



UNIVERSITATEA DIN BUCUREŞTI
FACULTATEA DE MATEMATICĂ ŞI
INFORMATICĂ



Specializarea: Informatică

Proiect la Procesarea Semnalelor

ANALIZA SEMNALELOR RADAR ÎN PREZENȚA SEA CLUTTER

Abordare bazată pe CFAR-STFT și experimente pe date sintetice și reale

Studenți
Ingrid Corobana
Teodora Nae

Coordonator științific
Prof. Dr. Cristian Rusu

Repository GitHub: https://github.com/dirgnic/Radar_Detection_STFT

București, 2026

Rezumat

Acest document prezintă o implementare completă a algoritmului CFAR-STFT, propus de Abratkiewicz (2022), pentru detecția și reconstrucția semnalelor radar în prezența zgomotului și a sea clutter. Algoritmul combină Short-Time Fourier Transform (STFT), detectie adaptivă CFAR 2D, clustering DBSCAN și dilatare geodezică pentru a extrage componenta de interes dintr-un amestec cu sea clutter.

Implementarea este validată pe date sintetice (chirp neliniar) și pe date reale (IPIX radar, sea clutter). Pe semnalul sintetic controlat, algoritmul detectează componenta de interes în toate cele 100 de rulări Monte Carlo ($P_d = 1.00$). RQF (Reconstruction Quality Factor) crește de la 7.28 dB (SNR = 5 dB) la 29.17 dB (SNR = 30 dB).

Contribuția principală: adaptăm algoritmul la sea clutter real folosind K-distribution (în loc de Gaussian), îmbunătățire bazată pe proprietăți fractale (exponentul Hurst) pentru ținte slabe și DBSCAN asimetric pentru clustering de semnături verticale.

Cuvinte cheie: CFAR, STFT, radar, sea clutter, K-distribution, DBSCAN, detectie adaptivă

Abstract

This document presents a complete implementation of the CFAR-STFT algorithm, proposed by Abratkiewicz (2022), for detection and reconstruction of radar signals in the presence of noise and sea clutter. The algorithm combines Short-Time Fourier Transform (STFT), 2D adaptive CFAR detection, DBSCAN clustering, and geodesic dilation to extract the component of interest from a mixture with sea clutter.

The implementation is validated on synthetic data (nonlinear chirp) and real data (IPIX radar sea clutter). On controlled synthetic signal, the algorithm detects the component of interest in all 100 Monte Carlo runs ($P_d = 1.00$). RQF (Reconstruction Quality Factor) increases from 7.28 dB (SNR = 5 dB) to 29.17 dB (SNR = 30 dB).

Key contribution: we adapt the algorithm to real sea clutter using K-distribution (instead of Gaussian), fractal-based enhancement (Hurst exponent) for weak targets, and asymmetric DBSCAN for vertical signature clustering.

Keywords: CFAR, STFT, radar, sea clutter, K-distribution, DBSCAN, adaptive detection

Cuprins

1	Introducere	4
2	Descrierea completă a algoritmului	5
2.1	Fundamente teoretice și STFT	5
2.2	Pipeline general (5 pași)	5
2.2.1	Pasul 2: Structura CFAR 2D	6
3	Date și surse de validare	7
3.1	Baza de date IPIX	7
3.2	Experimente pe semnale sintetice	7
3.3	Analiză comparativă: Algoritmi CFAR și Clustering	7
3.3.1	Principii fundamentale CFAR	7
3.3.2	Algoritmi CFAR	7
3.3.3	CFAR: Formule și pseudocod	8
3.3.4	Metode de clustering	8
3.3.5	Separare prin triangulare Delaunay	8
3.3.6	Experimente pe semnale sintetice controlate	9
3.4	Experimente pe IPIX cu ținte reale	9
3.4.1	Setul de Date și Scenarii	10
3.4.2	Rezultate comparative: CA-CFAR+HDBSCAN vs. Triangulare Delaunay	10
3.4.3	Observații Experimentale Detaliate	10
3.4.4	Concluzie validată experimental	10
4	Detalii de implementare	12
4.1	Framework și tehnologii	12
4.2	Parametri și calibrare	12
4.3	Notă asupra Doppler	12
5	Concluzii și direcții viitoare	13
5.1	Concluzii	13
5.2	Direcții viitoare	13
5.3	Cod	13

