

Project Report: Machine Learning-based Drone and Bird Classification by Micro-Doppler Radar Signatures

Zhongliang Guo
University of St Andrews
zg34@st-andrews.ac.uk

Samiur Rahman
University of St Andrews
sr206@st-andrews.ac.uk

Duncan A. Robertson
University of St Andrews
dar@st-andrews.ac.uk

Abstract

In this study, we present a comprehensive study on radar-based classification of drones and birds using multi-frequency radar systems. We first develop physical models for both drones and birds to understand their unique radar signal and corresponding micro-Doppler signatures. Data collection was performed using radar systems operating at 24GHz, 94GHz, and 207GHz frequencies, capturing real-world signatures from various targets. Through advanced signal processing techniques, we generated a diverse dataset comprising micro-Doppler images across four distinct categories: birds, drones, noise, and clutter. To evaluate the effectiveness of deep learning approaches in target classification, we implemented and compared several state-of-the-art neural network architectures using this dataset. Additionally, we investigated the cross-domain adaptability of our models by assessing how networks trained on real-world data perform when applied to simulated scenarios. This research contributes to the growing field of radar-based target classification by providing insights into the effectiveness of multi-frequency radar systems and deep learning approaches in distinguishing between biological and artificial aerial targets. The code, pre-trained weights, and training logs¹ for this project is available at [here](#), the dataset is available at [here](#).

1. Introduction

The skies are no longer just the domain of birds and aircraft — a new era of autonomous flying machines has quietly revolutionized our airspace. While these unmanned aerial vehicles (UAVs) demonstrate remarkable capabilities in commercial and humanitarian applications, they have simultaneously emerged as tools for those seeking to circumvent security measures [8]. This dual nature of drone technology presents a complex challenge for security systems, particularly in sensitive areas like airports, government fa-

cilities, and private properties.

Traditional security measures have proven inadequate against these agile aerial platforms, driving the search for more sophisticated detection methods. Among the emerging solutions, radar-based detection systems utilizing micro-Doppler analysis have shown exceptional promise [8]. The micro-Doppler effect, which captures the unique vibrational and rotational signatures of drone components, provides a wealth of information about the target's characteristics [6]. These distinctive patterns serve as a reliable basis for drone classification, enabling systems to differentiate between various drone models and their operational states [10]. Recent research has demonstrated that micro-Doppler signature analysis can achieve remarkable accuracy in drone identification, even in challenging environmental conditions [2].

However, effectively leveraging micro-Doppler signatures for reliable target classification presents several significant challenges. First, drones and birds produce distinct micro-Doppler signatures due to their different motion mechanics, yet classification remains challenging under varying operational conditions [9]. Second, environmental factors and clutter can significantly impact the quality of radar measurements, potentially masking or distorting the target signatures [11]. Third, the increasing variety of commercial drone models, each with unique physical characteristics and flight patterns, creates a complex classification problem that traditional signal processing approaches struggle to address effectively [15].

To tackle these challenges, we present a comprehensive approach that combines theoretical modeling, extensive data collection, and advanced deep learning techniques. Our work makes several key contributions:

- We develop theoretical models for both bird and drone radar signatures, providing a foundation for understanding their distinctive characteristics and enabling the generation of synthetic training data.
- We establish a large-scale, multi-frequency radar dataset comprising real-world measurements from three frequency bands (24GHz, 94GHz, and 207GHz), accompa-

¹The format is in [TensorBoard](#), implemented with [TensorBoardX](#).

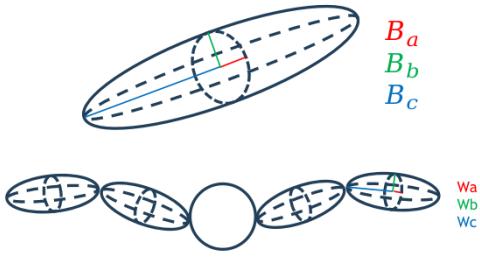


Figure 1. Illustration of the Body and Wing.

- nied by detailed annotations and quality control protocols.
- We implement and evaluate multiple state-of-the-art deep learning architectures for three increasingly complex classification tasks: binary drone detection, drone model classification, and four-class (bird-drone-clutter-noise) discrimination.
 - We conduct extensive transferability analysis between simulated and real-world data, providing insights into the practical applicability of simulation-based training for radar classification systems.

The remainder of this paper is organized as follows:

- Section 2 presents our theoretical modeling approach for bird and drone radar signatures.
- Section 3 details our data collection methodology and dataset generation process.
- Section 4 describes the implementation and evaluation of various deep learning architectures, and our experimental results and analysis.
- Section 5 concludes the paper with a discussion of our findings and future research directions.

2. Physical Simulation

In this section, we will illustrate the way how we model the bird and drone, including meanings of parameters names, value range for each parameters. We also design the Graphical User Interface (GUI) for the simulation code, which is available in Supplement A.

2.1. Simulation for Bird

The flight speed is set as constant as $c = 2.99792458 \times 10^8$.

We defined some parameters for birds' body. B_a : Body semi-axis length a , B_b : Body semi-axis length b and B_c : Body semi-axis length c . The parameters for birds' wings: W_a : Wing semi-axis length a , W_b : Wing semi-axis length b and W_c : Wing semi-axis length c .

For birds, we assume a simple kinematic model of a bird's wing with two connected parts, as shown in the Figure 1 and Figure 2.

Where the input parameters from the user are as follows: flapping frequency f_{flap} , length of upper arm L_1 , amplitude

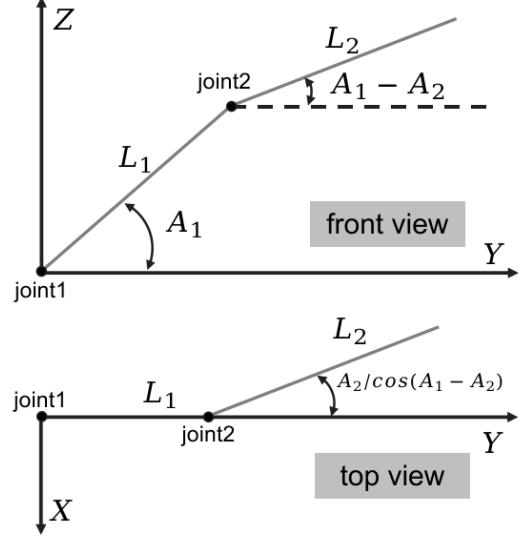


Figure 2. Illustration of a kinematic model of a bird's wing with two connected parts.

of flapping angel for upper arm ($^\circ$) A_1 , lag flapping angle for upper arm ($^\circ$) ψ_{10} , length of Forearm L_2 , amplitude of flapping angel for forearm ($^\circ$) A_2 , lag flapping angle for forearm ($^\circ$) ψ_{20} and Forearm sweep angle C_2 .

Radar parameters include time duration T , number of samples n_t , wavelength λ , range resolution $rangeres$, radar position $radarloc$.

The bird is initially located at the origin of an inertial coordinate system. Its velocities along the three coordinate axes are v_x , v_y and v_z respectively. The bird's flight direction points in the direction of its velocity vector. The bird's motion can be decomposed into two components: the main translational motion of its body and the micro-motion of the target.

Two coordinate systems are involved in this scenario: the body-fixed coordinate system and the inertial coordinate system. The bird's wing-flapping motion is more conveniently calculated in the body-fixed coordinate system. However, we ultimately need the coordinates of the bird's wings in the inertial coordinate system. Therefore, coordinate rotation and translation are used to transform from the body-fixed system to the inertial system.

Assuming the coordinates of a scattering center on the target in the body-fixed coordinate system are (x, y, z) , and the bird's flight direction is aligned with its velocity vector $v = (v_x, v_y, v_z)$, the coordinates of this scattering center in the inertial coordinate system after the bird has flown for time t can be calculated through the following steps:

- Calculate the unit vector of the velocity vector:

$$\mathbf{v}_{unit} = \frac{\mathbf{v}}{|\mathbf{v}|}, \quad (1)$$

where v is the velocity vector, v_{unit} is the normalized velocity vector (unit vector).

- Set the initial direction vector. The initial direction vector is set as the unit vector along the x-axis:

$$\mathbf{d}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (2)$$

- Calculate the rotation axis. The rotation axis is determined by the cross product of the initial direction vector and the velocity unit vector:

$$\mathbf{r}_{\text{axis}} = \frac{\mathbf{d}_0 \times \mathbf{v}_{\text{unit}}}{|\mathbf{d}_0 \times \mathbf{v}_{\text{unit}}|} \quad (3)$$

- Calculate the rotation angle. The rotation angle is calculated using the dot product of the initial direction vector and the velocity unit vector.

$$\theta = \cos^{-1}(\mathbf{d}_0 \cdot \mathbf{v}_{\text{unit}}) \quad (4)$$

- Construct the rotation matrix. Construct the rotation matrix \mathbf{R} using Rodrigues' rotation formula:

$$\mathbf{K} = \begin{bmatrix} 0 & -r_{\text{axis},3} & r_{\text{axis},2} \\ r_{\text{axis},3} & 0 & -r_{\text{axis},1} \\ -r_{\text{axis},2} & r_{\text{axis},1} & 0 \end{bmatrix} \quad (5)$$

$$\mathbf{R} = \mathbf{I} + \sin(\theta)\mathbf{K} + (1 - \cos(\theta))\mathbf{K}^2 \quad (6)$$

where \mathbf{I} is the identity matrix, and \mathbf{K} is a matrix constructed based on the rotation axis.

- Apply rotation and translation. For each point, apply rotation to its coordinates and then perform translation:

$$\mathbf{p}_{\text{new}} = \mathbf{R} \cdot \mathbf{p} + \mathbf{v} \cdot t \quad (7)$$

- Obtain the new coordinates: Finally, we obtain the updated coordinate matrix \mathbf{X}_{new} , representing the new coordinates after rotation and translation. v_x : Forward translation velocity x-component, v_y : Forward translation velocity y-component and v_z : Forward translation velocity z-component.

Parameter Summary and value ranges:

- n_t : number of samples, ranges from 10000 to 20000.
- f_c : carrier frequency of the radar, ranges from 70 GHz to 130 GHz.
- $\text{range}_{\text{res}}$: range resolution of the radar, ranges from 0.03 m to 0.07 m.
- radarloc : radar location in the x-y-z coordinate system, x ranges from 0 m to 20 m, y ranges from 0 m to 20 m, z ranges from -20 m to 0 m.
- f_{flap} : flapping frequency, ranges from 5 Hz to 20 Hz.
- A_1 : amplitude of flapping angle, ranges from 30° to 50°.

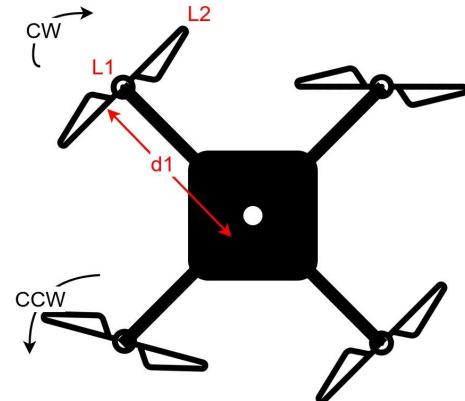


Figure 3. Illustration of the drone model.

- ψ_{10} : lag flapping angle, ranges from 10°.
- L_1 : length of segment 1, ranges from 0.3 m to 0.7 m.
- A_2 : amplitude of segment 2 flapping angle, ranges from 20° to 40°.
- ϕ_{20} : lag flapping angle for segment 2, ranges from 30° to 50°.
- L_2 : length of segment 2, ranges from 0.3 m to 0.7 m.
- C_2 : Forearm sweep angle, ranges from 15° to 25°.
- $v_{x,y,z}$: velocity of the bird along the x, y, z axis. v_x ranges from -5 m/s to 5 m/s, v_y ranges from -5 m/s to 5 m/s, v_z ranges from -5 m/s to 5 m/s.

