PolSAR Image Classification using Complex-Valued CNN and Squeeze-Excitation Network

by

Makhija Shradha Vipinkumar
202011029

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

May, 2022

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Shradha

Makhija Shradha Vipinkumar

Certificate

This is to certify that the thesis work entitled PolSAR Image Classification using Complex-Valued CNN and Squeeze-Excitation Network has been carried out by Makhija Shradha Vipinkumar for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my supervision.

Prof. Srimanta Mandal
Thesis Supervisor

Acknowledgments

This thesis is a testimony of the support that I have gotten from all the people around me. I would like to begin by extending my sincerest gratitude to my supervisor Prof. Srimanta Mandal. Without his motivation and unmatched patience, this thesis wouldn't have been possible. He made a point to always encourage me to explore new and relevant ideas and provided me with the finest directions. I am also grateful to late Prof. Suman Mitra who became my guiding light in the initial phase of this work. His knowledge and invaluable inputs have helped me throughout my thesis. His words of motivation have and will always remain with me in whatever I do.

I am also thankful to Dr. Sanid Chirakkal and Deepak Putrevu from SAC-ISRO for making me understand the intricacies of the kind of data that we are dealing with. Their constant help in understanding the fundamentals have made this thesis what it is today. I extend my deepest gratitude to SAC-ISRO for giving me this opportunity to work as a junior research fellow.

I would like to thank Nilam Chaudhari, who is now an alumna of DA-IICT. As my senior, she helped me understand the intuitions behind the data management of this domain and she has always been available to clear my doubts. I would also thank Tushar Gadhiya whose work in this domain has provided me a lot of useful insights. I would also like to thank other faculty members who taught me during my masters and the institute, DA-IICT, as a whole for providing me the environment and enormous learning opportunities.

I would like to thank Prof. Anil Roy for maintaining his encouraging demeanour throughout my stay at campus for the thesis work. Lastly, I would like to thank all my peers who have supported me throughout my work and studies. A very special thanks to my family for their unconditional love and support throughout this program and in general.

Contents

Abstract

Terrain classification is one of the most crucial tasks when utilisation of polarimetric SAR images comes into the picture. This work explores the efficiency of various supervised deep learning algorithms that make use of Convolutional Neural Networks in land cover classification of PolSAR images. The goal of this work is to classify terrain into different ground covers such as urban, crops, forests, water, etc., from polarimetric SAR (PolSAR) images. State-of-the-art classification approaches relish the advantage of deep learning techniques. However, conventional techniques such as convolutional neural networks (CNN), developed for optical images are not quite suitable for complex-valued PolSAR images. Hence, in this work, complex-valued CNN is employed to deal with complex values of PolSAR images. Further, the CNN focuses mainly on the spatial relationship within local receptive fields. However, the process entangles the channel correlation with spatial information. To address this issue, we use a squeeze-excitation network (SENet) along with complex-valued CNN to exploit the channel interdependencies. Thus, we utilize spatial as well as channel relationships in our work. This, in turn, helps in reducing the speckle noise in the images. Additionally, this work also tests the effectiveness of data augmentation techniques to increase the size of labeled training set of the three datasets used. This is done using various speckle noise suppression techniques. The experimental results on several datasets justify the importance of both spatial information as well as inter-channel correlation in classifying PolSAR images. The results after applying data augmentation techniques specific to the speckled nature of PolSAR images show improved performance of SENet architectures proposed in this work.

List of Tables

List of Figures

CHAPTER 1
Introduction

Synthetic Aperture Radar (SAR) has now become a well-explored remote sensing technique that provides large-scale two-dimensional images with high spatial resolution. These images are mainly that of the Earth's surface reflectivity. This reflectivity is active in nature as the SAR imaging system is a radar system that illuminates the surface with microwave pulses.

The successful launch of SEASAT satellite in 1978 proved that imaging radar is the most capable and an indispensable Earth remote sensing instrument. SAR imaging has established itself as the only viable and practical radar imaging system for acquiring high spatial resolution data. It has done so because of its portability when compared to other aperture of same calibre, and also due to the fact that radar can be mounted on airborne as well as spaceborne platforms. Moreover, the microwaves are capable of penetrating through various atmospheric barriers and the radar system is capable of illuminating the surface on its own. All these features make SAR imaging system independent of day time conditions as well as of atmospheric conditions.

The data is acquired based on the nature of signal used by the radar. This is classified into the polarization channels. These polarization channels capture various features on the same object. Usually SAR systems operate on a single band from C, L, P, and X. But a few modern SAR systems are capable of capturing multple frequency data making the data containing abundant information. SAR images have established their place in various research domains such as agriculture, urban, disaster management, etc for decades now.

An SAR image is a two-dimensional array of pixels. These pixels form a complex valued array containing amplitude and phase information. This information is stored in various formats such as coherency matrix, covariance matrix, etc using different statistical parameters. The data is stored across various matrices and to interpret such massive information, automated systems are required.

The following sections briefly discuss the objective of the work, contribution of the thesis and the organization of this thesis.

1.1 Objective

The goal of this work to develop various deep learning models to perform classification of polarimetric SAR images. The models are required to perform heterogeneous land cover classification such as urban, crops, forests, water, etc. The thesis focuses on the following aspects:

- Investigating the efficiency of Complex-Valued Convolutional Neural Networks (CV-CNNs) for the classification task.
- Application of deep learning models such as SENets using complex valued deep learning operations.
- Investigating the effects of incorporating skip connections in the deep learning models.
- Application of data augmentation techniques for the classification task and investigating its effects.

1.2 Contribution of Thesis

The significant features of this work can be summarized as:

- Understanding the efficiency of complex-valued CNNs for the classification task.
- Implementing Squeeze and Excitation blocks (SE Blocks) to extract spatial as well as channel-wise important features for PolSAR image classification.
- Modifying the SE blocks to accommodate the complex-valued deep learning operations.
- Eliminating the need of denoising the PolSAR images to reduce the speckle noise.
- Performing the data augmentation to increase the size of training data for better classification.

1.3 Organization of Thesis

The remaining thesis is organized as follows.

- Chapter 2 discusses Literature survey that contains various classification techniques employed for PolSAR classification.
- Chapter 3 contains details about how an SAR imaging system operates, how the data is acquired, the mathematical preliminaries associated with the PolSAR data, the datasets used for all the experiments, and the brief details about CNNs.
- Chapter 4 and Chapter 5 contain information about the architectures used for classification.
- Chapter 6 shows the effect of data augmentation using speckle denoising.
- Chapter 7 contains summary of all the work done and the comparison with the benchmark results described in leading research works.
- Chapter 8 discusses the conclusion and future scope of this work.

CHAPTER 2
Literature Survey

PolSAR image classification is about classifying each pixel of the input image into a set of predefined labels. The ultimate goal is to analyze the target images based on their orientation, shape, geometric structure, and configuration. It also provides insightful interpretations for many applications. All the classification techniques developed for PolSAR classification fall under either of two areas, statistical-based classification and machine learning-based classification.

PolSAR provides terrain information under day and night conditions and under all weather conditions. Considering the nature of data, many unsupervised statistical approaches have been proposed for classification task. For extracting the polarimetric characteristics of the Entropy-Alpha classes, analysis of the polarimetric signatures using H/α decomposition has been examined [35]. For fitting simple back-scatterer mechanisms to polarimetric SAR data, a scattering model employing three components has been proposed where single bounce, double bounce and volume scattering mechanisms have been explored [18]. Various Eigenvector-Eigenvalue based decomposition techniques are used and the classification is then done using Wishart supervised classifier [12].

Since the covariance and coherency matrix of PolSAR data follow Wishart distribution, Wishart mixture model has also been implemented for terrain classification where the model parameters are estimated using Expectation-Maximization algorithm [7]. When it comes to pattern recognition tasks, classification performance hugely depends on the feature extraction part. These extracted features include back-scattering elements [8], target decomposition-based features [3], and other statistic features [10]. Statistical methods designed to extract such features are usually class-specific and involve manual trial and error. These methods may provide better performance, but these methods require deep domain knowledge and also, knowledge about specific radar configurations is required for performing feature extraction [5]. This is where multistage deep learning models have scope for automation [53].

Machine learning classifiers such as SVM [21, 19] and KNN [40] have shown efficiency for the classification problem. Deep CNN [51], Dual-branch based deep CNN [20], semi supervised graph-based SPGraphCNN [4] are some of the methods proposed for PolSAR classification and have proven to be effective. Some problem specific CNNs such as Tc-CNN [23] which uses three channel CNN to extract spatial information has also been used. PDAS based model namely CVPDAS-CNN [15] has been developed.

CNNs have demonstrated superior performance with their hierarchical feature extraction capabilities [2]. The fundamental task of ship detection for SAR-based surveillance has been carried out using CNNs [9] on Sentinel-1 data. Furthermore, automatic target recognition for SAR has also been efficiently implemented by using a combination of CNNs with support vector machines [43]. In PolSAR images, the data is in the form of complex valued pixels and conventional CNNs accept only real numbers. Usually, polarimetric coherency matrix (T_3) or polarimetric covariance matrix (C_3) are used for image classification task. These matrices contain 6 channels of complex valued arrays. Using T_3 matrix, real valued CNN has been implemented where a new 6-D real vector representation has been specifically tailored for classification of PolSAR data [52].

In some works, real part and imaginary part of the data are fed separately as a 9-D feature vector [50]. A 6-D complex feature vector is also fed into Complex-Valued CNN where entire CNN architecture handles complex numbers [50][52]. Experiments are also performed with only the real valued coherency matrix elements (T_{11}, T_{22}, T_{33}), where a newly tailored 6-D feature vector representation has been proposed [44]. Backpropagation in complex valued CNN domain has been implemented using Complex Generalized Derivative and Complex Conjugate Derivative and it has shown improvement over real valued MLPs [25].

Squeeze and Excitation networks for suitably classifying the low-frequency and contour subbands-driven polarimetric data has also been implemented [37] as LC-PSENet. Terrain classification based on the local binary patterns using feature integrated CNN called CLBP-CNN has been implemented and its accuracy measured on Flevoland, San Francisco, and Oberpfaffenhofen [1]. Deep Residual Network (ResNet) has also been implemented for the classification task [32]. A lightweight network using DeepLabv3 has also been implemented [14].

Data scaling techniques have time and again proved to be effective for image classification using deep learning techniques [33].CNNs that are adaptively fine-tuned produce results after training on minimally labelled data [48], but the need of the hour in PolSAR image classification is data scaling. Data augmentation techniques have found its place in remote sensing imagery also. For improving the efficiency of object detection algorithms of multimodal remote sensing images, data scaling has been implemented [41]. Also, the issue with the datasets used in PolSAR classification is the size of these datasets which is small. Large scale data is needed for proper classification and these techniques are also shown to improve existing models' performance in classification [13].

Data augmentation in these experiments has been performed using speckle denoising. Polarimetric SAR speckle noise model in [31] clarifies the implications of 2D speckle noise. An adaptive algorithm exploring the effectiveness of mean-shift-based speckle filtering is discussed in [27]. Images with minimum speckle have been produced after applying polarimetric whitening filter (PWF) in [34] and it has produced comparable results. Many a times, during speckle filtering phase, crosstalk between polarization channels is detected. This issue was tackled in [28] and a classification was performed using Bayes maximum likelihood classification algorithm. Since speckle noise also carries terrain information, it is important to preserve the potential target information and it was achieved in [6]. Impact of speckle filtering was studied by performing classification of data using K Means clustering in [11] which also presents a comparative study of several speckle filters that are used widely.

CHAPTER 3

Background

This chapter describes the background of SAR, mathematical preliminaries of polarimetric SAR. It also throws some light on Convolution Neural Networks and how deep learning confronts few of the major challenges in PolSAR image classification.

3.1 Remote Sensing

With the advent of remote sensing, new doors have opened for research in Earth sciences and that has brought tremendous advancements in the technologies used for accomplishing these research related tasks. Remote sensing makes the task of information gathering from a target without the need of physical contact. This enables data gathering from sources or areas that are inaccessible.

Due to easy access of various landcovers and areas, remote sensing has made us capable of capturing the intricacies of climate change, natural disasters, rise of urbanization, resource depletion, and many more such applications. Figure 3.1 depicts the electromagnetic spectrum with different frequency bands along with their wavelengths. Out of these, remote sensing makes use of signals from microwave, infrared, and visible range. Optical sensing makes use of visible and infrared bands whereas non-optical or microwave sensing makes use of microwave bands. Though optical remote sensing has been around for a long time, its capabilities are limited to daytime because it needs sunlight or some source of illumination to work. Atmospheric barriers such as clouds, rainfall, and haze. Additionally, the information provided by the microwave sensing is different in nature altogether. Microwave sensing holds the scope of capturing information about the electric properties as well as geometry of the target. Besides, the range of microwave band ranges from 1mm to 1m which enables it to surpass clouds, dust, rainfall, mist, etc. Due to these characteristics, microwave imaging can operate during any time of the day under any type of atmospheric conditions. Moreover,

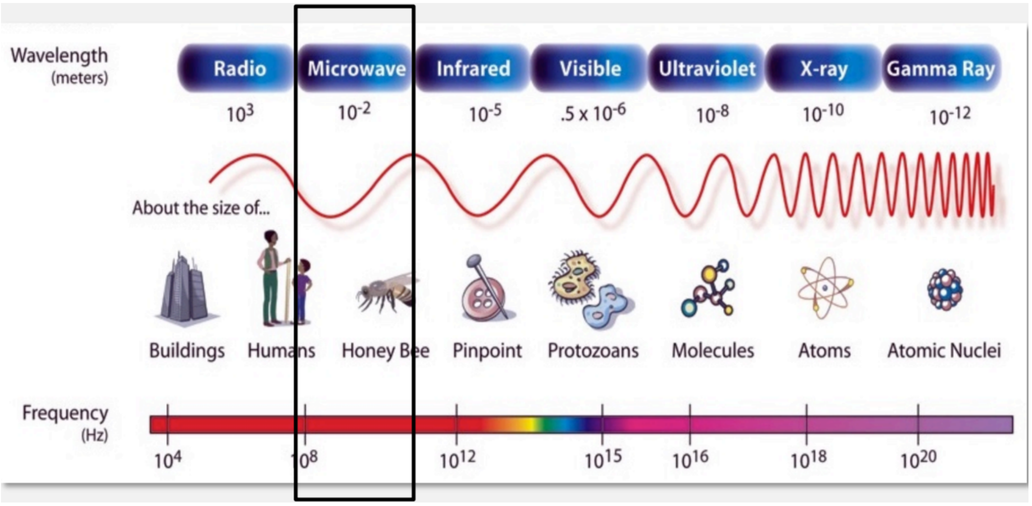


Figure 3.1: Electromagnetic spectrum with different bands [42]

microwave signals are also capable of penetrating the vegetation and a few subranges can also penetrate the ground. All these characteristics make microwave imaging apt for monitoring situations above and below the ground. However, the obtained images are different from the optical images, hence it is a challenge to interpret them.

Microwave sensing falls under two categories, active and passive. Antennas and receivers are a part of both of these categories. Active sensors illuminate the target with their own electromagnetic signals hence they don't require any kind of source for illumination. On the other hand, passive sensors require energy from a source of illumination or a source of heat and measure the emissions.

