Deep Learning Denoising Based Line Spectral Estimation

Yuan Jiang, Hongbin Li, *Fellow, IEEE*, and Muralidhar Rangaswamy, *Fellow, IEEE*

***Abstract*—Many well-known line spectral estimators may experience significant performance loss with noisy measurements. To address the problem, we propose a deep learning denoising based approach for line spectral estimation. The proposed approach utilizes a residual learning assisted denoising convolutional neural network (DnCNN) trained to recover the unstructured noise component, which is used to denoise the original measurements. Following the denoising step, we employ a popular model order selection method and a subspace line spectral estimator to the denoised measurements for line spectral estimation. Numerical results show that the proposed approach outperforms a recently introduced atomic norm minimization based denoising method and offers a substantial improvement compared with the line spectral estimation results obtained by directly applying the subspace estimator without denoising.**

***Index Terms*—line spectral estimation, signal denoising, deep learning.**

I. Introduction

LINE spectral estimation, involving recovering the frequencies, phases and amplitudes of a mixture of complex sinusoids from noisy samples, is a fundamental problem in statistical signal processing. Line spectral estimation problems arise in a variety of applications, such as direction of arrival estimation [1], [2], inverse scattering imaging [3], passive sensing [4], and many others. Compressed sensing (CS) based approaches, which have attracted much attention in recent years, can be employed to solve the line spectral estimation problem with several unique advantages (see, e.g., [5]); however, they may also suffer the so-called grid mismatch problem due to the use of a fixed discretized dictionary [6], [7]. Meanwhile, high-resolution spectral estimators based on subspace decomposition, e.g., MUSIC [8], exploit a low-rank structure of sinusoidal signals for line spectral estimation. Subspace methods are free of the grid mismatch problem, capable of offering accurate estimation result in low or moderate noise, but may degrade considerably with noisy measurements. Some recent works bridge the gap between discretized CS algorithms and subspace methods, by utilizing an atomic norm formulation [9], joint compressed sensing and dictionary refinement formulation [10]–[14], and others, although the computational complexity of these methods is usually high.

Another approach to line spectral estimation is based on denoising. An atomic norm minimization (ANM) approach was proposed in [15], which applies soft thresholding to the noise-corrupted measurements in the atomic norm for spectral recovery. Soft thresholding is a widely used denoiser for sparse signal recovery problems [16]. A denoiser is an algorithm that seeks to reduce noise or perturbation in the observed signal. To subtract the noise, a denoiser usually leverages some inherent structure, e.g., sparsity, low rank, etc., of the signal. These methods often need to solve a complex optimization problem and have a high computational complexity. In addition, their denoising performance in the presence of densely spaced frequencies is limited.

Aside from structure-based denoising approaches, data-driven denoising methods have been of significant interest in last few years. Rather than relying on prior knowledge of the data structure, deep convolutional neural networks can be trained to automatically capture the signal structure [17]. A denoising convolutional neural network (DnCNN) was introduced in [18] for image denoising, based on the so-called *residual learning*. It was found that, when the observation consists of a highly structured signal along with an unstructured noise, a deep network can be configured to remove the desired signal in a more efficient way than directly removing the undesired noise [19]. In essence, such networks perform denoising in an indirect manner, by first removing the signal to yield an estimate of the noise and then subtracting the estimated noise from the observation to get a denoised signal.

We propose herein a deep learning denoising based approach for line spectral estimation. The proposed approach consists of a DnCNN, which is configured to perform denoising for noisy sinusoidal signals, a model order selection process, and a line spectral estimator applied on the denoised signal. We explain how the DnCNN is composed and the associated training process. Numerical results show the proposed approach can yield a substantial improvement in estimation accuracy over the ANM method as well as the one that directly applies MUSIC to the original observation. The benefit is attributed to the DnCNN denoiser, which is able to reshape the eigenspectrum of the sample covariance matrix of the observed signal, leading to better model order estimation and signal/noise subspace separation.