2.2. Simulation for Drone

We simulate the drone in a CW radar system. Before starting, we define some constants, where $c = 2.99792458 \times 10^8$ is the light speed, $\theta = 0$ is the offset, $\sigma = 1$ is the RCS of the point scatterer, $ck = 0.200$ is the coefficient for the drone body, $i = \sqrt{-1}$ is the imaginary unit.

Given a drone model as shown in Figure 3, we define NR to indicate the number of rotors; N to indicate the number of blades for each rotor; L_1 is the distance (m) from blade root; L_2 is the distance (m) from blade root to blade tip; d_1 is the distance (m) from blade root to the gravity center of the drone; v is the velocity (m/s) of the drone; β is the elevation angle (°); f_r is the rotation rate (revolution per second) for each rotor, the positive value means clockwise rotation (CW), the negative value means counterclockwise rotation (CCW), notable, the length of f_r should be same as the NR ; R is the range (m) from the radar to the drone.

For the radar, we have N_{sample} to indicate total number of samples during a simulation; f_c is the carrier frequency (GHz); samplerate is the number of sampling per second.

For each time interval, here is $\Delta t = \frac{1}{\text{samplerate}}$, and if we assume the start time point is 0, the time point for

each sample can be calculated as $t = \Delta t \cdot N_{\text{sample}}^{(m)}$, where m represents the m -th sample. For the wavelength λ , here is $\lambda = \frac{c}{f_c}$. The center of a blade is $l_b = L_2 - L_1$.

From the EM scattering point of view, the distance from the radar to the origin of the reference coordinates is R_0 , when the speed v is not 0, the R_0 for each time point is related to the R and t , which can be calculated as $R_0 = R + v \cdot t$; the initial rotation angle of the point P at $t = 0$ is ϕ_0 , then at time t the rotation angle becomes $\phi_t = \phi_0 + \Omega t$. Thus, for each rotor, the range from the radar to the point scatter P becomes:

$$R_P(t) = \sqrt{R_0^2 + d_1^2 - 2d_1R_0 \cos(\theta + \theta_r)}, \quad (8)$$

where θ_r is the angle (degree) between the current rotor and the first rotor, we assume that the angle between each rotor is the same, thus $\theta_r = qrn \frac{360}{NR}$, qrn refers to the n -th rotor. The radar observed elevation angles of the origin of the reference coordinates are β , if the elevation angle β are not zero, the phase function can be modified as:

$$\Phi_P(t) = \frac{4\pi}{\lambda} \left[R_p(t) + l_b \cos \beta \sin \left(2\pi f_r^{qrn} t + 2\pi \frac{n_b}{N} \right) \right], \quad (9)$$

where n_b indicated the n -th blade, the $2\pi \frac{n_b}{N}$ indicates the angle (in rad) between the current blade and first blade. The amplitude of the radar return signal from the blade is:

$$AL(t) = \sigma l_b \exp \{i[2\pi f_c t - \Phi_P(t)]\}. \quad (10)$$

X is a phase term used within the context of the sinc function to model the distribution and spread of the radar return over the blade's length. The actual radar return signal from a blade is represented by the combination of AL and the sinc function applied to X , summed up in the return signal:

$$X(t) = \frac{4\pi}{\lambda} l_b \cos \beta \sin \left(2\pi f_r^{qrn} t + 2\pi \frac{n_b}{N} \right). \quad (11)$$

The return signal of a blade at the time point will be:

$$S_R(t) = AL(t) \cdot \text{sinc} \left(\frac{X_0}{\pi} \right). \quad (12)$$

Parameter summary and value ranges:

- NR : Number of rotors, ranges from 2 to 8.
- N : number of blades for each rotor, ranges from 2 to 6.
- L_1 : the distance blade root shifted, fixed to 0 m.
- L_2 : the distance from blade root to blade tip, ranges from 0.05 m to 0.2 m.
- d_1 : the distance from blade root to the gravity center of the drone, ranges from 0.2 m to 0.8 m.
- v : the velocity of the drone, ranges from 0 m/s to 10 m/s.
- β : the elevation angle, ranges from 0° to 90° .

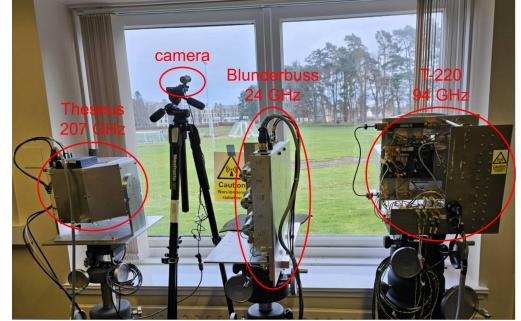


Figure 4. Radars and camera position in the trial place.



Figure 5. Illustration of the experimental trial location.

- f_r : the rotation rate, i.e., revolution per second, ranges from -200 to 200 for each rotor, where positive value represents clockwise rotation, the negative value represents counterclockwise rotation.
- R : the range from the radar to the drone, ranges from 50 m to 200 m.
- samplerate : the number of sampling per second, ranges from 100 kHz to 250 kHz.
- f_c : carrier frequency of the radar, ranges from 70 GHz to 130 GHz.

3. Data Collection and Processing

3.1. Experiment setup

Our field experiments were conducted at the James Gregory Observatory, St Andrews, Scotland, where we established a comprehensive multi-radar measurement system. As shown in Figure 4, we deployed three radar systems operating at different frequencies, each equipped with its dedicated control PC, along with a 720P camera for visual ground truth recording.

To ensure precise temporal alignment of measurements across different systems, we implemented a robust time synchronization protocol. All control PCs were synchronized to a common time reference through an NTP server, which was established using a separate laptop equipped with a GPS dongle. This synchronization was crucial for cor-

Table 1. Key specifications of the three radar systems. Parameters include operating frequency (f), polarization mode (Pol., where HH denotes horizontal-horizontal and CP denotes circular polarization), instrumented range (R_{ins}), range resolution (ΔR), and maximum unambiguous velocity (v_{max}).

Parameter	Blunderbuss	T-220	Theseus
f (GHz)	24	94	207
Pol.	HH	CP	HH
R_{ins} (m)	153.6	102.4	153.6
ΔR (cm)	60	20	7.5
v_{max} (m/s)	± 15.6	± 10.32	± 5.36

relating data across different radar systems and the video recordings. Each data file was automatically timestamped at creation, allowing for accurate temporal matching between radar measurements and visual observations.

The measurement setup was designed for autonomous operation without requiring external trigger switches. Each radar system operated independently through its dedicated control interface, providing real-time monitoring capabilities through live range profile and Range-Doppler profile updates on their respective GUIs. During the experiments, the camera continuously recorded video footage of the measurement area (shown in Figure 5), with each video file named according to its starting timestamp. This coincidental video recording served as ground truth data for subsequent analysis and validation of radar measurements.

3.1.1 Radars

The specifications of our three FMCW radar systems are detailed in Table 1. The Blunderbuss radar operates at 24 GHz with horizontal-horizontal polarization (HH), equipped with a 20 dB attenuator in its receiving channel to suppress phase noise effects. The T-220 operates at 94 GHz with circular polarization (CP). The Theseus radar operates at 207 GHz with HH polarization. As shown in the table, these systems offer complementary capabilities in terms of instrumented range (R_{ins}), range resolution (ΔR), and maximum unambiguous velocity (v_{max}), with higher frequency systems generally providing better range resolution at the cost of reduced velocity coverage.

It is worth noting that the 94 GHz radar also offers two extra operational modes for drone measurements: CW mode with CP, and scanning mode with CP and a 30° scan angle.

3.1.2 Birds

For birds, we conducted data collection on February 6, 2024, under moderate wind conditions with intermittent gusts reaching maximum speeds of approximately 22 mph.

Table 2. Details of birds we collected from.

	wingspan	length	weight
Bald eagle	200 cm	72 cm	9.5 lb
Harris hawk	106 cm	45 cm	1 lb 10.5 oz
Caracara	128 cm	54 cm	3 lb 5 oz
Tawny owl	88 cm	34 cm	15 oz
Redback buzzard	124 cm	40 cm	1 lb 7.5 oz
Kestrel	78 cm	35 cm	7.5 oz



Figure 6. Birds that we collected data during the field experiments.

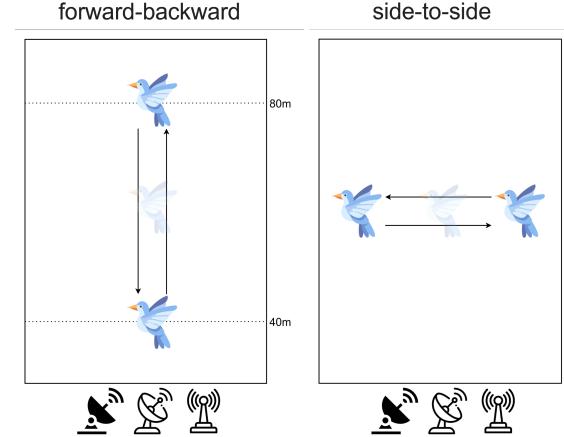


Figure 7. Flight patterns that we collected data for 7 species of birds.

FMCW data was collected across all radar systems for six different species as shown in Figure 6: (a) bald eagle, (b) redback buzzard, (c) tawny owl, (d) kestrel, (e) caracara, and (f) Harris hawk. The detailed specifications of each bird can be found in Table 2. As illustrated in Figure 7, we implemented two distinct flight patterns for systematic data



Figure 8. Drones that we collected data during the field experiments

Table 3. Details of drones we collected from.

	(a)	(b)	(c)	(d)	(e)
length	15.9 cm	34.7 cm	55.8 cm	55.1 cm	88.7 cm
width	20.3 cm	28.3 cm	45.7 cm	48.2 cm	88.0 cm
height	5.6 cm	10.7 cm	10.8 cm	30.9 cm	37.8 cm
N_{rotors}	4	4	4	6	4

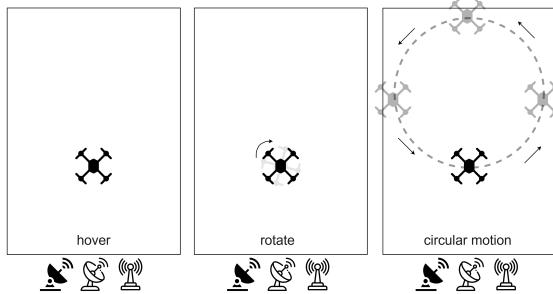


Figure 9. Flight patterns that we collected data for 5 type of drones.

collection:

- forward-backward:
where birds maintained their lateral position while flying back and forth at different distances from the radar
- side-to-side:
where birds maintained a fixed distance from the radar while flying laterally between two points

3.1.3 Drones

For drones, data collection was conducted on March 1, 2024, under mild weather conditions in the morning, with wind speeds increasing to approximately 17 mph by noon. FMCW data across all radars was collected using five different drone models as shown in Figure 8 and detailed in

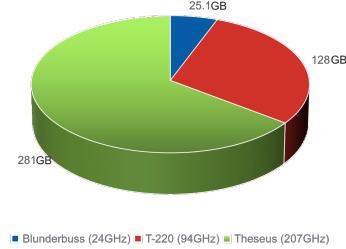


Figure 10. Illustration of the composition of raw data collected from birds.

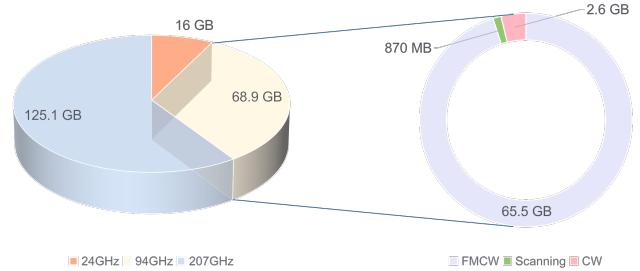


Figure 11. Illustration of the composition of raw data collected from drones.

Table 3: (a) DJI Mini 2, (b) DJI Mavic 3, (c) Autel Evo II, (d) Yuneec H520E, and (e) DJI Matrice 210.

Additional measurements included CW data for models (a), (c), and (e), as well as scanning data for the combination of (c) and (e).

