T.C

ISTANBUL AYDIN UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

AUTOMATIC TARGET RECOGNITION FOR SYNTHETIC APERTURE RADAR DATA

MASTER'S THESIS

Hasna EL HASNAOUY

Department of Electrical and Electronics Engineering

Electrical and Electronics Engineering Program

August 2023

T.C

ISTANBUL AYDIN UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

AUTOMATIC TARGET RECOGNITION FOR SYNTHETIC APERTURE RADAR DATA

MASTER'S THESIS

Hasna EL HASNAOUY

(Y2013.300025)

Department of Electrical and Electronics Engineering

Electrical and Electronics Engineering Program

Thesis Advisor: Assist. Prof. Dr. Necip Gökhan KASAPOĞLU

August 2023

ONAY FORMU

DECLARATION

I hereby declare with the respect that the study “Automatic Target Recognition for Synthetic Aperture Radar Data”, which I submitted as a Master thesis, is written without any assistance in violation of scientific ethics and traditions in all the processes from the project phase to the conclusion of the thesis and that the works I have benefited are from those shown in the References. (03/08/2022)

Hasna EL HASNAOUY.

FOREWORD

I would like to express my heartfelt gratitude to my supervisor Assist. Prof. Dr. Necip Gökhan KASAPOĞLU, for his invaluable guidance, support, and encouragement throughout the course of my master's degree. His insights and expertise have been invaluable in helping me to complete my thesis.

Special thanks go to my beloved parents, who have always believed in me and supported my academic goals. Their love and tireless support throughout my studies. Their encouragement and belief have been a constant source of motivation and has helped me to overcome any challenges that I faced.

I am also deeply grateful to my friends, who have been there for me through thick and thin and provided a much-needed distraction from the stresses of graduate school. I am also grateful to Aydin University for providing me with the resources and facilities that I needed to complete my research and the academic staff for their help and support.

August 2023 Hasna EL HASNAOUY

AUTOMATIC TARGET RECOGNITION FOR SYNTHETIC APERTURE RADAR DATA

ABSTRACT

Automatic Target Recognition (ATR) in images generated from Synthetic Aperture Radar (SAR) has become a significant focus of research in contemporary society and represents a crucial avenue of inquiry within the realm of image processing. This study presents a thorough investigation of ATR techniques applied to SAR data. The widely used MSTAR dataset is utilized for evaluating the proposed methodologies. The initial stage of the study involves feature extraction techniques which aim to capture the relevant information from SAR data and reduce its dimensionality. The extracted features are used as inputs for various classifiers including Support Vector Machine (SVM). The performance of these classifiers is compared and evaluated based on their classification accuracy. To address the issue of speckle noise inherent in SAR imagery, mean and median filters are applied as preprocessing steps before feature extraction to investigate how noise reduction techniques affect the recognition accuracy of the ATR system.

The expected outcomes of this research are twofold. First, it aims to determine the most effective feature extraction method and classifier combination for SAR-based ATR tasks. Second, it intends to assess the influence of noise reduction techniques on classification performance, providing insights into the trade-off between noise reduction and classification efficiency. By comprehensively analyzing feature extraction methods, classifiers, and the impact of noise reduction techniques, this study contributes to advancing the field of ATR for SAR data. The findings will aid researchers and practitioners in selecting suitable methodologies for SAR-based target recognition, ultimately enhancing the capabilities of SAR systems in various applications.

Keywords: Automatic Target Recognition, Synthetic Aperture Radar, Feature Extraction, Support Vector Machine, Speckle Noise Reduction

YAPAY AÇIKLIKLI RADAR VERİLERİ İÇİN OTOMATİK HEDEF TANIMA

ÖZET

Yapay Açıklıklı Radar (SAR) görüntülerde Otomatik Hedef Tanıma (ATR), SAR görüntü işleme ve hedef belirleme alanlarında önemli bir araştırma konusudur. Bu çalışma da, SAR verilerine uygulanan ATR teknikleri üzerine kapsamlı bir araştırma yapılmıştır. Önerilen metodlar yaygın olarak kullanılan MSTAR veri seti üzerinde uygulanmıştır. Çalışmanın ilk aşaması, SAR verilerinden ilgili öznitelikleri çıkarmayı ve boyut indirgemeyi amaçlayan öznitelik çıkarma tekniklerini içerir. Çıkarılan öznitelikler, Destek Vektör Makinesi (SVM) dahil olmak üzere çeşitli sınıflandırıcılar için girdi olarak kullanılır. Bu sınıflandırıcıların performansı, sınıflandırma doğruluğuna göre karşılaştırılır ve değerlendirilir. SAR görüntülerinde bulunan benek gürültüsü etkisini ele almak, gürültü azaltma tekniklerinin ATR sisteminin tanıma performansını nasıl etkilediğini araştırmak için öznitelik çıkarımından önce ön işleme adımları olarak ortalama ve medyan filtreler uygulanmıştır.

Bu araştırmanın beklenen sonuçları iki yönlüdür. İlk olarak, SAR tabanlı ATR görevleri için en etkili öznitelik çıkarma yöntemini ve sınıflandırıcı kombinasyonunu belirlemektir. İkinci olarak, gürültü azaltma tekniklerinin sınıflandırma performansı üzerindeki etkisini değerlendirmeyi ve gürültü azaltma ile sınıflandırma verimliliği arasındaki başarım ilişkisini ortaya çıkarmaktır. Öznitelik çıkarma yöntemlerini, sınıflandırıcıları ve gürültü azaltma tekniklerinin etkisini kapsamlı bir şekilde analiz eden bu çalışma, SAR verileri için ATR uygulamasında seçilen yöntemler içerisinde en iyi performans ve öznitelik çıkarım metod çiftinin belirlenmesini sağlamıştır. Bulgular, araştırmacılara ve uygulayıcılara SAR tabanlı hedef tanıma için uygun metodların belirlenmesin de yardımcı olacaktır.

Anahtar Sözcükler: Otomatik Hedef Tanıma, Yapay Açıklıklı Radar, Öznitelik Çıkarma, Destek Vektör Makinesi, Benek Gürültüsü Azaltma.

TABLE OF CONTENTS

DECLARATION I

FOREWORD II

ÖZET III

ABSTRACT IV

TABLE OF CONTENTS V

LIST OF FIGURES... VIII

LIST OF TABLES... IX

I. INTRODUCTION..1

A. Synthetic Aperture Radar (SAR) 1

1. SAR Mechanism of Working..2

2. Scattering Mechanisms 3

3. SAR Azimuth resolution...4

4. Range resolution 6

5. Confusion Matrix 8

B. Automatic Target Recognition...11

C. MSTAR Dataset..12

D. Literature Review..13

II. SPECKLE NOISE REDUCTION AND SAR DATA FEATURES EXTRACTION 15

A. Speckle Noise Reduction 15

1. Introduction..15

2. Speckle Pre-filtering 16

3. Mean Filter 17

4. Median Filter 17

B. SAR Data Features Extraction 19

1. Introduction 19

2. Principle Component Analysis (PCA) 20

3. Kernel Principal Component Analysis (KPCA) 22

4. Independent Component Analysis (ICA) 26

III. SAR DATA CLASSIFICATION 29

A. Introduction 29

B. K-Nearest Neighbors 31

C. Support Vector Machine 34

1. Linear support vector machine 35

2. Radial Basis Function (RBF) kernel 37

D. Random Forest 40

IV. DESIGNED EXPERIMENTS AND RESULTS 44

A. Data Base Used 44

B. Experiment Preprocessing Description 45

1. Reshaping Data 45

2. Normalizing Data 45

3. Mean-Centering Data 45

C. Classification of Reduced Number of Samples 46

1. Classification of Original data 46

2. Reduced Number of Data 46

D. Classification after Features Extraction 49

E. Effect of Speckle Noise Reduction on Classification 50

1. Classification Accuracy after Applying Mean Filter 50

2. Classification after Applying Median Filter 51

3. Training and Testing Confusion matrix 52

V. CONCLUSION AND FUTURE WORKS...56

REFERENCES 59

RESUME 66

LIST OF FIGURES

Figure 1. The synthesized antenna by the forward movement of the shuttle. 3

Figure 2. The scattering of radar signal by rough surface. 3

Figure 3. Reflection of radar signals from different types of surfaces. 4

Figure 4. The azimuth resolution of Synthetic aperture radar. 6

Figure 5. SAR system geometric model. 8

Figure 6. Confusion matrix actual and predicted results. 9

Figure 7. Precision vs Recall for a confusion matrix...10

Figure 8. Steps of the target recognition process...11

Figure 9. Standard MSTAR targets. (a) Visual image. (b) SAR image. 13

Figure 10. Mean filter applied to SAR image of 2S1 battle tank. 17

Figure 11. Median filter applied to SAR image of 2S1 battle tank. 18

Figure 12. Illustration of Principal Component Analysis concept. 21

Figure 13: The comparison on two data clouds that are Gaussian-based is made for two different neighbourhood sizes, K = 1 and K = 20.........................34

Figure 14: The potential solutions for the problem of finding a separating plane. The second solution is anticipated to result in better generalization compared to the first. 37

Figure 15: The graphical representation of the distribution of the RBF kernel......40

Figure 16: The random forest classifier processing diagram.................................41

LIST OF TABLES

Table 1: Details of the targets from MSTAR dataset used for the experiments.....44

Table 2: 1/3 Reduced number of samples of MSTAR ten target classes. 47

Table 3: 1/5 Reduced number of samples of MSTAR target classes. 48

Table 4: Training and testing accuracy for original and reduced features. 49

Table 5: Accuracy results for SAR data without noise reduction 50

Table 6: Accuracy results after reducing noise with Mean filter............................51

Table 7: Accuracy results after reducing noise with Median filter. 52

Table 8: Confusion table of true and predicted training classes of SVM-RBF classifier with PCA feature extraction and without speckle filtering. ...53

Table 9: Confusion table of true and predicted training classes of SVM-RBF classifier with PCA feature extraction and without speckle filtering. ...54

Table 10: Confusion table of true and predicted training classes of SVM-RBF classifier with PCA feature extraction after speckle filtering. 55

Table 11: Confusion table of true and predicted training classes of SVM-RBF classifier with PCA feature extraction after speckle filtering. 55

I. INTRODUCTION

A. Synthetic Aperture Radar (SAR)

Synthetic Aperture Radar (SAR) is a remote sensing technology that uses radar to create images of the Earth's surface. The technology was first developed in the 1950s, but it has undergone significant development and refinement over the years. One of the key innovations in SAR technology was the development of the synthetic aperture, which allowed for the creation of high-resolution images using smaller antennae. This was accomplished by using the motion of the antenna to effectively synthesize a larger aperture, allowing for higher resolution images to be produced.

SAR has been used for a wide range of applications, including mapping, surveillance, and environmental monitoring. It has also been used to study the Earth's oceans and ice caps, as well as to monitor the environment for natural disasters and other events. SAR can be used to detect targets by sending a radar pulse towards the region of interest (ROI) and analyzing the reflected signal. The reflected signal can be used to create a SAR image of the ROI, which can then be analyzed to detect the presence and location of targets within the image.

SAR technology has continued to evolve over the years, with new innovations and advances being made in areas such as imaging modes, data processing, and image analysis. Today, SAR is an important tool for a wide range of applications, and it is likely to continue to play a key role in remote sensing and imaging in the future.

One of the unique features of SAR is its ability to operate independently of weather and lighting conditions. This is because the radar signal is not affected by clouds or other obstacles, allowing it to produce images of the Earth's surface even in poor visibility conditions. Additionally, SAR has a high spatial resolution, meaning it can produce detailed images of objects or features on the ground. SAR can also be used to detect objects or features that are hidden or difficult to see with the naked eye, such as underground utilities or changes in the Earth's surface. This is because the radar signal can penetrate through obstacles, allowing it to reveal hidden features.

To produce high-resolution images, SAR uses a technique called synthetic aperture, which synthesizes a large antenna by taking advantage of the Doppler history of the radar echoes generated by the forward motion of the spacecraft. This allows for high azimuth resolution in the resulting image despite a physically small antenna. The process of generating a SAR image involves transmitting a pulse at each position as the radar moves and recording the return echoes in an echo store.

Overall, SAR is a powerful tool for a variety of applications due to its ability to operate in any weather, its high spatial resolution, and its ability to reveal hidden features.

1. SAR Mechanism of Working

Synthetic aperture radar (SAR) is a radar technology that creates high-resolution images of the Earth's surface by sending and receiving signals with an antenna. The system calculates the distance to an object by measuring the time it takes for a transmitted signal to bounce back after hitting the object. SAR is used for various purposes, including mapping, surveillance, and remote sensing. The synthesis of a substantial antenna is made possible by utilizing the Doppler effect of radar echoes caused by the spacecraft's motion. This approach enables the resulting image to have a high level of detail in terms of horizontal resolution, even though the physical antenna is small. During the radar's movement, pulses are transmitted and the echoes they generate are stored in a designated location.

Synthetic Aperture Radar necessitates a sophisticated assortment of navigational and control systems onboard the spacecraft. These systems rely on both Doppler and inertial navigation equipment to ensure accurate positioning. This holds true for sensors like ERS-1/2 SAR and ENVISAT ASAR, which maintain an orbit at a distance of roughly 900km from Earth., a single transmitted pulse covers an area on the ground approximately 5km long in the along-track (azimuth) direction.

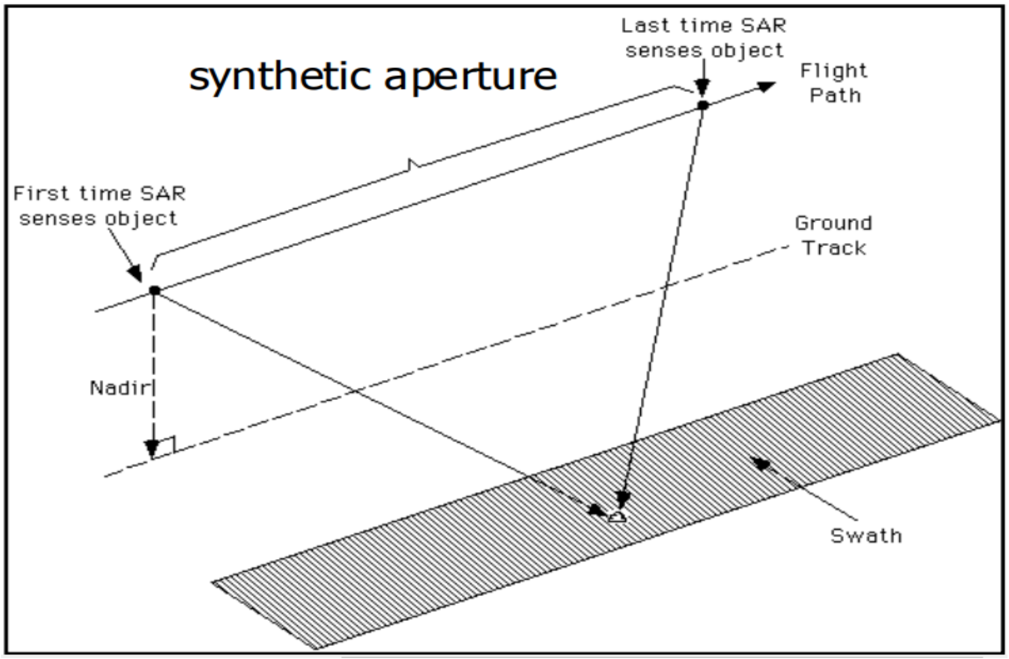


Figure 1. The synthesized antenna by the forward movement of the shuttle.

2. Scattering Mechanisms

In SAR images, the intensity of the backscattered radar signal is depicted, with darker areas indicating low backscatter and brighter areas indicating high backscatter. Low backscatter implies that little of the radar energy was reflected, while high backscatter means that a significant portion of the radar energy was reflected.

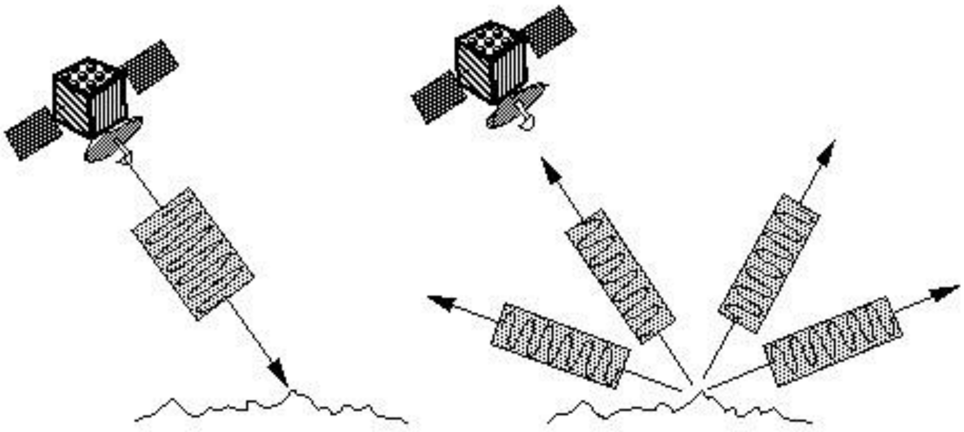


Figure 2. The scattering of radar signal by rough surface.

The amount of radar energy reflected to the sensor from a specific target area can be affected by various factors, including the size and electrical properties of the objects within the area, as well as the moisture content. Wetter objects tend to appear brighter in SAR images, while drier objects appear darker.

