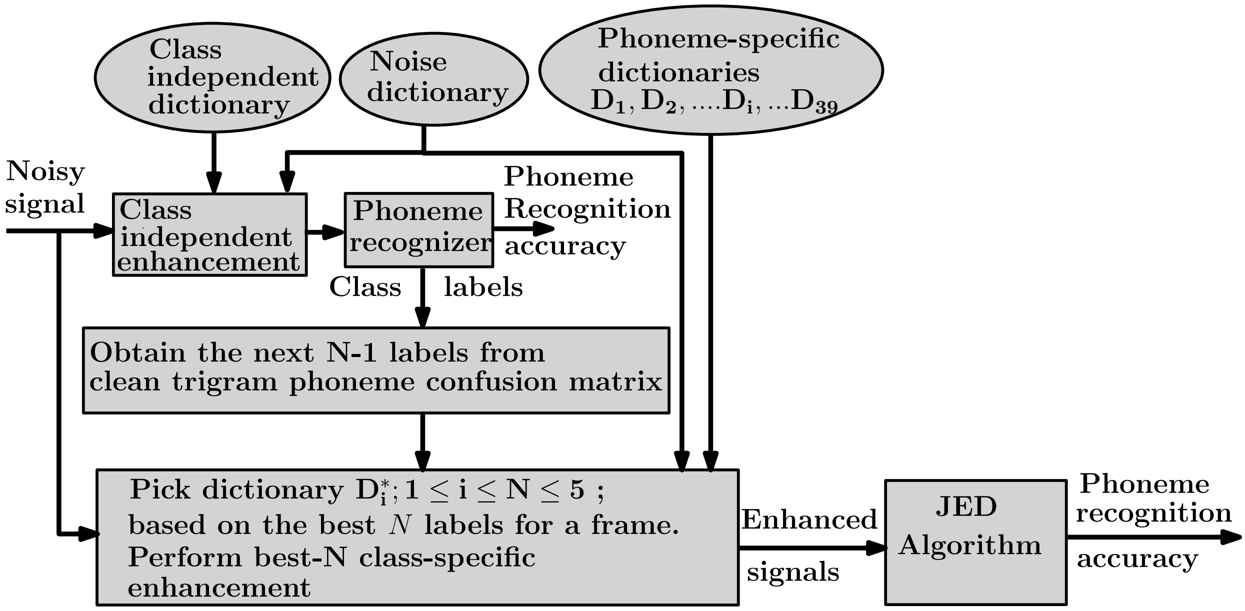
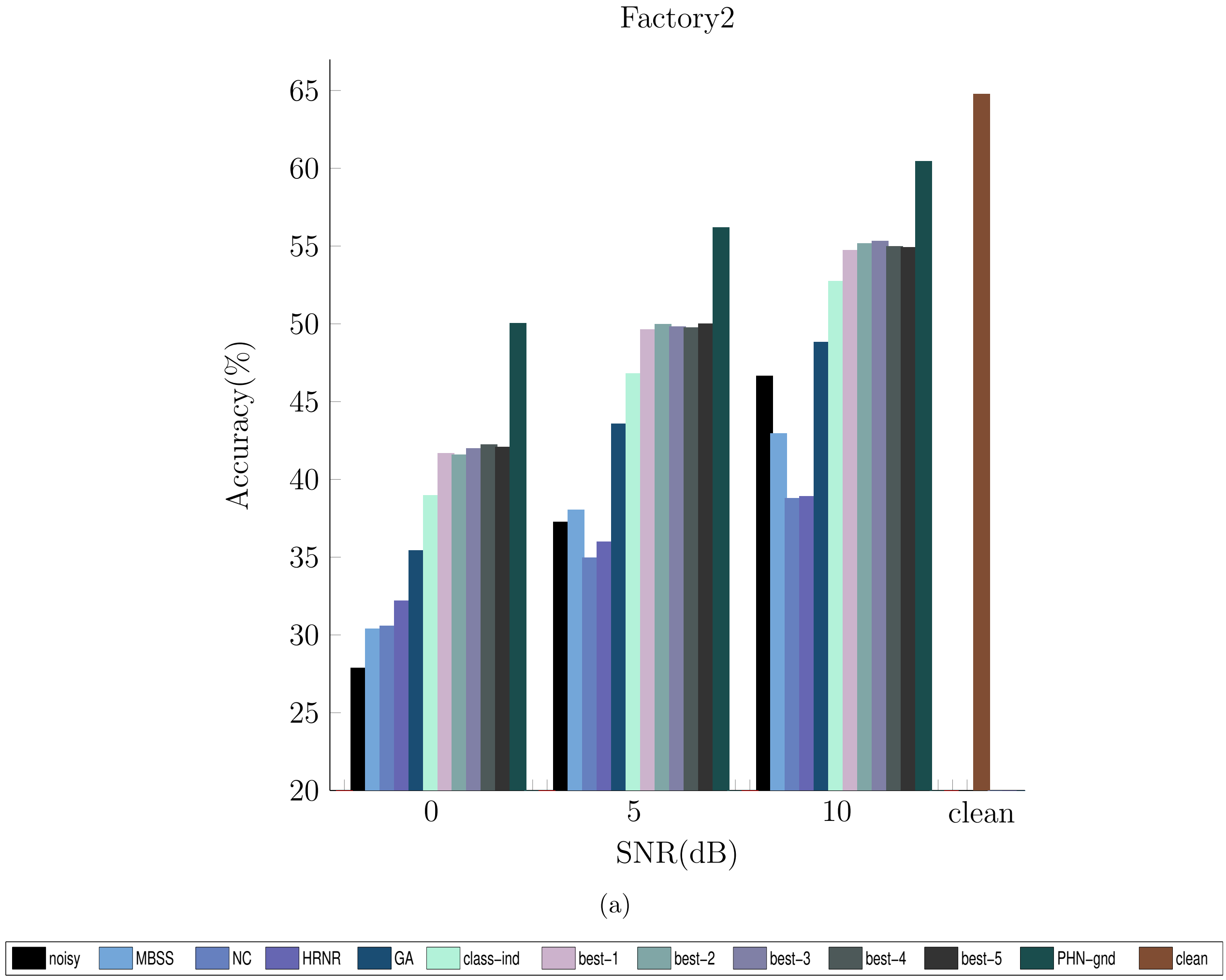


Figure 3.7: Best-N class-specific dictionary based enhancement-recognition scheme using JED with trigram confusion matrix

3.3.3.1 Results and discussion

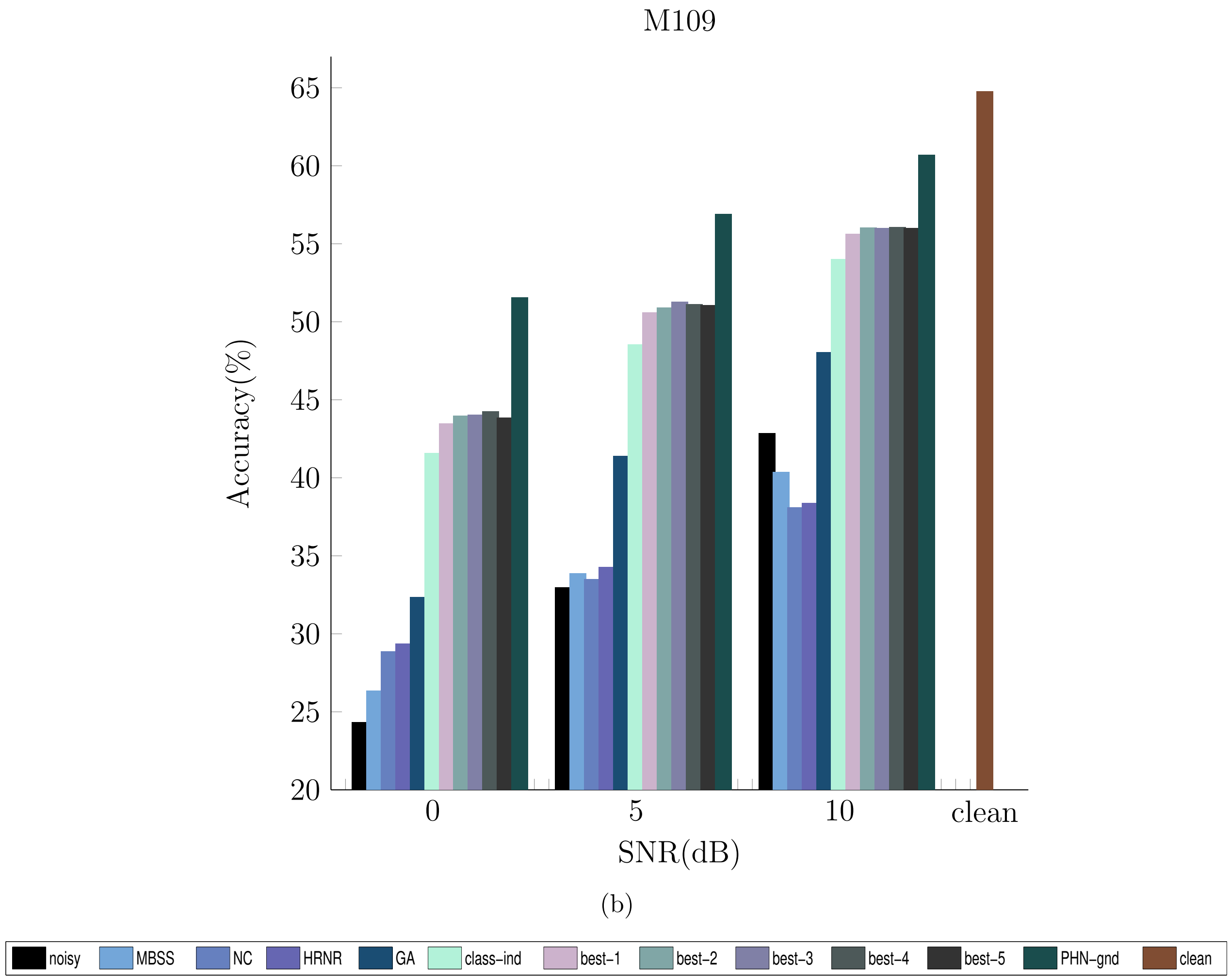
Figure 3.8 (a-e) shows the improvements in the phoneme recognition accuracies for the best-N class-specific enhancement-recognition scheme using JED using trigram confusion matrix for N varying from 1 through 5. The results for speech corrupted with noises (a) factory 2, (b) m109, (c) leopard, (d) babble and (e) volvo are shown. Similar to the previous cases, the recognition accuracies are compared with class-independent scheme, MBSS [18], NC [38], HRNR [39] and GA [19].

For speech corrupted with factory 2 noise the average RAI values (when averaged over 0, 5 and 10 dB SNRs) over class-independent case, for the scheme using trigram confusion matrix are 6.0%, 6.3%, 6.3% and 6.3% for $N = 2$ to 5. For m109 case, the values are 4.8%, 5.1%, 5.2% and 4.8 %. For leopard noise, the values are 3.8%, 4.5%, 4.4% and 4.4%. The RAI values for babble noise are 0.6%, 0.8%, 0.7% and 0.6%. For volvo noise case, the RAI values over noisy case are 0.4%, 0.3%, -0.2% and -0.3%. However, just like the previous two cases of using monogram and bigram confusion matrices, the performance is better than class-independent as well as the other four enhancement techniques MBSS, NC, HRNR and GA. The average RAI values over class independent scheme are 2.6%, 2.5%, 2.1% and 1.9%.



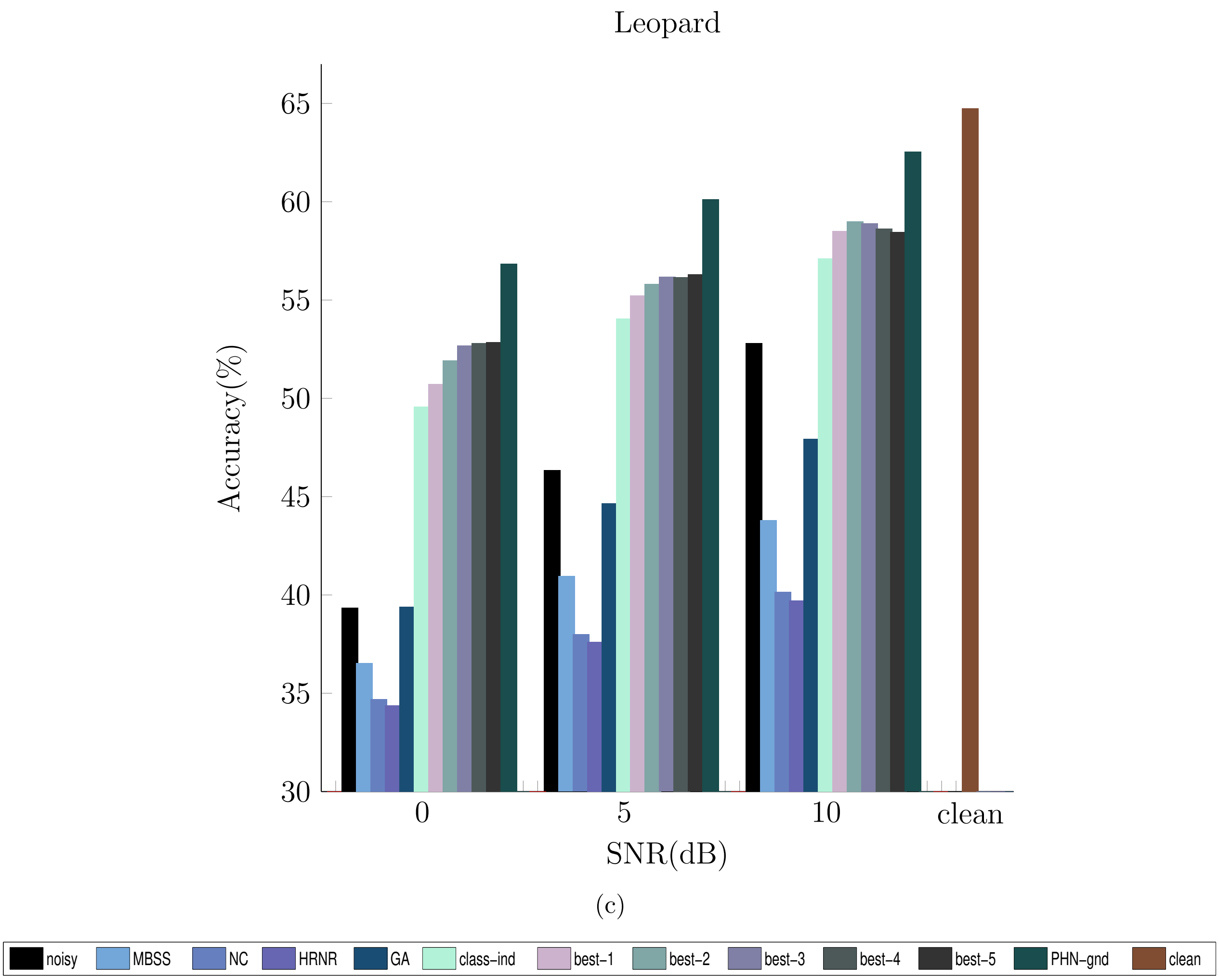
(a)

Figure 3.8: Performance of JED in terms of phoneme recognition accuracies for (a) Factory2 noise for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with trigram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.



(b)

Figure 3.8: Performance of JED in terms of phoneme recognition accuracies for (b) M109 noise for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with trigram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.



(c)

Figure 3.8: Performance of JED in terms of phoneme recognition accuracies for (c) Leopard noise for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with trigram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.

