一种致力于实时全频段语音增强的DSP+深度学习混合方法

Jean-Marc Valin, Mozilla Corporation, Mountain View, CA, USA, [jmvalin@jmvalin.ca](mailto:jmvalin@jmvalin.ca)

摘要 — 尽管噪声抑制是信号处理中的成熟领域, 但它仍然高度依赖于评估器算法和参数的微调。在本文中, 我们演示了一种DSP +深度学习混合方案来抑制噪声。我们致力于在保持尽可能低的复杂度的同时, 仍要实现高质量的增强语音。具有四个隐藏层的深层递归神经网络用于估计关键频带的最优增益, 而更传统的基音滤波器则衰减音调谐波之间的噪声。与传统的最小均方误差(MMSE)频谱估计器相比, 该方法可实现更高的质量, 同时将复杂度保持在足够低的水平, 以便在低功耗CPU上以48 kHz进行实时操作。

索引词 — 噪声抑制, 递归神经网络

# I.引言

至少从70年代开始, 噪声抑制一直是人们关注的话题。尽管质量有了显着提高, 但高层结构基本保持不变。某种形式的频谱估计技术依赖于噪声频谱估计器, 其本身由语音活动检测器(VAD)或类似算法驱动, 如图1所示。

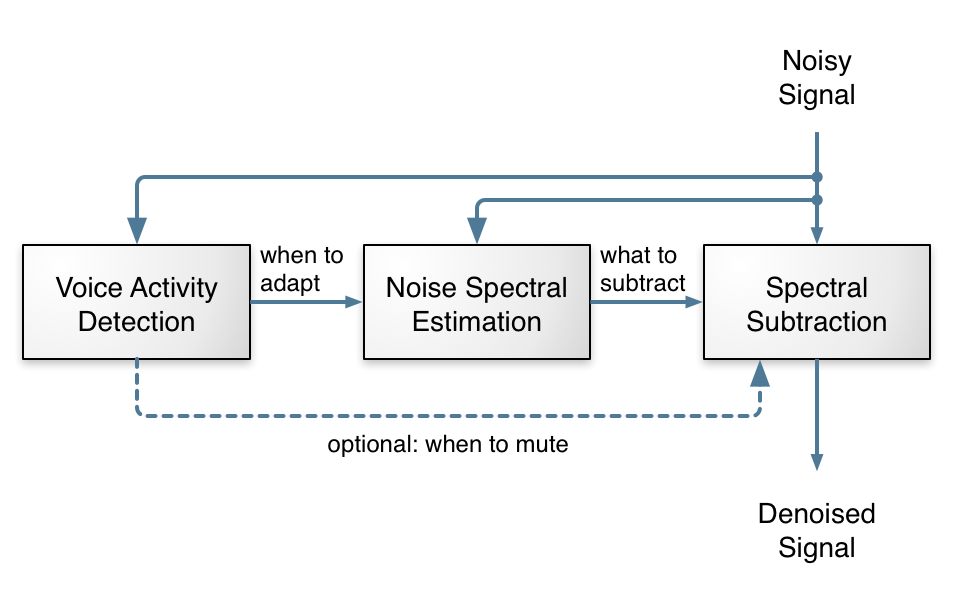


图1. 大多数噪声抑制算法的高层结构

这3个组件中的每一个都需要精确的估计器, 并且难以调整(tune)。例如, 粗放的初始噪声估计器和基于频谱减法的频谱估计器[1]已被更准确的噪声估计器[2], [3]和频谱幅度估计器[4]取代。尽管有改进, 但这些估计器仍然难以设计, 并且需要大量的人工调整工作。因此, 深度学习技术的最新进展吸引了噪声抑制领域的注意。深度学习技术已经用于噪声抑制[5], [6], [7], [8], [9]。许多提议的方法都针对不需要低等待时间的自动语音识别(ASR)应用。而且, 在许多情况下, 神经网络的规模太庞大, 这使得在没有GPU时要做到实时很困难。