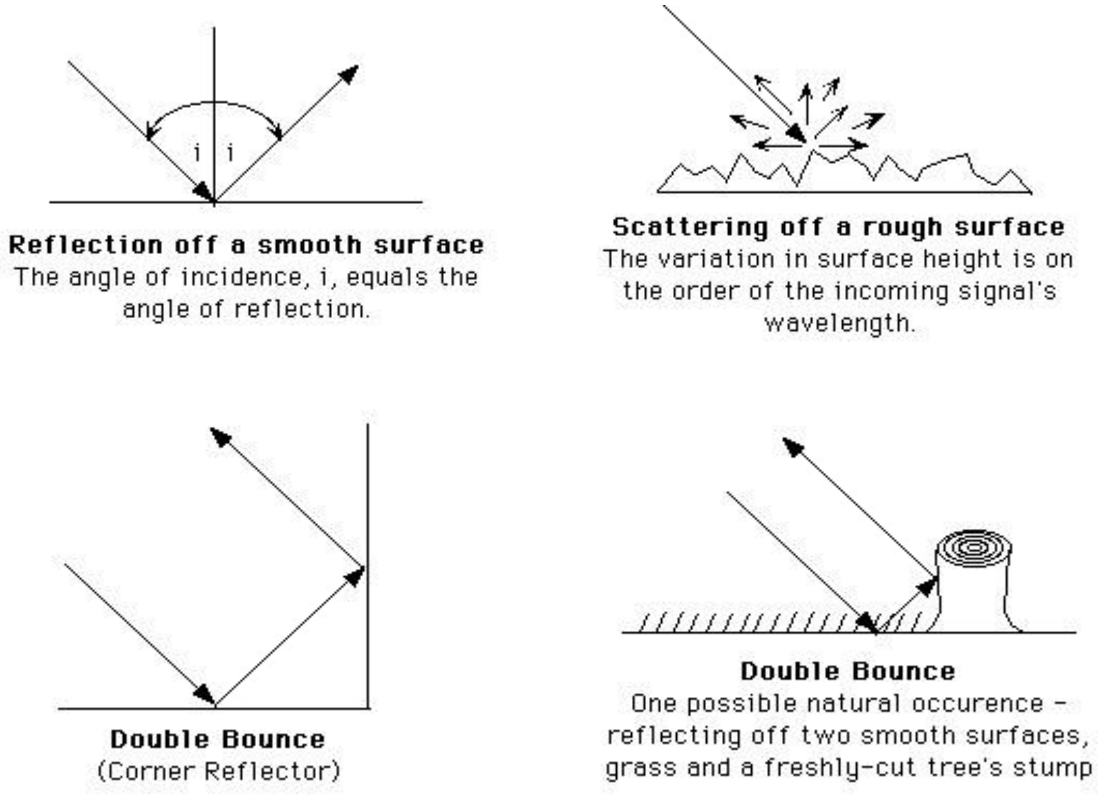

i
i
Reflection off a smooth surface
The angle of incidence, i, equals the angle of reflection.
Scattering off a rough surface
The variation in surface height is on the order of the incoming signal's wavelength.
Double Bounce
(Corner Reflector)
Double Bounce
One possible natural occurence – reflecting off two smooth surfaces, grass and a freshly-cut tree's stump

Figure 3. Reflection of radar signals from different types of surfaces.

However, smooth bodies of water, which act as flat surfaces and reflect the approaching beats absent from the sensor, will show up dark. In addition, the characteristics of the SAR pulses, such as the wavelength and polarization, and the observation angles can also influence the backscatter.

3. SAR Azimuth resolution

SAR utilizes a synthetic aperture technique to enhance the resolution of its antenna by mimicking the effect of a larger antenna. This is achieved by processing the signals and stages gotten from moving targets with a little radio wire in a complex way. This synthetic aperture method allows SAR to produce a beam with the same width as a radar of the same length within the azimuth course. The azimuth determination is at that point improved to that of a radar with half the opening estimate. This is done using pulse compression, similar to the technique used for the range direction.

- $\beta = \lambda / D$ The real beam width
- $Ls = \beta. R = \Delta L$ The Synthetic aperture length represents the real resolution
- $\beta s = D / (2. R) \quad \lambda / 2. Ls$ The Synthetic beam width
- $\Delta Ls = D / 2 = \beta s. R$ The Synthetic resolution

With, the wavelength (λ), the radar aperture (D), and the distance between the antenna and the object (R) are factors considered in the equation.

For conventional radar from the antenna's center, Azimuth resolution could be given as follow:

$$\delta \mathrm{y} = \frac{\lambda_c \mathrm{r}_c}{yl} \tag{1.1}$$

where:

δ_{y}: Azimuth resolution

λ_{c}: Wavelength of the radar signal

r_{c}: Cross-range cell size

y_{l}: Antenna length in the cross-range direction

Synthetic aperture technique enables the creation of an effect equivalent to use a large antenna by combining gotten signals and stages from moving targets with a little receiving wire, resulting in a synthetic aperture length that can project a beam with range width in the azimuth direction. This allows SAR to have an azimuth resolution that is half that of real aperture radar, even at great distances or high altitudes.

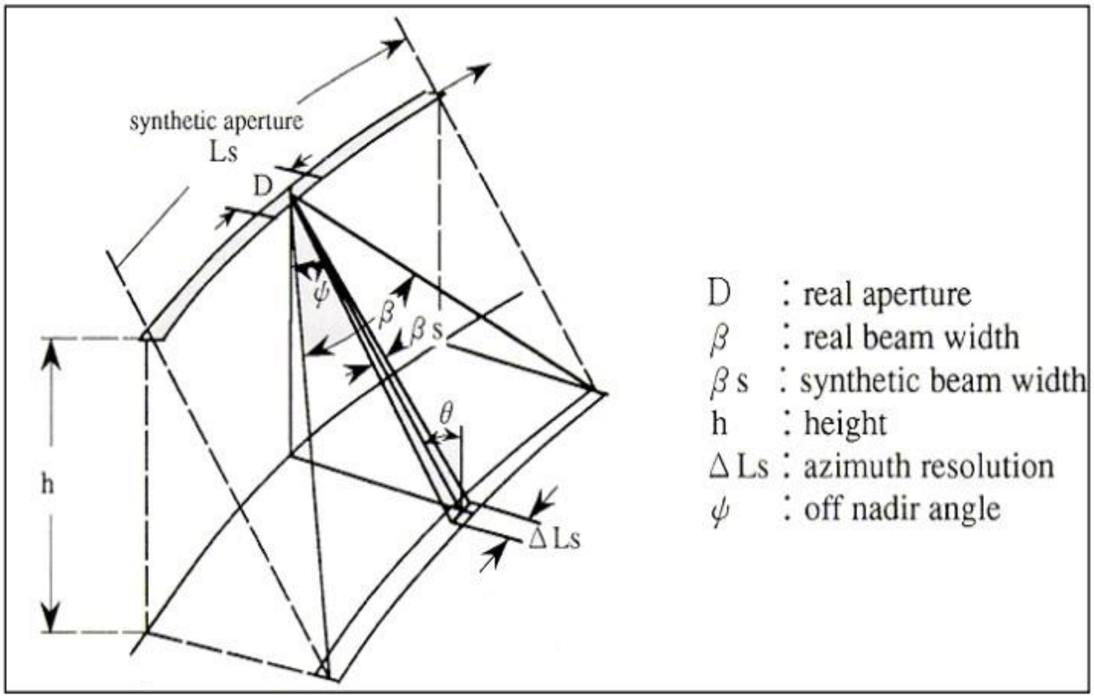


Figure 4. The azimuth resolution of Synthetic aperture radar.

4. Range resolution

The range resolution of a synthetic aperture radar (SAR) system is a measure of the ability of the system to distinguish between two objects or features that are located at different ranges (distances) from the antenna. It is typically expressed in meters or centimeters and is contrarily relative to the transfer speed of the radar signal. The range resolution is determined by several factors, including the width of the transmitted pulse, the characteristics and size of the target being imaged, and the efficiency of the receiver.

Range resolution can be calculated using the following formula:

$$\delta \mathrm{r} = \frac{\mathrm{C}^0}{2\tau} \qquad (1.2)$$

δr: Range resolution

C_0: Speed of light

τ: Bandwidth (the width of the radar signal in Hz).

Range resolution is an important parameter in SAR imaging because it determines the level of detail that can be resolved in the image. A system with a high range resolution will be able to recognize between objects or highlights that are found near to each other, while a system with a low range resolution will not be able to distinguish between such objects or features. Range resolution is also affected by other factors such as the pulse duration and the pulse repetition frequency of the radar signal. In general, systems with shorter pulse durations and higher pulse repetition frequencies will have higher range resolutions. It is an important property of SAR systems and is used in a variety of applications including remote sensing, mapping, and surveillance. It is also an important factor to consider when selecting a SAR system for a specific application.

Ground range resolution is a measure of the variation in ground range as it relates to slant range, and is represented by:

$$\delta\mathrm{x} = \frac{\delta\mathrm{r}}{\sin\theta} \tag{1.3}$$

Where the incidents angle is represented by θ.

In a SAR system, the range resolution is determined by the width of the transmitted pulse, the characteristics and size of the target, and the performance of the receiver. A geometric model for SAR systems involves the space separating the antenna from the ground pixel is referred to as slant range, the distance between the ground track and the ground pixel is known as ground range as we can see in the figure 3.

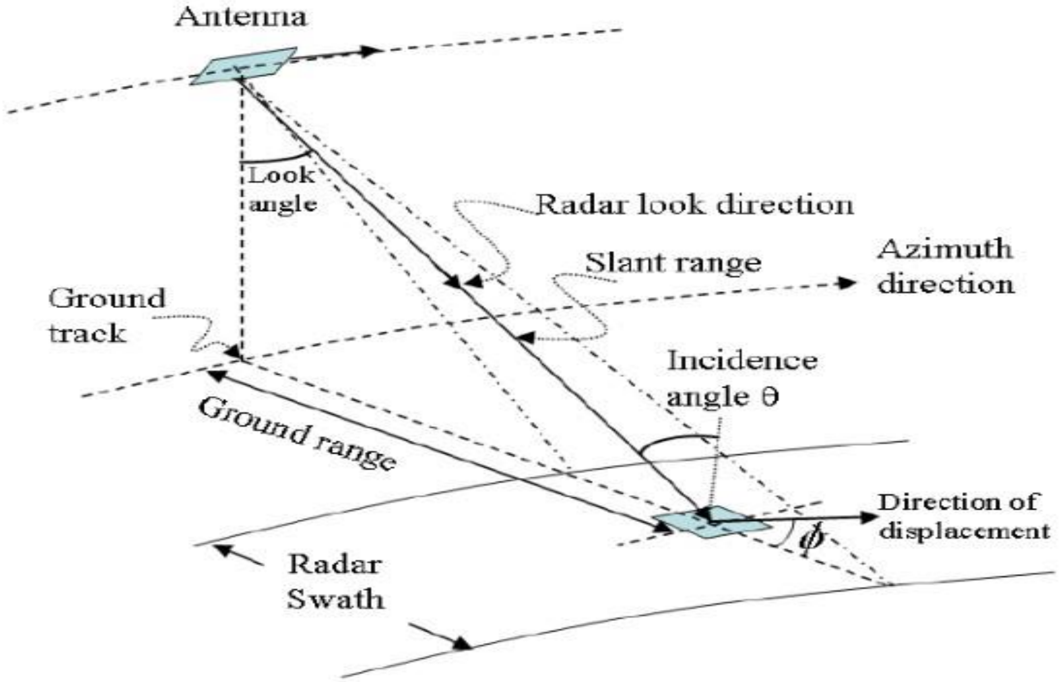


Figure 5. SAR system geometric model.

5. Confusion Matrix

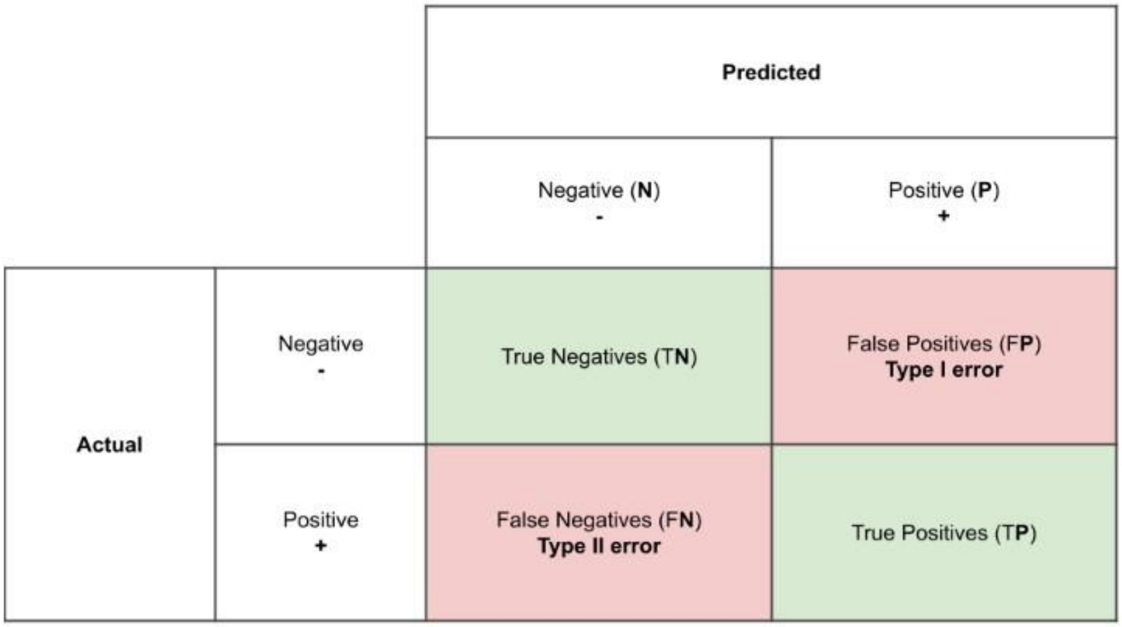
The study conducts multiple classification methods using various machine learning algorithms. A confusion matrix is created to showcase the performance of each model when applied to the given datasets. The confusion matrix stores the actual and predicted results of each class in the datasets. It is also known as an error matrix.

The matrix is usually used to describe the performance of a binary classifier, although it can also be used for multiclass classification. It produces four primary outcomes: true positives, true negatives, false positives, and false negatives.

- True positives (TP) are the cases when the classifier predicted the positive class, and it was correct.
- True negatives (TN) are the cases when the classifier predicted the negative class, and it was correct.
- False positives (FP) are the cases when the classifier predicted the positive class, but it was wrong.

- False negatives (FN) are the cases when the classifier predicted the negative class, but it was wrong.

The usage of the confusion matrix enables the calculation of different measures of performance such as the accuracy score, precision, recall, specificity, etc. These metrics can help in understanding the trade-offs between different classification algorithms and help in choosing the best one for a specific problem.



		Predicted	
		Negative (**N**) -	Positive (**P**) +
Actual	Negative -	True Negatives (T**N**)	False Positives (F**P**) **Type I error**
	Positive +	False Negatives (F**N**) **Type II error**	True Positives (T**P**)

Figure 6. Confusion matrix actual and predicted results.

The outcomes of the confusion matrix were used to compute four metrics to evaluate the performance of each model:

- Accuracy: it pertains to the quantity of samples that the classifier has correctly categorized.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1.4}$$

- Precision: is a metric that helps to understand the ability of a classifier to not classify a negative sample as positive. One way to determine this is by dividing the number of accurately predicted positive observations by the overall number of predicted positive observations.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1.5}$$

- Recall: describes how many of the actual positive instances are being predicted as positive by the classifier, it helps to understand the ability of a classifier to find all the positive instances.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{1.6}$$

- F1-Score: It is a way to balance the trade-off between accuracy and completeness of information. If precision and recall are both high, then there will be a high F1-score. One can achieve a maximum F1-score of 1, and the lowest possible value is 0. A classifier with a high F1-score is considered to have a good balance between precision and recall.

$$F1 - Score = \frac{\mathrm{2x(PrecisionxRecall)}}{\mathrm{Precision} + \mathrm{Recall}} \tag{1.7}$$

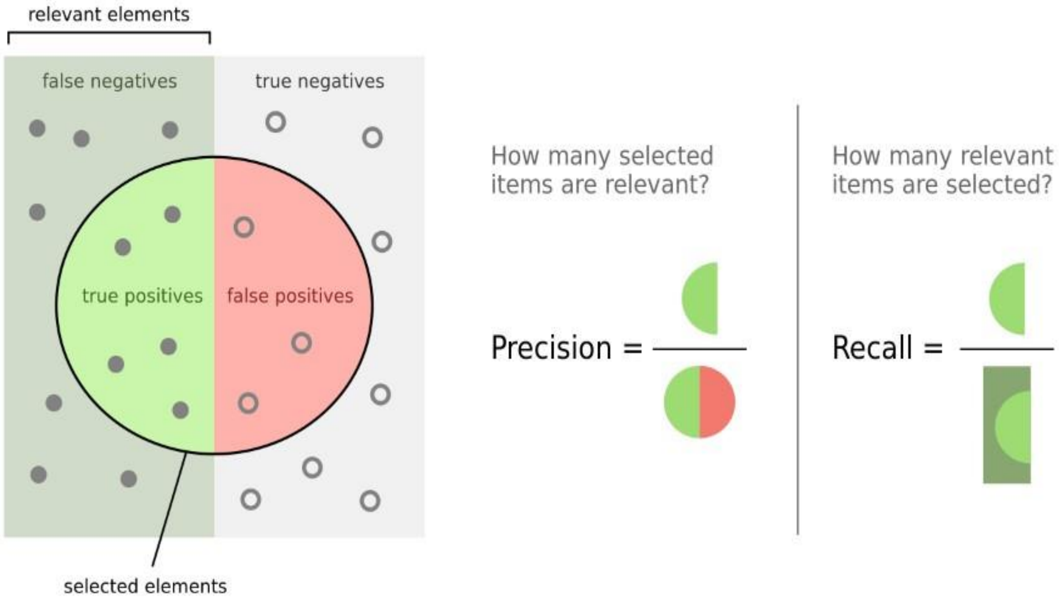


Figure 7. Precision vs Recall for a confusion matrix.

The confusion matrix is beneficial as it allows us to identify the successes and failures of the model, particularly when the labels are unbalanced. In simpler terms, it allows us to gain more insight than just the model's overall accuracy.

B. Automatic Target Recognition

The identification and detection of targets in synthetic aperture radar (SAR) images has become more widespread in the last years, but it remains a difficult task because of the significant amount of noise present in these images. Processing the automatic recognition of target in SAR images involves two stages: first, external factors that may cause false alarms (such as trees, cars, or buildings) must be removed from the images, and then feature extraction and classification algorithms are applied.

Recently, the field of identifying and locating targets in SAR images has become a topic of interest. However, the high level of noise in these images can make this task challenging. The process of Automatic Target Recognition (ATR) on SAR images involves two steps. The first step is to eliminate external factors such as trees, buildings, and cars that can lead to false alarms in the images. During the second phase, the application of feature extraction and classification algorithms takes place. A variety of methods, including those from computer vision and those specifically designed for SAR data, have been implemented to enhance the performance of target recognition using SAR data.