Manuscript received June 20, 2019; revised August 2, 2019; accepted August 20, 2019. Date of publication September 2, 2019; date of current version September 17, 2019. This work was supported in part by the Air Force Office of Scientific Research under Grant FA9550-16-1-0243 and in part by the National Science Foundation under Grants ECCS-1609393 and ECCS-1923739. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Steeve Zozor. *(Corresponding author: Hongbin Li.)*

Y. Jiang and H. Li are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: yjiang25@stevens.edu; hli@stevens.edu).

M. Rangaswamy is with the AFRL/RYAP, Wright-Patterson Air Force Base, OH 45433-7132 USA (e-mail: muralidhar.rangaswamy@us.af.mil).

Digital Object Identifier 10.1109/LSP.2019.2939049

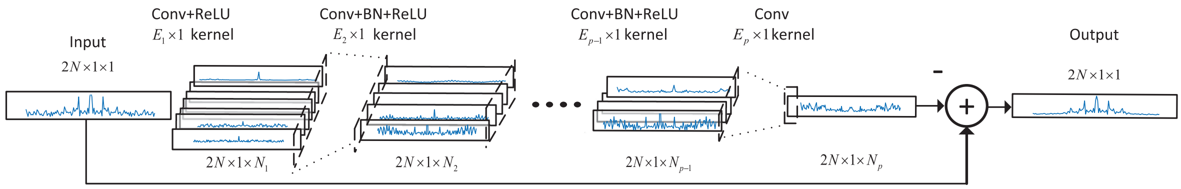

Fig. 1. Proposed DnCNN for sinusoidal signal denoising. The spectra of the signal before and after denoising, as well as of the intermediate signals produced by the DnCNN, are included in the sketch.

II. Problem Formulation

Consider a signal consisting of K complex sinusoids:

$$z_n = \sum_{k=1}^{K} \alpha_k e^{-j2\pi(n-1)f_k}, \quad n = 1, \cdots, N, \tag{1}$$

where $f_k \in [-0.5, 0.5)$ and α_k denote the normalized frequency and the complex amplitude of the k-th component, respectively. The observation can be written as $y_n = z_n + e_n$, where the signal z_n is corrupted by a complex additive white Gaussian noise e_n. By rewriting it into a vector form, we have

$$\mathbf{y} = \mathbf{z} + \mathbf{e}, \tag{2}$$

where $\mathbf{y} = [y_1, \cdots, y_N]^T$, $\mathbf{z} = [z_1, \cdots, z_N]^T$ and $\mathbf{e} = [e_1, \cdots, e_N]^T$. Let $\mathbf{h}(f_k) = [1, e^{-j2\pi f_k}, \cdots, e^{-j2\pi(N-1)f_k}]^T$. Then $\mathbf{z} = \mathbf{H}(\mathbf{f})\boldsymbol{\alpha}$, where $\mathbf{H}(\mathbf{f}) = [\mathbf{h}(f_1), \cdots, \mathbf{h}(f_K)]$, $\mathbf{f} = [f_1, \cdots, f_K]^T$ and $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_K]^T$. The problem of interest is to estimate sinusoidal parameters $\mathbf{f}$ and $\boldsymbol{\alpha}$ from noisy observation $\mathbf{y}$ when the model order K is unknown. In this work, we consider a denoised approach that first obtains a denoised observation $\hat{\mathbf{z}}$ from $\mathbf{y}$, and then performs spectral estimation using $\hat{\mathbf{z}}$.

III. Proposed Method

We employ a denoising convolutional neural network (DnCNN) for denoising. In the following, we describe the DnCNN used for denoising and the associated training process, and then explain how the denoised observation is used for model order selection and spectral estimation.

