ResNet Applied for a Single-Snapshot DOA Estimation

Marcio L. Lima de Oliveira
Computer Architecture for Embedded Systems
University of Twente
Enschede, The Netherlands
m.l.limadeoliveira@utwente.nl

Marco J. G. Bekooij
Department of Embedded Software and Signal Processing
NXP Semiconductors, University of Twente
Eindhoven, The Netherlands
marco.bekooij@nxp.com

2022 IEEE Radar Conference (RadarConf22) | 978-1-7281-5368-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/RADARCONF2248738.2022.9763905

***Abstract*—In this paper, we discuss the usage of Residual Neural Networks (ResNets) for calculating the Direction Of Arrival (DOA) of MIMO radars and the estimation of the number of targets, using Minimum Redundancy Array (MRA). In addition, we are considering only the case of one snapshot at a time. This means that most techniques would deliver a poor estimation performance, whereas the Maximum Likelihood Estimation (MLE) algorithm delivers decent performance at a high computational cost.**

ResNet appears to be a viable alternative solution for this problem, being able to outperform MLE in some cases while having a less computational cost for scenarios with at least two different targets.

***Index Terms*—Direction of Arrival, Maximum Likelihood Estimation, Number of Source Estimator, Residual Neural Network**

I. Introduction

Radars are versatile and have been used already for several decades. However, in recent years, they have received additional attention because of the new trend towards autonomous systems - through the development of reliable, low-cost mmWave radars - and driven by Deep Learning (DL) research. The applications of Frequency-Modulated Continuous-Wave (FMCW) radars with Deep Neural Networks (DNN) become more diversified every year. Reducing noise on Range-Doppler maps [1], generating synthetic radar data [2], and object classification in radars [3] are a few examples of this.

Estimating the Direction Of Arrival (DOA) is a well-known problem with many ways to approach it. For this work, we have considered the problem of multiple DOAs with a single snapshot using Minimum Redundancy antenna Array (MRA). Since most of our work was done with MRA, most of our comparisons used Maximum Likelihood Estimation (MLE), as algorithms like MUSIC and ESPRIT would not perform well with this type of scenario [4]. At the end of our work, to facilitate a comparison with the results of other works, we have also considered a Uniform Linear antenna Array (ULA).

The Maximum Likelihood Estimation (MLE) [4]–[6] is a reliable algorithm for estimating the DOA. It works by trying all possible angle combinations that would satisfy the provided signal model and choosing the best angle combination that minimizes the residual error of the arriving snapshot. The limitations of MLE lie in its exponentially increasing computational complexity for multiple sources. Furthermore, according to Häcker and Yang [5], MLE is the most reliable technique for estimating the DOA when using a single snapshot.

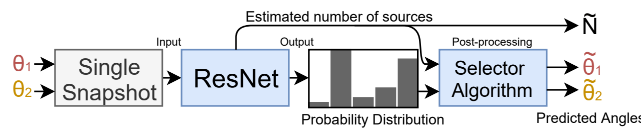

Fig. 1: A simplified diagram of our ResNet model.

More recently, with the successful introduction of Deep Neural Networks (DNN) techniques into the radar domain, some researchers have tried applying Neural Networks (NN) for estimating the DOA. In the researches of both Bialer et. al. [7] and Fuchs et. al. [8], deep Fully Connected Neural Networks (FCNN) were used for estimating the DOA using a single snapshot. Both used a neural network with regression to solve the problem. Bialer et. al. [7] used Root Mean Squared Error (RMSE) as a custom loss function to calculate the error for the regression and could solve the DOA problem up to four angles, while Fuchs et. al. [8] used both the average and distance between two targets as a way to calculate the error for the regression and solve the DOA problem for two sources. Another work, by Elbir [9], used regression, several snapshots, and various Convolutional Neural Networks (CNN) in parallel to achieve a MUSIC-like spectrum and solve the DOA problem for up to six distinct angles.

In this paper, we propose a novel ResNet application to estimate single-snapshot DOAs and to estimate the number of targets. Fig. 1 shows a simplified diagram of our design. Unlike previous authors, our method relies on classification rather than regression. In addition, unlike previous techniques, we do not always assume that the number of sources is previously known before estimating the DOA. Our experiments suggest that ResNet achieves at least similar performance to MLE for both MRA and ULA, outperforming it in some scenarios.