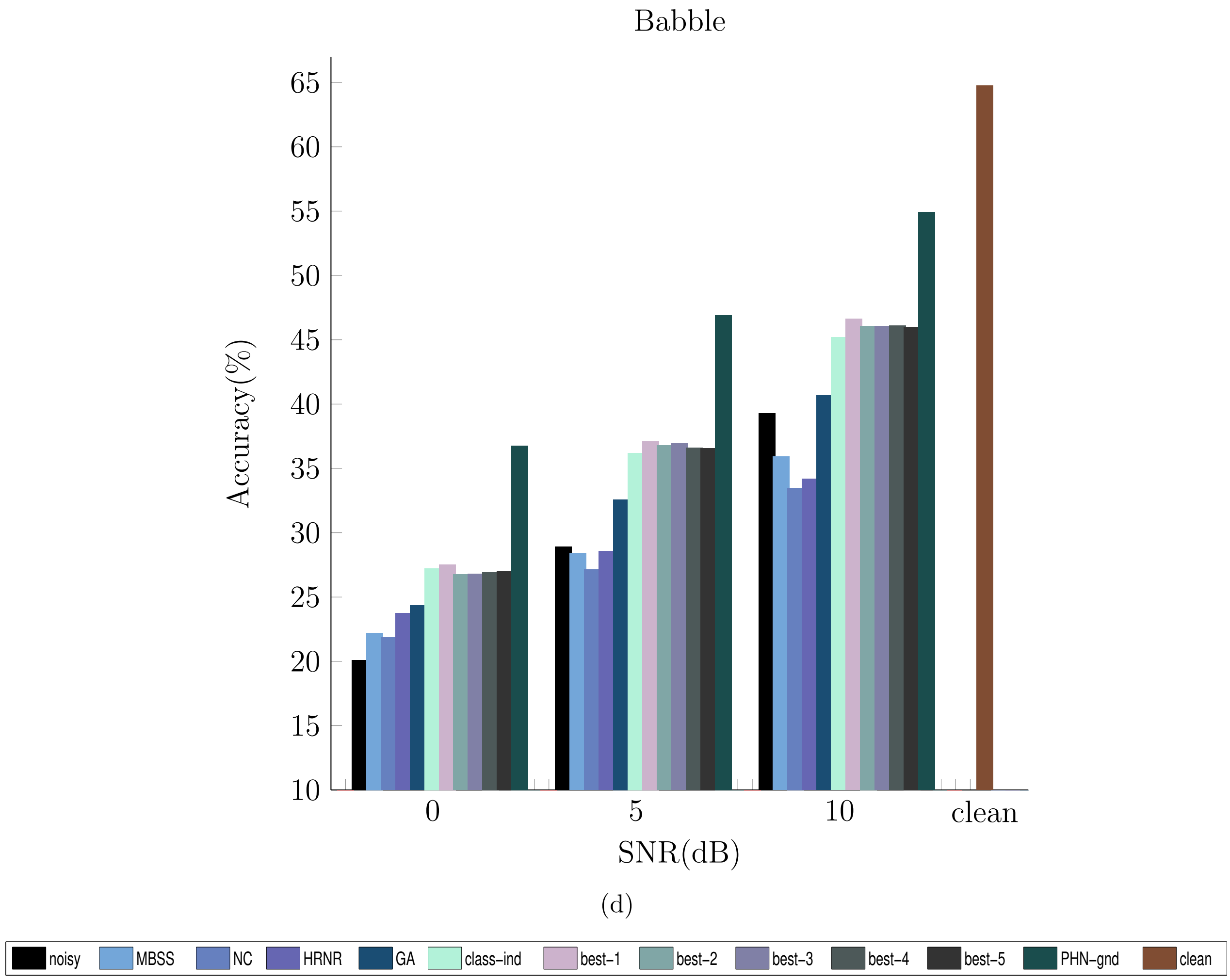
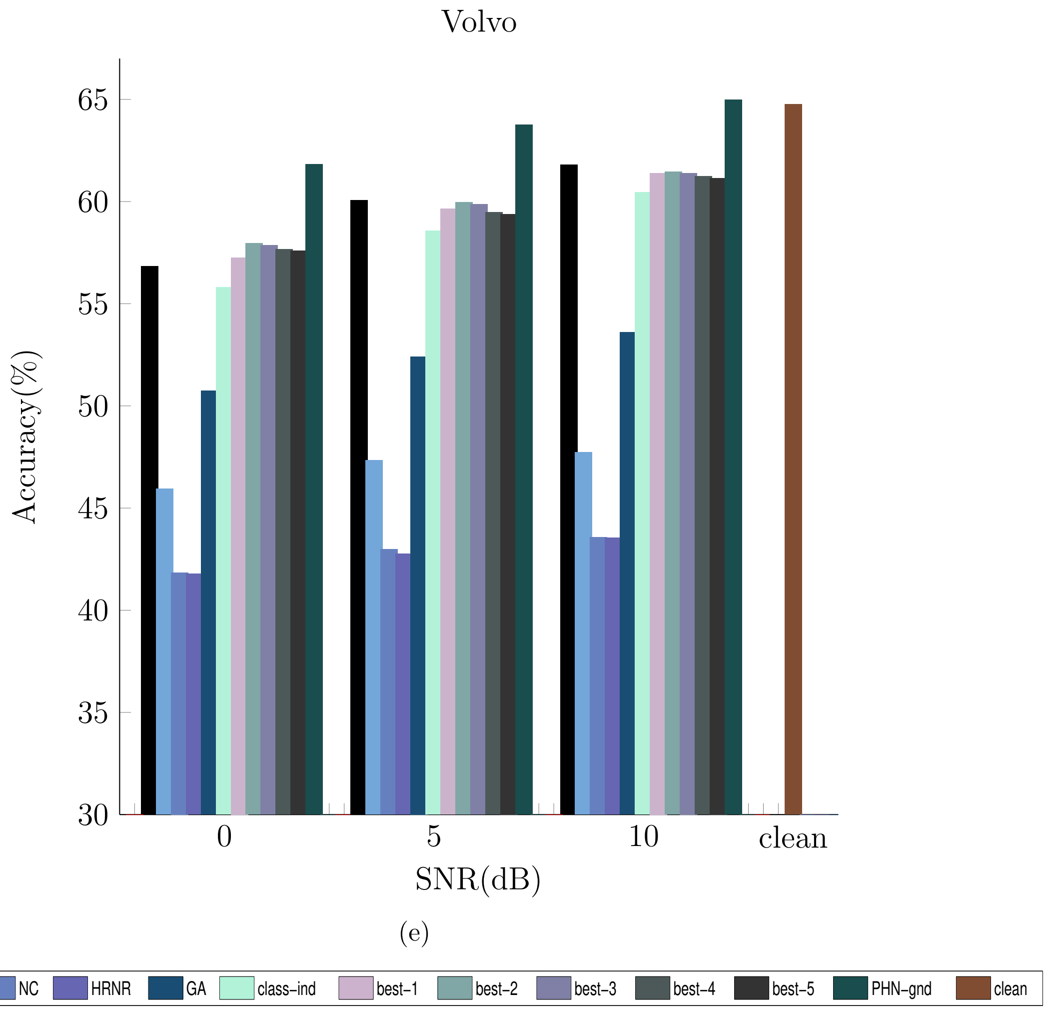


Figure 3.8: Performance of JED in terms of phoneme recognition accuracies for (d) Babble noise for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with trigram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.



(e)

Figure 3.8: Performance of JED in terms of phoneme recognition accuracies for (e) Volvo noise for SNRs of 0, 5 and 10 dB. For each SNR, the recognition accuracies are given for noisy speech, speech enhanced using MBSS, NC, HRNR, GA, class-independent scheme and best-N class-specific enhancement (best-N) schemes with trigram confusion matrix, for N varying from 1 to 5. PHN-gnd refers to the ideal case, when the ground truth phoneme labels are used for enhancement.

3.3.4 Comparison of phoneme recognition accuracies of best-N enhancement - recognition scheme using JED using monogram, bigram and trigram confusion matrices.

To get a better picture of the phoneme recognition results of the schemes using monogram, bigram and trigram confusion matrices given in secs. 3.3.1, 3.3.2.1 and 3.3.3.1, we average the results over all the three SNRs and five noises as shown in Table 3.3. It is observed that over the case of using a monogram confusion matrix, the bigram and trigram cases give only marginal improvements.

Table 3.3: Phoneme recognition accuracies for best-N ; N = 2 to 5 using n-gram confusion matrix (n = 1 to 3) averaged over Factory 2, Babble, Leopard, M109 and Volvo noises for 0, 5 and 10 dB SNRs

	best-2	best-3	best-4	best-5
Monogram	50.0	50.3	49.9	49.8
Bigram	50.2	50.3	50.2	50.2
Trigram	50.2	50.4	50.3	50.2

3.4 Conclusions

We analyzed the phoneme recognition performance of JED using best-N class-specific dictionaries, where the best-N labels could be selected based on a monogram, bigram or trigram confusion matrix. The recognition performance varies with N, giving the best values at N = 2 or 3 in most cases. The use of bigram and trigram confusion matrices for selection does not result in any marked improvement in performance compared to the monogram case. Further, the performance also depends on the type of noise corrupting the speech. Thus, in a real life scenario, the training data for noise model can be obtained from speech pauses using a voice activity detector to create the noise dictionary specific to the current environment [59].

The input observations for JED algorithm need not necessarily be enhanced using class-specific dictionary based approaches. The recognition performance of different enhancement techniques varies substantially over different noise types and SNRs [10]. Hence one can choose any other denoising technique depending upon the noise type and SNR. The proposed algorithm can thus be used to find the best enhancement scheme and recognition label for an input speech with any noise. We intend to explore in this direction in future.

Chapter 4

Monte Carlo dropout for low SNR, non-stationary, unseen noise reduction from speech

We propose methods to use dropout as a Bayesian estimator to improve the generalizability and performance of deep neural network (DNN) models for speech enhancement even with unseen and non-stationary noise. DNN model using Monte Carlo (MC) dropout performs better than the one using conventional dropout in unseen noisy conditions for low SNRs. A DNN is trained on speech with Factory2, M109, Babble, Leopard and Volvo noises at SNRs of 0, 5 and 10 dB. In another experiment, separate DNN models are trained, each on speech with one of the above noises at the same SNRs, using MC dropout. The trace of the covariance matrix (Var) of the output samples, resulting from different MC dropout trials, is used as a measure of the model precision (as a proxy for squared error) to select one out of these five models for each frame. We propose another algorithm with a threshold on Var to choose noise-classifier-based or model-precision-based selection scheme. Speech with unseen noises of White, Pink and Factory1 and all the five seen noises is used for testing. We also explore more realistic scenarios, where speech contains a mixture of noises or random number of segments of speech have different randomly chosen noises. In another significant experiment, we record real world, traffic noise and evaluate the performance of speech corrupted with this noise. Our algorithm performs well on real world, traffic noise from 10 down to -10 dB.

4.1 Introduction

Speech enhancement algorithms aim at the reduction of the noise associated with it without degrading the quality of speech. Several techniques have been proposed in the past for speech denoising, whose applications include speech recognition, speaker identification etc. Unsupervised enhancement techniques such as Wiener filter [109], spectral subtraction [19, 108], residual-weighting schemes [112–114], minimum mean-square error (MMSE) estimators [29] and Gaussian prior distribution based estimators [35, 115] are quite popular in the field. But these methods fail in the case of non-stationary noisy conditions.

Supervised learning techniques such as [50, 51, 53] have gained popularity due to their improved performance as they make use of prior information. With the introduction of multi layer perceptron (MLP), the denoising performance improved as they were able to better learn the complex mapping between noisy and clean speech [60, 62]. Xie and Compernolle [61] proposed the use of MLPs as nonlinear spectral estimators for noise reduction. But these networks are shallow and hence the mapping cannot be learned completely.

Deep architectures [63, 64] have revolutionized this field recently as they are able to learn this complex mapping between noisy and clean speech much better. Mass *et.al* proposed a model, which uses a deep recurrent auto encoder neural network for the enhancement of input features for a noise robust ASR [65].

A major issue encountered by DNN based enhancement is the degradation of enhancement performance when the characteristics of the noise corrupting the input speech is different from that of the training set [67, 68]. This kind of noise is referred to as unseen noise. The performance degrades for those noises for which the network is less adapted. Though not dealt separately, several techniques have been proposed in the past to address this problem. One intuitive way of doing this is to increase the training data by incorporating a variety of acoustic conditions. Wang *et.al.* [66] propose a DNN-SVM (support vector machine) system which is trained by including different acoustic conditions for a huge amount of time. They use 100 environmental noises to corrupt the training data. Xu *et. al.* proposed a wide DNN-based regression model with an RBM pre-training scheme. To improve the enhancement performance, they use more acoustic information and about 100 hours of noisy training data [67]. A noise aware DNN as a regression is proposed in [68], where noise information of the utterance is also appended along with the input vector to the DNN. In the above study, they use 104 different noise types leading to around 2500 hours of training data.

Ouyang *et.al.* [116] propose a DNN-based harmonic noise model (HNM) parameter estimator to learn the HNM parameters of clean speech from the spectrum of noisy speech. A hybrid signal processing/deep learning scheme is proposed in [117], where deep learning is used to estimate the noise statistics, which is then integrated into the Wiener filter-based enhancement structure to enhance speech. However, both [116] and [117] have not specifically addressed the problem of improving the generalizabity of the DNN employed, especially for unseen noises.

The experiments in this study aim at improving the generalizability of an existing DNN model for enhancement by replacing the conventional dropout [118, 119] by using Monte Carlo dropout proposed by Gal and Ghahramani in [87], particularly for unseen noisy scenario. In real life, when a noisy speech is encountered, one is not always sure that one knows the characteristics of the noise. When speech is recorded inside a moving car, for example, one knows that the constant, stationary noise of the engine is added to the speech. However, when the characteristics of the noise is unknown, there are two possibilities. The noise belongs to one of the classes of noise, with which the system (or model) has been trained. In this case of seen noise, the best way to handle it is to identify the class of the noise and use a model that is best suited for the same. However, if it is suspected that it does not belong to any of the classes of noise with which the system has been trained or if the noise is highly non-stationary, then it is better to use some ad-hoc strategy that is expected to best reduce the noise in the signal being enhanced. Considering such a scenario, we try to find a way of picking the best DNN model, where multiple DNN models are available for enhancement of speech corrupted with unseen noise, utilizing the intrinsic uncertainty of the models. This is the first work utilizing MC dropout for speech enhancement, to the best of our knowledge. Finally, we attempt at enhancing speech with time varying, unseen noise as well as real world, traffic noise.

4.2 Related work

The concept of dropout was first introduced to reduce overfitting while training a DNN model [118, 119]. Even though dropout omits certain neurons during training, all the neurons are active during the inference stage and contribute to the predicted output.

Gal and Ghahramani [87] propose a tool for modeling uncertainty in DNN using dropout during the inference stage. They show a probabilistic interpretation of dropout and its mathematical equivalence to a Gaussian process. Kendall *et al.* [120] use dropout as Bayesian approximation for the problem of camera relocalization and show that by averaging the results of multiple stochastic forward passes during inference, the performance could be improved. They use the term MC dropout to refer to this technique, as the output samples could be considered as Monte Carlo (MC) samples from the model posterior. This can be considered as a way of obtaining samples from the posterior distribution of models, from which an estimate of the uncertainty of the models can be obtained.

In this work, we use MC dropout to improve the generalizability of a DNN enhancement model and hence improve the performance when the input speech is corrupted by an unseen noise to which the DNN is less adapted [121]. In an initial set of experiments, we show that in the case of noisy speech corrupted with unseen noises, MC dropout models can give a better denoised output than conventional dropout models. To show this, we train two DNN models on multiple noises and SNRs, one employing MC dropout and another employing the conventional dropout and compare the performance of the two.

