A Qualitative Deep Learning Method for Inverse Scattering Problems

He Yang [1] **and Jun Liu** [2]

[1]Department of Mathematics, Augusta University, Augusta, GA 30912, USA
hyang1@augusta.edu

[2]Department of Mathematics and Statistics, Southern Illinois University Edwardsville, Edwardsville, IL 62026, USA
juliu@siue.edu

Abstract **–** In this paper, we propose a novel deep convolutional neural network (CNN) based qualitative learning method for solving the inverse scattering problem, which is notoriously difficult due to its highly nonlinearity and ill-posedness. The trained deep CNN accurately approximates the nonlinear mapping from the noisy far-field pattern (from measurements) to a disk that fits the location and size of the unknown scatterer. The used training data is derived from the simulated noisy-free far-field patterns of a large number of disks with different randomly generated centers and radii within the domain of interest. The reconstructed fitting disk is also very useful as a good initial guess for other established nonlinear optimization algorithms. Numerical results are presented to illustrate the promising reconstruction accuracy and efficiency of our proposed qualitative deep learning method.

Index Terms **–**convolutional neural network, deep learning, inverse acoustic scattering, qualitative method.

I. INTRODUCTION

Inverse scattering problems [1] arise in many fields of science and engineering, such as radar and sonar, biomedical imaging, and non-destructive testing. In the last few decades, many numerical algorithms have been developed for solving such nonlinear and ill-posed inverse problems, see, e.g., [2, 3]. In this paper, we propose a new deep learning based qualitative method.

Existing algorithms can be roughly categorized into two groups: (i) nonlinear optimization methods, and (ii) qualitative methods. The nonlinear optimization methods [4, 5] often need to solve a direct (forward) scattering problem at each iteration. Although such methods require less amount of data, they indeed require *a priori* knowledge of the boundary conditions of the unknown scatterer (e.g, sound-soft or not), which may not be available. Furthermore, if the initial guess is far away from the true solution, the optimization iterations may converge to a local minimum, leading to an inaccurate reconstruction of the true scatterer. On the other hand, the qualitative methods [6–9], including the linear sampling method (LSM) [6], the factorization method (FM) [10], and the direct sampling method (DSM) [3], have the advantage of not requiring much *a priori* information about the unknown scatterer. In addition, such qualitative methods were shown to be computationally faster than the nonlinear optimization methods and are highly parallelizable. However, it is well-known that both LSM and FM suffer from the severely ill-conditioned discretized far-field operator, which requires a costly Tikhonov regularization in order to achieve a robust approximation accuracy in the presence of noise in measured far-field data. To reduce the computational cost, an adaptive quadrature-based factorization method (AFM) was recently developed in [11, 12], which dramatically speeds up the standard FM while gives comparable reconstructions. Nevertheless, an effective implementation of the Tikhonov regularization relies on the knowledge of the unknown noise level. The recently developed DSM [3] and direct factorization method (DFM) [13] were shown to be capable of achieving similar reconstruction accuracy, without resorting to any such Tikhonov regularization processes. Therefore, such regularization-free DSM and DFM are more suitable to the real world applications.

All the above-discussed qualitative methods strive to reconstruct the accurate location and precise shape of the unknown scatterer, which sometimes may be unnecessary for some practical applications. In such situations, we merely need to find a rough estimation of the scatterer's location and size (support), which hopefully can be achieved with lower computational cost or even less amount of data. For example, the approach in [14] only approximately recovers a convex hull of obstacles by using limited aperture data. The range test method in [15] obtains a convex support of the scatterer as the intersection of many convex test domains. More recently, an interesting extended sampling method (ESM) was proposed in [16] and further improved in [17], where a fitting disk was identified to estimate the support (location and size) of the scatterer. In other words, the precise shape of unknown scatterer is not of primal interest anymore, but

Submitted On: November 17, 2019
Accepted On: January 10, 2020

its size (or support) is the reconstruction target.

In this paper, we propose a deep learning based qualitative method for solving the inverse scattering problems. Nowadays, most deep learning methods are based on the artificial neural networks, which are effective models for approximating certain functions. There are many types of neural networks (NNs), including the feedforward and recurrent NN. The feedforward NN is a network from the input layer to the output layer without any loops. Some widely used NNs, including the multilayer perceptron [18], the autoencoder [19], the convolutional NN [20] and the U-Net [21], belong to the category of feedforward NNs. The recurrent NNs, by contrast, can pass data forward and backward. For example, long short-term memory (LSTM) [22] is a type of recurrent NN. Since the pioneering work by Krizhevsky, Sutskever and Hinton [23], deep learning methods have been applied in various fields, including image recognition, medical imaging and language translation. Recently, deep learning approaches [24] have also been used to solve the inverse scattering problems. In particular, the authors in [25] proposed an U-net convolutional neural network to reconstruct the permittivities of dielectric scatters from the scattering data. In [26], the authors developed a convolutional neural network for the far-field subwavelength imaging. In [27], the authors proposed the so-called SwitchNet to reconstruct the scatterer field which is a mixture of Gaussians. Different from the aforementioned work, our present paper focuses on developing a *qualitative* method to approximate the location and size of the scatterer as a fitting disk, rather than its exact shape. Therefore, the training process of our proposed CNN is more efficient and the prediction is also robust with respect to noise. Moreover, our designed CNN utilizes three dropout layers which effectively prevent overfitting phenomena.

