R. B. ÇORUK

ATILIM UNIVERSITY 2025

ISAR IMAGING OF DRONE SWARMS USING MMWAVE RADAR

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

REMZİYE BÜŞRA ÇORUK

A DOCTOR OF PHILOSOPHY THESIS
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2025

ISAR IMAGING OF DRONE SWARMS USING MMWAVE RADAR

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

REMZİYE BÜŞRA ÇORUK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2025

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

Prof. Dr. Ender Keskinkılıç
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Atilim University**.

Prof. Dr. Reşat Özgür Doruk
Head of Department

This is to certify that we have read the thesis ISAR IMAGING OF DRONE SWARMS USING MMWAVE RADAR submitted by REMZİYE BÜŞRA ÇORUK and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Ali Kara
Co-Supervisor

Prof. Dr. Elif Aydın
Supervisor

Examining Committee Members:

Prof. Dr. Reşat Özgür Doruk
Electrical Electronics Eng. Dept., Atılım Uni.

Prof. Dr. Elif Aydın
Electrical Electronics Eng. Dept., Çankaya Uni.

Prof. Dr. Nursel Akçam
Electrical Electronics Eng. Dept., Gazi Uni.

Prof. Dr. Asım Egemen Yılmaz
Electrical Electronics Eng. Dept., Ankara Uni.

Asst. Prof. Dr. Barış Gürcan HAKANOĞLU
Electrical Electronics Eng. Dept., Atılım Uni.

Date: January 22, 2025

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : REMZİYE BÜŞRA ÇORUK

Signature :

ABSTRACT

ISAR IMAGING OF DRONE SWARMS USING MMWAVE RADAR

Çoruk, Remziye Büşra

Ph.D., Department of Electrical and Electronics Engineering

Supervisor : Prof. Dr. Elif Aydın

Co-Supervisor : Prof. Dr. Ali Kara

January 2025, 56 pages

With the increasing use of drones and drone swarms in technology, the importance of anti-drone technologies has grown significantly. However, detecting drones and drone swarms in scenarios with limited fields of view remains a persistent challenge in the literature. This thesis focuses on the classification of Inverse Synthetic Aperture Radar (ISAR) images of drone swarms reconstructed at millimeter-wave (mmWave) frequency bands based on their formation, size, and payload configurations. ISAR images of drone swarms were generated using the ANSYS High-Frequency Simulation Software (HFSS) electromagnetic simulation software. Swarm structures were modeled using quadcopter drones, and the formation types were designed with basic geometric shapes, such as line, cross, square, and triangle. The drones in the swarms were categorized into three sizes: medium, small, and mini. Additionally, payload drones were included in the swarm configurations. Radar and simulation parameters were optimized to obtain high-resolution ISAR images. To expand the dataset, ISAR images were reconstructed at various look angles (from 0° to 350° in 10° increments). The determination of swarm formation types using ISAR images was carried out through a Convolutional Neural Network (CNN) in the image recognition phase. The detection of drone size and payload configurations was conducted using the You Only Look Once (YOLO) algorithm in the object detection phase. The results

achieved in this thesis are highly promising. By presenting an extended dataset and detection algorithm, this work contributes significantly to the literature and advances the field of drone swarm detection.

Keywords: Radar imaging, millimeter-wave radar, inverse synthetic aperture radar, drone swarm, deep learning

ÖZ

MMDALGA RADAR KULLANARAK DRONE SÜRÜLERİNİN ISAR GÖRÜNTÜLENMESİ

Çoruk, Remziye Büşra

Doktora, Elektrik ve Elektronik Mühendisliği

Tez Yöneticisi : Prof. Dr. Elif Aydın

Ortak Tez Yöneticisi : Prof. Dr. Ali Kara

Ocak 2025, 56 sayfa

Teknolojide drone ve drone sürülerinin kullanımının artmasıyla birlikte, anti-drone teknolojilerinin kullanımı önemli ölçüde artmıştır. Ancak, sınırlı görüş alanına sahip senaryolarda drone ve drone sürülerinin tespiti literatürde kalıcı bir zorluk olmaya devam etmektedir. Bu tez, milimetre dalga (mmDalga) frekans bantlarında yeniden oluşturulan drone sürülerinin Ters Sentetik Açıklıklı Radar (ISAR) görüntülerinin, oluşumlarına, boyutlarına ve yük yapılandırmalarına göre sınıflandırılmasına odaklanmaktadır. Drone sürülerinin ISAR görüntüleri, ANSYS Yüksek Frekanslı Yapısal Simülatör (HFSS) elektromanyetik simülasyon yazılımı kullanılarak üretilmiştir. Sürü yapıları, quadcopter dronlar kullanılarak modellenmiş ve oluşum tipleri, çizgi, çarpı, kare ve üçgen gibi temel geometrik şekillerle tasarlanmıştır. Sürülerdeki dronlar, orta, küçük ve mini olmak üzere üç boyutta kategorize edilmiştir. Ek olarak, yük dronları sürü yapılandırmalarına dahil edilmiştir. Yüksek çözünürlüklü ISAR görüntüleri elde etmek için radar ve simülasyon parametreleri optimize edilmiştir. Veri setini genişletmek için, ISAR görüntüleri çeşitli bakış açılarında (0° ile 350° arasında 10° artışlarla) oluşturulmuştur. ISAR görüntüleri kullanılarak sürü oluşumu tiplerinin belirlenmesi, görüntü tanıma aşamasında bir Evrişimsel Sinir Ağı (CNN) aracılığıyla

gerçekleştirildi. Bunu takiben, nesne algılama aşamasında Sadece Bir Kez Bak (YOLO) algoritması kullanılarak drone boyutu ve yük tespiti gerçekleştirildi. Bu tezde elde edilen sonuçlar oldukça ümit vericidir. Genişletilmiş bir veri seti ve tespit algoritması sunarak, bu çalışma literatüre önemli katkıda bulunmaktadır.

Anahtar Kelimeler: Radar görüntüleme, milimetre dalga radarı, ters sentetik açıklıklı radar, drone sürüsü, derin öğrenme

To my little prince, Bulut, who lifted me above the clouds.

ACKNOWLEDGMENTS

In this challenging yet invaluable journey, I would like to express my heartfelt gratitude to my advisors, Prof. Dr. Elif AYDIN and my co-advisor, Prof. Dr. Ali KARA, for their support, knowledge, and dedication.

Prof. Dr. Elif AYDIN has not only been a scientific mentor throughout my academic journey but also a guiding light who has always encouraged me, boosted my motivation, and reinforced my confidence. Her dedicated approach, attention to detail, and consistently encouraging attitude have made this process significantly more productive and meaningful for me.

I would also like to extend my deepest gratitude to Prof. Dr. Ali KARA. His scientific rigor, disciplined approach, and constructive feedback have strengthened and enhanced my dissertation process. The academic ethics, discipline, and analytical thinking skills I have gained from him will serve as a great guide for my future studies. I am truly grateful for his generosity in sharing his knowledge and experience, as well as for his patience and dedication.

Furthermore, I would like to express my sincere appreciation to the esteemed jury members who contributed significantly to my academic development. I am deeply thankful to Prof. Dr. Asım Egemen YILMAZ and Prof. Dr. Nursel AKÇAM for their constructive feedback and valuable suggestions during all my thesis monitoring committee meetings. I also extend my gratitude to Prof. Dr. Reşat Özgür DORUK and Asst. Prof. Dr. Barış Gürcan HAKANOĞLU for their valuable evaluations and contributions during my thesis defense. Their support and guidance have played a crucial role in building a stronger scientific foundation for my dissertation.

The greatest supporters throughout this journey have been my beloved family, who stood by me with their patience and love, giving me strength. My dear husband, Samet, my life partner, who has always made me feel secure, supported me with

endless understanding, and kept me motivated even in the most challenging moments of this process... You were the reason I stood up again and again... My precious son, Bulut, who always puts a smile on my face and makes me feel his love in every moment... Completing this journey would have been much more difficult without you both. I am deeply grateful for all your sacrifices, love, and understanding.

Additionally, I would like to extend my sincerest thanks to both my husband's and my own families for their unconditional support, understanding, and love throughout this process. Their presence has always been a source of strength and encouragement for me. Their patience, uplifting words, and moral support have helped me navigate this journey more smoothly.

I am also deeply grateful to my dear friend, Bengisu Yalçınkaya, who has been a companion in my academic journey, always supporting me and keeping me motivated.

To my valued friends from SDT, Merve, Yarkın, Emine, Irmak, and Faruk, you have always been a source of joy and motivation for me with your kind words and encouragement. Your support and friendship have made this journey more enjoyable.

Lastly, I extend my gratitude to everyone who has been by my side, provided support, and in some way touched my life throughout this process.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Drones are defined as Unmanned Aerial Vehicles (UAVs), and drones may be operated remotely and automatically. They have been widely utilized in both civil and military applications in recent years [1]. Due to technical advancements, drones are becoming more and more common, which increases the risk of mishaps in airspace. And, because drones are easily created using inexpensive Commercial off the Shelf (COTS) development kits and can be readily purchased online, they may be a threat to privacy and security [2]. Furthermore, drones cooperating to accomplish a particular objective have rapidly gained importance. This multi-drone configuration, known as a drone swarm, offers numerous benefits in the military and defense technology [3]. In light of the increasing usage of drone swarms in contemporary technology, creating new technology for anti-drone applications has gained popularity [4]. The authors of [5] present a You Only Look Once (YOLO)-based lightweight UAV swarm detection technique. The low detection accuracy and significant computer resource consumption issues are intended to be resolved by this approach. An Intrusion Detection System (IDS) based on Timed Probabilistic Automata (TPA) is introduced in another paper [6] to simulate drone swarm behavior and detect variations that would indicate an incursion. The authors of [7] outline a machine learning-based technique for drone swarm flight. Real flight data is used to test the suggested approach. An approach for detecting and tracking UAV swarms utilizing videos and an annotated dataset called UAV Swarm is provided in [8]. For this, YOLOX and faster Region based Convolutional Neural Networks (R-CNN) are employed. Regarding detection in anti-drone swarm defense technologies that have only begun to surface in the literature, there are still issues that need to be resolved in this context [9].

One potential technique for detecting drones or drone swarms is radar technology, particularly when the field of view is constrained [10]. The two most popular radar techniques have been studied in the literature. The first method uses frequency measurement to determine the drones' micro-doppler [11]. The second method, based on range (time) measurement [12], images a drone using Synthetic Aperture Radar (SAR) and Inverse-SAR (ISAR). SAR and ISAR images of the target are obtained by using the movements of the radar and the target [13]. The target stays stationary while the radar travels or rotates when employing the SAR technique. In this case, the synthetic antenna aperture is provided by the motion of the radar [14]. Because the target moves while the radar beam aimed at it stays constant, reflectivity data resulting from the different reactions of the target's scatters is obtained. Signal processing techniques are used to reconstruct SAR images [15]. On the other hand, ISAR approach, a signal processing technique, can detect moving objects in both down-range and cross-range resolution. Because it may be used to detect moving objects like tanks, ships, drones, and airplanes, it is more commonly used in military and defense applications. SAR/ISAR systems are deployed using both manned and unmanned aerial platforms, such as satellites or aircraft [16]. With the most recent advancements in radar technology, high resolution is a crucial problem in radar imaging. Another common way to attain high resolution is to use a radar system that operates in the millimeter wave (mmWave) frequency ranges, which allows for more efficient target tracking and detection, particularly in inclement weather, day or night [17].

In recent years, ISAR images have become increasingly popular in the literature. The literature contains ISAR image data of a variety of targets, including basic objects, satellites, ships, aircraft, and drones. Several works [18, 19] use ISAR images that are produced by utilizing a turntable to rotate a few basic objects. Also, authors used satellites to obtain ISAR images [20, 21, 22]. The suggested approach makes use of the same satellite dataset, which includes satellites. Four distinct satellite types' ISAR images are acquired, and the classification results are examined using various deep-learning techniques. A technique for obtaining ISAR images of ships in motion at sea is presented in [23]. Additionally, ship images are used for the classification stage. Aircraft are employed as targets in a variety of works [24, 25, 26]. Deep learning techniques are used to classify different kinds of aircrafts. Drones are another target that

radar imaging uses. Drone ISAR images are analyzed in the literature. [27] presents a new method for drone target recognition and tracking applications using wide-angle ISAR images. The outcomes of employing the k-space approach to create ISAR images are displayed. Another study reports the results of experiments in the C and Ku bands using the ultra-wideband ISAR sensor [28]. Two distinct kinds of drones are the targets. The results show that whether or not the plastic wings rotate, there is minimal change in the ISAR image. The authors also point out that even non-metallic components are readily visible in the ISAR image. Furthermore, in an experiment, the authors of [29] examined the fully polarimetric radar for the fixed-wing drone in an anechoic room. The drone is displayed in VV, VH, HV, and HH polarization modes in Radar Cross Section (RCS) and ISAR images. [30] introduces wide-angle and ultra-wideband ISAR imaging for UAVs. The k-space imaging technique is used to create ISAR images. A Bayesian Statistical Machine Learning-based ISAR imaging method for drones is proposed by the authors of [31]. The method is discussed by presenting the results of experimental and simulated experiments. In [32], the ISAR image approach is used to identify drones using Multiple Input - Multiple Output (MIMO) radar. The backpropagation technique is employed to obtain high resolution in ISAR images. Three different drone structures are used in the detection process. In [33], the same authors create drone ISAR images using the time domain correlation approach. Research on the classification of drone ISAR images has lately acquired attention due to the developments in anti-drone technology. The use of ISAR images for drone identification is investigated in [34, 35]. One approach that incorporates drone classification is provided by the study in [34]. There are two types of drones: commercial quadcopters and explosives-equipped quadcopters. Another work [35] presents a multi-band radar system to get high classification displayed.

With the development of technology, the ability to recognize drone swarms, which are increasingly used, is becoming more and more important every day. In the literature, camera images are usually used to recognize drone swarms. Drone detection using ISAR images is quite limited. Moreover, even if drones are detected by generating ISAR images, it continues to be a challenge in the detection of drone swarms in the literature. In addition, no study has detected a dataset with different drone swarm structures. In this study, drone swarms are detected using ISAR images according

to the formation (Line, Square, Triangle, Cross), size (Medium, Small, Mini), and payload (Payload/Unpayload) configurations. Radar and simulation parameters are discussed in order to prepare image data for the detection process and to produce high-resolution images. The data augmentation process provides a solution to the overfitting problem and increases the dataset. The results show that different drone objects are successfully detected with high accuracy. This thesis presents an extended dataset including ISAR images of drone swarms and the detection model to the literature.

The thesis's following sections are organized as follows.