In another set of experiments, we use a measure of the model uncertainty for the selection of DNN models, where multiple noise-specific models are available for speech enhancement and compare the results with a DNN classifier-based model selection scheme [122]. The sample variance due to uncertainty [120] could be used as an estimate of prediction error for an input and hence this could be used to pick the optimal model for enhancing that particular frame. Model-specific enhancement techniques [11, 91, 110, 123, 124] have gained popularity recently which depend on a model selector, which ensures that the model chosen for enhancing each frame entails an overall improved performance. In practice there are cases where the noise conditions are known, and such knowledge could be used as addressed in [123]. In such scenarios a noise-specific approach is shown to be more useful. They employ multiple noise-specific DNN regression models for robust SNR estimation. On the other hand, in the case of an unknown noise, they use a DNN-based classifier to find the model matching closest to the input noise. This technique of using a DNN classifier for model selection is promising, but does not ameliorate the original problem of mismatch in training and testing conditions. Since our method uses the uncertainty information from the model output itself, it could be considered a better representative of the prediction error and also circumvents the issue of training mismatch, since, according to [87], the model uncertainty itself is an indicator of unseen data. We find this technique to be particularly useful for the case of unseen noises compared to that of using a classifier for model selection, since the model uncertainty is more for the case of unseen noise and hence picking a model out of the available ones which gives the minimum uncertainty could be promising. However, we find that blindly using uncertainty as a selection criterion could lead to a poor performance in the case of seen noises compared to classifier-based selection scheme. To circumvent this issue, we propose a threshold-based algorithm to switch between model uncertainty-based selection scheme and classifier-based model selection scheme. The algorithm is found to be useful for unseen noise cases at the same time giving comparable performance to that of classifier-based scheme for seen noises.

We also show some promising results in the enhancement performance of the above uncertainty-based algorithms on speech corrupted with a mixture of multiple noises. Another set of results is also shown for the case, where different segments of speech are corrupted by different noises. In another real world experiment, we record real world, traffic noise and add to clean speech in order to test our models.

Augmenting the baseline system with MC dropout could learn the distribution over the weights and give the uncertainty of the outputs. During testing, the input $Y_f \in \mathbb{R}^{R\times 1}$, which is the magnitude STFT of a frame of the noisy speech, is fed into the network using MC dropout. Multiple passes are made through the network, dropping out different random units each time. Thus J repetitions are performed by dropping of random units each time during testing. This results in J different outputs for a given input Y_f ; $\{\hat{S}_j(Y_f)\}; 1 \leq j \leq J$. Averaging these outputs of the forward passes through the network is equivalent to Monte Carlo integration over a Gaussian process posterior approximation, as shown in [87, 120].

For the uncertainty measurement, we use the trace of the covariance matrix of the output samples (Var) [120]. As explained in [120], the trace can be used as an efficient proxy measure of the model uncertainty.

4.3 Speech enhacement using DNN model

Let $y(m)$ be the m^{th} sample of speech, $s(m)$ corrupted with an additive noise $x(m)$;

$$y(m) = s(m) + x(m) \tag{4.1}$$

The short time Fourier transform (STFT) representation of the above is;

$$Y(\omega_k) = S(\omega_k) + X(\omega_k) \tag{4.2}$$

where k is the frequency index; $k = 0, 1, 2...R - 1$ and R is the number of frequency bins.

In our work, only the magnitude STFT is considered to train the DNN model. The phase of the noisy signal is retained for the reconstruction of the enhanced speech, considering the fact that the human ear is less sensitive to any phase distortion due to the noise. The magnitude STFT vector of the noisy speech can be approximated as

$$Y \approx S + X \in \mathbb{R}^{R\times 1} \tag{4.3}$$

where S and X represent the spectra of the clean speech and the noise, respectively.

The DNN based regression models are trained with the magnitude STFT features of the noisy speech as input and that of clean speech as target. During the inference stage, the input noisy test feature $Y_f \in \mathbb{R}^{R\times 1}$ of the fth frame is fed into the network to obtain the estimated enhanced feature vector $\hat{S}_f$. The inverse Fourier transform of $\hat{S}$ (together with the phase of the noisy speech) gives the enhanced speech signal.

4.3.1 DNN architecture for enhancement

The baseline DNN that we use consists of 3 fully connected layers of 2048 neurons and an output layer of 257. ReLu activation function is used in all the three layers as well as the output layer due to the nonnegative nature of magnitude STFT. The mean square logarithmic error (E_{lg}) between the noisy and clean magnitude spectra is minimized as the loss function:

$$E_{lg} = \frac{1}{R}\sum_{k=1}^{R}(log(S_k + 1) - log(\hat{S}_k + 1))^2 \tag{4.4}$$

where $\hat{S}_k$ and S_k denote the estimated and reference spectral features, respectively, at frequency index k. The architecture is based on the best performing DNN configuration in [68]. The baseline model is trained using conventional dropout. Figure 4.1 shows the basic framework of a DNN model for speech enhancement.

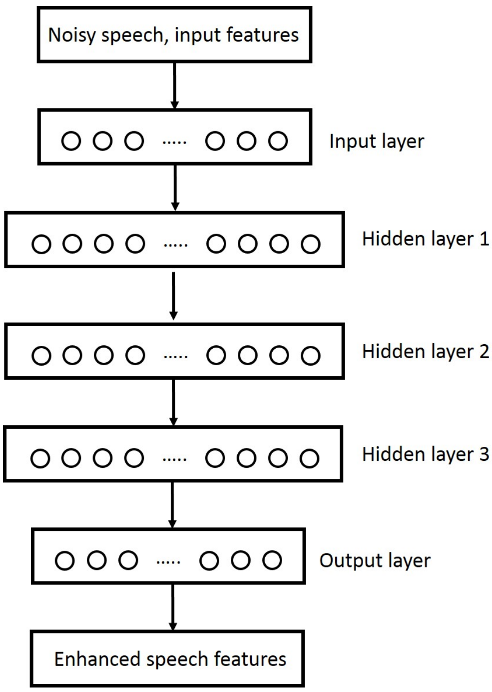


Figure 4.1: Framework of a DNN model for speech enhancement.

4.3.1.1 Dropout and MC dropout

Conventional dropout [118, 119] is a technique to prevent overfitting and combines many different neural network architectures efficiently. Dropout involves randomly dropping out units temporarily from networks along with their input and output connections during each mini-batch training. Thus training a neural network with dropout is similar to training a collection of thinned networks with reduced width. During testing time, a single neural network without dropout is used. If p is the probability with which a unit is retained during training, during testing, the outgoing weights of that unit are multiplied by p. Thus weights during test time are scaled down versions of the trained weights.

In Gal and Ghahramani's work [87], they extend the idea of dropout to approximate Bayesian inference and propose a method for modeling the uncertainty in DNN. Here dropout is used in a similar fashion during testing as during the training. Multiple forward passes of the input through the network during testing by dropping out random units results in empirical samples from an approximate predictive posterior.

4.4 MC dropout to improve generalization

Our study explores two main approaches. In the first approach, we show that MC dropout based estimation improves the generalization performance of a single DNN model trained on multiple noises and SNRs and apply this to speech enhancement.

In the second approach, we use model uncertainty to optimally choose one among multiple DNN models so that the reconstruction error is minimum in a scenario, where multiple models are available for enhancement. This analysis involves two sets of frameworks as explained in Secs. 4.4.2.2 and 4.4.2.3.

4.4.1 Single-MC: Single DNN model using MC dropout, trained with multiple noises

In this method, we use MC dropout to improve the generalizablity of a baseline DNN model trained on multiple noises and SNRs. We train a DNN model using MC dropout and evaluate the performance against the one using conventional dropout.

For this experiment, a single DNN model is trained on the magnitude STFT of speech corrupted with a set of noises and SNRs. During the inference stage, the test noisy speech is divided into frames and their magnitude STFT are obtained. The basic block diagram is

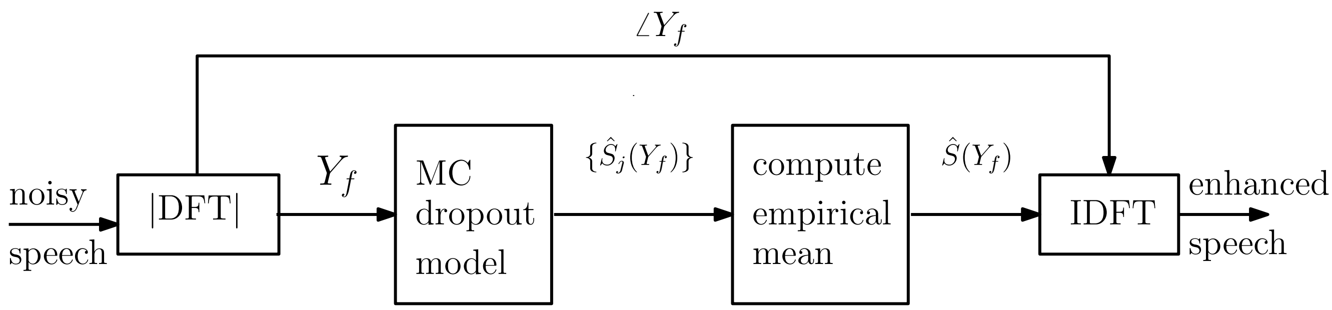


Figure 4.2: Single-MC : Enhancement using a single DNN-MC dropout model. The model is trained on speech corrupted with five noises and three SNRs. Y_f is the magnitude STFT vector of the f^{th} input frame of noisy speech.

shown in Fig. 4.2. Let $Y_f \in \mathbb{R}^{R\times 1}$ denote the magnitude STFT feature for the f^{th} frame. Given an input frame, J forward passes are carried out by dropping random units each time, giving J different outputs, $\{\hat{S}_j(Y_f)\}; 1 \leq j \leq J$. The empirical mean of these outputs is taken as the estimated enhanced frame $\hat{S}(Y_f)$. The time-domain enhanced speech signal is obtained by taking the inverse Fourier transform of this output with the original noisy speech phase, followed by an overlap-add method.

$$\hat{S}(Y_f) \approx \frac{1}{J}\sum_{j=1}^{J} \hat{S}_j(Y_f) \tag{4.5}$$

$$\hat{s}(y_f) = IDFT(\hat{S}(Y_f)\angle Y_f) \tag{4.6}$$

where $\hat{s}(y_f)$ is the enhanced speech estimate for the f^{th} frame of the noisy speech input y_f.

4.4.2 Choosing one out of multiple noise-specific MC dropout models for enhancing each input frame

Several model-specific enhancement techniques have been proposed in the past. The key idea is to select an appropriate model from a group of models so that there is an overall improvement in the enhancement performance for a given input [11, 91, 110, 124]. Given a framework of multiple DNN models for enhancement, one needs to select the appropriate model to enhance an input noisy speech frame. One possible method is to use a noise classifier [123, 125] to select the appropriate noise model. However, in scenarios where the input speech is corrupted with an unseen noise, the noise classifier might fail to pick the optimal model. In these cases, we need to ensure that the model chosen is the one that gives the lowest error and hence a better enhancement performance. In our methods proposed in Sec.s 4.4.2.2 and 4.4.2.3, we use a measure of the model uncertainty estimated from the output samples of each MC dropout model (Var) as an estimate of the prediction error and choose a model based on that. Our experiments show that higher the correlation between this uncertainty and the squared error, better is the enhancement performance.

4.4.2.1 Classifier-based model selection for comparison

For evaluating the performance, we compare our algorithms with the one where a classifier is used to pick the noise model. Here the noise model could be using MC dropout (class-MC) or conventional dropout (class-C).

4.4.2.2 Var-MC: Multiple models using MC dropout with predictive variance (model uncertainty) as the model selection criterion

This work explores the idea that a measure of the model uncertainty could be used as an estimate of the model error. Figure 4.3 shows the block diagram of this Var-MC model for enhancement.

M different DNN models with MC dropout are trained with speech corrupted with M distinct noises at various SNRs. The architecture of each model is as mentioned in section 4.3.1. During testing, the input noisy speech is first divided into frames and magnitude STFT is obtained. The magnitude STFT feature $Y_f \in \mathbb{R}^{R\times 1}$ of the f^{th} input frame is fed into each of these M models. J forward passes, by dropping out random nodes each time, are carried out and J outputs are obtained: $\{\hat{S}^i_j(Y_f)\}; 1 \leq j \leq J; 1 \leq i \leq M$; where i is the model index. Var values of each of these M output vectors are computed and the output of the model i^* with the minimum variance, $\{\hat{S}^{i^*}_j(Y_f)\}; 1 \leq j \leq J; 1 \leq i^* \leq M$, is selected. The corresponding model is considered the best for that particular input Y_f. The enhanced output $\hat{S}$ is estimated as the empirical mean of the J outputs: $\{\hat{S}^{i^*}_j(Y_f)\}; 1 \leq j \leq J$.

$$\hat{S}(Y_f) \approx \frac{1}{J}\sum_{j=1}^{J}\hat{S}^{i^*}_j(Y_f) \tag{4.7}$$

The enhanced speech signal is obtained as the inverse Fourier transform of $\hat{S}$ with the phase of the noisy speech signal and overlap-add method.

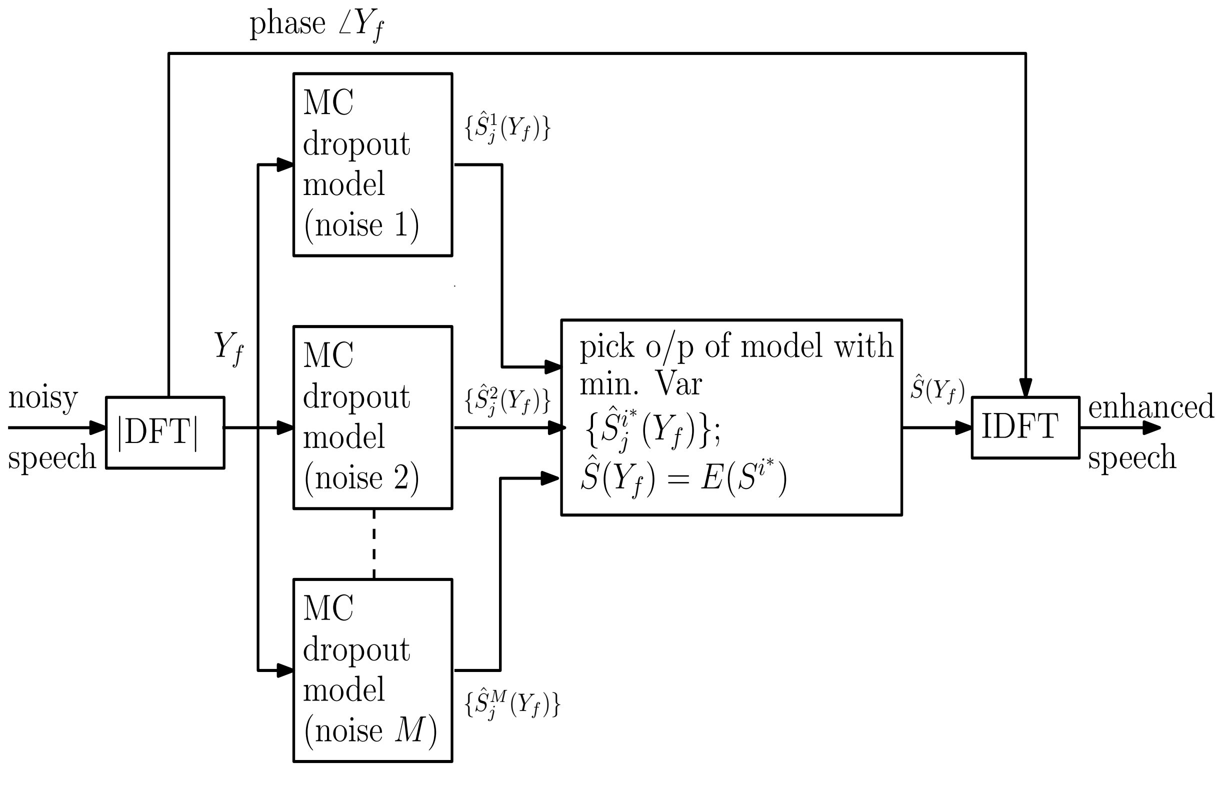


Figure 4.3: Var-MC : Enhancement using multiple DNN-MC dropout models with Var as the model selection criterion. Each model is trained on speech corrupted with a specific noise at three SNRs.