The remaining of the paper is organized as follows. In section II, we briefly review the standard inverse obstacle scattering problem. In section III, we describe our proposed deep convolutional neural network (CNN) for the inverse scattering problems. In section IV, we show some numerical results to demonstrate the performance of our proposed method. Finally, some concluding remarks are given in section V.

II. THE INVERSE OBSTACLE SCATTERING PROBLEM

Following [1], we briefly describe the standard inverse obstacle scattering problem. Let $D \subset \mathbb{R}^2$ be a bounded impenetrable sound-soft obstacle with a C^2 boundary ∂D. Let θ be an incident direction on the unit circle $\mathbb{S}$ and $\kappa > 0$ be the wave number (with wavelength $\lambda = 2\pi/\kappa$ meters). Given a time-harmonic incident plane wave field $u^i(x) = e^{i\kappa x\cdot\theta}$, its propagation in the presence of the obstacle D, which is situated in a homogeneous medium, will lead to a scattered wave field u^s. Then the obtained total field $u = u^i + u^s$ is the solution to the following scalar exterior Helmholtz equation:

$$\Delta u(x) + \kappa^2 u(x) = 0, \quad x \in \mathbb{R}^2\backslash\overline{D} \tag{1}$$

subject to the Dirichlet boundary condition (sound-soft)

$$u = 0, \quad \text{on} \quad \partial D \tag{2}$$

and the Sommerfeld radiation condition (here $|x|$ denotes the distance between x and the origin):

$$\lim_{|x|\to\infty} |x|^{\frac{1}{2}} \left(\frac{\partial u^s}{\partial |x|} - i\kappa u^s\right) = 0. \tag{3}$$

The above direct scattering problem (1-3) admits a unique solution $u \in C^2(\mathbb{R}^2\backslash\overline{D})\bigcap C^1(\mathbb{R}^2\backslash D)$. Moreover, the scattered field u^s has the asymptotic behavior:

$$u^s(x) = \frac{e^{i\kappa|x|}}{|x|^{1/2}} u^\infty(\hat{x},\theta) + O(|x|^{-3/2})$$

(as $|x| \to \infty$) uniformly in all directions, where $\hat{x} = x/|x|$ is the observation direction on the unit circle $\mathbb{S}$ and u^∞ is called the far-field pattern. Obviously, the measurable far-field pattern $u^\infty : \mathbb{S}\times\mathbb{S} \to \mathbb{C}$ depends nonlinearly on the obstacle's shape ∂D that is to be determined.

The standard inverse obstacle scattering problem is to recover the obstacle's shape ∂D from the measured noisy far-field pattern data $u^\infty(\hat{x},\theta)$ with a fixed $\kappa > 0$ for all incident directions $\theta \in \Gamma_s$ and observation directions $\hat{x} \in \Gamma_m$. In this paper, we assume full aperture data i.e. $\Gamma_s = [0, 2\pi]$ and $\Gamma_m = [0, 2\pi]$.

More precisely, we essentially need to invert the following nonlinear abstract operator equation:

$$\mathcal{F}(\partial D) = u^\infty(\hat{x},\theta), \quad \hat{x},\theta \in \mathbb{S},$$

where the forward operator $\mathcal{F}$ maps the boundary of the obstacle D to the corresponding far-field pattern for all pairs of directions $(\hat{x},\theta)$. This abstract operator equation turns out to be highly nonlinear and severely ill-posed, and it has been solved by Newton's method [4,28], with the Fréchet derivative of $\mathcal{F}$ being inverted using Tikhonov regularization at each iteration. Such a locally convergent nonlinear iterative method is costly in practical computations and its effectiveness highly depends on the initial guess, i.e., the *a priori* information, which may lead to incorrect approximations.

The main difficulty in the accurate reconstruction of ∂D lies in the nonlinearity and ill-posedness of $\mathcal{F}$, although ∂D is indeed uniquely determined by u^∞ on the unit disk. The inverse map $\mathcal{F}^{-1}$ from u^∞ to ∂D is not easy to compute numerically, but it may be easier to approximate if we simply estimate ∂D by a disk $B(z;r)$ with center $z = (x,y) \in [a,b]\times[c,d]$ and radius $r \in [r_{\min}, r_{\max}]$. Inspired by several recent works [25–27, 29–33] in this direction, we propose to qualitatively approximate $\mathcal{F}^{-1}$ by a convolutional neural network (CNN), denoted by $\mathcal{G}$, based on the simulated training data of randomly generated disks within a prescribed domain containing D. The trained CNN $\mathcal{G}$ approximately

maps the noisy far-field pattern data to a fitting disk (including its center location (x, y) and radius r), which is poised to estimate the support of the underlying unknown scatterer. The availability of a large amount of simulated training data is crucial to optimize such a robust qualitative CNN, so that it achieves a satisfactory approximation accuracy. Theoretically, one can also use any other shapes (e.g., ellipse) of interest to simulate the training data, but it will become computationally more expensive due to more degrees of freedom in parameterizing the targeting irregular shapes. The disk shape is the simplest since its far-field pattern is invariant with respect to rotation, and only 3 parameters (x, y, r) are sufficient to identify a disk.