Chapter 2 reviews ISAR imaging theory, including radar imaging principles, image reconstruction, resolution parameters, simulation parameters, and image quality metrics.

Chapter 3 covers computer vision techniques, focusing on image recognition with CNNs and object detection using the YOLO model.

Chapter 4 presents the ISAR dataset, detailing drone swarm scenarios based on formation, size, and payload configurations.

Chapter 5 discusses the results, including image recognition and object detection performance.

Chapter 6 summarizes the thesis findings and suggests future research directions.

CHAPTER 2

ISAR IMAGE THEORY

This section discusses radar imaging based on SAR and ISAR techniques. Also, the theory of ISAR image reconstruction steps, the parameters of image resolution (down-range and cross-range resolution) required to acquire high resolution, and simulation parameters calculated based on radar parameters are detailed.

2.1 Radar Imaging

Radar technology is based on receiving reflected signals from targets by emitting radio waves and creating data. Imaging radar creates images by processing the signals reflected in the radar, which is used to create images of the targets. Reflected waves in different forms will provide details regarding the targets. By processing radar data, a 2D or 3D image of the target can be created. Since radar can obtain images using radio waves, it provides an effective method even in bad weather conditions where visibility is limited. It can also offer penetrating imaging in the presence of obstacles, walls, and water. These are the most important advantages of imaging radar. The most commonly used imaging techniques today are SAR and ISAR technologies.

SAR is used as an imaging technique for targets and is used for situations where the radar is moving and the target remains stationary. Thanks to the synthetic aperture provided by the movement of the radar, a longer aperture is created than the real aperture. On the other hand, ISAR technique is used when the radar is usually stationary and the target is moving. The moving target can move on a rotating or fixed path. SAR and ISAR setups are given in Figure 2.1.

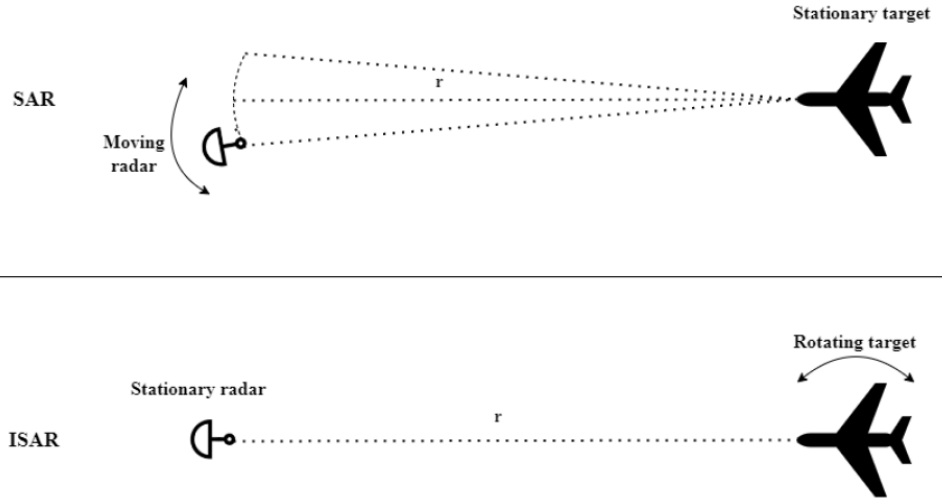

Figure 2.1: Top: SAR setup, Bottom: ISAR setup

2.2 Doppler Effect

A target sends a signal to radar, which then receives an echo signal from the target. Radar can determine the target's range by measuring the time delay of the signal it receives. The frequency of the received signal will differ from the frequency of the transmitted signal if the target is moving. It is called the doppler effect. The moving target's radial velocity, or the velocity component in the direction of the Line of Sight (LOS), determines the Doppler frequency shift. The fourier transform of the received signal is typically used to measure the Doppler frequency shift in the frequency domain. Doppler frequency can be formulated in Equation 2.1.

$$f_D(t) = -f\left[\frac{2v(t)}{c}\right] \tag{2.1}$$

where $v(t)$ is the radial velocity and f is the transmitted carrier frequency. Also, the transmitted signal can be assume in Equation 2.2.

$$S_T(t) = \cos(2\pi f t) \tag{2.2}$$

And, the received signal can be expressed as a model in Equation 2.3.

$$S_R(t) = A\cos\left[2\pi(f + f_D)t\right] = A\cos\left[2\pi ft + \varphi(t)\right] \tag{2.3}$$

where A is the amplitude of the signal, $\varphi(t)$ is the phase shift on the signal. Coherent demodulator is applied with in-phase(I) and quadrature(Q) channels. By mixing of received and transmitted signals, the output of I channel can be obtain in Equation 2.4.

$$S_R(t)S_T(t) = \frac{A}{2}\cos\left[4\pi ft + \varphi(t)\right] + \frac{A}{2}\cos\varphi(t) \tag{2.4}$$

And, after appliying low-pass filter, the output of I channel rewritten in Equation 2.5.

$$I(t) = \frac{A}{2}\cos\varphi(t) \tag{2.5}$$

90° phase shifted transmitted signal can be expressed in Equation 2.6.

$$S_T^{90^\circ}(t) = \sin(2\pi ft) \tag{2.6}$$

Mixed signal with the transmitted and received signal is written in Equation 2.7.

$$S_R(t)S_T^{90^\circ}(t) = \frac{A}{2}\sin\left(4\pi ft + \varphi(t)\right) - \frac{A}{2}\sin\varphi(t) \tag{2.7}$$

And, after appliying low-pass filter, the output of Q channel rewritten in Equation 2.8.

$$Q(t) = \frac{A}{2}\sin\varphi(t) \tag{2.8}$$

Complex signal with I and Q outputs is obtain in Equation 2.9.

$$S_D(t) = I(t) + jQ(t) = \frac{A}{2}e^{-j\varphi(t)} = \frac{A}{2}e^{-j2\pi f_D t} \tag{2.9}$$

Doppler frequency and radial velocity are obtain in Equation 2.10 and Equation 2.11, respectively.

$$f_D(t) = \frac{1}{2\pi}\frac{d\varphi(t)}{dt} \tag{2.10}$$

$$v = \frac{\lambda}{2}f_D = \frac{c}{2f}f_D. \tag{2.11}$$

where λ is the wavelength of the transmitted signal.

2.3 ISAR Image Reconstruction

The relative rotational movement between the radar and the target creates the ISAR image. This rotation allows the target to be seen on the radar side from different aspect angles and is defined in local system (xy) coordinates as seen in Figure 2.2.

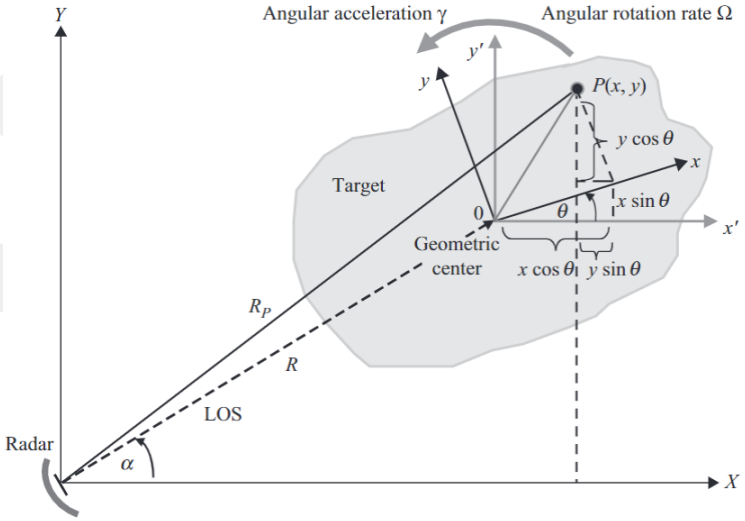

Figure 2.2: Radar and target geometry

The origin of the global system (XY) is stated as the radar place, and the geometric center is stated as the target's origin. It is considered that the target turns with an angular rotation rate along the z-axis of the local system. The angular rotation rate is formulated in Equation 2.12.

$$\Omega(t) = \Omega + \gamma t \tag{2.12}$$

where γ is the angular acceleration and Ω is the angular velocity induced rotation rate. Additionally, target rotation is expressed using the reference coordinate system (x', y'). The origin of this system is at the xy origin and is located parallel to the XY system. The distance between the radar and the scatterer $P(x, y)$ is denoted by $R_p(t)$, whereas R is the distance between the radar and the geometric center and $R_p(t)$ is formulated in Equation 2.13.

$$R_p(t) \cong R + xcos[\theta(t) - a] - ysin[\theta(t) - a] \tag{2.13}$$

and by Taylor expansion

$$\theta(t) = \theta_0 + \Omega t + \frac{1}{2}\gamma t^2 \tag{2.14}$$

where α is defined as the azimuth angle of the target and θ_0 is defined as the initial rotation angle in the reference system. Radar signal backscattered by the scatterer $(P(x, y))$ is formulated in Equation 2.15.

$$S_p(t) = \rho(x_p, y_p)exp[-j2\pi f[2R_p(t)/c]] \tag{2.15}$$

where $\rho(x_p, y_p)$ is defined as the reflectivity density function. The signals from all of the target's scatterers can be combined to create the signal that is sent back to the radar from a single point scatterer, as seen in Equation 2.16.

$$S_R(t) = \iint_{-\infty}^{\infty} \rho(x_p, y_p)exp[-j\frac{4\pi f}{c}R_p(t)]dxdy \tag{2.16}$$

Equation 2.17 is used to formulate the range of the point scatterer at (x, y) when a target is moving both translationally and rotationally at the same time and the azimuthal

angle an is equal to zero.

$$R_p(t) = R(t) + xcos\theta(t) - ysin\theta(t) \tag{2.17}$$

Thus, the signal returned to the radar from a single point scatterer of the target is formulated in terms of frequency and wavenumber. In terms of the frequency, it can be formulated in Equation 2.18.

$$S_R(t) = exp[-j4\pi f\frac{R(t)}{c}] \iint_{-\infty}^{\infty} \rho(x_p, y_p) exp\{-j2\pi[xf_x(t) - yf_y(t)]\}dxdy \tag{2.18}$$

where $f_x(t)$ and $f_y(t)$ are the components of frequency. $f_x(t)$ and $f_y(t)$ are formulated in Equation 2.19 and Equation 2.20, respectively.

$$f_x(t) = 2fcos\theta(t)/c \tag{2.19}$$

and

$$f_y(t) = 2fsin\theta(t)/c \tag{2.20}$$

Wavenumber is formulated in terms of frequency in Equation 2.21.

$$k = 2\pi f/c \tag{2.21}$$

The radar returned signal can be formulated in terms of wavelength in Equation 2.22.

$$S_R(t) = exp[-j4\pi f\frac{R(t)}{c}] \iint_{-\infty}^{\infty} \rho(x_p, y_p) exp\{-j2\pi[xk_x(t) - yk_y(t)]\}dxdy \tag{2.22}$$

where $k_x(t)$ and $k_y(t)$ are the components of wavenumber. Component for x and component for y are formulated in Equation 2.23 and Equation 2.24, respectively.

$$k_x(t) = 2fcos\theta(t)/c = kcos\theta(t) \tag{2.23}$$

and

$$k_y(t) = 2f sin\theta(t)/c = ksin\theta(t) \tag{2.24}$$

Consequently, by taking the Inverse Fourier Transform (IFT) of the $S_R(t)$, the reflectivity density function, $p(x, y)$, of the target is formulated in Equation 2.25.

$$\rho(x_p, y_p) = IFT\{S_R(t)exp[j4\pi fR(t)/c]\} \tag{2.25}$$

Also, the returned signal can expressed in range and doppler domain in Equation 2.26.

$$S_R(t, f) = \iint_{-\infty}^{\infty} \rho(\tau, f_D) \exp\{-j2\pi[ft - f_D\tau]\}\, d\tau df_D \tag{2.26}$$

Equation 2.27 and Equation 2.28 are used for the transformation from the x and y domains to the range-doppler domain, respectively.

$$\tau = \frac{2x}{c} \tag{2.27}$$

$$f_D = \frac{2f\Omega y}{c} \tag{2.28}$$

Consequently, the target reflectivity function can rewritten in the range-doppler domain in Equation 2.29.

$$\rho(\tau, f_D) = \text{IFT}\{S_R(t, f)\} \tag{2.29}$$

Transformation from the x and y domains to the range-doppler domain is shown in Figure 2.3.

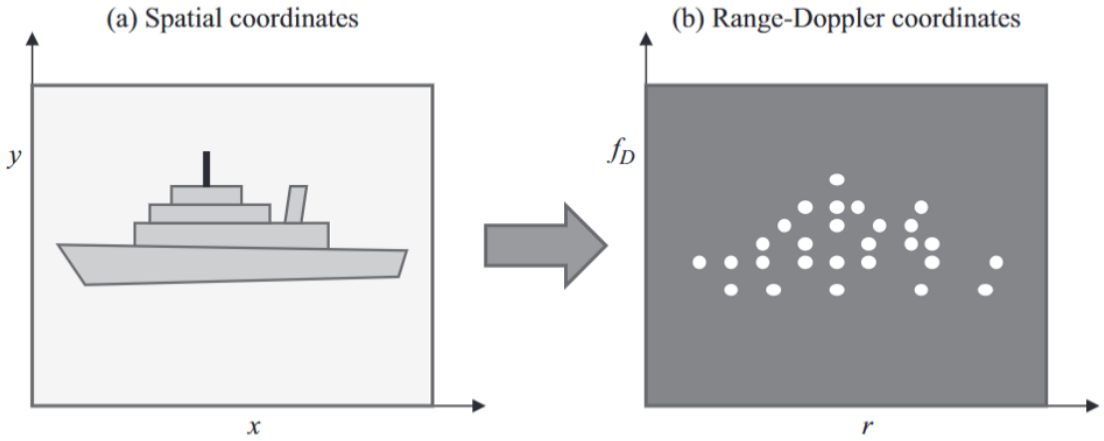

Figure 2.3: Transformation from spatial coordinates to range-doppler coordinates

Target reflectivity function also express as range-doppler image. It can be examined in Figure 2.4. A series of N pulses are released by the radar. The total number of range cells, M, for each transmitted pulse is determined using the size of the range cell and the maximum range covered. However, the total quantity of pulses, N, determines the doppler resolution. The baseband complex I and Q data are arranged into a $M \times N$ two-dimensional complex matrix in the radar receiver following pulse compression and $I - Q$ down-conversion. The range profile shows the received signal's energy distribution as a function of range. Range profile $G(r_m, t_n)$ is obtained using range and doppler tracking procedures. The range-doppler image is reconstructed after applying the Fast Fourier Transform (FFT) to the range profile at each range cell, M. The expression for a range-doppler image of size $M \times N$ is given in Equation 2.30.

$$I(r_m, f_n) = \mathrm{FFT}_n\{G(r_m, t_n)\} \tag{2.30}$$

where $n = 1, 2, \ldots, N$ and $m = 1, 2, \ldots, M$ [36].