All flight operations were conducted within a designated zone ranging from 30 to 100 meters radial to the radar beam. As shown in Figure 9, three primary flight patterns were implemented:

- hovering at fixed positions (at altitudes of 5, 8, and 13 meters)
 - rotating while hovering (at altitudes of 5, 8, and 13 meters)
 - radial back-and-forth motion (between 30-100 meters)
- Additionally, we collected data for multi-drone scenarios with two specific configurations:
- Autel Evo II and DJI Matrice 210 [(c) + (e)] executing simultaneous figure-of-8 patterns (captured in both FMCW and scanning modes)
 - DJI Mini 2 and DJI Matrice 210 [(a) + (e)] performing simultaneous figure-of-8 patterns (FMCW mode only)

3.2. Summary of Collected Data

3.2.1 Birds

As shown in Figure 10, The birds dataset consists of synchronized FMCW radar measurements collected from three different radar systems operating at different frequency bands. The 207GHz radar contributed 281GB of data, while

Table 4. Examples for labeled bird data.

File name	radar	Video file name	Bird	movement	Absolute start time	clear RTI	Comments
2024-02-06_10-24-39.24GHz.FMCW.dat	Blunderbuss_24GHz T220.94GHz	2024-02-06_10-20-33.mkv	Bald Eagle	forward-backward	10-24-42	T T T	10-24-42 to 12-24-50 fly from 80m to 40m. 3-11
2024-02-06_10-24-28.94GHz.FMCW.dat	Blunderbuss_24GHz T220.94GHz				10-25-22	TF HH HF	10-25-22 to 10-25-28: fly from 40m to 80m; 0-6 10-25-45 to 10-25-57: fly from 80m to side and back to 80m. 23-25
2024-02-06_10-25-22.24GHz.FMCW.dat	Blunderbuss_24GHz T220.94GHz				10-26-29	TT TF TT	10-26-29 to 10-26-36: fly from 80m to 40m; 0-7 10-26-42 to 10-26-50: fly from 40m to 80m. 13-21
2024-02-06_10-25-12.94GHz.FMCW.dat	Blunderbuss_24GHz T220.94GHz				10-28-38	TT TT HT	10-28-38 to 10-28-45: fly from 80m to 40m; 0-7 10-28-51 to 10-28-58: fly from 40m to 80m. 13-20
2024-02-06_10-26-10-29.24GHz.FMCW.dat	Blunderbuss_24GHz T220.94GHz				10-29-41	TT TT TT	10-29-41 to 10-29-49: fly from 80m to 40m; 0-8 10-29-53 to 10-30-00: fly from 40m to 80m. 12-19
2024-02-06_10-26-17.207GHz.FMCW.dat	Blunderbuss_24GHz T220.94GHz			
2024-02-06_10-28-18.94GHz.FMCW.dat	Blunderbuss_24GHz T220.94GHz
2024-02-06_10-28-16.207GHz.FMCW.dat	Blunderbuss_24GHz T220.94GHz
2024-02-06_10-29-41.24GHz.FMCW.dat	Blunderbuss_24GHz T220.94GHz
2024-02-06_10-29-18.94GHz.FMCW.dat	Blunderbuss_24GHz T220.94GHz
2024-02-06_10-29-16.207GHz.FMCW.dat	Blunderbuss_24GHz T220.94GHz

Table 5. Examples for labeled FMCW drone data.

File names	hover	rotate	outbound	inbound	maneuvar	DJI_Mavic_3	DJI_Mini_2	Autel_Evo.II	Yuneec_H520E	DJI_Matrice_210	3m	5m	8m	10m	13m	5 to 13m	clear RTI
2024-03-01_09-25-06.207GHz.FMCW.dat	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
2024-03-01_09-28-20.207GHz.FMCW.dat	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
2024-03-01_09-29-22.207GHz.FMCW.dat	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
2024-03-01_09-30-36.207GHz.FMCW.dat	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1
2024-03-01_09-30-56.207GHz.FMCW.dat	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1
2024-03-01_09-31-27.207GHz.FMCW.dat	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1
...

Table 6. Examples for labeled Scanning drone data.

File name	drone	height	comments
2024-03-01_11-36-59_FMCW.dat	DJI matrice 210 + Autel Evo II	5 to 13m	can observe 2 drones, the Autel Evo II disappears on radar from frame 28 to 43.
2024-03-01_11-37-39_FMCW.dat	DJI matrice 210 + Autel Evo II	5 to 13m	can observe 2 drones.
2024-03-01_11-39-12_FMCW.dat	DJI matrice 210 + Autel Evo II	5 to 13m	only one drone is observed.
...

the 94GHz and 24GHz generated 128GB and 25.1GB of data, respectively. In total, the multi-frequency FMCW dataset amounts to 434.1GB, providing comprehensive observations of the target scenes across different frequency bands.

3.2.2 Drones

As shown in Figure 11, the collected drone data comprises radar measurements collected from three radar systems operating at different frequency bands. The 207GHz radar contributed 125.1GB of FMCW data, while the 24GHz radar generated 16GB of FMCW data. The 94GHz radar collected data in three different modes: 65.5GB of FMCW measurements, 870MB of scanning mode data, and 2.6GB of CW measurements, totaling 68.9GB. The complete multi-frequency dataset amounts to 210GB, providing diverse radar observations across different frequency bands and operational modes.

3.3. Data Labeling

3.3.1 Birds

The data annotation process was primarily driven by video-based validation and verification. Out of the 191 collected radar files, 188 were successfully annotated, with only three files excluded due to the lack of corresponding video cover-

Table 7. Examples for labeled CW drone data.

File name	drone	movement	height	blade flash observed
2024-03-01_12-05-23.CW.dat	DJI Matrice 210	hover	5m	T
2024-03-01_12-06-15.CW.dat	DJI Matrice 210	hover	5m	T
2024-03-01_12-06-30.CW.dat	DJI Matrice 210	hover	5m	T
2024-03-01_12-06-57.CW.dat	DJI Matrice 210	hover	10m	T
2024-03-01_12-07-21.CW.dat	DJI Matrice 210	hover	10m	T
2024-03-01_12-07-50.CW.dat	DJI Matrice 210	hover	15m	T
2024-03-01_12-08-03.CW.dat	DJI Matrice 210	hover	15m	T
2024-03-01_12-08-20.CW.dat	DJI Matrice 210	hover	15m	T
2024-03-01_12-08-52.CW.dat	DJI Matrice 210	outbound-inbound	8m	T
...

age during their recording periods.

The annotation workflow followed a systematic approach, where each video recording was carefully analyzed to document all bird activities and human interactions in chronological order. These observations were then cross-referenced with the radar data files to ensure temporal alignment between the radar measurements and the observed events. This cross-validation process was crucial for establishing the ground truth of the dataset.

The annotation schema was comprehensive, documenting data file identifiers, corresponding radar system names, associated video file references, bird species identification, flight pattern classification, event timing information, if the corresponding Range Time Intensity (RTI) plot has clear trace or not, and additional contextual comments.

To maintain clear distinction between data from different radar systems, a color-coding system was implemented

in the annotation process, with each of the three radar systems assigned a unique color identifier. This systematic approach ensured consistent and traceable documentation of all observed events across the multi-radar dataset. Table 4 showcases an example that how the annotation is formatted.

3.3.2 Drones

The drone data annotation process employed a systematic approach utilizing MATLAB's Image Labeler application², which significantly improved the efficiency and accuracy of the labeling process compared to manual annotation methods used in bird data analysis. The annotation schema was designed to capture both behavioral characteristics and scene-specific attributes.

FMCW data For FMCW radar data, the annotation framework incorporated two main categories of labels. The first category focused on drone behavior classification, including hover, rotation, near-to-far movement, far-to-near movement, and maneuver patterns. The 'maneuver' designation was specifically reserved for multi-drone scenarios where complex flight patterns made precise behavior classification challenging.

The second category addressed scene-specific attributes, documenting drone models, flight altitudes, and the visibility of radar traces in RTI plots. A notable adaptation was made for multi-drone scenarios where altitude variations required a broader classification range (5-13m). This broader range accommodated scenarios where drones operated at varying altitudes, with the provision for later refinement using precise flight logs from the drone manufacturer. The annotation data was exported in MATLAB's native format (.mat), containing comprehensive information including file identifiers, label definitions with corresponding descriptions, and the annotated segments. As the final results shown in Table 5, post-processing involved data cleaning and standardization, resulting in a binary classification matrix where positive and negative attributes were represented as 1 and 0, respectively.

Scanning and CW data As shown in Table 6 and Table 7, the annotation approach for Scanning and CW radar data followed a simplified schema compared to the FMCW data labeling process. These datasets were annotated using a tabular format, documenting essential parameters including file identifiers, drone model specifications, flight altitudes, and significant event markers. This streamlined annotation format was designed to capture the fundamental characteristics of drone activities while maintaining consistency across different radar operational modes.

²<https://mathworks.com/help/vision/ug/get-started-with-the-image-labeler.html>

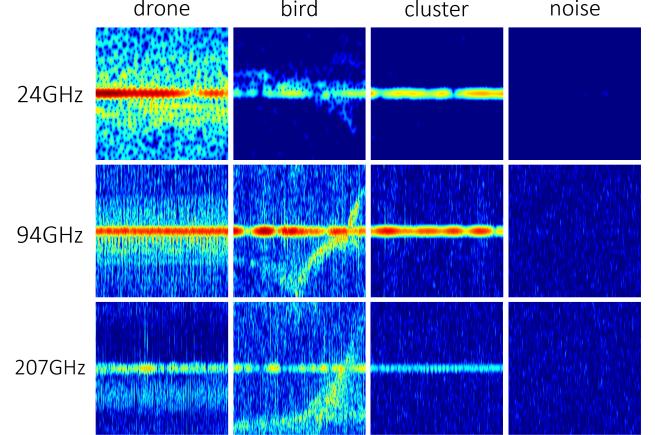


Figure 12. Examples of generated micro-Doppler images for different categories and different frequencies.

3.4. Deep Learning Dataset Generation

To be specific, we aim to develop deep learning models for 3 tasks:

- binary classification: the deep learning model will classify whether the image is drone or non-drone.
- drone classification: the deep learning model will classify which drone type the input image is in.
- bird-drone-clutter-noise classification: the deep learning model will classify which category list above the input image is in.

We illustrate (1) how we convert raw files collected by radars to sequences of square micro-Doppler images within the same time duration. (2) how we split micro-Doppler image dataset into trainset and testset. It is worth noting that, to align with the bird data, which was only collected in FMCW mode, we only use the FMCW drone data for further steps. Figure 12 illustrates the generated images for different categories and frequencies.

3.4.1 Extra Annotations and Hyperparameters

For all data, we first abstract adjustable parameters in spectrogram generation code, enabling adaptability to different radar signal processing parameters. This will facilitate generation of images under various radar signal processing parameter settings, meeting the potential requirements of different neural networks.

To ensure the reproducibility, extra annotations were conducted. CSV files were used to document the start and end range of the signal representing interested objects in RTI plot. To be specific, we focus on three main objects: birds, drones, and clutters. We also identified some signals as noise.