3.2 Synthetic Aperture Radar

Figure 3.2 shows the basic principle of Radio Detection and Ranging (RADAR) system mounted on a moving platform for synthesizing an image. A radar emits electromagnetic waves on the surface and these waves then get backscattered by the target surface upon the incident. A proportion of these scattered waves is received by the radar. The target image is constructed using these backscattered waves,

Depending on the size of the beam sent out by the antenna, the spatial resolution of the image is characterized. For capturing the finer details of the target area, a narrow beam makes more sense than a wider beam. The length of the antenna also known as the aperture is inversely proportional to the width of the

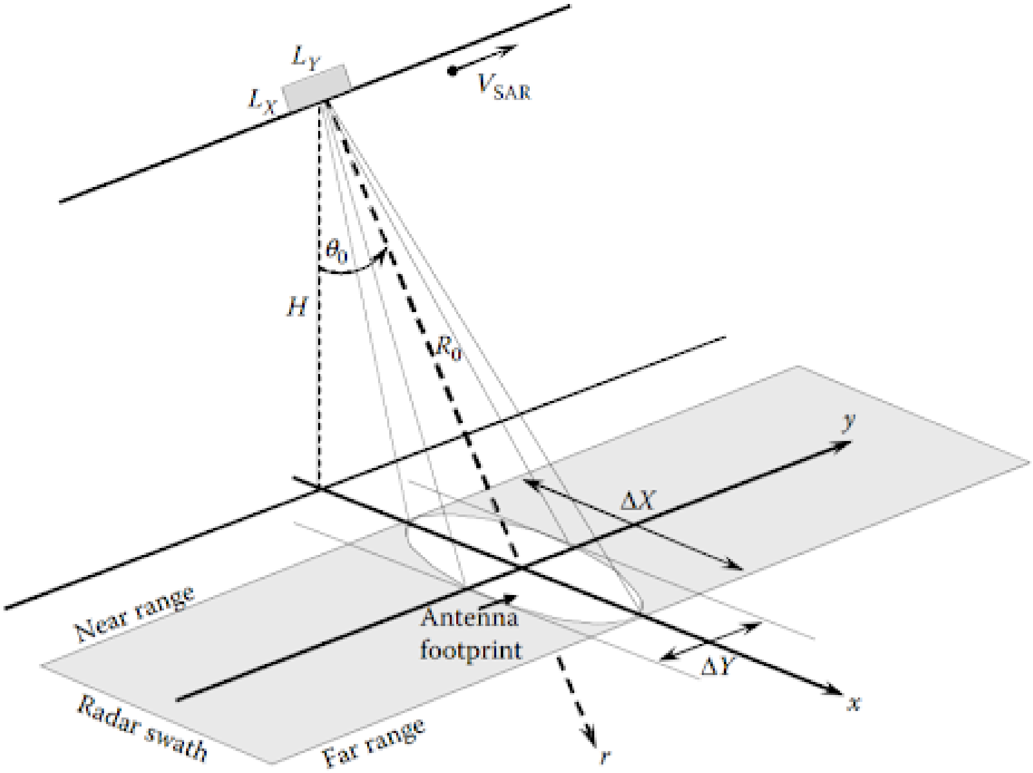


Figure 3.2: SAR system mounted on a moving platform [30]

beam. Hence, a radar with larger aperture is needed to capture high resolution images. But for a really high resolutioned image, the kind of large aperture that is required theoretically, is not feasible practically. Hence, this limitiation has been overcome by a synthetic aperture which simulates a very long antenna by moving and covering the surface. Figure 3.2 shows the working of an SAR imaging system. An SAR imaging system is situated at height H moving with velocity V_{SAR}. The antena is perpendicular to the flight direction. The beam is directed slantwise toward the ground with angle θ_0. The area covered by the antenna beam is called “antenna footprint”. The entire area covered by the radar during a duration is “radar swath”.

Multiple backscattered signals are received by the antenna at the same time from the same target. But since the antenna positions are different, the whole image of the bigger target scene is reconstructed and this image has a much higher resolution. This high resolution image mimicking or synthesizing the behavior of a larger antenna represents the core principle of SAR imaging. SAR systems can be airborne as well as spaceborne. In airborne systems, the radar is mounted on an

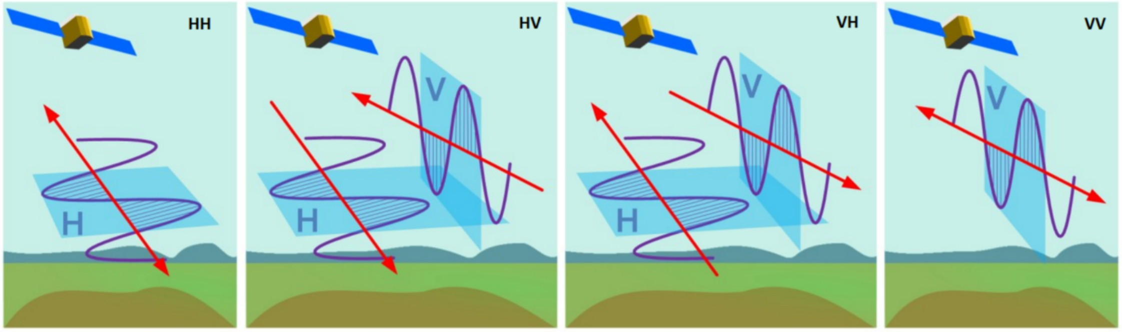


Figure 3.3: Illustration of four different polarization states of scattered waves

aircraft as in AIRSAR and EMISAR. In spaceborne systems, the radar is mounted on satellites as in RADARSAT, ALOS, and ENVISAT.

3.3 Polarimetric SAR

Polarization refers to the plane in which the electric field of the electromagnetic waves propagates. The data is acquired based on the nature of signal used by the radar. This is classified into the polarization channels. These polarization channels are horizontal transmitting and horizontal receiving (HH), horizontal transmitting and vertical receiving (HV), vertical transmitting and horizontal receiving (VH), and vertical transmitting and vertical receiving (VV).

The electromagnetic wave consists of two orthogonal components, electric field and magnetic field varying with time. In the polarimetric SAR, the information of the target is obtained by analyzing the polarization of transmitted and received signals. Most of the radar systems are designed based on horizontal and vertical polarization. If the electric field oscillates in the vertical plane of the wave propagation then it is called vertically polarized, and if it oscillates in the horizontal plane then it is called horizontally polarized. Depending on the shape of the electric field ellipse (ellipticity), the polarization is categorized into main three types: linear, circular, and elliptical. In the case of linear polarization, the electric field oscillates only in one direction. Whereas, in the case of circular and elliptical polarization, the electric field rotates in the direction of wave propagation. Figure 3.3 illustrates different types of polarization. Any PolSAR system operates in one of the three modes - Single polarization (HH or VV), Dual polarization (HH and HV or VH and VV), and Quad/Full polarization (HH, HV, VH, and VV).

The full polarized mode contains much more information than single or dual polarized. Most of the airborne and spaceborne sensors such as EMISAR, AIRSAR, RADARSAT-2, and ALOS-PALSAR operate in full polarized mode. The information of the target can be characterized by observing the behavior of a signal using different polarization channels. When the polarized signal is transmitted to the earth's surface, its polarization states are modified upon interaction with the target. The modification is dependent upon the target properties such as geometry, orientation, and roughness. This even helps to characterize the target better than optical sensors.

3.3.1 Mathematical Preliminaries of PolSAR Data

As discussed above, when the radar illuminates a surface with a horizontally polarized wave, the backscattered wave can either have horizontal or vertical polarization. Same behavior is repeated for the vertically polarized wave illuminated by the radar. This set of backscattering properties is contained in 2 × 2 Sinclair matrix as shown in equation 3.1.

$$S = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \tag{3.1}$$

Here, S_{HH} is for the scattering element signifying horizontal transmission and horizontal reception of polarization channel. The other three terms in the matrix above are to be interpreted in the same manner. Also, $S_{HV} = S_{VH}$ in the case of monostatic backscattering. Hence, S for our case becomes the associated target vector Ω, that can be written as,

$$\Omega = \begin{bmatrix} S_{HH} & \sqrt{2}S_{HV} & S_{VV} \end{bmatrix}^T \tag{3.2}$$

Using equation 3.2, the 3 × 3 Covariance matrix is obtained when the outer product of Ω is calculated along with its conjugate transpose Ω^T. Hence, C_3 is written as,

$$C_3 = \langle \Omega . \Omega^{*T} \rangle = \left\langle \begin{bmatrix} |\Omega_1|^2 & \Omega_1\Omega_2^* & \Omega_1\Omega_3^* \\ \Omega_2\Omega_1^* & |\Omega_2|^2 & \Omega_2\Omega_3^* \\ \Omega_3\Omega_1^* & \Omega_3\Omega_2^* & |\Omega_3|^2 \end{bmatrix} \right\rangle \tag{3.3}$$

Putting values of equation 3.2 in equation 3.3, equation 3.3 becomes,

$$C_3 = \begin{bmatrix} \langle S_{HH}S^*_{HH} \rangle & \langle S_{HH}S^*_{HV} \rangle & \langle S_{HH}S^*_{VV} \rangle \\ \langle S_{HV}S^*_{HH} \rangle & \langle S_{HV}S^*_{HV} \rangle & \langle S_{HV}S^*_{VV} \rangle \\ \langle S_{VV}S^*_{HH} \rangle & \langle S_{VV}S^*_{HV} \rangle & \langle S_{VV}S^*_{VV} \rangle \end{bmatrix} \tag{3.4}$$

3.4 Datasets

This section discusses the five datasets [17] used for all the experiments conducted. The details of these datasets is as given in Table 3.1.

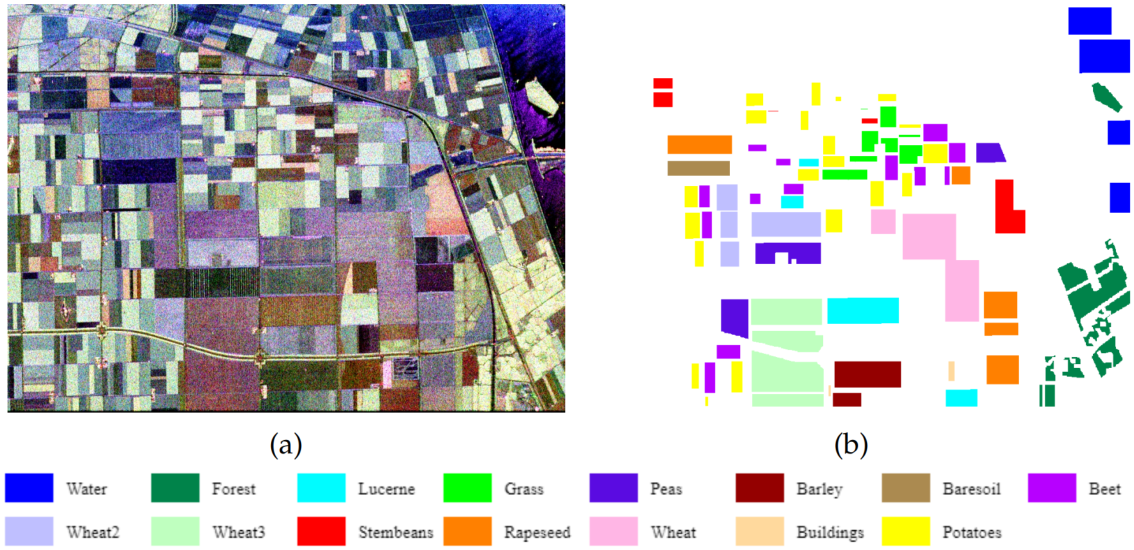
Name	Region	Sensor	Acquired on	Size	No. of Classes
Flevoland15	Flevoland, Netherlands	AIRSAR	16th August, 1989	750 × 1024	15
Flevoland7	Flevoland, Netherlands	AIRSAR	16th June, 1991	750 × 700	7
Landes	Landes, France	AIRSAR	19th June, 1991	1050 × 1000	6
SFAIRSAR	San Francisco Bay Area	AIRSAR	9 April, 2008	900 × 1024	5
SFRS2	San Francisco Bay Area	RADARSAT-2	August, 1989	1800 × 1380	5

Table 3.1: Details of data sets used for experimentation

The first two datasets namely Flevoland15 and Flevoland7 cover agricultural area which contains different crops. This number of classes in Flevoland15 is fifteen and that is Flevoland7 is seven. The Pauli decomposed image along with the ground truth and labels of Flevoland15 is as shown in Figure 3.4.

The Pauli decomposed image along with the ground truth and labels of Flevoland7 is as shown in Figure 3.5.The third dataset namely Landes consists of six classes and its Pauli RGB image, ground truth and labels are as shown in Figure 3.6. The Pauli decomposed image along with the ground truth and labels of both the San Francisco datasets is shown in Figure 3.7 and figure 3.8.

The data representation that is used for all the experiments below is in the form of covariance matrix C_3 which is discussed in the previous section of this chapter. Some other representations are in the form of coherency matrix T_3 which contains a 3_D Pauli feature vector k (shown in equation 3.5 instead of the Ω as in C_3. k contains three elements as shown below.


(a)

(b)

Figure 3.4: Flevoland15 data set with a) Pauli RGB image b) ground truth with 15 classes

$$k = \frac{1}{\sqrt{2}} \left[S_{XX} + S_{YY} \;\; S_{XX} + S_{YY} \;\; 2 \times S_{YY} \right]^{T} \tag{3.5}$$

As discussed further, C_3 consists of six matrix elements out of which three matrices are real valued and three matrices are complex-valued.

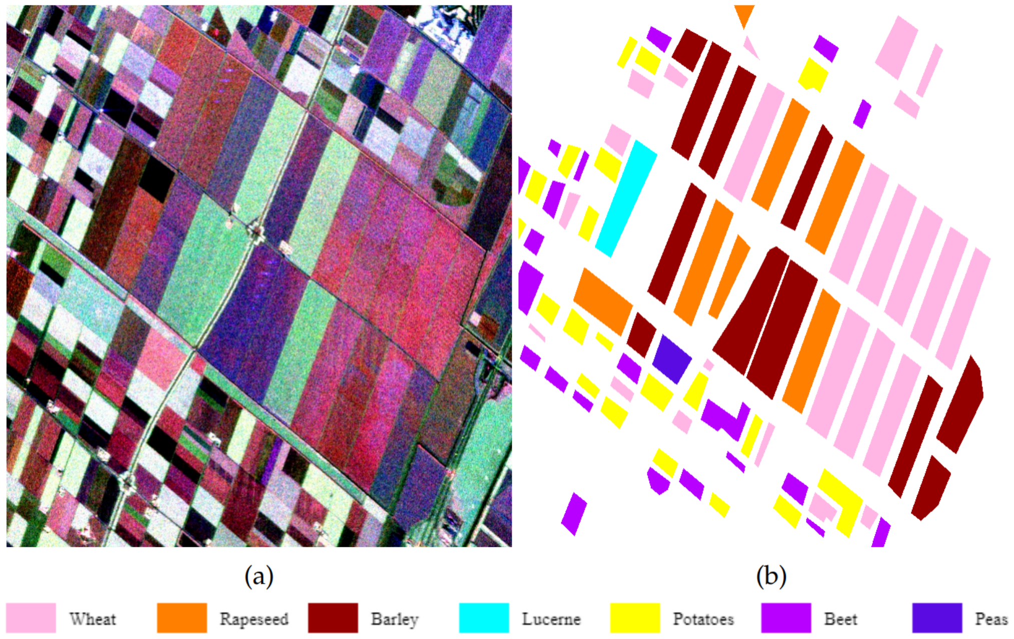

(a)
(b)

Figure 3.5: Flevoland7 data set with a) Pauli RGB image b) ground truth with 7 classes