在本方法中, 我们专注于低复杂度的实时应用程序(例如视频会议)。我们还专注于全频带(48 kHz) 语音。为了实现这些目标, 我们选择了一种混合方法(第二部分), 该方法依靠可靠的信号处理技术并使用深度学习(第三部分)来替代传统上难以正确调整的估计量。该方法与所谓的端到端系统形成对比, 在端到端系统中, 大多数或所有信号处理操作都被机器学习取代。这些端到端系统已经清楚地展示了深度学习的功能, 但是它们通常以显著增加复杂性为代价。我们表明, 所提出的方法具有可接受的复杂性 (第IV节), 并且比更常规的方法 (第V节) 具有更好的质量。我们在第VI节中总结了进一步改进此方法的方向。

# II. 信号模型

我们提出了一种混合方法来抑制噪声。目标是将深度学习用于需要仔细调整的噪声抑制方面, 而将基本信号处理构建块用于不需要 (深度学习) 的部分。主处理循环基于具有50％重叠(偏移为10 ms)的20 ms窗口。分析和综合都使用相同的Vorbis窗[10], 它满足Princen-Bradley准则[11]。该窗口定义为

(1)

其中N是窗口长度。该系统的信号级框图如图2所示。大部分抑制是先使用递归神经网络(RNN)计算增益, 然后在低分辨率频谱包络上执行此增益。这些增益是理想比率掩膜(IRM)的平方根。另一个更精细的抑制步骤是使用基音梳状滤波器来衰减基音谐波之间的噪声。

