

# Arquitetura Multi-DeepNet para Localização com Baixo SNR e Poucos Snapshots

## Rede Neural Convolutacional

Paulo Rogério Alfredo Cândido Junior

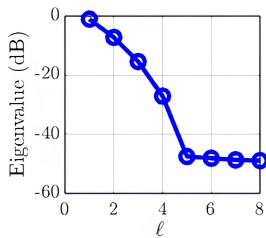
Department of Electrical Engineering.  
State University of Londrina

March 27, 2025

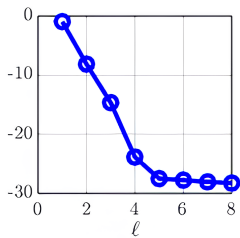
Técnicas tradicionais como MUSIC apresentam degradação expressiva de desempenho devido a sensibilidade ao ruído durante a decomposição espectral (*threshold effect*). Logo torna-se desafiador realizar o sensoriamento para;

- Condição severa de baixa latência, (apenas um snapshot)
- Fontes que transmitem poucos miliwatts. (IoT)
- Cenário com SNR muito baixa

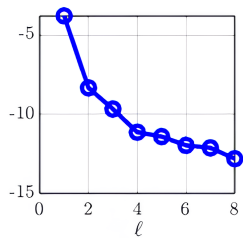
Por mais que o sistema seja determinístico devido a linha de visada (LoS), é de difícil inversão, altamente não linear, com ambiguidade e interferência de ruído.



(a)  $\sigma_n^2 = -90$  dBm



(b)  $\sigma_n^2 = -70$  dBm



(c)  $\sigma_n^2 = -50$  dBm

**Figura:** Diferença entre autovalores consecutivos para 8 antenas, com 4 fontes. Para uma SNR elevada, podemos realizar essa estimativa de forma simples, através da separação natural entre os autovetores.

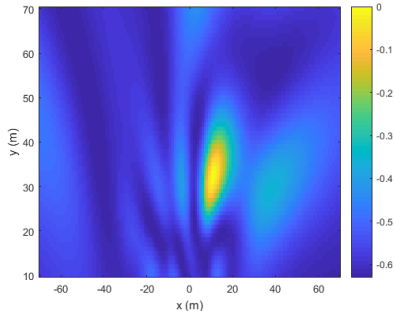
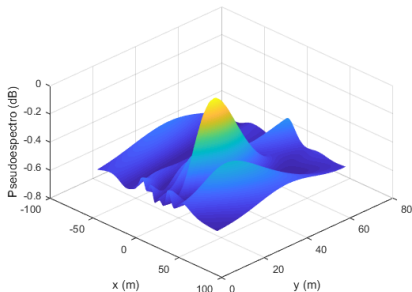
A solução proposta é baseada em aprendizagem supervisionada usando *Deep Learning*, adaptando o método Multi-DeepNet originalmente desenvolvido para radares FMCW-MIMO (Su et al., 2023), especificamente para o contexto de localização multiusuário com um conjunto de antenas linear (ULA). A abordagem atende diretamente a necessidade de soluções robustas sob baixa SNR, necessidade de baixa latência e multiuser.



Su, Yunye and Lan, Xiang and Shi, Jinmei and Sun, Lu and Wang, Xianpeng

*Fast target localization in FMCW-MIMO radar with low SNR and snapshot via multi-DeepNet.*

*Remote Sensing*, vol. 15, no. 1, pp. 66, 2022.



**Figura:** o pseudospectro MUSIC (função objetivo) obtido a partir do subespaço de ruído mostra uma resposta complexa e não convexa, para uma SNR de 3dB e uma única fonte. Embora exista um único máximo global que representa a posição correta da fonte, a presença de máximos locais e a topologia ondulada da função dificultam a inversão direta do modelo, principalmente em ambientes ruidosos.

- Aprendizagem supervisionada; o treinamento é feito usando dados rotulados.
- Aprendizagem passiva; já possui os pares rotulados definidos.
- Tarefa preditiva de classificação; divide o espaço de busca em  $N$  regiões discretas e classifica se há alguma fonte em cada uma.
- Aprendizagem offline; temos um banco de dados estático, de forma que o agente aprende de uma vez só.
- Rede Neural Convolutacional (CNN).
- Aquisição dos sinais recebidos no arranjo de antenas, que formam a matriz de covariância ( $\mathbf{R}_Y$ ) necessária para entrada da rede.
- Construção de um tensor tridimensional  $\mathcal{T} \in \mathbb{R}^{M \times M \times 3}$  para entrada da CNN.