The image generation process involved creating standardized 512×512 pixel images from the annotated radar

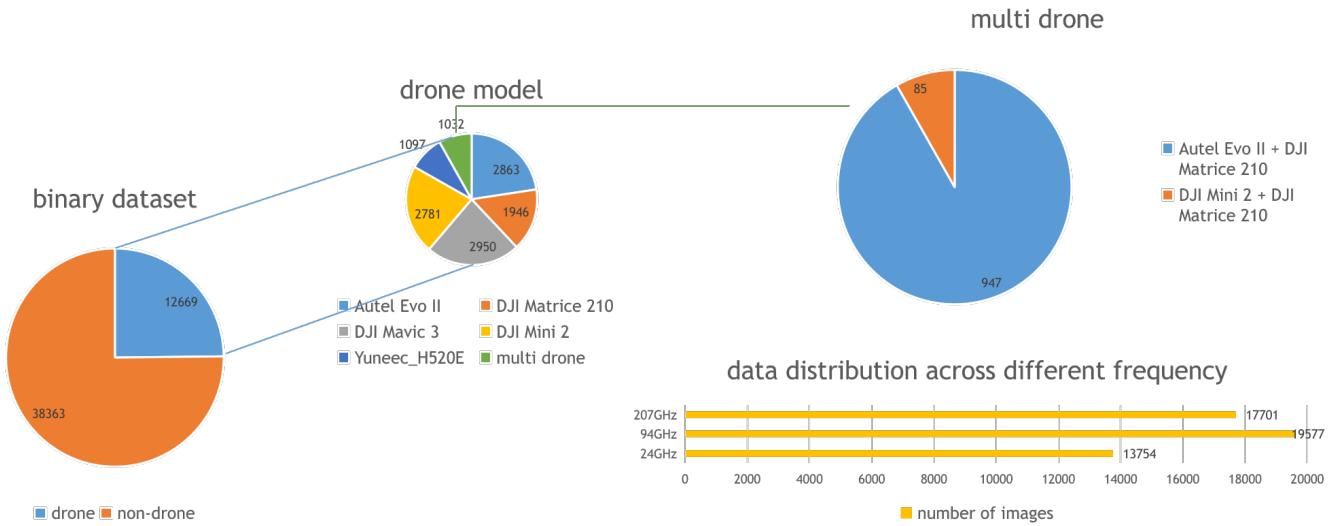


Figure 13. Dataset distribution for all generated birds' and drones' micro-Doppler images.

data. Each image was generated using specific temporal parameters: a window length of 256 samples, a time interval of 0.25 units, and a time overlap of 0.01 units between consecutive windows. The generated images were systematically stored in timestamped folders following the format 'YYYY-MM-DD_HH-MM-SS', which is matched with that of the corresponding raw radar data file.

To ensure data quality, a manual validation process was implemented. This included the removal of inadequate data segments, particularly those too brief to provide meaningful information. The validated images underwent a cross-labeling procedure, where all 512×512 images were cataloged in a comprehensive CSV file. Each entry was verified by cross-referencing the timestamped folder names with drone labels to ensure consistency in drone type and radar frequency specifications.

Following the established data generation and quality control procedures, we compiled a preliminary multi-radar dataset as shown in Figure 13. The binary dataset comprises 51,032 images in total, with 12,669 drone images and 38,363 non-drone images. For more detailed drone classification tasks, the drone images are further categorized into six classes: Autel Evo II (2,863 images), DJI Matrice 210 (1,946 images), DJI Mavic 3 (2,950 images), DJI Mini 2 (2,781 images), Yuneec H520E (1,097 images), and multi-drone scenarios (1,032 images).

For multi-drone scenarios, which account for 1,032 images in total, we primarily observed two types of combinations: Autel Evo II with DJI Matrice 210 (947 images) and DJI Mini 2 with DJI Matrice 210 (85 images). This multi-drone subset provides valuable data for analyzing radar signatures of simultaneous drone operations.

The dataset exhibits a comprehensive frequency distribution across three radar bands: 19,577 images from

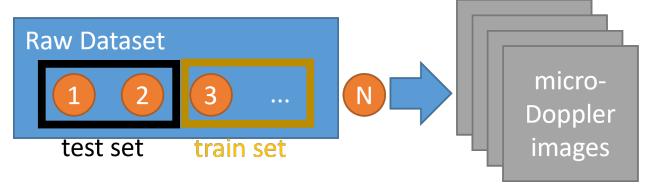


Figure 14. Illustration of dataset split protocol.

94GHz, 17,701 images from 207GHz, and 13,754 images from 24GHz radar systems. This multi-frequency approach ensures robust coverage across different radar operational modes, capturing diverse aspects of target signatures.

For the subsequent construction of balanced datasets for all three tasks, we implemented a strategic data selection process to maintain class equilibrium. While our data generation pipeline produced a substantial amount of radar images, we deliberately used only a subset which maintain a balance among each category.

3.4.2 Dataset Splitting Protocol

To maintain the integrity of model evaluation and prevent data leakage, we implemented a single-trial-independent data splitting strategy. Since multiple consecutive images were generated from each raw radar file, conventional random splitting at the image level could potentially lead to data leakage, where highly correlated images from the same trial might appear in both training and testing sets. Therefore, as shown in Figure 14, we adopted a trial-level randomization approach, where the splitting was performed on the raw radar files before image generation. This approach ensures that all images generated from the same radar trial are assigned to the same dataset partition. This splitting pro-

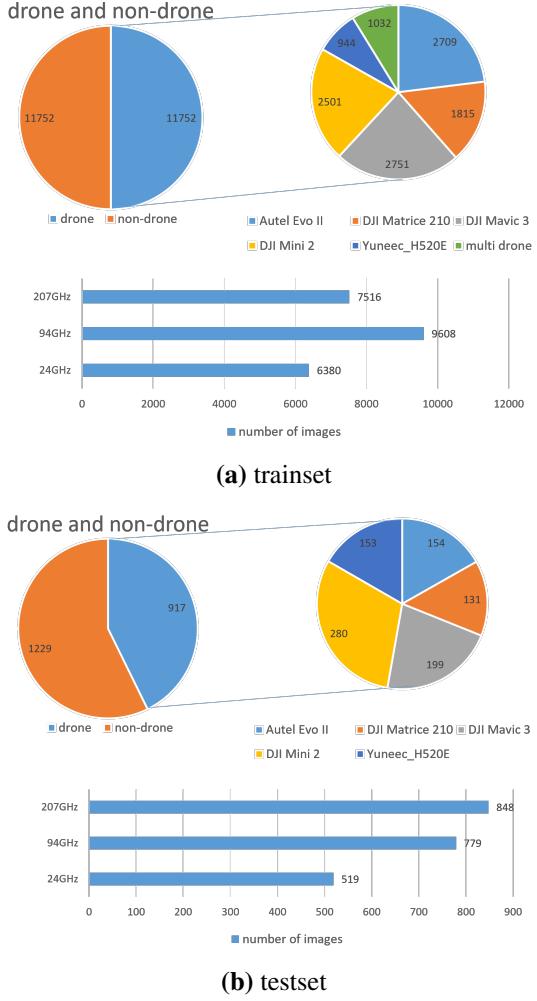


Figure 15. Data distribution of the split dataset for binary classification and drone classification.

tocol effectively maintains the independence between training and testing sets, providing a more reliable evaluation of model performance and preventing overly optimistic assessment due to temporal correlation in the radar measurements.

3.4.3 Dataset for Binary and Drone Classification

As shown in Figure 15, the dataset was partitioned into training and testing sets following the single-trial-independent splitting protocol, with an approximate ratio of 10:1. The training set consisted of 23,504 images, with an equal distribution of 11,752 images for both drone and non-drone classes, ensuring a balanced binary classification task. Within the drone category, the data encompassed various drone models: Autel Evo II (2,709 images), DJI Matrice 210 (1,815 images), DJI Mavic 3 (2,751 images), DJI Mini 2 (2,501 images), Yuneec H520E (944 images), and

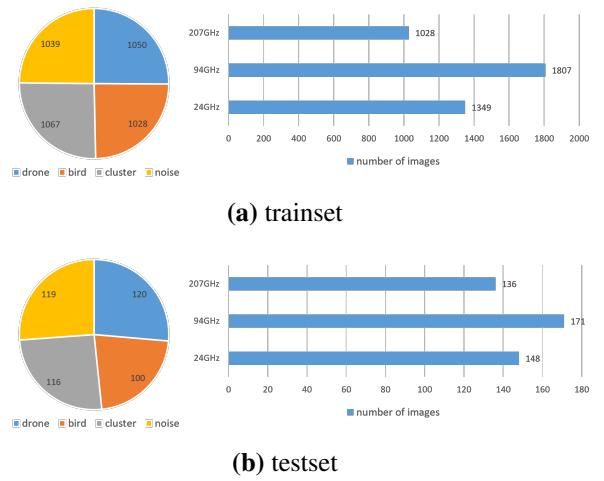


Figure 16. Data distribution of the split dataset for bird-drone-clutter-noise classification.

multi-drone scenarios (1,032 images). The training data was collected across three frequency bands, with 9,608 images from 94GHz, 7,516 images from 207GHz, and 6,380 images from 24GHz radar systems.

The testing set, representing approximately 10% of the total dataset, comprised 2,146 images, with 917 drone images and 1,229 non-drone images. The drone images were distributed among five different models: Autel Evo II (154 images), DJI Matrice 210 (131 images), DJI Mavic 3 (199 images), DJI Mini 2 (280 images), and Yuneec H520E (153 images). The frequency distribution in the testing set showed 848 images from 207GHz, 779 images from 94GHz, and 519 images from 24GHz radar systems.

3.4.4 Dataset for Bird-Drone-Clutter-Noise Classification

For the Bird-Drone-Clutter-Noise classification task, the dataset was similarly partitioned following the single-trial-independent splitting protocol with an approximate ratio of 10:1. The training set contained a total of 4,184 images, with a well-balanced distribution across all four classes: 1,050 drone images, 1,028 bird images, 1,067 cluster images, and 1,039 noise images. The training data was collected across three frequency bands, with 1,807 images from 94GHz, 1,349 images from 24GHz, and 1,028 images from 207GHz radar systems.

The testing set comprised 455 images, representing approximately 10% of the total dataset. The distribution among classes maintained similar balance with 120 drone images, 100 bird images, 116 cluster images, and 119 noise images. The frequency distribution in the testing set showed 171 images from 94GHz, 148 images from 24GHz, and 136

Table 8. Accuracy of different neural network architectures on three tasks with various combination of initializing setup and computational budget. Notably “w/ pt” refers to initialized with ImageNet pretrained weights, while “w/o pt” refers to initialized from scratch.

(a) w/o pt for 20 epochs.				(b) w/ pt for 20 epochs.			
architecture	binary	drone	four-class	architecture	binary	drone	four-class
AlexNet	99.95%	91.28%	88.79%	AlexNet	100%	97.60%	94.29%
ConvNeXt-base	97.72%	67.50%	83.30%	ConvNeXt-base	100%	99.02%	92.09%
ConvNeXt-tiny	98.14%	69.36%	79.56%	ConvNeXt-tiny	100%	98.91%	91.43%
EfficientNetV2-m	98.51%	33.48%	69.23%	EfficientNetV2-m	100%	96.73%	99.34%
EfficientNetV2-s	99.63%	47.98%	79.12%	EfficientNetV2-s	100%	98.04%	97.36%
ResNet18	100%	94.77%	87.25%	ResNet18	100%	96.29%	94.73%
ResNet50	99.86%	80.04%	77.80%	ResNet50	100%	94.66%	93.63%
ResNeXt50	99.91%	77.86%	82.42%	ResNeXt50	100%	96.84%	91.43%

(c) w/o pt for 100 epochs.				(d) w/ pt for 100 epochs.			
architecture	binary	drone	four-class	architecture	binary	drone	four-class
AlexNet	100%	98.80%	98.02%	AlexNet	100%	98.26%	94.29%
ConvNeXt-base	99.86%	93.46%	87.03%	ConvNeXt-base	100%	99.89%	95.16%
ConvNeXt-tiny	99.77%	94.44%	86.59%	ConvNeXt-tiny	100%	99.56%	98.02%
EfficientNetV2-m	100%	95.20%	84.62%	EfficientNetV2-m	100%	99.35%	99.34%
EfficientNetV2-s	100%	92.80%	85.05%	EfficientNetV2-s	100%	99.35%	98.68%
ResNet18	100%	97.60%	91.21%	ResNet18	100%	96.84%	94.73%
ResNet50	100%	90.73%	83.96%	ResNet50	100%	96.84%	99.34%
ResNeXt50	100%	88.11%	86.37%	ResNeXt50	100%	98.69%	93.63%

images from 207GHz radar systems. This balanced distribution across both classes and frequency bands ensures robust evaluation of the classification model’s performance.