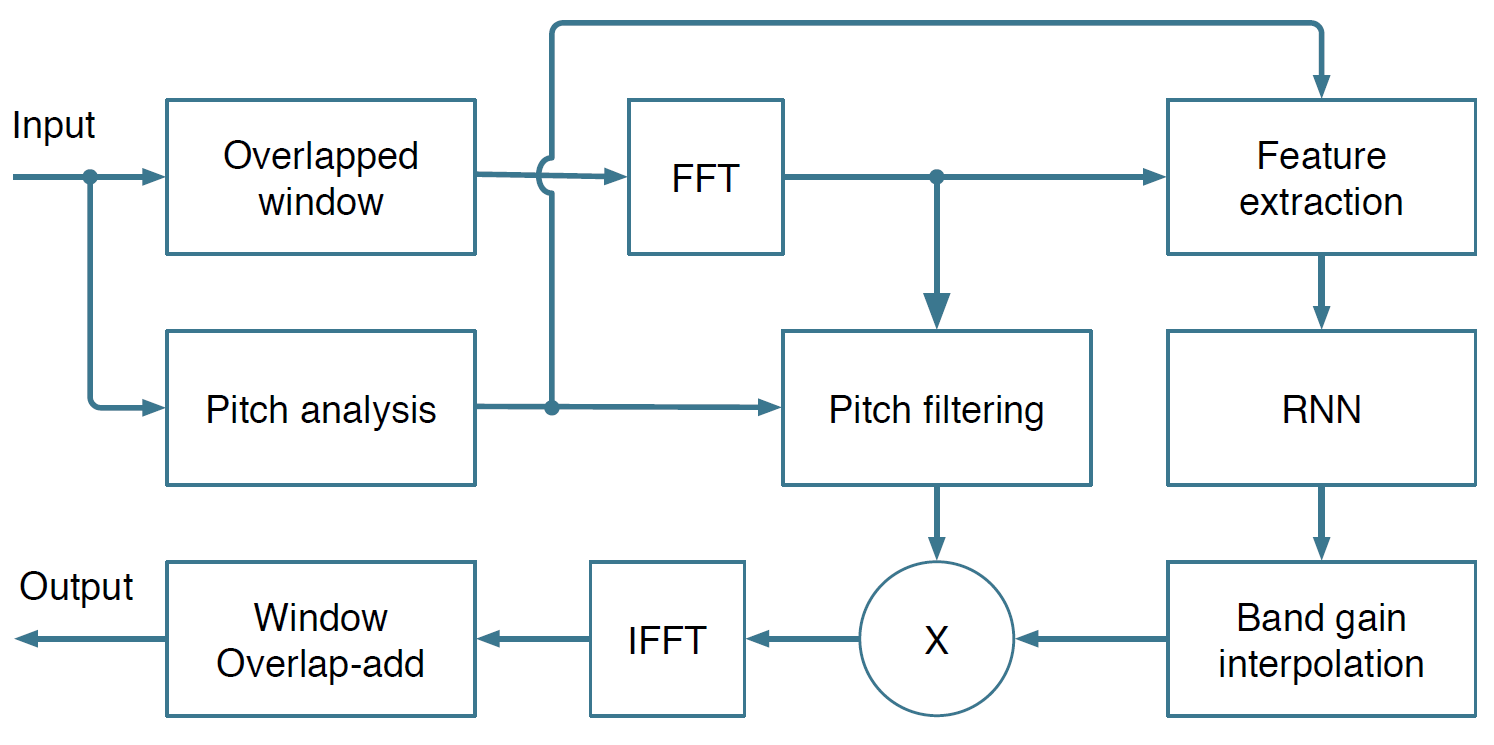


图2. 模块框图

## A. 频带结构

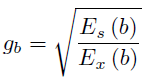
在[5]所采用的方法中, 神经网络被用来直接估计频仓(bin)的幅度, 并需要总共6144个隐藏单元和接近1000万个权重才能处理8 kHz语音。对应到帧长为20 ms的 48 kHz语音, 将需要一个具有400个输出 (0至20 kHz) 的网络, 其复杂性显然超出了我们的承受范围。

避免该问题的一种方法是假设语音和噪声的频谱包络足够平坦, 以使用比频率仓(frequency bin)更粗糙的分辨率。另外, 我们不直接估算频谱幅度, 而是估算关键频带的最优增益, 该增益具有限制在0和1之间的显着优势。Opus编解码器 [13] 中在使用巴克刻度 (Bark Scalre) [12] 的近似版本, 本文也采用了此频带划分。也就是说, 这些频带在高频段遵循巴克刻度, 但在低频段始终至少为4个频率仓(bin)。我们使用三角形窗而不是矩形窗, 峰值响应位于频带之间的边界。结果是总共有22个频带。因此, 我们的网络仅需输出22个范围在 [0, 1] 之间的值。

设wb(k)为频带b在频率k处的振幅, 令∑bwb(k)=1(归一化)。对于变换后的信号X(k), 频带中的能量由下式给出:

 (2)

每个频带增益定义为gb

 (3)

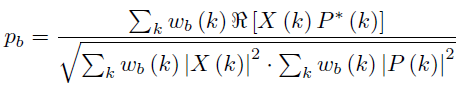
其中, Es(b)是纯净语音(标注数据, 基准数据)的能量, 而Ex(b)是输入语音(有噪声)的能量。考虑最优频带增益, 以下内插增益应用于每个频率仓 (bin):

 (4)

## B. 基音 (pitch) 滤波

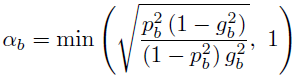
使用巴克(刻度)衍生频带来计算增益的主要缺点是我们无法在频谱中对更精细的细节建模。实际上, 这不利于抑制基音谐波之间的噪声。作为替代方案, 我们可以在基音周期使用梳状滤波器来消除谐波间噪声, 这与语音编解码器后置滤波器的工作方式相似[14]。因为语音信号的周期性严重依赖于频率(尤其是对于48 kHz采样率), 因此基音滤波器使用频带相关的滤波系数αb, 并在频域中处理。令P(k)为加时延的基音信号x(n-T)的加窗截取后的DFT, 通过计算X(k)+ αbP(k)进行滤波, 接着重整信号, 使每个频带的信号具有和原始信号X(k)相同的能量。

频带b的基音相关度定义为:

  (5)