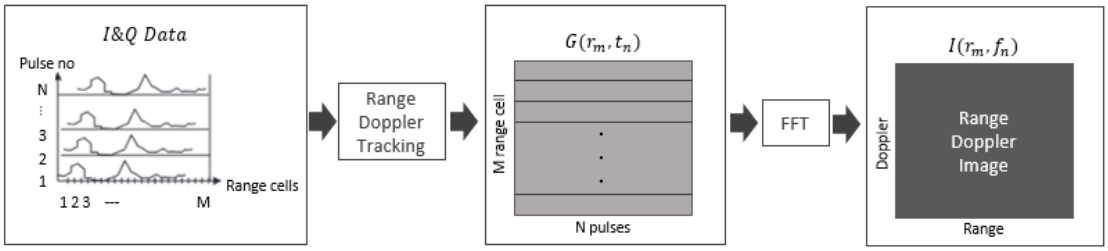

Figure 2.4: Range-doppler image reconstruction

2.4 Image Resolution

To obtain ISAR images obtained as high-resolution, image resolution parameters must be adjusted well. Image resolution parameters can be defined as resolving separated scatters. Image resolution parameters are calculated in two dimensions: down-range and cross-range [37].

2.4.1 Down-range resolution

Down-range resolution is the component in the radar direction that takes into account the synthetic aperture made by the signals from the radar on the target. It ensures that two targets placed in the direction extension of the radar signal are detected and displayed as two different targets on the radar. It is formulated in Equation 2.31.

$$\Delta x = \frac{c}{2B} \tag{2.31}$$

where B can be defined as the bandwidth of the signal.

2.4.2 Cross-range resolution

Cross-range resolution is the component parallel to the radar's flight line that takes into account the synthetic aperture made by the signals from the radar on the target.

It allows two targets positioned at the same distance from the radar to be detected as two separate targets and displayed on the radar. It is formulated in Equation 2.32.

$$\Delta y = \frac{c}{2f\Omega T} \tag{2.32}$$

where T is the observation time and Ω is the angular velocity induced rotation rate.

2.5 Simulation Parameters

ISAR images are reconstructed in two axes (down-range and cross-range). Thus, the appropriate choice of image resolution parameters affects image quality. Moreover, image resolution parameters are significant for using the calculation of simulation parameters. ISAR images are reconstructed using ANSYS. ANSYS is simulation software generally used in engineering applications. High-Frequency Structural Simulator (HFSS) is described as an asymptotic electromagnetic simulator. It is usually used at high frequencies. Additionally, HFSS simulator includes a Shooting Bouncing Ray (SBR+) solver for high-speed computation. SBR+ has hybrid structures, which are Geometric Optics (GO) and Physical Optics (PO) [38]. PO is extended to numerous bounces by HFSS SBR+ using GO.

Simulation parameters are calculated to acquire high-resolution ISAR images using radar parameters. Simulation parameters include frequency step, angle resolution, bandwidth, and aspect angle. Frequency step and angle resolution formulas are given in Equation 2.33 and Equation 2.34, respectively.

$$\Delta f = \frac{c}{2 * x_{max}} \tag{2.33}$$

$$\Delta \varphi = \frac{\lambda}{2 * y_{max}} \tag{2.34}$$

where x_{max} is the down-range extents, y_max is the cross-range extents, and λ is the wavelength. Later on, to calculate bandwidth and aspect angle parameters, frequency

step and angle resolution values are used. The frequency step and angle resolution formulas are given in Equation 2.35 and Equation 2.36, respectively.

$$B = \Delta f * N_x \tag{2.35}$$

$$\Delta\phi = \Delta\varphi * N_y \tag{2.36}$$

where N_x is the number of samples in x axis which is described as $x_{max}/\Delta x$ and N_y is the number of samples in y axis which is described as $y_{max}/\Delta y$.

2.6 Image Quality Metrics

The quality of inputs like images and videos is estimated using image quality metrics. In addition, it is used to measure the similarity between two images. Examples of these metrics include Mean-Square-Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and Image Entropy (IE) [39, 40, 41]. The reconstructed image can be described as $I = (I_{xy})_{X\times Y}$ and the reference image can be described as $I' = (I'_{xy})_{X\times Y}$. MSE and PSNR metrics are formulated in Equation 2.37 and Equation 2.38, respectively.

$$\text{MSE} = \frac{1}{XY}\sum_{x=1}^{X}\sum_{y=1}^{Y}(I_{xy} - I'_{xy})^2 \tag{2.37}$$

$$\text{PSNR} = 10\log_{10}\frac{\text{MAX}^2}{\text{MSE}} \tag{2.38}$$

where MAX is the value of the maximum pixel of I'. SSIM metric is formulated in Equation 2.39.

$$\text{SSIM} = \frac{(2\mu_I\mu_{I'} + c_1)(2\sigma_{II'} + c_2)}{(\mu_I^2 + \mu_{I'}^2 + c_1)(\sigma_I^2 + \sigma_{I'}^2 + c_2)} \tag{2.39}$$

where σ_I^2 and $\sigma_{I'}^2$ are the variance of I and I', $\sigma_{II'}$ is the covariance of I and I', and μ_I and μ_I' are the mean value of I and I'. IE metric is formulated in Equation 2.40.

$$IE = \sum_{x=1}^{X} \sum_{y=1}^{Y} \frac{I_{xy}^2}{\text{sum}_I} \ln \frac{I_{xy}^2}{\text{sum}_I} \tag{2.40}$$

where sum_I means $\sum_{x=1}^{X} \sum_{y=1}^{Y} I_{xy}^2$.

CHAPTER 3

COMPUTER VISION

Artificial Intelligence (AI) can be defined as the ability to perform human-like tasks with computer programs. A subfield of AI called computer vision enables computers to comprehend, analyze, and decide based on visual information, such as digital images or movies [42]. It seeks to create algorithms and software that mimic the human visual system to get meaning from the visual environment. Computer vision is divided into image recognition and object detection methods. Image recognition method is the process of classifying images. Object detection method detects, labels, and localizes objects in the images [43].

3.1 Image Recognition

A computer vision technology that allows a computer to identify and categorize objects, scenes, faces, or certain patterns in a digital image is known as image recognition. The goal is to obtain meaningful results by analyzing visual data. It strives to mimic the ability of the human brain to recognize an object or scene [44]. Image recognition methods are traditional image processing method, machine learning-based method and deep learning-based method.

Traditional image processing method includes Edge Detection and Histogram-Based Methods. Edge detection is the process of identifying an object's edges using algorithms like Canny and Sobel. Histogram-Based Methods are done on color intensities and distribution.

Machine learning-based method includes Support Vector Machines (SVM) and K-

Nearest Neighbors (KNN). SVN is used to categorize image features. In KNN, images with comparable attributes can be categorized.

Deep learning-based method includes CNN, Transfer learning and Autoencoder. It classifies items by learning local properties in images. The most used deep learning algorithm for visual data analysis is CNN. It learns local features in images and classifies objects. Example models: AlexNet, VGGNet, ResNet. Transfer Learning is retraining previously trained models on fresh images is known as transfer learning. To find the basic patterns and structures in literature, autoencoders are utilized.

3.1.1 Convolutional Neural Network

CNNs are deep learning models that analyze an image provided as input. When it was first released, it was tested in fields like biomedical, audio, image, and natural language processing. However, it has been determined that the best results are obtained from image inputs. It allows the identification of the image or objects in the image [45]. CNN consists of convolution operations comprised of many layers. The filters used for the convolution operation play an important role in determining the characteristics of the inputs [46]. The general CNN structure is composed of convolutional, pooling (downsampling), flattening, and fully-connected layers as seen in Figure 3.1.

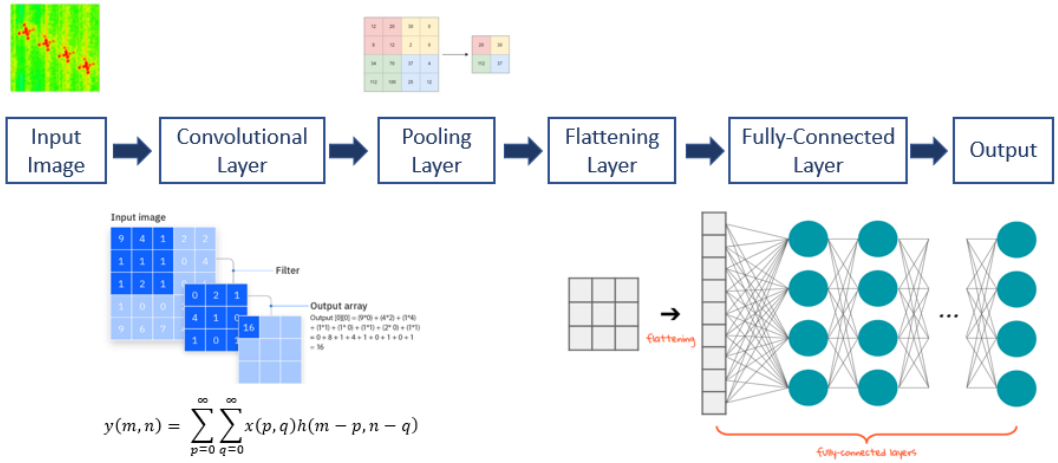

Figure 3.1: General structure of CNN model

3.1.1.1 Convolutional Layer

The convolution layer is the most important step of the CNN model. The features of the input image are determined in this part. A filter (kernel) is applied to the image. After these processes, the feature map of the CNN model is formed. The convolutional processing for image input is given in Equation 3.1.

$$y(m,n) = \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} x(p,q)h(m-p,n-q) \tag{3.1}$$

where $y(m,n)$ states the output image, $h(m,n)$ is the impulse response, and $x(p,q)$ can be defined as the input image [47].

3.1.1.2 Pooling (Downsampling) Layer

The pooling layer is located between convolution layers and is used to reduce computational complexity. There are average, L2-norm, and max pooling types, but the most commonly used is the max pooling layer. The general logic in the max pooling method is to progress by taking the largest number in the area covered by the filter while processing the image.

3.1.1.3 Activation Functions

Activation functions are very important in deep learning algorithms to train the model using complex data. Activation functions adjust weight and bias values in neural networks to ensure the model learns correctly. There are varieties such as sigmoid, tanh, ReLU, and leaky ReLU. The most commonly used of these activation functions is ReLU. In ReLU, the function output is equal to zero for negative values of z. In the case of zero and positive z, the function has the same amplitude value. ReLU formula is given in Equation 3.2.

$$a(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases} \tag{3.2}$$

Another activation function, softmax, is mostly used in the last part of the model, and the probability result for each class is determined here. The softmax formula is given in Equation 3.3 [48].

$$a(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.3}$$

3.1.1.4 Flattening and Fully-Connected layer

After the convolution and pooling layers, the data is reduced to a single dimension with the flattening layer, and then the classification algorithm results are revealed with the fully connected layer.

3.2 Object Detection

Object detection, which is one of the computer vision methods, may locate and identify objects in a digital image or video. As a result, it generates more intricate and detailed findings than image recognition. The field of object detection, which has gained a lot of popularity recently and encompasses a variety of techniques, is still evolving. Object detection is divided into two main subgroups: traditional and deep learning-based methods.

Traditional method often blend image processing and machine learning methodologies. The methods include Haar Cascades, Selective Search, and Histogram of Oriented Gradients (HOG) + SVM. The Haar Cascades technique employs classifiers to detect objects and extract certain properties. Face recognition and other basic object identification tasks are used. SVM is used to classify the image's edge and texture features after HOG + SVM removes them. Selective Search divides the image into

several stages of segmentation to find possible object regions. Although these techniques are comparatively quick, their performance is limited in situations involving multiple objects or complicated scenes.

Deep learning-based method is used for object detection problems. The primary instruments in this discipline are CNNs. It is separated into two categories: Single Shot MultiBox Detector (SSD) and Region-Based Approaches. Various iterations of Region-based (R-CNN) are part of region-based approaches. R-CNN uses CNN to classify each region after extracting features from various areas of the image. A quicker variant of R-CNN is called Fast R-CNN. All regions are processed at once after the feature map has been extracted. Faster R-CNN employs a network that produces region suggestions in addition to Fast R-CNN. This makes object detection quicker and more precise. SSD does object detection directly, without the region proposal phase. There are three methods: RetinaNet, SSD, and YOLO. YOLO creates a grid out of the image and concurrently predicts the locations and item classes of each cell. In real-time applications, it is widely used. SSD detects things at various scales by using several feature maps. On unbalanced datasets, RetinaNet performs better. It uses a loss function known as targeted loss and concentrates on learning challenging objects.

The YOLO family of models has also had a big influence on the field of computer vision. Numerous variations have been created since Joseph Redmon first introduced it in 2014, and it is widely used in technology. Since the speed component is crucial, it is preferred in real-time applications over the YOLO CNN object detection group since it only runs the image through the model once before estimating the item [49].

3.2.1 YOLO Model

Figure 3.2 represents the general architecture of the YOLO object detection model, which follows a streamlined pipeline for efficient and accurate detection. The process begins with an input image that undergoes preprocessing, such as resizing and normalization, to ensure compatibility with the model. The backbone is a deep convolutional neural network responsible for extracting essential spatial and semantic features from the image. These features are then passed to the neck, which enhances

multi-scale feature representation. This step improves the model's ability to detect objects of varying sizes. The head processes the refined feature maps and predicts object categories, bounding box coordinates, and confidence scores. YOLO employs an anchor-based approach with grid-based predictions, ensuring real-time performance with high accuracy. Finally, post-processing techniques refine the predictions by eliminating redundant detections. This modular architecture allows YOLO to achieve fast and accurate object detection, making it widely used in real-world applications.

Figure 3.2: General structure of YOLO model

Figure 3.3 displays the YOLO flow diagram. First, the YOLO model receives the input image. YOLO divides the image into $N \times N$ grids, each of which assesses whether an object is present and, if so, whether the center point is within its region. Bounding boxes are drawn around each region's objects. The letters (x, y, w, h) stand for the center coordinates, width, height of each bounding box. Object probabilities, P_r(Object), and Intersection over Union (IoU) are multiplied to determine confidence ratings, C_i. The chance that an object will exist inside the box is denoted by P_r(Object), which has values between 0 and 1. The accuracy of the box, or IoU, shows how much the true and predicted bounding boxes overlap. The object detection task is then carried out using the vector $[P(c), B_x, B_y, B_w, B_h, C_1, C_2, \ldots, C_N]$, and the conditional class probabilities , $P_r(Class_i|Object)$, and the class-specific confidence scores, $P(c)$, are computed. B_x is x point of the center point for the bounding box. B_y is y point of the center point for the bounding box. B_w is the width of the bounding box, B_h is the height of the bounding box, and $C_1, C_2, \ldots, C_N$ denotes each class [50].