A. Deep Learning Based Denoiser

As depicted in Fig. 1, a DnCNN has a layered structure, consisting of P layers that perform convolution (Conv), batch normalization (BN), and/or rectified linear unit (ReLU) functions, the details of which are to be discussed shortly. Instead of estimating the desired signal, a DnCNN is trained to estimate the noise. This mechanism is known as residual learning, which enables the deep network to remove the structured signal rather than the unstructured noise. Compared with the conventional approach of directly learning the desired signal, residual learning can speed up the training process and improve the denoising performance [18], [19].

In the first layer, DnCNN takes the real and imaginary part of observation $\mathbf{y}$ and reshapes it into a $2N \times 1 \times 1$ tensor, denoted by $\mathbf{Y}^{(0)} \in \mathbb{R}^{2N\times1\times1}$. The first layer consists of N_1 convolutional filters $\mathbf{W}_i^{(1)} \in \mathbb{R}^{E_1\times1\times1}$, $i = 1, \cdots, N_1$, with the common input $\mathbf{Y}^{(0)} \in \mathbb{R}^{2N\times1\times1}$. The convolutional output is expressed as $\mathbf{W}_i^{(1)} * \mathbf{Y}^{(0)}$, which has the same dimension of $2N \times 1 \times 1$ as the input with proper zero padding. A bias term $\mathbf{b}_i^{(1)} \in \mathbb{R}^{2N\times1\times1}$ is frequently added into the convolution result. Thus the output of the convolution of the first layer can be expressed as

$$\mathbf{Y}_i^{(1)} = \mathrm{ReLU}(\mathbf{W}_i^{(1)} * \mathbf{Y}^{(0)} + \mathbf{b}_i^{(1)}), \quad i = 1, \cdots, N_1, \tag{3}$$

where the activation function $\mathrm{ReLU}(\cdot)$ is given by [20] $\mathrm{ReLU}(x) = \max(x, 0)$. The convolution and activation operations enable the deep neural network to gradually separate signal structure from the noisy observation through the hidden layers.

The p-th layer, $p = 2, \cdots, P-1$, consists of N_p convolutional filters $\mathbf{W}_i^{(p)} \in \mathbb{R}^{E_p\times1\times N_{p-1}}$, $i = 1, \cdots, N_p$, with a common input $\mathbf{Y}^{(p-1)} \in \mathbb{R}^{2N\times1\times N_{p-1}}$ from the $(p-1)$-st layer. These $P-2$ inner layers also implement the BN and ReLU functions. Thus, the output of the p-th layer can be expressed as

$$\mathbf{Y}_i^{(p)} = \mathrm{ReLU}\Big(\mathrm{BN}\big(\mathbf{W}_i^{(p)} * \mathbf{Y}^{(p-1)} + \mathbf{b}_i^{(p)}\big)\Big), i = 1, \cdots, N_p, \tag{4}$$

where $\mathbf{b}_i^{(p)} \in \mathbb{R}^{2N\times1\times1}$ contains the bias terms added into the convolution results and $\mathrm{BN}(\cdot)$ denotes the batch normalization unit, along with a $\mathrm{ReLU}(\cdot)$ function. BN is introduced to alleviate the so-called *internal covariate shift* by incorporating a normalization step and a scale/shift step before the nonlinearity in each layer [21]. It can be mathematically expressed as

$$\mathrm{BN}(\mathbf{x}; \gamma_p, \beta_p) = \gamma_p \frac{\mathbf{x} - E[\mathbf{x}]}{\sqrt{\mathrm{Var}[\mathbf{x}] + \epsilon}} + \beta_p, \tag{5}$$

where γ_p and β_p are the scaling and shift factors for the p-th layer, which can be learned from the training data, while ϵ is a small-valued constant that prevents the denominator $\sqrt{\mathrm{Var}[\mathbf{x}] + \epsilon}$ from becoming (close to) zero.

