AI-Enabled Radar for Drone Operations: Detection, Tracking, and Classification

Dr. Shishir Mishra
BCTRC Nagpur

***Abstract*—This paper presents a practical framework for integrating radar signal processing with lightweight machine learning to enable robust UAV perception for collision avoidance, target classification, and autonomous navigation. We review core signal-processing components (FMCW/IQ acquisition, FFT-based range mapping, Doppler processing), adaptive detection using CFAR, micro-Doppler feature extraction, and edge deployment of compact neural networks (TensorFlow Lite / TinyML). A reference pipeline is described alongside implementation strategies for embedded platforms and evaluation methodology using simulated and micro-Doppler datasets. Results indicate that combining classical radar detection with small CNN/ML classifiers provides reliable drone vs. bird discrimination and improves situational awareness in degraded-visibility conditions.**

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are increasingly operating in cluttered airspaces where reliable sensing is critical for safety. Optical sensors such as cameras and LiDAR provide rich scene information but are limited under adverse weather and low-light conditions. Radar offers complementary sensing capabilities: it operates effectively in night, fog, dust, and other obscurants, and captures kinematic signatures encoded in frequency and phase (Doppler and micro-Doppler). Recent research demonstrates that micro-Doppler signatures can effectively discriminate UAVs from birds and other objects, and that compact ML models can run on edge hardware for real-time classification. This work synthesizes these concepts into a deployable perception pipeline suitable for drone operations.

II. RELATED WORK

Micro-Doppler analysis and radar-based classification have been extensively studied. Surveys highlight micro-Doppler's utility in UAV identification, and recent studies show the growing adoption of deep learning on radar spectrograms for classification. RF/Radar UAV detection research examines both passive and active radar methods, highlighting challenges posed by small radar cross-section (RCS) targets. Edge ML frameworks such as TensorFlow Lite and TinyML enable deployment of these models on resource-constrained embedded systems.

III. SYSTEM OVERVIEW & PIPELINE

A practical onboard radar-AI perception stack comprises the following modules:

1. Signal Acquisition: FMCW or pulsed radar produces complex I/Q samples per chirp/frame. Phase coherence is essential for accurate Doppler and micro-Doppler extraction.

2. Preprocessing: DC removal, windowing (Hann/Hamming), and calibration.

3. Range FFT: Per-chirp FFT produces range bins. Efficient FFT libraries (FFTW, KissFFT, or vendor DSP libraries) are recommended for real-time operation.

4. Doppler / Micro-Doppler Processing: Compute Doppler FFTs over slow-time frames to form time-frequency spectrograms, revealing rotor/propeller modulations useful for UAV vs. bird discrimination.

5. Detection (CFAR): Adaptive thresholding (CA-CFAR, OS-CFAR) locates candidate targets while maintaining controlled false-alarm rates. Parameter tuning is critical in cluttered environments.

6. Tracking: Use Kalman filters or advanced multi-target trackers to estimate kinematic state (range, range-rate) over time.

7. Feature Extraction & Classification (AI): Extract micro-Doppler patches per detection and classify using compact CNN/ML models exported to TensorFlow Lite / TinyML for edge inference.

8. Decision & Autonomy: Fuse classification and tracking to trigger avoidance maneuvers, path replanning, or mission-level decisions.

Pipeline summary:
Raw IQ → Range FFT → Range-Doppler / Spectrogram → CFAR → Tracks → Classifier → Autonomy

IV. SIGNAL PROCESSING DETAILS

4.1. Range FFT and Windowing
FFT size (N) determines range resolution and bin width. Apply taper/windowing to reduce sidelobes. Fixed- or floating-point optimized FFTs are recommended for constrained MCUs.

4.2. Doppler & Micro-Doppler
Stack M successive pulses and compute Doppler FFTs across slow-time frames to obtain velocity information. Micro-Doppler signatures, resulting from rotating/flapping parts, are represented as spectrograms and provide discriminative cues for small UAV vs. bird classification.

4.3 CFAR Detection
CFAR computes adaptive thresholds from neighboring training cells while guarding the test cell(s). Variants include CA-CFAR, OS-CFAR, and 2D/3D CFAR on range-Doppler maps. CFAR maintains a constant false alarm rate under non-stationary noise/clutter.