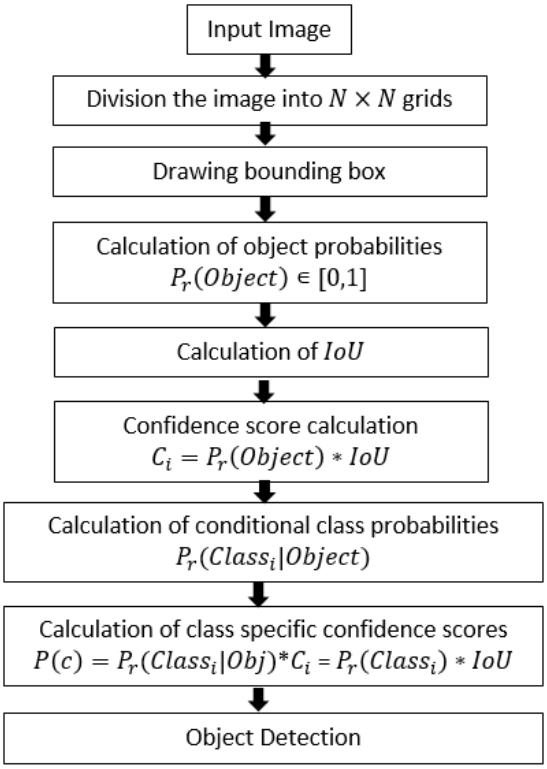

Figure 3.3: Flow diagram of YOLO model

3.3 Evaluation Metrics

Fundamental evaluation metrics are employed to assess the model's performance; Mean Average Precision (mAP), recall, and precision. True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) data are used to compute accuracy, precision and recall measure, and are shown as the confusion matrix in Figure 3.4

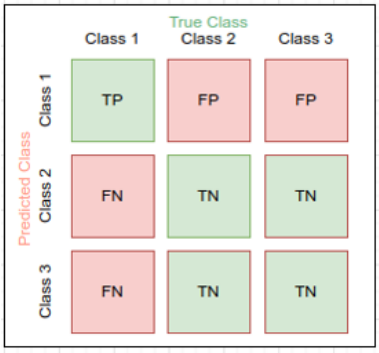

Figure 3.4: General structure of confusion matrix

Accuracy is the measure of correct prediction among all the predictions and it is formulated in Equation 3.4.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{3.4}$$

Precision is the measure of correct positive predictions among all the predicted positives and it is formulated in Equation 3.5.

$$\text{Precision} = \frac{TP}{TP + FN} \tag{3.5}$$

Recall is the measure of correct positive predictions among all the actual positives and it is formulated in Equation 3.6.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.6}$$

Average Precision (AP), or the average accuracy of each class, is represented by the area under the precision–recall curve for single class, and it is formulated in Equation 3.7.

$$AP = \int_0^1 P(R)\, dR \tag{3.7}$$

The mean of AP is known as the mAP metric across all classes, and it is formulated in Equation 3.8.

$$mAP = \frac{\sum_{i=1}^{N_{\text{cls}}} AP_i}{N_{\text{cls}}} \tag{3.8}$$

where the number of classes is N_{cls}. Two methods are used to evaluate the mAP value based on the IoU value: mAP(50) and mAP(50-95). The computed value is mAP(50) while the IoU threshold value is set to 0.5, and mAP(50-95) when the IoU threshold value shifts from 0.5 to 0.95 [51]. .

CHAPTER 4

ISAR IMAGE DATASET

Due to the requirements of deep learning algorithms, extended datasets must be studied to answer the classification problem with high accuracy without using deep networks. ISAR images of drone targets are reconstructed using the monostatic mmWave radar system. A quadcopter drone is selected as a target.

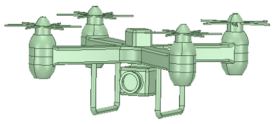

Figure 4.1: Quadcopter drone target

The quadcopter drone is placed in XYZ coordinate system in part a; the ISAR image structure is given with downrange and cross-range of image size in part b; and the result of the ISAR image for the individual drone is shown in part c in Figure 4.2.

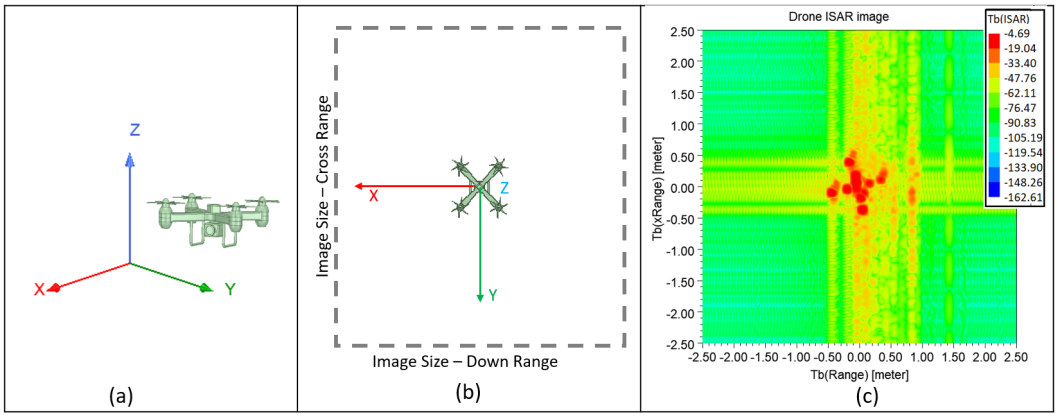

Figure 4.2: ISAR image reconstruction steps a) drone target in xyz coordinate system; b) ISAR image structure c) ISAR image result of individual drone

4.1 Drone Swarm Scenarios

A drone swarm is a technique in which several drones work together to accomplish particular tasks without centralized control. These swarms, modeled after natural bee or bird swarms, function as collective intelligence by continuously interacting with one another via artificial intelligence and communication protocols. Drone swarms, which have become quite widespread today, are created in various forms. In this study, drones with different sizes and payloads are included in the swarm to provide drone diversity. Drone swarm structures are created based on the formation, size, and payload configurations.

4.1.1 Formation

To retain the same formation in a swarm structure, drones move in unison with one another and adhere to the borders of structures that have been allocated to them [50]. A static structure with a commonly recognized geometric shape (line, square, cross, or triangle) is used to construct a swarm [49, 52]. The number of drones can change based on the purpose, and formation structures can be given alternative names. Figure 4.3 shows an example of formation types including line, square, triangle, and cross.

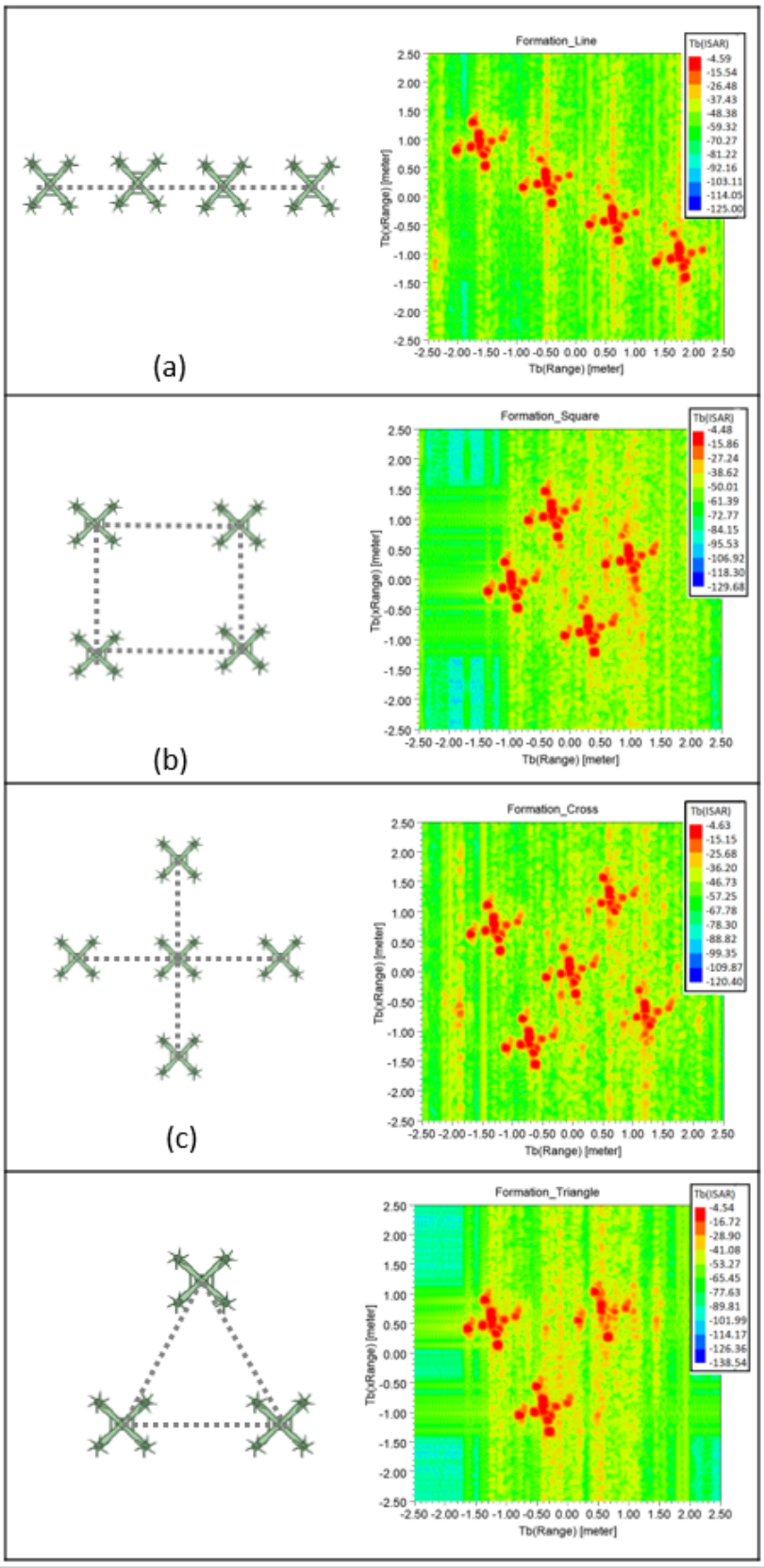

Figure 4.3: Examples of formation scenarios a) Line b) Square c) Cross d) Triangle

4.1.2 Size

There may be drones of different sizes in the drone swarm structure. Therefore, in addition to formation detection, it is also important to determine the size of the drone. Drone sizes are included in the literature as nano, micro, mini, small, medium and large [53] to suit different purposes. In this study, the drones to be included in the drone swarm were created by taking the most commonly used mini, small and medium drone types as seen in Figure 4.4.

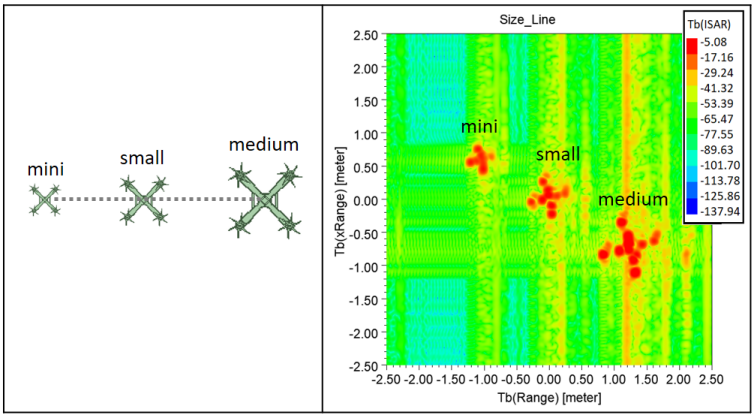

Figure 4.4: Example of size scenarios

4.1.3 Payload

The ability to precisely determine if a drone is carrying a payload has become more and more important as drone technology develops. This capacity is essential for some applications, such as improving security measures, preventing illicit activity, and guaranteeing airspace safety [54]. The circular load design is chosen for this thesis. Figure 4.5 provides a payload drone illustration.

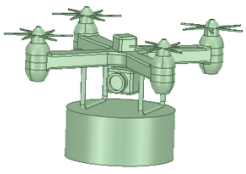

Figure 4.5: Payload drone target

The payload drone in ISAR image is seen in Figure 4.6. The data is concentrated in

the drone's arms, particularly in the direction of the propellers.