The process of identifying and understanding targets in SAR images can be broken down into three stages: identifying the presence of a target, determining the category of the target, and identifying specific instances of the same category of target. The first stage, target identification, simply determines the difference between targets in the image. The second stage, target categorization, uses this initial identification to predict the class of the target. The final stage, target recognition, uses the information from the previous two stages to confirm the specific identity of a target within a class. It is important to note that the term "target recognition" is often used specifically to refer to this final stage, the highest level of understanding a target in an SAR image.

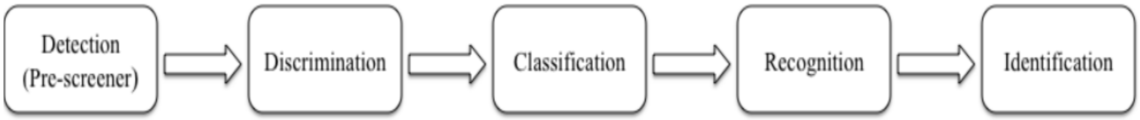


Figure 8. Steps of the target recognition process.

In summary, ATR using SAR data is a process that involves two main stages: first, removing external factors that may cause false alarms, and second, using feature extraction and classification algorithms to accurately identify targets in the SAR images. This process can be further divided into three levels: discrimination, classification, and recognition. These methods are evaluated on two sets of target configurations: standard and extended operation conditions. The standard conditions typically include ten classes of targets, while the extended conditions involve variations in acquisition geometry, target state, and local target deployment, as well as intraclass variability.

C. MSTAR Dataset

The Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset is employed to assess the performance of algorithms for identifying targets automatically using synthetic aperture radar imagery. It consists of images of military vehicles captured from different angles, as well as clutter and noise. The MSTAR dataset is extensively utilized in studies on automatic target recognition methods. and has been used in multiple benchmark studies. It is available for download from the Defense Technical Information Center (DTIC) website.

This dataset is a collection of synthetic aperture radar images that are used to evaluate algorithms based on automatic target recognition. These images contain military vehicles taken from different angles, as well as clutter and noise. The dataset is widely used in research on ATR algorithms and has been used in multiple benchmark studies. The MSTAR dataset is publicly available and can be downloaded from the Defense Technical Information Center (DTIC) website. The MSTAR dataset comprises of X-band Synthetic Aperture Radar images that have a resolution of 1x1 ft, can rotate 360-degrees with an interval of 1 degree and has an image dimension of 128x128 pixels. The targets frequently found in the MSTAR dataset are illustrated in Figure 2

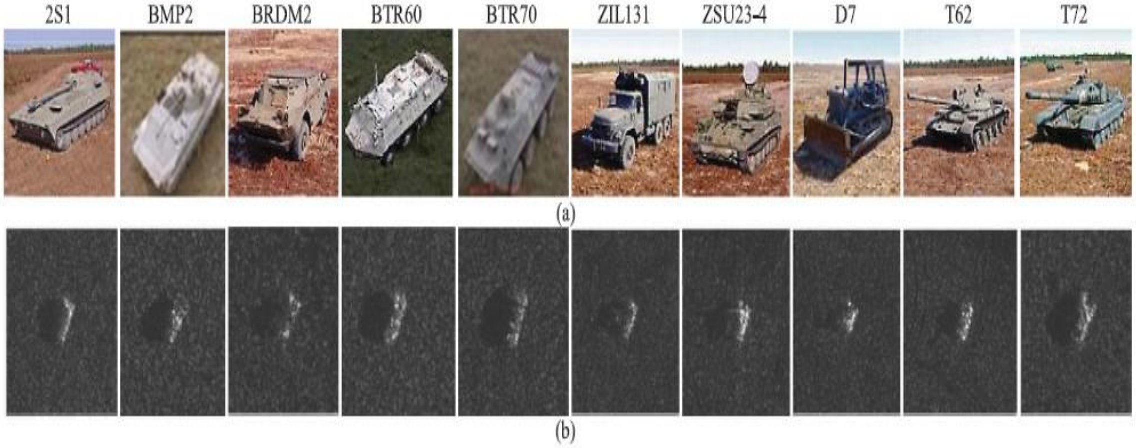


Figure 9. Standard MSTAR targets. (a) Visual image. (b) SAR image.

D. Literature Review

Template matching, a traditional method, has been found to be inadequate for detecting targets in Synthetic Aperture Radar (SAR) images. The main issue is that noise present in SAR images can alter the appearance of targets, making it difficult to identify them. Researchers have attempted to tackle this problem by utilizing different techniques such as obtaining local and global features, using sparse representations, and implementing canonical correlation analysis of sparse matrices to fuse features at different scales. These fused features are then used for target recognition. Liu et al. 2013 used SAR images to extract features, which were evaluated using multiple classifiers and then combined for improved classification results.

The Automatic Target Recognition (ATR) of SAR images is a persistent challenge. Obtaining raw SAR images is a significant difficulty and the images themselves often contain high levels of noise. The MSTAR dataset is commonly employed by researchers for conducting ATR tasks involving the recognition and acquisition of both still and moving targets, which was introduced by Liu and Li in 2013. The MSTAR dataset has been used in several studies, with varying levels of accuracy. Novak et al. 1998 achieved 66.2% and 77.4% accuracy for 20-class and 10-class classification respectively. Martone et al. 2009 applied k-means clustering to identify objects in motion within areas covered in trees. Gorovyi and Sharapov achieved 90.7% accuracy on the MSTAR dataset using SVM. Techniques such as speckle noise reduction and feature extraction are commonly used in these studies.

In this study, using Python coding program, different features extractions were used on ten classes of data provided by the benchmarking dataset MSTAR, which has 96x96 pixel size on different categories for training then testing a classification model. First, we apply different preprocessing algorithms to our data then we extract the most important feature using different types of features extraction before we train our model using different classifiers. Moreover, the reduction of speckle noise is done by applying a mean and median filter. Finally, a comparison between different classifiers.

II. SPECKLE NOISE REDUCTION AND SAR DATA FEATURES EXTRACTION

A. Speckle Noise Reduction

1. Introduction

When a radar beam hits a surface that has small variations in its texture on the scale of the radar wavelength, to identify objects in motion within areas covered in trees. The signal that is reflected consists of various waves that originate from several scatterers present in a particular resolution cell. Because of the roughness of the surface, the distance between the scatterers and the receiver changes, which means that the waves are no longer in phase with each other. If the waves combine constructively, the signal is strong, but if they combine destructively, the signal is weak. The production of a SAR image is achieved through the consolidation of radar signals from numerous pulses. This results in variations in the intensity of each pixel, called speckle, which can appear as a granular pattern. This can make it difficult to extract information from SAR images, especially in areas that are heterogeneous. Despite this, speckles are still considered to be noise in most image processing applications.

Speckle noise is a common issue in synthetic aperture radar (SAR) imagery, which could pose a challenge in accurately identifying and categorizing items. It is caused by the interference of multiple reflections of the radar signal of objects in the image. Thus, the overall appearance is granular due to the collective contribution of numerous elementary scatterers within each resolution cell. Speckle noise can significantly degrade the quality of SAR images, making it difficult to distinguish delicate details and features.

There are several methods that have been developed to reduce speckle noise, including conventional techniques like mean and median filtering and adaptive techniques such as the Lee, Kuan, and Frost filters. Generally, these techniques employ a specified filter window to approximate the regional noise variation in a speckled picture and employ a distinct filtering course. While these methods can effectively reduce noise in homogenous areas, they can also lead to over-smoothing and loss of detail in areas consisting of diverse elements. The Lee filter is a frequently used point of comparison. due to its ability to effectively reduce noise while maintaining image sharpness. Wavelet-based denoising techniques have been shown to be effective in reducing speckle noise in SAR images. These algorithms involve transforming the SAR image using a logarithmic transformation and then applying shrinkage to the wavelet coefficients. Some studies have considered combining the use of wavelet shrinkage denoising methods with an edge detector and showed an effectiveness reducing speckle noise while preserving small details and edges in the SAR image.

2. Speckle Pre-filtering

Two types of methods are used to reduce or eliminate speckle noise in images. The first method is to take multiple images or "looks" by combining various polarization states and sections of the azimuth spectral bandwidth, it is possible to create an averaged version which is known as multiple-look processing. The second method involves using digital image processing techniques to smooth the image after it has been formed. There are two main techniques of digital filtering in the frequency domain, such as the Wiener filter or wavelet transformation.

In the realm of image enhancement, the second technique takes a captivating journey through the spatial domain. Here, the ethereal task of noise elimination is accomplished through the artistry of averaging or the alchemy of statistically molding the values of neighboring pixels. A tapestry of contrasting approaches is skillfully woven by the venerable Hervet E., Fjørtfort, R., Marthon, P., and A. Lopes, offering a delightful juxtaposition between the frequency domain's enchanting wavelet-based filters and the spatial domain's majestic statistical filters. Our study embarks on the captivating path of the second approach, as we delve into the splendid realm of digital filters reigning over the spatial domain, their noble purpose being to quell the unruly speckles adorning the realm of SAR imagery.

3. Mean Filter

Mean filter is a type of low-pass filter that is used to smooth an image where the pixel values can be substituted with a computed average value derived from the surrounding pixel values. It is applied by convolving the image with a kernel that has all its values set to 1 and dividing the sum by the total number of elements in the kernel. The size of the kernel determines the amount of smoothing applied to the image.

To reduce speckle noise in SAR data using mean filter, the mean filter is applied to the SAR image using a kernel of a suitable size. The kernel is moved over the image, and for each pixel, the average value of the neighbouring pixels is calculated and replaced with the original pixel value. The aforementioned procedure is replicated iteratively across each individual pixel within the image, thereby producing a smoothed image which exhibits a diminished amount of speckle noise.

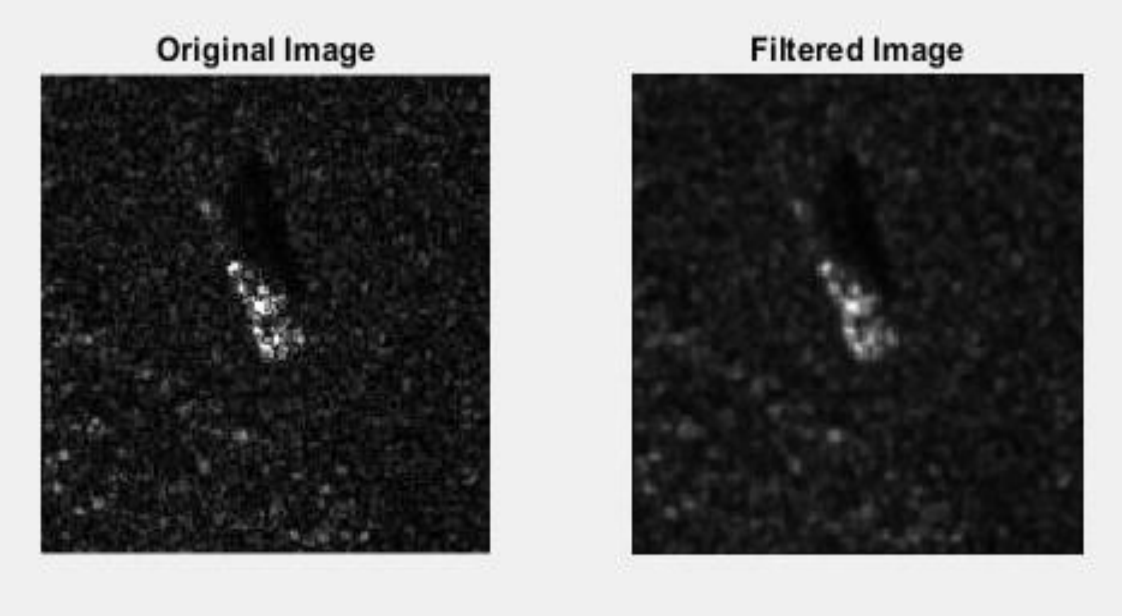


Figure 10. Mean filter applied to SAR image of 2S1 battle tank.

4. Median Filter

The median filter is a prevalent non-linear image filtering methodology utilized for noise removal in images, including those obtained from Synthetic Aperture Radar (SAR) systems. The underlying principle of the median filter involves substitution of the value of a designated pixel in an image with the median value computed from the pixel set within a defined neighbourhood around that pixel. The neighbourhood is typically defined as a square window of a certain size, and the median filter is applied to each pixel in the image by sliding this window over the entire image.

The main advantage of the median filter is its ability to effectively remove impulse noise, also known as "salt and pepper" noise, which is characterized by the presence of random pixels with very high or very low values. This type of noise is common in SAR images due to the presence of outliers, such as ground clutter or other types of interference.

The median filter, along with its various adaptations, operates by substituting the central pixel of a window that is sliding through an image with the median intensity of all the pixels located within that window. The filter in question exhibits efficacy in eliminating impulse or short-duration noise; nevertheless, it is not optimally designed for attenuating speckle noise. Common issues with this filter include blurring of edges, erasure of thin linear features, and distortion of object shapes.

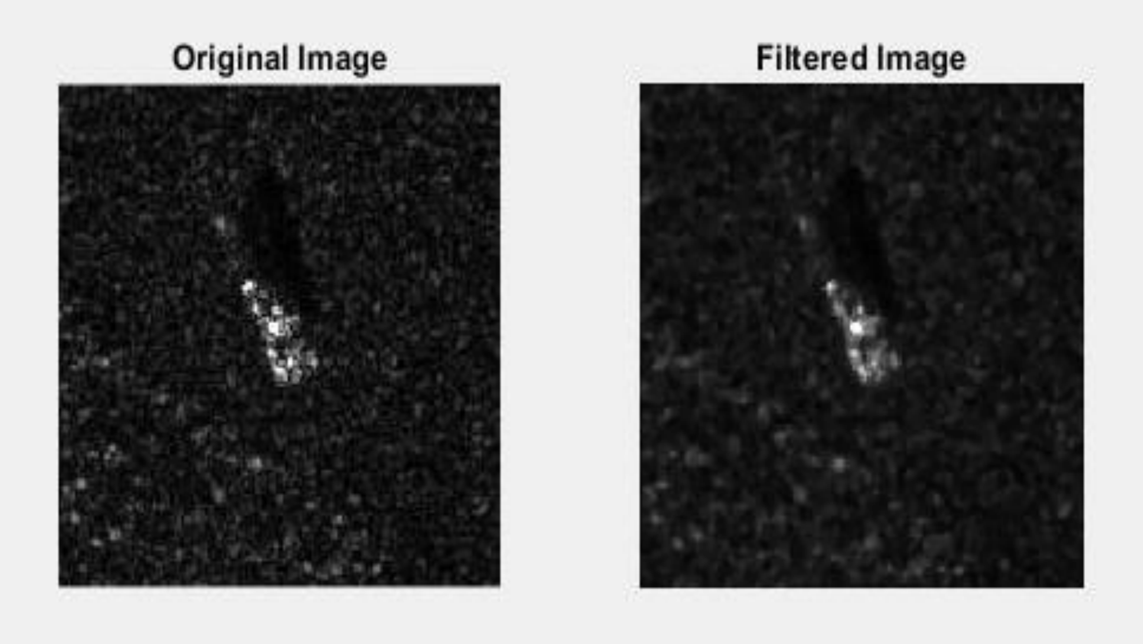


Figure 11. Median filter applied to SAR image of 2S1 battle tank.

B. SAR Data Features Extraction

1. Introduction

Feature extraction is the procedure of recognizing and obtaining meaningful information from raw data, such as images or signals. This information can be used for various tasks such as pattern recognition, classification, and anomaly detection. In the field of synthetic aperture radar (SAR) data, feature extraction is used to identify and extract specific features of interest, such as buildings or vehicles, from the radar images.

The history of feature extraction dates back to the early days of image processing and computer vision, with early techniques such as edge detection and texture analysis. With the advent of machine learning and deep learning, feature extraction has become more sophisticated and powerful, with methods such as convolutional neural networks (CNNs) and autoencoders being used to extract high-level features from complex data. The importance of feature extraction for SAR data lies in its ability to extract relevant information from the data, making it easier for downstream tasks such as classification and target detection. This can improve the accuracy and efficiency of these tasks, as well as enabling the use of SAR data in new applications, such as change detection and urban planning. Additionally, feature extraction can serve as a technique to lower the dimensionality of data, thereby rendering it more manageable for storage and analysis. In many cases, the raw input data is high-dimensional and contains redundant or irrelevant information. For example, in image classification tasks, the raw pixel values of an image are high-dimensional and contain information about the lighting conditions, camera noise, and other factors that are irrelevant to the classification task. Therefore, feature extraction techniques are used to transform the raw input data into a lower-dimensional feature space that captures the essential characteristics of the data. A diverse range of techniques can be employed for executing feature extraction, including statistical methods, transform-based methods, and deep learning methods. Frequently utilized in various fields, statistical techniques such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) hold prominence to reduce the dimensionality of the data and extract representative features. Transform-based methods such as discrete wavelet transform (DWT) and Fourier transform (FT) are used to decompose the input data into a set of frequency components that capture the essential characteristics of the data. The use of deep learning methodologies, specifically convolutional neural networks (CNNs), is a widely adopted strategy for extracting features in image classification endeavours. These methods use a hierarchy of convolutional layers to learn increasingly complex features from the input data. The features learned by these networks are often more discriminative and representative than the features extracted by traditional feature extraction methods.

Before classifying SAR data, the process of feature extraction is of critical importance in data preparation for the classification task. The goal of feature extraction is to extract relevant information from the data that can be used to distinguish between different classes. This process can involve extracting various types of features, such as texture, shape, and intensity, that can be used to represent the data. Once the relevant features have been extracted, they can be used as input to a classifier, which is a model that assigns a label or class to each input. The classifier is trained on a set of labelled data, where the inputs are the extracted features, and the outputs are the class labels. During the training process, the classifier learns to associate specific feature patterns with specific class labels. When the classifier is used to classify new data, it takes the extracted features as input and assigns a class label based on the patterns it learned during training. By extracting relevant features before classification, the classifier is able to make more accurate and efficient predictions.

2. Principle Component Analysis (PCA)

Principal Component Analysis is a method created by Karl Pearson in 1901, aims to reduce the number of variables by identifying a set of uncorrelated factors known as Principal Components from a larger set of data that are linear combinations of the original variables. It is a technique for feature extraction in which the original data is transformed into a new set of variables. These new variables are uncorrelated and ordered by the amount of variance they explain in the data. PCA can be used for dimensionality reduction, noise reduction, and data visualization. Figure 7 illustrates the concept of Principal Component Analysis.