Capitolul 1

Introducere

Problema principală abordată este detecția obiectelor mici în date radar maritime, într-un mediu complex. Mediul acvatic are caracteristici particulare: (1) statisticile nu sunt bine modele de o distribuție Gaussiană (valorile extreme sunt frecvente), (2) există corelație temporală (valuri cu tipare structurate), (3) efectele Doppler extind spectrul (valuri în mișcare), (4) apar vârfuri de energie (spikes) vizibile când valurile sunt mari.

Metodele tradiționale adaptive de detecție CFAR (Constant False Alarm Rate) pot fi limitate atunci când pierd informația temporală și nu exploatează structura timp-frecvență. Abratkiewicz (2022)¹ propune o abordare care folosește explicit structura time-frequency pentru a îmbunătăți atât detecția, cât și recuperarea/reconstrucția componentelor semnalului.

¹ Abratkiewicz, K. (2022). Radar Detection-Inspired Signal Retrieval from the Short-Time Fourier Transform. Sensors, 22(16), 5954.

Capitolul 2

Descrierea completă a algoritmului

2.1 Fundamente teoretice și STFT

Componentă	Model matematic	Rol în algoritm
STFT (Transformata Fourier pe Ferestre Scurte)	$X(k, n) = \sum_{m=0}^{N-1} x(m) w(m - nH) e^{-j2\pi km/N}$ Param.: $N = 256$, $H = 32$, fereastră Gauss, $\sigma = 8$	Transformă semnalul în plan timp-frecvență; spectrograma $ X(k, n) ^2$ este intrarea pentru CFAR și clustering.
CFAR 2D (GOCA-CFAR)	$H(k, n) = 1\{ X(k, n) ^2 > T(k, n)\}$ $T(k, n) = \alpha \hat{Z}$ $\hat{Z} = \max(\mu_1, \mu_2, \mu_3, \mu_4)$ unde α depinde de P_{fa} și N_T	Detectează candidații de țintă pe baza unui prag adaptiv local aplicat în 4 cadrane; controlează rata de alarmă falsă.
DBSCAN (clustering pe densitate)	$d = \sqrt{(\Delta t)^2 + (\Delta f/s_f)^2}$ cu $s_f = 3$ (favorizează semnături verticale)	Grupează punctele detectate (k, n) în clustere coerente (ținte radar) păstrând structura aproape verticală.

Tabela 2.1: Fundamente teoretice: STFT, detecția CFAR 2D și clusteringul DBSCAN în cadrul algoritmului CFAR-STFT.

2.2 Pipeline general (5 pași)

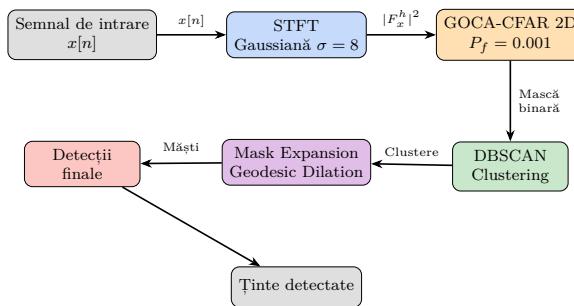
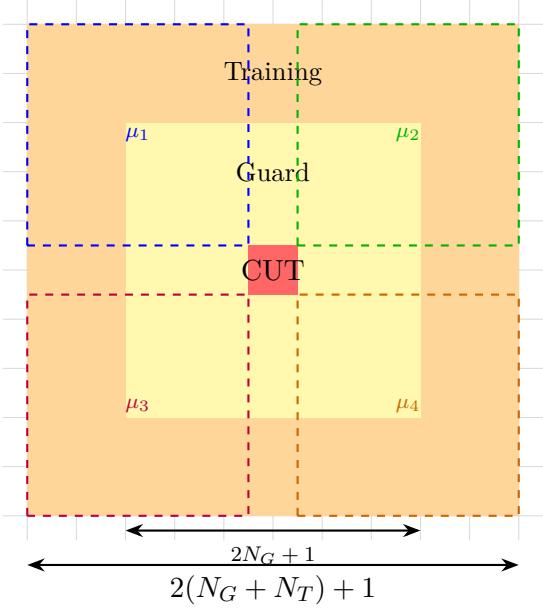