III. A DEEP CONVOLUTIONAL NEURAL NETWORK

In this section, we introduce our deep convolutional neural network (CNN) for qualitatively solving the above inverse obstacle scattering problems. Generally speaking, a CNN can be regarded as the composition of a sequence of functions, with each function representing a layer that takes the output of the previous layer and computes the input for the next layer. For the classification tasks, for example in [34], the architecture of a CNN generally consists of convolutional layers, pooling layers, activation layers, dropout layers and a loss layer. For the tasks of regression, however, sometimes the pooling layer is not necessary in the CNN architecture [35].

In this work, we construct and train a deep convolutional neural network using the disk-shape scatterers and the corresponding far-field data matrices. All lengths are measured in meters (m). We first generate 5000 circles with their x- and y-coordinates randomly distributed between $a = c = 0.5$ and $b = d = 10$, and their radii are randomly distributed in $[r_{\min} = 0.5, r_{\max} = 5]$. For each circular obstacle, we further simulate the corresponding noisy far-field data matrix of size 32×32, representing 32 incident and observation directions. Since each far-field data matrix is a 32-by-32 complex matrix, it can be treated as a $32 \times 32 \times 2$ tensor, which is then used as the input of our CNN. The output of the CNN is a 3-by-1 vector representing the x- and y-coordinates as well as the radius of the corresponding circular obstacle. Next, we use the dataset (far-field data matrices) for the input of the deep CNN of size $32 \times 32 \times 2 \times 5000$, and their corresponding labels (ground truth centers and radii) of size $3 \times 1 \times 5000$ to train and test our designed CNN. It is worthwhile to emphasize that a much larger size of dataset can be easily simulated if a higher accuracy is pursued, which however takes longer training time. We remark that the architecture of our proposed convolutional neural network is quite different from those networks developed in [25–27, 30–33]. One special feature of our designed network architecture is that there are three dropout layers. Numerical simulations show that the network with these dropout layers leads to much better results than the network without dropout layers. This can prevent overfitting efficaciously so that the prediction are very accurate for both the training and test dataset. Our proposed CNN mainly serves as a proof of concept to demonstrate the feasibility of such a qualitative deep learning approach, which we believe has a lot more room for improvement in terms of different network architecture and better reconstruction accuracy.

The architecture of our deep convolutional neural network is depicted in Fig. 1, where the detailed configuration of each layer is summarized in Table 1. The output of the current layer is of the same size as the input of the next layer. The first layer of the network is a convolutional layer with 30 filters of size 5×5, stride size of 2×2, and the same size padding. The first layer is followed by a rectified linear unit (ReLU) layer. The third layer is a dropout layer which randomly selects the input neurons to zero according to 10% probability. Such a layer is used to prevent overfitting efficiently [23, 36]. The fourth to the sixth layers are the convolutional layer, ReLU layer and the dropout layer, respectively. The seventh to the ninth layers are arranged in the same pattern. The fourth layer, i.e., the convolutional layer, has 60 filters of size 3×3, with stride size of 3×3 and the same size padding. Another convolutional layer, i.e., the seventh layer, has 180 filters of size 3×3 with stride size of 2×2 and the same size padding. The sixth and ninth layers are both dropout layers which set the input neurons to zero according to 10% probability. The tenth layer is a fully connected layer which combines all the information from previous layers and predicts the radius and coordinates of the fitting disk. The output of this layer will be further used to compute the mean-square error between the predicted and the ground truth labels (3×1 vectors).

Mathematically, if we reshape each layer of neurons as a column vector, then our convolutional neural network $\mathcal{G}$ can be represented by a nonlinear mapping:

$$\mathcal{G}(v) := A_4 \,\mathrm{ReLU}(A_3 \,\mathrm{ReLU}(A_2 \,\mathrm{ReLU}(A_1 v))) + g,$$

where v is a 2048-by-1 vector representing the input far-field data, A_1, A_2 and A_3 are the sparse Toeplitz matrices due to the convolution operators, ReLU is the element-wise Rectified Linear Unit function, A_4 is a dense weight matrix and g is a bias vector. Both A_4 and g are from the last fully connected layer. The output $\mathcal{G}(v)$ is a 3×1 vector. Given m pairs of training data (v_i, w_i) for $i = 1, 2, \ldots, m$, where v_i is the far-field data and w_i is the corresponding ground truth label, the training process is to minimize the total mean-square error, i.e.,

$$\min_{A_k, g} \quad \frac{1}{m} \sum_{i=1}^{m} \|\mathcal{G}(v_i) - w_i\|_2^2. \tag{4}$$

In our simulations, about half million parameters (includ-

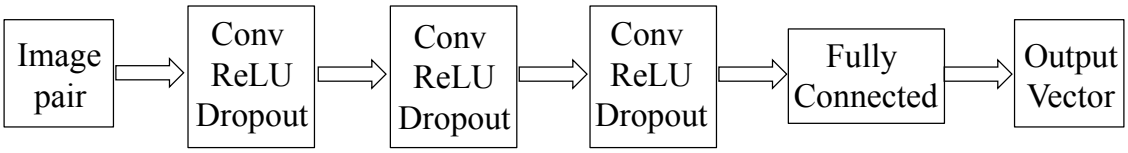

Fig. 1. Our proposed network architecture $\mathcal{G}$. The input image is of size $32 \times 32 \times 2$. There are three convolutional layers, three activation layers, three dropout layers, and one fully connected layer.