The last layer consists of $N_P = 1$ convolutional filter $\mathbf{W}_i^{(P)} \in \mathbb{R}^{E_P\times1\times N_{P-1}}$ with input $\mathbf{Y}^{(P-1)} \in \mathbb{R}^{2N\times1\times N_{P-1}}$ from the $(P-1)$-st layer. The final noise estimate is reconstructed as

$$\mathbf{Y}^{(P)} = \mathbf{W}^{(P)} * \mathbf{Y}^{(P-1)} + \mathbf{b}^{(P)}, \tag{6}$$

where $\mathbf{b}^{(P)} \in \mathbb{R}^{2N\times1\times1}$ contains the bias terms added into the convolution result.

Mathematically, DnCNN can be treated as a mapping $\boldsymbol{\mathcal{F}}(\mathbf{y}, \boldsymbol{\Theta})$ taking the observation $\mathbf{y}$ as input and defined by the parameter set $\boldsymbol{\Theta}$ of the DnCNN. The parameter set $\boldsymbol{\Theta}$ contains the convolution filters $\{\mathbf{W}_i^{(p)}\}$, bias $\{\mathbf{b}_i^{(p)}\}$, and scaling and shift factors $\{\gamma_p, \beta_p\}$. As $\mathbf{Y}^{(P)}$ forms an estimate of the real and imaginary parts of the noise vector in (2), we can use it to form a complex-valued noise estimate, denoted by $\hat{\mathbf{e}}$. Then, we subtract it from the observation $\mathbf{y}$, and obtain a denoised signal $\hat{\mathbf{z}} = \mathbf{y} - \hat{\mathbf{e}}$.

B. Offline Training

To effectively remove noise in dynamic environments, we need to train the DnCNN with a diverse set of training signals with different frequencies and signal-to-noise ratio (SNR) conditions. The frequencies of the training signals are generated as uniformly distributed random variables over a range that covers the signals of interest, assuming some prior knowledge of the signal bandwidth (BW) is available. The BW of training signals does not have to be strictly matched to that of the test signal. It suffices to use a loose upper bound of the signal BW for training signal generation. Another important factor in generating the training data is the SNR. Since the SNR of the test signal is unknown, DnCNN has to be trained with training data with a range of SNR values.

In our proposed scheme, the DnCNN is trained to estimate the residual or noise. The objective of training is to minimize the difference between the output of the DnCNN and the noise with respect to the DnCNN parameter $\mathbf{\Theta}$. Let $\mathbb{S}_{\mathrm{t}} = \left\{\{\mathbf{y}^{(j)}\}_{j=1}^{S}, \{\mathbf{z}^{(j)}\}_{j=1}^{S}\right\}$ be the training set that consists of S noise-corrupted observations and the noiseless signals. The loss function for training is:

$$\boldsymbol{\mathcal{L}}(\mathbb{S}_{\mathrm{t}}, \mathbf{\Theta}) = \sum_{j=1}^{S} \|\boldsymbol{\mathcal{F}}(\mathbf{y}^{(j)}, \mathbf{\Theta}) - (\mathbf{y}^{(j)} - \mathbf{z}^{(j)})\|_2^2. \quad (7)$$

Through the training process, the DnCNN parameter set $\mathbf{\Theta}$, which includes the convolution filters $\{\mathbf{W}_i^p\}$, bias $\{\mathbf{b}_i^p\}$, and scale/shift parameters $\{\gamma_p, \beta_p\}$), is obtained by minimizing the above loss function. In our simulation, we use Matlab's Deep Learning Toolbox to train the DnCNN and obtain $\mathbf{\Theta}$.