4.4.2.3 μ-MC: A Var threshold (μ) based algorithm to choose either classifier-based or model-uncertainty-based selection of model

The experimental results of the Var-MC algorithm show superior performance for most of the unseen noises. However, the performance on seen noise shows significant degradation compared to classifier-based selection scheme. This can be rectified using a conditional selection criterion for the noise models. Using this condition, selection of noise models can be switched from model uncertainty-based to classifier-based.

A threshold is set for the Var value of all the five models, so that the model for enhancing a noisy frame could either be selected on the basis of minimum variance scheme or on the basis of the decision of a noise classifier, as shown in Fig. 4.4.

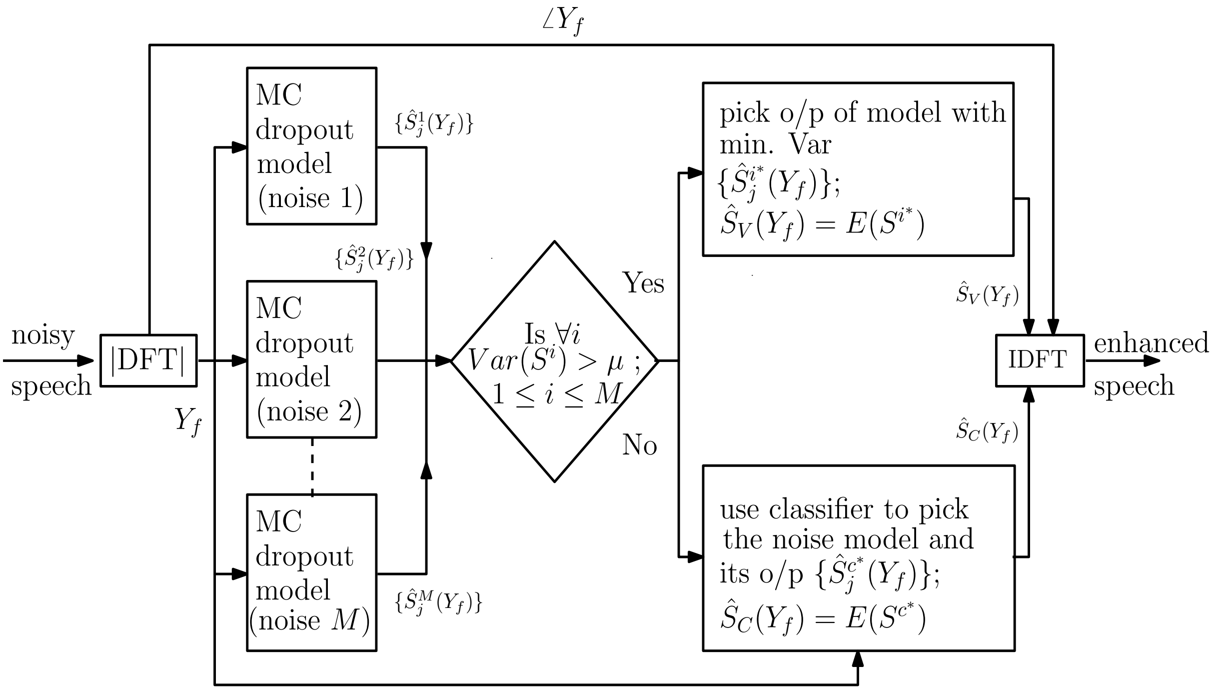


Figure 4.4: μ-MC : A Var threshold (μ) based algorithm for enhancement using multiple models trained on distinct noises. The appropriate model output is selected for each input frame of noisy speech, using model uncertainty as a selection criterion, or a noise classifier.

The input noisy feature of a frame $Y_f \in \mathbb{R}^{R\times 1}$, is fed into all the five MC dropout models. The input is passed J different times by dropping out random units each time. The corresponding outputs are $\{\hat{S}_j^i(Y_f)\}; 1 \leq j \leq J; 1 \leq i \leq M$; where i is the model index and $M = 5$. Then the $Var(S^i)$ of J outputs is computed for each of these M models. If all the M uncertainty values are above a threshold, say μ, it could be taken as an indication that the noise corrupting the given input speech belongs to none of these M noise models. In such a case, the model which gives the minimum Var value is considered as the best model to enhance the input noisy speech feature Y_f. The corresponding output, $\hat{S}_V(Y_f)$ is obtained as the empirical mean of the J outputs: $\{\hat{S}_j^{i^*}(Y_f)\}; 1 \leq j \leq J; 1 \leq i^* \leq M$.

$$\hat{S}_V(Y_f) \approx \frac{1}{J}\sum_{j=1}^{J} \hat{S}_j^{i*}(Y_f) \tag{4.8}$$

On the other hand, if the uncertainty values are below the threshold μ, the input feature Y_f is fed into a classifier to decide the best model, c^* for enhancing the frame. Let the outputs of the corresponding model be; $\{\hat{S}_j^{c^*}(Y_f)\}; 1 \leq j \leq J; 1 \leq c^* \leq M$. As mentioned previously, taking the empirical mean of these J different outputs gives the enhanced output $\hat{S}_C(Y_f)$.

$$\hat{S}_C(Y_f) \approx \frac{1}{J}\sum_{j=1}^{J} \hat{S}_j^{c*}(Y_f) \tag{4.9}$$

Inverse Fourier transform is applied on $\hat{S}$ with the noisy phase information to obtain the enhanced output.

4.5 Details of the experiments conducted

The speech and noise databases used for the experiments are as explained in sec. 2.3.1. The entire TIMIT training data is used for training and the test data is randomly chosen from the TIMIT test utterances. The DNN models are trained on magnitude STFT computed using a frame size of 30 ms with 10 ms frame shift after applying a Hamming window. Only the first 257 points are used out of the 512-point FFT due to the symmetry of the spectrum.

During the inference stage, the number of repetitions J is chosen as 50. Each DNN based regression model is trained with the magnitude STFT of noisy speech as input and clean speech as target. The Adam optimizer [126] is chosen. The dropout rate is set to 20%.

4.5.1 Single-MC experimental setup

For experiments using a single DNN (Sec.4.4.1), a baseline DNN with conventional dropout (single-C) [118, 119] and a DNN using MC dropout (single-MC) are trained using speech corrupted with factory 2, m109, leopard, babble and volvo noises at 0, 5 and 10 dB SNRs. The architectures of both the models are as mentioned in section 4.3.1. The DNN is trained using the entire TIMIT training data after adding five noises at three different SNRs.

4.5.2 Var-MC and μ-MC experimental setup

For multiple DNN model based experiments, five DNN models are separately trained on speech corrupted with factory2, m109, leopard, babble and volvo noises, each at SNRs 0, 5 and 10 dB. Each DNN model is trained separately using MC and conventional dropout, using the entire TIMIT training data after adding noises at SNRs 0, 5 and 10 dB. In this case also, the architecture of the models are as defined in section 4.3.1.

The testing is done using TIMIT test set corrupted with unseen noises white, pink and factory1 and seen noises factory2, m109, leopard, babble and volvo at SNRs varying from -10 dB to 10 dB..

4.5.2.1 Experiments with mixed, time-varying and real world, traffic noises

We also evaluate the performance of our Var-MC and μ-MC algorithms by mixing two unseen noises factory 1 and pink (mix) and corrupting the speech file with this new noise at SNRs varying from -10 dB to 10 dB. In another time-varying (TV1) noise experiment, the given speech waveform is divided into three segments and white (unseen), factory2 (seen) and factory 1 (unseen) noise is added to the distinct segments. We also show the evaluation on another non-stationary scenario where each test utterance of 2 to 3 seconds is divided into a random number (chosen to lie between 5 and 10) of segments of random lengths. One of the unseen noises white, factory1 and pink is randomly chosen to be added to these segments (TV2). This, we believe, is the closest one can simulate non-stationary time-varying noise, while still having access to the ground truth clean speech, for evaluating the enhancement performance of the algorithm in question.

We also have performed a real world experiment, where we record real world, traffic noise and add to clean speech to evaluate the performance of our algorithm. The noise is recorded from 'CV Raman road' for the experiments. We believe this experiment is significant, since in most cases DNN models for speech enhancement might be untrained on these real world noises, consequently giving poor performances on the same.

4.5.3 Noise classifier

For those experiments, where a DNN classifier is used to pick the model (class-MC and class-C), the classifier consists of 3 fully connected layers of 2048 neurons and an output layer of 5 neurons for the five noises. ReLu activation function is used in all the three layers and softmax activation function is used in the output layer. Categorical cross entropy is used as the loss function. The classifier is trained on the entire TIMIT training data, corrupted with factory2, babble, leopard, m109 and volvo noises at SNRs 0, 5 and 10 dB.

4.6 Results and discussion

4.6.1 Performance of single-MC model for unseen and seen noises

Table 4.1 shows the results obtained in terms of sum squared error (SSE), and segmental SNR (SSNR) [69] for the single DNN-MC dropout model (single-MC) over the baseline (single-C) for unseen and seen noises. For comparison, the SSE and the SSNR values of the input noisy speech are also shown in the first column of the Table. SSE is computed in the magnitude STFT domain. We use white, pink and factory 1 noises as unseen noises and factory2 as a seen noise. The results are averaged over 50 files randomly selected from TIMIT [95] test set. Table 4.1 shows the superior performance of the single-MC over single-C, for unseen noises at lower SNRs. It could be observed that the performance is similar to single-C at higher SNRs, though in terms of SSE, the model performs better than single-C. We believe that the improvements in terms of SSE could be significant for the potential use of this enhancement approach in applications like speech recognition [11]. For seen noise Factory2 , the performance of the MC dropout model is comparable to that of single-C. Thus the conventional dropout of any DNN model for speech enhancement could be replaced with MC dropout to improve the enhancement performance in unseen noise case without degrading the seen noise performance. These observations are in line with our aim of improving the generalizability of a DNN model for enhancement of speech corrupted with unseen noises.

Table 4.1: Performance evaluation of single DNN model with MC dropout (single-MC) for speech corrupted with three unseen noises, (white, pink and factory1) and one seen noise (factory2) at SNRs -10, -5, 0, 5 and 10 dB averaged over 50 files, randomly selected from TIMIT test set. SSE: sum squared error; SSNR: segmental SNR. single-C: Single DNN model with conventional dropout, as the baseline system for comparison. The SSE values listed have been scaled down by a factor of 1000 and this multiplicative factor is given in the metric column of the Table.

		White (unseen)			Pink (unseen)			Factory1 (unseen)			Factory2 (seen)		
SNR (dB)	Metric	Noisy input	single-C	single-MC	Noisy input	single-C	single-MC	Noisy input	single-C	single-MC	Noisy input	single-C	single-MC
10	SSE x10^3	.308	.270	**.267**	.341	.118	**.116**	.309	.107	**.106**	.382	0.056	**0.055**
	SSNR	2.0	2.7	**2.7**	2.2	4.7	**4.7**	2.3	5.0	**5.0**	2.6	8.9	**8.9**
5	SSE x10^3	1.03	0.844	**0.827**	1.12	0.291	**0.288**	1.02	0.244	**0.242**	1.24	0.069	**0.069**
	SSNR	-1.6	-0.7	**-0.7**	-1.4	1.7	**1.7**	-1.3	2.2	**2.2**	-0.9	7.1	**7.1**
0	SSE x10^3	3.41	2.81	**2.60**	3.71	0.858	**0.843**	3.41	0.682	**0.671**	4.01	0.104	**0.104**
	SSNR	-4.6	-3.9	**-3.8**	-4.5	-1.5	**-1.4**	-4.4	-0.7	**-0.7**	-4.1	5.1	**5.1**
-5	SSE x10^3	11.2	9.60	**9.13**	12.2	2.70	**2.51**	11.2	2.13	**2.00**	12.9	0.198	**0.197**
	SSNR	-7.2	-6.6	**-6.5**	-7.1	-4.3	**-4.2**	-6.9	-3.51	**-3.50**	-6.7	3.05	**3.08**
-10	SSE x10^3	36.4	33.6	**31.4**	39.6	8.74	**8.48**	36.9	7.20	**7.0**	41.3	0.467	**0.461**
	SSNR	-8.9	-8.5	**-8.4**	-8.8	-6.7	**-6.6**	-8.7	-6.0	**-5.9**	-8.5	1.0	**1.0**

4.6.2 Performance of Var-MC model for unseen noises

Tables 4.2 show the performance of our Var-MC algorithm in terms of SSNR for unseen noises. The corresponding SSE comparisons for unseen noises are given as bar charts in Fig. 4.5. It can be inferred from the Table and the Figure that, Var-MC gives superior performance over class-C and class-MC algorithms especially at lower SNRs for most of the unseen noise cases. For unseen noises, the performance of Var-MC drops as SNR increases. This drop in performance could be explained by the correlation plots illustrated in Fig. 4.6, which show the correlation between Var and the frame-wise squared error (SE) of the output frames for the five MC dropout models. The plots are for input speech corrupted with white noise at SNRs varying from -10 dB to 10 dB. From these plots, it could be seen that the correlation is stronger for lower SNRs, -10 dB and -5 dB, but weakens as SNR increases. This peculiar pattern needs further exploration. The observation also matches with that in [120], that the uncertainty increases for the case where the properties of test set are far from those of training set. This is reflected in our Var-MC results as well (Table 4.2, Fig. 4.5), since there is not much improvement over the class-C and class-MC as the SNR increases. The values at higher SNRs are still comparable to class-C and class-MC values for most unseen noise cases.

Table 4.2: **Results on unseen noises:** Performance comparison (in terms of **SSNR: segmental SNR**) of Var-MC and μ-MC algorithms with class-C and class-MC for speech corrupted with white, pink and factory1 noises at SNRs -10, -5, 0, 5 and 10 dB averaged over 50 files randomly selected from TIMIT test set. Improvement could be noticed especially for low SNRs.