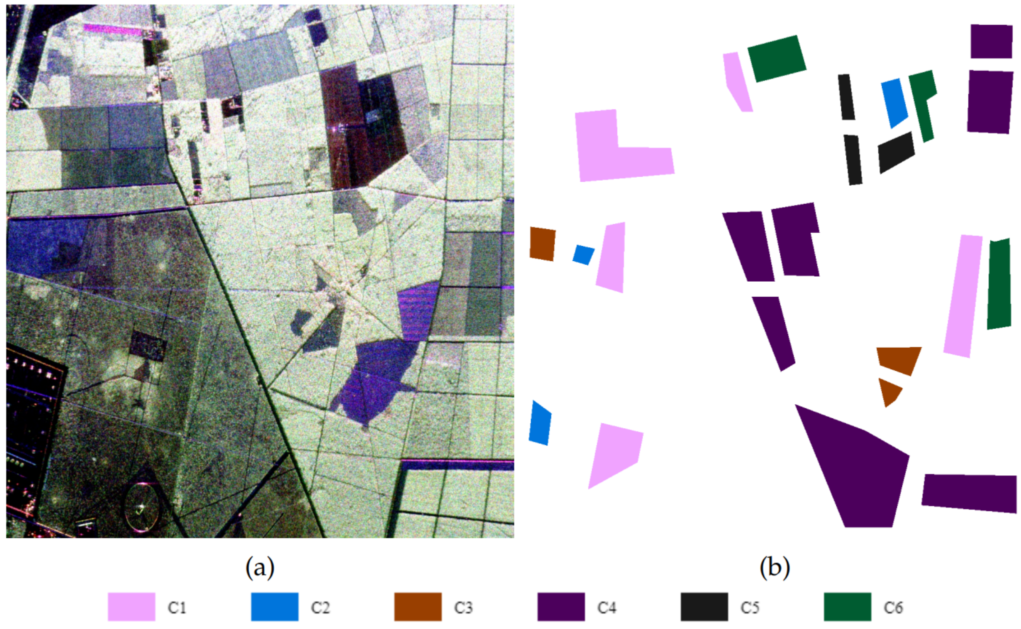

(a)
(b)

Figure 3.6: Landes data set with a) Pauli RGB image b) ground truth with 6 classes

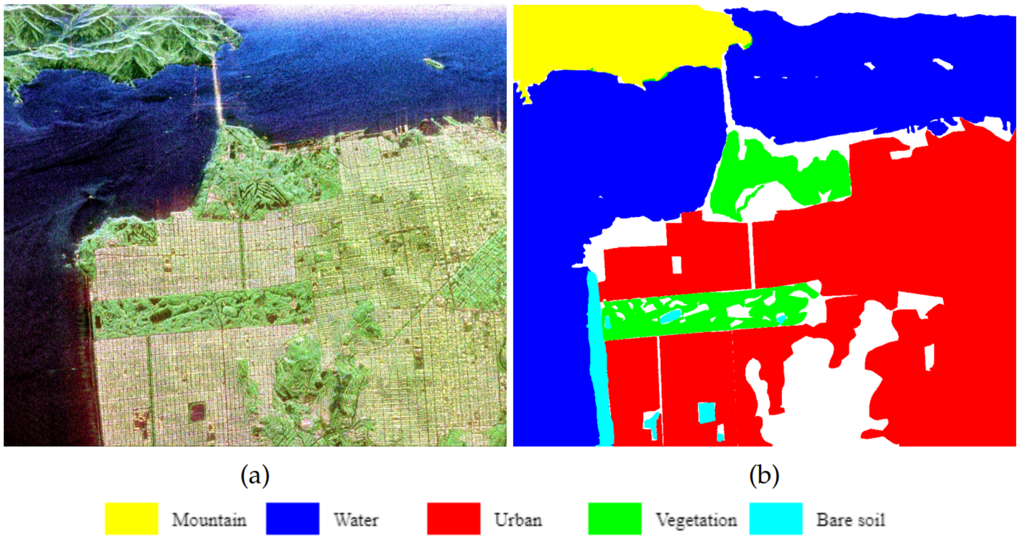

(a) (b)

Figure 3.7: SFAIRSAR data set with a) Pauli RGB image b) ground truth with 6 classes

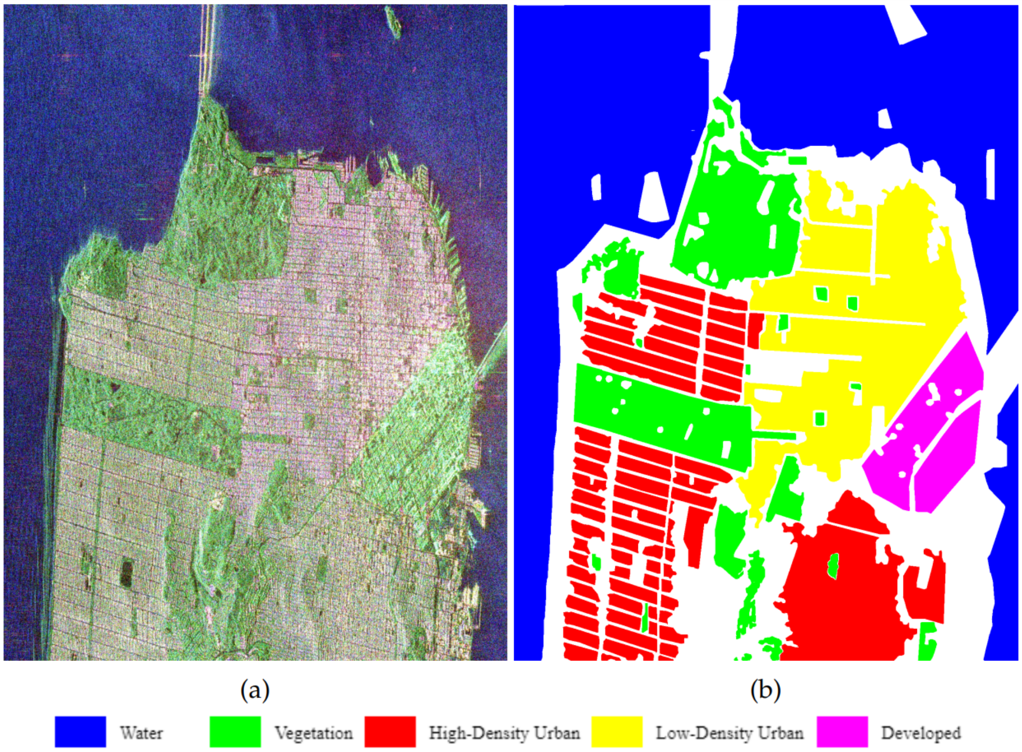

(a) (b)

Figure 3.8: SFRS2 data set with a) Pauli RGB image b) ground truth with 6 classes

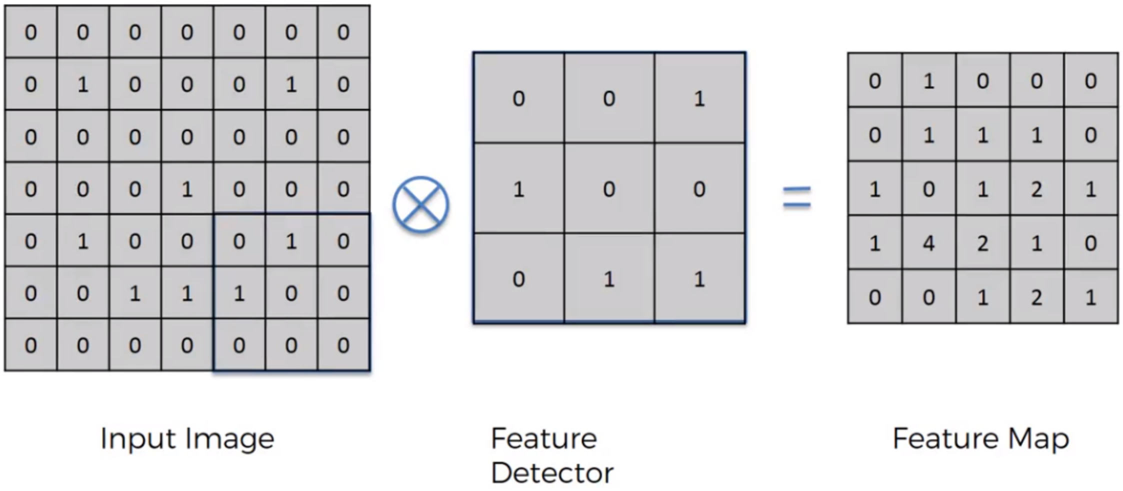
All the five datasets are benchmark datasets. For performing speckle filtering operations discussed in later sections, SNAP tool is used.

3.5 Convolutional Neural Networks in PolSAR Image Classification

Convolutional Neural Networks dominate the image classification task because of the ways in which they operate. Any CNN employs five elemental operations namely convolution, activation, max pooling, flattening, and a fully connected layer in the end.

Convolution

During this step, a portion of the input image is multiplied with the feature detector usually referred to as Kernel as shown in the Figure 3.9. In case of PolSAR images, the portion of input size is chosen as a 12×12 patch. This step generates a feature map which gives us the number of convolved features. The kernels are of highly specific nature (blur detect, edge detect, etc) and multiple kernels can be used during one convolution operation.



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

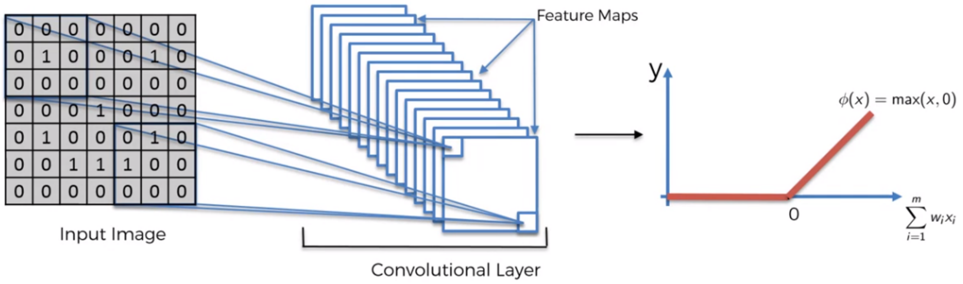
0	0	1
1	0	0
0	1	1

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Figure 3.9: Convolution Operation [36]

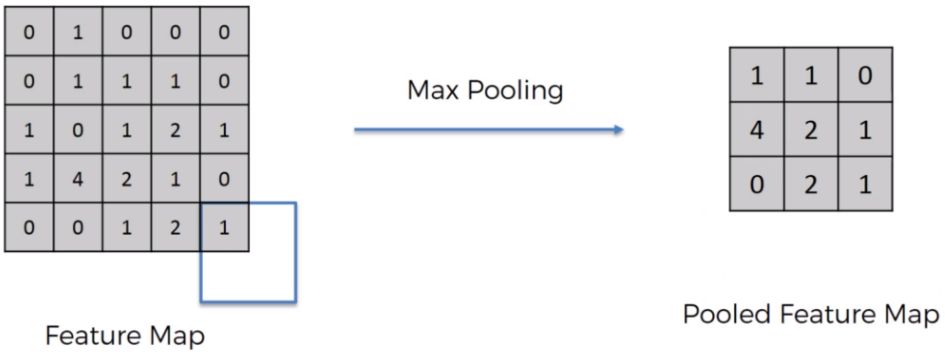
ReLU

An activation function like ReLU is normally used to bring a non-linearity effect. Such non-linearity is required because images are non-linear in nature. The function of ReLU is as shown in equation 3.6,



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Figure 3.10: ReLU [36]



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

1	1	0
4	2	1
0	2	1

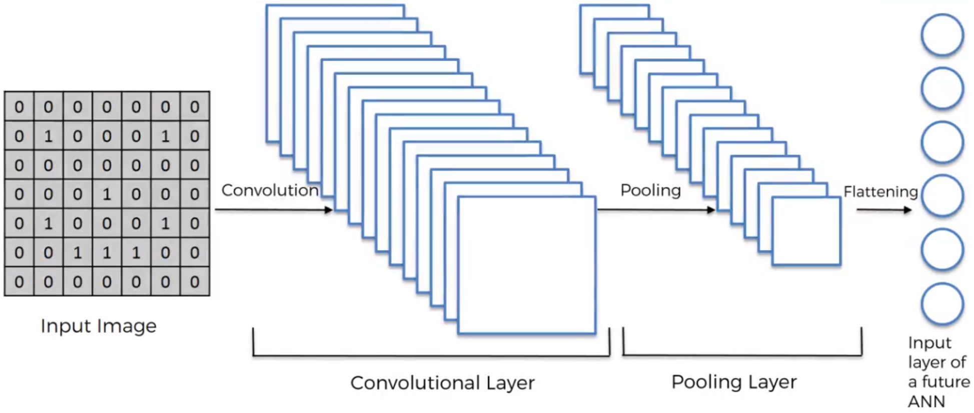
Figure 3.11: Max Pooling [36]

$$ReLU(x) = max(x, 0) \tag{3.6}$$

Using multiple ReLUs with increasing number of epochs makes a lot more sense as compared to other activation functions such as tanh and sigmoid as it learns specific features very well with increased training. Figure 3.10 shows how ReLU fits the CNN architecture.

Max Pooling

When working with multiple images of the same object, it is necessary for the network to be able to extract the said object from various environments at different orientations. For this very purpose, max pooling is used. A 2×2 max pool window returns the maximum value of the window which is traversed over the feature map. This process can be seen in Figure 3.11.



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Figure 3.12: Flattening [36]

Flattening

Flattening is performed to arrange the feature map vector in a single column. A single columned vector is needed at this stage because it is to be fed to a fully connected layer for further processing. Flattening is described in Figure 3.12.

Fully Connected Layer

The output of the flattening layer is fed into a fully connected layer for the purpose of combining all the features into specific attributes for the class prediction task.

The final probability after each iteration or epoch is decided by Softmax function. And for further reducing the loss obtained using losses such as Cross Entropy loss, optimizers such as Adam Optimizer are used.

The entire process of CNNs can be summarized as in Figure 3.13 where all the layers discussed above individually come together for performing the classification task.

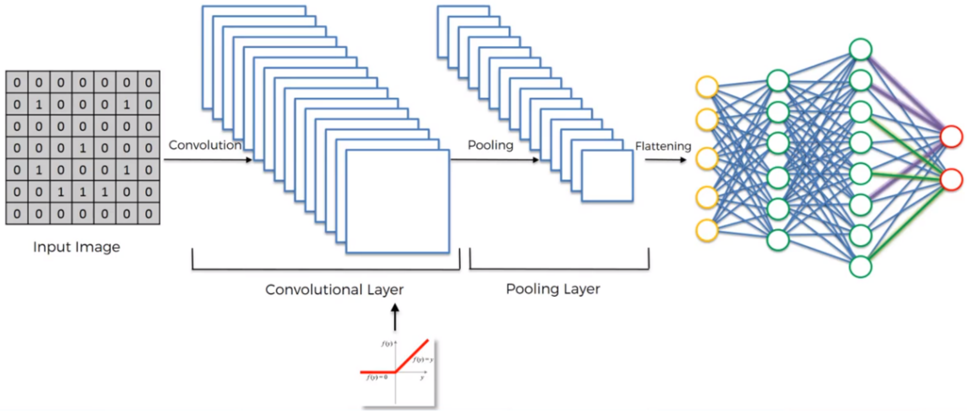


Figure 3.13: All the steps of a CNN [36]

A conventional CNN is designed to operate only on real values. And the PolSAR data is represented in terms of complex values. Various studies as shown in previous sections, have been done using real-valued CNNs however, using complex-valued CNNs with each operation discussed above, equipped to handle complex values, makes more sense.

Importance of Batch Normalization

PolSAR data follows Wishart distribution [29] and CNNs are shift invariant in nature. Hence, performing batch normalization aids the network in understanding the particular distribution nature of the data in a better way. In all the implementations discussed from now on, batch normalization has been used at least once to induce the effect of Wishart distribution for the classification task.