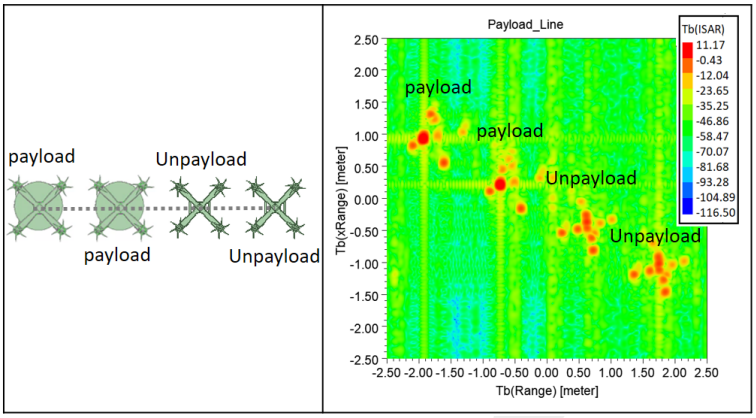

Figure 4.6: Example of payload/unpayload scenarios

A variety of formations, sizes, and payload configurations are covered in ISAR images. The dataset is created by expanding it at different look angles (0° to 350° with 10° increments). ISAR image results for cross formation type at different look angles are given in Figure 4.7

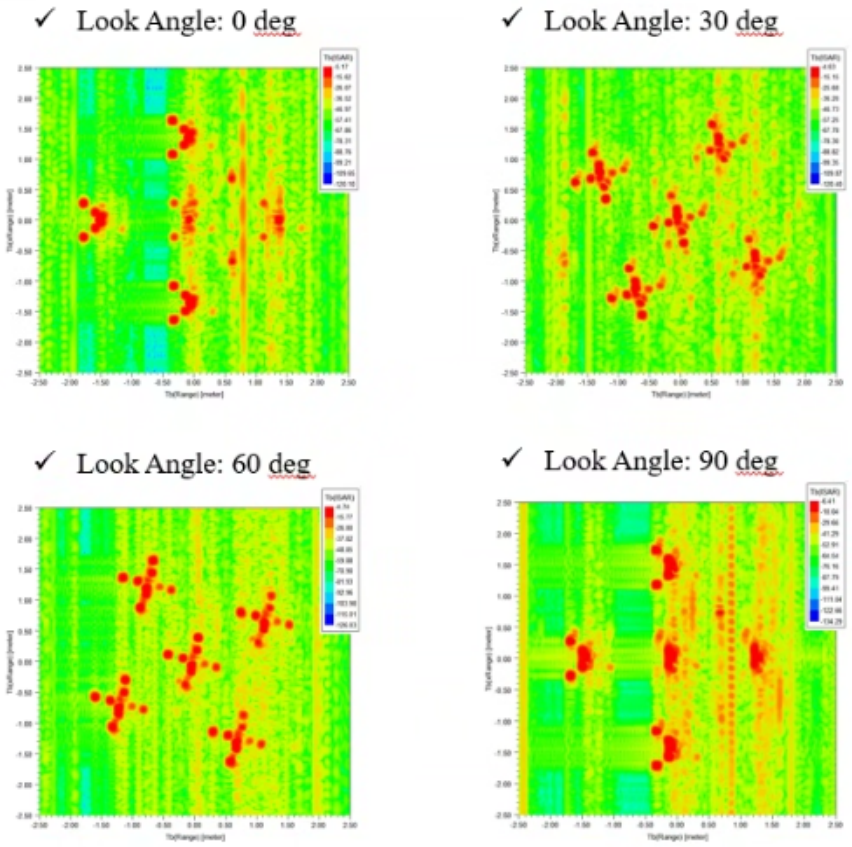

Figure 4.7: ISAR image results at various look angles

The dataset is created with 216 images from each class and 864 images in total. ISAR image scenarios created for the four types of formations are displayed in Figure 4.8.

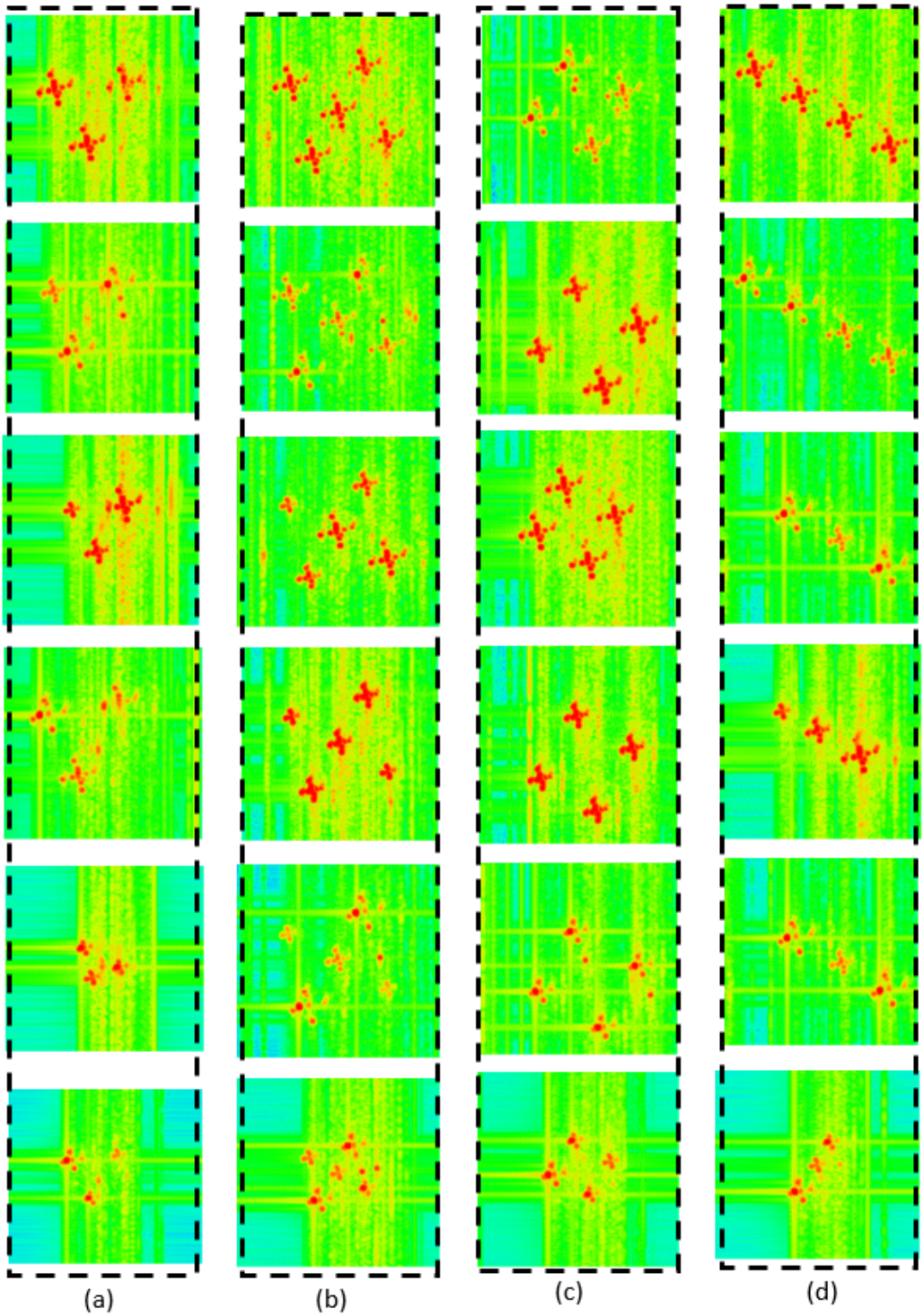

Figure 4.8: ISAR image scenarios a) Triangle b) Cross c) Square d) Line

CHAPTER 5

RESULTS

5.1 Image and Simulation Parameters

ISAR images are created using ANSYS, an engineering simulation application with RADARPre and RADARPost which are Application Customization Toolkit (ACT) extensions. To produce ISAR images of the targets, radar parameters are established and the required procedures (plane wave incidence angles, frequency sweeps, and data reports) are completed [35]. In an ISAR image, the angular resolution in the synthetic aperture determines the cross-range resolution, while the bandwidth determines the down-range resolution. Depending on the properties of the dispersed for various measurement frequencies, the signals reflected off the target include varying amounts of phase information. The target's ISAR picture is impacted by these phase values. Because of this, ISAR images of the same object at several frequencies may not seem the same. ISAR images at mmWave frequency ranges are reconstructed. Radar parameters are given in Table 5.1.

Table 5.1: Radar Parameters

Parameters	**Values**
Frequency (f)	77 GHz
Image Size - Down-range ($x_{\max}$)	5 m
Image Size - Cross-range ($y_{\max}$)	5 m
Down-range Resolution (Δx)	0.05 m
Cross-range Resolution (Δy)	0.05 m
Polarization	Phi-Phi

HFSS SBR+ solver radar parameters are used to calculate simulation parameters. Aspect resolution, bandwidth, aspect angle, and frequency step are crucial simulation parameters. The ISAR image reconstruction process is carried out following the completion of the simulation parameter computation. Simulation parameters are given using radar parameters in Table 5.2.

Table 5.2: Simulation Parameters

Parameters	**Values**
Bandwidth (B)	3 GHz
Aspect Angle ($\Delta\phi$)	2.23°
Frequency Step (Δf)	30 MHz
Aspect Resolution ($\Delta\phi$)	0.02°

ISAR images of individual drone are reconstructed using various types of window functions. Using IQM, the window function is selected based on metrics results. PSNR, SSIM, and IE metrics are used for window functions. Window functions are chosen based on the most used of the literature for ISAR images [55, 56, 57]. The window functions evaluated are chosen from the most frequently used ones in ISAR literature. IQM results are given in Table 5.3.

Table 5.3: PSNR, SSIM, and IE values for various window functions

Window Function	**PSNR**	**SSIM**	**IE**
Bartlett	14.3348	0.8428	3.7952
Bartlett Hanning	13.5208	0.8428	3.8272
Blackman	12.1440	0.7939	3.9136
Blackman Harris	11.6335	0.7728	3.9911
Hanning	12.4323	0.8072	3.9250
Hamming	16.3645	0.9067	3.6576
Triangular	13.2950	0.8358	3.9293

When the table is examined, it can be seen that the hamming window has the highest SSIM, the smallest IE, and the PSNR. This shows that the hamming window function is the best choice for this image input. Hamming windows strike a balance between reducing spectral leakage and preserving good resolution characteristics, making them an excellent choice for ISAR imaging, where precise target details are essential. As a

result, drone swarm ISAR images are extracted using hamming window.

5.2 Image Recognition and Object Detection

After the radar and simulation parameters are adjusted well, high resolution, one of the crucial criteria in the image classification process, is attained. Drone swarms are created in line, square, cross, and triangle formations. As long as they cooperate following the swarm's objective, drone swarms can be created in a variety of configurations without being constrained. ISAR images are used to produce a formation dataset. These images are reconstruct at several look angles, ranging from 0° to 360°, with a 10° variation for each type of swarm formation. ISAR images are classified with computer vision algorithms in Google Colab platform. Google Colab is a free cloud-based platform that allows users to write and execute Python code in a Jupyter Notebook environment. It provides free access to GPUs and TPUs, making it useful for computer vision, data analysis, and scientific computing. The formation type of drone swarm are recognized in the image recognition part and, drones in swarms are detected based on size and payload in the object detection part.

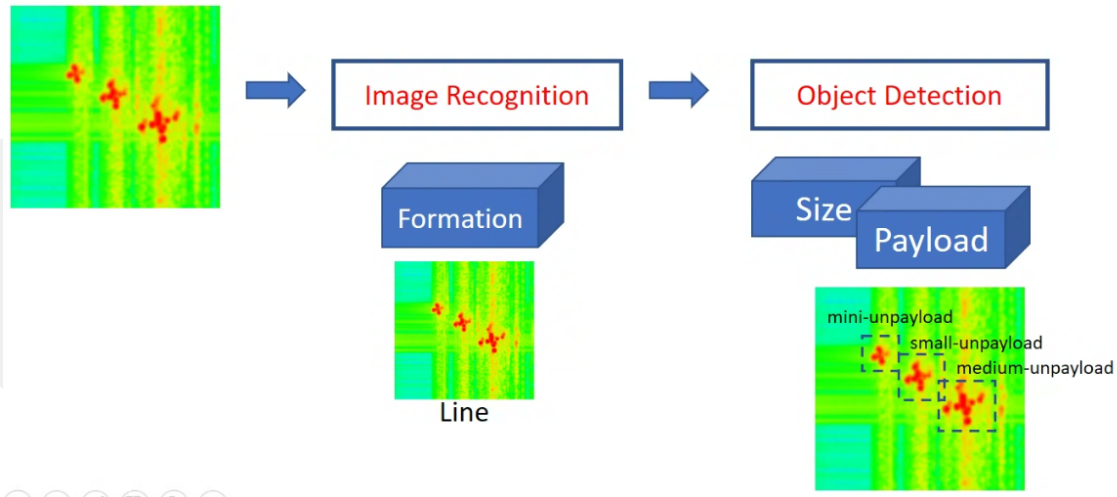

Figure 5.1: Image recognition and object detection parts

The dataset is used to classify drone swarm formation types using a variety of classification algorithms. It consists of 864 images in total (216 images for each class). The created CNN model for the dataset is then used in the classification step. Figure 5.2

displays CNN model with all of the layers with varying output sizes and the number of channels (N) values.

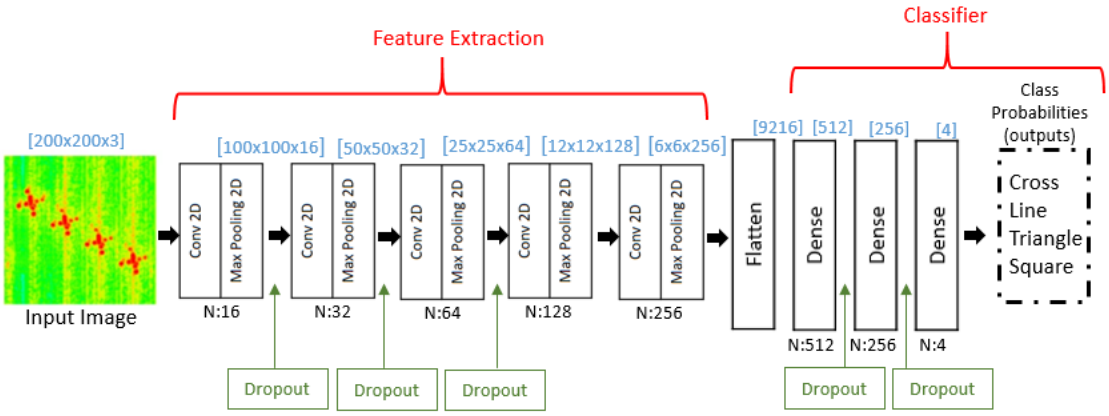

Figure 5.2: CNN model for image recognition part

Three-by-three kernel filters are part of the convolutional layers. Additionally, the batch size is 32. Adam, or the stochastic gradient descent algorithm, is used as an optimizer, and sparse categorical cross-entropy is used as a loss function. Additionally, the model uses dropout, one of the regularization approaches, to improve performance. Typically, dropout processes utilize 20–30% ratios; for the proposed model, a 20% ratio is chosen.

The data augmentation method plays an important role in improving performance in deep learning applications. Data augmentation methods have so many advantages. It efficiently expands the dataset's size by generating new samples from the available data, which may improve model performance. By giving the model a more varied collection of samples, it adds more variables to the data, which can help avoid overfitting. It improves the model's ability to generalize to examples that haven't been seen before by exposing it to a greater variety of data variants. Geometric transformations and photometric (or color space) transformations are the two categories into which many tutorials and non-academic resources divide image data augmentation. Both involve manipulating image files in a comparatively straightforward manner. Techniques that affect the original image's space and arrangement, such as scaling, zooming, or orientation adjustments, are categorized under the first one. To achieve diversity, ISAR images are reconstructed at varied look angles. Additionally, data augmentation is

applied to increase the size of the dataset. Data augmentation activities are carried out minimally to avoid distorting the formation shape of the drone swarm structure, as the formation kinds of the drone swarm in the image will be classified. Rotation, zoom, and flip functions are employed [58]. Sample images acquired after the data augmentation process are shown in Figure 5.3.