	White (Unseen)					Pink (Unseen)					Factory1 (Unseen)				
SNR (dB)	Noisy input	Class-C	Class-MC	Var-MC	μ-MC $\mu = 0.16$	Noisy input	Class-C	Class-MC	Var-MC	μ-MC $\mu = 0.16$	Noisy input	Class-C	Class-MC	Var-MC	μ-MC $\mu = 0.16$
10	2.0	2.6	2.6	**2.7**	**2.7**	2.2	4.8	4.8	4.5	4.7	2.3	4.9	4.9	4.8	**4.9**
5	-1.6	-0.8	-0.8	**-0.7**	**-0.7**	-1.4	1.7	1.7	1.6	**1.7**	-1.3	2.0	2.0	**2.0**	**2.0**
0	-4.6	-4.1	-4.0	**-3.8**	**-4.0**	-4.5	-1.6	-1.6	**-1.3**	**-1.6**	-4.4	-1.1	-1.1	**-0.83**	**-1.1**
-5	-7.2	-6.7	-6.6	**-6.5**	**-6.6**	-7.1	-4.5	-4.5	**-3.7**	**-4.5**	-6.9	-4.1	-4.1	**-3.3**	**-4.0**
-10	-8.9	-8.7	-8.6	**-8.4**	**-8.5**	-8.8	-7.1	-7.1	**-5.4**	**-6.9**	-8.7	-6.6	-6.6	**-5.3**	**-6.3**

4.6.3 Observations on the performance of Var-MC model for seen noises

Tables 4.3 and 4.4 and Fig. 4.7 for seen noises show that Var-MC performs really poorly compared to class-C and class-MC for seen noises like factory2, m109, leopard, babble and

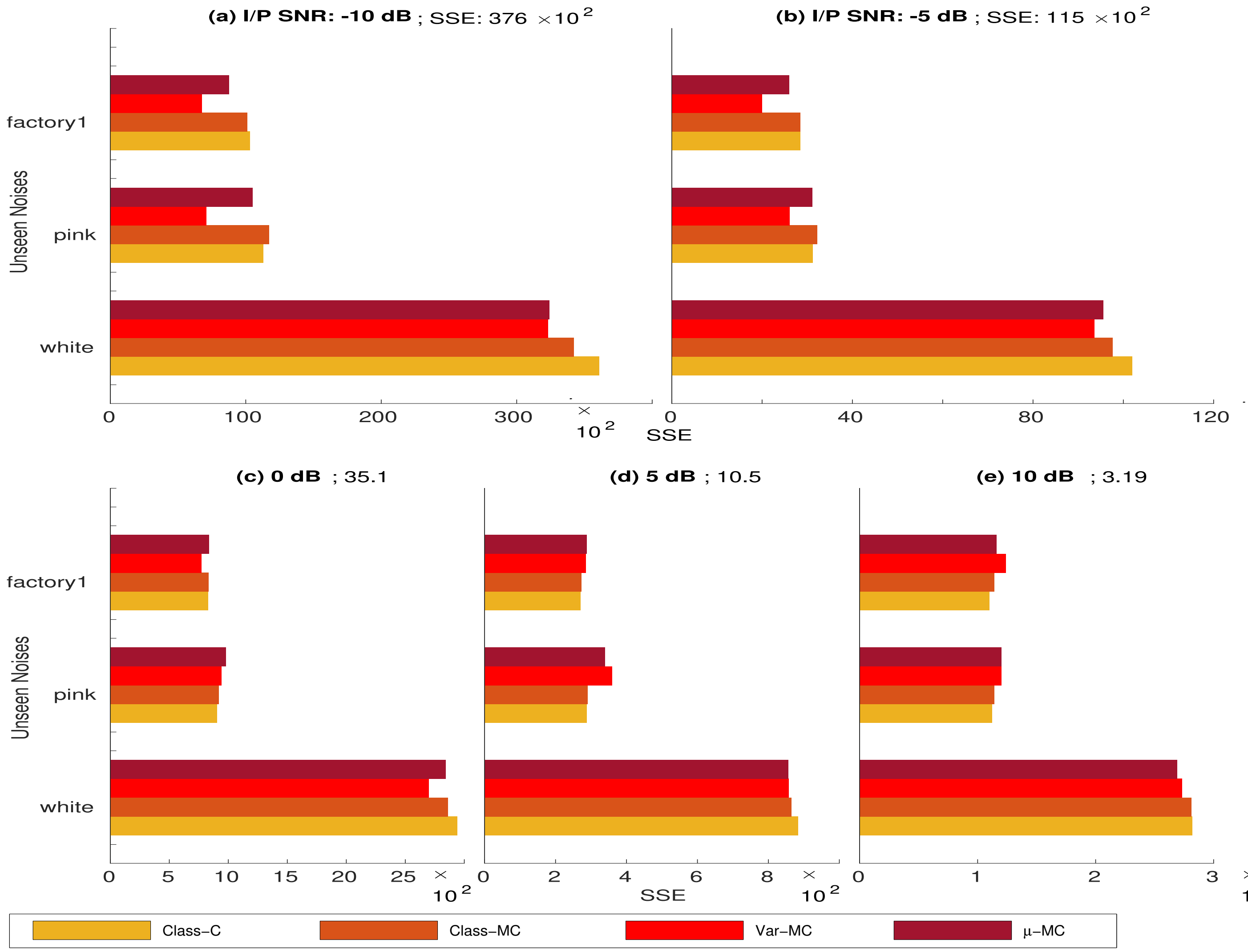


Figure 4.5: Performance comparison in terms of SSE (sum squared error) of Var-MC and μ-MC algorithms with class-C and class-MC for speech corrupted with unseen noises white, pink and factory1 at SNRs (a) -10 dB (b) -5 dB (c) 0 dB (d) 5 dB and (e) 10 dB averaged over 50 files randomly selected from TIMIT test set. The noisy speech SSE values averaged over all the three noises for (a) -10 dB: 376×10^2 (b) -5 dB: 115×10^2 (c) 0 dB: 35.1×10^2 (d) 5 dB: 10.5×10^2 (e) 10 dB: 3.19×10^2 (These noisy speech SSE values are also shown along with each SNR in the plots; the scaling factor of $\times 10^2$ is omitted. The noisy speech SSE bar is omitted as the values are too high in comparison to the rest).

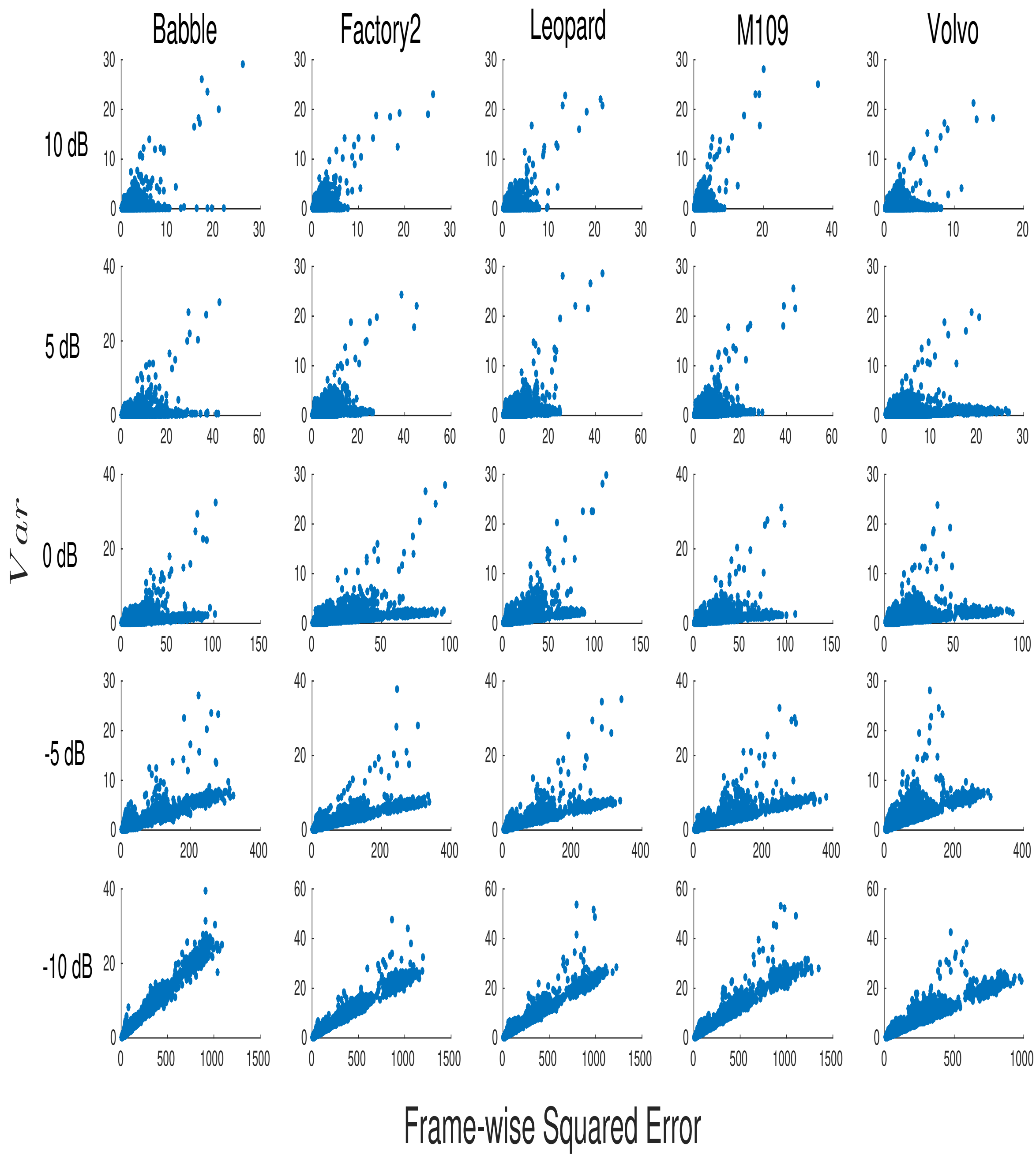


Figure 4.6: Correlation plot between Var and the squared error of the estimated output frames for all the five MC models for the case of speech corrupted with the unseen white noise at -10, -5, 0, 5 and 10 dB SNRs as input. It could be seen that the correlation is stronger for lower SNRs, -10 dB and -5 dB, but weakens as SNR increases. This is reflected in our results as well , since there is not much improvement over the class-C and class-MC as the SNR increases

volvo. μ-MC algorithm compensates for this performance drop by using per frame threshold μ to select between the Var-MC and class-MC schemes.

The threshold is selected based on the experiments on a validation set of 30 files from TIMIT corrupted with seen noises factory 2, m109, leopard, babble and volvo and unseen pink noise at SNRs -10, -5 , 0, 5 and 10 dB. For our experiments, this threshold is set at $\mu = 0.16$.

In the case of volvo noise, the performance of Var-MC over class-C and class-MC is too poor compared to its performance on other seen noises. This could be explained by the highly band-limited and predictable nature of volvo noise resulting in the poor performance of uncertainty based selection.

Table 4.3: **Results on seen noises:** Performance evaluation (in terms of **SSNR: segmental SNR**) of Var-MC and μ-MC algorithms compared to class-C and class-MC for speech corrupted with seen noises, namely, factory2, leopard and m109, at SNRs -10, -5, 0, 5 and 10 dB averaged over 50 files randomly selected from TIMIT test set.

	Factory 2 (Seen)					Leopard (Seen)					M109 (Seen)				
SNR (dB)	Noisy input	Class-C	Class-MC	Var-MC	μ-MC $\mu = 0.16$	Noisy input	Class-C	Class-MC	Var-MC	μ-MC $\mu = 0.16$	Noisy input	Class-C	Class-MC	Var-MC	μ-MC $\mu = 0.16$
10	2.6	9.5	9.5	8.1	**9.5**	2.5	8.9	8.9	8.5	**8.9**	2.5	9.1	9.1	8.1	**9.1**
5	-0.9	7.7	7.7	5.8	7.6	-1.1	7.4	7.4	7.0	**7.4**	-1.1	7.3	7.3	6.3	**7.3**
0	-4.1	5.8	5.8	3.3	**5.8**	-4.3	5.9	5.9	5.6	**5.9**	-4.2	5.3	5.3	4.3	**5.3**
-5	-6.7	4.0	4.0	1.3	3.9	-6.8	4.3	4.4	4.2	**4.3**	-6.8	3.5	3.5	2.5	**3.5**
-10	-8.5	2.1	2.1	0.5	**2.1**	-8.6	2.7	2.9	2.7	**2.8**	-8..6	1.9	1.9	1.0	**1.9**

Table 4.4: **Results on seen noises:** Performance evaluation (in terms of **SSNR: segmental SNR**) of class-C, class-MC, Var-MC and μ-MC algorithms, for speech corrupted with seen noises, babble and volvo at SNRs -10, -5, 0, 5 and 10 dB averaged over 50 files randomly selected from TIMIT test set.

	Babble (seen)					Volvo (seen)				
SNR (dB)	Noisy input	Class-C	Class-MC	Var-MC	μ-MC $\mu = 0.16$	Noisy input	Class-C	Class-MC	Var-MC	μ-MC $\mu = 0.16$
10	2.6	7.5	7.5	7.0	**7.5**	3.3	12.9	12.9	7.5	12.8
5	-1.0	5.8	5.8	5.3	**5.8**	-0.3	12.1	12.1	4.6	12.0
0	-4.1	4.2	4.2	3.8	**4.2**	-3.6	10.8	10.8	2.1	**10.8**
-5	-6.7	2.7	2.7	2.4	**2.7**	-6.3	9.1	9.1	0.8	**9.1**
-10	-8.5	1.5	1.5	1.3	**1.5**	-8.2	6.7	6.7	0.2	**6.7**

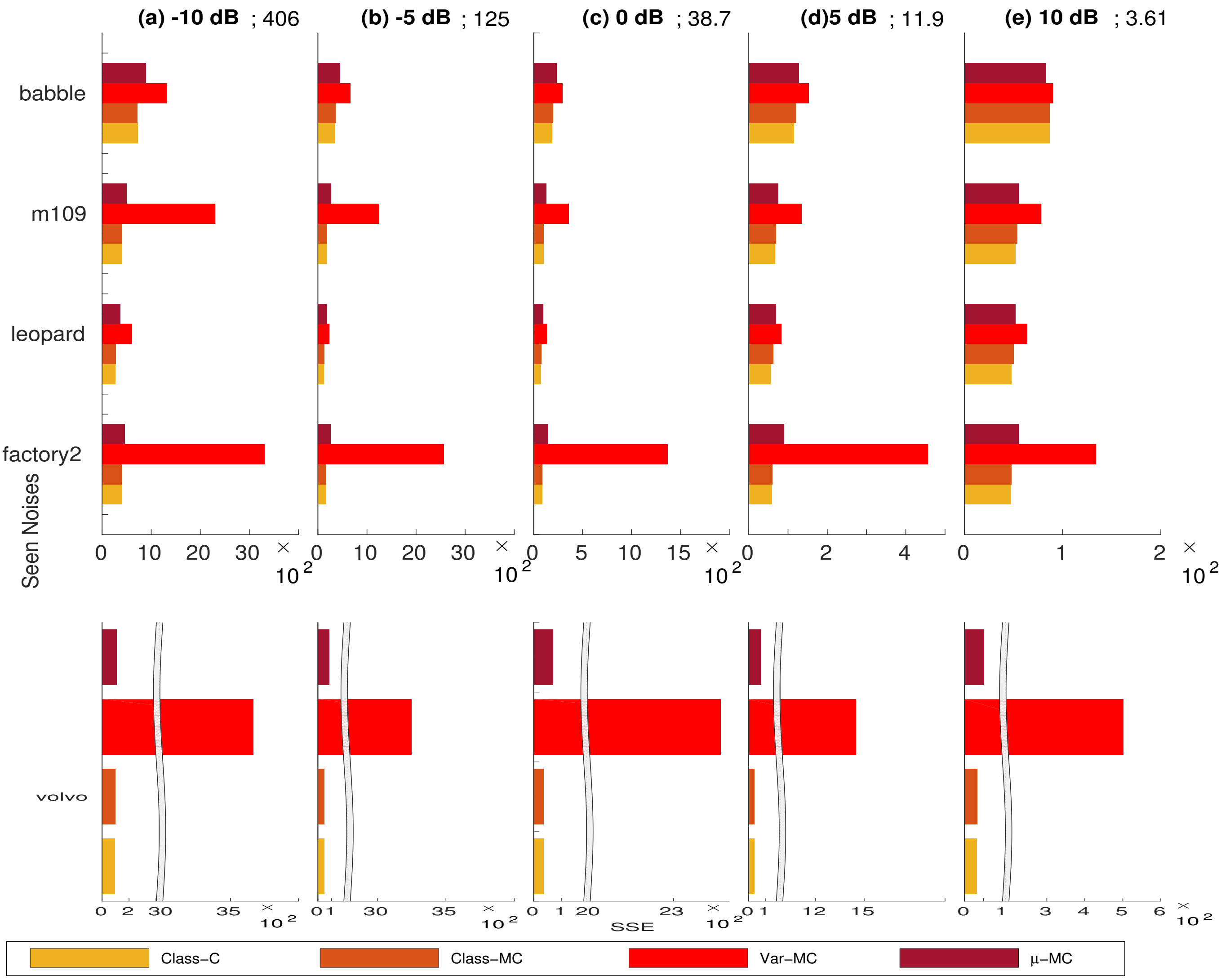


Figure 4.7: Performance comparison in terms of SSE (sum squared error) of Var-MC and μ-MC algorithms with class-C and class-MC for speech corrupted with seen noises babble, m109, leopard, factory2 and volvo at SNRs (a) -10 dB (b) -5 dB (c) 0 dB (d) 5 dB and (e) 10 dB averaged over 50 files randomly selected from TIMIT test set. The noisy speech SSE values averaged over all the five noises for (a) -10 dB: 406×10^2 (b) -5 dB: 125×10^2 (c) 0 dB: 38.7×10^2 (d) 5 dB: 11.9×10^2 (e) 10 dB: 3.61×10^2 (These noisy speech SSE values are also shown along with each SNR in the plots; the scaling factor of $\times 10^2$ is omitted.)