II. Signal Model

Consider M antennas uniformly separated (ULA), with the distance between two antennas being d. Moreover, considering that all N sources are far-field, respecting the condition of

978-1-7281-5368-1/22/$31.00 ©2022 IEEE

$0 \leq N \leq M-1$. The received snapshot $x \in \mathbb{C}^M$ follows the expression:

$$x = \mathbf{A}(\theta) \cdot s + \eta \,, \tag{1}$$

where $s \in \mathbb{C}^N$ is the signal emitted by the sources, $\eta \in \mathbb{C}^M$ is the added Gaussian noise, and $\mathbf{A}(\theta) \in \mathbb{C}^{M \times N}$ is the steering matrix (also known as the beamforming matrix). The steering matrix is a function of the angles of arrival $\theta = \theta_1, ..., \theta_N$, and it is expressed as:

$$\mathbf{A}(\theta) = \begin{bmatrix} e^{j2\pi\lambda^{-1}d\theta_1} & ... & e^{j2\pi\lambda^{-1}d\theta_N} \\ ... & & ... \\ e^{j2\pi\lambda^{-1}Md\theta_1} & ... & e^{j2\pi\lambda^{-1}Md\theta_N} \end{bmatrix}, \tag{2}$$

in which λ is the radar's wavelength, $d = \lambda/2$, and the positions of each antenna of the array vary following $d, 2d, ..., Md$.

A. Maximum Likelihood Estimation

The Maximum Likelihood Estimation (MLE) algorithm is commonly used for estimating the DOA when using a single snapshot, having a good performance at it [5]. MLE tries all possible angle combinations to find the most likely answer for the problem - the angle combination that provides the least amount of residual error.

Consider $\varepsilon(\theta) \in \mathbb{C}^M$ as the residual error vector and $\tilde{\mathbf{A}}(\theta)$ as a certain estimated steering matrix. In short, the MLE algorithm attempts to minimize its residual error by trying different matrices $\tilde{\mathbf{A}}(\theta)$ containing different combinations of angles:

$$\varepsilon(\theta) = x - \tilde{\mathbf{A}}(\theta) \cdot (\tilde{\mathbf{A}}^{+}(\theta) \cdot x) \,, \tag{3}$$

where $\tilde{\mathbf{A}}^{+}(\theta)$ is the pseudo-inverse matrix of $\tilde{\mathbf{A}}(\theta)$. The estimated DOAs $\tilde{\theta} = \tilde{\theta}_1, ..., \tilde{\theta}_N$ can be calculated by the minimal residual error:

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmin}} \, ||\varepsilon(\theta)||^2 \,, \tag{4}$$

in which $\varepsilon_{\min} = \min\{\varepsilon(\theta)\}$ is the minimal residual noise obtained after repeating eq. 3 several times and finding its minimal value.

The number of times the MLE algorithm repeats eq. 3 depends on the size of the search grid. For example, using a Field of View (FOV) of 180 degrees, a resolution of 1 degree, 8 antennas, and 3 targets, the number of times the MLE algorithm will need to pseudo-invert the matrix $\tilde{\mathbf{A}}(\tilde{\theta}) \in \mathbb{C}^{8 \times 3}$ is nearly a million times. Hence, this algorithm is very computationally intensive when estimating the DOA for more than 2 sources.

The MLE algorithm can also be used for calculating the estimated number of sources $\tilde{N}$ (shown in Algorithm 1) if the scene noise η is previously known. For this, the algorithm first assumes that there is only one angle, $\tilde{N} = 1$. The 1D MLE finds the angle that produces the least amount of residual error. If the difference between the residual error $\varepsilon_{1\min}$ and the known scene noise η is too large, then there is (probably) more than 1 source in the scene. Sequentially, the algorithm assumes that two angles of arrival exist and calculates these angles and residual error by running a 2D MLE. If the difference between them is still too big, it assumes there is still another source. The same happens for 3D MLE. If there are more than 3 targets, the DOAs are not calculated due to the computational intensity. If $||\eta - \varepsilon_{\min}|| < c_1$, then we can consider that $\eta \approx \varepsilon_{\min}$, where c_1 is small and found empirically.