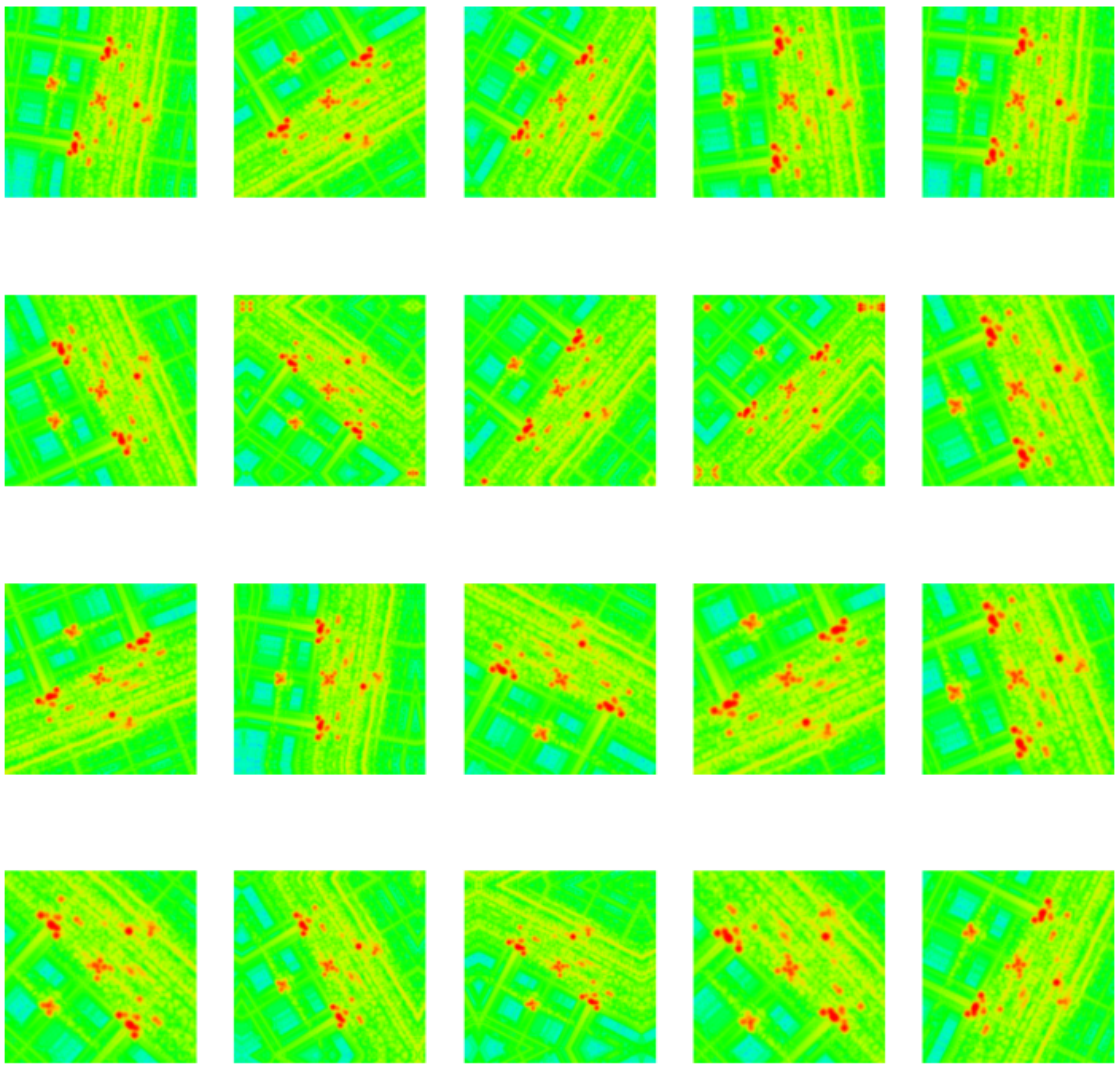

Figure 5.3: Examples of data augmented images

To improve model performance, the dataset is separated into training, validation, and testing phases. The ratios of 70–80% of the data for training, 10-15% for validation, and 10–20% for testing are the most commonly utilized and acknowledged in the literature. The ratio of 80:10:10 (training, validation, and testing) is often more widely acknowledged [59, 60]. Additionally, this ratio serves as a benchmark for this thesis.

With the help of the validation set, the model is fitted during the training phase. The model's hyperparameters are adjusted using the validation set to prevent overfitting. Additionally, the test set generates classification performance and assesses the model. Thirty epochs are used to train the model. Figure 5.4 shows the loss and accuracy outcomes of training and validation versus epoch.

Figure 5.4: Train and validation graph versus epoch for the image recognition part

Performance results for training and validation increase in tandem across a series of epochs, peaking at the conclusion. This demonstrates how minimal the overfitting issue is. A test subset of 90 images is then used to evaluate the trained model, and the overall performance result is found to be 98.8%. In Figure 5.5, the confusion matrix is shown. Analysis of the confusion matrix reveals that just one triangle formation type is confused with the square formation type. As a result, the classification performance of the model is encouraging.

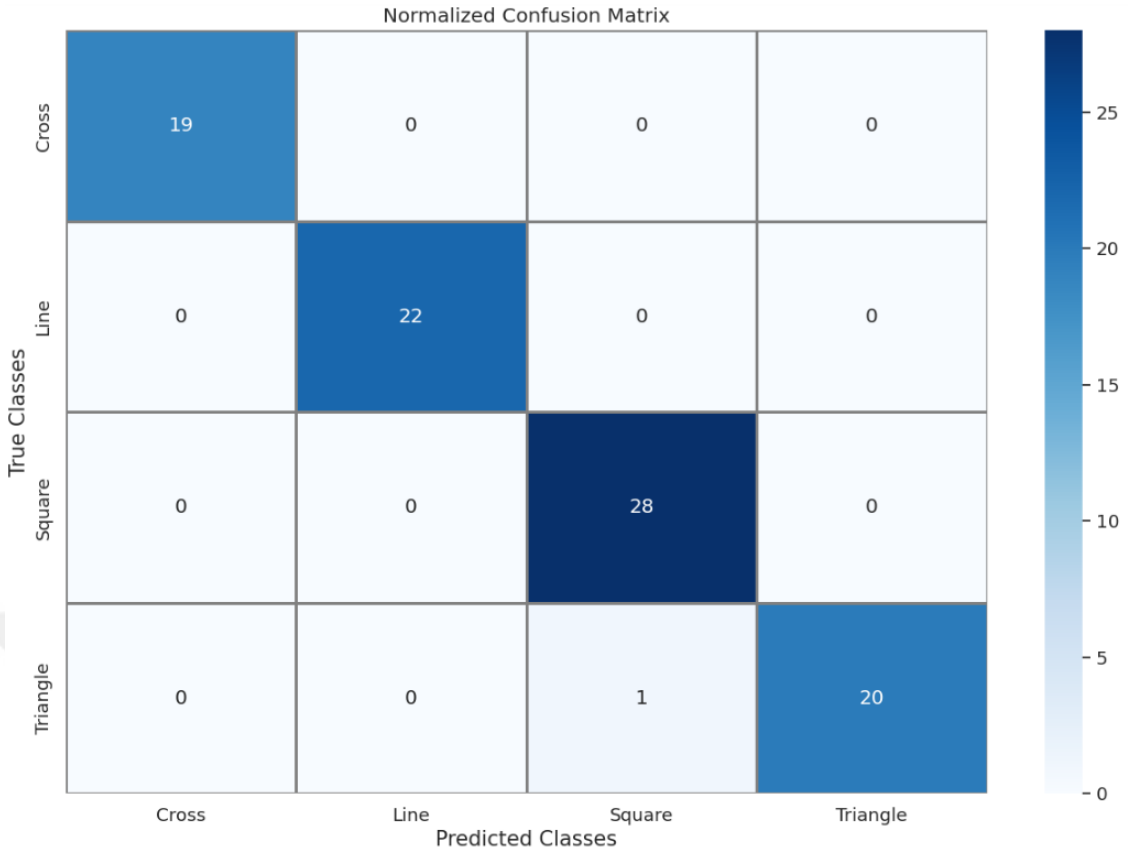

Figure 5.5: Confusion matrix for the image recognition part

After the classification of the formation type of the drone swarm, drone types in the swarm are detected based on size and payload scenarios using the YOLO11 model, which is an object detection method. YOLO11 expands on the impressive developments of previous YOLO versions with significant architecture and training method enhancements, providing a versatile choice for a range of computer vision workloads. Figure 5.6 shows some instances of ISAR images produced throughout the training procedure.

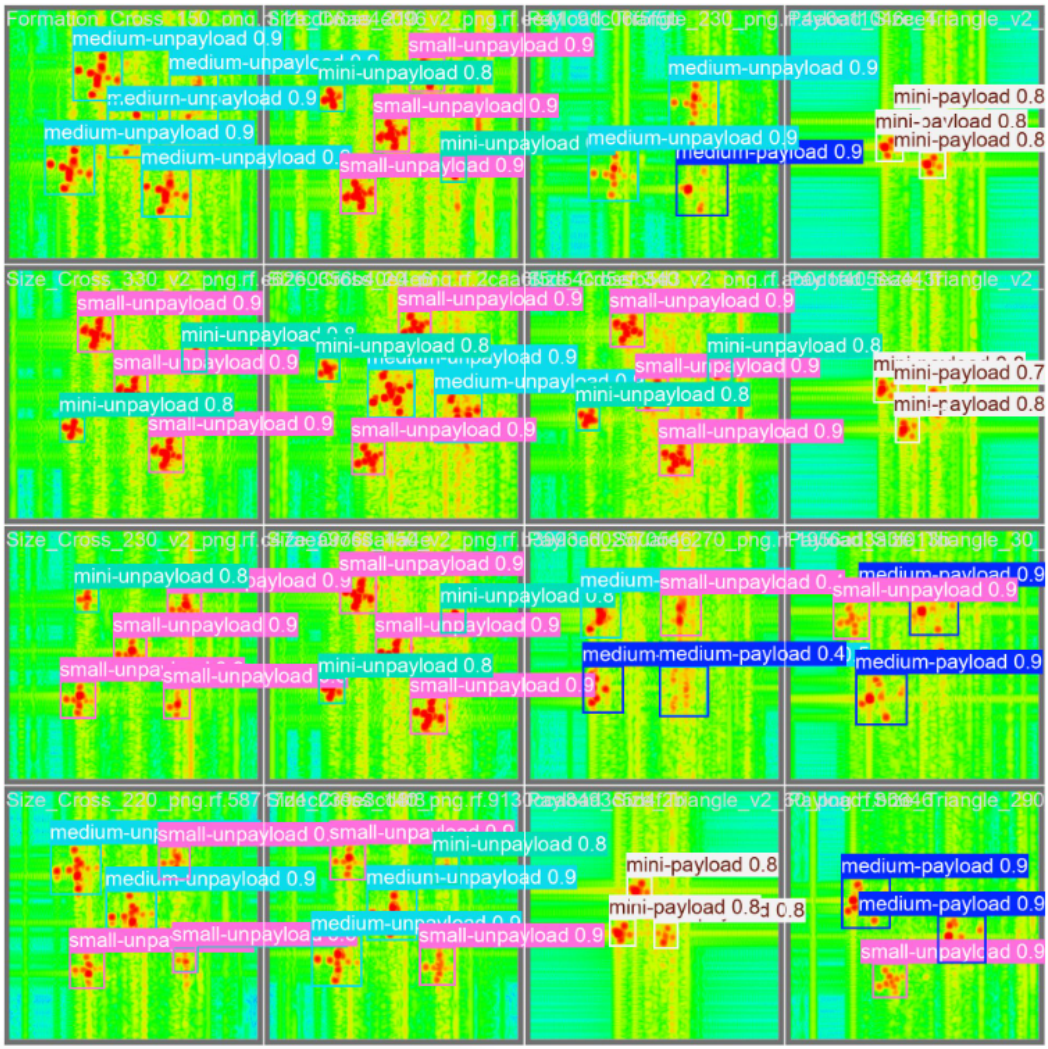

Figure 5.6: Examples of images in the training process for the object detection part

Figure 5.7 provides the evaluation metrics used to assess the model's performance. According to the mAP50 graph, the mAP50 value rises gradually during training and eventually reaches about 98.8%. The curve often rises, despite minor initial oscillations, suggesting that the model's performance results increase over time. Throughout the training period, the mAP50-95 value rises, reaching roughly 75.4%, as the mAP50-95 graph illustrates. Since it is a more complex and broad evaluation, this metric has a smaller value than mAP50. The increasing trend indicates that at various IoU levels, the model's overall performance improves. It demonstrates that even though it is a more challenging metric, it produces a successful outcome. Therefore, it can be concluded that the model performs successfully and promisingly in the object detection task.

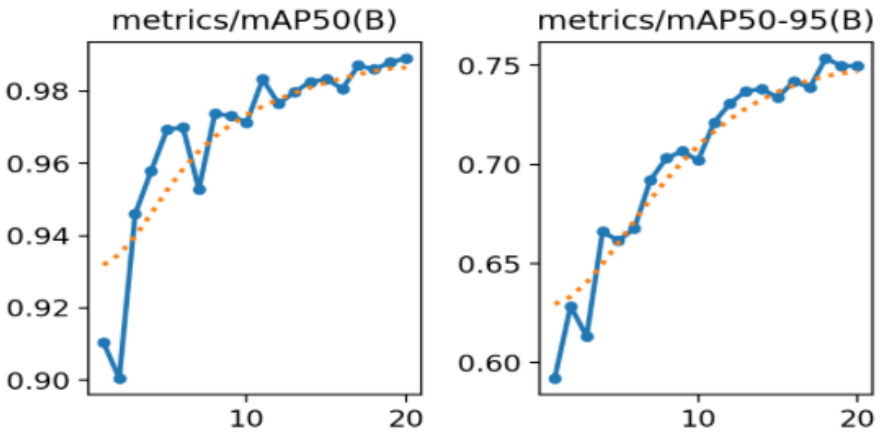

Figure 5.7: Evaluation metrics for the object detection part

The trained model is then assessed using test images, and the overall classification performance is 75.0% for the mAP50-95 metric and 98.9% for the mAP50. The confusion matrix is examined in the Figure 5.8. One mini-unpayload drone is predicted as mini-payload when the confusion matrix is analyzed, but a medium-payload drone is seen as being present even though it is not in the image. Furthermore, despite their presence in the image, one mini-payload and one mini-unpayload drone are not observed. According to the results of the confusion matrix, the model typically produces performance that is satisfactory and does not make many mistakes between classes.