CHAPTER 4

Complex Valued Convolutional Neural Networks

Complex-valued CNN is a variation of conventional CNN hierarchical model where instead of real values, the CV-CNN network can handle the complex values. This is in line with how PolSAR data is represented which is in the form of complex numbers. So, the proposed models in this report contain the usual layers such as input layer, several combinations of convolutional layers followed by pooling layers, fully connected layers, and the classifier layer in the end, just like any other CNN model, but the layers are specifically adjusted mathematically for handling complex numbers.

Input to a CNN model is in the form of 2-D matrix or multiple 2-D matrices distributed over various channels of an image. Then, the convolution layer performs the feature extraction part. This output is then fed to the nonlinear activation to generate the feature maps. This is then followed by down-sampling done in the pooling layer. The output is multi-channel 1-D or 2-D matrices.

4.1 Various Operations in Complex-Valued CNNs

In a CV-CNN, the elements discussed above need to be fully CV compatible. Following sections throw some light on how complex data is handled in various CV-CNN layers.

Convolution

A complex filter matrix $w_{ik}^{(l+1)} \in C^{B_1 \times H_1 \times K}$ is convolved over a complex vector $M_k^{(l)} \in C^{F \times F \times K \times I}$ which represents previous layer's input feature maps. This convolution is calculated as follows.

The output feature map of each layer after Rectified Linear Unit (ReLU) activation is calculated using equation 4.1.

$$M_i = ReLU(\Re(W_i^{l+1})) + iReLU(\Im(W_i^{l+1})) \tag{4.1}$$

The weighted sum of inputs in layer $l+1$ to the *ith* output feature map M_l is calculated as follows.

$$\begin{aligned}
W_i^{l+1} &= \sum_{k=1}^{K} w_{ik}^{l+1} * M_k^{(l)} + b_i^{l+1} \\
&= \sum_{k=1}^{K} (\Re(w_{ik}^{(l+1)}) \cdot \Re(M_k^{(l)})) - \Im(w_{ik}^{(l+1)}) \cdot \Im(M_k^{(l)}) \\
&+ j \sum_{k=1}^{K} (\Re(w_{ik}^{(l+1)}) \cdot \Im(M_k^{(l)})) + \Im(w_{ik}^{(l+1)}) \cdot \Re(M_k^{(l)}) \\
&+ b_i^{l+1}
\end{aligned} \tag{4.2}$$

Pooling

For performing max pooling in complex domain, firstly the magnitude of the neighbouring complex numbers is calculated and then as per the rule of max pooling, the complex number with maximum magnitude is selected to replace the entire $N \times N$ pooling cell.

Fully Connected Layer

The output feature map for fully connected layer remains same as shown in equation 4.1. The equation for weighted sum W for complex-valued fully connected layers can be written as,

$$W_i^{l+1} = \sum_{k=1}^{K} w_{ik}^{l+1} \cdot M_k^{(l)} + b_i^{l+1} \tag{4.3}$$

Complex Backpropogation

The total classification error in case of complex valued CNN model is calculated using output M of each layer which is in complex domain and the label L which is also represented in complex domain. Hence, the total classification error E is calculated as,

$$
\begin{aligned}
E =& \frac{1}{2}\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}[(\Re(L_k[n]) - \Re(M_k[n]))^2 \\
&+ (\Im(L_k[n]) - \Im(M_k[n]))^2]
\end{aligned}
\tag{4.4}
$$

To minimize E in equation 4.4, weights and bias are iteratively adjusted using the equations below.

$$
\begin{aligned}
w_{ik}^{(l+1)}[t+1] &= w_{ik}^{(l+1)}[t] + \Lambda w_{ik}^{(l+1)}[t] \\
&= w_{ik}^{(l+1)}[t] - \eta \frac{\partial E[t]}{\partial w_{ik}^{(l+1)}[t]}
\end{aligned}
\tag{4.5}
$$

$$
\begin{aligned}
b_{i}^{(l+1)}[t+1] &= b_{i}^{(l+1)}[t] + \Lambda b_{i}^{(l+1)}[t] \\
&= b_{i}^{(l+1)}[t] - \eta \frac{\partial E[t]}{\partial b_{i}^{(l+1)}[t]}
\end{aligned}
\tag{4.6}
$$

4.2 Implemented Methods

In this work, we mainly explore five CNN models. Out of which, two are real valued CNN models and the other three are similar complex valued CNN models with variations just in terms of layers. In the experiments demonstrated here, covariance matrix C_3 is used. The data is C_3 is distributed over 6 channels as follows,

$$
C_3 = (C_{11}, C_{12}, C_{13}, C_{22}, C_{23}, C_{33}
\tag{4.7}
$$

The two RV-CNNs are very different in terms of the kind of inputs they take. In equation 11, C_{11}, C_{22}, and C_{33} contain only the real part and imaginary part is 0. On the other hand, C_{12}, C_{13} and C_{23} contain complex valued pixels with both real and imaginary parts. The entire image is divided into 12×12 patches that are fed into the CNN model. Depending on the size of the data, this size can be experimented with, for optimal performance.

4.2.1 Real Valued CNN implementation

The first implementation of RV-CNN architecture is shown in Figure 1. It takes a 6-D feature vector which contains only the real values. Hence, data from C_{11}, C_{22}, and C_{33} is taken as it is and only the real part of C_{12}, C_{13} and C_{23} into this feature vector shown in equation 4.8.

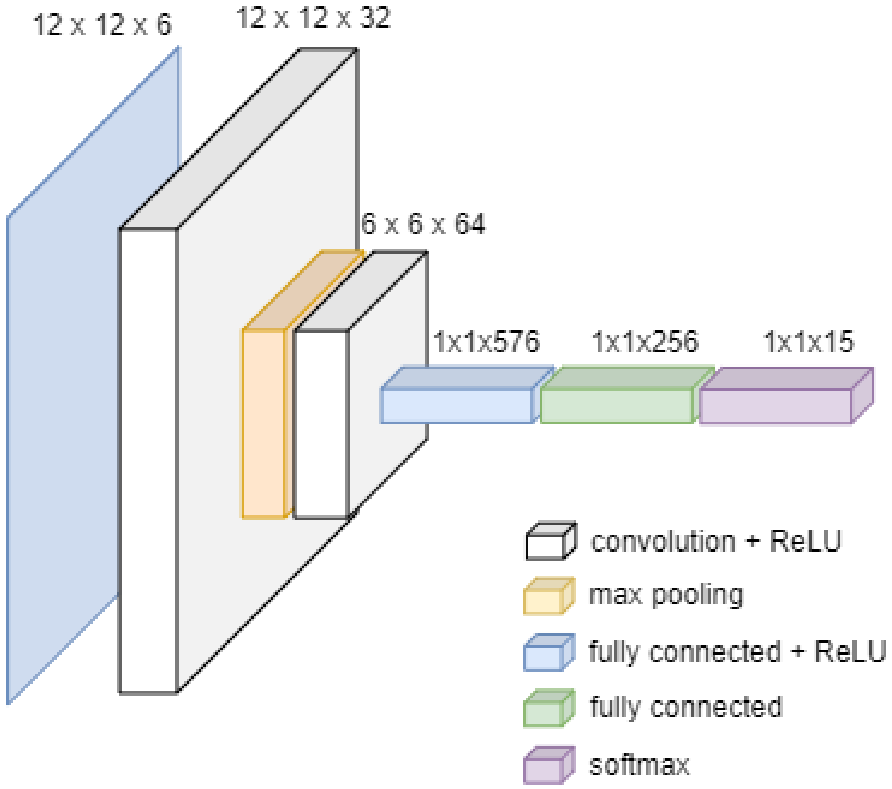


Figure 4.1: Real-Valued CNN model

$$\begin{aligned} data = (&\Re(C_{11}), \Re(C_{12}), \Re(C_{13}), \\ &\Re(C_{22}), \Re(C_{23}), \Re(C_{33})) \end{aligned} \tag{4.8}$$

The input layer of $12 \times 12 \times 6$ is fed into the network. Here, 6 signifies the number of channels which signifies the 6 vectors discussed in equation 4.7. The first convolution layer uses 32 filters to extract features after which this input feature map is shrunk using a 2×2 max-pooling. Then a convolution layer with 64 filters is implemented on this shrunken feature map. This is followed by 2 fully connected layers where ReLU activation is applied. The last fully connected layers extracts features using 128 filters after which the Softmax classifier classifies the data into the number of classes in the given dataset.

The second implementation considers entire data mentioned in equation 4.7 but feeds it into RV-CNN even though there is complex data present. This is achieved by constructing a 9-D feature vector where real and imaginary parts are

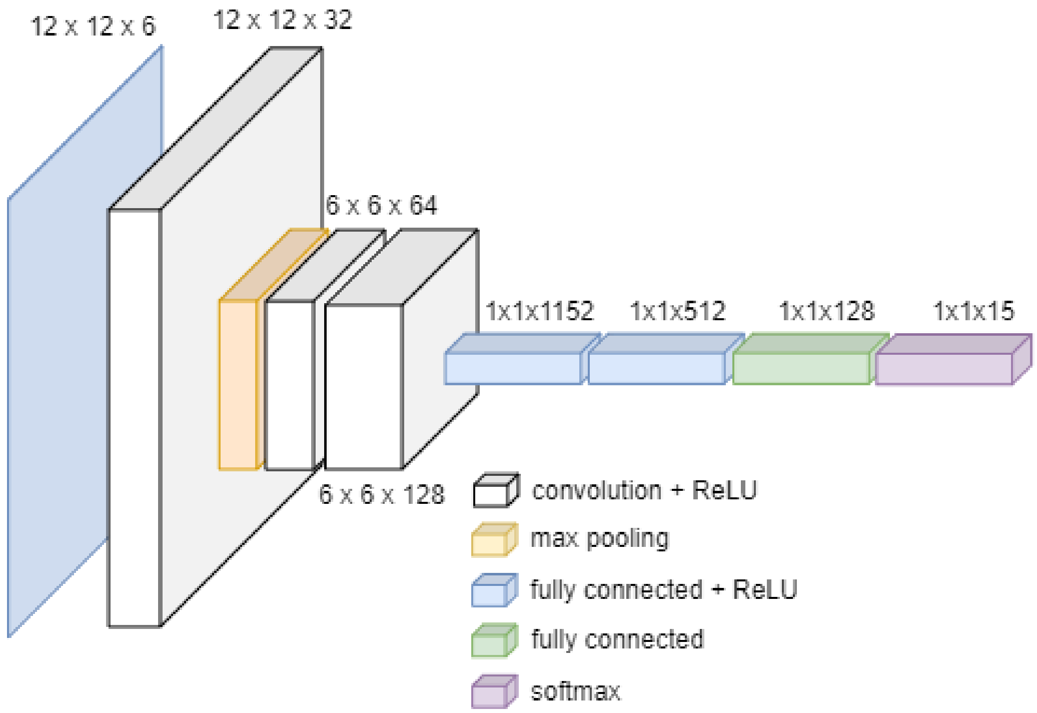


Figure 4.2: Complex-Valued CNN model (Model 3 of CV-CNN variations)

fed separately. This 9-D feature vector is as follows,

$$\begin{aligned} data = (&\Re(C_{11}), \Re(C_{12}), \Re(C_{13}), \\ &\Re(C_{22}), \Re(C_{23}), \Re(C_{33}), \\ &\Im(C_{12}), \Im(C_{13}), \Im(C_{23})) \end{aligned} \tag{4.9}$$

The feature vector in equation 4.9 is fed into the architecture shown in Figure 4.1 as explained above.

4.2.2 Complex Valued CNN implementation

The complex-valued CNN is implemented as per the architecture of individual layers discussed in the previous sections. Here, the network is fed a 6-D feature vector where each pixel is a complex number. The input data representation for this network is as follows,

$$\begin{aligned} data = (&\Re(C_{11}), \Re(C_{12}) + \Im(C_{12}), \\ &\Re(C_{13}) + \Im(C_{13}), \Re(C_{22}), \\ &\Re(C_{23}) + \Im(C_{23}), \Re(C_{33})) \end{aligned} \tag{4.10}$$

As shown in Figure 4.2, the input layer of 12 × 12 × 6 is fed into the complex valued CNN network.The first convolution layer uses 32 complex filters to extract features after which this complex input feature map (M in equation 4.1) is shrunk using a 2 × 2 max-pooling. Then, two more convolution layers are applied to further extract the features using 64 and 128 complex filters. This feature map is then fed into two CV compatible fully connected layers where ReLU activation is applied. The last fully connected layer extracts features using 128 filters after which the Softmax classifier classifies the data into the number of classes in the given dataset.

4.2.3 Variations of CV-CNN

The architecture discussed in Figure 4.2 contains three convolution layers and three fully connected layers. Before arriving at this model, experiments were carried out with other variations also. As shown in Figure 4.3, the first CV-CNN model (Model 1) consists of 2 convolution layers and 2 fully connected layers followed by the Softmax classifier in the end.

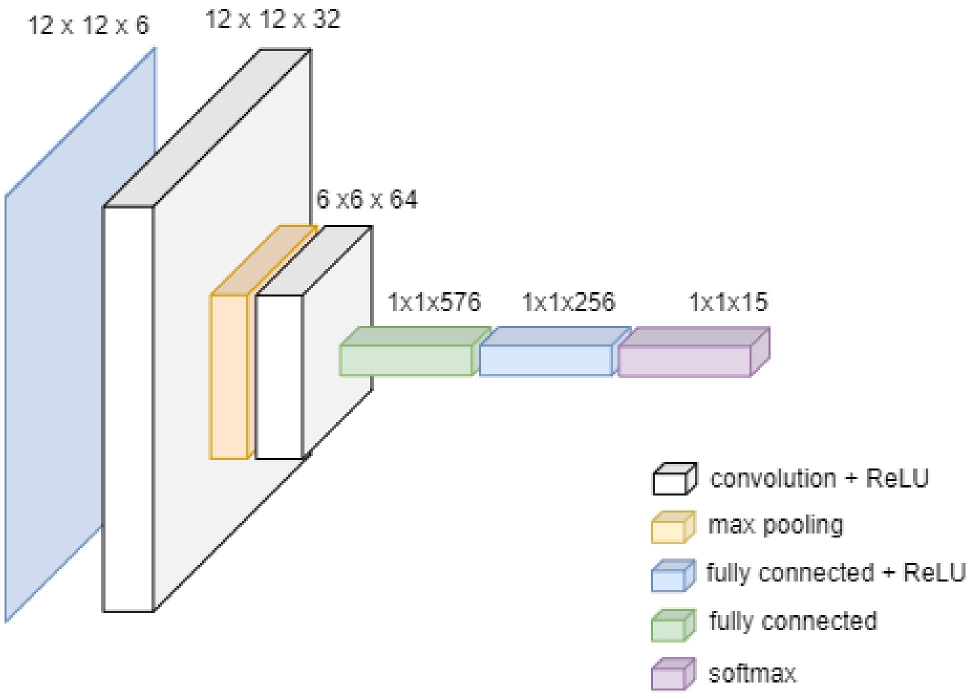


Figure 4.3: Model 1 of CV-CNN variations

The depth of this model 1 is increased in the the second CV-CNN implementation (Model 2) which consists of 3 convolution blocks followed by 2 fully connected layers as shown in Figure 4.4. This depth is further increased in Model 3 which is the one shown in Figure 4.2.