Algorithm 1 Number of Sources Estimator using MLE

$\tilde{N} = 1$
$[\varepsilon_{1\min}; \tilde{\theta}_1] = \text{MLE}_{1D}(x)$
if $||\eta - \varepsilon_{1\min}|| > c_1$, **then**
  $\tilde{N} = 2$
  $[\varepsilon_{2\min}; \tilde{\theta}_1, \tilde{\theta}_2] = \text{MLE}_{2D}(x)$
  if $||\eta - \varepsilon_{2\min}|| > c_1$, **then**
    $\tilde{N} = 3$
    $[\varepsilon_{3\min}; \tilde{\theta}_1, \tilde{\theta}_2, \tilde{\theta}_3] = \text{MLE}_{3D}(x)$
    if $||\eta - \varepsilon_{3\min}|| > c_1$, **then**
      $\tilde{N} > 3$
    end if
  end if
end if

III. RESNET - DATA AND ARCHITECTURE

In this paper, we propose the usage of a Residual Network (ResNet) for estimating both the DOA and the number of sources. Therefore, it is essential to understand how this technique works and how it processes data.

A. ResNet Data

ResNet was first introduced as an image recognition technique [10]. Since images are 3D matrices (the third dimension being used for colors), we have used something similar as the input for our network, adapting the received snapshot x. As in the study by Elbir [9], we have used the covariance matrix $\mathbf{R} \in \mathbb{C}^{M \times M}$ as the input to our NN. For our case, since the complete covariance matrix is not available for real applications, we can approximate the covariance matrix as:

$$\mathbf{R} = x \cdot x^H, \tag{5}$$

where x^H is the conjugate transpose (or Hermitian transpose) of the snapshot x.

To obtain the normalized NN's input $\mathbf{X} \in \mathbb{R}^{M \times M \times 2}$, we must separate the real and imaginary parts of $\mathbf{R}$, store them separately into the third dimension of the 3D matrix $\mathbf{X}$ and then normalize it:

$$\begin{matrix} \mathbf{X}_{i,j,1} = \text{Re}\{\mathbf{R}_{i,j}\}/c_2 \\ \mathbf{X}_{i,j,2} = \text{Im}\{\mathbf{R}_{i,j}\}/c_2 \end{matrix} \,, \tag{6}$$

where $i = 1, ..., M$ and $j = 1, ..., M$. The constant c_2 is used to normalize the input. This value was obtained empirically as a number that would always keep all values of $\mathbf{X}$ between -1 and 1.

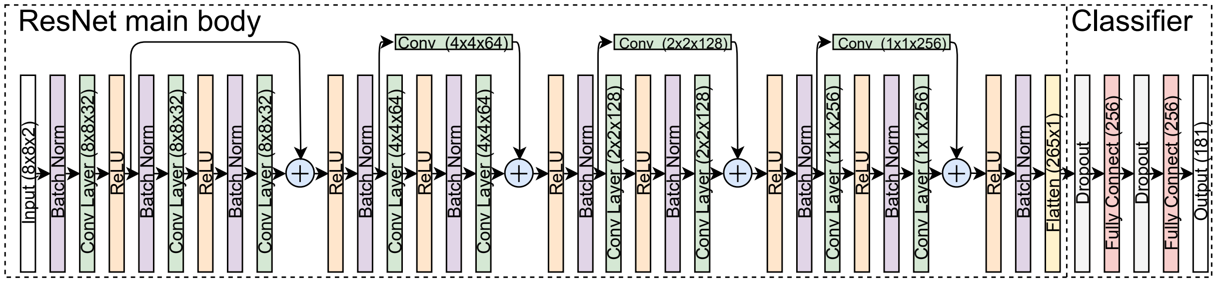

Fig. 2: The architecture of our proposed ResNet applied to the DOA problem.

Our output layer is a fully connected dense layer, where each neuron represents one possible angle, and the value of this neuron represents the probability that a source exists at that possible angle. Unlike previous papers [7], [8], we do not perform a regression but a classification. The advantage of doing this is that we do not need to alter the neural network to produce results for different numbers of sources N. One disadvantage is the need for post-processing to analyze the probability distribution coming out of the neural network. Another disadvantage is the need for retraining if either the FOV or resolution changes. In our experiments, the advantages outweighed the disadvantages, and our testings using classification were superior to the test models using regression. Since each neuron is related to a possible angle, the number of neurons in the output layer depends on the desired FOV and resolution. Therefore, the output of the NN is a vector y containing $(\mathrm{FOV}/\mathrm{resolution}) + 1$ elements. Each element of this output y corresponds to a single neuron output. In this research, we have used 181 neurons for MRA (FOV of 180° and a resolution of 1°) and 201 neurons for ULA (FOV of 20° and a resolution of 0.1°).