3.4.5 Simulation Dataset for Transferability

In addition to the real-world radar datasets described above, we generated a simulation dataset to evaluate model transferability across different radar scenarios. This synthetic dataset complements our real-world data by providing controlled test conditions with precisely known parameters. Following the modeling approach detailed in Section 2, we created a balanced dataset comprising 2,000 training images (1,000 each for birds and drones) and 200 testing images (100 each for birds and drones).

To maximize the dataset’s representativeness and challenge the models’ generalization capabilities, we employed a comprehensive parameter sampling strategy. Rather than using fixed parameter combinations as in controlled real-world data collection, we randomly sampled parameters from their predefined valid ranges for each simulated image. This approach ensures broad coverage of potential real-world scenarios, including various combinations of target velocities, radar cross sections, flight trajectories, and environmental conditions. Such diverse parameter combinations help assess whether models trained on our real-world dataset can effectively generalize to previously unseen target signatures and operational conditions.

4. Neural Networks Modeling

4.1. Backbone Models

For this radar image classification task, we selected a diverse set of CNN architectures as backbone models, ranging from classic networks to modern efficient designs:

- AlexNet [4]: Despite its relatively simple architecture, it serves as a baseline model and provides insights into whether our radar classification tasks require deep architectural complexity. Its straightforward structure also offers good interpretability of feature learning processes.
- ResNet 18 and 50 [3]: These models represent different depths of the classical residual learning framework. ResNet-18 provides a lightweight option while maintaining good performance, whereas ResNet-50 offers deeper feature extraction capabilities. Their skip connections help mitigate the vanishing gradient problem, which is particularly important for radar signal patterns.
- ResNeXt 50 [14]: Building upon ResNet’s architecture, it introduces cardinality through grouped convolutions, potentially beneficial for capturing diverse radar signal characteristics at different frequencies and target behaviors.
- EfficientNet S and M [13]: These models, optimized through compound scaling, offer an excellent balance between computational efficiency and model capacity. Their adaptive receptive field sizing could be particularly advantageous for detecting targets at various ranges in radar images.
- ConvNeXt Tiny and Base [5]: As modern architectures

Table 9. Models transferability evaluation results on the simulation dataset. The models are trained on real-world radar data and tested on simulated data. Notably “w/ pt” refers to initialized with ImageNet pretrained weights, while “w/o pt” refers to initialized from scratch.

(a) w/o pt for 20 epochs.				(b) w/ pt for 20 epochs.					
	bird		drone			bird		drone	
	binary	four-class	binary	four-class		binary	four-class	binary	four-class
AlexNet	99%	29%	50%	21%	AlexNet	100%	81%	81%	73%
ConvNeXt-base	93%	63%	67%	30%	ConvNeXt-base	100%	77%	47%	92%
ConvNeXt-tiny	97%	68%	69%	25%	ConvNeXt-tiny	95%	48%	73%	77%
EfficientNetV2-m	99%	62%	52%	39%	EfficientNetV2-m	99%	59%	38%	78%
EfficientNetV2-s	94%	100%	34%	21%	EfficientNetV2-s	94%	50%	5%	84%
ResNet18	100%	93%	18%	20%	ResNet18	97%	58%	88%	45%
ResNet50	100%	100%	45%	21%	ResNet50	73%	22%	59%	84%
ResNeXt50	100%	100%	32%	21%	ResNeXt50	94%	65%	75%	91%

(c) w/o pt for 100 epochs.				(d) w/ pt for 100 epochs.					
	bird		drone			bird		drone	
	binary	four-class	binary	four-class		binary	four-class	binary	four-class
AlexNet	93%	58%	68%	23%	AlexNet	100%	81%	86%	73%
ConvNeXt-base	98%	78%	70%	24%	ConvNeXt-base	100%	78%	51%	84%
ConvNeXt-tiny	100%	68%	92%	17%	ConvNeXt-tiny	98%	28%	68%	79%
EfficientNetV2-m	100%	94%	65%	42%	EfficientNetV2-m	99%	55%	46%	62%
EfficientNetV2-s	98%	99%	87%	51%	EfficientNetV2-s	100%	43%	3%	78%
ResNet18	100%	91%	22%	36%	ResNet18	97%	58%	83%	45%
ResNet50	100%	100%	21%	16%	ResNet50	78%	29%	50%	85%
ResNeXt50	100%	100%	52%	18%	ResNeXt50	100%	48%	62%	91%

inspired by Vision Transformers but implemented purely with convolutions, they provide robust feature extraction with modernized design choices. Their hierarchical structure could be especially effective for capturing both fine-grained radar signatures and broader movement patterns.

4.2. Experiment Setup

To ensure consistent and reproducible results, we maintained fixed hyperparameters across all experiments: a learning rate of 1×10^{-5} , batch size of 64, and a random seed of 3407 [7]. To make the training process faster, we also set a early stop point, where the difference of loss between current epoch and last epoch is less than 1×10^{-4} .

For model initialization, we explored two strategies: random initialization and the weights pretrained on ImageNet [1]. The adoption of pre-trained weights is particularly beneficial for our radar image classification tasks despite the domain difference between natural images and radar spectrograms. This approach leverages the hierarchical feature extraction capabilities already learned from large-scale image datasets, where lower-level features (such as edges, textures, and basic patterns) are often transferable across domains. Moreover, transfer learning from ImageNet pre-trained models can potentially accelerate convergence and improve model stability, especially valuable when dealing with specialized radar data where the training samples are relatively limited compared to natural image datasets.

We established two distinct computational budgets to evaluate model performance under different training sce-

narios: a resource-constrained setting with 20 epochs and an extended training regime with 100 epochs. This dual-budget approach allows us to assess both the rapid adaptation capabilities of different architectures and their ultimate performance given sufficient training time. The 20-epoch setting is particularly relevant for practical applications where quick model deployment is required, while the 100-epoch setting helps understand the full potential of each architecture when computational resources are not a limiting factor.

4.3. Results

As shown in Table 8, the experimental results demonstrate several clear trends across different network architectures, weights initialization, and computational budgets.

To provide insights into the decision-making process of different architectures, we conducted an extensive visualization analysis using Grad-CAM [12] across all experimental configurations. Given the comprehensive nature of our experiments, we generated a substantial collection of visualization results. While the complete set of Grad-CAM visualizations would be impractical to include in this paper due to space constraints, we have made the full collection available on our [GitHub repository](#).

4.3.1 Analysis of Weights Initialization

The use of ImageNet pre-trained weights for network weights initialization shows significant advantages across all architectures and tasks. This benefit is particularly pro-

Table 10. Cross-domain evaluation results of models trained on simulation data. The models are tested on both real data and simulation data. Notably “w/ pt 100” refers to initialized with ImageNet pretrained weights and trained for 100 epochs, while “w/o pt” refers to initialized from scratch and trained for 100 epochs.

(a) accuracy on testset of simulation data.								
	w/o pt 100		w/ pt 100		(b) accuracy on testset of real data.			
	bird	drone	bird	drone		w/o pt 100	w/ pt 100	
AlexNet	100%	100%	100%	100%	AlexNet	6%	98%	33% 100%
ConvNeXt-base	100%	100%	100%	100%	ConvNeXt-base	67%	38%	15% 99%
ConvNeXt-tiny	100%	100%	100%	100%	ConvNeXt-tiny	66%	41%	0% 100%
EfficientNetV2-m	100%	100%	100%	100%	EfficientNetV2-m	19%	95%	13% 93%
EfficientNetV2-s	100%	100%	100%	100%	EfficientNetV2-s	51%	88%	2% 100%
ResNet18	100%	100%	100%	100%	ResNet18	2%	100%	0% 100%
ResNet50	100%	100%	100%	100%	ResNet50	35%	63%	0% 100%
ResNeXt50_32x4d	100%	100%	100%	100%	ResNeXt50_32x4d	33%	80%	0% 100%

nounced in the 20-epoch setting, where pre-trained models consistently outperform their randomly initialized counterparts. For instance, in the drone classification task at 20 epochs, EfficientNetV2-m improved dramatically from 33.48% to 96.73% with pre-training. This validates our hypothesis that transfer learning, even from a different domain, provides valuable initialization for radar image classification tasks.

4.3.2 Analysis of Computational Budget

Extending the training from 20 to 100 epochs shows different impacts between pre-trained and randomly initialized models. Randomly initialized models benefit substantially from the increased training time, notably closing the performance gap with their pre-trained counterparts. However, pre-trained models achieve high performance even with limited training epochs, suggesting they are more suitable for scenarios with computational constraints.

4.3.3 Analysis of Neural Network Architecture

Among all tested architectures, several models showed consistently strong performance:

- ConvNeXt (both base and tiny variants) demonstrated exceptional performance in drone classification with pre-training, achieving up to 99.89% accuracy. Their modern architecture design appears particularly well-suited for capturing complex radar signatures.
- EfficientNetV2 models, especially with pre-training, showed remarkable performance in the four-class task (up to 99.34% accuracy), suggesting their compound scaling principles effectively capture multi-class radar signal characteristics.
- ResNet18, despite its relatively simple architecture, proved highly competitive across all tasks, offering an ex-

cellent balance between performance and computational efficiency.

4.3.4 Analysis of Tasks

Binary classification All architectures achieved near-perfect performance (100%) in the binary classification task with pre-training, indicating that the drone/non-drone distinction is well-captured by modern CNN architectures regardless of their specific design choices.

Multi-class Performance The more challenging drone classification and four-class tasks revealed greater differentiation between architectures. Modern architectures (ConvNeXt and EfficientNetV2) generally outperformed older designs, particularly with pre-training, suggesting their architectural innovations better capture the subtle differences between different drone types and radar signatures.

4.3.5 Transferability on Simulation Data

As shown in Table 9, the transferability evaluation on our simulation dataset reveals several interesting patterns across different architectures and training configurations.

Overall, the results highlight that while our models can effectively transfer knowledge for bird detection, drone detection presents more challenges in bridging the gap between real and simulated data.

Performance on bird detection Models demonstrate strong transferability in bird-related tasks, particularly in binary classification. Most architectures achieve high accuracy (>90%) in binary bird classification, with several models reaching 100% accuracy. This suggests that the bird signatures learned from real radar data generalize well to simulated scenarios.

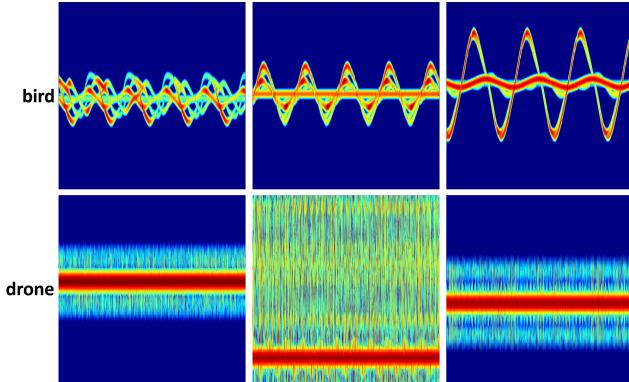


Figure 17. Some examples for simulated micro-Doppler images. Those images are crafted with arbitrary parameter combinations, which can cover various objects.

Performance on drone detection In contrast, drone detection shows notably lower transferability. Binary classification accuracy for drones ranges from 3% to 92%, indicating that drone signatures learned from real data may not fully capture the characteristics present in simulated scenarios. This disparity could be attributed to the greater complexity and variability in drone flight patterns and radar signatures compared to birds.

Impact of pre-training Pre-trained models generally show better transferability, particularly evident in the four-class classification task. For instance, with pre-training and 20 epochs, ConvNeXt-base achieves 92% accuracy on four-class drone classification, compared to only 30% without pre-training. This suggests that ImageNet pre-training helps models learn more robust and generalizable features.

Architecture-specific observations Modern architectures (ConvNeXt and ResNeXt50) demonstrate more consistent transferability across tasks, particularly in the four-class scenario with pre-training. EfficientNetV2 models show mixed results, with strong performance in bird classification but variable results in drone detection. Simpler architectures like AlexNet show surprisingly robust transferability, maintaining relatively stable performance across different configurations.