4.6.4 Results of μ-MC model on unseen and seen noises

Table 4.2 shows the performance improvements of μ-MC algorithm over class-C and class-MC in terms of SSNR for unseen noises pink, white and factory 1. The comparison plot for SSE for unseen noises can be seen in Fig 4.5. Tables 4.3 and 4.4 and Fig 4.7 show the same for seen noises factory 2, m109, leopard, babble and volvo. $\mu - MC$ gives better performance than class-C and class-MC in most of the unseen noise cases, especially at lower SNRs, though Var-MC gives the best performance of all. The algorithm also compensates for the poor performance of Var-MC algorithm for seen noises and gives performance comparable to class-C and class-MC.

The variation of SSE with the threshold μ, for the test data of 50 random files corrupted with all the five seen and three unseen noises for -10 dB SNR is shown in Fig. 4.8. It is seen that as the threshold increases, the performance on unseen noises degrades, while that on seen noises improves. Thus, the threshold μ can be used to trade-off between the performance on seen and unseen noise cases for the μ-MC algorithm.

4.6.5 Observations on mixed and time-varying noises

Table 4.5 and Fig. 4.9 show the performance evaluation of mix and TV1 experiments in terms of SSNR and SSE, respectively. It can be observed that μ-MC algorithm gives performance superior or comparable to Class-C and Class-MC in all the cases. The algorithm Var-MC gives the best performance of all, for those cases for which the DNN is less adapted and hence where the correlation between squared error and variance is strong.

Table 4.6 shows the SSNR performance evaluation for the TV2 experiment. The performance in terms of SSE is shown in Fig. 4.9. Here also we can see that, μ-MC algorithm gives better performance for low SNRs -10 and -5 dB than Class-C and Class-MC and comparable performance for higher SNRs. The performance of Var-MC algorithm is the best at low SNRs and degrades at higher SNRs, as expected.

4.6.6 Observations on real world, traffic noise

Results reported in Table 4.7 show the SSNR performances of Var-MC and μ-MC algorithms on speech corrupted with real world, traffic noise that we have recorded. Figure 4.10 shows the performance evaluation in terms of SSE. It can be observed that both Var-MC and μ-MC algorithms give performances superior to class-MC and class-C at all SNRs varying from -10 dB to 10 dB for this case.

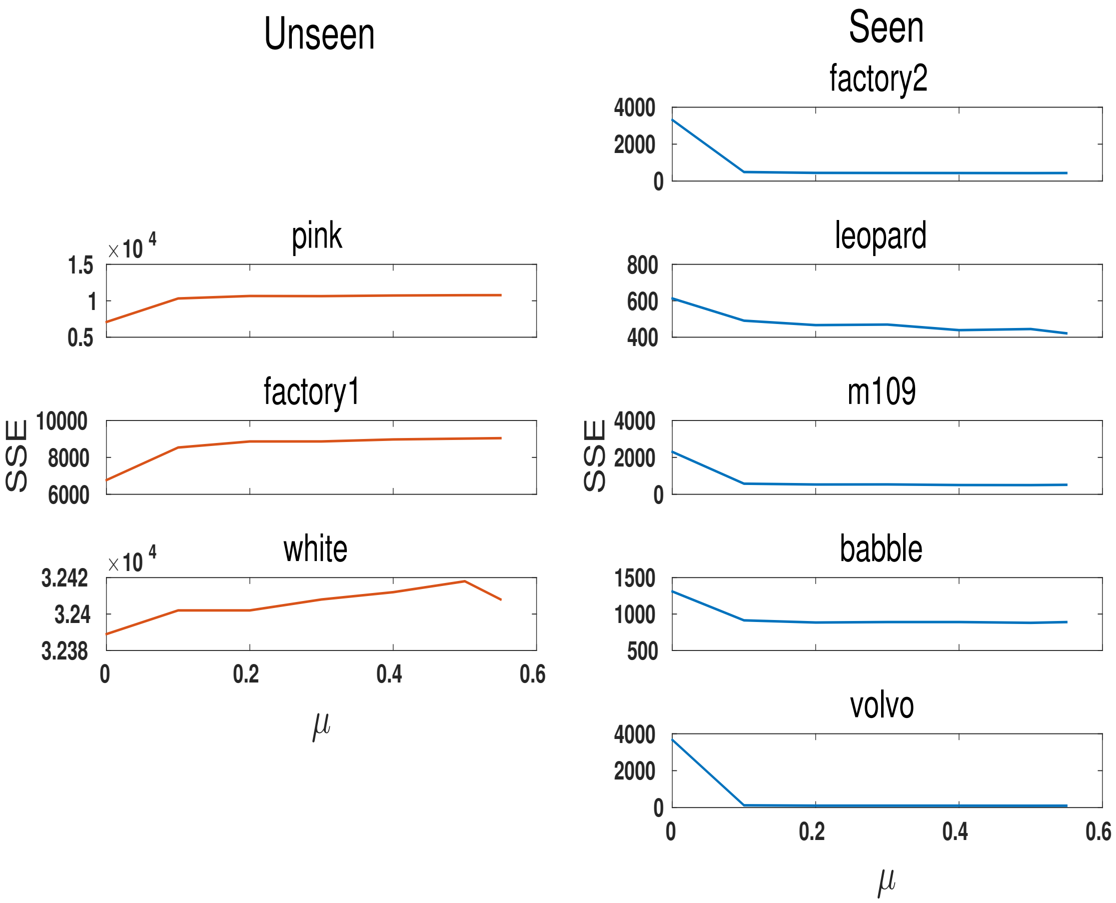


Figure 4.8: Variation of SSE with μ averaged over the test data of 50 random files corrupted with three unseen and five seen noises at -10dB SNR. As the threshold increases, the performance on unseen noises degrades, while that on seen noises improves. Thus, the threshold μ can be used to trade-off between the performance of seen and unseen noise cases.

Table 4.5: **Mixed or time varying noise experiments:** Performance evaluation (in terms of **SSNR: segmental SNR**) of Var-MC and μ-MC algorithms for two cases. In the first case, speech is corrupted with a mixture of unseen noises, factory1 and pink. In the second case, the given speech waveform is divided into three segments and white, factory2 and factory 1 noises are added to the different segments. The results averaged over 50 files randomly selected from TIMIT test set show improvement for low SNRs of -5 and -10 dB.

	mix: Additive noise Factory1+Pink (unseen)					TV1: White-Factory2-Factory1 noises added segment-wise				
SNR (dB)	Noisy input	Class-C	Class-MC	Var-MC	μ-MC $\mu = 0.16$	Noisy input	Class-C	Class-MC	Var-MC	μ-MC $\mu = 0.16$
10	2.2	4.8	4.8	4.6	**4.8**	4.4	6.8	6.8	6.3	**6.9**
5	-1.3	1.8	1.8	**1.8**	**1.8**	0.7	4.5	4.5	3.8	**4.5**
0	-4.5	-1.3	-1.3	**-1.0**	**-1.3**	-2.5	1.9	1.9	1.2	**1.9**
-5	-7.0	-4.3	-4.3	**-3.5**	**-4.1**	-5.2	-0.9	-0.9	-1.0	**-0.8**
-10	-8.8	-6.8	-6.8	**-5.5**	**-6.5**	-7.2	-3.5	-3.5	**-2.8**	**-3.4**

Table 4.6: **Non-stationary unseen noise experiments:** Performance comparison (in terms of **SSNR: segmental SNR**) of Var-MC and μ-MC algorithms with class-C and class-MC for simulated nonstationary noise. Each test utterance of duration 2 to 3 sec. is divided into a random number (5 to 10) of segments of random length and unseen noises white, factory1 and pink are added randomly to these segments. The results are averaged over 50 files randomly selected from TIMIT test set.

	TV2: White-Factory1-Pink noises (unseen) added randomly to speech segments of random length				
SNR (dB)	**Noisy input**	**Class-C**	**Class-MC**	**Var-MC**	**μ-MC** $\mu = 0.16$
10	3.0	4.9	4.9	4.7	**4.9**
5	-0.6	1.9	1.9	**1.9**	**1.9**
0	-3.9	-1.4	-1.4	**-1.2**	**-1.4**
-5	-6.5	-4.3	-4.3	**-3.7**	**-4.2**
-10	-8.4	-6.9	-6.9	**-5.6**	**-6.7**

Table 4.7: **Real world, traffic noise experiments:** Performance comparison (in terms of **SSNR: segmental SNR**) of Var-MC and μ-MC algorithms with class-C and class-MC for speech corrupted with real world, traffic noise. The results are averaged over 50 files randomly selected from TIMIT test set.

	Traffic: Speech corrupted with real world, traffic noise (unseen)				
SNR (dB)	**Noisy input**	**Class-C**	**Class-MC**	**Var-MC**	**μ-MC** $\mu = 0.16$
10	3.4	4.9	4.9	**5.0**	**5.0**
5	-0.2	2.0	2.0	**2.2**	**2.0**
0	-3.4	-1.1	-1.1	**-0.8**	**-1.0**
-5	-6.0	-4.1	-4.1	**-3.6**	**-3.9**
-10	-7.9	-6.6	-6.6	**-6.1**	**-6.2**

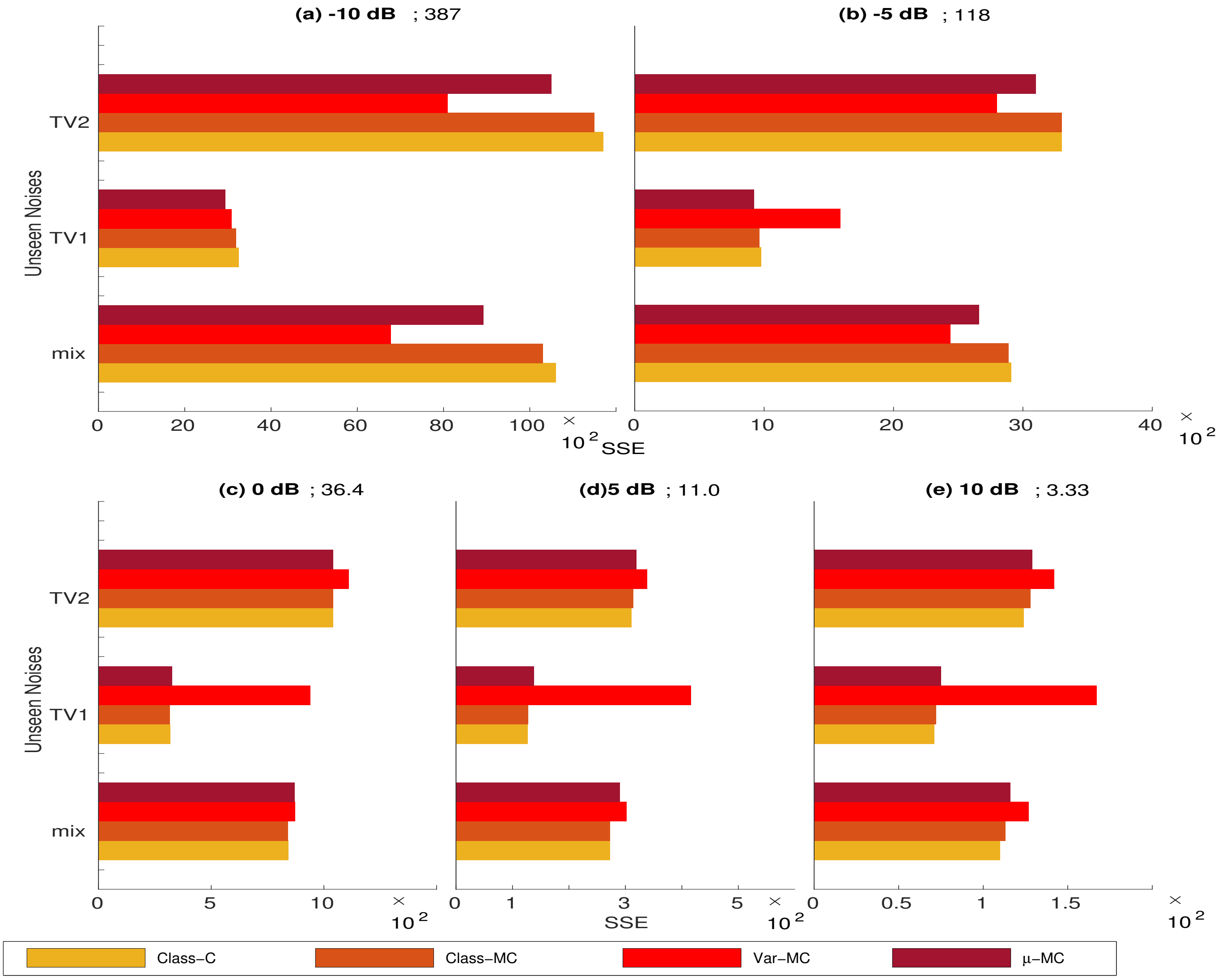


Figure 4.9: **Results on non-stationary, time-varying noises:** Performance comparison in terms of SSE (sum squared error) of Var-MC and μ-MC algorithms with class-C and class-MC for mix, TV1 and TV2 cases at SNRs (a) -10 dB (b) -5 dB (c) 0 dB (d) 5 dB and (e) 10 dB averaged over 50 files randomly selected from TIMIT test set. The noisy speech SSE values averaged over all the three noises for (a) -10 dB: 387×10^2 (b) -5 dB: 118×10^2 (c) 0 dB: 36.4×10^2 (d) 5 dB: 11.0×10^2 (e) 10 dB: 3.33×10^2 (These noisy speech SSE values are also shown along with each SNR in the plots; the scaling factor of $\times 10^2$ is omitted).
mix: mixture of unseen noises, factory1 and pink; **TV1**: the given speech waveform is divided into three segments and white, factory2 and factory 1 noises are added to the different segments; **TV2**: each test utterance of duration 2 to 3 sec. is divided into a random number (5 to 10) of segments of random length and unseen noises white, factory1 and pink are added randomly to these segments