- Uma CNN inicial classifica a região ampla onde os usuários estão situados, fornecendo uma estimativa grosseira.
- Na saída teremos um vetor de probabilidades com  $N$  classes representando a distribuição de confiança sobre as regiões.
- Utiliza-se uma segunda rede CNN para o refinamento, onde realizamos uma classificação com alta granularidade, sobre  $M$  regiões, onde  $M \gg N$ .

Figura: Busca grosseira primeira etapa.

**Tabela:** PARÂMETROS DO SISTEMA UTILIZADO NO ARTIGO

Parâmetro	Valor
Realizações por SNR	$\mathcal{M} = 5 \cdot 10^5$
Snapshots por amostra	$N = 10$
Número de antenas	$M = 10$ (ULA)
Espaçamento entre elementos	$\delta = \frac{\lambda}{2}$
Comprimento de onda	$\lambda = \frac{c}{f_c}$
Frequência portadora	$f_c = 78,74$ GHz
Potência do sinal transmitido	$\mathcal{P} = 30$ mW
Expoente de perda de percurso	$\alpha = 2$
Intervalo de SNR utilizado	$\text{SNR} \in [-20:2:20]$ dB
Intervalo de ângulos de incidência	$\theta_k \sim \mathcal{U}(-60^\circ, 60^\circ)$
Intervalo de distâncias	$R_k \sim \mathcal{U}(1, 10)$ m
Número de alvos por amostra	$K \sim \mathcal{U}\{1, 6\}$
Quantização angular (coarse)	$10^\circ$ por bin



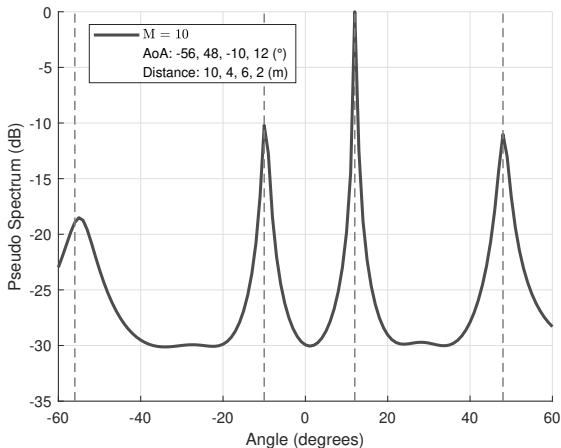
Na entrada será utilizado dados sintéticos de sinais recebidos no ULA.  
Utilizando *MATLAB* Montamos a matriz de sinais recebidos ( $\mathbf{Y}$ ) a partir de séries temporais de sinais recebidos em cada antena;

$$\mathbf{Y} = \mathbf{A}\mathbf{S} + \mathbf{N}, \quad \mathbf{N} \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2 \mathbf{I}_M)$$

## Modelagem da matriz de sinais recebidos ( $\mathbf{Y}$ );

```
function [Y, Z] = signals(M, snapshots, delta, ...
    lambda, AoA, numSources, d, SNRdB)
    PL = (lambda/(4*pi)).^2 ./ (d.^2);
    beta = sqrt(PL);
    H = zeros(M, numSources);
    for s = 1:numSources
        a = responsearray(M, delta, lambda, AoA(s));
        H(:,s) = beta(s) * a;
    end
    X = (randn(numSources, snapshots) + ...
        1j*randn(numSources, snapshots)) / sqrt(2);
    Y_sig = H * X;
    P_signal = mean(abs(Y_sig).^2, 'all');
    noiseVar = P_signal / (10^(SNRdB/10));
    Z = sqrt(noiseVar) * (randn(M, snapshots) + ...
        1j*randn(M, snapshots)) / sqrt(2);
    Y = Y_sig + Z;
end

function a = responsearray(M, delta, lambda, theta)
    gamma = 2*pi * delta / lambda;
    a = exp(-1j * gamma * (0:M-1)' * sind(theta));
end
```



**Figura:** Pseudo espectro normalizado em decibéis obtido a partir da modelagem dos sinais, onde as linhas verticais tracejadas em cinza indicam os ângulos de chegada (AoA) verdadeiros dos alvos. A coincidência entre os picos do pseudoespectro e os AoAs valida a geração dos sinais recebidos no ULA.