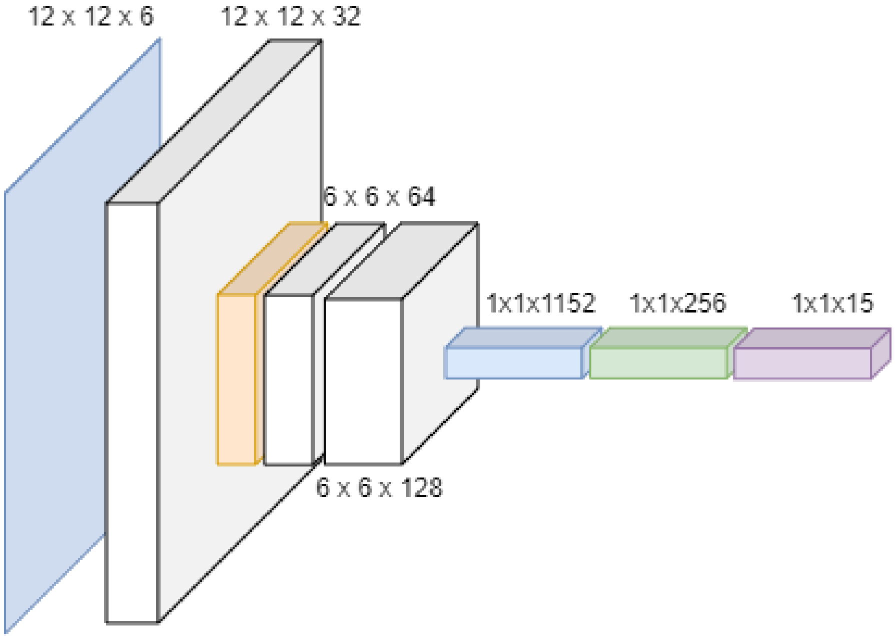


Figure 4.4: Model 2 of CV-CNN variations

4.3 Experimental Results

Result of all the three CV-CNN models on Flevoland15 is shown in Table 4.1. Although the CV-CNN models are able to give comparable results, these could not perform better than the real valued CNNs (results shown in Table 4.2) on Flevoland15 dataset for the three models explored in this report.

Table 4.2 shows the comparison of Wishart Mixture Model (WMM) classifier [7] with the two RV-CNN architectures explored in this report. The WMM models chosen for comparison are those that use unfiltered data. The real-valued CNN model which used 9-D feature vector as input is shown to perform better than the normal RV-CNN model shown in Figure 4.1. It is clear that all the CNN implementations of this report give better results than statistical methods on raw unfiltered data.

Figure 4.5 shows the classification output for Flevoland15 generated by Model 3 of CV-CNN on the right side. The figure also shows the ground truth of Flevoland15 on the left.

Flevoland15 Results on CV-CNN			
Class Label	**CV-CNN (Model 1)**	**CV-CNN (Model 2)**	**CV-CNN (Model 3)**
Water	86.36	91.18	100.00
Forest	87.76	86.84	78.72
Lucerne	91.67	100.00	85.00
Grasses	35.29	50.00	43.75
Peas	90.91	100.00	100.00
Barley	87.50	100.00	82.35
BareSoil	80.00	0	100.00
Beat	91.30	96.30	95.45
Wheat2	30.77	76.17	100.00
Wheat3	97.87	90.20	88.00
Steambeans	92.31	100.00	100.00
Rapeseed	50.00	37.50	5.88
Wheat	86.11	76.47	94.44
Buildings	50.00	0	100.00
Potatoes	96.67	94.44	100.00
OA	**80.219**	**82.417**	**81.868**

Table 4.1: Various Architectures on Flevoland15 Dataset

Flevoland15 Result Comparison				
Class Label	**WMM K-means**	**WMM Global K-means**	**RV-CNN**	**RV-CNN (R+C)**
Water	95.65	100.00	89.20	89.26
Forest	59.00	59.35	100.00	94.87
Lucerne	52.23	52.27	91.30	94.44
Grasses	33.96	33.83	31.25	36.84
Peas	47.49	47.51	94.12	100.00
Barley	54.43	54.39	94.12	100.00
BareSoil	65.59	66.12	0	16.67
Beet	45.16	45.17	88.00	94.74
Wheat2	33.74	33.75	80.00	86.96
Wheat3	46.79	46.88	92.19	97.37
Steambeans	50.80	50.51	78.57	88.89
Rapeseed	34.84	34.49	25.00	37.50
Wheat	46.44	46.84	73.68	90.48
Buildings	40.46	46.15	100.00	0.00
Potatoes	55.28	55.83	94.59	93.18
OA	**49.92**	**50.04**	**81.593**	**84.89**

Table 4.2: Comparison of RV-CNN with Statistical Methods over Unfiltered Data

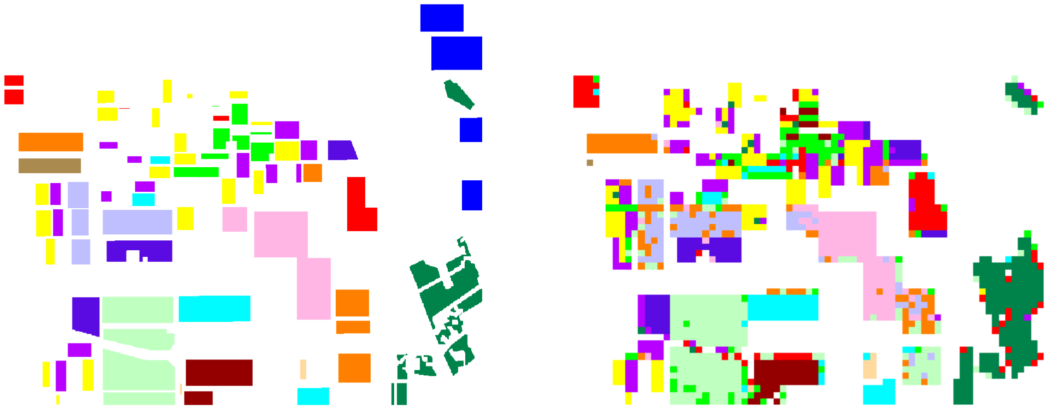


Figure 4.5: Flevoland15 - Ground Truth vs. Results on Model 3

Flevoland7					
Labels	**RV-CNN**	**RV-CNN (R+C)**	**CV-CNN (Model 1)**	**CV-CNN (Model 2)**	**CV-CNN (Model 3)**
Water	90.30	99.23	99.22	99.22	99.22
Rapeseed	96.30	100.00	98.57	100.00	97.01
Barley	85.83	98.20	98.10	95.45	99.09
Lucerne	100.00	77.78	100.00	83.33	91.67
Potatoes	96.43	97.73	100.00	100.00	100.00
Beet	87.80	86.21	86.84	89.19	75.68
Peas	85.71	66.67	100.00	100.00	100.00
OA	**90.102**	**96.701**	**97.716**	**96.954**	**96.447**

Table 4.3: Various Architectures on Flevoland7 Dataset

The result of all the five models on Flevoland7 Dataset is shown in Table 4.3. Model 1 and Model 2 of CV-CNN architecture perform better than both the Real-Valued CNNs on Flevoland7 data. Model 1 discussed in Figure 4.3 performs the best.

The result of all the five models on Landes Dataset is shown in Table 4.4. All the three models of CV-CNN architecture perform better than both the Real-Valued CNNs on Landes data. Model 2 discussed in Figure 4.4 performs the best.

Figure 4.6 shows the classification output for Flevoland7 generated by Model 3 of CV-CNN on the right side. The figure also shows the ground truth of Flevoland7 on the left.

Figure 4.7 shows the classification output for Landes generated by Model 3 of CV-CNN on the right side. The figure also shows the ground truth of Landes on the left.

Figure 4.8 shows the result generated by Wishart classifier explored in [7]. The results shown in this figure are densely pixelated because the classification is at

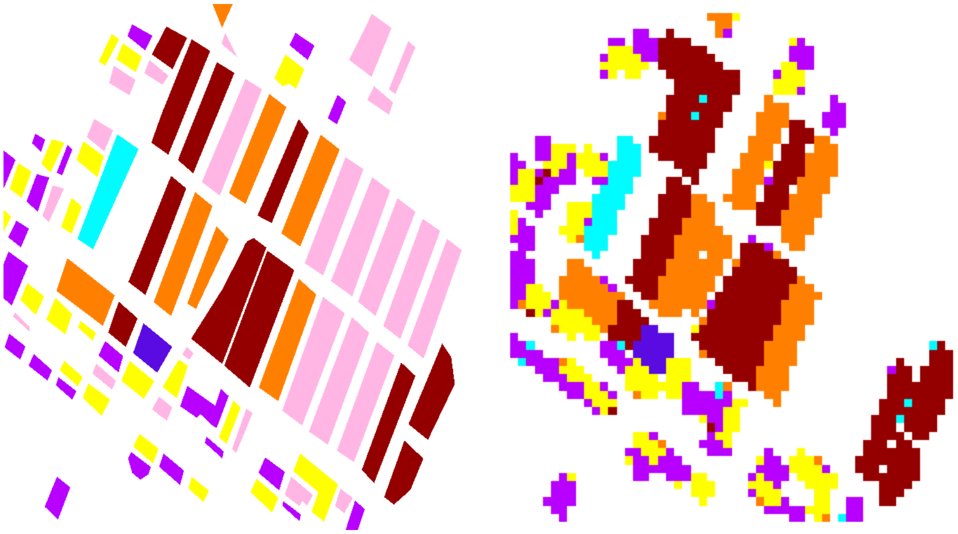


Figure 4.6: Flevoland7 - Ground Truth vs. Results on Model 3

Landes					
Labels	**RV -CNN**	**RV-CNN (R+C)**	**CV-CNN (Model 1)**	**CV-CNN (Model 2)**	**CV-CNN (Model 3)**
C1	70.27	87.39	93.46	91.49	92.66
C2	95.00	100.00	100.00	90.00	100.00
C3	87.50	100.00	100.00	100.00	95.83
C4	99.56	99.49	100.00	100.00	99.44
C5	76.00	80.77	92.00	95.00	96.00
C6	82.98	79.31	76.60	95.35	82.50
OA	**88.546**	**92.731**	**94.885**	**96.675**	**95.396**

Table 4.4: Various Architectures on Landes Dataset

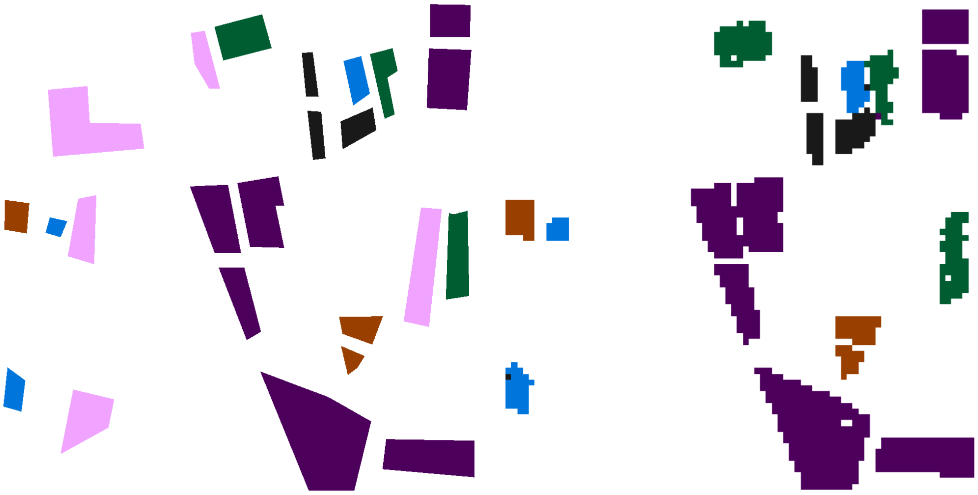


Figure 4.7: Landes - Ground Truth vs. Results on Model 3

pixel level where as the classification of deep learning models shown in results of Figures 4.5-4.7 follow patch-level classification.

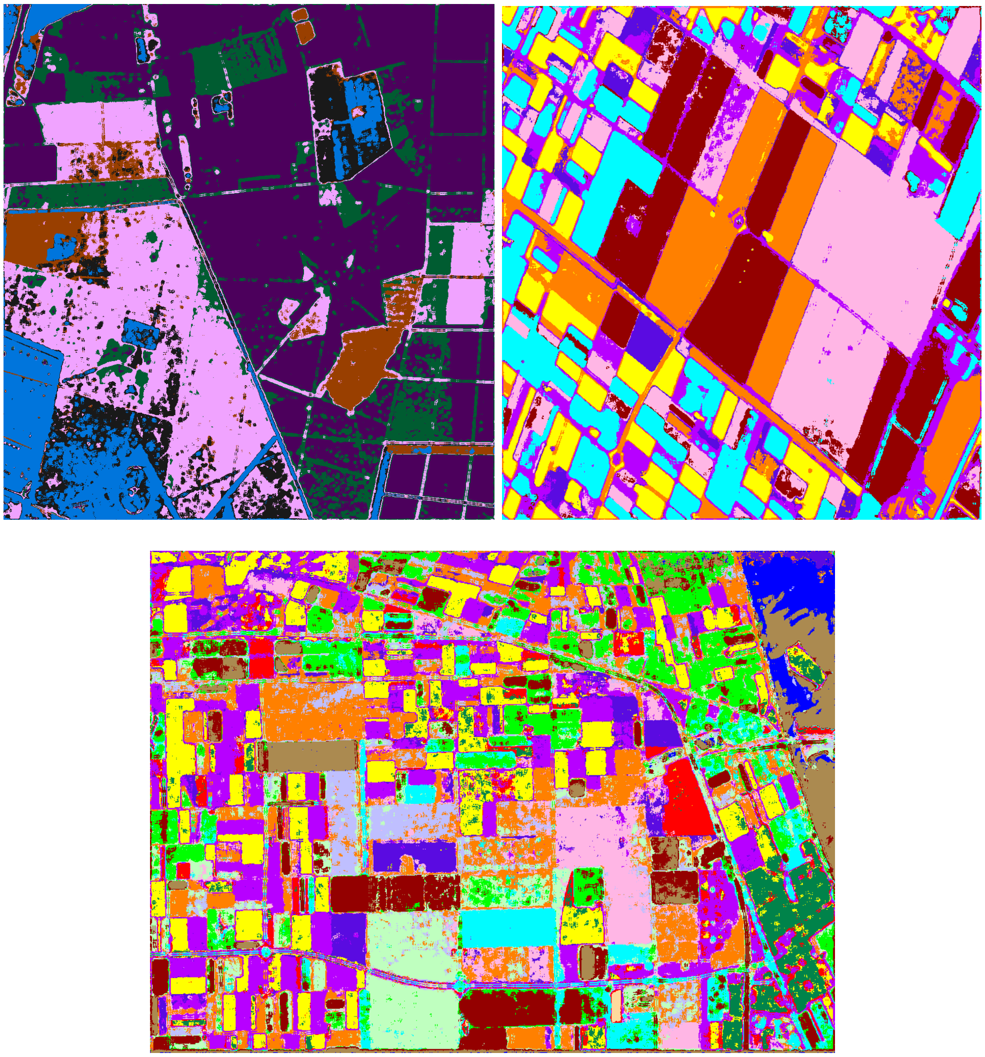


Figure 4.8: Results of Wishart Classifier on Landes, FL7, and FL15

CHAPTER 5
Squeeze and Excitation Networks

Convolutional Neural Network (CNN) focuses mainly on the spatial relationship within local receptive fields. However, the process entangles the channel correlation with spatial information. To address this issue, we use a squeeze-excitation network (SENet) along with complex-valued CNN to exploit the channel interdependencies. Thus, we utilize spatial as well as channel relationships in our work. This, in turn, helps in reducing the speckle noise in the images.

SENet incorporate a mechanism using CNNs only. This mechanism enhances the interpretation of channel interdependencies at no additional computational cost. This improvement is achieved by introducing a content aware mechanism that weighs each channel on a global level adaptively. This is in contrast to a conventional CNN where each channel is weighed equally.