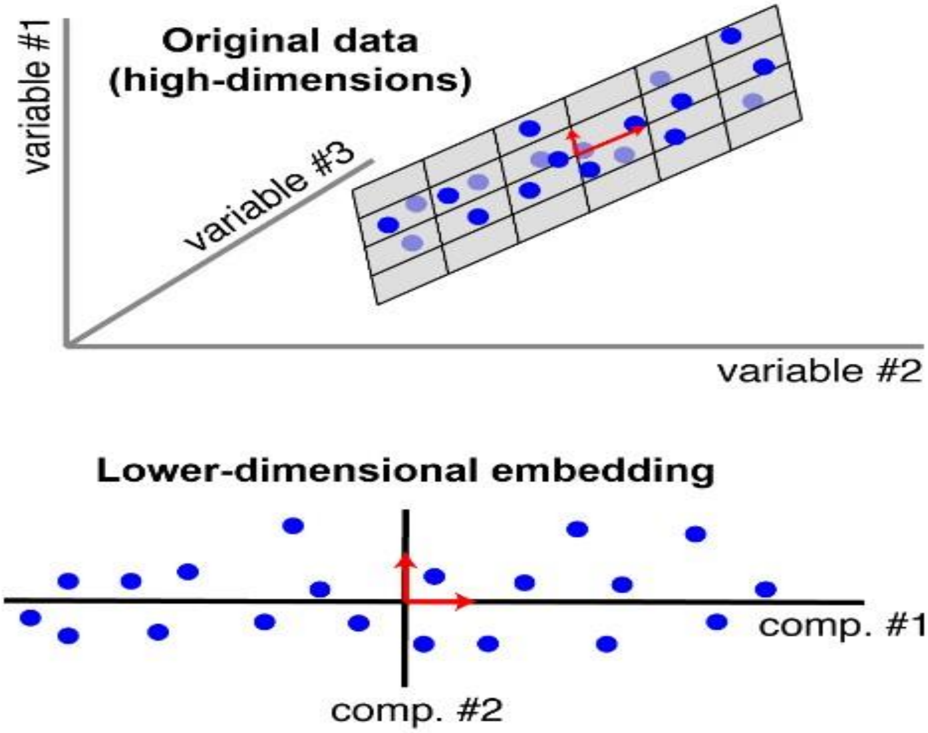


Figure 12. Illustration of Principal Component Analysis concept.

In the context of synthetic aperture radar (SAR) data classification, PCA can be used as a pre-processing step to extract relevant features from the radar data. This can help improve the classification performance by reducing the dimensionality of the data and removing noise and redundancy. The principal components can also be used as input features to a classifier. However, it's important to note that PCA is categorized as an unsupervised method that doesn't guarantee the identification of the most distinctive characteristics for classification purposes. The utilization of PCA in various fields, such as social science and space science, has been utilized for the goal of reducing data size and identifying key features. While it has been utilized in SAR image classification by remote sensing experts, it has not been commonly employed in the field of automatic target recognition. In this study, PCA is implemented in the context of SAR-ATR using the MSTAR data set.

PCA is a technique that is used to transform highly correlated image bands in SAR data into a set of linearly uncorrelated variables, which can then be used to extract meaningful information from the image bands. Let x be the pixel vector of the image, which can be represented as $X_n = [x_{n1}\, x_{n2} \dots\, x_{nF}]^T$ with each pixel location in the hypercube having S pixel values x_1, x_2, …, x_S. Here, n represents the n^{th} pixel number from S. The hypercube is denoted by D, with a size of $F \times S$, where $S = X \times Y$. To calculate the mean vector, M, of all the image vectors, we use the equation:

$$M = \frac{1}{S}\sum_{n=1}^{S} X_n. \tag{2.1}$$

To calculate an approximation of the covariance matrix we use the following equation:

$$C = \frac{1}{S} II^T. \tag{2.2}$$

Where the variable I represents a zero-mean image, which is obtained by subtracting the mean vector M from the pixel vector of the SAR image. Specifically, it is represented as $I = [I_1\ I_2 \ldots\ I_n]$, where $I_n = \mathrm{x}_n - M$ = $[x_{n1}\ x_{n2} \ldots\ x_{nF}]^T$. Next, to perform eigenvalue decomposition, the covariance matrix C is calculated, which is given by:

$$C = VEV^T. \tag{2.3}$$

In this process, the matrix V is formed using the F-dimensional eigenvectors $V_1, V_2, \ldots, V_F$ to create an orthonormal matrix. Similarly, the matrix E, which is a diagonal matrix, is formed by arranging the eigenvalues $E_1, E_2, \ldots, E_F$ of the covariance matrix C. These eigenvectors are also called principal components (PCs). To create a new feature subspace, k eigenvectors are selected, resulting in an F × k dimensional matrix called W, where k is less than or equal to F and is often much smaller than F. Numerous methodologies exist for selecting k eigenvectors, including sorting them in a descending order and selecting the top k principal components, engaging in divergence or discriminant analysis, among other approaches. The resultant vector Y, comprising Principal Component Analysis transformed pixel values, can be attained through the multiplication of the transposed matrix W and the initial pixel vector I as follow:

$$Y = W^T \times \mathrm{I}. \tag{2,4}$$

3. Kernel Principal Component Analysis (KPCA)

Kernel Principal Component Analysis (KPCA) is a prominent method utilized for the purpose of feature extraction within various domains of research, as for Synthetic Aperture Radar (SAR) data classification. SAR data is typically high-dimensional and non-linear, making it challenging to classify using traditional methods. KPCA can be used to transform the high-dimensional SAR data into a lower-dimensional space while preserving important non-linear relationships between the data points. In KPCA feature extraction for SAR data classification, the SAR data undergoes an initial process of mapping into a feature space of high-dimensionality, employing a non-linear transformation, and then the principal components of the transformed data are computed in this space. These principal components can then be used as features for classifying the SAR data.

The importance of KPCA feature extraction for SAR data classification is:

- Improved classification accuracy: By transforming the high-dimensional and non-linear SAR data into a lower-dimensional space, KPCA can help in improving the accuracy of SAR data classification algorithms.
- Better feature representation: KPCA can provide a better representation of the SAR data compared to linear feature extraction methods, making it easier to identify patterns and relationships in the data.
- Improved computational efficiency: KPCA can reduce the dimensionality of the SAR data, which can result in improved computational efficiency for classification algorithms.

Kernel methods are a type of machine learning method that that can address nonlinearity in data. These techniques facilitate the analysis of data by projecting it onto multidimensional spaces, which enhance the efficacy of linear procedures. By doing so, Kernel methods have the capacity to identify non-linear patterns present within the given data while still maintaining the computational simplicity of matrix algebra. In the context of process monitoring, kernel learning is primarily used in the feature extraction step of analysing plant data. Linear dimensionality reduction is the only option provided by standard PCA, which may not be effective if the data has complex structures that cannot be accurately represented within a linear subspace. Fortunately kernel PCA offers a solution by allowing for nonlinear dimensionality reduction. This generalisation of standard PCA enables the analysis of more complex data strictures.

Assuming the presence of a non-linear transformation function, denoted as *$\phi(x)$*, which facilitates the mapping of data points within a feature space possessing *D* dimensions, with *M* normally surpassing *D*. By performing a projection of individual data points x_i onto a designated point *$\phi(x_i)$*, it becomes feasible to apply standard PCA in the newly derived feature space. Nevertheless, this methodology has been observed to be computationally demanding and ineffective. In order to mitigate the aforementioned concern, kernel methods can be employed as a means of optimizing computational efficiency.

Initially, it is postulated that the newly acquired traits have been obtained through projection have a mean value of zero:

$$\frac{1}{N}\sum_{i=1}^{N}\Phi(x_i) = 0. \tag{2,5}$$

We compute the covariance matrix as an M x M matrix of the projected features as follow:

$$C = \frac{1}{N}\sum_{i=1}^{N}\Phi(x_i)\Phi(x_i)^T. \tag{2,6}$$

Herein, the covariance matrix's eigenvalues and eigenvectors are disclosed:

$$Cv_k = \lambda\, v_k \tag{2,7}$$

Where *k 1, 2, 3…, M.* from the equations of covariance (2,6) and eigenvectors (2,7) we get:

$$\frac{1}{N}\sum_{i=1}^{N}\Phi(x_i)\{\Phi(x_i)^T v_k\} = \lambda_k\, v_k\,, \tag{2,8}$$

Which we can write as:

$$v_k = \sum_{i=1}^{N} a_{ki}\,\Phi(x_i). \tag{2,9}$$

By replacing the value of v_k in the equation (2,8) with the value obtained from equation (2,9) we get:

$$\frac{1}{N}\sum_{i=1}^{N}\Phi(x_i)\Phi(x_i)^T\sum_{i=1}^{N} a_{ki}\,\Phi(x_i) = \lambda_k\sum_{i=1}^{N} a_{ki}\,\Phi(x_i)\,. \tag{2,10}$$

The definition of the kernel function is given by:

$$k(x_i, x_j) = \phi(x_i)^T \phi(x_j) \qquad (2,11)$$

If we multiply by the transpose of $\phi(x_l)$ both sides of the equation (2,10) we will have:

$$\frac{1}{N}\sum_{i=1}^{N} k(x_l, x_i) \sum_{k=1}^{N} a_{kj} k(x_i, x_j) = \lambda_k \sum_{i=1}^{N} a_{kj} k(x_l, x_i). \qquad (2,12)$$

The matrix notation can be represented as follow:

$$K^2 a_k = \lambda_k N K a_k, \qquad (2,13)$$

Where

$$K_{i,j} = k(x_i, x_j) \qquad (2,14)$$

The vector a_k is a column vector of size N, with a_{kj} as its elements.

$$a_k = [a_{k1}\ a_{k2} \ldots a_{kN}]^T \qquad (2,15)$$

We can solve a_k by

$$K a_k = \lambda_k N a_k \qquad (2,16)$$

The concluded kernel principal components can be obtained using:

$$y_k(x) = \phi(x)^T v_k = \sum_{i=1}^{N} a_{ki} k(x, x_i) \qquad (2,17)$$

In case the projected dataset $\phi(x_i)$ doesn’t have a mean of zero, we can use the Gram matrix $\widetilde{K}$ instead of the kernel matrix K. This Gram matrix is obtained by:

$$\widetilde{K} = K - 1_N K - K 1_N + 1_N K 1_N \qquad (2,18)$$

The matrix 1N is an N × N matrix where each element is equal to $\frac{1}{N}$. The benefits of kernel methods lie in the fact that it obviates the requirement of explicitly calculating the transformation $\phi(x_i)$. Instead, we can construct the kernel matrix directly from the training dataset $\{x_i\}$. There are two widely used kernels, namely the polynomial kernel

$$k(x, y) = (x^T y)^d \qquad (2,19)$$

Or

$$k(x, y) = (x^T y + c)^d \qquad (2,20)$$

where the constant c is greater than 0, and the gaussian kernel is represented by:

$$k(x, y) = e^{-\frac{\|x-y\|^2}{2\,\sigma^2}} \tag{2,21}$$

where σ is parameter.

Kernel PCA dimensionality reduction can be performed in the following steps:

- Construct the kernel matrix K from the input dataset $\{x_i\}$ using the equation (2,14)
- Calculate the Gram matrix $\widetilde{K}$ using the equation (2,18) when the projected dataset $\{\phi(x_i)\}$ does not have zero mean.
- Use equation (2,16) to find the vectors a_i, with the substitution of K with $\widetilde{K}$.
- The principal components of the kernel, denoted as $y_k(x)$, can be computed through the utilization of equation (2,17).

4. Independent Component Analysis (ICA)

The utilization of Independent Component Analysis (ICA) as a feature extraction technique has become prevalent in the study of Synthetic Aperture Radar (SAR) data analysis. The goal of ICA is to find a set of independent, non-Gaussian signals that are mixed in the SAR data. The technique assumes that the underlying signals in the data are statistically independent, and it aims to separate these signals by maximizing their independence. The ICA algorithm starts by assuming that the mixed signals are linear combinations of the independent sources. It then estimates the mixing matrix and the independent sources by minimizing the mutual dependence between the signals. Once the independent sources have been estimated, they can be used as features for further analysis, such as classification or target detection. ICA has been found to be particularly useful for SAR data because it can effectively separate the signals from different scatterers in the scene, such as buildings and vegetation, which improves the interpretability of the data. It is also robust to noise and can handle non-Gaussian distributions. By applying ICA to SAR images from MSTAR data before classification, it is possible to extract useful features that are highly correlated with specific targets, such as building or vehicle signatures. These features can then be used to train a classifier, such as a neural network, to accurately identify and classify the different targets in the scene. Additionally, ICA is robust to noise, which is particularly important in MSTAR data, where the signals of interest may be weak or obscured by noise. As a result, ICA can be used to enhance the ratio of signal to noise within the given data, which leads to better classification performance.

The generative approach applied in Principal Component Analysis (PCA), wherein the latent variables are mixed to obtain observations, which are independents components can be concluded by assuming we have m linear mixture that we can observe $x_1,\ x_2, \dots$, x_m of n independent components.

$$x_j = \ a_{j1}s_1 + a_{j1}s_1 + \cdots + a_{jn}s_n \qquad (2.22)$$

The mixtures x_j and the independent components s_k each of them is a random variable. We can represent the equation: $x = As$ using the vector-matrix notation. The estimation of the latent variable s_i in ICA is characterized by the absence of direct observation, thereby necessitating concurrent estimation of the mixing matrix *A*. The underlying principle of ICA posits that the unobserved variables s_i exhibit statistical independence. This implies that the joint probability density function can be factored into the product of their corresponding marginal densities:

$$P(s) = \prod_{n=0}^{N} P_i\,(s_i) \qquad (2.23)$$

According to the principles of probability theory, the central limit theorem posits that when specific criteria are satisfied, the distribution of the summation of unconnected random variables tends towards a Gaussian or normal distribution. This theorem can be employed to postulate that a combination of s_i variables exhibit greater conformity to a Gaussian distribution as opposed to each variable taken individually. Therefore, A viable approach for estimating the independent components entails minimizing the Gaussian distribution of the variables s_i. This can be achieved by employing non-Gaussian measures, such as kurtosis and negentropy, to assess the degree of non-Gaussianity. An alternative method, that derives inspiration from information theory, entails harnessing the notion of differential entropy, with the objective of reducing mutual information.

$$I(y_1, y_2, \dots, y_n) = \sum_{i=0}^{n} H(y_i) - H(y) \qquad (2.24)$$

Mutual information is a metric that quantifies the level of interdependence between random variables, taking into account higher-order statistical properties. The aforementioned expression is represented as the Kullback-Leibler divergence between the joint density function $f(y)$ and the product of its corresponding marginal density functions. It has been observed that there is a significant relationship between negentropy, mutual information, projection pursuit, and their interconnectedness. Since negentropy is invariant for invertible transformation that minimizes mutual information similar to identifying is maximized. The endeavour of determining a singular constituent that optimizes the measure of negentropy is classified as a specific manifestation of projection pursuit and can be construed as a univariate feature.

III. SAR DATA CLASSIFICATION

A. Introduction

Classification is a fundamental approach utilized to partition a provided dataset into discrete classes, leveraging observable features derived from the training set. In the context of Synthetic Aperture Radar (SAR) imagery, this process involves the discernment and categorization of objects or targets present within the image into their respective classes. Specifically, the classification of marine targets within a SAR image is of interest, with the objective being to identify and allocate them based on discernible features. Nonetheless, the accurate classification of targets in SAR imagery remains a significant hurdle due to the inherent limitations in the quality of SAR images, leading to challenges associated with misclassification. The classification algorithm may encounter difficulties in discerning between relevant and non-relevant features, thereby impeding the accurate determination of target types for appropriate classification. Additionally, identifying similar-looking targets poses an additional challenge in SAR image classification, as distinguishing features that differentiate between such targets may have been suppressed or overlooked during the noise reduction process employed by the classifier. Consequently, the classification of similar-looking targets in SAR images remains an active area of research. The literature offers various approaches to SAR image classification, each presenting its own merits and limitations. These approaches will be comprehensively examined in the subsequent subsections, allowing for a detailed exploration of their strengths and weaknesses.

Classification involves predicting the class labels of unknown patterns based on observations. Consider a set of observations $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ of q-dimensional patterns $\mathcal{X} = \{x\}_{i=1}^{N} \subset \mathbb{R}^q$, and a corresponding set of labels $\mathcal{Y} = \{y_i\}_{i=1}^{N} \subset \mathbb{R}^q$, the objective of classification is to memorize a useful demonstrate f that can anticipate the class label y for a new pattern x with reasonable accuracy. Patterns without labels must be assigned labels of similar known patterns, such as those that are close in distance, belong to the same distribution, or fall on the same side of a decision boundary. However, learning from observations can be challenging due to noisy training data, unknown important features, difficulties in defining similarity between patterns, and inadequate descriptions of observations using simple distributions. Moreover, learning simple functional models can be challenging as classes may not be separable by linear decision boundaries or simple mathematical equations.

Automatic target recognition (ATR) in SAR images can be a challenging task due to the variability in the radar backscatter characteristics of targets under different conditions and the presence of other types of clutter in the image. Classification can be used to aid in the ATR process by identifying specific targets of interest within the SAR image based on their radar backscatter characteristics. The basic idea behind classification-based ATR is to train a classifier on a labeled dataset of SAR images containing known targets, such as the MSTAR dataset. The classifier can then be applied to new, unlabeled SAR images to identify potential targets. The classifier can be trained to recognize different types of targets by utilizing the unique radar backscatter characteristics of each one of them. For example, different types of tanks have different radar cross-section (RCS) signatures, which can be used to distinguish them from other types of clutter. Additionally, targets can be viewed from different angles, which can also affect their RCS signature. Once the classifier is prepared, it can be connected to unused SAR pictures to recognize ranges of the image that match the radar backscatter characteristics of the tanks of interest. These areas can then be flagged as potential targets for further analysis.