V. MACHINE LEARNING FOR CLASSIFICATION

5.1. Feature Choices
1D spectral vectors (amplitude vs. frequency) for small CNN/ML models.

2D micro-Doppler spectrograms for CNN classifiers.

Handcrafted features: SNR, RCS estimate, Doppler variance, spectral centroid.

5.2. Model Types & Edge Deployment
Lightweight CNNs or MLPs quantized to int8 can be deployed via TensorFlow Lite / TinyML on SBCs or MCUs. Model selection depends on latency, memory, and accuracy requirements.

VI. IMPLEMENTATION GUIDELINES

Radar Frontend: Short-range FMCW radars (24/60/77 GHz) or mmWave modules (e.g., TI SDK) are suitable. Phase coherence and sampling fidelity are key.

FFT & DSP: FFTW/KissFFT or vendor DSP libraries; fixed/floating-point support depends on the host.

Edge ML: Use TensorFlow Lite / TF-Micro; quantize models and test on representative datasets.

CFAR & Tracking: Implement CA/OS-CFAR; use gating and nearest-neighbor for simple tracking, or Hungarian/assignment methods for multi-target scenarios.

VII. EVALUATION APPROACH

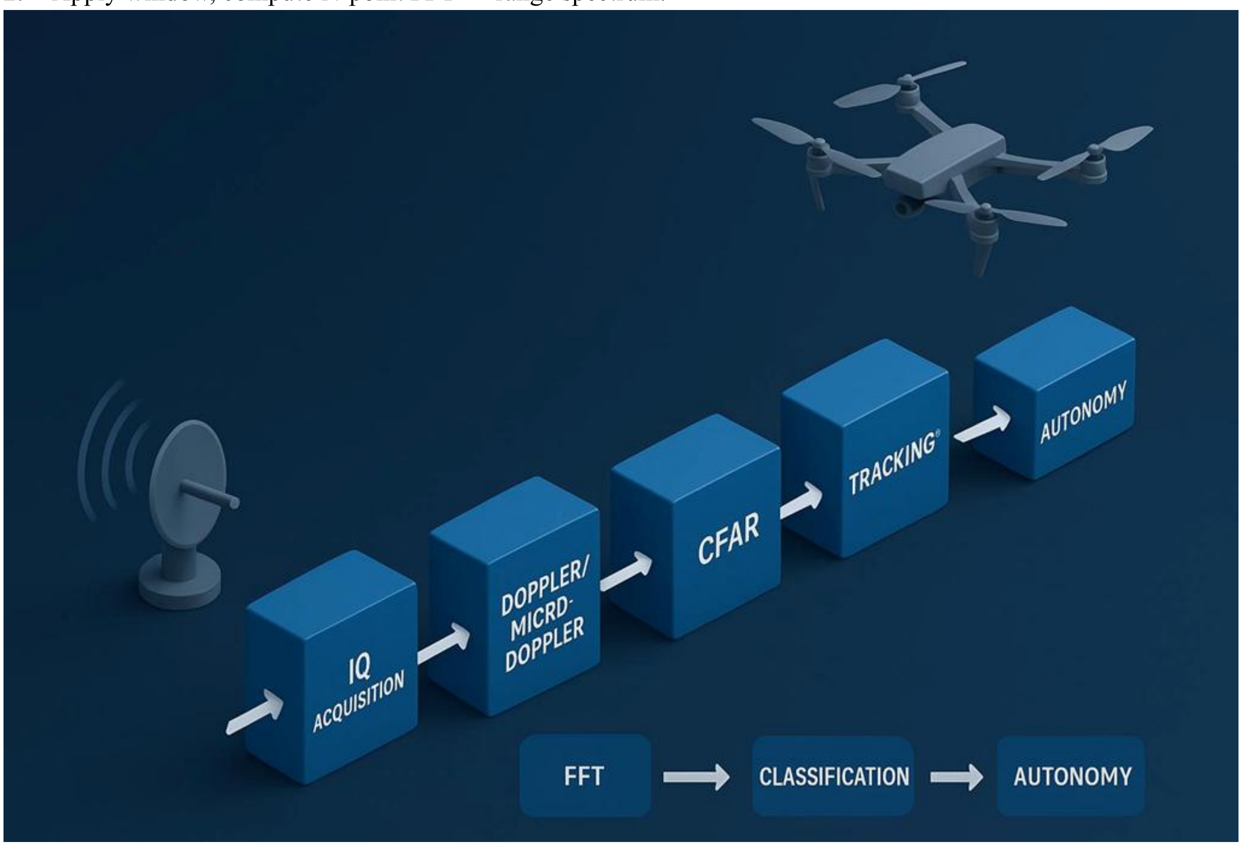
```c
/* radar_detect.c
   Minimal radar detection skeleton in C.
   - Simulated input: complex samples
(float I,Q) interleaved
   - Uses an external FFT function
(replace with FFTW/kissFFT on your
platform)
   - Simple CFAR-like detector
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>

#define N 1024          // FFT size (range
bins)
#define GUARD 2         // CFAR guard cells
#define TRAIN 16        // CFAR training
cells each side
#define PROB_FA 1e-4    // not used
directly, but for design

typedef struct { float re, im; } cplx;

/* ----- Replace this with real FFT
library call ----- */
/* Input: in[] complex array length N;
output: out[] complex fft result */
void fft_forward(cplx *in, cplx *out, int
n) {
    // Placeholder naive DFT (O(n^2)) for
clarity. Replace with FFTW/kissFFT in
practice.
    for (int k = 0; k < n; ++k) {
        float re = 0, im = 0;
        for (int t = 0; t < n; ++t) {
```