4.3.6 Train with Simulation Data

To further investigate the bidirectional transferability between simulated and real-world radar data, we conducted a complementary experiment by training models on the simulation dataset (shown in Figure 17) and evaluating their performance on both simulated and real-world test sets. This experiment aims to assess whether models trained on simulated data can effectively generalize to real-world scenarios,

and to understand the domain gap between synthetic and real radar signatures. As shown in Table 10, analysis of the results reveals several interesting findings.

Performance on simulation data All architectures achieve perfect accuracy (100%) on the simulated test set, regardless of pre-training status and architecture choice. This indicates that the models can effectively learn and classify the synthetic patterns when both training and testing data come from the same simulated distribution.

Performance on real data

Drone detection: Models show surprisingly good transferability in drone detection, with most architectures achieving high accuracy (>90%) on real data, particularly with pre-training. This suggests that our simulation approach effectively captures the essential characteristics of drone radar signatures.

Bird detection: In contrast, the transfer performance for bird detection is notably poor, with accuracies ranging from 0% to 67%. This significant performance drop indicates a substantial domain gap between simulated and real bird signatures, suggesting that our current simulation approach may not fully capture the complexity and variability of bird radar signatures in real-world conditions.

Pre-training impact: Interestingly, pre-training shows a negative impact on bird detection performance while generally improving drone detection accuracy, suggesting different transfer dynamics for these two classes.

5. Conclusion

In this study, we presented a comprehensive investigation of radar-based target classification, from theoretical modeling to practical deep learning implementation. Our work encompasses several key contributions and findings:

- **Radar Signature Modeling:** We established theoretical models for both bird and drone radar signatures, providing a foundation for understanding their distinctive characteristics in radar observations and enabling the generation of simulation data.
- **Data Collection and Processing:** We conducted extensive real-world data (6 species of birds and 5 types of drones) collection using multi-frequency radar systems (24GHz, 94GHz, and 207GHz), followed by systematic annotation and processing procedures. This resulted in well-curated datasets for three classification tasks: binary drone detection, drone type classification, and four-class (bird-drone-clutter-noise) discrimination.
- **Dataset Generation:** We developed two complementary datasets:
 - A real-world dataset with carefully balanced class distributions and a single-trial-independent splitting pro-

- tocol to prevent data leakage.
- A simulation dataset based on our theoretical models, enabling controlled experiments and transferability analysis.
 - Deep Learning Implementation: Through comprehensive experiments with various CNN architectures, we found that:
 - Modern architectures like ConvNeXt and EfficientNetV2 demonstrate superior performance in complex classification tasks
 - ImageNet pre-training significantly improves model performance, especially under limited computational budgets
 - All tested architectures achieve excellent performance in binary drone detection
 - Cross-Domain Analysis: Our investigation into model transferability between real and simulated radar data revealed interesting asymmetries in how models generalize across domains, particularly the contrast between bird and drone detection capabilities.

These findings suggest several promising directions for future research:

- Refining theoretical models to better capture real-world radar signatures
- Improving simulation fidelity to reduce the domain gap
- Developing radar-specific neural network architectures
- Exploring advanced transfer learning techniques for cross-domain generalization

Our results demonstrate both the potential and challenges of deep learning in radar-based target classification, while providing valuable insights for future development in this field.

References

- [1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [12](#)
- [2] Jiangkun Gong, Deren Li, Jun Yan, Huiping Hu, and Deyong Kong. Comparison of radar signatures from a hybrid vtol fixed-wing drone and quad-rotor drone. *Drones*, 6(5):110, 2022. [1](#)
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [11](#)
- [4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. [11](#)
- [5] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, 2022. [11](#)
- [6] Pavlo Molchanov, Ronny IA Harmann, Jaco JM de Wit, Karen Egiazarian, and Jaakko Astola. Classification of small uavs and birds by micro-doppler signatures. *International Journal of Microwave and Wireless Technologies*, 6(3-4):435–444, 2014. [1](#)
- [7] David Picard. Torch. manual_seed (3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision. *arXiv preprint arXiv:2109.08203*, 2021. [12](#)
- [8] Samiur Rahman and Duncan A Robertson. Millimeter-wave micro-doppler measurements of small uavs. In *Radar Sensor Technology XXI*, pages 307–315. SPIE, 2017. [1](#)
- [9] Samiur Rahman and Duncan A Robertson. Classification of drones and birds using convolutional neural networks applied to radar micro-doppler spectrogram images. *IET radar, sonar & navigation*, 14(5):653–661, 2020. [1](#)
- [10] Arne Schroder, Matthias Renker, Uwe Aulenbacher, Axel Murk, Urs Boniger, Roland Oechslin, and Peter Wellig. Numerical and experimental radar cross section analysis of the quadrocopter dji phantom 2. In *2015 IEEE Radar Conference*, pages 463–468. IEEE, 2015. [1](#)
- [11] Ulzhalgas Seidaliyeva, Lyazzat Ilipbayeva, Kyrmyzy Tais-sariyeva, Nurzhigit Smailov, and Eric T Matson. Advances and challenges in drone detection and classification techniques: A state-of-the-art review. *Sensors*, 24(1):125, 2023. [1](#)
- [12] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. [12](#)
- [13] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. [11](#)
- [14] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [11](#)
- [15] Jun Yan, Huiping Hu, Jiangkun Gong, Deyong Kong, and Deren Li. Exploring radar micro-doppler signatures for recognition of drone types. *Drones*, 7(4):280, 2023. [1](#)

Project Report: Machine Learning-based Drone and Bird Classification by Micro-Doppler Radar Signatures

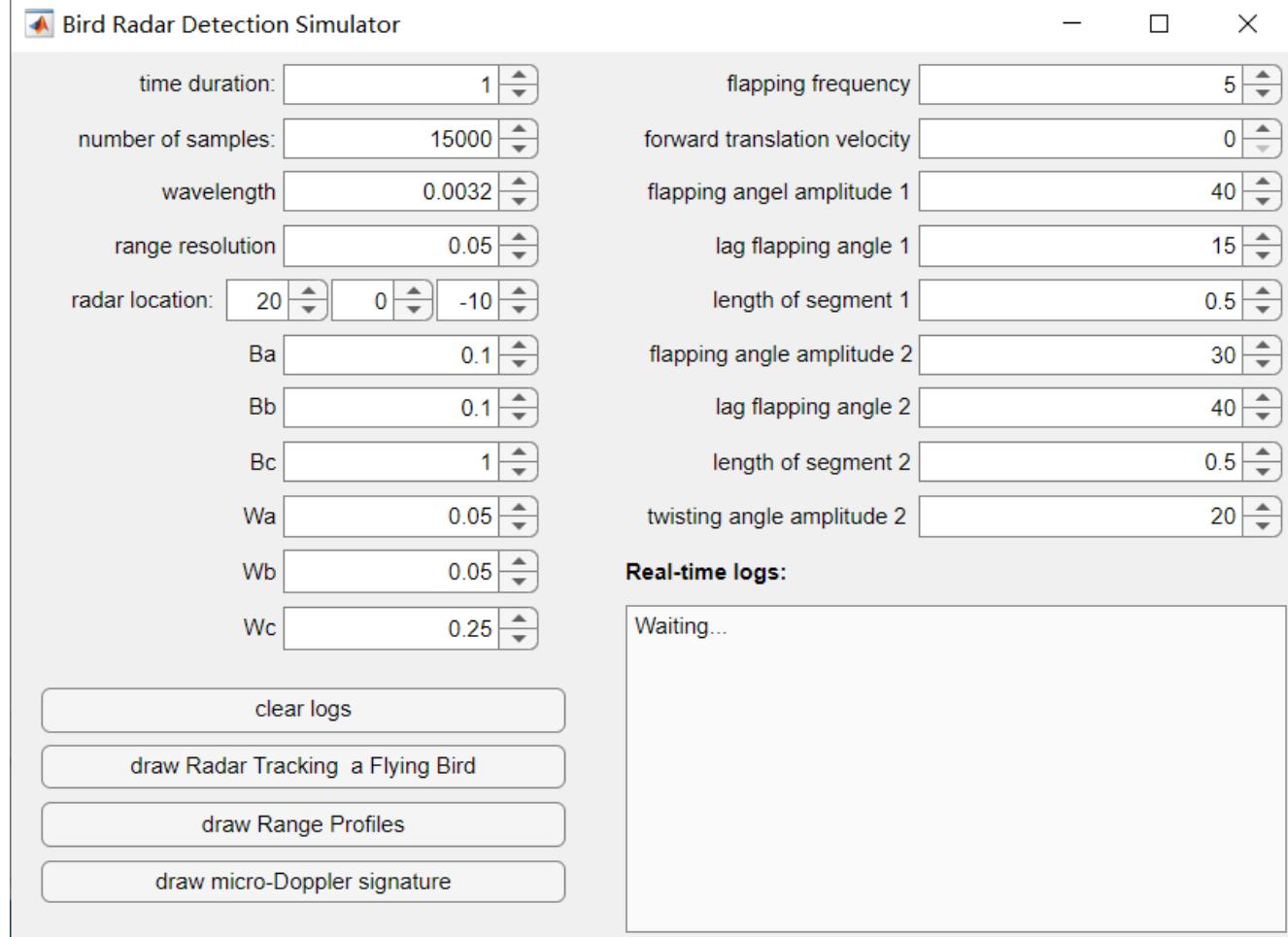
Supplementary Material

A. Graphical User Interface

A.1. GUI for Bird Simulation

A.1.1 Main Interface

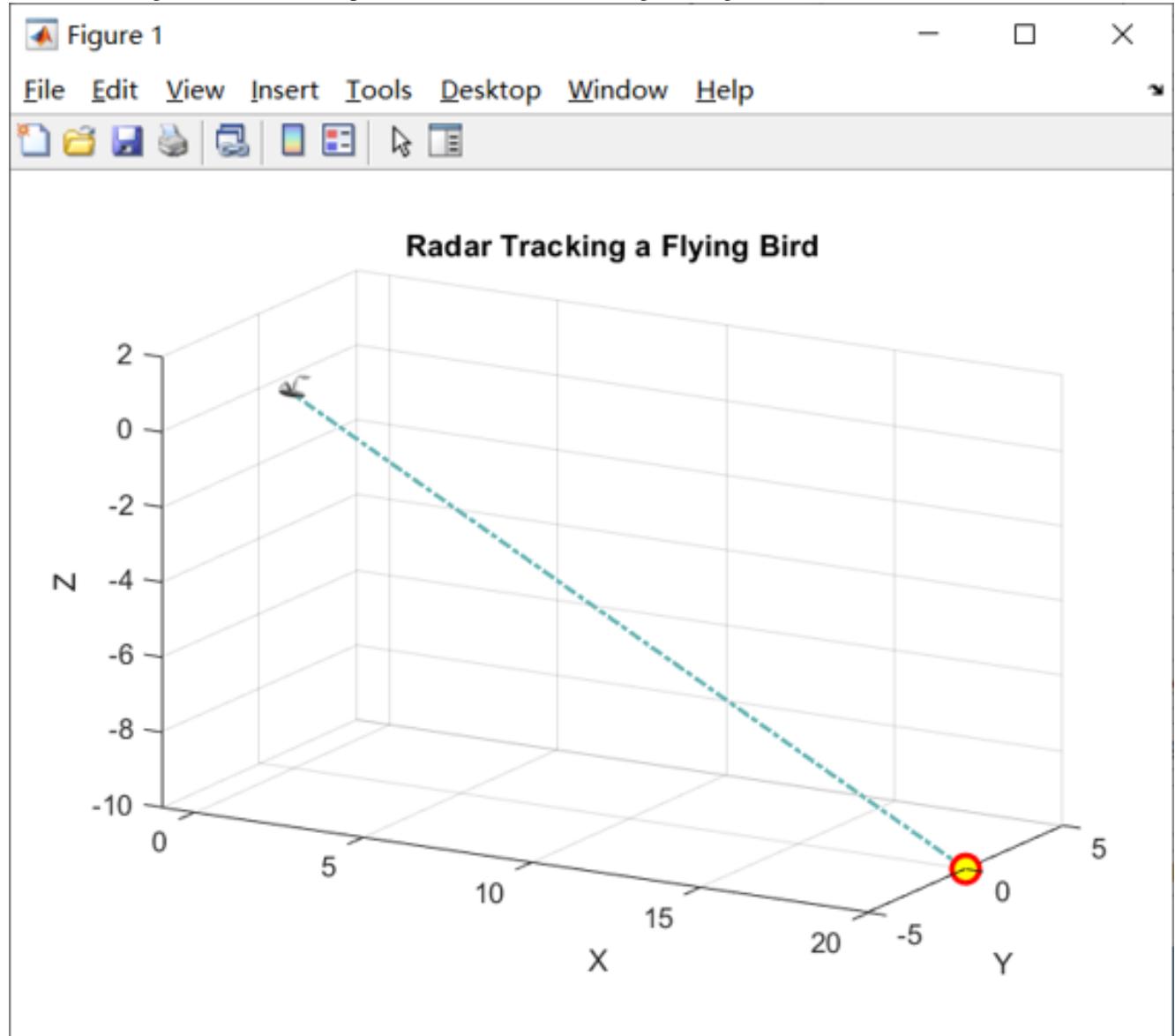
For bird radar signature simulation, we developed a graphical user interface (GUI) that allows flexible parameter configuration, as shown in Figure:



The interface enables users to adjust key simulation parameters including temporal settings (time duration, number of samples), radar specifications (wavelength, range resolution, radar location), body and wing reflection coefficients (B_a , B_b , B_c and W_a , W_b , W_c respectively), and bird motion characteristics (flapping frequency, forward translation velocity, flapping angle amplitudes, and segment lengths). The GUI provides three visualization options: radar tracking of the flying bird, range profiles, and micro-Doppler signature generation. A real-time log window displays the simulation status and progress.