As shown in Figure 5.1, one SE-block consists of three operations as follows:

Squeeze

During squeeze operation, the global information comprised in all the channels is extracted using an averaging operation such as global or adaptive average pooling. The input to this operation is a feature map which is output of the previous convolution layer. The squeeze operation is the key feature of SE blocks as it is essential to extract the global information of the feature map. In our experiments, adaptive average pooling has been used during this operation.

Excitation

Once the feature map is reduced to a smaller dimension by the squeeze operation, this information is now fed into a small deep learning network module during the excitation phase. This module contains a bottleneck structure with two fully connected layers, each followed by their separate activation layers of ReLU and Sigmoid respectively.

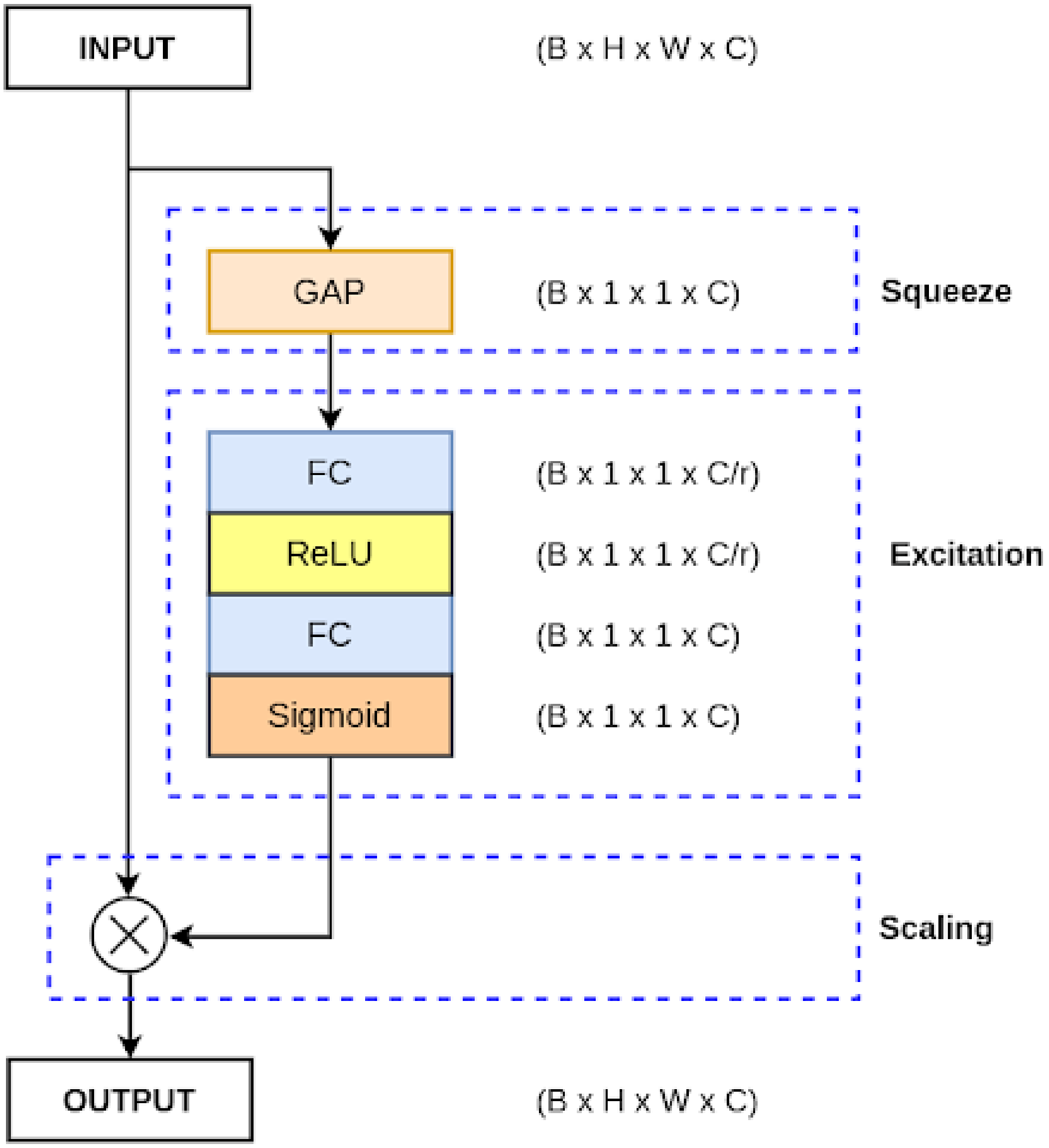


Figure 5.1: SEBlock architecture

Scaling

After the excitation part, during the scaling phase, the output of the Sigmoid layer is multiplied with the input feature map of the convolution layer that was added before the SEBlock. This element-wise multiplication produces the values for each channel between 0 and 1, assigning 0 or lower values to channels that are of less importance as compared to channels with higher values towards 1.

5.1 Skip Connections

A lot of deep learning networks suffer from the problem of vanishing gradient. When this issue occurs, the loss stops decreasing further but it is still far from reaching the desired value. For understanding how skip connections work, we first need to understand the chain rule.

Chain Rule

Suppose the gradient of a loss function z with respect to functions x and y with t as the layer parameter, then let f, g, h be the functions describing different layers on the network such as

$$z = f(x,y) \;\; x = g(t) \;\; y = h(t) \tag{5.1}$$

To express the gradient of z with respect to the input parameters in x and y, the chain rule in multi-variable calculus applies such that,

$$\frac{\partial z}{\partial t} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t} \tag{5.2}$$

Now, when the gradient is calculated, if instead of multiplication as shown in equation 5.2. if addition could be performed, then the problem of vanishing gradient is eliminated. This is precisely what is accomplished using skip connections.

Skip Connections via ResNet

There are two ways in which skip connections can be incorporated. One is through addition and another is through concatenation. In our implementations, skip connections are incorporated through addition by using residual connection block. In

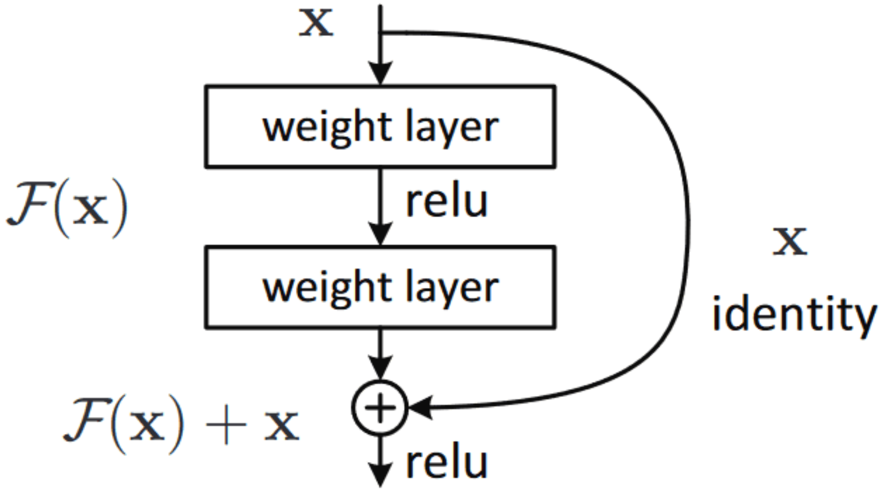

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 5.2: Skip Connections using Residual block

a ResBlock as shown in Figure 5.2, a selected block of network is skipped from the main network and later it's feature map is added to the original path.

5.2 Complex-Valued Squeeze-Excitation Network

The main component of Squeeze-Excitation (SE) Network is the complex SE block [22]. It enhances the convolutional features by modelling channel inter-correlation. It enables significant information propagation towards the next layer. SE block performs squeeze and excitation operation. We modify the conventional SE block to deal with complex values. The squeeze operation is performed by

$$\mathcal{F}_{sq}(M_i) = \frac{1}{H \times W} \sum_{x=1}^{H} \sum_{y=1}^{W} \left(\Re\{M_i(x,y)\} + j\Im\{M_i(x,y)\} \right). \tag{5.3}$$

This is equivalent to average pooling per channel, which represents the channel-wise statistics. The dependencies among the channel is exploited by excitation operation by a gating mechanism with sigmoid activation as

$$e_i = \mathcal{F}_{ex}(s_i) = \frac{1}{1 + e^{-\Re\{W_1' \tilde{s}_i\}}} + j\frac{1}{1 + e^{-\Im\{W_1' \tilde{s}_i\}}} \tag{5.4}$$

where $s_i = \mathcal{F}_{sq}(M_i)$, $\tilde{s}_i = max(W_1 s_i, 0)$. e_i is then multiplied channel-wise with M_i to produce the output feature o_i.

We use the SE block with and without skip connections, separately. While using SE block without skip connection, we use the above mentioned expressions

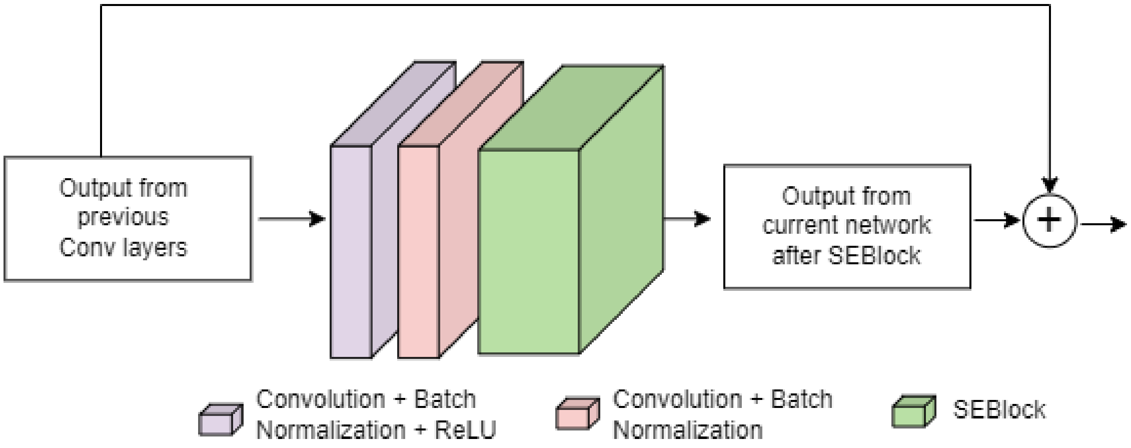

Output from previous Conv layers
Output from current network after SEBlock
+
Convolution + Batch Normalization + ReLU
Convolution + Batch Normalization
SEBlock

Figure 5.3: ResBlock along with SEBlock

to find the feature map o_i. In skip-connection configuration (Figure 5.3), the SE block is associated with two conv blocks. One conv block consists of one CV-CNN layer, batch normalization and complex ReLU. The other conv block consists of one CV-CNN layer along with batch normalization.

The output feature map of this configuration is added with the input map $\hat{M}_i = M_i + o_i$.

The advantage of using SENet for PolSar images is to capture channel-wise dependencies that are learned by extracting non-mutually-exclusive and nonlinear relationships between the individual channels. This is achieved by the excitation module which follows the squeeze model. The main advantage of using a squeeze network is to interpret the local descriptors that capture the statistics for the whole image. One more reason for exploring the efficiency of SE blocks for SAR image classification is that the speckle noise is spread across channels. The SE block helps in reducing the noise by suppressing unwanted information.

5.3 Experimental Results

Three datasets [17] namely Flevoland15, Flevoland7, and Landes have been considered for demonstrating the effectiveness of our architecture. Flevoland15 and Flevoland7 are collected by AIRSAR over Flevoland region of Netherlands. Flevoland15 data is an SAR image of dimensions 750 × 1024 with 15 classes. Flevoland7 data is a 750 × 700 dimensional SAR image captured by AIRSAR with 7 classes. Landes is a 1050 × 1000 SAR image with 6 classes. It is important to note that all the experiments are carried out on unfiltered data. Entire image of the dataset is divided into 12 × 12 patches. All the experiments are carried out using PyTorch framework.

Furthermore, experiments can be divided into two parts. First part explores the efficiency of Complex Valued SENet on the original datasets without any data scaling. The second part explores the same architectures along with data scaling techniques.

5.3.1 SENet with and without skip connections

The datasets are divided into training data and testing data where 25% images have been chosen for testing. Tesla P100-PCIE with 16GB graphics RAM is used to train the models. For Flevoland15, we get 1212 number of patches. Number of patches for Flevoland7 and Landes is 1313 and 1303, respectively. For training, Adam optimizer is used on cross-entropy loss.

Class Label	**WMM K-Means [7]**	**WMM Global K-Means[7]**	**Proposed Net Without Skip Connections**	**Proposed Net With Skip Connections**
Water	95.65	100.00	100.00	73.01
Forest	59.00	59.35	100.00	100.00
Lucerne	52.23	52.27	100.00	100.00
Grasses	33.96	33.83	77.78	81.82
Peas	47.49	47.51	100.00	94.44
Barley	54.43	54.39	66.67	92.31
BareSoil	65.59	66.12	100.00	100.00
Beet	45.16	45.17	100.00	100.00
Wheat2	33.74	33.75	100.00	100.00
Wheat3	46.79	46.88	100.00	100.00
Steambeans	50.80	50.51	100.00	100.00
Rapeseed	34.84	34.49	95.83	100.00
Wheat	46.44	46.84	97.44	100.00
Buildings	40.46	46.15	0	100.00
Potatoes	55.28	55.83	100.00	100.00
OA	**49.92**	**50.04**	**97.12**	**96.04**

Table 5.1: Results of SENet on Flevoland 15 dataset

Table 5.1 shows the comparison between the results obtained using statistical methods [7] and our results. It should be noted that statistical methods produce sub-optimal results on completely unprocessed data. Since, the convolution layers and the squeeze and excitation blocks are employed, the noise spread across the channels gets sufficiently suppressed and therefore isn't propagated through the feature maps. Hence, the proposed deep learning architecture handles the unfiltered data in an efficient way without any denoising techniques. Table 5.1 also shows the comparison between SENet with and without skip connections. On Felvoland15, the network without skip connections gives a higher accuracy of 97.12% than that of network with skip connections which gives accuracy of 96.04%. The proposed network with skip connection performs quite well for almost all the classes except for water. The framework is able to classify the building category, which was missed by without skip-connection configuration. This is because the skip connection allows a smoother information flow from the input towards output.

Table 5.2 shows the results of our method on Flevoland7 and Landes datasets. On Flevoland7, the network without skip connections gives a higher accuracy of 98.78% than that of network with skip connections which gives accuracy of 98.18%. On Landes dataset, the network with skip connections gives a higher accuracy of 97.24% than that of network without skip connections which gives accuracy of 97.24%.

Results of SENet on Flevoland7		
Class Label	**Without Skip Connections**	**With Skip Connections**
Water	97.50	99.24
Rapeseed	100.00	100.00
Barley	100.00	96.67
Lucerne	100.00	100.00
Potatoes	100.00	91.30
Beet	96.67	100.00
Peas	100.00	100.00
OA	**98.78**	**98.18**
Results of SENet on Landes		
C1	98.82	96.51
C2	78.57	100.00
C3	100.00	100.00
C4	100.00	100.00
C5	94.12	95.00
C6	75.00	85.29
OA	**95.71**	**97.24**

Table 5.2: Evaluations on more datasets

Figure 5.4 shows the comparison between results generated by the proposed architecture with and without skip connections for Flevoland7. We also compare with WMM-based method [7]. The left image in Figure 5.4 shows the result generated by the proposed method without skip connections. The right image in Figure 5.4 shows the result generated by the proposed method with skip connections. Whereas the bottom-left image shows the result produced by WMM method [7], and the bottom-right shows the ground truth. One can observe that the WMM method [7] is not able to classify different regions satisfactorily. Further, the method does not label the background separately. Hence, background is classified along with other classes. Moreover, it suffers from the speckle noise. Whereas, our method produces better results by suppressing the effect of speckle noise. Similar comparison can be made from Figure 5.5 for Landes dataset.