Table 1: Detail layers of our proposed deep convolutional neural network architecture $\mathcal{G}$

Layer	Type	Filter #	Filter Size	Stride Size	Input Size	Output Size
1	Conv	30	5×5	2×2	$32 \times 32 \times 2$	$32 \times 32 \times 30$
2	ReLU	-	-	-	$32 \times 32 \times 30$	$32 \times 32 \times 30$
3	Dropout	-	-	-	$32 \times 32 \times 30$	$32 \times 32 \times 30$
4	Conv	60	3×3	2×2	$32 \times 32 \times 30$	$32 \times 32 \times 60$
5	ReLU	-	-	-	$32 \times 32 \times 60$	$32 \times 32 \times 60$
6	Dropout	-	-	-	$32 \times 32 \times 60$	$32 \times 32 \times 60$
7	Conv	180	3×3	2×2	$32 \times 32 \times 60$	$32 \times 32 \times 180$
8	ReLU	-	-	-	$32 \times 32 \times 180$	$32 \times 32 \times 180$
9	Dropout	-	-	-	$32 \times 32 \times 180$	$32 \times 32 \times 180$
10	Fully Connected	-	-	-	$32 \times 32 \times 180$	3×1

ing $A_k (k = 1, 2, 3, 4)$ and g) of the CNN can be updated iteratively by the stochastic gradient descent (SGD) method (implemented in MATLAB Deep Learning Toolbox). Figure 2 shows the typical convergence history.

IV. NUMERICAL RESULTS

In this section, we provide several 2D inverse acoustic scattering examples from impenetrable and sound-soft obstacles to demonstrate the effectiveness of our proposed deep learning method. All simulations are implemented in MATLAB 2019a on a Dell Laptop with Intel(R) Core(TM) i7-7700HQ CPU@2.80GHz and 32GB RAM. The CPU time (in seconds) is estimated using timing functions `tic/toc`. To simulate the measurement noise, we added random noise to the simulated far-field data $F \in \mathbb{C}^{N \times N}$ according to:

$$F^{\delta} = F + \delta \|F\| \frac{S_1 + S_2 i}{\|S_1 + S_2 i\|},$$

where S_1 and S_2 are two $N \times N$ random matrices (with a standard normal distribution) generated by the MATLAB function `randn(N,N)`. Here the value of δ represents the level of noise based on relative error and the noise-free situation corresponds to the case with $\delta = 0\%$.

In our simulated training data, we choose the wave number $\kappa = 5$, the total number of incident and observation directions $N = 32$, and the noise level $\delta = 0\%$. As we have mentioned in the previous section, we first generate 5000 random circles of random radius between 0.5 and 5, and random x- and y-coordinates between 0.5 and 10. We then use these circles as the training scatterers, and generate the far-field data for each circle via the Nyström method [1]. We mention that numerical discretization errors are much smaller than the added noise. Our training dataset consists of $m = 4000$ far-field data matrices of size $32 \times 32 \times 2$, and 4000 corresponding labels of size 3×1. The remaining 1000 pairs are used as test dataset. We train our convolutional neural network using the training dataset and the SGD method with learning rate 0.01 and 1000 epochs. We use mini-batches of size 128 to speed up the computation time in each iteration. The training history of our convolutional neural network can be seen in Fig. 2. The blue curve is the base 10 logarithm of the mini-batch training loss, and the red curve the logarithm of the testing loss. Figure 2 indicates the convergence of the stochastic gradient descent method in the training procedure. We remark that different networks or learning rates may lead to different convergence rate.

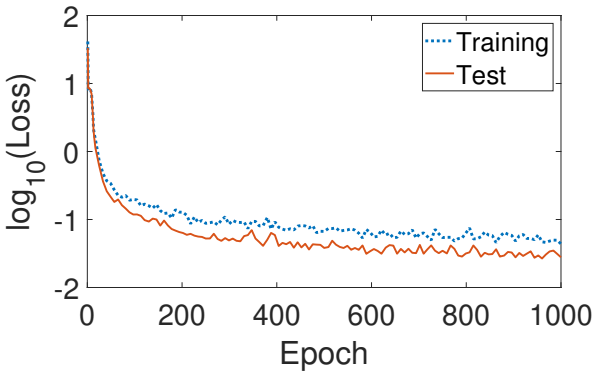

Fig. 2. Training history of our convolutional neural network. Horizontal axis: epoch number; vertical axis: logarithm (base 10) of the mini-batch loss.