其中表示复数值的实部, .\*表示复共轭。注意, 对于单个频带, (5) 等效于时域上的基音相关度。

推导滤波系数αb的最优值是困难的, 并且最小化均方误差(MMSE)的值在感知上并不是最佳的。而我们采用了基于以下约束和观察摸索出来的方法。由于噪声会导致基音相关性降低, 因此平均来说pb会小于gb, 因此对于任何具有pb≥gb的频带, 我们都使用αb = 1。因为当没有噪声时, 我们不想使信号失真, 所当gb = 1时, 我们使用αb =0。类似地, 当pb = 0时, 我们不需要增强基音, 因此αb=0。考虑所有这些约束以及它们之间的平滑行为, 使用以下表达式来计算滤波器系数：

  (6)

虽然我们在这里使用FIR基音滤波器, 也可以基于形式为的IIR基音滤波器来计算P(k), 这会更好的抑制(基音)谐波之间的噪音, 代价是失真会稍微增加。

## C. 特征提取

神经网络的输入和输出的频带数目应该相同, 而且应该包含噪音信号的对数谱。为了改善训练数据的条件, 我们将对数频谱进行DCT (离散余弦变换), 得到22个巴克频率倒谱系数 (BFCC) 。(特征集) 还包含前6个BFCC的时域上的一二阶导数。计算整个频带上基音相关度的DCT, 并将其前6个系数包括到特征集中。最后, 加上基音周期以及有助于语音检测的频谱非平稳性度量。总共使用42个输入特征(22+6x2+6+1+1)。与语音识别中通常使用的特征不同, 这里不使用倒谱均值归一化, 并且包含第一倒谱系数。这是有意的选择, 因为我们需要跟踪噪声的绝对水平, 这确实会使特征对信号的绝对幅度和信道频率响应敏感。这将在章节III-A中讨论。

# 三. 深层学习架构

神经网络紧密遵循噪声抑制算法的传统结构，如图3所示。该设计基于以下假设：三个循环层分别构成图1中的基本组成部分之一。在实践中，神经网络可以完全不是如我们所规划这样的承担这些功能 (并且可能在某种程度上确实如此）。它总共包括215个单元，4个隐藏层，最大的一层有96个单元。我们发现增加单元数量并不能显着提高噪声抑制的质量。但是，损失函数和我们构造训练数据的方式对最终质量有很大影响。我们发现门控循环单元（GRU）[15]在此任务上略胜于LSTM，同时也更简单。

尽管不是那么必要，但该网络仍包含VAD输出。额外的复杂性成本很小（额外增加了24个权重），并且这通过确保相关GRU确实学会了区分语音和噪声，改善了训练效果。

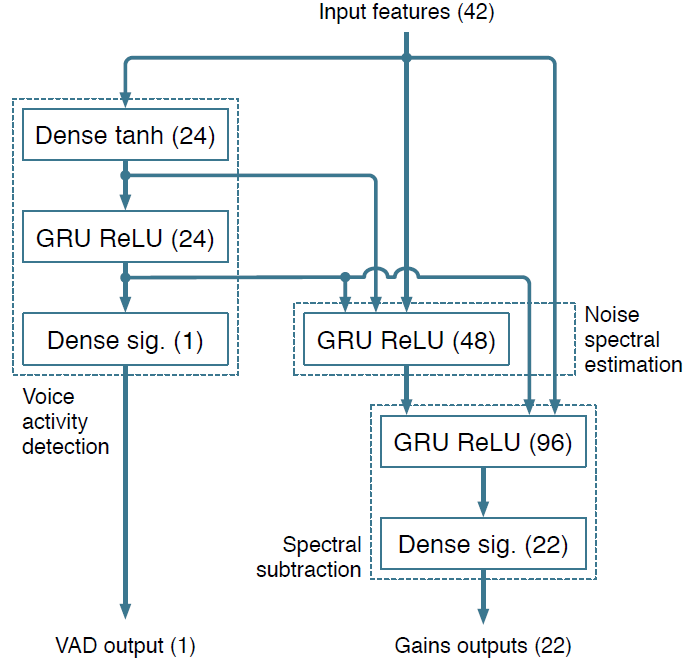
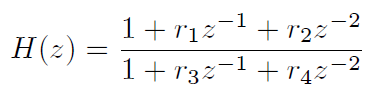