C. Model Order Selection and Spectral Estimation

After denoising, model order selection and sinusoidal parameter estimation are performed by using the denoised data $\hat{\mathbf{z}}$. Many model order selection methods are available, from the classical AIC [22] and MDL [23], to more recent techniques tailored for specific spectral estimators [24], multi-dimensional spectral analysis [25], and spectral analysis with quantized data [26]. For our problem, we employ the MDL criterion for model order selection due to its consistent performance [27]. Let $L \geq K$ denote an upper bound of the model order K, and define $\hat{\mathbf{z}}_t = [\hat{z}_t, \cdots, \hat{z}_{t+L-1}]$, $t = 1, \cdots, N-L+1$. Usually, L is selected such that $N/3 \leq L \leq N/2$ [27], which provides a good trade-off among bias, variance, and spectral resolution. The sample covariance matrix is given by

$$\hat{\mathbf{R}} = \tfrac{1}{N-L+1} \sum_{t=1}^{N-L+1} \hat{\mathbf{z}}_t \hat{\mathbf{z}}_t^{\mathrm{H}}. \quad (8)$$

When data is limited (N is small), the sample covariance matrix can be improved by forward-backward smoothing [28]: $\tilde{\mathbf{R}} = \frac{1}{2}(\hat{\mathbf{R}} + \mathbf{J}\hat{\mathbf{R}}^{\mathrm{T}}\mathbf{J})$, where $\mathbf{J}$ is the exchange matrix with ones on the anti-diagonal and zeros elsewhere.

Given $\tilde{\mathbf{R}}$, the signal and noise subspaces can be obtained by eigendecomposition, $\tilde{\mathbf{R}} = \mathbf{U}\mathbf{S}\mathbf{U}^{\mathrm{H}}$, where $\mathbf{S}$ denotes a diagonal matrix of the eigenvalues $\{\lambda_l\}_{l=1}^{L}$ (arranged in a non-increasing order) and $\mathbf{U}$ contains the corresponding eigenvectors. The MDL cost function can be written as [23]

$$\mathrm{MDL}(l) = c_1(l) \log \left(\frac{\frac{1}{L-1} \sum_{m=l+1}^{L} \lambda_l}{\prod_{m=l+1}^{L} \lambda_m^{1/(L-1)}} \right) + c_2(l), \quad (9)$$

where $c_1(l) = (L-l)(N-L+1)$ and $c_2(l) = \frac{1}{2} l(2L - l)\log(N-L+1)$. The model order estimate is $\hat{K} = \arg\min_{1 \leq l \leq L} \mathrm{MDL}(l)$.

Many methods are available to solve the spectral estimation problem. Here, we use the MUSIC algorithm since it is simple and is free of the grid mismatch problem. Given the model order estimate $\hat{K}$, the eigenvectors can be split in two groups, $\mathbf{U} = [\mathbf{U}_s, \mathbf{U}_e]$, where $\mathbf{U}_s \in \mathbb{C}^{L \times \hat{K}}$ spans the signal subspace and $\mathbf{U}_e \in \mathbb{C}^{L \times (L-\hat{K})}$ the noise subspace. The frequency estimates can be obtained by the root-MUSIC algorithm, which computes the roots of the following polynomial [27]:

$$\mathbf{a}^{\mathrm{T}}(z^{-1}) \mathbf{U}_e \mathbf{U}_e^{\mathrm{H}} \mathbf{a}(z) = 0, \quad (10)$$

where $\mathbf{a}(z) = [1, z^{-1}, \cdots, z^{-L+1}]^T$. Specifically, the frequency estimate $\hat{\mathbf{f}}$ can be obtained from the phase of the $\hat{K}$ roots that are inside and closest to the unit circle. Finally, the amplitude estimate $\hat{\boldsymbol{\alpha}}$ can be obtained by least squares.