Table 2: Mean square errors for the inverse scattering problem of circular scatterers with various random noise levels δ in the far-field data

Noise Level δ	Mean Square Error
0%	0.17249
10%	0.20229
20%	0.41218
30%	0.66666
40%	1.02110
50%	1.33220

Our results also show that the mean square errors of the training dataset and the test dataset are 0.13648 and 0.17249, respectively. Such convergence results indicate that our deep convolutional neural network model can generalize well from our training dataset to the unseen test dataset. A larger dataset would lead to better accuracy and it is also possible to use noisy far-field data as training dataset for better robustness.

We then investigate the performance of our deep neural network in reconstruction given far-field data with various noise levels. In Table 2, we present the mean square errors for the inverse scattering problem of 6000 circular scatterers (1000 scatterers for each test), with random noise levels $\delta = 0\%, 10\%, 20\%, 30\%, 40\%$ and 50% in the far-field data. The mean square errors are computed based on the prediction of our trained network, including the predicted radii, x- and y-coordinates of the centers. We observe that the mean square errors for $\delta = 0\%$ and 10% are quite similar. As the noise level goes up to 20%, the mean square error becomes twice as much as the error when $\delta = 10\%$ is used, but the roughly linear growth of error is still acceptable. When $\delta = 50\%$, the mean square error increases more dramatically due to the large amount of noise. Since we use circular objects with random locations and sizes, it is meaningful to check the relative errors of the predicted results. More specifically, (1) for $\delta = 0\%$, we find that 938 out of 1000 testing scatterers have the relative error of radius less than 0.1; 911 out of 1000 testing scatters have the relative error of the x-coordinate of the center less than 0.1; 937 out of 1000 testing scatterers have the relative error of the y-coordinate of the center less than 0.1. (2) For the dataset with 10% relative noise, 931 out of 1000 testing scatterers have the relative error of radius less than 0.1; 885 out of 1000 testing scatters have the relative error of the x-coordinate of the center less than 0.1; 890 out of 1000 testing scatterers have the relative error of y-coordinate of the center less than 0.1. (3) For $\delta = 20\%$, the numbers of testing scatterers that have relative error of radius, x- and y-coordinates to be less than 0.1 are 915, 718 and 745 out of 1000, respectively. These statistics indicate that our trained CNN can solve the inverse scattering problems of a circular object accurately in a high probability.

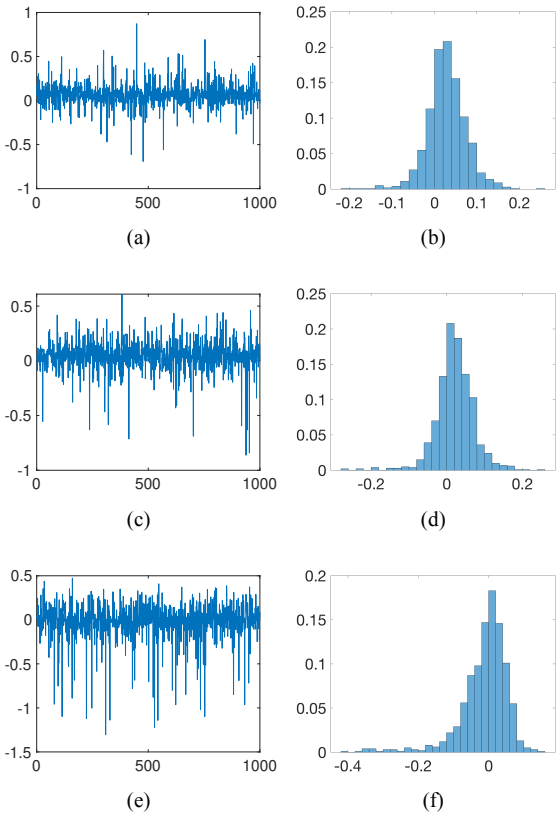

Fig. 3. The distribution of the prediction errors of the radii (left column) and the histogram of the relative errors (right column) with 3000 random far-field data under various noise levels: $\delta = 0\%$ (top row), 10% (middle row) and 20% (bottom row).

Moreover, among all three predicted parameters, the predicted radius is usually more accurate than the other parameters. This is due to the fact that the maximum of radii is smaller than that of x- and y-coordinates of the centers. In Fig. 3, we present the distribution of the prediction errors of the radii (in the left column) and the histogram of the relative errors (in the right column), where randomly perturbed far-field data with various noise levels are used. The prediction errors are roughly proportional to the added noise magnitude. We observe that the distributions of the relative errors in radius for $\delta = 0\%$ and 10% are quite similar, which means that our trained network is not sensitive to a small noise level.

Next, we investigate the performance of the proposed deep learning method for reconstructing randomly located scatterers with various shapes. In Figs. 4-6, we show the reconstructed fitting disks for obstacles in the shape of a circle, ellipse, rectangle, kite, peanut and triangle with $\delta = 0\%, 20\%$ and 40%, respectively. The center of mass for each object mostly lies in the reconstructed disk, although we do observe slightly increasing discrep-

ancy as the noise level becomes larger. For $\delta = 0\%$ and 20%, it is easy to see that the circular and triangular objects lead to the most accurate results. The reconstructed results for the obstacles in the shape of ellipse, kite and peanut for $\delta = 0$ and 20% are comparable. However, the reconstructed result for an object in the shape of a rectangle is worse when $\delta = 20\%$. For far-field data with $\delta = 40\%$, our trained network can still capture the unknown objects well except for the rectangular one.