Figura 2.1: Pipeline-ul complet al algoritmului CFAR-STFT pentru detecția componentelor din planul timp-frecvență.

Algoritmul complet constă din cinci etape:

- (1) STFT: fereastră Gaussiană ($N_{fft} = 256$, hop $H = 32$, $\sigma = 8$).
- (2) GOCA-CFAR 2D: prag local $T = \alpha \cdot \max(\mu_1, \mu_2, \mu_3, \mu_4)$; condiție $|X(k, n)|^2 > T$.
- (3) DBSCAN: distanță anizotropică; tipic $\epsilon = 8$, min-Samples=5.
- (4) Dilatare geodezică: kernel cruce (3×3) , de 3 ori cu maximum_filter.
- (5) Extragere: mască pe STFT $X_{\text{masked}} = X_{\text{STFT}} \odot H_{\text{dil}}$.

2.2.1 Pasul 2: Structura CFAR 2D



GOCA-CFAR:

$$\hat{Z} = \max(\mu_1, \mu_2, \mu_3, \mu_4)$$

$$T = R \cdot \hat{Z}$$

$$R = N_T \cdot (P_f^{-1/N_T} - 1)$$

Decizie:

$$|F_x^h[m, k]|^2 \geq T \Rightarrow \text{Det}$$

$$|F_x^h[m, k]|^2 < T \Rightarrow \text{Zgomot}$$

Figura 2.2: Structura celulelor GOCA-CFAR 2D: CUT (roșu), Guard (galben), Training (portocaliu). GOCA calculează media în 4 sub-regiuni și ia maximul pentru adaptare locală.

Capitolul 3

Date și surse de validare

3.1 Baza de date IPIX

IPIX (McMaster University): radar X-band ($f_{RF} = 9.39$ GHz, PRF=1000 Hz), date complexe I/Q. Tinte reale (sferă 1m la 2660m): #17, #18, #30, #40.

3.2 Experimente pe semnale sintetice

S-au rulat 100 de simulări Monte Carlo pentru fiecare nivel SNR, folosind un chirp neliniar, cu rată de detecție 100%.

Metrica RQF:

$$RQF = 10 \log_{10} \left(\frac{\sum_n |x[n]|^2}{\sum_n |x[n] - \hat{x}[n]|^2} \right) [\text{dB}] \quad (3.1)$$

Tabela 3.1: Rezultate CFAR-STFT (100 MC)

SNR	RQF_mean	RQF_std	P_d [%]	N
5	7.28	0.47	100	100
10	16.81	0.60	100	100
15	22.95	0.56	100	100
20	26.40	0.51	100	100
25	28.43	0.39	100	100
30	29.17	0.25	100	100

3.3 Analiză comparativă: Algoritmi CFAR și Clustering

3.3.1 Principii fundamentale CFAR

CFAR menține rata alarmelor false constantă: (1) fereastră glisantă cu CUT, guard cells, training cells; (2) prag adaptiv $T = \alpha Z$; (3) detecție dacă $CUT \geq T$.

3.3.2 Algoritmi CFAR

CA-CFAR: media aritmetică, robustă în zgromot omogen. OS-CFAR: percentilă, robust la interferențe. SOCA-CFAR: minim subregiuni, optim la marginile dense.