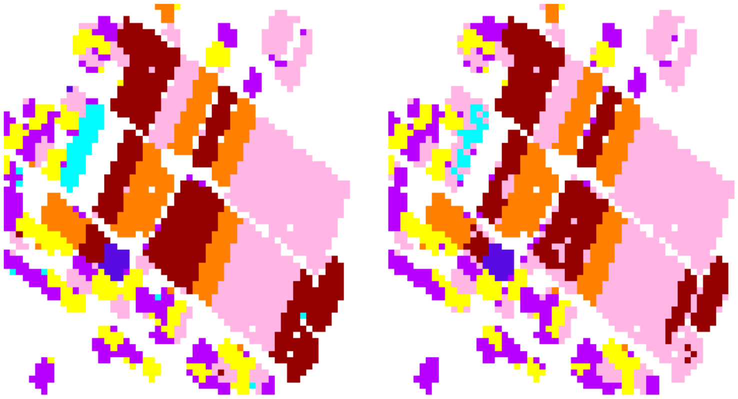


Figure 5.4: Flevoland7 - left: our result without skip connections; right: results with skip connections

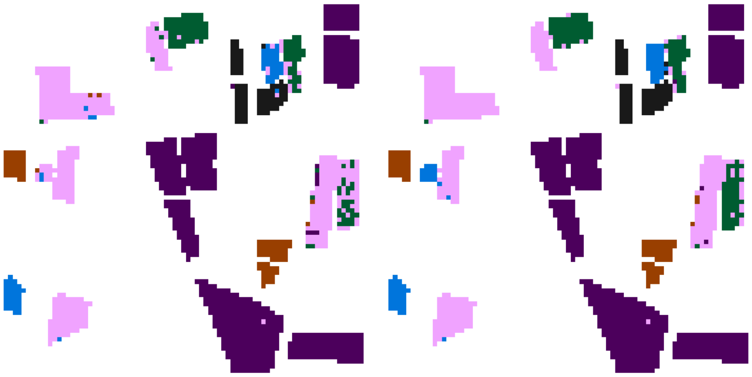


Figure 5.5: Landes - left: our result without skip connections; right: results with skip connections

CHAPTER 6

Data Augmentation for Improved Classification

As seen previously, the size of the datasets that we are dealing with is very small for feeding them into such robust deep learning networks. Hence, by increasing the size of training data, data augmentation has been performed. This section gives a brief idea about various speckle filtering techniques and how they are incorporated in this work for data scaling.

6.1 Speckle Filtering for Polarimetric Data

The reason for appearance of speckle in SAR imagery is the coherent interference of waves that arise due to scattering from surface. Various types of scattering are volume scattering, surface scattering, multiple scattering and wave attenuation. The scattering phenomenon is linear coherent addition of individually scattered waves that are emitted from various point of discrete scatters. Speckle is deterministic in nature and helps in classifying certain land covers. However, Speckle causes significant changes in pixel values by not only appearing in the intensity images of polarization channel, but it also appears in the complex, cross-product terms. Speckle appears in the form of granular noise and hinders an accurate image interpretation as it suppresses small details in the images. Though speckle noise is deterministic in nature, it is to be treated properly using statistical methods.

Polarimetric SAR speckle noise model in [31] clarifies the implications of 2D speckle noise. An adaptive algorithm exploring the effectiveness of mean-shift-based speckle filtering is discussed in [27]. Images with minimum speckle have been produced after applying polarimetric whitening filter (PWF) in [34] and it has produced comparable results. Many a times, during speckle filtering phase, crosstalk between polarization channels is detected. This issue was tackled in [28] and a classification was performed using Bayes maximum likelihood classification algorithm. Since speckle noise also carries terrain information, it is important to preserve the potential target information and it was achieved in [6]. Impact of speckle filtering was studied by performing classification of data using K Means clustering in [11] which also presents a comparative study of several speckle filters that are used widely.

6.1.1 Various Speckle suppression filters

This section throws some light on the Speckle noise filters used in our methods for producing differently filtered images for data augmentation.

Median Filter

It is a non-linear image filtering technique, generally used to eliminate impulse noise such as Speckle noise from an image. The center pixel is replaced with the median value of the filter window. This type of noise removal is a common preprocessing step for improving the output results; one such example is that under some established conditions this filter can help preserve edges while reducing noise.

Frost Filter

This filter borrows from the concept of minimum mean square error algorithm, while adapting to local statistical distribution of the image. The frost filter can be used to mitigate the speckle noise while maintaining the edge information in radar images. This filter has an exponential damping factor. The value that replaces the center pixel is computed from a combination of distance from the filter center, local variance and the damping factor. The exponential damping factor is vital in managing the smoothness of the image.

$$y = \frac{\sum(x * W)}{\sum W} \tag{6.1}$$

where x is the pixel values in the local window, W is the weight for each pixel in the local window.

Boxcar Filter

A boxcar function is equal to a constant value over a range/interval and zero everywhere else. The boxcar filter is roughly based on the boxcar function. The center pixel is replaced by the average of the pixel values of filter window.

$$Y_{i,j} = \langle X_{i,j} \rangle_{N_w} = \frac{1}{N_w^2} \sum_{p=-N_w/2}^{N_w/2} \sum_{q=-N_w/2}^{N_w/2} X_{i+p,j+q} \tag{6.2}$$

where, $Y_{i,j}$ is the final estimate of filtered value at position i, j. $X_{i,j}$ is the value of the input pixel and N_w is the size of window.

Lee Sigma Filter

This filter uses the principle of Gaussian distribution model. Only the pixels belonging to a certain standard deviation range are averaged.

$$Y_{i,j} = \overline{K} + W * (C - \overline{K}) \tag{6.3}$$

where, $Y_{i,j}$ is the despeckled image, $\overline{K}$ is the mean of the window, W is the weighing function and C is the center element in the window. W is calculated using the equation below.

$$W = \frac{\sigma_k^2}{(\sigma_k^2 + \sigma^2)} \tag{6.4}$$

where, σ^2 is the variance of the image and σ_k^2 is the variance of the pixels in the window.

After experimenting with different kinds of combonations for data scaling, boxcar filter and Lee Sigma filter have been selected for data augmentation in this work as discussed below.

For data augmentation, the size of training set has been increased three times as compared to the original dataset. This is achieved using speckle noise suppression techniques discussed in the previous section. Firstly, the boxcar filter is applied on the 75% of original patches selected for training set. Then Lee Sigma filter is applied on the original images in the same way.

$$data\_og = (C_{11}, C12, C_{13}, C_{22}, C_{23}, C_{33}) \tag{6.5}$$

$$data\_box = (C_{11}, C12, C_{13}, C_{22}, C_{23}, C_{33}) \tag{6.6}$$

$$data\_lee = (C_{11}, C12, C_{13}, C_{22}, C_{23}, C_{33}) \tag{6.7}$$

These filtered images are then added with the collection of training patches along with their labels as shown in equation 6.8

$$augmented\_data = (data\_og, data\_box, data\_lee) \tag{6.8}$$

6.2 Experimental Results

After the training set is ready, the training is done using the SENet architecture with and without skip connections as discussed above. The classwise accuracies of employing these techniques on Flevoland15 dataset is shown in Table 6.1. Figure 6.1 shows the set of labels generated for pixels of Flevoland 15 dataset.

When compared with the results in Table 5.1, it can be seen that the accuracy after employing data augmentation has increased for Flevoland15. It should also be noted that for classes with very few labels such as the "buildings" class in Flevoland15, the accuracy without data augmentation was very less and it significantly improves after data scaling.

Class Label	**Without Skip Connections**	**With Skip Connections**
Water	100.00	85
Forest	99.06	100.00
Lucerne	100.00	100.00
Peas	100.00	98
Barley	95.08	98.36
BareSoil	97.37	94.74
Beet	94.44	77.78
Wheat2	98.77	100.00
Wheat3	94.12	97.06
Steambeans	100.00	100.00
Rapeseed	100.00	100.00
Wheat	89.19	94.59
Buildings	100.00	99.17
Potatoes	100.00	100.00
OA	**98.17**	**97.25**

Table 6.1: Results on Flevoland 15 after data augmentation

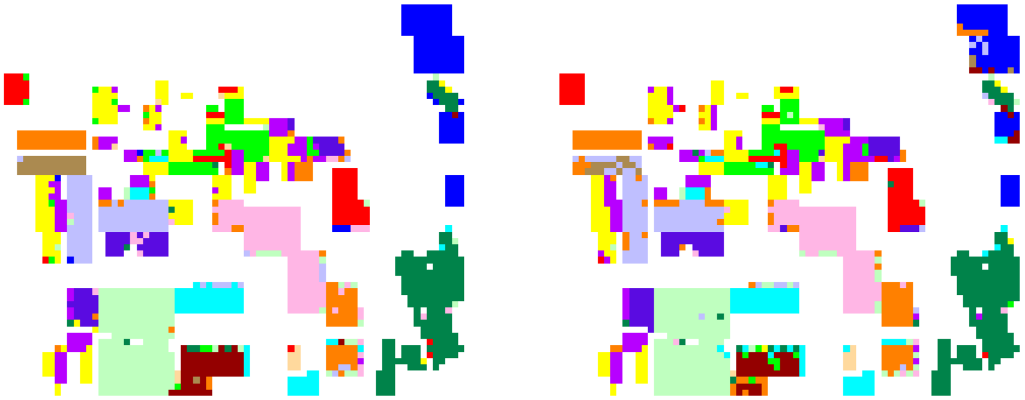


Figure 6.1: After Data Augmentation - left: results without skip connections on F15; right: results with skip connections on F15

Table 6.2 shows the results of classification using various SENet architectures after using Data augmentation techniques on Flevoland7. Figure 6.2 shows the visual result for the result shown in the Table 6.2. As is the case with Flevoland15, data augmentation process has improved the accuracy for Flevoland7 too.

Results on Flevoland7 after Data Augmentation		
Class Label	**Without Skip Connections**	**With Skip Connections**
Water	100.00	100.00
Rapeseed	100.00	100.00
Barley	100.00	99.70
Lucerne	100.00	100.00
Potatoes	100.00	99.15
Beet	96.36	97.27
Peas	100.00	100.00
OA	**99.66**	**99.57**

Table 6.2: Evaluation on Flevoland7

Table 6.3 shows the results of classification using various SENet architectures after using Data augmentation techniques on Landes. Figure 6.3 shows the visual result for the result shown in the Table 6.3. A slight improvement in accuracy for Landes has also been achieved for Landes dataset after data augmentation.

Table 6.4 shows the accuracy achieved on SFAIRSAR for SENets without skip connections without and with data augmentation. Same type of results for SFRS2 have been shown in Table 6.5. Figure 6.4 shows the visual results for the results shown in Table 6.4.

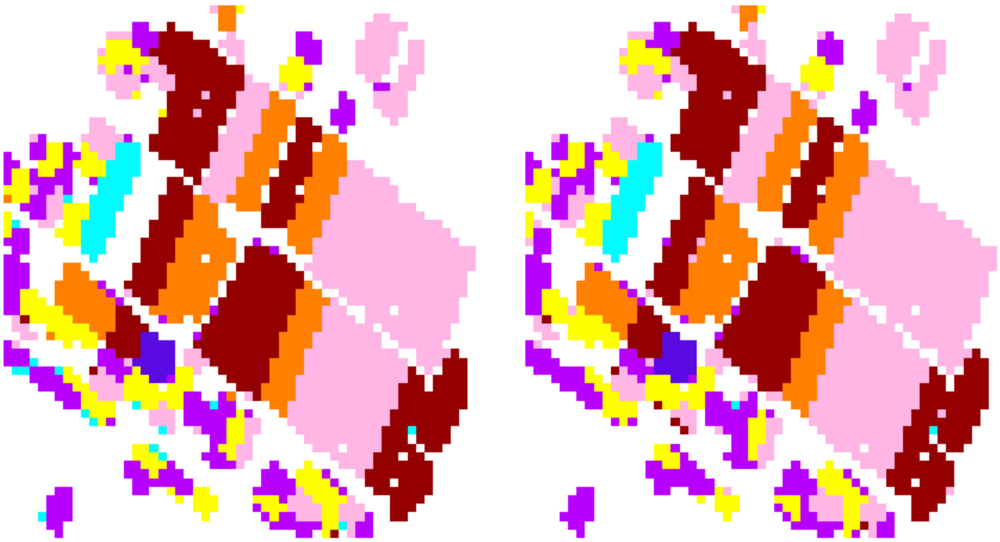


Figure 6.2: After augmentation - left: results with- out skip connections on F7; right: results with skip connections on F7

Results on Landes after Data Augmentation		
Class Label	**Without Skip Connections**	**With Skip Connections**
C1	97.61	96.42
C2	95.35	97.67
C3	100.00	100.00
C4	100.00	100.00
C5	100.00	100.00
C6	96.18	97.71
OA	**98.72**	**98.64**

Table 6.3: Evaluation on Landes

Table 6.4: Experiments on SFAIRSAR dataset

Class Label	**SENet Without Data Augmentation**	**SENet With Data Augmentation**
Mountain	71.19	94.34
Water	98.05	99.81
Urban	99.83	98.67
Vegetation	91.09	86.35
BareSoil	91.35	97.05
OA	**97.22**	**98.30**

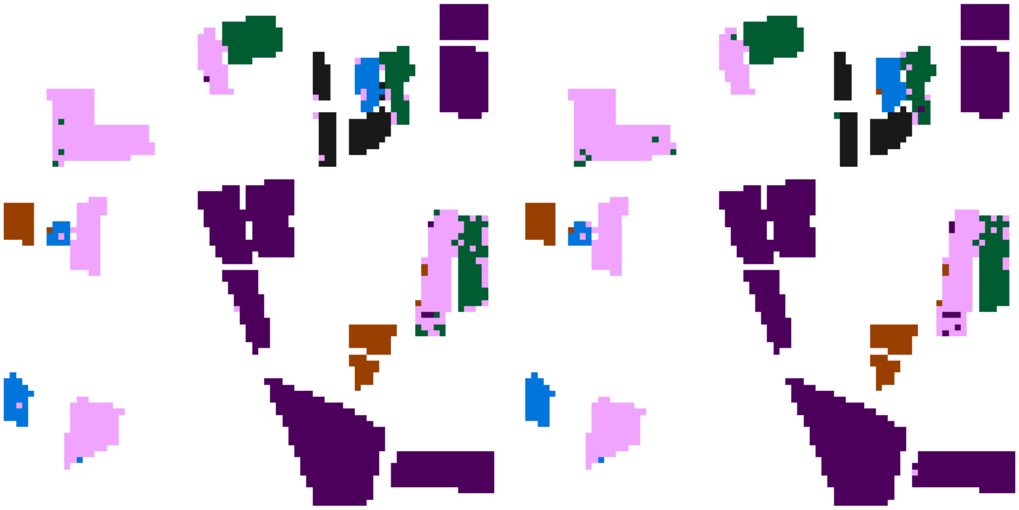


Figure 6.3: After augmentation - left: results with skip connections on Landes; right:results with skip connections on Landes

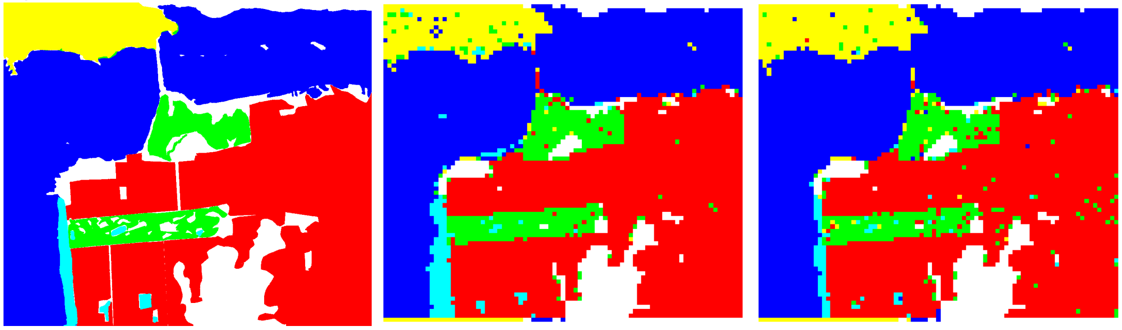


Figure 6.4: Results on SFAIRSAR a) Ground Truth. b) Result of SENet without data augmentation. c) Result of SENet with data augmentation

Table 6.5: Experiments on SFRS2 dataset

Class Label	**SENet Without Data Augmentation**	**SENet With Data Augmentation**
Water	100.00	100.00
High Density Urban	97.60	97.06
Low Density Urban	97.57	97.42
Developed	92.97	97.55
Forest	96.32	98.66
OA	**98.41**	**98.84**

CHAPTER 7

Summary and Comparison

To summarize all the models implemented in this work, we have divided the implementations in terms of CNN architectures and SENet Architectures.