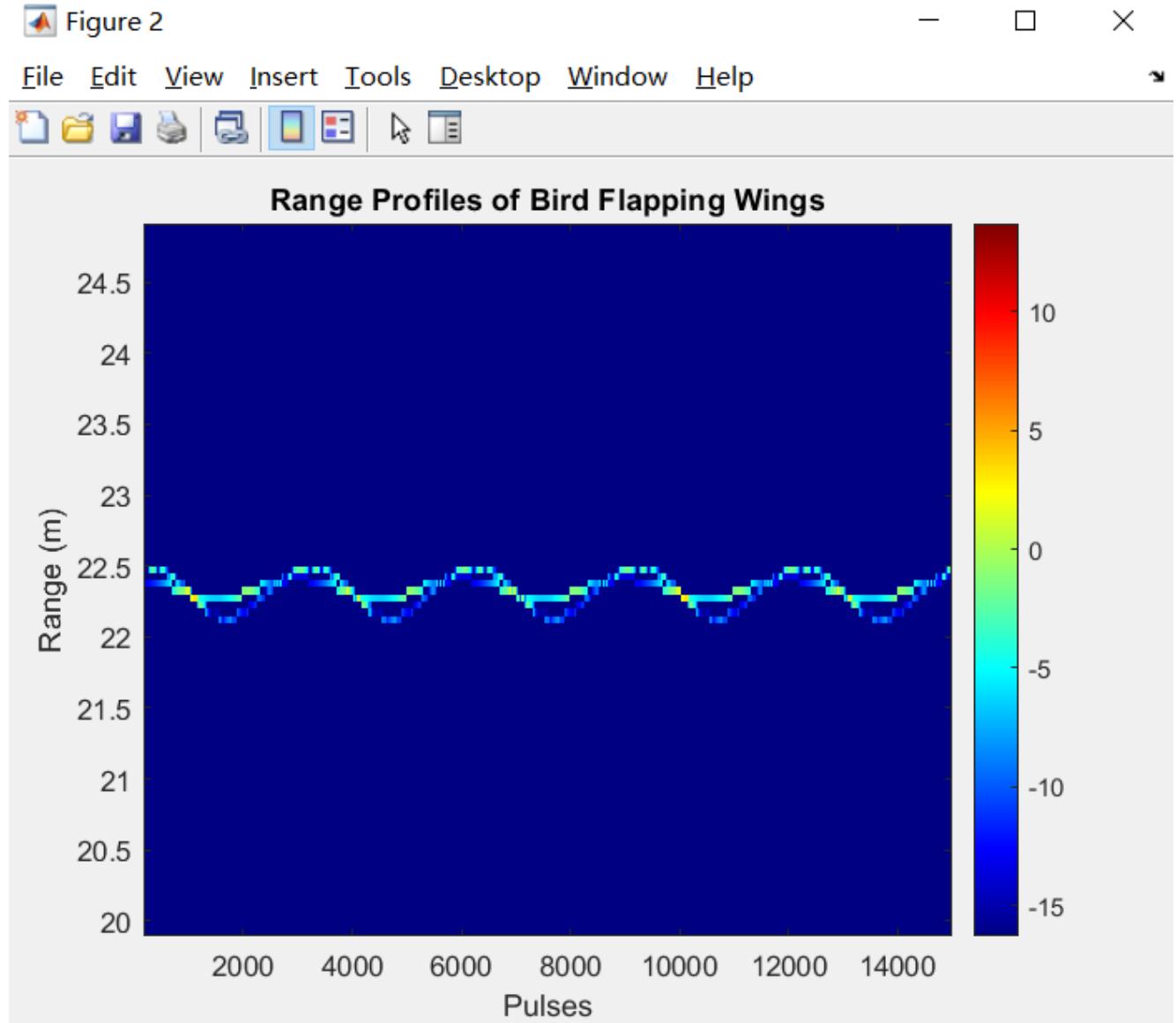
A.1.2 Bird Flying Tracking

The “Radar Tracking a Flying Bird” function generates a real-time 3D plot displaying the spatial relationship between the radar and the bird target. In the visualization, the radar position is indicated by a yellow-red marker, while the bird’s trajectory is represented by a dotted cyan line. The plot uses a Cartesian coordinate system with X, Y, and Z axes, allowing users to observe the relative positions and movement patterns in three-dimensional space. This visualization helps users intuitively understand the geometric relationship between the radar and the target during the simulation.



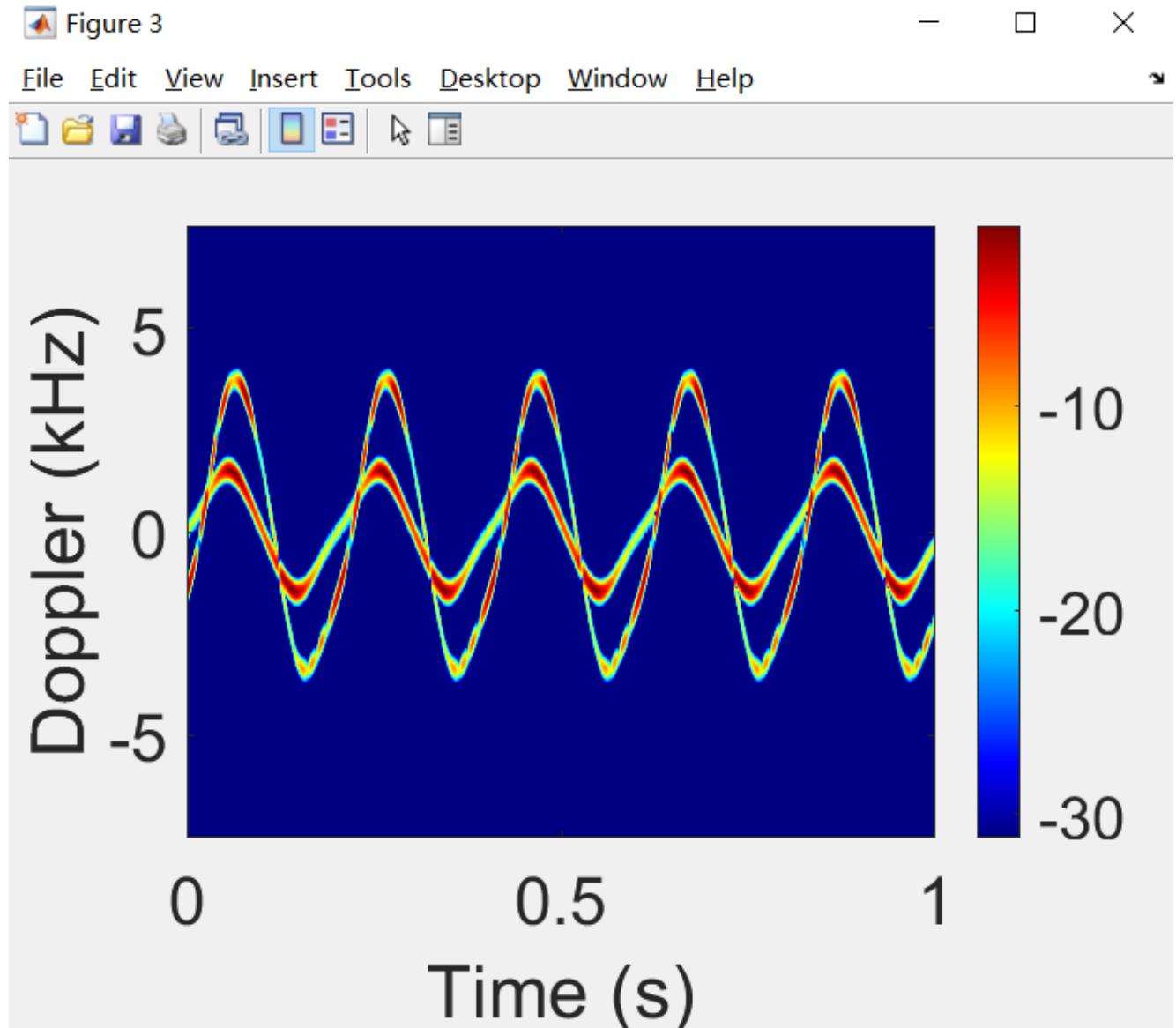
A.1.3 Range Time Intensity

The “Range Profiles of Bird Flapping Wings” function displays the range-time representation of the simulated bird radar returns. The visualization presents range on the vertical axis (in meters) and pulse number on the horizontal axis, with signal intensity depicted using a color map ranging from dark blue (lowest) through cyan to red (highest). The periodic pattern visible in the range profile corresponds to the bird’s wing flapping motion, with the sinusoidal-like variation in range representing the alternating forward and backward movement of the wings during flight.