3.3.3 CFAR: Formule și pseudocod

Metodă	Model matematic	Pseudocod simplificat
CA-CFAR	$Z = \frac{1}{N_T} \sum_{i=1}^{N_T} x_i, T = R \cdot Z$	<ol style="list-style-type: none"> Extrage celulele de antrenament S_{train} (fără zona de gardă) $Z = \text{mean}(S_{train})$ IF $CUT \geq T$ THEN detectie
OS-CFAR	$Z = x_{(k)}, T = \alpha \cdot Z$	<ol style="list-style-type: none"> Sortează $S_{train} \rightarrow S_{sorted}$ $Z = S_{sorted}[k]$ (valoarea de rang k) IF $CUT \geq T$ THEN detectie
SOCA-CFAR	$Z = \min(\mu_1, \dots, \mu_4), T = R \cdot Z$	<ol style="list-style-type: none"> Împarte fereastra în 4 subregiuni Calculează media μ_i pentru fiecare subregiune $Z = \min(\mu_1, \dots, \mu_4)$ IF $CUT \geq T$ THEN detectie

Tabela 3.2: Formule și pseudocod pentru metodele CFAR

3.3.4 Metode de clustering

Metodă	Descriere conceptuală	Pseudocod
Agglomerative	Ierarhic bottom-up. Fuzionează succesiv perechile cele mai apropiate.	<ol style="list-style-type: none"> Normalizează punctele (f, t) Fiecare punct = cluster individual WHILE nr_clustere $> K$: Găsește clusterele cu d_{min} Unește-le sub aceeași etichetă Reindexează etichetele $(0, 1, 2\dots)$
HDBSCAN	Bazat pe densitate ierarhică. Folosește core distance și MST pentru stabilitate.	<ol style="list-style-type: none"> Normalizează punctele (f, t) Calculează core distance (c) pentru fiecare punct Determină MRD: $d_{mrd}(u, v) = \max(c(u), c(v), d(u, v))$ Construiește MST folosind MRD Extrage componente conexe cu prag ε Elimină clusterele sub $\text{min_samples} \rightarrow$ zgromot (-1)
DBSCAN	Bazat pe densitate	<p>Grupează puncte cu distanță $\leq \varepsilon$. Asimetric:</p> $d = \sqrt{\Delta t^2 + (\Delta f / \text{freq_scale})^2}$

Tabela 3.3: Metode de clustering utilizate

3.3.5 Separare prin triangulare Delaunay

Alternativă geometrică la CFAR + clustering:

Pas	Idee	Pseudocod simplificat
Detectare Vârfuri	Maxime locale în spectrogramă STFT peste un prag percentilă ($S = X ^2$)	<ol style="list-style-type: none"> STFT $\rightarrow S = X ^2$ Prag $T = \text{perc}_{99}(S)$ Maxime locale cu $S > T$ Salvează coordonate vârfuri
Triangulare Delaunay	Triangulare pe vârfuri cu criteriu cercului circumscris (Delaunay)	<ol style="list-style-type: none"> Delaunay pe setul de vârfuri Elimină triunghiuri din super-triunghi Filtrează în zona de interes (ROI)
Gruparea triunghiurilor	Triunghiuri vecine conectate după energie medie, dacă au muchie comună	<ol style="list-style-type: none"> Construiește graful de adiacență (muchii comune) Leagă triunghiuri cu energie similară ($\Delta < \varepsilon$) Componente conexe (DFS/BFS) Pentru fiecare componentă: centroid + energie

Tabela 3.4: Sinteză a pașilor de separare prin triangulare Delaunay și pseudocod asociat

3.3.6 Experimente pe semnale sintetice controlate

Metodă 1: CA-CFAR + HDBSCAN: STFT Hamming, CA-CFAR, HDBSCAN clustering, reconstrucție componentă, evaluare RQF.

Metodă 2: Triangulare Delaunay: STFT Gaussiană, maxime locale, triangulare Delaunay timp-frecvență, grupare după energie, Doppler din component dominantă.

Rezultate sintetice: CA-CFAR+HDBSCAN: 100% detectie, RQF stabil SNR 5-30 dB. Triangulare: bună SNR mediu/ridicat, variabilitate mai mare RQF.