Several pre-processing steps that are typically performed on SAR data before classification to improve the accuracy of the classification results. Some of these pre-processing steps include radiometric calibration since SAR data may contain radiometric errors due to antenna gain fluctuations, atmospheric attenuation, and other factors. Radiometric calibration is performed to normalize the data and correct these errors. Speckle filtering, the coherent nature of the SAR imaging process can reduce the overall quality and clarity of the image. This issue could be solved by removing the high-frequency noise in the image while preserving the useful information. This is done by applying a spatial or frequency domain filter to the image in the spatial domain, to smooth and improve the image quality. SAR data is affected by the topography of the area being imaged, and terrain correction is performed to remove these effects and produce a more accurate image. Feature extraction is also an important step before the classification process. Since SAR data contains a large amount of information, feature extraction techniques are used to recognize and extricate the foremost important highlights for the classification task. The main purpose of feature extraction is to transform the raw input data into a set of representative and discriminative features that can be used by a classification algorithm to accurately classify the input data. The extracted features should capture the essential characteristics of the input data and should be invariant to irrelevant variations in the data. By performing these pre-processing steps, the SAR data can be prepared for classification, and the resulting classification results will be more accurate and reliable.

B. K-Nearest Neighbors

The k-nearest neighbour decision rule (k-NN) is a prominent classification algorithm utilized in the domain of statistical pattern recognition. The methodology encompasses representation of each class through a determined collection of exemplars a set of training pattern vectors that correspond to that particular class. When classifying a novel vector, the k-nearest neighbours are identified from the pool of prototype vectors, and the resulting class label is assigned through a majority voting scheme. To prevent the occurrence of overlaps in class regions requiring resolution by the process of tying, it is advisable to utilize an odd value for variable k. Whilst the k-NN (k-Nearest Neighbours) rule possesses a relatively uncomplicated and refined process, its error rate has been observed to be low in actual applications. As the quantity of prototype samples increases, the asymptotic error rate approaches the optimal Bayes error rate with a corresponding increase in k. Henceforth, the k-nearest neighbour algorithm serves as a conventional benchmark technique for assessing the performance of novel classifiers, including neural networks.

K-Nearest Neighbours k-NN is a simple, non-parametric classifier that is often used for image classification tasks, including Synthetic Aperture Radar (SAR) images. The basic idea behind the k-NN classifier is to identify the k-number of closest "neighbours" to a given data point in feature space, and then classify the point based on the majority class of those neighbours. In the case of SAR image classification, the k-NN algorithm would first convert the image pixels into feature vectors, which represent the relevant characteristics of the image. These feature vectors are then used to identify the k-nearest neighbours to a given pixel, and the algorithm assigns the majority class of those neighbours to that pixel. One of the main advantages of the k-NN classifier is its simplicity and ease of implementation. There are few parameters to tune, and the algorithm can be applied to a wide range of image classification tasks. Additionally, k-NN is a robust algorithm that can handle noisy and missing data. However, k-NN also has some limitations. One discernible obstacle is the computational expense associated with the algorithm, particularly when dealing with sizable datasets or feature spaces with high dimensions. Additionally, the choice of k-value can have a significant impact on the performance of the algorithm and finding the optimal k-value can be challenging. Within the realm of SAR image classification, a prevalent approach towards feature extraction involves the utilization of texture features including but not limited to the grey-level co-occurrence matrix (GLCM), grey-level run-length matrix (GLRLM), grey-level size zone matrix (GLSZM), and grey-level difference matrix (GLDM). These texture features serve as viable candidate inputs for algorithms such as the k-NN. It is a type of lazy learning algorithm that classifies new data points based on the majority class of its closest "k" training examples. The distance between the new data point and the training examples is typically measured using the Euclidean distance formula:

$$dxy = \sqrt{\sum_{i=1}^{n} (x_i + y_i)^2} \tag{3.1}$$

Nonetheless, the primary limitation of the k-NN decision rule is its substantial computational complexity stemming from the extensive number of distance computations required. In pattern spaces characterized by realistic dimensions, identifying a variant of the rule that significantly reduces computational intensity compared to the brute force method becomes a challenging task. The brute force method involves calculating distances between the unknown pattern vector and the prototype vectors, requiring computations for all possible combinations. To address this limitation, alternative adaptations of the nearest neighbour classifiers have been proposed, frequently incorporating editing, or pruning techniques. These modifications aim to reduce the number of prototypes while maintaining classification accuracy, offering potential solutions to mitigate computational demands. The algorithm for classification works by identifying the nearest patterns in data space to a target pattern x, which is the pattern for which we want to predict the label. The class label assigned to the target pattern is based on the majority class label among its K-nearest neighbours. To use k-NN, a similarity measure must be defined in data space. The Minkowski metric (p-norm) is a commonly used similarity measure in $\mathbb{R}^q$.

$$\left\|x' - x_j\right\|^p = \left(\sum_{i=1}^{q} \left|(x)' - (x_i)_j\right|^p\right)^{\frac{1}{p}} \tag{3.2}$$

where the Euclidean distance corresponds to $p = 2$. However, for other data spaces, appropriate distance functions must be selected. For instance, the Hamming distance can be used in $\mathbb{B}^q$. In binary classification problems, where the label set is $\mathcal{Y} = \{1, -1\}$, k-NN is defined as follows:

$$f_{KNN}(x') = \begin{cases} 1 \quad if \ \sum_{i \subset \mathcal{N}_K(x\prime)} y_i \geq 0 \\ -1 \, if \ \sum_{i \subset \mathcal{N}_K(x\prime)} y_i < 0 \end{cases} \tag{3.3}$$

Given a specific neighbourhood size, denoted as *K*, and a set of indices $\mathcal{N}_K(x')$ that represents the K-nearest patterns to a target pattern.

The selection of *K* determines how localized KNN will be. When *K = 1*, it tends to form small neighbourhoods in regions where patterns from various classes are dispersed. However, larger neighbourhood sizes, such as *K = 20*, disregard patterns with labels in the minority. To illustrate the classification difference between KNN with *K = 1* and

K=20, consider a simple two-dimensional dataset with two overlapping data clouds, each containing 50 Gaussian-sampled red and blue points.

When the neighbourhood size is small, KNN tends to overfit and become local. On the other hand, for larger neighbourhood sizes, KNN generalizes and ignores small clusters of patterns. Bright blue areas represent data space locations classified as blue, while white areas represent those classified as red. KNN with *K = 1* results in a local prediction, meaning an outlier blue point located at the center of the red cloud is identified as blue. Meanwhile, the classifier generalizes for larger *K* values, disregarding small clusters of patterns. KNN establishes a Voronoi tessellation in data space. When applied to vast datasets, KNN must scan a subset of patterns to identify the K-nearest neighbours, but it can still provide a good approximation.

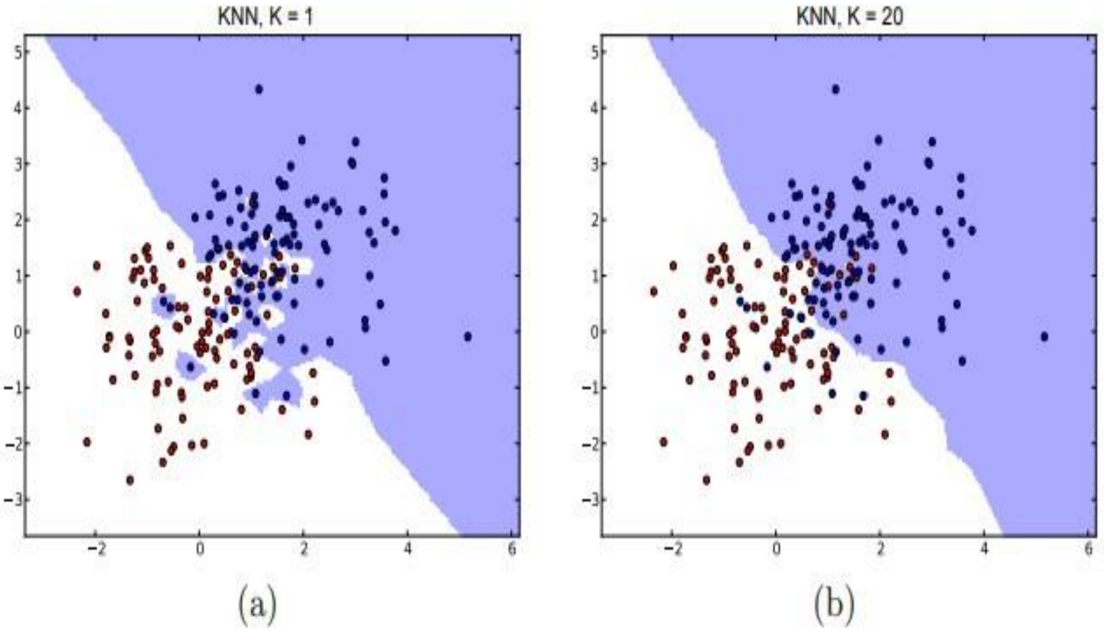


Figure 13: The comparison on two data clouds that are Gaussian-based is made for two different neighbourhood sizes, K = 1 and K = 20.

C. Support Vector Machine

The support vector machine (SVM) is a machine learning algorithm used for identifying subtle patterns in complex datasets (Vapnik, 1998). It is a supervised learning algorithm that learns from previous examples to predict the classification of new, unseen data. Cristianini and Shawe-Taylor (2000) have demonstrated the broad application of the Support Vector Machine (SVM) algorithm across diverse domains, including text categorization, image recognition, and hand-written digit recognition. In recent years, SVMs have emerged as a powerful tool in various bioinformatics domains. Notably, they have been successfully employed in the identification of translation start sites (Zien et al., 2000), detection of protein remote homology (Jaakkola et al., 1999; Liao and Noble, 2002; Leslie et al., 2003), recognition of protein folds (Ding and Dubchak, 2001), analysis of gene expression in microarray data (Brown et al., 2000; Guyon et al., 2001; Mukherjee et al., 1999; Furey et al., 2001; Vert and Kanehisa, 2003), functional classification of promoter regions (Pavlidis et al., 2001), prediction of protein-protein interactions (Bock and Gough, 2001), and identification of peptides from mass spectrometry data (Anderson et al., 2003). These applications highlight the versatility and effectiveness of SVMs in addressing various challenges in the field of bioinformatics.

The SVM algorithm is popular due to four main reasons. The proposed algorithm is underpinned by a robust theoretical framework rooted in VC dimension and the principle of structural risk minimization. Secondly, it can handle large datasets. Thirdly, it's flexible and has been applied to various domains, thanks to the robustness of the algorithm and the use of kernel functions for parameterization. The kernel function can be adjusted to include prior knowledge of a classification task. Finally, the SVM algorithm is highly accurate, which has contributed to its widespread application, despite the underlying theory not fully explaining its success.

1. Linear support vector machine

Linear SVM, a commonly employed machine learning algorithm, is highly effective for tackling classification problems. Specifically designed for linear classification scenarios, where data can be segregated by a single straight line or hyperplane, this variant of SVM excels. By aiming to locate the optimal hyperplane that distinctly separates the data into respective classes, the algorithm maximizes the margin between these classes. The margin, quantified as the distance between the hyperplane and the nearest data points (known as support vectors) from each class, plays a critical role in achieving optimal classification performance. The objective of the linear SVM is to maximize the margin and choose the hyperplane that has the largest distance from the closest data points of each class. The decision boundary of a linear SVM is a hyperplane in the feature space that separates the data into different classes.

Let’s assume that $x \subset \mathbb{R}^q$, the linear discriminant function is a mathematical expression used for classification tasks. It takes a pattern x as input and produces an output based on a linear combination of the components of x. The coefficients of this linear combination, called weights or parameters, are learned from a training dataset. The goal is to find the optimal values of the weights that maximize the separation between patterns from different classes. Once the weights are learned, the discriminant function can be used to predict the class of new patterns.

$$f(x) = \langle w, x \rangle + b, \tag{3.4}$$

A decision hyperplane is determined by a scalar product, weight $w \in \mathbb{R}^q$, bias $b \in \mathbb{R}$, and is also referred to as an inner product.

$$\langle w, x \rangle = \sum_{j=1}^{q} w_j x_j \tag{3.4}$$

The function $f(x)$ partitions the data space into two halves based on its sign. The SVM's learning objective is to determine the most suitable values for w and b parameters, which can effectively separate the two classes and ensure the accurate classification of pattern x'.

$$f_{LIN}(x') = \begin{cases} +1 & if\ f(x') \geq 0 \\ -1 & if\ f(x') < 0 \end{cases} \tag{3.5}$$

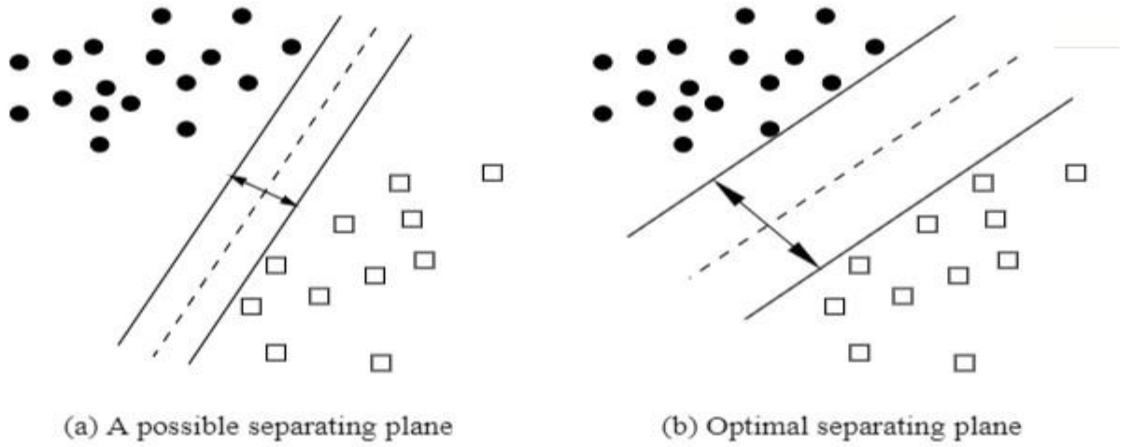
The linear classifier is also referred to as a linear decision hyperplane and is optimized to correctly classify patterns. The optimal decision hyperplane should have a large distance to its closest patterns, known as the margin, and can be represented as 1/w2. In the hard margin SVM, a linear decision hyperplane is defined to correctly classify all patterns. The optimization problem can be formulated as:

$$minimize_{w,b}\frac{1}{2}\|w\|^2$$

$$subjectto\colon y_i(\langle w, x_i\rangle + b) \geq 0 \tag{3.6}$$

$$for\ i = 1, \dots, N$$

From a geometrical standpoint, the objective is to identify the hyperplane that maximizes the total distance to the nearest positive and negative instances used in training. This distance is commonly referred to as the margin (as shown in Figure 12), and the optimal hyperplane is derived by maximizing the expression $\frac{2}{\|w\|}$ or, alternatively, minimizing $\|w\|$ while subject to the constraint $y_i(\langle w, x_i\rangle + b) \geq 0$.


(a) A possible separating plane

(b) Optimal separating plane

Figure 14: The potential solutions for the problem of finding a separating plane. The second solution is anticipated to result in better generalization compared to the first.

To enhance generalization capability, the distance between the separating hyperplane $f(x)$ and patterns should be maximized, provided that all training patterns are correctly classified. Since patterns are frequently not separated linearly, the separability constraint is loosened by introducing *slack variables* ξ, which quantify the deviation of patterns from the associated hyperplane.

2. Radial Basis Function (RBF) kernel

The kernel function is a mathematical function that can convert input data from a lower dimension to a higher one, by mapping it into a new feature space. This new feature space will make the data linearly separable, allowing for the use of a support vector machine to find a hyperplane that separates the data. For example, if the input data is two-dimensional, the kernel function can map it into a three-dimensional space where it will be linearly separable.

The fundamental concept behind kernel methods is to transform an input pattern x into a feature space represented by $\phi(x)$. For instance, a polynomial feature space can be used to represent a two-dimensional input pattern $x = (x_1, x_2)^T$ under a quadratic polynomial transformation as $\phi(x) = ({x_1}^2, {x_2}^2, \sqrt{2}x_1x_2)^T$. The kernel trick is an essential part of most kernel algorithms, where the dot product of two input patterns x and x' in feature space $\phi(x).\, \phi(x')$ is needed to be computed by a kernel function $k(x, x')$. This eliminates the need to work explicitly in the feature space. In the case of the quadratic example, the corresponding kernel is $(x.x')^2$. Other algorithms, such as neural networks or tree ensembles, extract intrinsic properties of data points through a kernel function, which can result in more features than kernel algorithms. This is particularly useful for problems involving handwritten recognition, face detection, and other tasks.

The development of kernel machines was initially intended to address supervised learning tasks, such as classification and regression. The predictor at a given test point x can be represented as $h(f(x))$, where $f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$. The function h is given and may be the sign function for a two-class classification problem or the identity function for a regression problem, The $\{x_i\}_{i=1}^{n}$ set comprises the patterns used for training, and the α_i coefficients are real numbers. Such prediction methods are not uncommon in statistics. Kernel machines encompass various techniques, such as spline methods (e.g., Wahba 1990; Green and Silverman 1994) and Gaussian process models, which operate in the realm of function space regularization and reproducing kernel Hilbert spaces. Gaussian process models have gained substantial prominence and have found extensive applications in geo-statistics and machine learning (e.g., Williams and Rasmussen 1996). These models leverage the concept of Gaussian processes to effectively model and analyze complex data patterns, making them valuable tools in diverse domains.

The support vector machine differs from other methods in the expansion $f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$ which is usually sparse. The support patterns are those training patterns with non-zero α’s. The reason for this sparsity can be explained by considering a classification problem. When considering spline prediction techniques, it is customary to utilize the function $f(x)$ to effectively represent the logarithmic odds ratio $log\, p(y = +1|x) = p(y = -1|x)$. This is equivalent to a term for the negative logarithm of the likelihood, which is $log(1 + e^{-y_i f_i})$ with $f_i = f(x_i)$. Nevertheless, the SVM utilizes a different "data-fit" expression $[1 - y_i f_i]_+$, which is similar but has a key difference that it is exactly 0 for $y_i f_i \geq 0$ ¸ 1. This term $[1 - y_i f_i]_+$ is considered a soft constraint. If $y_i f_i \geq 1$, the constraint is satisfied; Otherwise, a penalty is incurred for its violation. The characteristic of Support Vector Machines (SVM) induces sparsity, whereby a training instance is deemed safely classified. (When $y_i f_i \geq 1$), If a training point is not required to satisfy the constraint (i.e., the constraint is not active), then this point will not be included in the expansion.