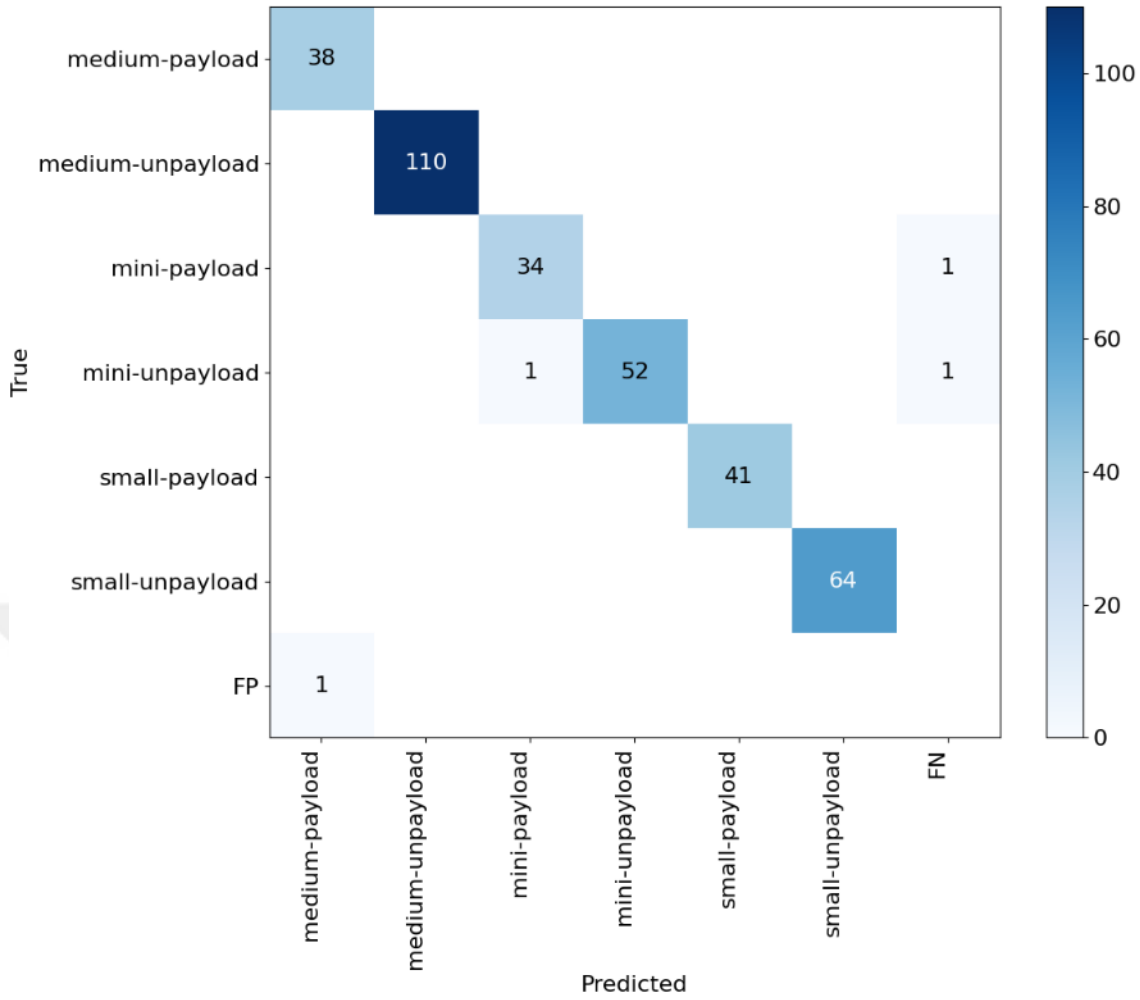

Figure 5.8: Confusion matrix for the object detection part

CHAPTER 6

CONCLUSIONS

The importance of an anti-drone application has increased with the growing use of drones and drone swarms in technology. In particular, detection systems designed for scenarios with a constrained field of view continue to be difficulty in the literature. This thesis first recognizes the formation structure of the drone swarm utilizing ISAR images produced in mmWave frequency bands. Next, the drones in the swarm are detected based on their size and payload configurations. SBR+ solvers in ANSYS-HFSS electromagnetic simulation software are used to generate ISAR images of drone swarms. A quadcopter drone structure is used to create swarm structures. The hamming window is selected for the ISAR image reconstruction step based on IQM results. Using the formation types of simple geometric shapes like line, cross, square, and triangle, drone swarm structures are developed. The three sizes of the drones in the swarm were intended to be medium, small, and mini. Furthermore, numerous types of drones are developed based on whether or not they carry a payload. Each drone swarm structure is replicated at different look angles (10° increments between 0° and 350°) to increase the dataset. It consists of 864 images in total (216 images for each class). The ratio of 80:10:10 (training, validation, and testing phases) is used for the dataset partition. Furthermore, the data augmentation technique is used to address the overfitting issue and diversity of the data. Rotation, zoom, and flip functions are employed. CNN model is used to identify the type of formation from ISAR images of drone swarms. The overall performance result is found to be 98.8%. Following the identification of the drone swarm's formation type, six classes based on size and payload configurations are identified using the YOLO object detection model. The overall detection performance is 75.0% for the mAP50-95 metric and 98.9% for the mAP50.

A sizable dataset and detection algorithm made up of ISAR images of drone swarms are therefore added to the literature, and the findings produced in the thesis are highly encouraging. As a future work, ISAR images for drone swarms can be diversified based on various scenarios. The other image recognition and object detection models in the literature can be applied to the ISAR dataset.

REFERENCES

[1] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Progress in Aerospace sciences*, vol. 91, pp. 99–131, 2017.

[2] A. Coluccia, G. Parisi, and A. Fascista, "Detection and classification of multirotor drones in radar sensor networks: A review," *Sensors*, vol. 20, no. 15, p. 4172, 2020.

[3] Y. Zhou, B. Rao, and W. Wang, "Uav swarm intelligence: Recent advances and future trends," *Ieee Access*, vol. 8, pp. 183 856–183 878, 2020.

[4] J. Tang, H. Duan, and S. Lao, "Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: A comprehensive review," *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4295–4327, 2023.

[5] C. Wang, L. Meng, Q. Gao, J. Wang, T. Wang, X. Liu, F. Du, L. Wang, and E. Wang, "A lightweight uav swarm detection method integrated attention mechanism," *Drones*, vol. 7, no. 1, p. 13, 2022.

[6] V. Subbarayalu and M.A. Vensuslaus, "An intrusion detection system for drone swarming utilizing timed probabilistic automata," *Drones*, vol. 7, no. 4, p. 248, 2023.

[7] H. Ahn, H.L. Choi, M. Kang, and S. Moon, "Learning-based anomaly detection and monitoring for swarm drone flights," *Applied Sciences*, vol. 9, no. 24, p. 5477, 2019.

[8] C. Wang, Y. Su, J. Wang, T. Wang, and Q. Gao, "Uavswarm dataset: An unmanned aerial vehicle swarm dataset for multiple object tracking," *Remote Sensing*, vol. 14, no. 11, p. 2601, 2022.

[9] J. Cai, F. Wang, and B. Yang, "Exploration of uav cluster defense technology," *Aerodyn. Missile J*, vol. 12, no. 5, 2020.

[10] G. Ciattaglia, G. Temperini, S. Spinsante, and E. Gambi, "mmwave radar features extraction of drones for machine learning classification," in *2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. IEEE, 2021, pp. 259–264.

[11] V.C. Chen, *The micro-Doppler effect in radar*. Artech house, 2019.

[12] G. Franceschetti and R. Lanari, *Synthetic aperture radar processing*. CRC press, 2018.

[13] F. Bovenga, *Synthetic Aperture Radar (SAR) Techniques and Applications*. MDPI-Multidisciplinary Digital Publishing Institute, 2020.

[14] B.Y. Gökdoğan, R.B. Çoruk, E. Aydın, and A. Kara, "2d millimeter-wave sar imaging with automotive radar," *Journal of Science, Technology and Engineering Research*, vol. 5, no. 1, pp. 68–77.

[15] V. Borkar, A. Ghosh, R. Singh, and N. Chourasia, "Radar cross-section measurement techniques," *Defence Science Journal*, vol. 60, no. 2, pp. 204–212, 2010.

[16] M. Cheney and B. Borden, *Fundamentals of radar imaging*. SIAM, 2009.

[17] J. Guanghu, G. Xunzhang, and D. Zhen, "Two-dimensional length extraction of ballistic target from isar images using a new scaling method by affine registration," *Defence Science Journal*, vol. 64, no. 5, p. 458, 2014.

[18] X. Ye, F. Zhang, Y. Yang, D. Zhu, and S. Pan, "Photonics-based high-resolution 3d inverse synthetic aperture radar imaging," *IEEE Access*, vol. 7, pp. 79 503–79 509, 2019.

[19] M. Martorella, E. Giusti, A. Capria, F. Berizzi, and B. Bates, "Automatic target recognition by means of polarimetric isar images and neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 11, pp. 3786–3794, 2009.

[20] P. Ni, Y. Liu, H. Pei, H. Du, H. Li, and G. Xu, "Clisar-net: A deformation-robust isar image classification network using contrastive learning," *Remote Sensing*, vol. 15, no. 1, p. 33, 2022.

[21] R. Xue, X. Bai, and F. Zhou, "Saisar-net: A robust sequential adjustment isar image classification network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.

[22] P. Ni, J. Sheng, L. Jiang, and G. Xu, "Sequential isar images classification using cnn-bi-lstm method," in *2022 3rd China International SAR Symposium (CISS)*. IEEE, 2022, pp. 1–5.

[23] S. Musman, D. Kerr, and C. Bachmann, "Automatic recognition of isar ship images," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1392–1404, 1996.

[24] K.T. Kim, D.K. Seo, and H.T. Kim, "Efficient classification of isar images," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 5, pp. 1611–1621, 2005.

[25] X. He, N. Tong, and X. Hu, "Automatic recognition of isar images based on deep learning," in *2016 CIE International Conference on Radar (RADAR)*. IEEE, 2016, pp. 1–4.

[26] S.J. Lee, M.J. Lee, K.T. Kim, and J.H. Bae, "Classification of isar images using variable cross-range resolutions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2291–2303, 2018.

[27] M. Barbary, A.S. Hafez, M.H. Abd El-azeem, T. Soliman, A.E. Elbanna, M. Shalaby, W.M. Saad, and M. Mabrouk, "Extended drones tracking from isar images with doppler effect and orientation based robust sub-random matrices algorithm," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 12, pp. 12 648–12 666, 2022.

[28] C.J. Li and H. Ling, "An investigation on the radar signatures of small consumer drones," *IEEE Antennas and Wireless Propagation Letters*, vol. 16, pp. 649–652, 2016.

[29] Y. Yang, X.S. Wang, Y.Z. Li, and L.F. Shi, "Rcs measurements and isar images of fixed-wing uav for fully polarimetric radar," in *2019 International Radar Conference (RADAR)*. IEEE, 2019, pp. 1–5.

[30] C.J. Li and H. Ling, "Wide-angle, ultra-wideband isar imaging of vehicles and drones," *Sensors*, vol. 18, no. 10, p. 3311, 2018.

[31] J. Xu, M. Liu, F. Zhao, K. Cheng, and L. Yang, "Statistical isar imagery for low-altitude and small-size uav based on sparse bayesian learning," in *2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*. IEEE, 2019, pp. 1–6.

[32] K. Ogawa, D. Tsagaanbayar, and R. Nakamura, "Isar imaging for drone detection based on backprojection algorithm using millimeter-wave fast chirp modulation mimo radar," *IEICE Communications Express*, 2024.

[33] K. Ogawa, D. Tsagaanbayar, and R. Nakamura, "Isar imaging of drones based on time domain correlation algorithm using millimeter-wave fast chirp modulation mimo radar," in *2024 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNeT)*. IEEE, 2024, pp. 76–79.

[34] A.N. Sayed, O.M. Ramahi, and G. Shaker, "In the realm of aerial deception: Uav classification via isar images and radar digital twins for enhanced security," *IEEE Sensors Letters*, 2024.

[35] W.K. Lee and K.M. Song, "Enhanced isar imaging for surveillance of multiple drones in urban areas," in *2018 International Conference on Radar (RADAR)*. IEEE, 2018, pp. 1–4.

[36] V. Chen, M. Martorella *et al.*, *Inverse synthetic aperture radar imaging: principles, algorithms and applications*. IET, 2014.

[37] C. Oliver and S. Quegan, *Understanding synthetic aperture radar images*. SciTech Publishing, 2004.

[38] H. Ansys, "Ansys hfss: Best-in-class 3d high frequency electromagnetic simulation software," *HFSS and Ansys, Feb*, 2023.

[39] X. Li, X. Bai, and F. Zhou, "High-resolution isar imaging and autofocusing via 2d-admm-net," *Remote Sensing*, vol. 13, no. 12, p. 2326, 2021.

[40] H. Wang, K. Li, X. Lu, Q. Zhang, Y. Luo, and L. Kang, "Isar resolution enhancement method exploiting generative adversarial network," *Remote Sensing*, vol. 14, no. 5, p. 1291, 2022.

[41] C. Zeng, W. Zhu, X. Jia, and L. Yang, "Sparse aperture isar imaging method based on joint constraints of sparsity and low rank," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 168–181, 2020.

[42] R. Klette, A. Koschan, K. Schluns, and K. Achluns, *Computer vision*. Springer-Verlag New York, 1998.

[43] G. Stockman and L.G. Shapiro, *Computer vision*. Prentice Hall PTR, 2001.

[44] R.B. Coruk, B.Y. Gokdogan, M. Benzaghta, and A. Kara, "On the classification of modulation schemes using higher order statistics and support vector machines," *Wireless Personal Communications*, vol. 126, no. 2, pp. 1363–1381, 2022.

[45] R.B. Çoruk, A. Kara, and E. Aydın, "Drone swarm classification from isar imaging," *Journal of Science, Technology and Engineering Research*, vol. 5, no. 2, pp. 127–134, 2024.

[46] S. Khan, H. Rahmani, S.A.A. Shah, and M. Bennamoun, "Convolutional neural network," in *A guide to convolutional neural networks for computer vision*. Springer, 2018, pp. 43–68.

[47] V. Kurian and V. Jacob, "Importance of cnn in the classification of remote sensing images," in *2023 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*. IEEE, 2023, pp. 1–4.

[48] O.A. Montesinos López, A. Montesinos López, and J. Crossa, "Convolutional neural networks," in *Multivariate statistical machine learning methods for genomic prediction*. Springer, 2022, pp. 533–577.