Project - MultiDeepNet-FMCW-MIMO				
Views	All	Project (36)	Modified (39)	Layout: Tree
Files				
Dependency Analyzer				
Labels				
Classification				
Git				
Current branch: master				
Branch status: Normal				
Ahead of /origin/master				
Details				

Name ^	Status	Classification	Git
datasets	✓		•
dataset_coarse_SNR-02.mat	✓	Dataset	●
dataset_coarse_SNR-04.mat	✓	Dataset	●
dataset_coarse_SNR-06.mat	✓	Dataset	●
dataset_coarse_SNR-08.mat	✓	Dataset	●
dataset_coarse_SNR-10.mat	✓	Dataset	●
dataset_coarse_SNR-12.mat	✓	Dataset	●
dataset_coarse_SNR-14.mat	✓	Dataset	●
dataset_coarse_SNR-16.mat	✓	Dataset	●
dataset_coarse_SNR-18.mat	✓	Dataset	●
dataset_coarse_SNR-20.mat	✓	Dataset	●
dataset_coarse_SNR+00.mat	✓	Dataset	●
dataset_coarse_SNR+02.mat	✓	Dataset	●
dataset_coarse_SNR+04.mat	✓	Dataset	●
dataset_coarse_SNR+06.mat	✓	Dataset	●
dataset_coarse_SNR+08.mat	✓	Dataset	●
dataset_coarse_SNR+10.mat	✓	Dataset	●
dataset_coarse_SNR+12.mat	✓	Dataset	●
dataset_coarse_SNR+14.mat	✓	Dataset	●
dataset_coarse_SNR+16.mat	✓	Dataset	●
dataset_coarse_SNR+18.mat	✓	Dataset	●
dataset_coarse_SNR+20.mat	✓	Dataset	●

**Figura:** Datasets gerados, totalizando uma base de dados com 500,000 dados rotulados.

```

edgesCoarse = linspace(-60,60,13); nCoarse = numel(edgesCoarse)-1;
for iSNR = 1:numel(SNRdB_list)
    snr_dB = SNRdB_list(iSNR);
    Tcoarse = zeros(M, M, 3, nSamples, 'single');
    Ylabel = false(nCoarse, nSamples);
    for idx = 1:nSamples
        K = randi([1 maxSources]);
        AoA = (rand(K,1)*120 - 60); dist = rand(K,1)*9 + 1;
        [Y, ~] = signals(M, snapshots, delta, ...
            lambda, AoA, K, dist, snr_dB);
        Rs = (Y * Y') / snapshots;
        Tcoarse(:,:,1,idx) = real(Rs);
        Tcoarse(:,:,2,idx) = imag(Rs);
        Tcoarse(:,:,3,idx) = angle(Rs);
        label = false(nCoarse,1);
        for k = 1:K
            bin = find(AoA(k) >= edgesCoarse, 1, 'last');
            if bin>=1 && bin<=nCoarse
                label(bin) = true;
            end
        end
        Ylabel(:,idx) = label;
    end
    fname = sprintf('dataset_coarse_SNR%+03d.mat', snr_dB);
    save(fname, 'Tcoarse', 'Ylabel', '-v7.3');
    fprintf('Salvo %s (nSamples=%d)\n', fname, nSamples);
end

```

end



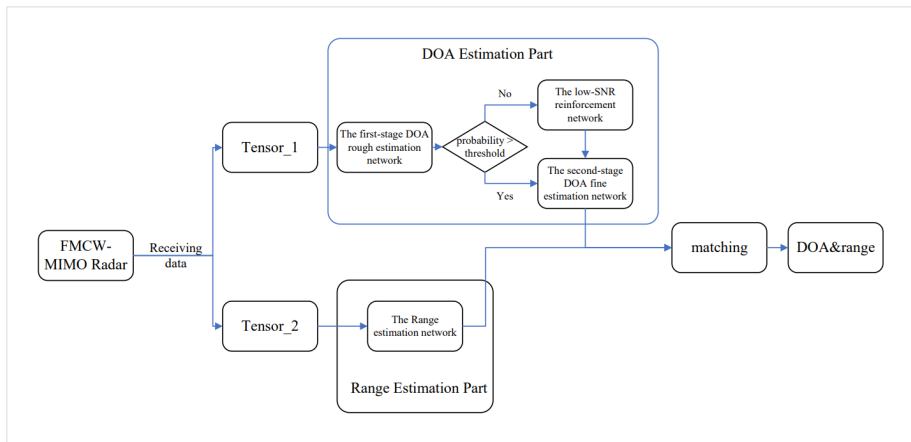
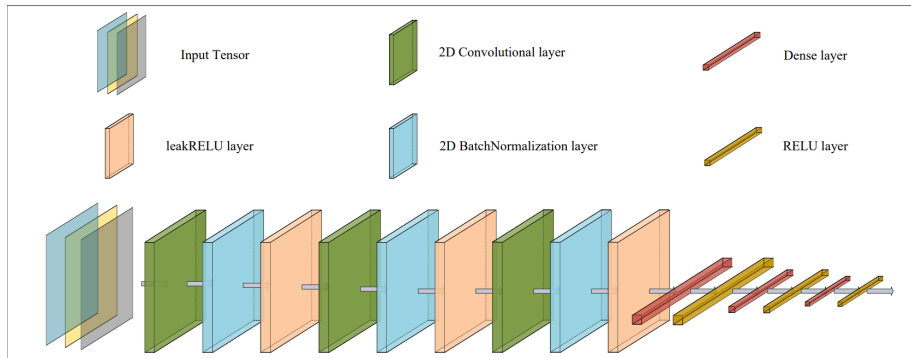


Figura: O processo completo de estimativa do Multi-DeepNet (Su et al., 2023).