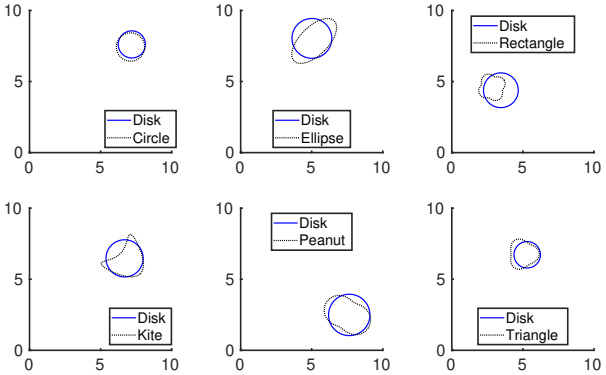

Fig. 4. The reconstructed fitting disks for randomly located different testing scatterers (noise level $\delta = 0\%$).

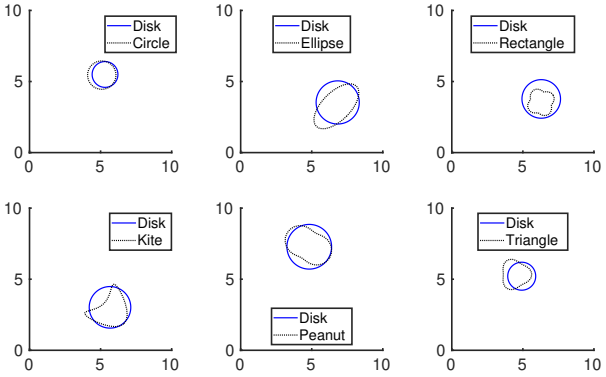

Fig. 5. The reconstructed fitting disks for randomly located different testing scatterers (noise level $\delta = 20\%$).

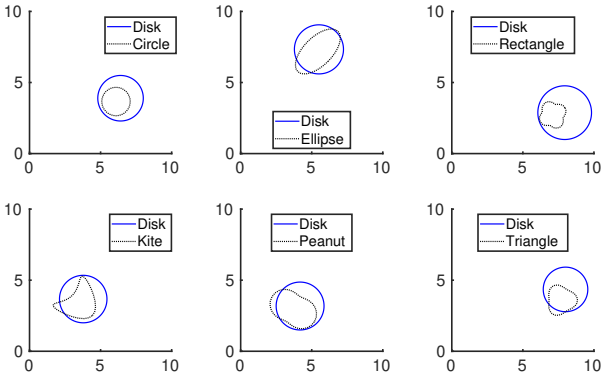

Fig. 6. The reconstructed fitting disks for randomly located different testing scatterers (noise level $\delta = 40\%$).

Figure 7 shows the reconstruction results for scatterers of various sizes. While most of the predicted disks are accurate, the rectangular-shape object was overestimated again. In particular, the radius of small scatterers (ellipse and peanut) is about 0.1, which is smaller than the minimal radius 0.5 used in the training dataset.

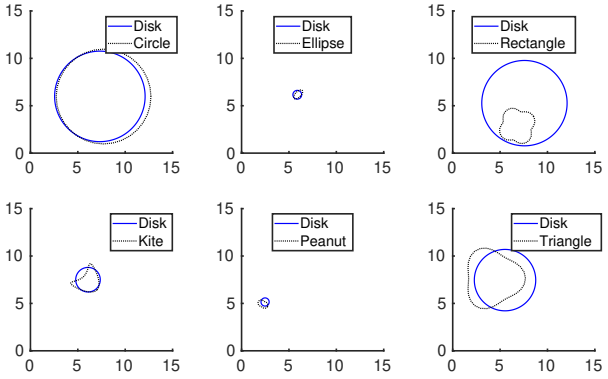

Fig. 7. The reconstructed fitting disks for randomly located scatterers of various sizes (noise level $\delta = 0\%$).

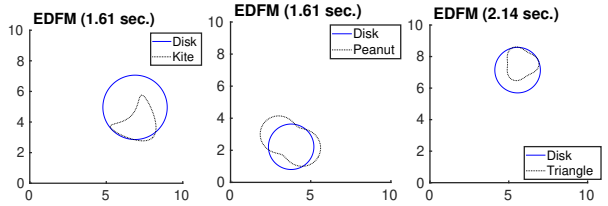

Fig. 8. The reconstructed fitting disks by EDFM [17] for randomly located scatterers (noise level $\delta = 40\%$). The EDFM is based on one incident direction at $\pi/2$ and 32 observation directions with a 500×500 sampling mesh.