B. ResNet Architecture

Our ResNet was inspired as a reduced and adapted version of the original network proposed by He et. al. [10]. Our version has 1 identity block, 3 convolution blocks, and 2 fully connected layers. Through our testing, we have found this light architecture to produce the fewest of errors (difference between the estimated and actual angle of arrival). Fig. 2 shows our proposed neural network, as we used it for MRA, with an output of 181 neurons and an input of $8 \times 8 \times 2$. For our ULA experiments, the input used was $16 \times 16 \times 2$, and we had 201 neurons at the output.

The main body of the ResNet contains the convolutional layers, and it is where the feature extraction happens. The classifier is composed of fully connected layers. It interprets the extracted features and separates them into various classes - angles in our case. The output y of the ResNet is a probability distribution, where each output neuron has a value between 0 and 1. The closest to 1, the higher the probability that a target exists at that particular angle. It is important to note that this probability measures how strongly the NN believes a target exists (or not) at a certain angle, not being a real probability representation of the problem.

C. Post-Processing

After obtaining the probability distribution as the output y of the NN, a post-processing step is required for selecting the most probable DOAs. The post-processing step selects the $\hat{N}$ most likely DOAs from the probability distribution given a previously known ($\hat{N} = N$) or estimated ($\hat{N} = \tilde{N}$) number of sources. In the beginning, the most probable angles $\hat{\theta} = \hat{\theta}_1, ..., \hat{\theta}_{2\hat{N}}$ contain the estimated angles $\tilde{\theta} = \tilde{\theta}_1, ..., \tilde{\theta}_{\hat{N}}$. In other words, $\tilde{\theta} \in \hat{\theta}$. After all steps, all inadequate angles are deleted, and the result is that $\tilde{\theta} = \hat{\theta}$.

The post-processing algorithm works by following this sequence:

1) Get the $2\hat{N}$ most probable angles $\hat{\theta} = \hat{\theta}_1, ..., \hat{\theta}_{2\hat{N}}$ of y.
2) Delete any angle that is at least 100 times smaller than the most strong probable angle.
3) Find $\hat{N}$ peaks in $\hat{\theta}$.
4) If the remaining size of $\hat{\theta}$ is bigger than $\hat{N}$, delete the less likely angles until reaching $\hat{N}$.
5) If the size of $\hat{\theta}$ is smaller than $\hat{N}$, use the most likely angles that are in the vicinity of the peaks and belong to the $2\hat{N}$ most probable angles.
6) If the size of $\hat{\theta}$ is still smaller than $\hat{N}$, repeat the most strong peak until reaching $\hat{N}$.

Another way to illustrate the dynamic between ResNet and the post-processing step is through Fig. 3, which shows the output of ResNet with its probability distribution, the most probable angles, and the final estimated angles. In this example, $\hat{N} = 3$, therefore $2\hat{N} = 6$. The correctly estimated DOAs were $-33°$, 40°, and 45°. It is interesting to note that the $-33°$ angle was correctly selected (because it is a peak), even though the 44° angle has a stronger neuron output.

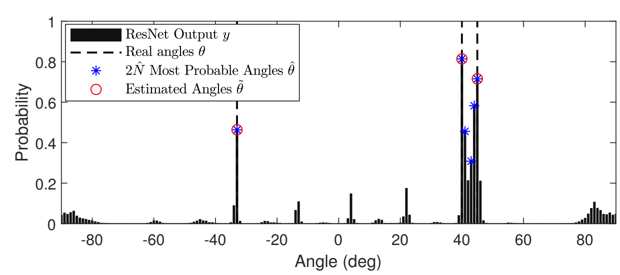

Fig. 3: A ResNet output and post-processing example.

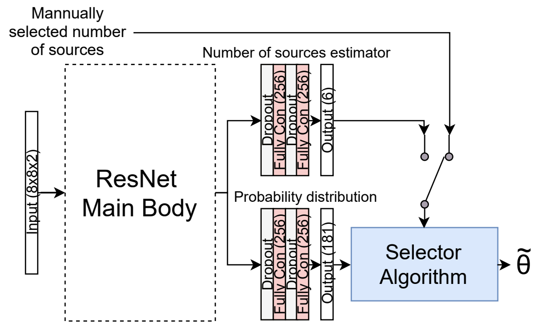

Fig. 4: The number of sources estimator architecture with ResNet.

D. Estimating Number of Sources

In addition to estimating DOAs, we have considered the estimation of the number of sources. As seen in Fig. 4, instead of using a new neural network or another solution, we have just created a new classifier with new fully connected layers, using the features extracted by the main body of the ResNet.