IV. Numerical Results

In this section, we present simulation results to demonstrate the performance of the deep learning based approach for denoising and line spectral estimation. For simplicity, the proposed method is called DnCNN-MUSIC. The SNR is defined as $\mathrm{SNR} = N\|\boldsymbol{\alpha}\|^2/\sigma^2$, where σ^2 is the noise variance. The performance metric is the mean squared error (MSE):

$$\mathrm{MSE} \triangleq \sum_{k=1}^{K} \min_{\hat{f} \in \hat{\mathbf{f}}} E[\hat{f} - f_k]^2. \quad (11)$$

We compare DnCNN-MUSIC with three other methods including MUSIC, which applies the root-MUSIC algorithm directly on the original data $\mathbf{y}$, ANM [15], and Lasso [5]. Lasso uses a fixed-size dictionary formed from 1024 uniformly spaced frequency points.

In our simulation, the DnCNN is implemented in Matlab. It has $P = 11$ layers along with the following parameters: $\{E_p\}_{p=1}^{11} = \{8, 4, 4, \cdots, 4, 16\}$ and $\{N_p\}_{p=1}^{11} = \{128, 64, 32, \cdots, 32, 16, 1\}$. Before training the network, we first generate the training data and validation data under $\mathrm{SNR} = 15, 20, 25, 30, 35, 40$ dB, with 1600 training signals and 400 validation signals for each SNR. The frequency set contains $K = 5$ frequencies uniformly distributed over the BW $(-0.25, 0.25)$ and varying independently from one training signal to another. We use the mini-batch gradient descent method to train the network. Each mini-batch contains 10 training signals. In each iteration, one mini-batch is randomly picked to compute the gradient. The Adam optimizer is employed to determine the optimal neural parameters [29]. During the training process, we operate the validation every 200 iterations. Training continues until the loss does not drop for consecutive iterations. The initial learning rate is set to be 10^{-3} and the learning rate drop factor is 0.8.

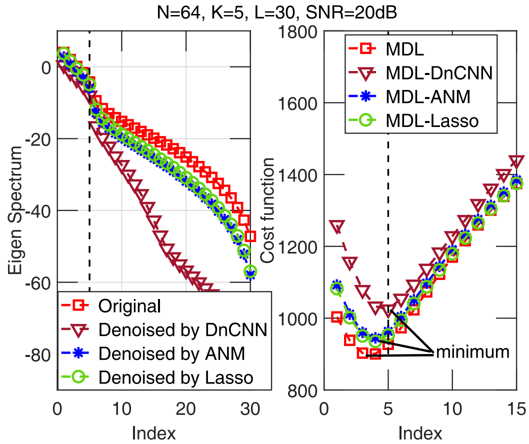

Fig. 2. Eigenspectrum (left) and MDL cost function (right) before and after denoising. The black dashed line indicates the true model order $K = 5$.

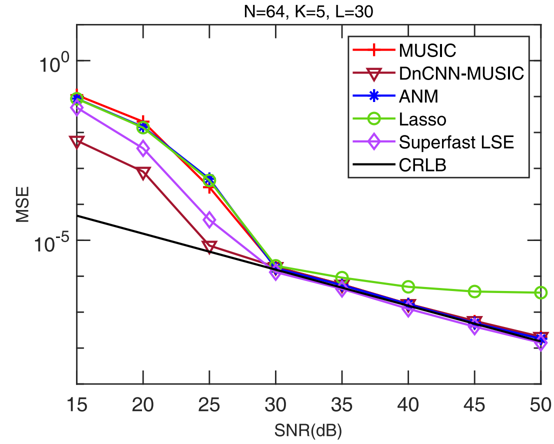

Fig. 3. MSE of the frequency estimates versus SNR.

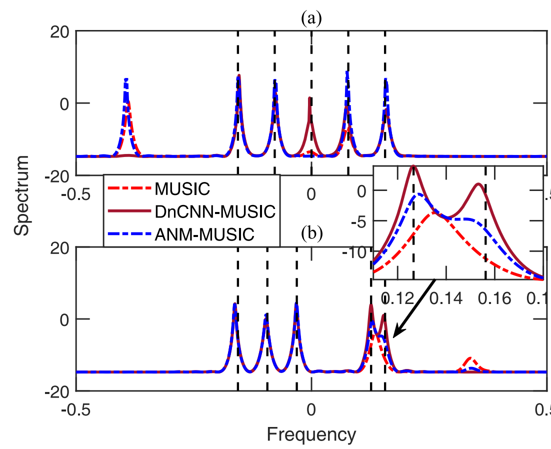

Fig. 4. Spectral estimates with (a) well-separated, and (b) closely space frequencies. The black dashed lines indicate the locations of the true frequencies.