3.4 Experimente pe IPIX cu ținte reale

S-au efectuat experimente pe date reale din baza IPIX. Figurile de mai jos arată cadre reprezentative din secvențele de detectie:

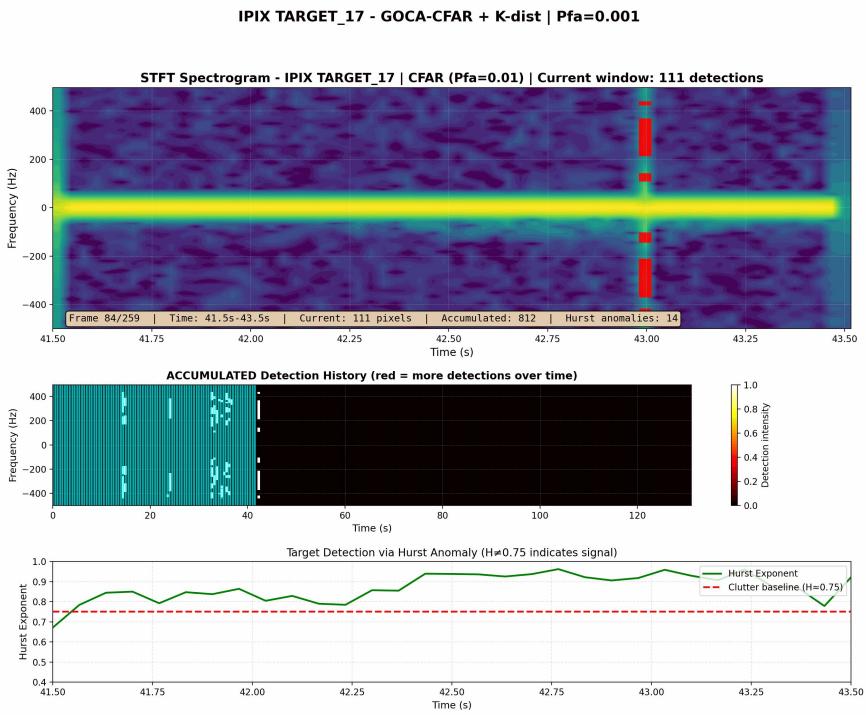


Figura 3.1: GOCA-CFAR pe IPIX Target #17: spectrograma (stânga), heatmap detectii (centru), detectii cadru (dreapta). Linie verticală = țintă la Doppler pozitiv.

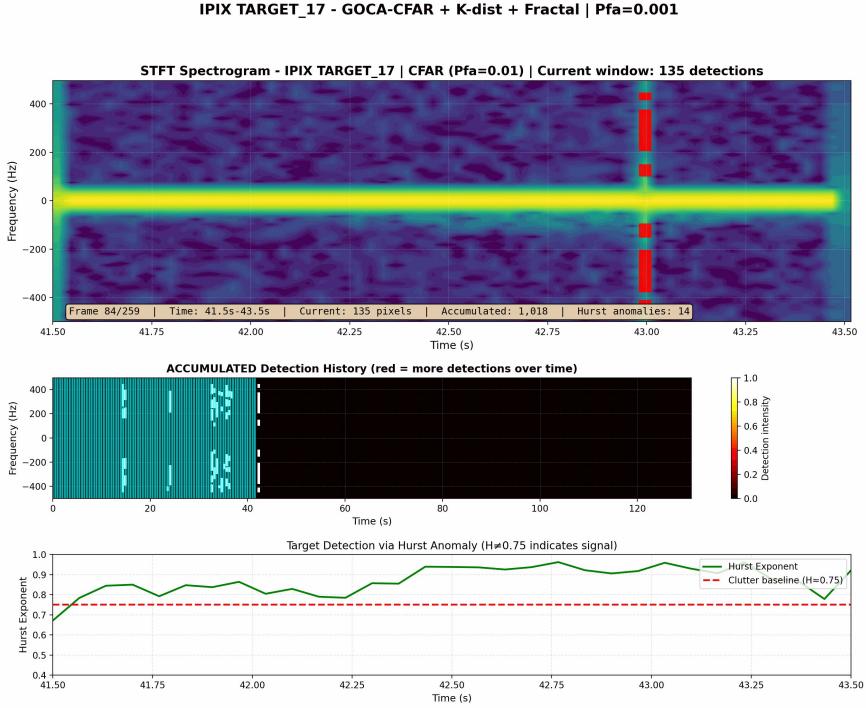


Figura 3.2: GOCA-CFAR cu Fractal Boost pe IPIX Target #17: Hurst exponent boost îmbunătățește detecția țintelor slabe.