Our Number of Source Estimator (NSE) has 6 neurons in the output layer, meaning that it can estimate the following number of targets: $\tilde{N} = 0$, 1, 2, 3, 4, or $>$4 sources. It is important to note that our estimator considers the absence of sources ($\tilde{N} = 0$) and that anything above 4 targets is classified in only one category, without distinction ($\tilde{N} > 4$).

IV. Evaluation

We have performed various experiments during this research. Firstly we tested the best deep learning architecture. Moreover, we have compared ResNet with MLE in different scenarios. For the evaluation, we have considered a few metrics such as Root Mean Square Error (RMSE), percentage of angles estimated correctly within resolution (accuracy), and percentage of estimated DOAs that are too far from the correct answer (outliers).

Most researchers [7]–[9] estimating the DOA tend to choose Root Mean Square Error (RMSE) as at least one of their metrics. In our work, this was no different. The RMSE used by our research is given by:

$$\mathrm{RMSE} = \frac{1}{L}\sum_{l=1}^{L}\sqrt{\frac{1}{K \times N}\sum_{k=1}^{K}\sum_{n=1}^{N}(\theta_{n,k} - \tilde{\theta}_{n,k,l})^2}, \quad (7)$$

where K is the number of samples, N is the number of sources, L is the number of Monte-Carlos trials, $\tilde{\theta}_{n,k,l}$ is the estimated angle, and $\theta_{n,k}$ is the actual angle. For the situation where the number of sources is unknown, N is replaced by $\tilde{N}$.

One of the problems of RMSE is that it does not show the complete picture. A higher RMSE may be due to a few incorrect but far away angles, while a lower RMSE could be due to many incorrect but closer angles. That is why we also use accuracy as a metric. We define in this paper accuracy as the percentage of angles that are correctly estimated within the resolution. If the resolution of 1° is used, anything below 0.5° should be considered a correct estimate. In addition, we also analyzed the percentage of what we called outliers. In most cases, it makes no difference if a particular target is incorrectly estimated by 15 or 50 degrees. Therefore, in our work, any error (difference between the estimated and the actual angles) above $(5^\circ \times \mathrm{resolution})$ is considered an outlier.

TABLE I: Different DNN architectures and configurations.

DNN Type	**Number of Layers**	**Accuracy for 2 angles with SNR = 30 dB***
MLE_{2D}	-	88.9 %
FCNN	8 FC Layers	82.29 %
CNN	4 Conv. + 2 FC Layers	79.7 %
ResNet	12 Conv. + 2 FC Layers	90.1 %

A. Evaluating Different Architectures

For the first experiment, the training was done with a reduced dataset, so the time to train and evaluate could be shortened. Therefore this does not show the full potential of the neural networks. Although, since they are scalable, better performance with a limited dataset should also mean better performance with a larger dataset. For this experiment, we have used 8 antennas in MRA configuration (where the distance between antennas are: 0, 1, 4, 6, 13, 14, 17, 19 $\times\lambda/2$), single-snapshot, a resolution of 1 degree, a FOV of 180 degrees, and 2 targets. The neural networks were evaluated using 10 thousand data points (K) and 100 Monte-Carlo trials (L). The number of sources was previously known. In Table I, we can see that the ResNet has the best performance when compared to the Fully Connected Neural Network (FCNN) and Convolutional Neural Network (CNN). It is important to note that, although ResNet has more layers, it is less computationally demanding than CNN (multiply-accumulate: 5.1×10^6 and 2.5×10^7, respectively) and a bit more demanding than FCNN (multiply-accumulate: 3.3×10^6) due to its architecture.

B. Experiments using MRA Antenna Array

The following experiments use the same parameters as the first experiment, 8 antennas in the same MRA configuration. However now, our ResNet was trained with a generated dataset with 40 million data points, containing from 0 to 5 targets, amplitudes from sources varying from 0.5 to 1, and SNR ranging between 0 and 30 dB. Although 40 million might seem like a large number, when considering all possible scenarios, it just represents a small fraction of all possibilities. Hence, the NN needs to learn how to generalize during training.

Due to the complexity of MLE, we had to reduce its FOV (the FOV for ResNet is always 180° for MRA) and the amount of evaluating data as the number of sources increased in order to produce the results within a feasible amount of time.