1. Simulated Data: Synthetic IQ signals with embedded rotor tones and noise for pipeline verification.
2. Controlled Field Trials: Collect labeled micro-Doppler data from UAVs, birds, and clutter at multiple ranges/aspect angles.
3. Metrics: Detection probability (Pd), false alarm rate (Pfa), classification accuracy, confusion matrices, latency, and resource usage.
4. Ablation Studies: Analyze impact of window size, FFT length, CFAR parameters, and classifier architecture on performance

VIII. EXAMPLE REFERENCE IMPLEMENTATION

Each frame:

1. Acquire N-sample IQ chirp.
2. Apply window, compute N-point FFT → range spectrum.



3. Stack K chirps → Doppler FFTs / spectrograms.
4. Apply 1D/2D CFAR → candidate cells.
5. Extract micro-Doppler patches → normalize → TF-Lite classification.
6. Update Kalman tracks and fuse labels with kinematic confidence → trigger avoidance if required.

Deployable on Raspberry Pi / Jetson Nano, later portable to MCU with TF-Micro.

IX. DISCUSSION

Strengths: Radar + AI complements vision sensors, providing all-weather, low-light operation and unique kinematic information. Micro-Doppler aids rotorcraft identification.
Limitations: Small RCS and low SNR at long ranges challenge detection. Classifiers require representative data across angles, ranges, and environmental conditions. Passive radar and sensor fusion can mitigate this.

Future Directions: Multi-sensor fusion (camera + radar + IMU), joint communications & sensing (JC&S), distributed sensor networks, semi-supervised learning, advanced 3D CFAR, and robust multi-target tracking.

X. CONCLUSION

Integrating classical radar detection (FFT, CFAR, tracking) with compact machine learning yields an effective perception stack for UAV operations. Micro-Doppler spectrograms enable discriminative drone vs. bird classification, while edge ML runtimes make onboard inference feasible. Careful design, dataset curation, and regulatory compliance are essential for safe deployment. PinakShakti Aerospace envisions such hybrid radar-AI stacks as key enablers of safer, more autonomous aerial systems.

REFERENCES

[1] S. Fan et al., Micro-Doppler Signature Detection and Recognition, PMC / Sensors.
[2] R. David et al., Embedded Machine Learning on TinyML Systems, TensorFlow Lite Micro / TinyML Overview.
[3] Survey: Detection, Classification, and Tracking of UAVs using RF/Radar Systems, arXiv, 2025.
[4] MATLAB Phased Array / CFAR Tutorials.
[5] TI, Machine Learning on the Edge with mmWave Radar, micro-Doppler and CNN approaches.