7.1 Convolutional Neural Networks

As discussed previously, five models for CNNs have been implemented. These are as shown in Table 7.1.

Model	Real/Complex	Input Channels	Layers
RV-CNN	Real	6	5
RV-CNN	Real	9	5
CV-CNN	Complex	6	5
CV-CNN	Complex	6	6
CV-CNN	Complex	6	7

Table 7.1: All CNN models

7.2 Squeeze-and-Excitation Newtorks

As discussed previously, four variations of SENet have been implemented. And they can be summarized as in Table 7.2.

Model	Skip Connections	Data Augmentation
SENet	Without Skip Connections	Without Data Augmentation
SEResNet	With Skip Connections	Without Data Augmentation
SENet	Without Skip Connections	With Data Augmentation
SEResNet	With Skip Connections	With Data Augmentation

Table 7.2: All SENet Models

Table 7.3 shows the comparison among various methods and the four variations of the proposed methods that yield the highest results. Here, the efficiency of the proposed method is compared with seven models, including Wishart-AE model or Wishart-CAE model[47], Fisher Vectors [38], Self-Paced CNN [26], Three-Channel CNN [24], Multi-Channel fusion CNN based on scattering mechanism [46], and Complex valued CNN [50] for real-valued as well as complex-valued network. One can observe that the proposed method is able to produce best results as compared to the existing approaches.

Method	**Accuracy**
WCAE [47]	93.31
FV [38]	91.44
SPCNN [26]	96.90
Tc-CNN [24]	96.63
MCFCNN+Newloss [46]	95.83
CV-CNN [50]	96.20
RV-CNN [50]	95.30
Proposed method 1 **(SENet without skip connections)**	**97.12**
Proposed method 2 **(SENet with skip connections)**	**96.04**
Proposed method 1 **With Data Augmentation**	**98.17**
Proposed method 2 **With Data Augmentation**	**97.25**

Table 7.3: Comparisons on Flevoland15 Dataset

Using variations of Composite Kernel Method for PolSAR Image Classification Based on Polarimetric-Spatial, four different methods namely POL, MP, Vector sketching, and composite kernel were able to achieve overall accuracy of 86.9%, 89.6%, 92.6%, and 94.4% respectively [45]. Consequently, previous work using both statistical and deep learning-based methods [16] have produced accuracy as shown in Table 7.4. As shown in Table 7.5, SENet with complex valued CNNs have been able to achieve better performance than various deep learning methods discussed in [49] and [39].

Table 7.4: Result Comparison for SFAIRSAR

Method	**Overall Accuracy**
Wishart	81.49
RBF-SVM	81.34
CNN	90.38
CV-CNN	91.83
SF-CNN	92.01
DAS-CNN	93.43
CVDAS-CNN	94.32
Composite Kernel	94.40
SENet with Data Augmentation	**98.30**

Table 7.5: Result Comparison for SFRS2

Method	**Overall Accuracy**
CNN	81.00
AN-CNN	83.90
SW-CNN	74.41
Grid research CNN	89.78
Tensor Embedding	90.76
SENet with Data Augmentation	**98.84**

Chapter 8

Conclusion and Future Work

The implementations that use CNN models confirm the efficiency of CNNs real-valued and complex-valued both. The overall accuracy in these models itself proves that deep learning based classification does not require any preprocessing such as speckle filtering on the raw unlfiltered data. Moreover, the overall accuracy in these five models itself crosses the accuracy of statistical methods that use unfiltered data.

In the classification using squeeze and excitation blocks, spatial and channel-wise important information plays an essential role. The spatial correlation is taken care of by convolutional blocks, whereas the inter-channel dependencies are exploited by squeeze and excitation blocks. To deal with complex values, we have used complex-valued CNN. The SE block has also been modified accordingly. Furthermore, the data augmentation techniques are employed to provide enough examples during the training phase of the model.

The proposed models have produced promising results on unfiltered data confirming that the squeeze-excitation, when configured with complex-valued deep learning methods appropriately, can suppress the effect of speckle noise while classifying the PolSAR image.

The implementations discussed in this work explore the efficiency of squeeze and excitation networks with deep learning operations that are designed to use complex values instead of real values. SENets extract channel wise information efficiently. Work can be done to better implement pixel wise relations along with these channel wise relations for further improving the performance.

Data augmentation techniques have proven remarkable in solving the issue that the datasets available for experimentation are small in size. The data used for data augmentation is purely C-band data. For further improving the model performance, dual band data such as L- and C-band data can be used.

References

[1] J. Ai, M. Huang, F. Wang, X. Yang, and Y. Wu. Completed local binary patterns feature integrated convolutional neural network-based terrain classification algorithm in polarimetric synthetic aperture radar images. *Journal of Applied Remote Sensing*, 16(1):1 – 16, 2022.

[2] S. Albawi, T. A. Mohammed, and S. Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017.

[3] M. Arii, J. J. van Zyl, and Y. Kim. Adaptive model-based decomposition of polarimetric sar covariance matrices. *IEEE Transactions on Geoscience and Remote Sensing*, 49(3):1104–1113, 2011.

[4] H. Bi, J. Sun, and Z. Xu. A graph-based semisupervised deep learning model for polsar image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(4):2116–2132, 2019.

[5] Y. Cao, Y. Wu, P. Zhang, W. Liang, and M. Li. Pixel-wise polsar image classification via a novel complex-valued deep fully convolutional network, 2019.

[6] L. Chang, M.-H. Hung, Y.-L. Chang, and Y.-J. Chen. Improved speckle filtering of polarimetric SAR by target signature preserving. In J. M. Craven, J. A. Shaw, and F. Snik, editors, *Polarization Science and Remote Sensing IX*, volume 11132, pages 205 – 214. International Society for Optics and Photonics, SPIE, 2019.

[7] N. Chaudhari, S. K. Mitra, S. Chirakkal, S. Mandal, D. Putrevu, and A. Misra. Discrimination of multi-crop scenarios with polarimetric SAR data using Wishart mixture model. *Journal of Applied Remote Sensing*, 15(3):1 – 21, 2021.

[8] C.-T. Chen, K.-S. Chen, and J.-S. Lee. The use of fully polarimetric information for the fuzzy neural classification of sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 41(9):2089–2100, 2003.

[9] D. Cozzolino, G. Di Martino, G. Poggi, and L. Verdoliva. A fully convolutional neural network for low-complexity single-stage ship detection in sentinel-1 sar images. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 886–889, 2017.

[10] M. Dabboor, M. J. Collins, V. Karathanassi, and A. Braun. An unsupervised classification approach for polarimetric sar data based on the chernoff distance for complex wishart distribution. *IEEE Transactions on Geoscience and Remote Sensing*, 51(7):4200–4213, 2013.

[11] K. Dasari, L. Anjaneyulu, P. Jayasri, and A. Prasad. Importance of speckle filtering in image classification of sar data. In *2015 International Conference on Microwave, Optical and Communication Engineering (ICMOCE)*, pages 349–352, 2015.

[12] K. Dasari and A. Lokam. Eigen value and eigen vector based decomposition and wishart supervised classification on fully polarimetric sar data. *Indian Journal of Science and Technology*, 9, 03 2016.

[13] J. Ding, B. Chen, H. Liu, and M. Huang. Convolutional neural network with data augmentation for sar target recognition. *IEEE Geoscience and Remote Sensing Letters*, 13(3):364–368, 2016.

[14] H. Dong, L. Zhang, and B. Zou. Polsar image classification with lightweight 3d convolutional networks. *Remote Sensing*, 12(3), 2020.

[15] H. Dong, B. Zou, L. Zhang, and S. Zhang. Automatic design of cnns via differentiable neural architecture search for polsar image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(9):6362–6375, 2020.

[16] H. Dong, B. Zou, L. Zhang, and S. Zhang. Automatic design of CNNs via differentiable neural architecture search for PolSAR image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(9):6362–6375, sep 2020.

[17] ESA. Earth online - esa. https://step.esa.int/main/toolboxes/polsarpro-v6-0-biomass-edition-toolbox/.

[18] A. Freeman and S. Durden. A three-component scattering model for polarimetric sar data. *IEEE Transactions on Geoscience and Remote Sensing*, 36(3):963–973, 1998.

[19] S. Fukuda and H. Hirosawa. Support vector machine classification of land cover: application to polarimetric sar data. In *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No.01CH37217)*, volume 1, pages 187–189 vol.1, 2001.

[20] F. Gao, T. Huang, J. Wang, J. Sun, A. Hussain, and E. Yang. Dual-branch deep convolution neural network for polarimetric sar image classification. *Applied Sciences*, 7(5), 2017.

[21] R. S. Hosseini, I. Entezari, S. Homayouni, M. Motagh, and B. Mansouri. Classification of polarimetric sar images using support vector machines. *Canadian Journal of Remote Sensing*, 37(2):220–233, 2011.

[22] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[23] W. Hua, W. Xie, and X. Jin. Three-channel convolutional neural network for polarimetric sar images classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:4895–4907, 2020.

[24] W. Hua, W. Xie, and X. Jin. Three-channel convolutional neural network for polarimetric sar images classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:4895–4907, 2020.

[25] R. Hänsch and O. Hellwich. Classification of polarimetric sar data by complex valued neural networks. 05 2009.

[26] C. Jiao, X. Wang, S. Gou, W. Chen, D. Li, C. Chen, and X. Li. Self-paced convolutional neural network for polsar images classification. *Remote Sensing*, 11(4), 2019.

[27] F. Lang, J. Yang, D. Li, L. Shi, and J. Wei. Mean-shift-based speckle filtering of polarimetric sar data. *IEEE Transactions on Geoscience and Remote Sensing*, 52(7):4440–4454, 2014.

[28] J.-S. Lee, M. Grunes, and G. de Grandi. Polarimetric sar speckle filtering and its implication for classification. *IEEE Transactions on Geoscience and Remote Sensing*, 37(5):2363–2373, 1999.

[29] J. S. LEE, M. R. GRUNES, and R. KWOK. Classification of multi-look polarimetric sar imagery based on complex wishart distribution. *International Journal of Remote Sensing*, 15(11):2299–2311, 1994.

[30] J.-S. Lee and E. Pottier. *Polarimetric Radar Imaging: From Basics to Applications*. 02 2009.

[31] C. Lopez-Martinez and X. Fabregas. Polarimetric sar speckle noise model. *IEEE Transactions on Geoscience and Remote Sensing*, 41(10):2232–2242, 2003.

[32] X. Mei, W. Nie, J. Liu, and K. Huang. Polsar image crop classification based on deep residual learning network. In *2018 7th International Conference on Agro-geoinformatics (Agro-geoinformatics)*, pages 1–6, 2018.

[33] A. Mikołajczyk and M. Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pages 117–122, 2018.

[34] L. Novak and M. Burl. Optimal speckle reduction in polarimetric sar imagery. *IEEE Transactions on Aerospace and Electronic Systems*, 26(2):293–305, 1990.

[35] M. Ouarzeddine, B. Souissi, and A. Belhadj-Aissa. Target detection and characterization using h/alpha decomposition and polarimetric signatures. In *2006 2nd International Conference on Information Communication Technologies*, volume 1, pages 395–400, 2006.

[36] A. Perlato. Steps in convolutional neural networks. ://www.andreaperlato.com/aipost/cnn-and-softmax/.

[37] R. Qin, X. Fu, and P. Lang. Polsar image classification based on low-frequency and contour subbands-driven polarimetric senet. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:4760–4773, 2020.

[38] J. Redolfi, J. Sánchez, and A. G. Flesia. Fisher vectors for polsar image classification. *IEEE Geoscience and Remote Sensing Letters*, 14(11):2057–2061, 2017.

[39] B. Ren, B. Hou, J. Chanussot, and L. Jiao. Polsar feature extraction via tensor embedding framework for land cover classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(4):2337–2351, 2020.

[40] A. Richardson, D. Goodenough, H. Chen, B. Moa, G. Hobart, and W. Myrvold. Unsupervised nonparametric classification of polarimetric sar data using the k-nearest neighbor graph. pages 1867 – 1870, 08 2010.

[41] M. Sharma, M. Dhanaraj, S. Karnam, D. Chachlakis, R. Ptucha, P. Markopoulos, and E. Saber. Yolors: Object detection in multimodal remote sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, PP:1–1, 11 2020.

[42] E. Spectrum. Electromagnetic spectrum. https://www.siyavula.com/read/science/grade-10/electromagnetic-radiation/11-electromagnetic-radiation-03.

[43] S. A. Wagner. Sar atr by a combination of convolutional neural network and support vector machines. *IEEE Transactions on Aerospace and Electronic Systems*, 52(6):2861–2872, 2016.

[44] H. Wang, F. Xu, and Y.-Q. Jin. A review of polsar image classification: from polarimetry to deep learning. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 3189–3192, 2019.

[45] X. Wang, Z. Cao, Y. Ding, and J. Feng. Composite kernel method for polsar image classification based on polarimetric-spatial information. *Applied Sciences*, 7(6), 2017.

[46] Y. Wang, J. Cheng, Y. Zhou, F. Zhang, and Q. Yin. A multichannel fusion convolutional neural network based on scattering mechanism for polsar image classification. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

[47] W. Xie, L. Jiao, B. Hou, W. Ma, J. Zhao, S. Zhang, and F. Liu. Polsar image classification via wishart-ae model or wishart-cae model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(8):3604–3615, 2017.

[48] Y. Xue. An active deep learning approach for minimally supervised polsar image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[49] A. Zhang, X. Yang, L. Jia, J. Ai, and Z. Dong. Sar image classification using adaptive neighborhood-based convolutional neural network. *European Journal of Remote Sensing*, 52(1):178–193, 2019.

[50] Z. Zhang, H. Wang, F. Xu, and Y.-Q. Jin. Complex-valued convolutional neural network and its application in polarimetric sar image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7177–7188, 2017.

[51] Y. Zhou, H. Wang, F. Xu, and Y.-Q. Jin. Polarimetric sar image classification using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 13(12):1935–1939, 2016.

[52] Y. Zhou, H. Wang, F. Xu, and Y.-Q. Jin. Polarimetric sar image classification using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 13(12):1935–1939, 2016.

[53] X. Zhu, S. Montazeri, M. Ali, Y. Hua, Y. Wang, L. Mou, Y. Shi, F. Xu, and R. Bamler. Deep learning meets sar: Concepts, models, pitfalls, and perspectives. *IEEE Geoscience and Remote Sensing Magazine*, pages 0–0, 2021.