As shown in Fig. 5, our proposed model performs well in a single-snapshot MRA environment. When considering one source, our NN had a performance close to that of MLE. For two sources, our model performed better than MLE. For three sources, the ResNet was better for low SNR levels and

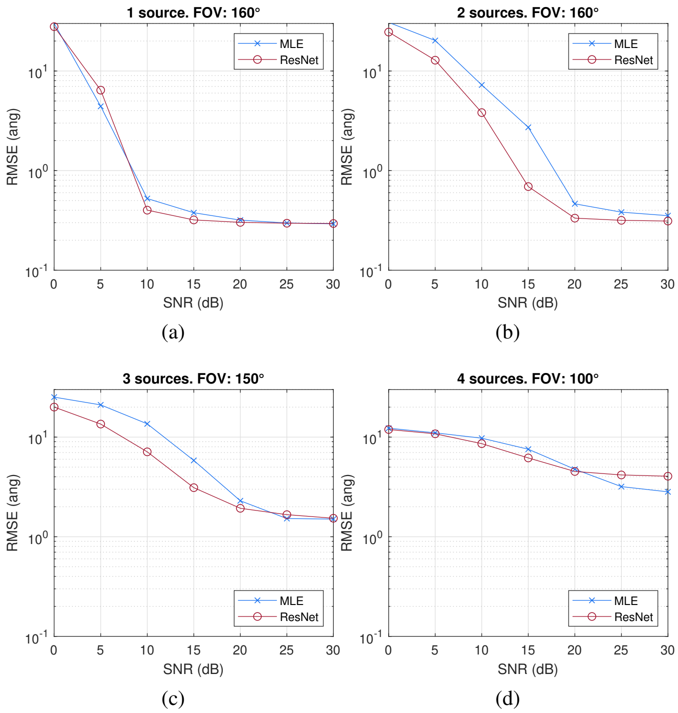

Fig. 5: RMSE vs. SNR for MLE and our proposed ResNet model between 1 and 4 sources. For this experiment, the number of sources was previously known. (a) 1 source, FOV of 160°, and 10,000 data points. (b) 2 sources, FOV of 160°, and 10,000 data points. (c) 3 sources, FOV of 150°, and 8,000 data points. (d) 4 sources, FOV of 100°, and 5,000 data points.

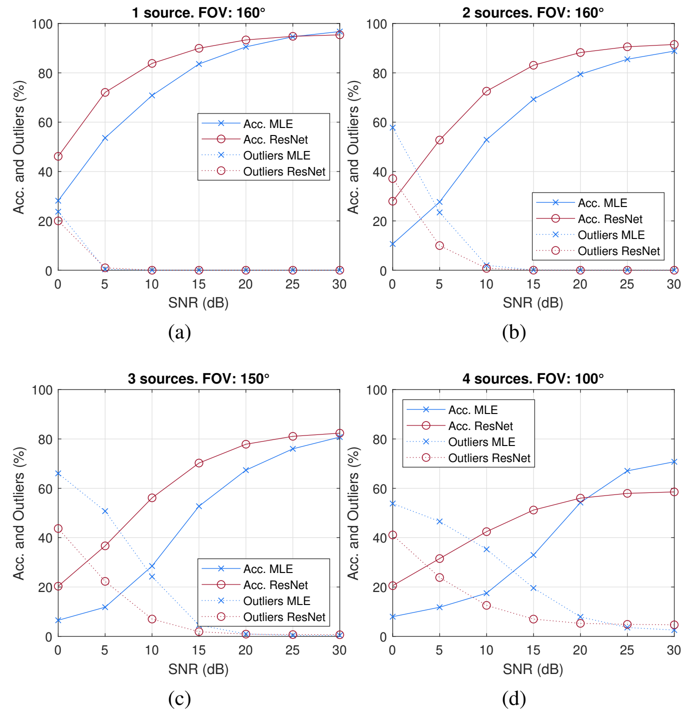

Fig. 6: Accuracy and Outliers vs. SNR for MLE and our proposed ResNet model between 1 and 4 sources. For this experiment, the number of sources was previously known. (a) 1 source, FOV of 160°, and 10,000 data points. (b) 2 sources, FOV of 160°, and 10,000 data points. (c) 3 sources, FOV of 150°, 8,000 data points. (d) 4 sources, FOV of 100°, and 5,000 data points.

slightly worse for high SNR levels. For four sources, the neural network was slightly better in low SNR and slightly worse in high SNR. In general, our model performs well under high noise levels and quite similarly to MLE when noise decreases.