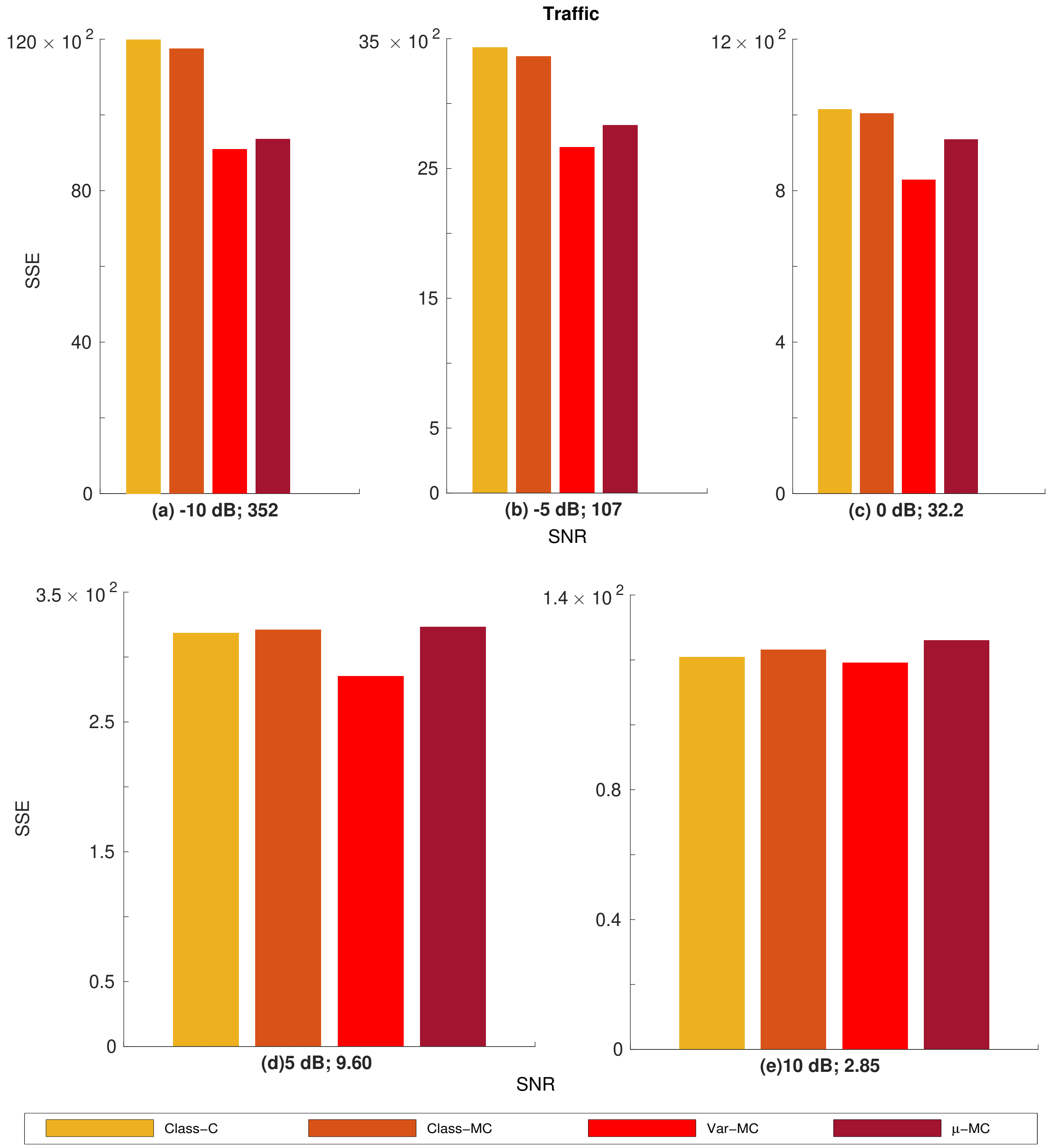


Figure 4.10: **Results on real world, traffic noise:** Performance comparison in terms of SSE (sum squared error) of Var-MC and μ-MC algorithms with class-C and class-MC for speech corrupted with real world, traffic noise at SNRs (a) -10 dB (b) -5 dB (c) 0 dB (d) 5 dB and (e) 10 dB averaged over 50 files randomly selected from TIMIT test set. The noisy speech SSE values are (a) -10 dB: 352×10^2 (b) -5 dB: 107×10^2 (c) 0 dB: 32.2×10^2 (d) 5 dB: 9.60×10^2 (e) 10 dB: 2.85×10^2 (These noisy speech SSE values are also shown along with each SNR in the plots; the scaling factor of $\times 10^2$ is omitted).

4.6.7 Impact on computational complexity

We conducted experiments by adding MC dropout between various layers of the DNN. Our experiments show that the addition of MC dropout just before the final layer improves the enhancement performance [120]. Hence the computational impact of using MC dropout is restricted to the time required for forward passes for the final layer alone as the rest of the layers are deterministic and can be shared. Consequently, the net impact on computational complexity of using MC dropout is minimal, since the additional time required for drawing the stochastic samples is marginal compared to the baseline model with conventional dropout.

4.7 Conclusions

We have proposed different techniques that use dropout as a Bayesian estimator for DNN models for speech enhancement, to improve their generalizability. In an initial set of experiment, we show that replacing a single DNN with conventional dropout, trained on multiple noises, with MC dropout helps in improving the enhancement performance of the model on speech with unseen noises. We also propose the application of the inherent uncertainty (predictive variance) of MC dropout models as an estimate of squared error, for frame-wise selection of one out of multiple noise-specific DNN models (Var-MC). The algorithm shows performances superior to a classifier-based model selection scheme for unseen noise cases.

We devise a method based on a threshold μ to switch between a noise classifier-based model selection and predictive variance-based model selection (μ-MC) to compensate for the poor performance of Var-MC compared to classifier-based model selection schemes for seen noises. We find that this method gives better enhancement performance than the classifier-based model selection for unseen noises at the same time giving comparable performances for the case of seen noises. The algorithms are also observed to be useful for scenarios where the speech signal is corrupted with non-stationary noises. This includes the case, where the speech is corrupted with a mixture of various noises and also where random number of segments of random lengths of speech get corrupted by different randomly chosen noises. This is the closest we can go in testing a model for its effectiveness on non-stationary noise, while still having the ability to evaluate its effectiveness, due to the availability of ground truth. In another significant result, we show the performance of the algorithms on speech corrupted by real world, traffic noise. This points towards the potential application of the algorithm in realistic scenarios. This work shows the effectiveness of MC dropout over standard dropout models and hence could be implemented on any state of the art system employing dropout.

Chapter 5

Conclusion and future work

In this thesis, we analyzed various speech-sound class-specific and noise-specific enhancement approaches and frame-wise selection methods for class-specific and noise-specific models.

5.1 Conclusion

In Chapter 2, we have analyzed the performance of our enhancement scheme, where we use various speech-sound class-specific dictionaries to enhance noisy speech. We have observed that even though in terms of objective quality measures such as PESQ and SSNR, the performance is not so promising in most noise cases compared to class-independent case, we obtain significant performance improvements in terms of phoneme recognition accuracy. To select the appropriate class dictionary for a frame, we have used the approximate labels obtained from an ASR, whose input is the speech enhanced using a class-independent dictionary. We have analyzed the performance using manner of articulation (MOA), place of articulation (POA) and phoneme-specific dictionaries. The phoneme-specific dictionary based enhancement outperforms the MOA and POA based schemes in most of the cases.

In Chapter 3, we have proposed a joint enhancement-decoding (JED) algorithm to overcome the dictionary selection errors in our class-specific scheme due to the errors in the estimated labels. N enhanced observations for each frame can be fed into the JED algorithm which then chooses the best observation that maximizes the overall likelihood to obtain the recognized labels. We have analyzed the phoneme recognition performance of JED for N varying from 1 to 5. The recognition performance varies with N, giving the best values at $N = 2$ or 3 in most cases. The best-N labels could be selected based on a monogram, bigram or trigram confusion matrix though we found that the use of bigram and trigram confusion matrices does not result in any marked improvement over monogram case.

In Chapter 4, we have proposed different techniques that use dropout as a Bayesian estimator for DNN models for speech enhancement, to improve their generalizability. The inherent uncertainty (predictive variance) of Monte Carlo dropout models is used as an estimate of squared error, for frame-wise selection of one out of multiple DNN models. For unseen noise scenarios, the above scheme of noise model selection (Var-MC) gives superior performance compared to a DNN classifier-based noise model selection scheme. However, the performance dropped for the case of seen noises compared to classifier-based selection scheme. To compensate for this performance drop, we have devised a method based on a threshold μ to switch between a noise-classifier-based model selection and predictive variance-based model selection (μ-MC). The μ-MC algorithm is found to be useful for unseen noises at the same time giving comparable performance to that of classifier-based scheme for seen noises. The algorithms are also observed to be useful for scenarios where the speech signal is corrupted with non-stationary noises. This includes the case, where the speech is corrupted with a mixture of various noises and also where random number of segments of random lengths of speech get corrupted by different randomly chosen noises. We also show that the algorithms give performances superior to classifier-based scheme in a real world, traffic noise scenario. We have also shown that replacing a single DNN with conventional dropout, trained on multiple noises, with MC dropout helps in improving the enhancement performance of the model on speech with unseen noises.

5.2 Future scope

In future we would like to extend our class-specific scheme using a DNN framework and explore the usefulness of other features such as multi-stream features [100]. In the case of JED algorithm we would like to explore the scenario where the multiple inputs to the algorithm are enhanced using techniques other than dictionary-based. Depending on the noise type and SNR, any other enhancement scheme can be used. We would also like to implement the JED algorithm in a DNN-based recognition framework. Even though the μ-MC algorithm compensates for the performance drop of Var-MC algorithm for seen noise cases, the algorithm does not perform as good as Var-MC, for unseen noise case. We would like to address this issue in future and find a selection criterion to further optimize the μ-MC algorithm.

Bibliography

[1] Jordan Novet. Look inside Microsoft's anechoic chamber, officially the quietest place on earth. 1 October 2015. 1

[2] Philipos C Loizou. *Speech enhancement: theory and practice*. CRC press, 2007. 1, 4, 6, 8, 9

[3] Harald Gustafsson. *Speech enhancement for mobile communications*. PhD thesis, 2000. 1

[4] Nils Westerlund. *Applied speech enhancement for personal communication*. PhD thesis, Blekinge Institute of Technology, 2003.

[5] David Yuheng Zhao. *Model based speech enhancement and coding*. PhD thesis, KTH, 2007. 1

[6] Harry Levitt. Noise reduction in hearing aids: A review. *Journal of rehabilitation research and development*, 38(1):111–122, 2001. 2

[7] José I Alcántara, Brian CJ Moore, Volker Kühnel, and Stefan Launer. Evaluation of the noise reduction system in a commercial digital hearing aid: Evaluación del sistema de reducción de ruido en un auxiliar auditivo digital comercial. *International Journal of Audiology*, 42(1):34–42, 2003. 2

[8] Jinqiu Sang, Hongmei Hu, Chengshi Zheng, Guoping Li, Mark E Lutman, and Stefan Bleeck. Evaluation of a sparse coding shrinkage algorithm in normal hearing and hearing impaired listeners. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1074–1078. IEEE, 2012. 2

[9] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. 2

[10] Kuldip K Paliwal, James G Lyons, Stephen So, Anthony P Stark, and Kamil K Wójcicki. Comparative evaluation of speech enhancement methods for robust automatic speech recognition. In *4th Int. Conf. on Signal Processing and Communication Systems*, 2010. 3, 23, 53, 81

[11] P M Nazreen, A G Ramakrishnan, and Prasanta Kumar Ghosh. A class-specific speech enhancement for phoneme recognition: A dictionary learning approach. *Proc. Interspeech*, pages 3728–3732, 2016. 3, 14, 53, 54, 85, 89, 95

[12] Mark R Weiss, Ernest Aschkenasy, and Thomas W Parsons. Study and development of the intel technique for improving speech intelligibility. Technical report, Nicolet Scientific Corp Northvale NJ, 1975. 4

[13] Steven Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on acoustics, speech, and signal processing*, 27(2):113–120, 1979. 4

[14] Jeffery J Faneuff and D Richard Brown III. Noise reduction and increased vad accuracy using spectral subtraction. In *Processing of the Global Signal Processing Exposition and International Signal Processing Conference (ISPC'03). Dallas, Texas*, 2003. 4

[15] Haitian Xu, Zheng-Hua Tan, Paul Dalsgaard, and Borge Lindberg. Spectral subtraction with full-wave rectification and likelihood controlled instantaneous noise estimation for robust speech recognition. In *Eighth International Conference on Spoken Language Processing*, 2004. 4

[16] Michael Berouti, Richard Schwartz, and John Makhoul. Enhancement of speech corrupted by acoustic noise. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 208–211. IEEE, 1979. 5

[17] Zenton Goh, Kah-Chye Tan, and TG Tan. Postprocessing method for suppressing musical noise generated by spectral subtraction. *IEEE Transactions on speech and audio processing*, 6(3):287–292, 1998.

[18] Sunil Kamath and Philipos Loizou. A multi-band spectral subtraction method for enhancing speech corrupted by colored noise. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 4164–4164. Citeseer, 2002. 30, 61, 69, 75

[19] Yang Lu and Philipos C Loizou. A geometric approach to spectral subtraction. *Speech communication*, 50(6):453–466, 2008. 5, 14, 30, 61, 69, 75, 83

[20] Norbert Wiener and Mass. Massachusetts Institute of Technology (Cambridge). *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. Technology Press, 1950. 5

[21] Pascal Scalart et al. Speech enhancement based on a priori signal to noise estimation. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 629–632. IEEE, 1996. 5

[22] Jae Soo Lim and Alan V Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, 67(12):1586–1604, 1979. 5

[23] TV Sreenivas and Pradeep Kirnapure. Codebook constrained wiener filtering for speech enhancement. *IEEE Transactions on speech and audio processing*, 4(5):383–389, 1996. 5

[24] Marvin Sambur. Adaptive noise canceling for speech signals. *IEEE Transactions on acoustics, speech, and signal processing*, 26(5):419–423, 1978. 5

[25] Jingdong Chen, Jacob Benesty, Yiteng Huang, and Simon Doclo. New insights into the noise reduction wiener filter. *IEEE Transactions on audio, speech, and language processing*, 14(4):1218–1234, 2006. 5

[26] Asmaa Amehraye, Dominique Pastor, and Ahmed Tamtaoui. Perceptual improvement of wiener filtering. In *International Conference on Acoustics, Speech and Signal Processing*, pages 2081–2084. IEEE, 2008. 5

[27] Fei Chen and Philipos C Loizou. Speech enhancement using a frequency-specific composite wiener function. In *International Conference on Acoustics, Speech and Signal Processing*, pages 4726–4729. IEEE, 2010. 5

[28] Robert McAulay and Marilyn Malpass. Speech enhancement using a soft-decision noise suppression filter. *IEEE Transactions on acoustics, speech, and signal processing*, 28(2): 137–145, 1980. 5

[29] Yariv Ephraim and David Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on acoustics, speech, and signal processing*, 32(6):1109–1121, 1984. 6, 83

[30] Yariv Ephraim and David Malah. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE Transactions on acoustics, speech, and signal processing*, 33(2):443–445, 1985. 6

[31] Guo-Hong Ding, Taiyi Huang, and Bo Xu. Suppression of additive noise using a power spectral density mmse estimator. *IEEE Signal processing letters*, 11(6):585–588, 2004. 6

[32] Sriram Srinivasan, Jonas Samuelsson, and W Bastiaan Kleijn. Codebook-based bayesian speech enhancement for nonstationary environments. *IEEE Transactions on audio, speech, and language processing*, 15(2):441–452, 2007. 6, 7

[33] Yoshihisa Uemura, Yu Takahashi, Hiroshi Saruwatari, Kiyohiro Shikano, and Kazunobu Kondo. Musical noise generation analysis for noise reduction methods based on spectral subtraction and MMSE STSA estimation. In *International Conference on Acoustics, Speech and Signal Processing*, pages 4433–4436. IEEE, 2009. 6

[34] Saeed Gazor and Wei Zhang. Speech enhancement employing laplacian-gaussian mixture. *IEEE Transactions on speech and audio processing*, 13(5):896–904, 2005. 6

[35] Jan S Erkelens, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. Minimum mean-square error estimation of discrete fourier coefficients with generalized gamma priors. *IEEE Transactions on audio, speech, and language processing*, 15(6):1741–1752, 2007. 6, 83

[36] Thomas Lotter and Peter Vary. Speech enhancement by map spectral amplitude estimation using a super-gaussian speech model. *EURASIP Journal on Advances in Signal Processing*, 2005. 6

[37] Philipos C Loizou. Speech enhancement based on perceptually motivated bayesian estimators of the magnitude spectrum. *IEEE Transactions on speech and audio processing*, 13(5):857–869, 2005. 6, 22

[38] Israel Cohen. Speech enhancement using a noncausal a priori SNR estimator. *Signal Processing Letters, IEEE*, 11(9):725–728, 2004. 6, 30, 61, 69, 75

[39] Cyril Plapous, Claude Marro, and Pascal Scalart. Improved signal-to-noise ratio estimation for speech enhancement. *IEEE Transactions on audio, speech, and language processing*, 14(6):2098–2108, 2006. 6, 30, 61, 69, 75

[40] Markos Dendrinos, Stelios Bakamidis, and George Carayannis. Speech enhancement from noise: A regenerative approach. *Speech Communication*, 10(1):45–57, 1991. 6

[41] Søren Holdt Jensen, Per Christian Hansen, Steffen Duus Hansen, and John Aasted Sorensen. Reduction of broad-band noise in speech by truncated qsvd. *IEEE Transactions on speech and audio processing*, 3(6):439–448, 1995. 6

[42] Simon Doclo, Ioannis Dologlou, and Marc Moonen. A novel iterative signal enhancement algorithm for noise reduction in speech. In *Fifth International Conference on Spoken Language Processing*, 1998.