The RBF kernel then uses the basic idea of Linear SVM to perform classification. To enhance the dimensional space, projecting the data into a higher level is a requisite, the RBF kernel employs the radial basis function, which can be expressed as:

$$k(x, x') = exp(-\frac{\|x - x'\|^2}{2\sigma^2}) \tag{3.7}$$

The term $\|x - x'\|^2$ represents the Squared Euclidean Distance between two data points x and x'. By tuning the σ parameter, The Radial Basis Function (RBF) kernel possesses the capability to transform efficiently and effectively non-linearly separable data points into a higher-dimensional space, thereby facilitating separation by means of a hyperplane.

By incorporating a new parameter $\gamma = \frac{1}{2\sigma^2}$, the equation can be expressed as:

$$k(x, x') = exp(-\gamma \|x - x'\|^2) \tag{3.8}$$

The RBF kernel function is characterized by a simple equation, wherein the Squared Euclidean Distance is multiplied by the gamma parameter, followed by taking the exponent of the entire product. By implementing this equation, the transformed inner products for mapping the data into higher dimensions can be computed directly, without transforming the entire dataset, thus avoiding any inefficiencies. This is precisely why the function is referred to as the RBF kernel function.

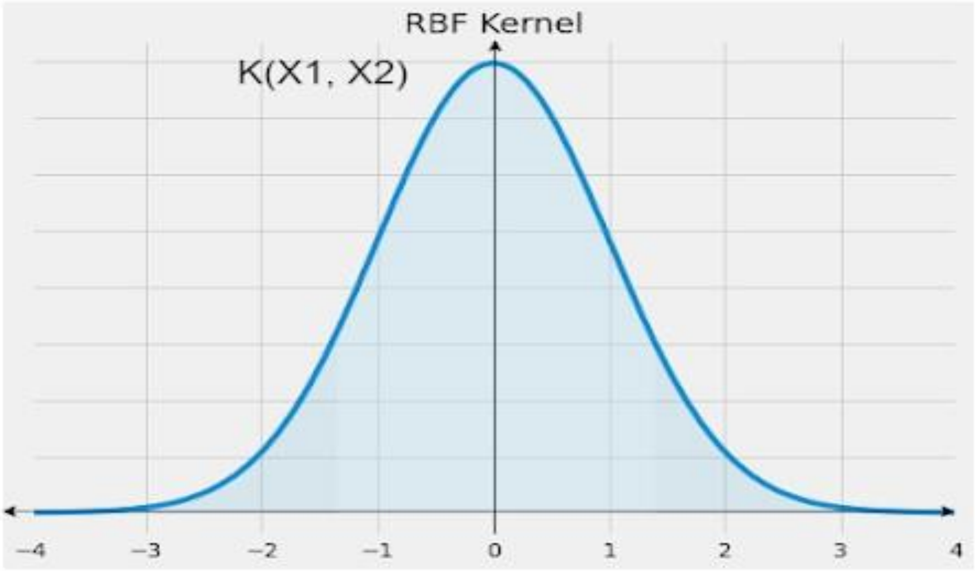


Figure 15: The graphical representation of the distribution of the RBF kernel.

The distribution plot of the RBF kernel resembles the Gaussian distribution curve, also called a bell-shaped curve. This is why the RBF kernel is often referred to as the Gaussian Radial Basis Kernel.

D. Random Forest

The Random Forest (RF) algorithm is a type of supervised machine learning technique commonly utilized for both classification and regression analysis. This approach constitutes an ensemble methodology whereby numerous decision trees' outcomes are integrated to foster a more precise and robust prediction. This method can be used to classify different objects or land covers in the image. The algorithm works by training multiple decision trees on different subsets of the SAR image data and features, and then averaging their predictions to classify each pixel in the image. The described algorithm demonstrates its versatility by being applicable to both Classification and Regression problems within the domain of machine learning. It operates on the principle of ensemble learning; wherein multiple classifiers are utilized to address intricate problems and enhance the overall performance of the model. By combining the insights and predictions from multiple classifiers, the algorithm harnesses the collective intelligence to achieve more accurate and robust results. This approach proves particularly effective in tackling complex problems where a single classifier may struggle to provide satisfactory outcomes. The Random Forest algorithm is a classification method that involves multiple decision trees, each of which are trained on diverse sub-samples of the input data. By computing the average of the predictions produced by the individual decision trees, the accuracy of the dataset is greatly enhanced. The Random Forest approach does not depend solely on a solitary decision tree, but instead determines the output by aggregating the majority votes derived from the predictive models generated by multiple trees. For the purpose of improving precision and minimizing overfitting, it is recommended to augment the quantity of trees within the forest.

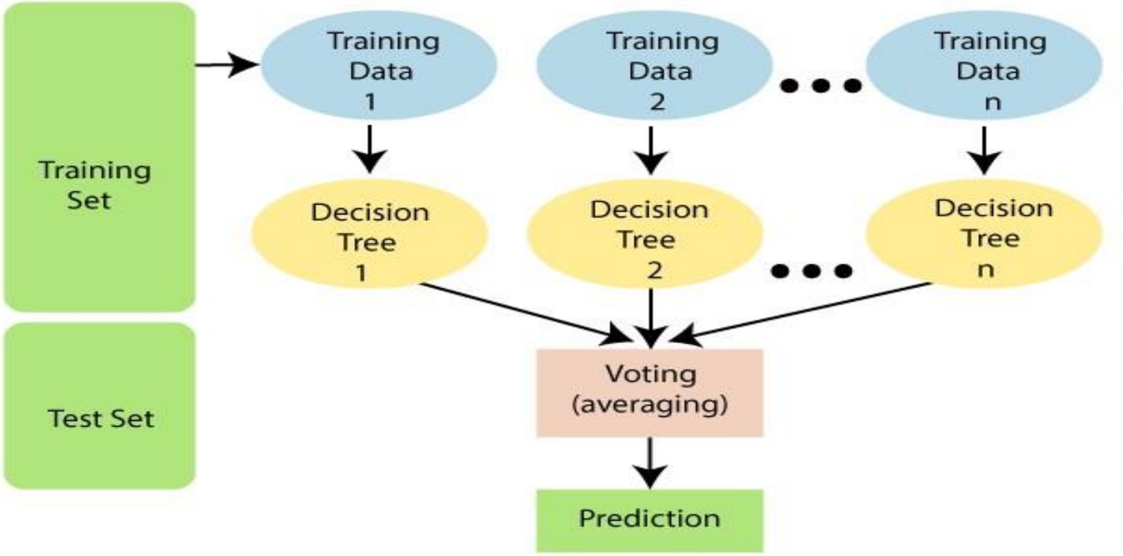


Figure 16: The random forest classifier processing diagram.

In recent years, there has been an increasing interest in the use of the random forest (RF) classifier (Breiman, 2001) due to its superior classification performance and fast processing speed, as demonstrated in various studies (Du et al., 2015, Pal, 2005, Rodriguez-Galiano et al., 2012). The RF classifier achieves accurate classification results by using predictions from a group of decision trees (Breiman, 2001). Moreover, this classifier can effectively identify and rank variables that have the highest discriminatory power between target classes. This feature is particularly valuable since selecting the most relevant variables from high-dimensional remotely sensed data is time-consuming

(Körting et al., 2013), error-prone, and subjective process (Belgiu et al., 2014a). Several studies have extensively examined the application of the RF classifier for categorizing hyperspectral data (Ham et al., 2005) and distinguishing land cover types like ETM+ (Pal, 2005) or MSS and DEM data (Gislason et al., 2006).

Random forest is a useful machine learning algorithm for handling large datasets with numerous predictor variables. However, in some cases, it may be necessary to minimize the number of predictors to improve efficiency. For instance, when creating a medical prediction model, it might be preferable to use only the most critical variables instead of all variables present in the electronic medical record. Variable selection is a technique that identifies the most critical predictors for developing a parsimonious model with optimal predictors. By selecting the optimal predictors based on statistical characteristics like accuracy or importance, variable selection can reduce the data collection burden and enhance prediction model efficiency. Considering the prevalent high-dimensional nature of contemporary datasets, the process of selecting variables has emerged as an indispensable element in the development of accurate prediction models.

In this context, we shall limit our attention to the binary classification scenario and disregard the consideration of multi-class complications, even though random forests have been demonstrated to be competent in addressing such complexities in Díaz-Uriarte and de Andrés (2006). In this problem (outlined in Devroye et al. 1996), the binary response Y has two possible outcomes $\{0, 1\}$. Given X, the value of Y must be predicted. A classifier or classification rule, m_n, is a Borel measurable function of X and D_n which aims to estimate the label Y from X and D_n. In this context, a classifier m_n is considered consistent if the probability of error is minimized.

$$L(m_n) = \mathbb{P}[m_n(X) \neq Y] \underset{n\to\infty}{\longrightarrow} L^* \tag{3.9}$$

The variable L^* refers to the error made by the Bayes classifier that is ideal but not known.

$$m * (x) = \begin{cases} 1 & if\ \mathbb{P}[y = 1|X = x] > \mathbb{P}[y = 0|X = x] \\ 0 & \qquad\qquad\qquad\qquad otherwise. \end{cases} \tag{3.10}$$

For classification, the random forest classifier is created by taking the majority vote of the classification trees.

$$m_{M,n}(x;\Theta_1,\ldots,\Theta_M,D_n) = \begin{cases} 1\; if\; \frac{1}{M}\sum_{j=1}^{M} m_n(x;\Theta_j,D_n) > 1/2 \\ 0 \qquad\qquad\qquad\qquad\qquad otherwise. \end{cases} \tag{3.11}$$

If a section of a tree represents an area labeled as A, then a classifier using randomized trees can take a straightforward shape:

$$m_{M,n}(x;\Theta_j,D_n) = \begin{cases} 1\; if \sum\limits_{i \subset D^*{}_n(\Theta_j)} 1_{X_i \subset A, Y_i=1} > 1_{X_i \subset A, Y_i=0}, x \in A \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad otherwise. \end{cases} \tag{3.12}$$

If the leaf corresponds to region A, then a randomized tree classifier adopts a simple approach where a majority vote is taken over all data points (X_i, Y_i) in the same region selected during the resampling step. The set $D^*{}_n(\Theta_j)$ contains the selected data points. In the event of a tie, class 0 is favored by convention. Algorithm 1 can be easily modified to perform two-class classification without changing the CART-split criterion. Consider a single tree with no subsampling step, where $Y \in \{0,1\}$. For any generic cell A, let $p_{0,n}(A)$ $(resp., p_{1,n}(A))$ denote the empirical probability that a data point in cell A has label 0 (resp., label 1). By noticing that $\bar{Y}_A = p_{1,n}(A) = 1 - p_{0,n}(A)$, the classification CART-split criterion reads, for any $(j,z) \in C_A$.

$$\begin{aligned} L_{class,n}(j,z) = p_{0,n}(A)p_{1,n}(A) - \frac{N_n(A_L)}{N_n(A)} \times p_{0,n}(A_L)p_{0,n}(A_L) \\ - \frac{N_n(A_R)}{N_n(A)} \times p_{0,n}(A_R)p_{0,n}(A_R) \end{aligned} \tag{3.13}$$

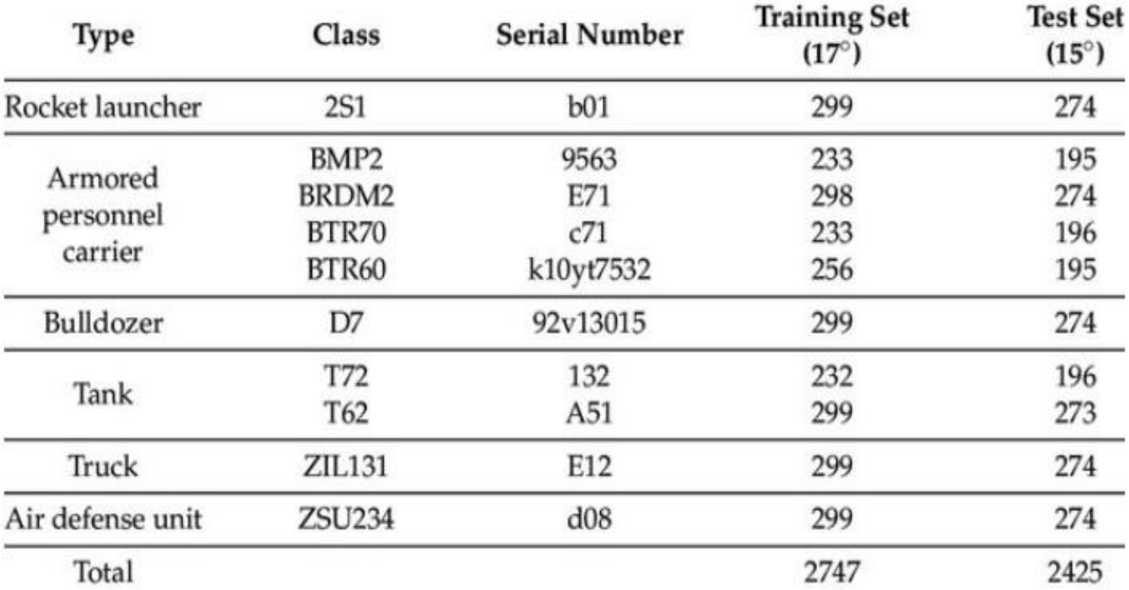
This measure is known as the Gini impurity measure (Breiman et al. 1984) and forms the basis of the criterion. Its interpretation is straightforward: given a data point that falls within a particular cell A, the probability of labeling it as ℓ is $p_{\ell,n}(A)$, with ℓ taking values in $\{0,1\}$. To estimate the probability that the item actually belongs to class ℓ, one can use $p_{\ell,n}(A)$, which is the empirical probability. The estimated error using this approach is the Gini index, which is equal to $2p_{0,n}(A)p_{1,n}(A)$. However, it is important to note that the prediction strategies differ in regression and classification. In the classification case, a local majority vote is used by each tree, while in regression, local averaging is used to make predictions.

IV. DESIGNED EXPERIMENTS AND RESULTS

A. Data Base Used

The validation of the proposed classifiers was done using a database of Synthetic Aperture Radar (SAR) images of ten military ground targets. These images were obtained as part of the Moving and Stationary Target Acquisition and Recognition (MSTAR) program, which is a project backed by the Defense Advanced Research Projects Agency (DARPA). The images were collected using a Twin Otter SAR sensor payload from Sandia National Laboratories and operated at X-band frequency.

Table 1: Details of the targets from MSTAR dataset used for the experiments.



Type	Class	Serial Number	Training Set (17°)	Test Set (15°)
Rocket launcher	2S1	b01	299	274
Armored personnel carrier	BMP2	9563	233	195
	BRDM2	E71	298	274
	BTR70	c71	233	196
	BTR60	k10yt7532	256	195
Bulldozer	D7	92v13015	299	274
Tank	T72	132	232	196
	T62	A51	299	273
Truck	ZIL131	E12	299	274
Air defense unit	ZSU234	d08	299	274
Total			2747	2425

The selected targets for the actual experiments are the 2S1 rocket launcher, and four armored personnel carrier BMP2, BRDM-2, BTR-60 and BTR-70, D-b Bulldozer, T62 tank, T72 tank, ZIL131 truck, and the ZSU-23-4 air defense unit.

The classifiers were trained using image clips gathered at an altitude of 17 degrees and the clips obtained at an altitude of 15 degrees were used as testing images. The resolution of the clips was adjusted to 96x96 pixels.

B. Experiment Preprocessing Description

1. Reshaping Data

Using Python 3.9, we reshape the training and testing data matrices by flattening them into a single vector by using the ‘numpy’ function ‘np.reshape()’. The new shape is determined by multiplying the second and third dimensions of the original shape. The first dimension of the new shape is the number of samples, which is the same as the first dimension of the original shape.

By reshaping the data into a single vector, the algorithm can treat each pixel in the image as a separate feature, which allows it to learn patterns in the image data. It is a very important preprocessing step for image data because it allows for more efficient computation during the training and testing of the model. Reshaping the data into a single vector makes it more easily comparable to other types of data, such as text, allowing for more flexibility in choosing machine learning models and techniques.

2. Normalizing Data

This step is being used to normalize the pixel values of the SAR images. Dividing each element of training and testing data by 255.0 will scale all pixel values to the range of 0 to 1. This is useful for machine learning algorithms as it can help to improve the stability of the training process and make the model more robust.

3. Mean-Centering Data

This is done by subtracting the mean value of each feature (i.e., each pixel value in the SAR image) across all samples from the feature values for each sample. This is important because it helps to standardize the data and remove any bias due to the mean of the data. Furthermore, it is an important step to adjust the overall brightness of the image to make it easier to analyze and it can help to reduce the impact of lighting conditions on the data. It is also important to note that before the mean centering the data is normalized by dividing it with 255.0 to ensure that the feature values are in the range of [0, 1].

C. Classification of Reduced Number of Samples

1. Classification of Original data

When we use a classifier on data that has been reshaped to a single vector and normalized, the classifier will treat each pixel value as a separate feature. The classifier will then use these features to determine the class label of the image.

In the context of linear classification, a hyperplane is typically sought to effectively and accurately segregate distinct categories within the given feature space. The placement and alignment of the hyperplane shall be ascertained by the coefficients of the model that are acquired through the process of training data. Once the hyperplane is found, the classifier will use it to make predictions on new images by computing the dot product of the image's feature vector with the coefficients of the model. If the result is positive, the image is classified as one class, and if it is negative, it is classified as the other class.

In the case of a non-linear classifier, it will learn a more complex decision boundary that separates the different classes in the feature space, this can be done by kernel trick or other methods. The classifier will then use this decision boundary to make predictions on new images.

2. Reduced Number of Data

To evaluate the importance of different rotation angles of the targets for the classification program and how it affects its accuracy, we selected and down sampled the training data for each of the 10 classes in the dataset. For this reason, a ‘for’ loop was used to iterate over each class.

For each class, the code selects only every 3rd example from the input arrays of the original data. The resulting arrays are then appended to a new variables which are used to store the selected samples, and the ‘np.concatenate()’ function is used to concatenate these lists to form a single training set. The new number of training samples and the difference angle between two successive samples are listed in Table 2.

Table 2: 1/3 Reduced number of samples of MSTAR ten target classes.