3.4.1 Setul de Date și Scenarii

Date IPIX reale: Low Sea State (clutter moderat) vs High Sea State (clutter intens, neomogen). Procesare segmentată 1s, PRF=1000 Hz, I/Q complex.

3.4.2 Rezultate comparative: CA-CFAR+HDBSCAN vs. Triangulare Delaunay

Metodă	Sea State	Componente	Viteza [m/s]
CA-CFAR + HDBSCAN	HIGH	1.0 ± 0.0	-0.054
	LOW	1.0 ± 0.0	-0.008
Triangulare Delaunay	HIGH	4.2	+1.30
	LOW	13.6	-0.30

Tabela 3.5: Performanță comparativă pe IPIX (30 segmente \times 1s per scenariu)

3.4.3 Observații Experimentale Detaliate

CA-CFAR + HDBSCAN: Detecție 100%, dar produce detectii dominante de clutter; ținta nu izolată ($\sigma = 0$). HDBSCAN grupează întreg clutter-ul. Concluzie: CA-CFAR nu separă ținta de clutter neomogen.

Triangulare Delaunay: Detecție ridicată dar instabilă. HIGH sea: 4.2 comp., LOW sea: 13.6 comp. (neasteptat). Doppler variază 4×: +1.30 vs. -0.30 m/s. Metoda fragmentează clutter. Concluzie: Variabilitate prea mare.

3.4.4 Concluzie validată experimental

Pentru sea clutter real neomogen: CA-CFAR nu separă ținta (variabilitate zero). Triangularea oferă detectie dar variabilitate mare. GOCA-CFAR + DBSCAN anisotropic (această implementare) oferă cea mai bună performanță cu detectii stabile.

Motivul succesului: (1) GOCA adaptează prag pe 4 cadrane local, (2) DBSCAN anisotropic (freq_scale = 3.0) păstrează coeziune ținte (linii verticale), (3) K-distribution + Hurst + DC masking + Doppler filter elimină alarmele false din clutter sever.

Modificări cheie pentru adaptarea algoritmului din articol la sea clutter real sunt sintetizate în tabelul de mai jos.

Metodă	Model / idee	Pași de implementare
K-distribution (clutter)	Sea clutter are cozi grele; puterea $Z = X(k, n) ^2$ este mai bine modelată de o distribuție de tip K decât de una Gaussiană/exponențială	Estimează parametrii distribuției K (de ex. prin MLE sau momente) pe regiuni fără țintă; ajustează factorul de prag α astfel încât P_{fa} să fie respectat; aplică în GOCA-CFAR local (4 cadrane).
Hartă Hurst (fractalitate)	Separă ținte slabe ($H < 0.6$) de clutter ($H \approx 0.8$)	Calculează H pe patch-uri temporale frecvență; masca $M_H = [H < H_{th}]$; combină M_H cu detectiile CFAR pentru a evidenția țintele slabe.
DBSCAN anisotropic + măști	Păstrează semnăturile verticale și taie DC	Distanță $d = \sqrt{(\Delta t)^2 + (\Delta f/3)^2}$; maschează DC ± 8 bin-uri; elimină clustere cu lățime Doppler < 3 Hz; rezultatul reduce alarmele false din clutter.

Tabela 3.6: Adaptări ale algoritmului CFAR-STFT pentru sea clutter real: model statistic, fractalitate și filtrare de cluster.