[43] Simon Doclo and Marc Moonen. Gsvd-based optimal filtering for single and multimicrophone speech enhancement. *IEEE Transactions on signal processing*, 50(9):2230–2244, 2002. 6

[44] Yariv Ephraim and Harry L Van Trees. A signal subspace approach for speech enhancement. *IEEE Transactions on speech and audio processing*, 3(4):251–266, 1995. 6

[45] Jun Huang and Yunxin Zhao. An energy-constrained signal subspace method for speech enhancement and recognition in white and colored noises. *Speech Communication*, 26(3): 165–181, 1998. 7

[46] Udar Mittal and Nam Phamdo. Signal/noise KLT based approach for enhancing speech degraded by colored noise. *IEEE Transactions on speech and audio processing*, 8(2): 159–167, 2000.

[47] Afshin Rezayee and Saeed Gazor. An adaptive KLT approach for speech enhancement. *IEEE Transactions on speech and audio processing*, 9(2):87–95, 2001.

[48] Yi Hu and Philipos C Loizou. A generalized subspace approach for enhancing speech corrupted by colored noise. *IEEE Transactions on speech and audio processing*, 11(4): 334–341, 2003.

[49] Hanoch Lev-Ari and Yariv Ephraim. Extension of the signal subspace speech enhancement approach to colored noise. *IEEE Signal Processing Letters*, 10(4):104–106, 2003. 7

[50] Yariv Ephraim. A Bayesian estimation approach for speech enhancement using hidden Markov models. *IEEE Transactions on signal processing*, 40(4):725–735, 1992. 7, 83

[51] Hossein Sameti, Hamid Sheikhzadeh, Li Deng, and Robert L Brennan. HMM-based strategies for enhancement of speech signals embedded in nonstationary noise. *IEEE Transactions on speech and audio processing*, 6(5):445–455, 1998. 7, 83

[52] Achintya Kundu, Saikat Chatterjee, A Sreenivasa Murthy, and TV Sreenivas. GMM based bayesian approach to speech enhancement in signal/transform domain. In *International Conference on Acoustics, Speech and Signal Processing*, pages 4893–4896. IEEE, 2008. 7

[53] Sriram Srinivasan, Jonas Samuelsson, and W Bastiaan Kleijn. Codebook driven short-term predictor parameter estimation for speech enhancement. *IEEE Transactions on audio, speech, and language processing*, 14(1):163–176, 2006. 7, 83

[54] Kevin W Wilson, Bhiksha Raj, and Paris Smaragdis. Regularized non-negative matrix factorization with temporal dependencies for speech denoising. In *Ninth Annual Conference of the International Speech Communication Association*, 2008. 7

[55] Nasser Mohammadiha, Timo Gerkmann, and Arne Leijon. A new linear mmse filter for single channel speech enhancement based on nonnegative matrix factorization. In *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 45–48. IEEE, 2011.

[56] Gautham J Mysore and Paris Smaragdis. A non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 17–20. IEEE, 2011.

[57] Nasser Mohammadiha and Arne Leijon. Nonnegative hmm for babble noise derived from speech hmm: Application to speech enhancement. *IEEE Transactions on audio, speech, and language processing*, 21(5):998–1011, 2013.

[58] Nasser Mohammadiha, Paris Smaragdis, and Arne Leijon. Supervised and unsupervised speech enhancement using nonnegative matrix factorization. *IEEE Transactions on audio, speech, and language processing*, 21(10):2140–2151, 2013. 7

[59] Christian D Sigg, Tomas Dikk, and Joachim M Buhmann. Speech enhancement using generative dictionary learning. *IEEE Transactions on audio, speech, and language processing*, 20(6):1698–1712, 2012. 7, 11, 14, 16, 17, 21, 55, 81

[60] Shinichi Tamura. An analysis of a noise reduction neural network. In *Acoustics, Speech, and Signal Processing, ICASSP, International Conference on*, pages 2001–2004. IEEE, 1989. 7, 83

[61] Fei Xie and Dirk Van Compernolle. A family of MLP based nonlinear spectral estimators for noise reduction. In *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, volume 2, pages II–53. IEEE, 1994. 83

[62] Eric A Wan and Alex T Nelson. Networks for speech enhancement. *Handbook of neural networks for speech processing. Artech House, Boston, USA*, 139:1, 1999. 7, 83

[63] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 7, 83

[64] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 83

[65] Andrew L Maas, Quoc V Le, Tyler M O'Neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y Ng. Recurrent neural networks for noise reduction in robust ASR. In *13th Annual Conf. , International Speech Communication Association*, 2012. 83

[66] Yuxuan Wang and DeLiang Wang. Towards scaling up classification-based speech separation. *IEEE Transactions on audio, speech, and language processing*, 21(7):1381–1390, 2013. 83

[67] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. An experimental study on speech enhancement based on deep neural networks. *IEEE Signal processing letters*, 21(1):65–68, 2014. 83

[68] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Transactions on audio, speech and language processing (TASLP)*, 23(1):7–19, 2015. 7, 83, 87

[69] Yi Hu and Philipos C Loizou. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on audio, speech, and language processing*, 16(1):229–238, 2008. 7, 14, 22, 95

[70] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE, 2001. 8

[71] ITU-T Recommendation. Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. *Rec. ITU-T P. 862*, 2001. 8

[72] Patricia Scanlon, Daniel PW Ellis, and Richard B Reilly. Using broad phonetic group experts for improved speech recognition. *IEEE Transactions on audio, speech, and language processing*, 15(3):803–812, 2007. 10, 14, 21

[73] The International Phonetic Alphabet. revised to 2005. 10, 14, 21

[74] Ramya Rasipuram et al. Multitask learning to improve articulatory feature estimation and phoneme recognition. Technical report, Idiap, 2011. 10, 14, 21

[75] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993. 11

[76] Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers. Record of The Twenty-Seventh Asilomar Conf.*, pages 40–44. IEEE, 1993. 11, 16

[77] Irina F Gorodnitsky and Bhaskar D Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *IEEE Transactions on signal processing*, 45(3):600–616, 1997. 11

[78] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001. 11

[79] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004. 11, 16

[80] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1. *Vision research*, 37(23):3311–3325, 1997. 11

[81] Michael S Lewicki and Bruno A Olshausen. Probabilistic framework for the adaptation and comparison of image codes. *JOSA A*, 16(7):1587–1601, 1999. 11

[82] Adam Coates and Andrew Y Ng. Learning feature representations with k-means. In *Neural networks: Tricks of the trade*, pages 561–580. Springer, 2012. 11

[83] Kenneth Kreutz-Delgado, Joseph F Murray, Bhaskar D Rao, Kjersti Engan, Te-Won Lee, and Terrence J Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2):349–396, 2003. 11

[84] J.A. TROPP. *Topics in sparse approximation*. PhD thesis, The University of Texas at Austin, 2004. 11

[85] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006. 11, 17, 55

[86] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *CS Technion*, 40(8):1–15, 2008. 11, 17, 20

[87] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016. 12, 84, 85, 86, 88

[88] Jort F Gemmeke, Tuomas Virtanen, and Antti Hurmalainen. Exemplar-based sparse representations for noise robust automatic speech recognition. *IEEE Transactions on audio, speech, and language processing*, 19(7):2067–2080, 2011. 14

[89] Emre Yilmaz, Jori F Gemmeke, et al. Noise-robust speech recognition with exemplar-based sparse representations using alpha-beta divergence. In *Acoustics, Speech and Signal Processing (ICASSP), IEEE Int. Conf.*, pages 5502–5506, 2014. 14

[90] Bhiksha Raj, Rita Singh, and Tuomas Virtanen. Phoneme-dependent NMF for speech enhancement in monaural mixtures. In *INTERSPEECH*, pages 1217–1220, 2011. 14, 53

[91] Zhong-Qiu Wang, Yan Zhao, and DeLiang Wang. Phoneme-specific speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016. 15, 53, 85, 89

[92] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 16

[93] Parya MomayyezSiahkal. *Integration of multiple feature sets for reducing ambiguity in automatic speech recognition*. PhD thesis, McGill University, 2008. 20, 21

[94] Kai-Fu Lee and Hsiao-Wuen Hon. Speaker-independent phone recognition using hidden markov models. *IEEE Transactions on acoustics, speech and signal processing*, 37(11): 1641–1648, 1989. 20, 21

[95] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon Technical Report N*, 93:27403, February 1993. 21, 22, 23, 95

[96] Andrew Varga and Herman J. M. Steeneken. Assessment for automatic speech recognition ii: Noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Commun.*, 12(3):247–251, July 1993. ISSN 0167-6393. 21

[97] Steve J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book Version 3.4*. Cambridge University Press, 2006. 21, 60

[98] Steven B Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on acoustics, speech and signal processing*, 28(4):357–366, 1980. 21, 60

[99] Yann Soon, Soo Ngee Koh, and Chai Kiat Yeo. Noisy speech enhancement using discrete cosine transform. *Speech communication*, 24(3):249–257, 1998. 22

[100] Sridhar Krishna Nemala, Kailash Patil, and Mounya Elhilali. A multistream feature framework based on bandpass modulation filtering for robust speech recognition. *IEEE Transactions on audio, speech, and language processing*, 21(2):416–426, 2013. 51, 109

[101] Mark JF Gales and Steve J Young. Robust continuous speech recognition using parallel model combination. *IEEE Transactions on speech and audio processing*, 4(5):352–359, 1996. 53

[102] Ding Pei and Cao Zhigang. An efficient robust automatic speech recognition system based on the combination of speech enhancement and log-add HMM adaptation. In *Info-tech and Info-net. Proceedings. ICII Beijing. Int. Conf.*, volume 3. IEEE, pages 367–371, 2001. 53

[103] Li Deng, Jasha Droppo, and Alex Acero. Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion. *IEEE Transactions on speech and audio processing*, 13(3):412–421, 2005.

[104] Jinyu Li, Li Deng, Dong Yu, Yifan Gong, and Alex Acero. A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions. *Computer Speech & Language*, 23(3):389–405, 2009. 53

[105] Sadaoki Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on acoustics, speech and signal processing*, 29(2):254–272, 1981. 53

[106] Hynek Hermansky and Nelson Morgan. RASTA processing of speech. *IEEE Transactions on speech and audio processing*, 2(4):578–589, 1994. 53

[107] Pedro J Moreno, Bhiksha Raj, and Richard M Stern. A vector Taylor series approach for environment-independent speech recognition. In *Acoustics, Speech, and Signal Processing. ICASSP. Proceedings. IEEE Int. Conf.*, volume 2, pages 733–736, 1996. 53

[108] Steven F Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on acoustics, speech signal processing*, 27(2):113–120, 1979. 53, 83

[109] Volker Stahl, Alexander Fischer, and Rolf Bippus. Quantile based noise estimation for spectral subtraction and Wiener filtering. In *Acoustics, Speech, and Signal Processing. ICASSP. Proceedings. IEEE Int. Conf.*, volume 3, pages 1875–1878, 2000. 53, 83

[110] PM Nazreen, AG Ramakrishnan, and Prasanta Kumar Ghosh. A joint enhancement-decoding formulation for noise robust phoneme recognition. In *14th IEEE India Council International Conference (INDICON)*, pages 1–6, 2017. 53, 85, 89

[111] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967. 53

[112] B Yegnanarayana, Carlos Avendano, Hynek Hermansky, and P Satyanarayana Murthy. Speech enhancement using linear prediction residual. *Speech communication*, 28(1):25–42, 1999. 83

[113] P Krishnamoorthy and SR Mahadeva Prasanna. Enhancement of noisy speech by temporal and spectral processing. *Speech Communication*, 53(2):154–174, 2011.

[114] Wen Jin and Michael S Scordilis. Speech enhancement by residual domain constrained optimization. *Speech Communication*, 48(10):1349–1364, 2006. 83

[115] Rainer Martin. Speech enhancement based on minimum mean-square error estimation and supergaussian priors. *IEEE Transactions on speech and audio processing*, 13(5): 845–856, 2005. 83

[116] Zhiheng Ouyang, Hongjiang Yu, Wei-Ping Zhu, and Benoit Champagne. A deep neural network based harmonic noise model for speech enhancement. In *Proc. Interspeech*, pages 3224–3228, 09 2018. 83, 84

[117] Shuai Nie, Shan Liang, Bin Liu, Yaping Zhang, Wenju Liu, and Jianhua Tao. Deep noise tracking network: A hybrid signal processing/deep learning approach to speech enhancement. In *Proc. Interspeech*, pages 3219–3223, 2018. 84

[118] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. Improving deep neural networks for LVCSR using rectified linear units and dropout. In *Acoustics, Speech and Signal Processing (ICASSP), International Conference on*, pages 8609–8613. IEEE, 2013. 84, 88, 93

[119] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 84, 88, 93

[120] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *Robotics and Automation (ICRA), International Conference on*, pages 4762–4769. IEEE, 2016. 84, 85, 86, 97, 107

[121] PM Nazreen and AG Ramakrishnan. DNN based speech enhancement for unseen noises using Monte Carlo dropout. In *12th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–6. IEEE, 2018. 85

[122] P M Nazreen and A G Ramakrishnan. Using Monte Carlo dropout for non-stationary noise reduction from speech. *arXiv preprint arXiv:1808.09432 [eess.AS]*, 2018. 85

[123] Pavlos Papadopoulos, Andreas Tsiartas, and Shrikanth Narayanan. Long-term SNR estimation of speech signals in known and unknown channel conditions. *IEEE/ACM Transactions on audio, speech, and language processing*, 24(12):2495–2506, 2016. 85, 89

[124] Shlomo E Chazan, Sharon Gannot, and Jacob Goldberger. A phoneme-based pre-training approach for deep neural network with application to speech enhancement. In *IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 1–5. IEEE, 2016. 85, 89

[125] KV Vijay Girish, AG Ramakrishnan, and TV Ananthapadmanabha. Hierarchical classification of speaker and background noise and estimation of SNR using sparse representation. In *INTERSPEECH*, pages 2972–2976, 2016. 89

[126] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *Proc. ICLR*, pages 1–13, 2015. 93