The test signals contain $K = 5$ sinusoids with frequencies and amplitudes randomly generated from each Monte Carlo trial. In our simulations, the testing frequencies are uniformly distributed over the BW $(-0.2, 0.2)$; hence the bandwidth of the training signals is slightly larger than that of the test signals. Note that the training data should have frequencies covering a BW no less than that of the test signal. The length of the observed signal is $N = 64$. The root-MUSIC uses $L = 30$ to compute the covariance matrix.

We first examine the model order selection performance. Fig. 2 shows the eigenspectrum and the MDL cost function by using the original and, respectively, denoised data produced by ANM, Lasso, and DnCNN denoiser. The eigenspectrum shows the eigenvalues of the sample covariance matrix $\hat{\mathbf{R}}$ sorted in a non-increasing order. For DnCNN, the eigenspectrum shows a more defined boundary between the first $K = 5$ eigenvalues, which are due to the sinusoidal signals, and the remaining eigenvalues, due to the noise. This implies the denoising provided by the DnCNN allows one to better separate the signal subspace from the noise subspace. The benefit is also reflected in the model order selection result, which shows that DnCNN can correctly estimate the model order with the minimum of $\mathrm{MDL}(l)$ located at $\hat{K} = 5$ in average. Meanwhile, it is seen that the ANM and Lasso yield limited improvement in terms of both the eigenspectrum and model order selection.

Fig. 3 compares the frequency estimation performance of the above methods, along with their corresponding model order selection estimates, and the Cramer-Rao lower bound (CRLB). In addition, we include a recently introduced Bayesian off-grid sparsity based method, called the superfast LSE [14], which was shown to yield competitive performance for linear spectral estimation. It can be seen from the figure that ANM and MUSIC have similar performance for all SNR values suggesting that ANM denoising offers limited gain. Superfast LSE outperforms ANM by about 2 dB. Lasso is the worst at high SNRs due to the fixed dictionary, which causes grid mismatch. The proposed DnCNN-MUSIC performs the best among all methods and has an improvement of about 5 dB over ANM in approaching the CRLB.

Finally, spectral estimates obtained by applying the spectral MUSIC algorithm [27] to either the original or denoised data, denoised by ANM or DnCNN, are presented in Fig. 4(a) when all frequencies are well separated, and in Fig. 4(b) when some frequencies are more closely spaced. In the first case, although the frequencies are well separated, MUSIC and ANM-MUSIC show one pseudo peak toward the left and miss one true peak in the middle, while DnCNN-MUSIC does not have such issues, indicating that DnCNN has a better denoising performance. The second case is more challenging, involving two closely spaced frequencies. It is seen that MUSIC and ANM-MUSIC show a single merged peak for two closely spaced frequencies and, furthermore, a pseudo peak toward the left. In contrast, DnCNN-MUSIC is able to obtain all signal peaks without any pseudo ones.

V. Conclusions

We proposed a deep learning denoising based approach to line spectral estimation. The proposed approach employs a DnCNN derived from residual learning for signal denoising and then the MDL/MUSIC algorithm applied on the denoised data for line spectral estimation. Numerical results show that the proposed approach can benefit from deep learning and outperforms a recently introduced ANM denosing method.

REFERENCES

[1] Z.-Q. He, Z.-P. Shi, L. Huang, and H. C. So, "Underdetermined DOA estimation for wideband signals using robust sparse covariance fitting," *IEEE Signal Process. Lett.*, vol. 22, no. 4, pp. 435–439, Apr. 2015.