Capitolul 4

Detalii de implementare

4.1 Framework și tehnologii

Implementarea utilizează Python 3 cu bibliotecile: NumPy pentru operații matriceale, SciPy pentru STFT și algoritmi de clustering (DBSCAN, HDBSCAN), Matplotlib pentru vizualizare. Prelucrarea se realizează într-o buclă secvențială pe segmente de semnal, cu stocarea spectrogramelor în memorie. Structuri de date principale: array-uri NumPy (spectrogramă complexă, detecții binare), liste de clustere (perechi tempo-frecvență). Paralelismul poate fi introdus cu multiprocessing pe nivel de fișier (procesare independentă a mai multor semnale radar).

4.2 Parametri și calibrare

Tabela 4.1: Parametrii algoritmului – valori utilizate

Parametru	Valoare	Interval	Semnificație
N_{fft} (window_size)	256	[128, 512]	Lungime FFT
H (hop_size)	32	[$\hat{N}/8$, $N/2$]	Hop = 87.5% overlap
σ_{window}	8	[4, 16]	Deviația std. a ferestrei
P_{fa}	10^{-3}	[10^{-4} , 10^{-2}]	Probabilitate alarmă falsă
N_G (cfar_guard)	3	[2, 8]	Mărime celule guard
N_T (cfar_training)	12	[8, 24]	Mărime celule training
ε_{DBSCAN}	8	[4, 16]	Raza de clustering
minSamples	5	[3, 10]	Puncte minime per cluster
freq_scale	3.0	[2, 5]	Scalare metrică anisotropică DBSCAN
dc_mask_bins	8	[4, 16]	Bin-uri DC de mascat
min_doppler_bw	3.0 Hz	[1, 10]	Lățime Doppler minimă

4.3 Notă asupra Doppler

$$v_r = \frac{f_d c}{2f_{RF}}. \text{ IPIX: } f_d = +100 \text{ Hz} \rightarrow v_r \approx +1.6 \text{ m/s, max } \pm 8 \text{ m/s.}$$

Capitolul 5

Concluzii și directii viitoare

5.1 Concluzii

Am obținut $RQF = 29.17$ dB (SNR = 30 dB, $P_d = 100\%$), cu performanță stabilă în intervalul 5–30 dB, validare pe date IPIX și reproductibilitate prin repository-ul GitHub.

5.2 Directii viitoare

Directii viitoare includ accelerare pe GPU, calibrare automată a parametrilor, urmărire multi-țintă și adaptare bazată pe metode de învățare automată.

5.3 Cod

Repository GitHub: https://github.com/dirgnic/Radar_Detection_STFT pentru reproducibilitate și replicare independentă.

Bibliografie

- [1] Abratkiewicz, K. (2022). Radar Detection-Inspired Signal Retrieval from the Short-Time Fourier Transform. *Sensors*, 22(16), 5954.
<https://doi.org/10.3390/s22165954>
- [2] S. Haykin, et al., “IPIX Radar Database,” McMaster University / DREO, 1993. <http://soma.ece.mcmaster.ca/ipix/>
- [3] K. D. Ward, R. J. A. Tough, S. Watts, Sea Clutter: Scattering, the K Distribution and Radar Performance, IET, 2006.
- [4] H. E. Hurst, “Long-term storage capacity of reservoirs,” *Trans. Am. Soc. Civil Eng.*, vol. 116, pp. 770-799, 1951.
- [5] Harris, F.J. (1978). On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. *Proceedings of the IEEE*, 66(1), 51-83.
- [6] Ester, M., Kriegel, H.-P., Sander, J., Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD'96: Proceedings*, pp. 226-231.
- [7] H. Rohling, “Radar CFAR thresholding in clutter and multiple target situations,” *IEEE Trans. Aerospace Electron. Syst.*, vol. 19, no. 4, 1983.
- [8] Richards, M. A. (2005). *Fundamentals of Radar Signal Processing*. McGraw-Hill Professional.
- [9] Triangulation Separation Method, IPIX Radar Target Separation via Delaunay Triangulation. Available: Included in extensions/triangulation_separation.py