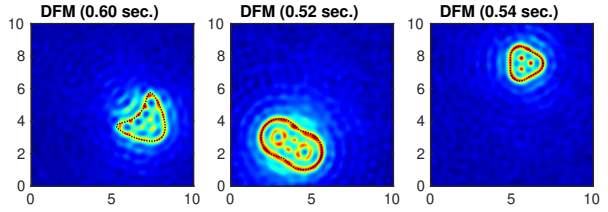

Fig. 9. The reconstructed profiles by DFM [13] for the same scatterers as in Fig. 7 (noise level $\delta = 40\%$). The DFM is based on 32 incident and observation directions with a 500×500 sampling mesh.

High computational efficiency is another remarkable advantage of our deep learning method, which of course relies on the expensive offline training that can be speed up by CPU/GPU parallel computing. Different from the extended sampling method [16] and the extended direct factorization method (EDFM) [17], where the radius is obtained with much more extra efforts after the center is chosen from a sampling process over the search domain, our deep learning method computes the radius and center simultaneously without any sampling procedure. With a pre-trained network, our deep learning method takes only *about 2 milliseconds* to estimate a fitting disk (using the

`predict` function provided by MATLAB).

As a comparison, Fig. 8 shows the corresponding results computed by the EDFM [17], where a similar fitting disk was estimated but with significantly higher CPU time. However, the EDFM can handle limited aperture data, which was not addressed in our deep learning model. In addition, Fig. 9 plots the constructed profiles by the highly efficient and vectorized DFM [13], where the same scatterers' shapes are qualitatively identified with more CPU time as well. Note the DFM gives much more detailed shape information that may not be necessary. In both EDFM and DFM, it would take much longer CPU time if using a finer sampling mesh in a larger domain.

Overall speaking, the trained CNN based solely on the circular-shape scatterers can handle the far-field data of the general shaped objects with various noise levels very well, where the non-circular shaped scatterers were never seen by the trained CNN. A possible explanation of the less satisfactory accuracy in reconstructing the rectangle scatterer is that its far-field pattern is vastly different from that of any nearby circular-shape scatterer, as illustrated in Fig. 10. Note that the far-field pattern of the kite is more similar to that of the circle, thus it leads to better reconstruction results. We would expect much improved reconstruction accuracy if the anticipated non-circular shapes were used in generating the training dataset. This is usually the case when tailored to a specific application where only certain shapes are of interest.

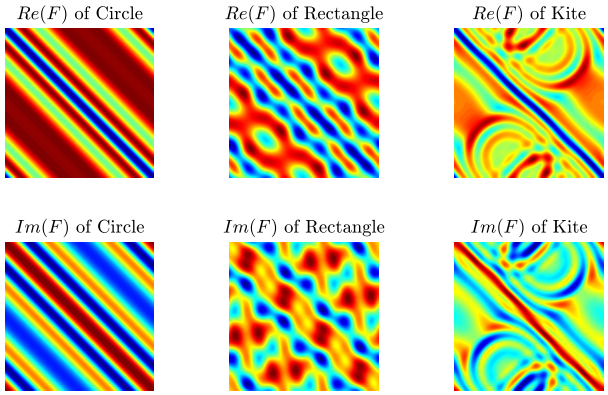

Fig. 10. The jet colormap of noise-free far-field data F of three different scatterers: Circle , Rectangle, and Kite.

V. CONCLUSION

In this paper, we propose a qualitative deep learning method for solving the inverse obstacle scattering problem. After constructing and training the deep convolutional neural network using the randomly generated disk-shape scatterers and the corresponding far-field data matrices, we can obtain a convolutional neural network which reliably maps the noisy far-field data to the center location and radius of the fitting disk that approximates the support of the unknown scatterer. The trained CNN has the advantage of solving the inverse scattering problem in much faster CPU time than other qualitative methods in literature. It takes only a few milliseconds (rather than a few seconds by EDFM and DFM) to estimate an accurate fitting disk with a PC laptop. Our numerical results show that our proposed deep CNN requires a small amount of training data to get very accurate predictions, and it performs very robust with noisy far-field data. The generalization of our proposed method on 2D problems (of finding a disk) to 3D problems (of finding a ball) is straightforward. It is possible to further improve the reconstruction accuracy by training the CNN with the dataset generated from other desired shapes (e.g., ellipse or airplanes). The application of our developed method to the inverse scattering problems with multiple obstacles or limited aperture data is currently undertaken.

REFERENCES

[1] D. Colton and R. Kress, *Inverse Acoustic and Electromagnetic Scattering Theory*, Springer New York, 2012.

[2] X. Liu and B. Zhang, "Recent progress on the factorization method for inverse acoustic scattering problems," *SCIENTIA SINICA Mathematica*, vol. 45, pp. 873–890, 2015.

[3] X. Liu, "A novel sampling method for multiple multiscale targets from scattering amplitudes at a fixed frequency," *Inverse Problems*, vol. 33, no. 8, p. 085011, 2017.

[4] A. Roger, "Newton-Kantorovitch algorithm applied to an electromagnetic inverse problem," *IEEE Transactions on Antennas and Propagation*, vol. 29, no. 2, pp. 232–238, 1981.