[2] B. Liao, S.-C. Chan, L. Huang, and C. Guo, "Iterative methods for subspace and DOA estimation in nonuniform noise," *IEEE Trans. Signal Process.*, vol. 64, no. 12, pp. 3008–3020, Jun. 2016.

[3] L. Borcea, G. Papanicolaou, C. Tsogka, and J. Berryman, "Imaging and time reversal in random media," *Inverse Problems*, vol. 18, no. 5, 2002, Art. no. 1247.

[4] L. Zheng and X. Wang, "Super-resolution delay-Doppler estimation for OFDM passive radar," *IEEE Trans. Signal Process.*, vol. 65, no. 9, pp. 2197–2210, May 2017.

[5] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning With Sparsity: The Lasso and Generalizations*. London, U.K.: Chapman and Hall, 2015.

[6] Y. Chi, L. L. Scharf, A. Pezeshki, and A. R. Calderbank, "Sensitivity to basis mismatch in compressed sensing," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2182–2195, May 2011.

[7] Z. Yang, C. Zhang, J. Deng, and W. Lu, "Orthonormal expansion l_1-minimization algorithms for compressed sensing," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6285–6290, Dec. 2011.

[8] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.

[9] E. J. Candès and C. Fernandez-Granda, "Super-resolution from noisy data," *J. Fourier Anal. Appl.*, vol. 19, no. 6, pp. 1229–1254, 2013.

[10] L. Hu, Z. Shi, J. Zhou, and Q. Fu, "Compressed sensing of complex sinusoids: An approach based on dictionary refinement," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3809–3822, Jul. 2012.

[11] Z. Yang, L. Xie, and C. Zhang, "Off-grid direction of arrival estimation using sparse Bayesian inference," *IEEE Trans. Signal Process.*, vol. 61, no. 1, pp. 38–43, Jan. 2013.

[12] J. Fang, F. Wang, Y. Shen, H. Li, and R. Blum, "Super-resolution compressed sensing for line spectral estimation: An iterative reweighted approach," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4649–4662, Sep. 2016.

[13] J. Dai, X. Bao, W. Xu, and C. Chang, "Root sparse Bayesian learning for off-grid DOA estimation," *IEEE Signal Process. Lett.*, vol. 24, no. 1, pp. 46–50, Jan. 2017.

[14] T. L. Hansen, B. H. Fleury, and B. D. Rao, "Superfast line spectral estimation," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2511–2526, May 2018.

[15] B. N. Bhaskar, G. Tang, and B. Recht, "Atomic norm denoising with applications to line spectral estimation," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5987–5999, Dec. 2013.

[16] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[18] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[20] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[22] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.

[23] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 2, pp. 387–392, Apr. 1985.

[24] R. Badeau, B. David, and G. Richard, "Selecting the modeling order for the ESPRIT high resolution method: An alternative approach," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, 2004, pp. 1025–1028.

[25] K. Liu, J. P. C. Da Costa, H. C. So, and L. Huang, "Subspace techniques for multidimensional model order selection in colored noise," *Signal Process.*, vol. 93, no. 7, pp. 1976–1987, 2013.

[26] C. Li, R. Zhang, J. Li, and P. Stoica, "Bayesian information criterion for signed measurements with application to sinusoidal signals," *IEEE Signal Process. Lett.*, vol. 25, no. 8, pp. 1251–1255, Aug. 2018.

[27] P. Stoica and R. L. Moses, *Spectral Analysis of Signals*. Upper Saddle River, NJ, USA: Prentice-Hall, 2005.

[28] H. Li, J. Li, and P. Stoica, "Performance analysis of forward-backward matched-filterbank spectral estimators," *IEEE Trans. Signal Process.*, vol. 46, no. 7, pp. 1954–1966, Jul. 1998.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015.