Target Class	Number of Training Samples	Angle between Samples (°)	Reduced Number of Training Samples	New Angle between Samples (°)
2S1	299	1.2	150	2.4
BMP2	233	1.54	117	3.08
BRDM2	298	1.2	149	2.4
BTR60	256	1.4	128	2.8
BTR70	233	1.54	117	3.08
D7	299	1.2	150	2.4
T62	298	1.2	149	2.4
T72	232	1.55	116	3.1
ZIL131	299	1.2	150	2.4
SZU234	299	1.2	150	2.4

After that, we reduced more features by selecting only every 5th example from the input arrays of the original data. The new number of training samples and the difference angle between two successive samples for each case, are listed in Table 3.

Reducing the amount of data in a dataset have advantages, such as reducing the computational time required to train the model, as it needs to process fewer examples, also avoiding overfitting, especially when the original dataset is large, and the model is complex. Overfitting is a phenomenon that arises when a model exhibits a close fit to the characteristics of training data but fails to extend that performance to novel, unseen data.

Table 3: 1/5 Reduced number of samples of MSTAR target classes.

Target Class	Number of Training Samples	Angle between Samples (°)	Reduced Number of Training Samples	New Angle between Samples (°)
2S1	299	1.2	60	6
BMP2	233	1.54	47	7.7
BRDM2	298	1.2	60	6
BTR60	256	1.4	52	7
BTR70	233	1.54	47	7.7
D7	299	1.2	60	6
T62	298	1.2	60	6
T72	232	1.55	47	7.75
ZIL131	299	1.2	60	6
SZU234	299	1.2	60	6

However, reducing the amount of data can also lead to a loss of information and lower the testing accuracy, as we can see in the accuracy results listed in Table 4.

Table 4: Training and testing accuracy for original and reduced features.

Target	Training Accuracy	Testing Accuracy
Original Features	0.999	0.969
1/3 Reduced Features	0.998	0.921
1/5 Reduced Features	1.0	0.853

D. Classification after Features Extraction

To improve the classification accuracy, we implemented different features extraction methods that were explained in detail in the second section. First, we added a code for shuffling the training and testing datasets by concatenating the feature data (X) and the target data (y) into a single matrix, then using the 'np.random.shuffle' function to randomly shuffle the rows of the matrix. Finally, returns the shuffled feature and target data as separate arrays.

This shuffling is important for training machine learning models because it helps ensure that the model is not overfitting to the specific order of the data, and that it has been observed that the model exhibits enhanced capability to extend its learning to novel, unencountered datasets, thereby demonstrating superior generalization abilities.

The Data have been classified after extracting the most significant features using Principal Component Analysis (PCA), then using Kernel Principal Component Analysis (KPCA), later using the independent Component Analysis (ICA).

We used three different classifiers with each feature extraction method, K- Nearest Neighborhood (K-NN), Random Forest (RF), Radial Basis Function kernel with Support Vector Machine (RBF-SVM) and Linear Support Vector Machine (SVM).

The accuracy of classifications has been calculated for each one of the mentioned methods. The results are listed in Table 5.

Table 5: Accuracy results for SAR data without noise reduction

Without Noise Reduction				
Features	Classification Accuracy			
	K-NN (k=10)	Random Forest	RBF-SVM	Linear SVM (C=1)
Original	82.72%	78.22%	96.97%	95.83%
ICA (80)	93.04%	90.76%	96.85%	55.05%
PCA (80)	95.79%	96.05%	97.73%	92.16%
KPCA (80)	95.77%	96.39%	97.63%	92.28%

E. Effect of Speckle Noise Reduction on Classification

1. Classification Accuracy after Applying Mean Filter

To mitigate the deleterious effects of stochastic noise within the data set under examination, a mean filter was employed prior to engaging in the process of classification.

This procedure contributes to enhancing the efficacy of the classifier by rendering the features depicted in the image distinguishable and amenable to classification. For this reason, we implement a code with a mean filter before reshaping the data. We define the "kernel" variable as a 2-dimensional array with shape (3, 3), which represents the filter that will be used for convolution. The values in the array are set to 1/9, so the average of the neighboring pixels will be computed as the sum of their values divided by 9. The code applies the mean filter to each image in the arrays of training and testing data by using the "convolve" function from a library such as ‘SciPy’ or ‘NumPy’. The convolve function performs a convolution operation between the image and the filter kernel, successfully supplanting each pixel esteem with the normal of its neighboring pixel values. We tested all the previously mentioned classification methods after implementing noise reduction and list the results in Table 6.

Table 6: Accuracy results after reducing noise with Mean filter.

After Mean Filter Noise Reduction				
Features	Classification Accuracy			
	K-NN (k=10)	Random Forest	RBF-SVM	Linear SVM (C=1)
Original	96.08%	86.14 %	98.10%	97.23%
ICA (80)	96.10%	91.91%	98.68%	63.71 %
PCA (80)	96.94 %	97.81%	98.43 %	94.96 %
KPCA (80)	96.86 %	97.89%	98.47 %	95.17 %

2. Classification after Applying Median Filter

Median filter was implemented to evaluate the difference in the accuracy results and see which filter gives the best outcome to improve the accuracy. The median filter replaces each pixel in the SAR image with the median of its neighboring pixels. The median is the middle value of the set of values, and is a more robust statistic than the mean, as it is less affected by extreme values and does not introduce blurring and preserve the resolution of the image.

For this reason, we used a code with two "for" loops to iterate through each image in the input arrays. In each loop, the current image is stored in a variable. The "ndimage.median_filter" function is then used to apply the median filter to the image, using a kernel size of (3,3). The filtered image is stored in a different variable, and then the original image is replaced with the filtered image. The accuracy results for the different classification methods that we trained are listed in Table7.

Table 7: Accuracy results after reducing noise with Median filter.

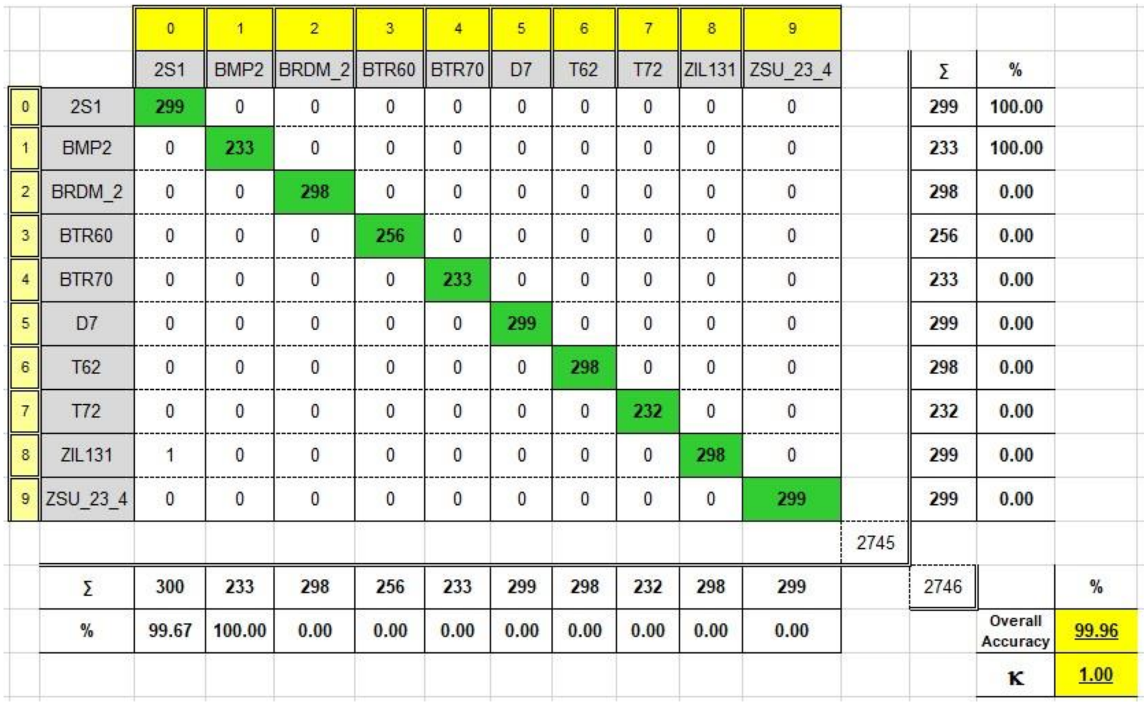
After Median Filter Noise Reduction				
Features	Classification Accuracy			
	K-NN (k=10)	Random Forest	RBF-SVM	Linear SVM (C=1)
Original	93.27%	85.27%	97.77%	96.57%
ICA (80)	93.03%	90.63%	97.81%	57.23%
PCA (80)	95.67%	97.07%	98.14%	96.57%
KPCA (80)	95.58%	97.07%	98.10%	93.97%

3. Training and Testing Confusion matrix

To evaluate the performance of our classification model, we trained a confusion matrix as it provides a comprehensive summary of the model's predictions, we chose the SVM-RBF model since it gives the best results. The framework shows the number of rectify and erroneous expectations made by the model and helps to identify patterns of misclassification and bias in the model's performance.

By visualizing the confusion matrix, it is possible to assess the accuracy, precision, recall and F1 score of the model, and compare its performance with other models.

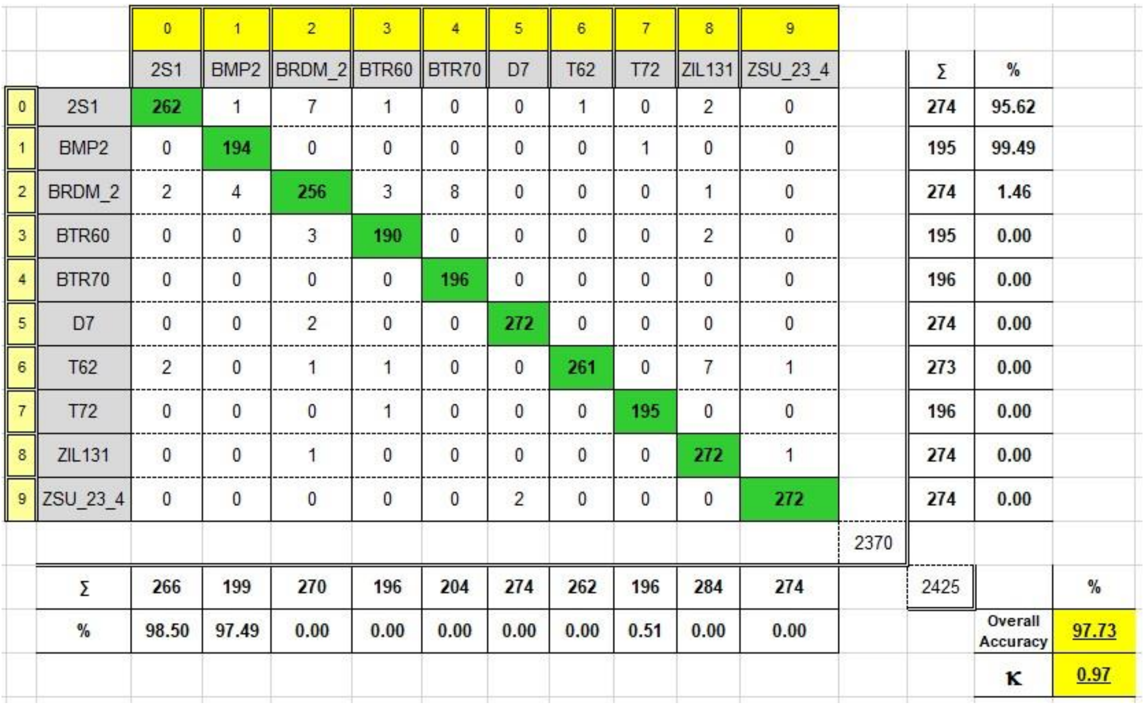
Table 8: Confusion table of true and predicted training classes of SVM-RBF classifier with PCA feature extraction and without speckle filtering.



		0	1	2	3	4	5	6	7	8	9				
		2S1	BMP2	BRDM_2	BTR60	BTR70	D7	T62	T72	ZIL131	ZSU_23_4		Σ	%	
0	2S1	299	0	0	0	0	0	0	0	0	0		299	100.00	
1	BMP2	0	233	0	0	0	0	0	0	0	0		233	100.00	
2	BRDM_2	0	0	298	0	0	0	0	0	0	0		298	0.00	
3	BTR60	0	0	0	256	0	0	0	0	0	0		256	0.00	
4	BTR70	0	0	0	0	233	0	0	0	0	0		233	0.00	
5	D7	0	0	0	0	0	299	0	0	0	0		299	0.00	
6	T62	0	0	0	0	0	0	298	0	0	0		298	0.00	
7	T72	0	0	0	0	0	0	0	232	0	0		232	0.00	
8	ZIL131	1	0	0	0	0	0	0	0	298	0		299	0.00	
9	ZSU_23_4	0	0	0	0	0	0	0	0	0	299		299	0.00	
												2745			
	Σ	300	233	298	256	233	299	298	232	298	299		2746		%
	%	99.67	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			Overall Accuracy	99.96
														κ	1.00

To build the Confusion Matrix, it is crucial to combine both the actual class labels and the classifier's predictions into one table (Table 8). The actual class labels are placed in the columns while the classifier predictions occupy the rows. This creates a square matrix with perfect classifications appearing along the main diagonal. A total of 2746 values were utilized for the training process, with the correct classifications being placed along the main diagonal of the Confusion Matrix. The Kappa coefficient measures the level of agreement between the classifications and the actual class labels. The results indicate that the Kappa value obtained from the training data is approximately 0.99, where a value of 1 represents perfect agreement and a value of 0 signifies no agreement.

Table 9: Confusion table of true and predicted training classes of SVM-RBF classifier with PCA feature extraction and without speckle filtering.

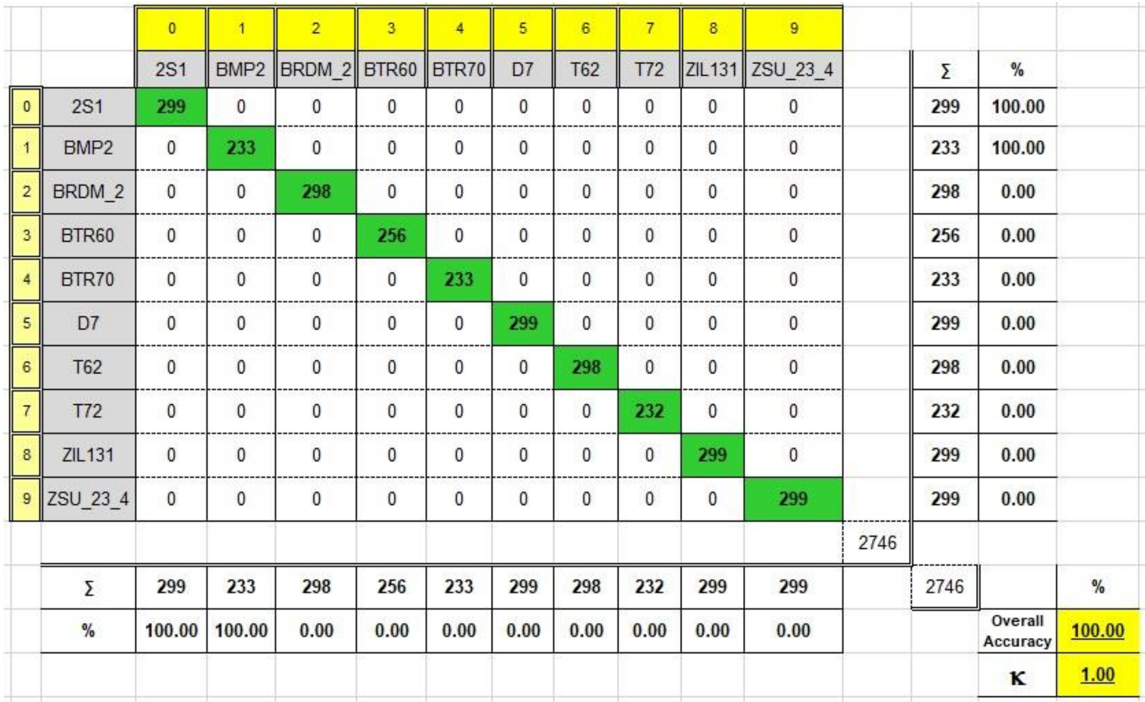


		0	1	2	3	4	5	6	7	8	9				
		2S1	BMP2	BRDM_2	BTR60	BTR70	D7	T62	T72	ZIL131	ZSU_23_4		Σ	%	
0	2S1	262	1	7	1	0	0	1	0	2	0		274	95.62	
1	BMP2	0	194	0	0	0	0	0	1	0	0		195	99.49	
2	BRDM_2	2	4	256	3	8	0	0	0	1	0		274	1.46	
3	BTR60	0	0	3	190	0	0	0	0	2	0		195	0.00	
4	BTR70	0	0	0	0	196	0	0	0	0	0		196	0.00	
5	D7	0	0	2	0	0	272	0	0	0	0		274	0.00	
6	T62	2	0	1	1	0	0	261	0	7	1		273	0.00	
7	T72	0	0	0	1	0	0	0	195	0	0		196	0.00	
8	ZIL131	0	0	1	0	0	0	0	0	272	1		274	0.00	
9	ZSU_23_4	0	0	0	0	0	2	0	0	0	272		274	0.00	
												2370			
	Σ	266	199	270	196	204	274	262	196	284	274		2425		%
	%	98.50	97.49	0.00	0.00	0.00	0.00	0.00	0.51	0.00	0.00			Overall Accuracy	97.73
														κ	0.97

A testing matrix, like the Confusion matrix used for training, is employed to determine the precision of the code that has been written (as shown in Table 9). Similar to the training matrix, the true classifications and the classifier's predictions are organized in a column-wise manner. In the training process, a total of 2425 values were utilized. Evaluating the trained data against the true classifications reveals an impressive overall accuracy of 97.73%. Additionally, the Kappa value calculated from the training data measures an impressive agreement of 0.97, indicating a high level of concordance between the predicted and true classifications. These results underscore the effectiveness and reliability of the trained model in accurately classifying the data.