图3: 神经网络架构, 包括前馈网络, 全连接(稠密)层, 和循环层, 包括激活函数和每次的单元数

## A. 训练数据

因为标记数据既需要嘈杂的语音, 也需要纯净的语音，因此必须通过在纯净的语音数据中添加噪声来人工构建训练数据。对于语音数据，我们使用McGill TSP语音数据库(<http://www-mmsp.ece.mcgill.ca/Documents/Data/>, 法语和英语) 和用于语音测量的NTT多语言语音数据库(NTT Multi-Lingual Speech Database, 21种语言, 使用44.1 kHz音频CD轨道而不是16 kHz数据文件)。使用各种噪声源，包括计算机风扇，办公室，人群，飞机，汽车，火车，建筑。噪声以不同的级别混音以产生范围很宽的信噪比，包括干净的语音和纯噪声段。

由于我们不使用倒谱均值归一化，因此我们使用数据增强来使网络对频率响应的变化具有鲁棒性。这是通过使用以下形式的二阶滤波器针对每个训练数据独立过滤噪声和语音信号来实现的:

（7）

其中，r1 … r4中的都是均匀分布在 [-3/8, 3/8] 之间的随机值。对信号幅度的鲁棒性是通过改变混合信号的最终电平来实现的。我们总共有6个小时的语音和4个小时的噪声数据，通过使用各种增益和滤波器组合并将数据重新采样到40 kHz至54 kHz之间的频率，可以产生140个小时的嘈杂语音。

## B. 优化过程

在训练过程中, 因为无法精确确定正确的增益, 损失函数被用于判断衰减是过多了还是少了。尽管二值交叉熵函数在优化[0,1] 中的值用的很普遍, 但因为它与感知效果不匹配, 在获取合适增益值时结果并不好。我们使用以下损失函数进行训练:

 (8)

其中为增益估计, gb为相应的标记. 指数γ是一个感知参数，控制抑制噪声的程度。由于log(x)，因此最小化对数能量的均方误差，如果gain太小会导致抑制的太狠(which would make the suppression too aggressive given the lack of a floor on the gain)。实践中发现, γ=1/2是个比较合适的值，此时相当于在能量是1/4的情况下最小化均方误差(is equivalent to minimizing the mean squared error on the energy raised to the power 1/4)。有时, 在特定频段中可能没有噪音且没有语音, 比如当输入为无声时或在高频段上的信号被低通滤掉时。如果发生这种情况，则将标记数据的增益明确标记为undefined，并且忽略该增益的损失函数，以避免破坏训练过程。

对于神经网络的VAD输出，我们使用标准的交叉熵损失函数。训练在Tensorflow后台和Keras库上进行。

## C.增益平滑

当使用增益抑制噪声时，有时输出信号听起来很干，缺少预期的混响成分. 通过限制跨帧的衰减，可以轻松解决该问题。通过下式获取平滑增益:

（9）

其中是前一帧的滤波增益，衰减因子λ=0.6等于135 ms的混响时间。

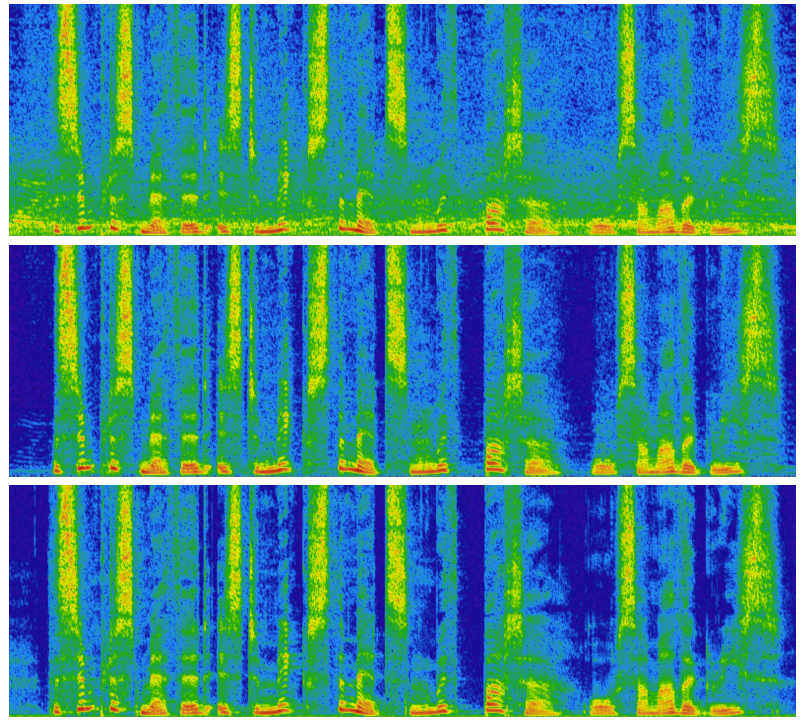


图4.在SNR为15 dB, 对嘈杂噪声 (babble Noise) 的噪声抑制示例语音的语谱图, (上：noisy，中：RNNoise，下：clean speech), 为了清楚起见，仅显示0-12 kHz频带。

# IV. 复杂度分析

为了使噪声抑制算法易于部署，希望保持大小和复杂度都较低。可执行文件的大小由代表神经网络中215个单位所需的87503个权重决定。为了使尺寸尽可能小，可以将权重量化为8位而不会降低性能。这样就可以将所有权重放入CPU的L2缓存中。由于每个权重在每帧的乘加运算中恰好使用一次，因此神经网络每帧需要175,000个浮点运算（我们将乘加算作两个运算），因此实时使用需要17.5 M flops。每帧的IFFT和两个FFT大约需要7.5 M flops，基音搜索（以12 kHz的频率运行）大约需要10 M flops。该算法的总复杂度约为40 M flops，与全频带语音编码器相当。该算法的非矢量C实现方式需要大约1.3％的单个x86内核（Haswell i7-4800MQ）来执行单个通道的48 kHz噪声抑制。在1.2 GHz ARM Cortex-A53内核（Raspberry Pi 3）上，相同浮点代码的实时复杂度为14％。作为比较，[9]中的16 kHz语音增强方法使用3个隐藏层，每个隐藏层具有2048个单位。这需要1250万的重量，导致1600 M flops的复杂度。即使量化为8位，其权重也不适合大多数CPU的缓存，其实时操作需要大约800 MB/s的内存带宽。

# V. 结果

我们使用训练集中未使用的语音和噪声数据测试噪声抑制的质量。我们将其与SpeexDSP库(https://www.speex.org/downloads/)中基于MMSE的噪声抑制器进行了比较。尽管噪声抑制工作在48 kHz，但由于宽带PESQ的局限性，必须将输出重新采样至16 kHz [16]。图5中的客观结果表明，通过使用深度学习（特别是对于非平稳噪声类型），质量得到了显着改善。通过随意聆听样品可以确认这种改进。图4展示了示例中的噪声抑制的效果。

在https://people.xiph.org/~jm/demo/rnnoise/上有所提到系统的交互式demo，其中包括Javascript版本的实时版本。在<https://github.com/xiph/rnnoise/> 上有所提到系统的软件实现, 基于BSD许可，产生的结果在hash码为91ef401的提交中。