A.1.4 Micro-Doppler Spectrogram

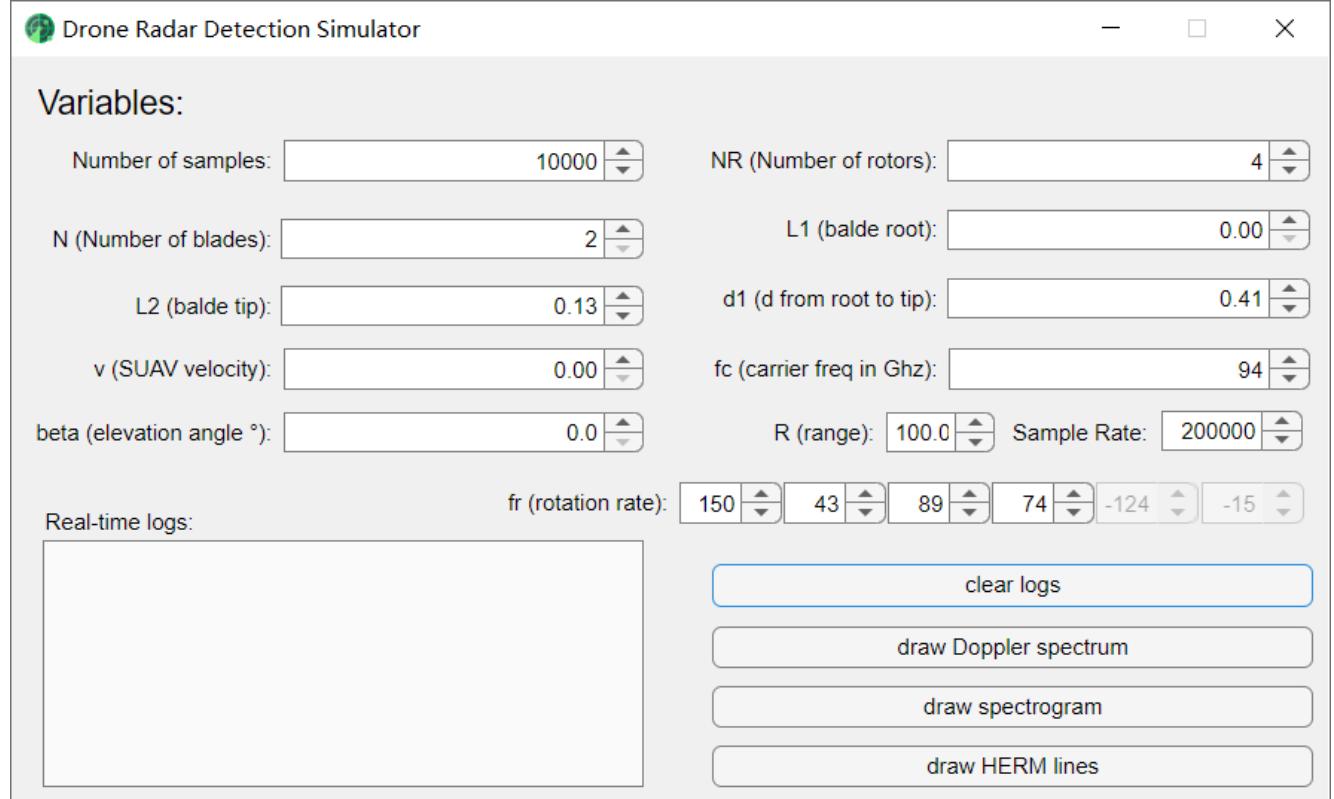
The “Micro-Doppler Signature” function generates a time-frequency representation of the simulated bird radar returns. The plot displays Doppler frequency (in kHz) on the vertical axis and time (in seconds) on the horizontal axis, with signal intensity represented by a color scale from dark blue (-30 dB) to red (10 dB). The periodic sinusoidal pattern in the micro-Doppler signature corresponds to the bird’s wing-beat motion, where the multiple harmonics visible at different Doppler frequencies represent the complex motion dynamics of the wings during the flapping cycle.



A.2. GUI for Drone Simulation

A.2.1 Main Interface

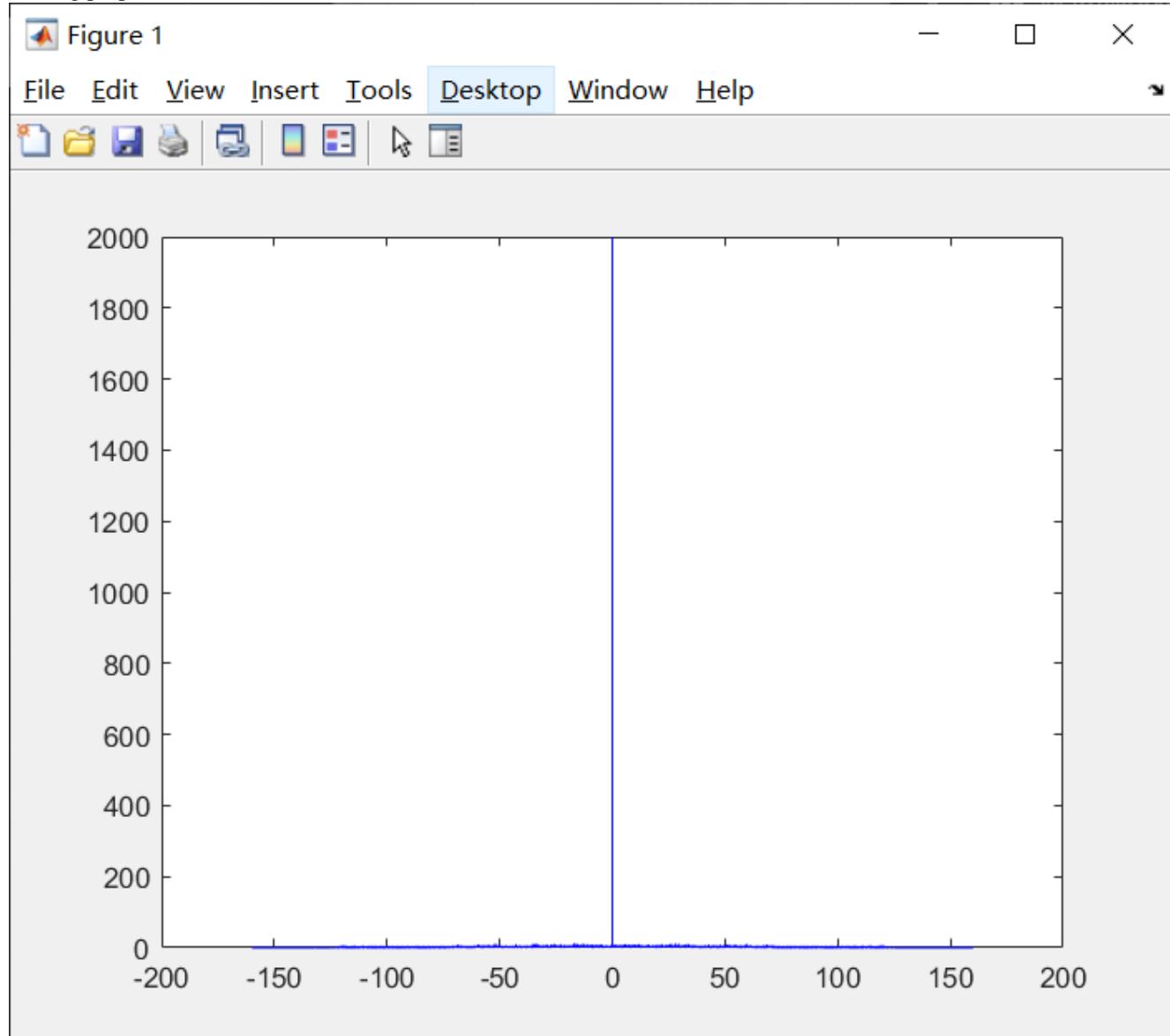
For drone radar signature simulation, we developed a dedicated GUI that enables comprehensive parameter configuration, as shown in Figure:



The interface allows users to specify both drone physical characteristics (number of rotors, number of blades per rotor, blade dimensions L_1 and L_2 , distance from root to tip d_1) and operational parameters (SUAV velocity, elevation angle, rotation rates for each rotor). Radar parameters can also be adjusted, including carrier frequency (f_c), range (R), and sample rate. The GUI provides three visualization functions: Doppler spectrum analysis, spectrogram generation, and HERM line visualization. A real-time log window monitors the simulation progress and status.

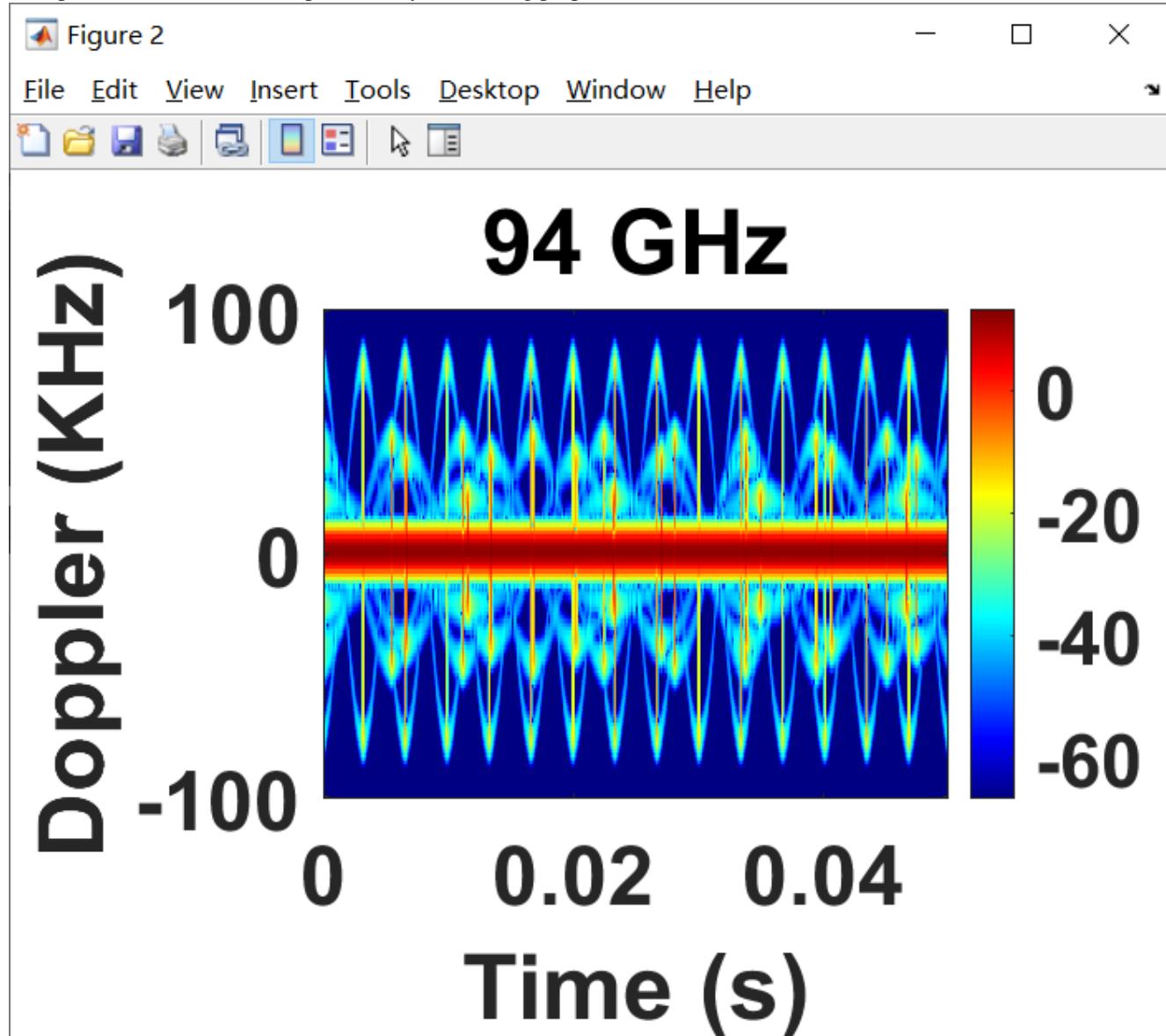
A.2.2 Spectrum

The “Doppler Spectrum” function displays the frequency-domain representation of the simulated drone radar returns. The plot shows frequency on the horizontal axis (ranging from -200 to 200) and signal amplitude on the vertical axis (0 to 2000). The spectrum reveals characteristic peaks corresponding to the drone’s propeller rotation rates, with the central peak representing the drone’s body reflection and symmetric sidebands indicating the micro-Doppler modulations induced by the rotating propellers.



A.2.3 Spectrogram

The “Spectrogram” function generates a time-frequency visualization of the drone’s micro-Doppler signature at 94 GHz. The plot presents Doppler frequency (in kHz) on the vertical axis ranging from -100 to 100 kHz, and time (in seconds) on the horizontal axis. Signal intensity is represented by a color scale from dark blue (-60 dB) to red (0 dB). The spectrogram clearly shows the characteristic features of a hovering drone: a strong central line representing the drone’s body, and symmetric blade flash patterns above and below, produced by the rotating propeller blades.



A.2.4 HERM Lines

The “HERM Lines” visualization displays the Helicopter Rotor Modulation (HERM) pattern of the drone. The plot shows Doppler frequency (in Hz) on the vertical axis ranging from -1×10^5 to 1×10^5 Hz, and time (in seconds) on the horizontal axis. Signal strength is indicated by a color scale from dark blue (20 dB) to red (65 dB). The visualization reveals characteristic HERM lines — horizontal traces at specific Doppler frequencies, which are harmonically related to the product of the number of blades and the rotation frequency of the propellers.