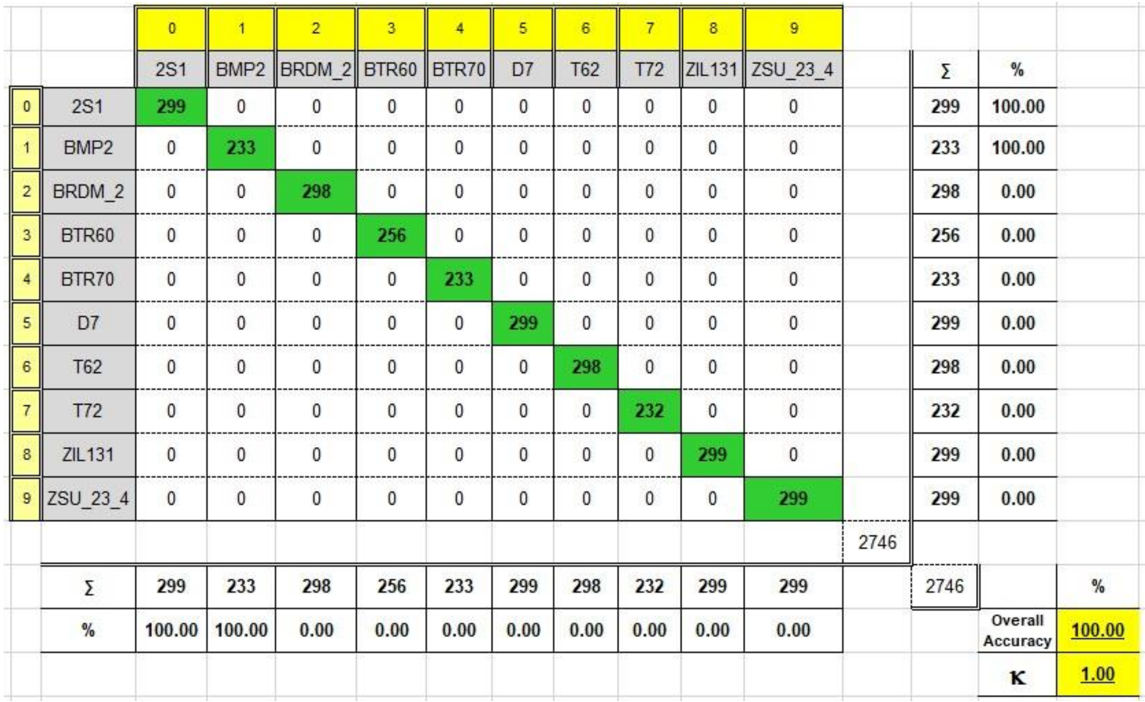
To have a clear vision about the effect of speckle noise filtering we generated training and testing confusion matrices for the same classifier which is SVM-RBF, always extracting features with PCA. But this time after we implemented mean filtering on the data, to see how the process of speckle noise reduction moves forward the accuracy of the classification model. The results are shown in Tables 10 and 11.

Table 10: Confusion table of true and predicted training classes of SVM-RBF classifier with PCA feature extraction after speckle filtering.



		0	1	2	3	4	5	6	7	8	9				
		2S1	BMP2	BRDM_2	BTR60	BTR70	D7	T62	T72	ZIL131	ZSU_23_4		Σ	%	
0	2S1	299	0	0	0	0	0	0	0	0	0		299	100.00	
1	BMP2	0	233	0	0	0	0	0	0	0	0		233	100.00	
2	BRDM_2	0	0	298	0	0	0	0	0	0	0		298	0.00	
3	BTR60	0	0	0	256	0	0	0	0	0	0		256	0.00	
4	BTR70	0	0	0	0	233	0	0	0	0	0		233	0.00	
5	D7	0	0	0	0	0	299	0	0	0	0		299	0.00	
6	T62	0	0	0	0	0	0	298	0	0	0		298	0.00	
7	T72	0	0	0	0	0	0	0	232	0	0		232	0.00	
8	ZIL131	0	0	0	0	0	0	0	0	299	0		299	0.00	
9	ZSU_23_4	0	0	0	0	0	0	0	0	0	299		299	0.00	
												2746			
	Σ	299	233	298	256	233	299	298	232	299	299		2746		%
	%	100.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			Overall Accuracy	100.00
														κ	1.00

Table 11: Confusion table of true and predicted training classes of SVM-RBF classifier with PCA feature extraction after speckle filtering.



		0	1	2	3	4	5	6	7	8	9				
		2S1	BMP2	BRDM_2	BTR60	BTR70	D7	T62	T72	ZIL131	ZSU_23_4		Σ	%	
0	2S1	299	0	0	0	0	0	0	0	0	0		299	100.00	
1	BMP2	0	233	0	0	0	0	0	0	0	0		233	100.00	
2	BRDM_2	0	0	298	0	0	0	0	0	0	0		298	0.00	
3	BTR60	0	0	0	256	0	0	0	0	0	0		256	0.00	
4	BTR70	0	0	0	0	233	0	0	0	0	0		233	0.00	
5	D7	0	0	0	0	0	299	0	0	0	0		299	0.00	
6	T62	0	0	0	0	0	0	298	0	0	0		298	0.00	
7	T72	0	0	0	0	0	0	0	232	0	0		232	0.00	
8	ZIL131	0	0	0	0	0	0	0	0	299	0		299	0.00	
9	ZSU_23_4	0	0	0	0	0	0	0	0	0	299		299	0.00	
												2746			
	Σ	299	233	298	256	233	299	298	232	299	299		2746		%
	%	100.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			Overall Accuracy	100.00
														κ	1.00

V. CONCLUSION AND FUTURE WORKS

The valuable research conducted in Synthetic Aperture Radar (SAR) Automatic Target Recognition (ATR) technology was made possible by the public data released by DARPA/SNL and ARFL in the form of Moving and Stationary Target Acquisition and Recognition (MSTAR). This thesis presents a method to evaluate and rank the performance of ATR algorithms through the computation of their accuracies and respective confusion matrices and calculation of their Kappa statistics, which is typically a good indicator of a classification algorithm's reliability.

The experiments were carried out on MSTAR data using Python programming codes with different classifiers such as KNN, Random Forest and SVM with its both technics, RBF SVM and Linear SVM. After applying multiple preprocessing methods and feature extraction as ICA, PCA and KPCA, we compared the effect of each of them on the classification accuracy, first without applying any speckle noise reduction filters and later after applying Mean and Median filters separately to evaluate their effect on the classification's accuracy. We conclude that the noise reduction process has increased the accuracy of the classifiers. The best results we reached with our experiments were with RBF- SVM after applying PCA that gives us an accuracy of 97.72% that we were able to amplify with reducing the speckle noise on our data with Mean filter to reach 98.47%.

To further advance the field, additional investigations can be undertaken to explore alternative methods for feature extraction, such as Non-Negative Matrix Factorization (NMF). This exploration would enable the analysis of various classification algorithms and their behavior. Moreover, the integration of deep learning techniques, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), holds promise for enhancing both the accuracy and robustness of automatic target recognition systems. Leveraging transfer learning, pre-trained deep learning models can be fine-tuned specifically for the target recognition task, minimizing the requirement for extensive labeled data. Furthermore, unsupervised learning techniques, such as clustering and dimensionality reduction, provide opportunities for the automatic identification and categorization of diverse target types without relying on labeled data. These avenues of research offer promising avenues for advancing automatic target recognition methodologies.

REFERENCES

BOOKS:

A MAITY, A. PATTANAIK, S. SAGNIKA, and S. PANI, **“A Comparative Study on Approaches to Speckle Noise Reduction in Images”,** in 2015 International Conference on Computational Intelligence and Networks, 2015, pp. 148–155.

ANTONINI, G., POPOVICI, V., & THIRAN, J.-P. (2003). **Independent Component Analysis and Support Vector Machine for Face Feature Extraction. Audio- and Video-Based Biometric Person Authentication**, 111–118.

BRUCE, J. S. (2020). **Automatic Target Recognition** (Vol. TT120). SPIE PRESS BOOK.

CHENG, G., ZHAO, W., ZHANG, J., & MAO, S. (2006). **A Practical Kernel Criterion for Feature Extraction and Recognition of MSTAR SAR Images**. 2006 8th International Conference on Signal Processing.

D. A. E. MORGAN, **“Deep convolutional neural networks for ATR from SAR imagery,”** Proc. SPIE, vol. 9475, p. 94750F, May 2015.

I. A. ABDULMAJEED AND I. M. HUSIEN, (2022) "**Machine Learning Algorithms and Datasets for Modern IDS Design**" 2022 IEEE International Conference on Cybernetics and Computational Intelligence (Cybernetics.Com), pp. 335-340, doi: 10.1109/CyberneticsCom55287.2022.9865255.

L. CHEN ET AL., **“A new deep learning network for automatic bridge detection from SAR images based on balanced and attention mechanism,”** Remote Sens., vol. 12, no. 3, p. 441, Jan. 2020

ARTICLES:

DONG, G., KUANG, G., WANG, N., ZHAO, L., & LU, J. (2015). **SAR target recognition via joint sparse representation of monogenic signal**. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8(7), 3316-3328.

GU, B.; SHENG, V.S.; TAY, K.Y.; ROMANO, W.; LI, S. **Incremental Support Vector Learning for Ordinal** Regression. *IEEE Trans. Neural Netw. Learn. Syst.* 2017, *26*, 1403–1416.

JIANXIONG, Z., ZHIGUANG, S., XIAO, C., & QIANG, F. (2011). **Automatic target recognition of SAR images based on global scattering center model**. IEEE transactions on Geoscience and remote sensing, 49(10), 3713-3729.

K. EL-DARYMLI, E. W. GILL, P. MCGUIRE, D. POWER AND C. MOLONEY, **"Automatic target recognition in synthetic aperture radar imagery**: **A state-of-the-art review"**, IEEE Access, vol. 4, pp. 6014-6058, 2016.

JOURNALS:

DANG, S.; CUI, Z.; CAO, Z.; LIU, N. **SAR target recognition via incremental nonnegative matrix factorization**. Remote Sens. 2018, 10, 374.

J. SCHMIDHUBER, **“Deep learning in neural networks: An overview,”** Neural Network., vol. 61, pp. 85–117, Jan. 2015.

LIU, H., & LI, S. (2013). **Decision fusion of sparse representation and support vector machine for SAR image target recognition**. Neurocomputing, 113, 97-104.

MARTONE, A., INNOCENTI, R., & RANNEY, K. (2009). **Moving target indication for transparent urban structures.** US Army Research Laboratory Adelphi United States

RAHMANIZADEH, A.; AMINI, J. **An integrated method for simulation of synthetic aperture radar (SAR) raw data in moving target detection**. *Remote Sens.* 2017, *9*, 1009.

TANG, S.; ZHANG, L.; SO, H. **Focusing high-resolution highly squinted airborne SAR data with manoeuvres**. *Remote Sens.* **2018**, *10*, 862.

ELECTRONIC JOURNALS:

BAMBRICK N., (2020). **Support Vector Machines**. [Online] Available at: https://www.kdnuggets.com/2016/07/support-vector-machines-simpleexplanation.html

Biau, G., Scornet, E. **A random forest guided** tour. TEST 25, 197–227 (2016). https://doi.org/10.1007/s11749-016-0481-7

KRAMER, O. (2013). **K-Nearest Neighbours. In: Dimensionality Reduction with Unsupervised Nearest Neighbours**. Intelligent Systems Reference Library, vol 51. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-38652-7_2

URL-1 "**Regularization in Machine Learning - Javatpoint**", www.javatpoint.com, 2022. [Online]. Available: https://www.javatpoint.com/regularization-in-machine-learning

URL-2 "**Regression Analysis in Machine learning - Javatpoint**", www.javatpoint.com, 2022. [Online]. Available: https://www.javatpoint.com/regression-analysis-in-machine-learning

R.-E. FAN, K.-W. CHANG, C.-J. HSIEH, X.-R. WANG, AND C.-J. LIN. **LIBLINEAR: A Library for Large Linear Classification**, Journal of Machine Learning Research 9(2008), 1871-1874. Sage Journals. [Online] Available at: https://journals.sagepub.com/doi/10.1177/000812561986492

PAPERS:

M. KANG, K. JI, X. LENG, X. XING, AND H. ZOU, **“Synthetic aperture radar target recognition with feature fusion based on a stacked autoencoder,”** Sensors, vol. 17, no. 12, Jan. 2017, Art. no. 192.

M. ROSTAMI, S. KOLOURI, E. EATON, AND K. KIM, **“Deep transfer learning for few-shot SAR image classification,”** Remote Sens., vol. 11, no. 11, p. 1374, Jun. 2019.

N. AGRAWAL AND K. VENUGOPALAN, **“Speckle reduction in remote sensing images,”** in 2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), 2011, pp. 195–199.

NOVAK, L. M., OWIRKA, G. J., & BROWER, W. S. (1998, November). **An efficient multi-target SAR ATR algorithm**. In Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers (Cat. No. 98CH36284) (Vol. 1, pp. 3-13). IEEE.

NOVAK, L. M., OWIRKA, G. J., & NETISHEN, C. M. (1993). **Performance of a high-resolution polarimetric SAR automatic target recognition system**. Lincoln Laboratory Journal, 6(1).

OKFALISA, I. GAZALBA, MUSTAKIM AND N. G. I. REZA**, "Comparative analysis of k-nearest neighbour and modified k-nearest neighbour algorithm for data classification,"** *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia, 2017, pp. 294-298, doi: 10.1109/ICITISEE.2017.8285514.

PAN, Z., QIU, X., HUANG, Z., & LEI, B. (2016). **Airplane recognition in TerraSAR-X images via scatter cluster extraction and reweighted sparse**

representation. IEEE Geoscience and Remote Sensing Letters, 14(1), 112-116.

PEI, J.; HUANG, Y.; HUO, W.; ZHANG, Y.; YANG, J.; YEO, T.-S. **SAR automatic target recognition based on multiview deep learning framework**. *IEEE Trans. Geosci. Remote Sens.* 2018, *56*, 2196–2210.

S. CHEN, H. WANG, F. XU, AND Y.-Q. JIN, **"Target classification using the deep convolutional networks for SAR images"**, *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806-4817, Aug. 2016.

SCHUMACHER AND K. ROSENBACH, **“ATR of battlefield targets by SAR–Classification results using the public MSTAR dataset compared with a dataset by QinetiQ, UK,”** in Proc. RTO SET Symp. “Target Identification Recognit. Using RF Syst.,” 2004, pp. 11–13, [Online].

S. WU, Q. ZHU, AND Y. XIE, **“Evaluation of various speckle reduction filters on medical ultrasound images.,”** Eng. Med. Biol. Soc. (EMBC), 2013 35th Annu. Int. Conf. IEEE, pp. 1148 – 1151, Jan. 2013.

U. A. M. E. ALI, M. A. HOSSAIN AND M. R. ISLAM, **"Analysis of PCA Based Feature Extraction Methods for Classification of Hyperspectral Image,"** 2019 2nd International Conference on Innovation in Engineering and Technology (ICIET), Dhaka, Bangladesh, 2019.

U. SRINIVAS, V. MONGA, AND R. G. RAJ**, “SAR automatic target recognition using discriminative graphical models,”** IEEE Trans. Aerosp. Electron. Syst., vol. 50, no. 1, pp. 591–606, Jan. 2014.

WANG, H.; LI, S.; ZHOU, Y.; CHEN, S. **SAR automatic target recognition using a Roto-translational invariant wavelet-scattering convolution network**. *Remote Sens.* 2018, *10*, 501.

YANG J., FRANGI A. F., ET AL. **“KPCA Plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition”.** IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(2), pp.230-244, 2005.

YANG YINAN, QIU YUXIA AND LU CHAO, **“Automatic Target Classification Experiments on the MSTAR SAR Images”**, Proceedings of the Sixth International Conference on SNPD/SAWN’05, 2005, pp. 2-7

Y. HE, S. Y. HE, Y. H. ZHANG, G. J. WEN, D. F. YU AND G. Q. ZHU**, "A forward approach to establish parametric scattering center models for known complex radar targets applied to SAR ATR"**, IEEE Trans. Antennas Propag., vol. 62, no. 12, pp. 6192-6205, Dec. 2014.

ZHOU, F.; WANG, L.; BAI, X.; HUI, Y. **SAR ATR of ground vehicles based on LM-BN-CNN**. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 7282–7293.

ZHOU, XIAOBING & CHANG, NI-BIN & LI, SHUSUN. (2009). **Review Applications of SAR Interferometry in Earth and Environmental** Science Research. Sensors (Basel, Switzerland). 9. 1876-912. 10.3390/s90301876.

DEMERS, DAVID AND Y, GARRISON COTTRELL. **Non-linear dimensionality reduction. In Advances in Neural Information Processing** Systems 5, pp. 580–587. Morgan Kaufmann, 1993.

TWINING, C. J. AND TAYLOR, C. J. **Kernel principal component analysis and the construction of non-linear active shape models**. In Proceedings of British Machine Vision Conference, pp. 23–32, 2001.

PAVLIDIS, P., WAPINSKI, I., & NOBLE, W. S. (2004). **Support vector machine classification on the web. Bioinformatics,** 20(4), 586–587.

CRISTIANINI,N. AND SHAWE-TAYLOR,J. (2000)**An Introduction to Support Vector Machines. Cambridge University Press,** MA.

Belgiu, M., & Drăguţ, L. (2016). **Random forest in remote sensing: A review of applications and future directions**. ISPRS Journal of Photogrammetry and Remote Sensing, 114, 24–31

HASNA EL HASNAOUY

Electrical Engineering

Electronics Engineering

Skills

- MATLAB
- Python
- C++
- Microsoft Office
- Problem-solver
- Team Player
- Self-Motivated
- Punctual
- Time management

Languages

- Arabic
- English
- Frensh
- Turkish

RESUME

Motivated and diligent master's degree student in Electronic Engineering with a strong passion for innovation and problem-solving. Proven ability to apply theoretical knowledge to practical engineering projects and deliver exceptional results.

Education

- March 2021- July 2023
  Istanbul Aydin University
  Master of Science in Electrical and Electronics Engineering
- September 2015 – July 2020
  Cadi Ayyad University- Limoges University
  Double diploma fundamental and experimental Bachelor degree in Electronics and Optics for Embedded Systems

Research experience

- Conference Presentations
  - ➢ Conference on Advances in Mechanical Engineering, Istanbul, Turkey, October 2021
    Paper title: Analysis of ternary organic solar cells
  - ➢ Recent Advances on Air and Spaces Technologies, Istanbul, June 2023
    Paper title: Paper title: Comparison of Feature Extraction Methods for Automated Target Recognition by Reducing Speckle Noise in SAR Data.
- Publications
  El Hasnaouy, H., Kasapoğlu, N. G., "Comparison of Feature Extraction Methods for Automated Target Recognition by Reducing Speckle Noise in SAR Data." In Proceedings of the Recent Advances on Air and Spaces Technologies Conference, Istanbul, Turkey, June 2023.