[5] G. Giorgi, M. Brignone, R. Aramini, and M. Piana, "Application of the Inhomogeneous Lippmann–Schwinger Equation to Inverse Scattering Problems," *SIAM Journal on Applied Mathematics*, vol. 73, no. 1, pp. 212–231, 2013.

[6] D. Colton, H. Haddar, and M. Piana, "The linear sampling method in inverse electromagnetic scattering theory," *Inverse Problems*, vol. 19, no. 6, pp. S105–S137, 2003.

[7] R. Potthast, "A survey on sampling and probe methods for inverse problems," *Inverse Problems*, vol. 22, no. 2, pp. R1–R47, 2006.

[8] T. Arens and A. Lechleiter, "The linear sampling method revisited," *Journal of Integral Equations and Applications*, vol. 21, no. 2, pp. 179–202, 2009.

[9] F. Cakoni and D. Colton, *A Qualitative Approach to Inverse Scattering Theory*, Springer US, 2013.

[10] A. Kirsch and N. Grinberg, *The Factorization Method for Inverse Problems*, OUP Oxford, 2008.

[11] K. H. Leem, J. Liu, and G. Pelekanos, "An adaptive quadrature-based factorization method for in-

verse acoustic scattering problems," *Inverse Problems in Science and Engineering*, vol. 27, no. 3, pp. 299–316, 2019.

[12] K. H. Leem, J. Liu, and G. Pelekanos, "Efficient Adaptive Qualitative Methods for 3D Inverse Scattering Problems," *ACES Journal*, vol. 33, no. 10, 2018.

[13] K. H. Leem, J. Liu, and G. Pelekanos, "Two direct factorization methods for inverse scattering problems," *Inverse Problems*, vol. 34, no. 12, p. 125004, 2018.

[14] M. Ikehata, E. Niemi, and S. Siltanen, "Inverse obstacle scattering with limited-aperture data," *Inverse Problems and Imaging*, vol. 6, no. 1, pp. 77–94, 2012.

[15] R. Potthast, J. Sylvester, and S. Kusiak, "A range test for determining scatterers with unknown physical properties," *Inverse Problems*, vol. 19, no. 3, pp. 533–547, 2003.

[16] J. Liu and J. Sun, "Extended sampling method in inverse scattering," *Inverse Problems*, vol. 34, no. 8, p. 085007, 2018.

[17] K. H. Leem, J. Liu, and G. Pelekanos, "An extended direct factorization method for inverse scattering with limited aperture data," *Inverse Problems in Science and Engineering*, p. to appear, 2019.

[18] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington DC, 1961.

[19] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, no. 2, pp. 233–243, 1991.

[20] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv preprint*, vol. arXiv:1505.04597, 2015.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[24] H. Kabir, Y. Cao, Y. Cao, and Q. Zhang, "Advances of neural network modeling methods for RF/microwave applications," *ACES Journal*, vol. 25, no. 5, p. 423, 2010.

[25] Z. Wei and X. Chen, "Deep-learning schemes for full-wave nonlinear inverse scattering problems," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 1849–1860, 2019.

[26] H. Yao, M. Li, and L. Jiang, "Applying deep learning approach to the far-field subwavelength imaging based on near-field resonant metalens at microwave frequencies," *IEEE Access*, vol. 7, pp. 63801–63808, 2019.

[27] Y. Khoo and L. Ying, "SwitchNet: a neural network model for forward and inverse scattering problems," *SIAM Journal on Scientific Computing*, vol. 41, no. 5, pp. 3182–3201, 2019.

[28] R. D. Murch, D. G. H. Tan, and D. J. N. Wall, "Newton-Kantorovich method applied to two-dimensional inverse scattering for an exterior Helmholtz problem," *Inverse Problems*, vol. 4, no. 4, pp. 1117–1128, 1988.

[29] J. Adler and O. Öktem, "Solving ill-posed inverse problems using iterative deep neural networks," *Inverse Problems*, vol. 33, no. 12, p. 124007, 2017.

[30] L. Li, L. G. Wang, F. L. Teixeira, C. Liu, A. Nehorai, and T. J. Cui, "DeepNIS: Deep neural network for nonlinear electromagnetic inverse scattering," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 3, pp. 1819–1825, 2018.

[31] Y. Sanghvi, Y. N. G. B. Kalepu, and U. Khankhoje, "Embedding Deep Learning in Inverse Scattering Problems," *IEEE Transactions on Computational Imaging*, 2019.

[32] L. Li, L. G. Wang, D. O. Acero, and F. L. Teixeira, "Deep Convolutional Neural Network Approach for Solving Nonlinear Inverse Scattering Problems," in *2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, pp. 219–220, IEEE, 2019.

[33] Z. Wei and X. Chen, "Physics-Inspired Convolutional Neural Network for Solving Full-Wave Inverse Scattering Problems," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 9, pp. 6138–6148, 2019.

[34] H. Yang, H. Yu, and G. Wang, "Deep learning for the classification of lung nodules," *arXiv preprint*, vol. axXiv:1611.06651v2, 2017.

[35] H. Yang, W.-X. Cong, and G. Wang, "Deep learning for dual-energy X-ray computed tomography," *Proceedings of The 14th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pp. 864–869, 2017.

[36] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.