In Fig. 6, we can see both accuracy and outliers for different scenarios. The solid line represents the accuracy (percentage of angles that are correctly estimated within the resolution), and the dotted line represents the percentage of outliers (when the model incorrectly estimates a source by more than 5 degrees). The trends observed in Fig. 5 are also repeated in Fig. 6. However, there is an exception. For three sources and high SNR, the neural network is more accurate, but due to having more outliers than MLE, it has a slightly worse RMSE.

For the following experiment (shown in Fig. 7), we have used the same scenario as the previous experiment, now, however, the number of sources is previously unknown, and we also need to estimate it. In this experiment, we compared the performance of the ResNet architecture, shown in Fig. 4, with Algorithm 1 that uses MLE. Due to the complexity of MLE, it was not feasible to apply Algorithm 1 for more than three sources. Graphs (a), (b), and (c) of Fig. 7 have two vertical axes. One for RMSE in semilog (left, full line) and the other for the correctly estimated number of sources, given in percentage (right, dashed line). Graph (d) has the accuracy of the Number of Source Estimator (NSE) for both the neural network (Fig. 4) and the maximum likelihood estimation (Algorithm 1).

In Fig. 7, we can observe that the RMSE performance of ResNet is similar but slightly higher than the one shown in Fig. 5. On the other hand, the performance of ResNet when estimating the number of targets is much superior to when using MLE and Algorithm 1. It is also important to note that, although the MLE is worse than ResNet when performing the number of targets estimation, its RMSE is not. This happens because eq. 7 only considers whether any of the estimated angles are similar to the real angles. For example, Algorithm 1 could have detected the presence of three targets when only one is present. However, if any of the three estimated angles are similar to the real target angle, the RMSE value would be the same as if the number of targets would have been correctly estimated.

C. Experiments using ULA Antenna Array

As seen before in Fig. 5, our model performs well for single-snapshot MRA. Moreover, this is also true for single-snapshot ULA. For this scenario, we tried to replicate the results of the work from Bialer et. al. [7], although our attempts were unsuccessful. Therefore, we have replicated the conditions of

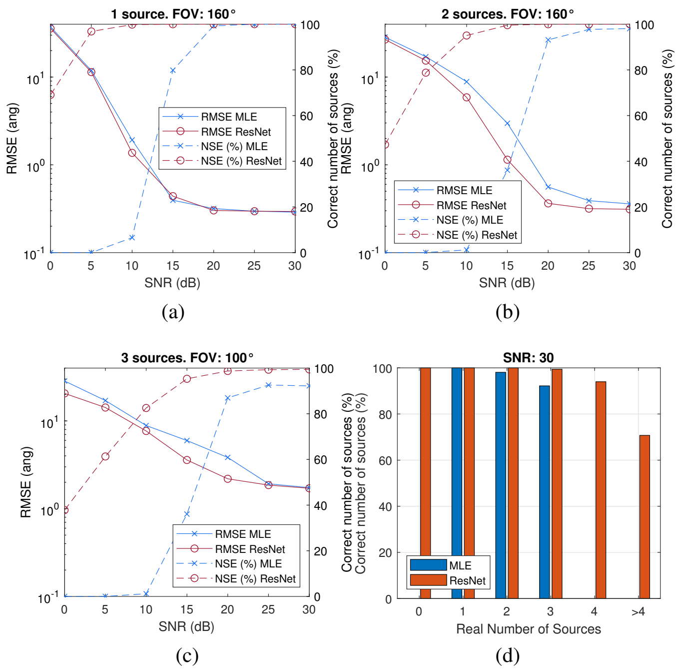

Fig. 7: The left vertical axis is RMSE vs. SNR. The right vertical axis is the percentage of the correctly estimated number of sources vs. SNR. The number of sources is unknown. The number of actual sources varies between 1 and 3. The number of sources was estimated using a Number of Source Estimator (NSE). (a) 1 source, FOV of 160°, and 10,000 data points. (b) 2 sources, FOV of 150°, and 8,000 data points. (c) 3 sources, FOV of 100°, and 3,000 data points. (d) Bar graph showing the accuracy of the NSE vs. the actual number of sources.