[49] B. Alkouz, A. Abusafia, A. Lakhdari, and A. Bouguettaya, "In-flight energy-driven composition of drone swarm services," *IEEE Transactions on Services Computing*, vol. 16, no. 3, pp. 1919–1933, 2022.

[50] W.Y.H. Adoni, S. Lorenz, J.S. Fareedh, R. Gloaguen, and M. Bussmann, "Investigation of autonomous multi-uav systems for target detection in distributed environment: Current developments and open challenges," *Drones*, vol. 7, no. 4, p. 263, 2023.

[51] S. Chifeng and W. Qianqian, "Object-centered petri net process prediction: A case study of multi-system intelligent healthcare," *IEEE Access*, 2024.

[52] B. Alkouz and A. Bouguettaya, "Formation-based selection of drone swarm services," in *MobiQuitous 2020-17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2020, pp. 386–394.

[53] M.F.F. Rahman, S. Fan, Y. Zhang, and L. Chen, "A comparative study on application of unmanned aerial vehicle systems in agriculture," *Agriculture*, vol. 11, no. 1, p. 22, 2021.

[54] F. Fioranelli, M. Ritchie, H. Griffiths, and H. Borrion, "Classification of loaded/unloaded micro-drones using multistatic radar," *Electronics Letters*, vol. 51, no. 22, pp. 1813–1815, 2015.

[55] Y. Sun, Q. Liu, J. Cai, and T. Long, "A novel method for designing general window functions with flexible spectral characteristics," *sensors*, vol. 18, no. 9, p. 3081, 2018.

[56] R.M. Nuthalapati, "High resolution reconstruction of isar images," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 2, pp. 462–472, 1992.

[57] K.M. Prabhu, *Window functions and their applications in signal processing*. Taylor & Francis, 2014.

[58] M. Barbeau, "Recognizing drone swarm activities: Classical versus quantum machine learning," *Digitale Welt*, vol. 3, no. 4, pp. 45–50, 2019.

[59] M. Ahishali, S. Kiranyaz, T. Ince, and M. Gabbouj, "Dual and single polarized sar image classification using compact convolutional neural networks," *Remote Sensing*, vol. 11, no. 11, p. 1340, 2019.

[60] J. Gawlikowski, M. Schmitt, A. Kruspe, and X.X. Zhu, "On the fusion strategies of sentinel-1 and sentinel-2 data for local climate zone classification," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 2081–2084.

CHAPTER 7

ALGORITHMS

7.1 Algorithm 1

The following python code is used for the image recognition part.

```
import matplotlib.pyplot as plt
import numpy as np
import os
import PIL
import sklearn.metrics
import tensorflow as tf

from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from sklearn.metrics import confusion_matrix,accuracy_score,
classification_report
import seaborn as sns
import pandas as pd
from keras import models
import pickle
from keras.utils import load_img

from google.colab import drive
drive.mount('/content/drive')

data_root='/content/drive/My Drive/ALL_dataset_tez/'
data_root_train='/content/drive/My Drive/ALL_dataset_tez/train/'
data_root_valid='/content/drive/My Drive/ALL_dataset_tez/valid/'
data_root_test='/content/drive/My Drive/ALL_dataset_tez/test/'

import pathlib
data_root = pathlib.Path(data_root)
image_count = len(list(data_root.glob('*/*')))
print(image_count)
```

```
batch_size = 32
img_height = 200
img_width = 200

train_ds = tf.keras.utils.image_dataset_from_directory(
data_root_train,
validation_split=0.01,
subset="training",
seed=123,
image_size=(img_height, img_width),
batch_size=batch_size)

val_ds = tf.keras.utils.image_dataset_from_directory(
data_root_valid,
validation_split=0.99,
subset="validation",
seed=123,
image_size=(img_height, img_width),
batch_size=batch_size)

test_ds = tf.keras.utils.image_dataset_from_directory(
data_root_test,
image_size=(img_height, img_width),
batch_size=batch_size
)

class_names = np.array(train_ds.class_names)
num_classes = len(class_names)
print(class_names)

images, labels = next(iter(train_ds))
images = images[:5]
labels = labels[:5]

fig, axes = plt.subplots(nrows=1, ncols=5, figsize=(20, 4))

for i in range(len(images)):
ax = axes[i]
ax.imshow(images[i].numpy().astype(np.uint8))
ax.set_title(f'Label: class_names[labels[i]]')

plt.show()

for image_batch, labels_batch in train_ds:
print(image_batch.shape)
print(labels_batch.shape)
break

AUTOTUNE = tf.data.AUTOTUNE
```

```
train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=AUTOTUNE)
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)

data_augmentation = keras.Sequential(
[
layers.RandomFlip("vertical",
input_shape=(img_height,
img_width,
3)),
layers.RandomRotation(0.2),
layers.RandomZoom(0.2),
]
)

plt.figure(figsize=(10, 10))
for images, _ in train_ds.take(1):
for i in range(20):
augmented_images = data_augmentation(images)
ax = plt.subplot(4,5, i + 1)
plt.imshow(augmented_images[0].numpy().astype("uint8"))
plt.axis("off")

num_classes = len(class_names)

model = Sequential([
data_augmentation,
layers.Rescaling(1./255),
layers.Conv2D(16, 3, padding='same', activation='relu'),
layers.MaxPooling2D(),
layers.Dropout(0.2),
layers.Conv2D(32, 3, padding='same', activation='relu'),
layers.MaxPooling2D(),
layers.Dropout(0.2),
layers.Conv2D(64, 3, padding='same', activation='relu'),
layers.MaxPooling2D(),
layers.Dropout(0.2),
layers.Conv2D(128, 3, padding='same', activation='relu'),
layers.MaxPooling2D(),
layers.Conv2D(256, 3, padding='same', activation='relu'),
layers.MaxPooling2D(),
layers.Flatten(),
layers.Dense(512, activation='relu'),
layers.Dropout(0.2),
layers.Dense(256, activation='relu'),
layers.Dropout(0.2),
layers.Dense(num_classes,  activation='softmax')
])
```

```
model.summary(line_length = 100)

model.compile(optimizer='adam',
loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
metrics=['accuracy'])

epochs=30

history = model.fit(
train_ds,
validation_data=val_ds,
epochs=epochs
)

fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(10, 4))

ax1.plot(history.history['loss'], label='train_loss')
ax1.plot(history.history['val_loss'], label='val_loss')
ax1.set_xlabel('Epoch')
ax1.set_ylabel('Loss')
ax1.legend()

ax2.plot(history.history['accuracy'], label='train_acc')
ax2.plot(history.history['val_accuracy'], label='val_acc')
ax2.set_xlabel('Epoch')
ax2.set_ylabel('Accuracy')
ax2.legend()

fig.tight_layout()

plt.show()

results = model.evaluate(test_ds)
print("    Test Loss: :.5f".format(results[0]))
print("Test Accuracy: :.2f%".format(results[1] * 100))

import numpy as np
import tensorflow as tf
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

predictions = np.array([])
labels = np.array([])

for x, y in test_ds:
predict_x = model.predict(x)
classes_x = np.argmax(predict_x, axis=1)
predictions = np.concatenate([predictions, classes_x])
```

```
labels = np.concatenate([labels, np.array(y)])

label_names = i: class_names[i] for i in range(len(class_names))

conf_matrix = confusion_matrix(labels, predictions, normalize='true')

plt.figure(figsize=(10, 7.5))
sns.heatmap(conf_matrix,
annot=True,cmap="Blues",linecolor="gray",
xticklabels=[label_names[x] for x in sorted(set(labels))],
yticklabels=[label_names[x] for x in sorted(set(labels))])
sns.set(font_scale=1.2)
plt.xlabel("Predicted Classes", fontsize=15)
plt.ylabel("True Classes", fontsize=15)
plt.title('Normalized Confusion Matrix')
plt.show()

cf_matrix = confusion_matrix(labels, predictions)
fig, ax = plt.subplots(figsize=(15,10))
xlabelss=label_names
ylabelss=label_names
sns.heatmap(cf_matrix, linewidths=1, annot=True,
cmap="Blues",linecolor="gray", ax=ax, fmt='g',
xticklabels=[label_names[x] for x in sorted(set(labels))],
yticklabels=[label_names[x] for x in sorted(set(labels))])
plt.xlabel("Predicted Classes", fontsize=15)
plt.ylabel("True Classes", fontsize=15)
plt.title('Normalized Confusion Matrix')
plt.show()

for x, y in test_ds:
predict_x = model.predict(x)
classes_x = np.argmax(predict_x, axis=1)
predictions = np.concatenate([predictions, classes_x])
labels = np.concatenate([labels, np.array(y)])

report = classification_report(labels, predictions,
target_names=class_names)

print(report)

print(predict_x[0].shape, predict_x[1].shape)
```

7.2 Algorithm 2

The following python code is used for the object detection part.

```
gpu_info = !nvidia-smi

!pip install ultralytics

from ultralytics import YOLO
import os
from IPython.display import display, Image
from IPython import display
display.clear_output()
!yolo mode=checks

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="xxxxxxx")
project = rf.workspace("xxxxxxxx").project("xxxxxx")
version = project.version(3)
dataset = version.download("yolov11")

!yolo task=detect mode=train model=yolo11m.pt
data=dataset.location/data.yaml epochs=30 imgsz=640

Image(filename=f'/content/runs/detect/train/confusion_matrix.png',
width=600)

Image(filename=f'/content/runs/detect/train/results.png',
width=600)

Image(filename=f'/content/runs/detect/train/val_batch0_pred.jpg',
width=600)

!yolo task=detect mode=val
model=/content/runs/detect/train/weights/best.pt
data=dataset.location/data.yaml

!yolo task=detect mode=predict
model=/content/runs/detect/train/weights/best.pt conf=0.25
source=dataset.location/test/images save=True

import glob
from IPython.display import Image, display
import matplotlib.pyplot as plt #for image and graph ploting

import glob
from IPython.display import Image, display

base_path = '/content/runs/detect/'
```

```
subfolders = [os.path.join(base_path, d) for d in os.listdir(base_path)
if os.path.isdir(os.path.join(base_path, d)) and d.startswith('predict')]

latest_folder = max(subfolders, key=os.path.getmtime)

image_paths = glob.glob(f'latest_folder/*.jpg')[:90]

!pip install supervision

import supervision as sv
from ultralytics import YOLO
import numpy as np
import matplotlib.pyplot as plt

dataset = sv.DetectionDataset.from_yolo(
images_directory_path="/content/xxxxxx/test/images",
annotations_directory_path="/content/xxxxxx/test/labels",
data_yaml_path="/content/xxxxxx/data.yaml")
model = YOLO("/content/runs/detect/train/weights/best.pt")

def callback(image: np.ndarray) -> sv.Detections:
result = model(image)[0]
return sv.Detections.from_ultralytics(result)

confusion_matrix = sv.ConfusionMatrix.benchmark(
dataset=dataset,
callback=callback)

matrix = confusion_matrix.matrix
correct_predictions = np.trace(matrix)
total_predictions = np.sum(matrix)
overall_accuracy = correct_predictions / total_predictions

print(f"Overall Accuracy: overall_accuracy:.2%")

plt.rcParams.update('font.size': 20)
confusion_matrix.plot()

plt.xticks(fontsize=16)
plt.yticks(fontsize=16)

plt.show()
```

CURRICULUM VITAE (ONLY FOR DOCTORAL THESIS)

Surname, Name: Çoruk, Remziye Büşra

EDUCATION

Degree	Institution	Year of Graduation
PhD	Atılım University Electrical and Electronics Engineering	2025
MS	Atılım University Electrical and Electronics Engineering	2020
BS	Kırıkkale University Electrical and Electronics Engineering	2015
High School	Nermin Mehmet Çekiç Anadolu High School, Ankara	2011

FOREIGN LANGUAGES

Advanced English, Beginner Spanish

PUBLICATIONS

1. Coruk, R. B., Yalcinkaya, B., and Kara, A. "On the design and effectiveness of Simulink-based educational material for a communication systems course", Computer Applications in Engineering Education, 28(6), 1641-1651 (2020)
2. Benzaghta, M., Coruk, R. B., Yalcinkaya, B., and Kara, A. "An Experimental Study on the Influence of Human Movement in Indoor Radio Channel at 28GHz", In 2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), (pp. 1-3) IEEE (2021, May)
3. Doğanay, B., Arslan, M., Demir, E. C., Çoruk, R. B., Gökdoğan, B. Y., and Aydin, E. "UAV Detection and Ranging with 77-81 GHz FMCW Radar", In 2022 30th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4) IEEE (2022, May)

4. Coruk, R. B., Gokdogan, B. Y., Benzaghta, M., and Kara, A. “On the classification of modulation schemes using higher order statistics and support vector machines”, Wireless Personal Communications, 126(2), 1363-1381 (2022)

5. Benzaghta, M., Gokdogan, B. Y., Coruk, R. B., and Kara, A. “Modeling and measurement of human body blockage loss at 28 GHz”, Journal of Electromagnetic Waves and Applications, 37(4), 538-548 (2023)

6. Gokdogan, B. Y., Coruk, R. B., Benzaghta, M., and Kara, A. “A Hybrid-Flipped Classroom Approach: Students’ Perception and Performance Assessment”, Ingeniería e Investigación, 43(3), 1 (2023)

7. Benzaghta, M., Gokdogan, B. Y., Coruk, R. B., and Kara, A. “Blockage loss and shadow fading behavior of millimeter-wave signals due to human bodies at 28 GHz”, Microwave and Optical Technology Letters, 66(1), e33997 (2024)

8. Yalcinkaya, B., Coruk, R. B., Kara, A., and Tora, H. “Hierarchical Classification of Analog and Digital Modulation Schemes Using Higher-Order Statistics and Support Vector Machines”, Wireless Personal Communications, 1-21 (2024)

9. Gökdoğan, B. Y., Çoruk, R. B., Aydın, E., and Kara, A. “2D Millimeter-Wave SAR Imaging with Automotive Radar”, Journal of Science, Technology and Engineering Research, 5(1), 68-77 (2024)

10. Çoruk, R. B., Kara, A., and Aydın, E. “Drone Swarm Classification from ISAR Imaging”, Journal of Science, Technology and Engineering Research, 5(2), 127-134 (2024).

11. Yalçınkaya B, Benzaghta M, Çoruk RB, Kara A. “Evaluating the Educational Effectiveness of Radar Systems Laboratory Sessions in the Undergraduate Curriculum“ Journal of Science, Technology and Engineering Research, 5(2), 186-198 (2024)

RESEARCH INTERESTS

- ➢ Radar Systems
- ➢ Communication Systems
- ➢ Artificial Intelligence
- ➢ Electronic Warfare
- ➢ Engineering Education