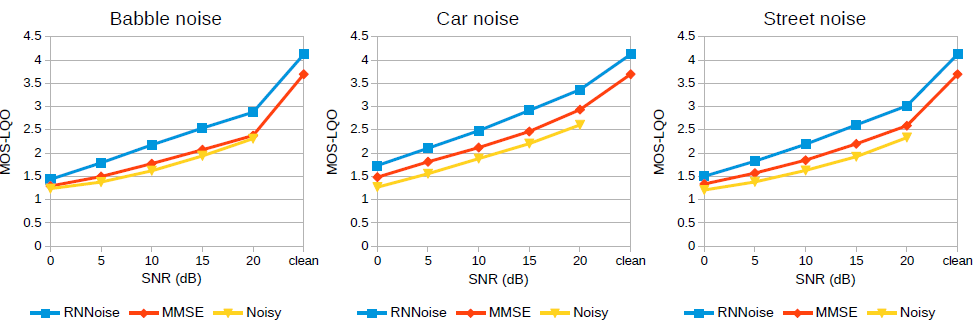


图5. PESQ MOS-LQO质量评估, 包括嘈杂(babble), 汽车(car)和街道(street)噪音场景

# VI. 结论

本文演示了一种噪声抑制方法，该方法将基于DSP的技术与深度学习相结合。通过仅将深度学习用于难以调节的噪声抑制方面，将问题简化为仅计算22个最优的关键频带增益，而且只需几个单元就可以有效地完成。然后，通过使用简单的基音滤波器解决频段的粗分辨率。所产生的低复杂度使该方法适用于移动或嵌入式设备，并且时延足够低，可用于视频会议系统。我们还证明，其质量明显高于基于纯信号处理的方法。我们认为，该技术可以轻松扩展到残留回声抑制，例如，通过将远端信号或滤波后的远端信号的倒频谱添加到输入特征中。同样，它应适用于麦克风阵列后过滤，方法是像[17]那样通过泄漏(leakage)估计来增强输入特征.

# 参考文献

[1] S. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” IEEE Transactions on acoustics, speech, and signal processing, vol. 27, no. 2, pp. 113–120, 1979.

[2] H.-G. Hirsch and C. Ehrlicher, “Noise estimation techniques for robust speech recognition,” in Proc. ICASSP, 1995, vol. 1, pp. 153–156.

[3] T. Gerkmann and R.C. Hendriks, “Unbiased MMSE-based noise power estimation with low complexity and low tracking delay,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no.4, pp. 1383–1393, 2012.

[4] Y. Ephraim and D. Malah, “Speech enhancement using a minimum mean-square error log-spectral amplitude estimator,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 33, no. 2, pp. 443–445, 1985.

[5] A. Maas, Q.V. Le, T.M. O’Neil, O. Vinyals, P. Nguyen, and A.Y. Ng, “Recurrent neural networks for noise reduction in robust ASR,” in Proc. INTERSPEECH, 2012.

[6] D. Liu, P. Smaragdis, and M. Kim, “Experiments on deep learning for speech denoising,” in Proc. Fifteenth Annual Conference of the International Speech Communication Association, 2014.

[7] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, “A regression approach to speech enhancement based on deep neural networks,” IEEE Transactions on Audio, Speech and Language Processing, vol. 23, no. 1, pp. 7–19, 2015.

[8] A. Narayanan and D. Wang, “Ideal ratio mask estimation using deep neural networks for robust speech recognition,” in Proc. ICASSP, 2013, pp. 7092–7096.

[9] S. Mirsamadi and I. Tashev, “Causal speech enhancement combining data-driven learning and suppression rule estimation.,” in Proc. INTERSPEECH, 2016, pp. 2870–2874.

[10] C. Montgomery, “Vorbis I specification,” 2004.

[11] J. Princen and A. Bradley, “Analysis/synthesis filter bank design based on time domain aliasing cancellation,” IEEE Tran. on Acoustics, Speech, and Signal Processing, vol. 34, no. 5, pp. 1153–1161, 1986.

[12] B.C.J. Moore, An introduction to the psychology of hearing, Brill, 2012.

[13] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos, “High-quality, low-delay music coding in the Opus codec,” in Proc. 135th AES Convention, 2013.

[14] Juin-Hwey Chen and Allen Gersho, “Adaptive postfiltering for quality enhancement of coded speech,” IEEE Transactions on Speech and Audio Processing, vol. 3, no. 1, pp. 59–71, 1995.

[15] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” in Proc. Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8), 2014.

[16] ITU-T, Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, 2001.

[17] J.-M. Valin, J. Rouat, and F. Michaud, “Microphone array post-filter for separation of simultaneous non-stationary sources,” in Proc. ICASSP, 2004.