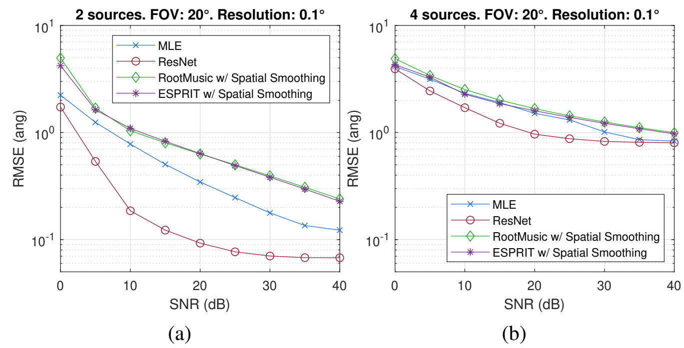

Fig. 8: RMSE vs. SNR for ULA, comparing 3 algorithms and our model. The number of sources is previously known. (a) 2 sources, FOV of 20°, resolution of 0.1 °, and 10,000 data points. (b) 4 sources, FOV of 20°, resolution of 0.1 °, and 1,000 data points.

their work and used our own MLE, RootMUSIC with spatial smoothing, and ESPRIT with spatial smoothing algorithms [4] to compare the performance of our model. For this experiment, we have retrained our model with 12 million data points containing: 16 uniformly separated antennas (which means that this input was $\mathbf{X} \in \mathbb{R}^{16\times16\times2}$), separated by 0.5λ, FOV of 20°, resolution of 0.1° (giving us 201 neurons in the output layer), 0 to 5 targets, amplitudes varying from 0.5 to 1, and SNR ranging between 0 and 30 dB.

In this last experiment, the results (seen in Fig. 8) show that our model outperformed all three other algorithms, even though MLE performs similarly for 4 targets and high SNR.

V. Conclusion

In this paper, we have proposed a novel approach to solve the direction of arrival problem for a single-snapshot using a Residual Neural Network, with both known and unknown numbers of targets. In this work, we have explored different fields of views, resolutions, and antennas configurations. Both Minimum Redundancy Linear Arrays (MRA) and Uniform Linear Array (ULA) have been analyzed in our work.

The results of our proposed ResNet look promising. Our model was able to outperform MLE, especially in high noise situations, and when the number of targets is estimated. This means that, at the very least, our proposed model has a performance similar to the Maximum Likelihood Estimation Algorithm, and it is superior when estimating the number of targets. The shortcoming of ResNet lies in its necessity of a fast computer for the training process, although, after training, the computational requirements are lower than MLE and do not change.

For future work, we would like to explore the reasons and causes that make, in our results, ResNet superior to MLE in some scenarios.

References

[1] M. L. Lima de Oliveira and M. J. G. Bekooij, "Deep convolutional autoencoder applied for noise reduction in range-Doppler maps of FMCW radars," *IEEE International Radar Conference (RADAR 2020)*, pp. 630–635, 2020.

[2] M. L. Lima De Oliveira and M. J. G. Bekooij, "Generating Synthetic Short-Range FMCW Range-Doppler Maps Using Generative Adversarial Networks and Deep Convolutional Autoencoders," *IEEE Radar Conference (RadarConf20)*, pp. 1–6, 2020.

[3] K. Aziz, E. De Greef, M. Rykunov, A. Bourdoux, and H. Sahli, "Radar-camera Fusion for Road Target Classification," *IEEE International Radar Conference (RADAR 2020)*, vol. 2020-Septe, 2020.

[4] H. L. Van Trees, *Optimum Array Processing*, 2002, vol. 4.

[5] P. Häcker and B. Yang, "Single snapshot DOA estimation," *Advances in Radio Science*, vol. 8, pp. 251–256, 2010.

[6] P. Stoica and K. C. Sharman, "Maximum Likelihood Methods for Direction-of- Arrival Estimation," *IEEE TRANSACTIONS ON ACOUSTICS. SPEECH, AND SIGNAL PROCESSING*, vol. 38, no. 7, pp. 1132–1143, 1990.

[7] O. Bialer, N. Garnett, and T. Tirer, "PERFORMANCE ADVANTAGES OF DEEP NEURAL NETWORKS FOR ANGLE OF ARRIVAL ESTIMATION," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3907–3911.

[8] J. Fuchs, R. Weigel, and M. Gardill, "Single-snapshot direction-of-arrival estimation of multiple targets using a multi-layer perceptron," *2019 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility, ICMIM 2019*, pp. 0–3, 2019.

[9] A. M. Elbir, "DeepMUSIC: Multiple Signal Classification via Deep Learning," *IEEE Sensors Letters*, vol. 4, no. 4, 2020.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.