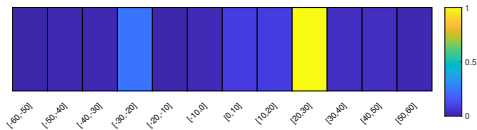


**Figura:** Estrutura da rede de estimativa DOA bruta do primeiro estágio (Su et al., 2023).

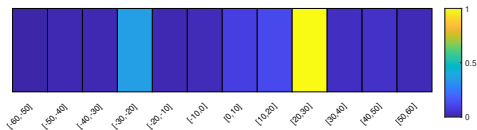
Época	Iteração	Loss (Train)	Loss (Val)	Acurácia	Precisão	Recall	F1-score
2	13 200	0,3254	0,3178	0,878	0,902	0,587	0,711

**Tabela:** Métricas de desempenho da melhor rede obtida durante o treinamento do primeiro estágio (estimativa grosseira).

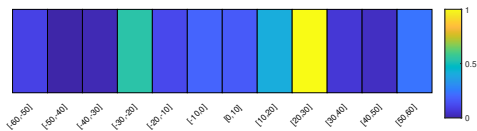




(a) 10 snapshots.



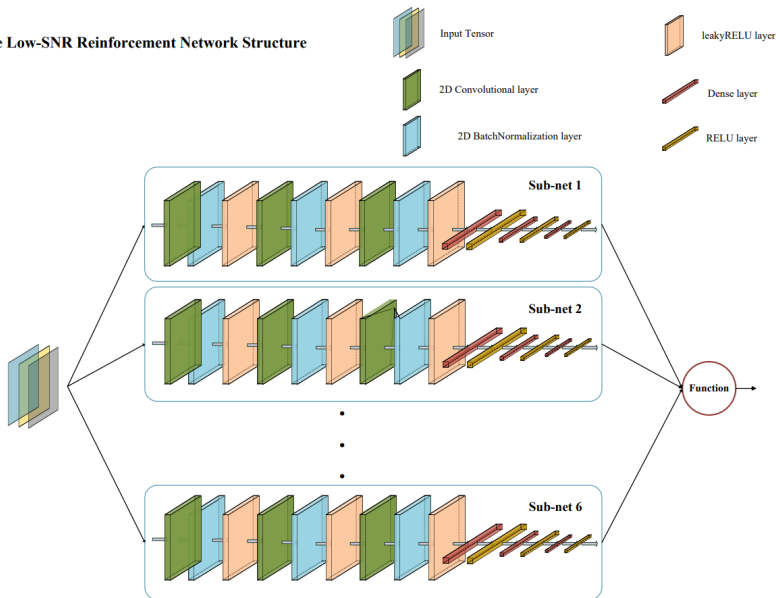
(b) 5 snapshots.



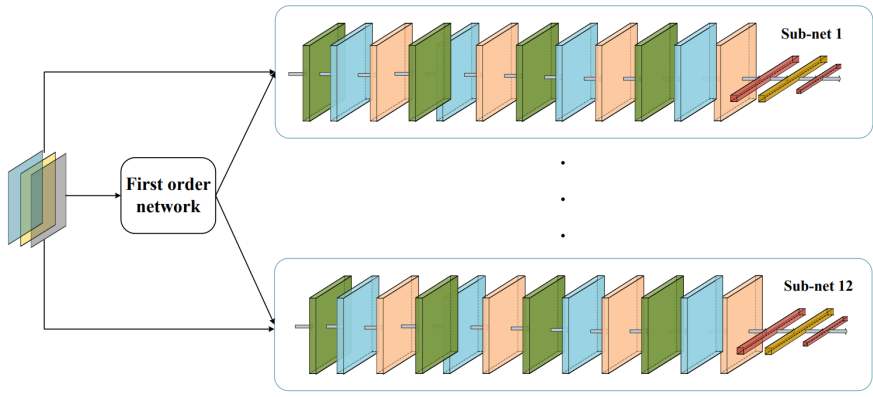
(c) 1 snapshot.

**Figura:** Vetor de probabilidades obtido na saída da rede do primeiro estágio, alimentada com uma amostra de 1 usuário com ângulo de chegada em  $25,32^\circ$

## The Low-SNR Reinforcement Network Structure



## The Second-order DOA Fine Estimation Network Structure



**Figura:** Estrutura de rede de estimativa fina de DOA de segunda ordem. (Su et al., 2023)