

# Polarimetric SAR Image Classification using Gaussian Context Transformer in Complex-Valued Convolutional Neural Networks

by

PANDYA UTKARSH SAMIRBHAI  
202111026

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**



July, 2023

## 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.

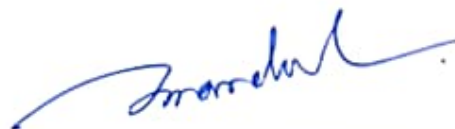


---

PANDYA UTKARSH SAMIRBHAI

## Certificate

This is to certify that the thesis work entitled **Polarimetric SAR Image Classification using Gaussian Context Transformer in Complex-Valued Convolutional Neural Networks** has been carried out by **PANDYA UTKARSH SAMIRBHAI** for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.



---

PROF. SRIMANTA MANDAL  
Thesis Supervisor

# Acknowledgments

I would like to thank my supervisor Prof. Srimanta Mandal for providing me with this opportunity to conduct this research and providing continuous efforts in shaping this Thesis. It is his insightful guidance that has helped in making this Thesis what it is. I also thank Prof. Tapas Kumar Meiti for his valuable suggestions.

I would also like to thank DAIICT and SAC-ISRO for giving me the opportunity to conduct this research. I would also like to thank Dr. Sanid Chirakkal and Deepak Putrevu for their guidance during this Thesis.

I would also like to thank my predecessors Ms. Shradha Makhija and Ms. Nilamben Choudhari for their contributions as JRFs in this project as well as for helping me get familiar with the basics of Remote Sensing Data and how to handle it. They have also been helpful in answering any queries throughout the Thesis.

I would also like to thank Prof. Rahul Mishra, Prof. PM Jat, Prof. Sujay Kadam, and Prof. Naleen Sharma for providing valuable inputs during the stage presentations. I would also like to thank all the members of the scientific community, and all the predecessors and colleagues who have worked or are working on developing Remote Sensing, PolSAR technology, and Machine Learning or Deep Learning techniques. It is on their shoulders that this Thesis stands.

I would also like to thank all the faculties for guiding us through the coursework and enabling me with the knowledge and tools to perform better in my studies and research. I would also like to thank my peers for supporting me and creating a conducive atmosphere for learning.

At last, I would like to thank my parents for always supporting me in my endeavors.

# Contents

<b>Abstract</b>	<b>v</b>
<b>List of Tables</b>	<b>vi</b>
<b>List of Figures</b>	<b>viii</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Objectives . . . . .	4
1.2 Contribution of Thesis . . . . .	4
1.3 Organization of Thesis . . . . .	5
<b>2 Literature review</b>	<b>6</b>
2.1 Classical Methods . . . . .	6
2.2 Deep-Learning Methods . . . . .	8
<b>3 Background</b>	<b>11</b>
3.1 Remote Sensing . . . . .	11
3.2 Synthetic Aperture Radar . . . . .	12
3.3 Polarimetric SAR . . . . .	14
3.4 Mathematical concepts for PolSAR data . . . . .	16
3.5 CNNs in PolSAR Image Classification . . . . .	17
<b>4 Dataset</b>	<b>19</b>
<b>5 Methodology</b>	<b>23</b>
5.1 Complex-Valued Convolutional Neural Network (CV-CNN) . . . . .	26
5.2 Complex-Valued Squeeze and Excitation Network . . . . .	27
5.3 Complex-Valued Squeeze and Excitation Residual Network . . . . .	29
5.4 Complex-Valued Gaussian Context Transformer . . . . .	30
5.5 Loss Function and Optimization . . . . .	32
5.6 Data Augmentation . . . . .	39

<b>6 Summary and Analysis</b>	<b>51</b>
6.1 Cumulative Analysis . . . . .	51
6.2 Ablation Study . . . . .	57
6.3 Comparison with State of the art Models . . . . .	75
<b>7 Conclusion</b>	<b>77</b>
<b>8 Future Work</b>	<b>78</b>
<b>References</b>	<b>79</b>

# Abstract

There have been many advancements in the field of terrain classification using Polarimetric SAR images/data. This Thesis explores different classical methods as well as deep learning methods for this task. The Covariance matrix of land sample data is classified into different terrains such as various crops, urban areas or water, etc. The PolSAR covariance matrix has both amplitude and phase components. Statistical techniques such as Wishart Classifier and Wishart Mixture Model with Conditional Random Field (WMM-CRF) approach exploit the inherent mathematical predispositions of the data while Deep Learning techniques such as Complex Valued-CNN and Squeeze and Excitation Networks utilize the brilliance of neural networks to study correlation in spatial data as well as inter-channel dependencies. There have been studies in order to retrofit deep learning models with components that can leverage the predetermined data patterns in any dataset. Gaussian Context Transformer is one such technique that allows the exploitation of inter-channel dependencies with predetermined mathematical inclinations while the rest of the model learns spatial-contextual parameters. In order to overcome noise, there are no available ground truth images, hence data augmentation is done with several image processing techniques such as Box-Car filter, Lee-Sigma filter, and Mean-Shift filters can be used to downsize the effects of the multiplicative noise as much as possible. The effects of Gaussian Context Transformers and Data augmentation on one Indian land sample, namely, Mysore and three European land samples, namely, Flevoland-7, Flevoland-15, and Landes show promising results.

# List of Tables

4.2	Details of Datasets used for experimentation . . . . .	19
5.1	Comparison Results on Flevoland 15 across GCT model with the best accuracy with and without skip connections . . . . .	34
5.2	Comparison Results on Flevoland 7 across GCT model with the best accuracy with and without skip connections . . . . .	36
5.3	Comparison Results on Landes across GCT model with the best accuracy with and without skip connections . . . . .	38
5.4	Comparison Results on Flevoland 15 with Data Augmentation across GCT model with the best accuracy with and without skip connections	41
5.5	Comparison Results on Flevoland 7 with Data Augmentation across GCT model with the best accuracy with and without skip connections	43
5.6	Comparison Results on Landes with Data Augmentation across GCT model with the best accuracy with and without skip connections .	46
5.7	Comparison Results on Mysore with Data Augmentation across GCT model with the best accuracy with and without skip connections	48
6.1	Comparison Results on Flevoland 15 across all models without GCT	52
6.2	Comparison Results on Flevoland 15 across all models with GCT .	52
6.3	Comparison Results on Flevoland 7 across all models without GCT	53
6.4	Comparison Results on Flevoland 7 across all models with GCT . .	54
6.5	Comparison Results on Landes dataset across all models without GCT . . . . .	55
6.6	Comparison Results on Landes dataset across all models with GCT	55
6.7	Comparison Results on Mysore dataset across all models . . . . .	57
6.8	Comparison Results on Flevoland 15 across all SENet + GCT variations . . . . .	58
6.9	Comparison Results on Flevoland 15 across all SEResNet + GCT variations . . . . .	59

6.10 Comparison Results on Flevoland 15 dataset with Data Augmentation across all SENet + GCT variations . . . . .	60
6.11 Comparison Results on Flevoland 15 dataset with Data Augmentation across all SEResNet + GCT variations . . . . .	62
6.12 Comparison Results on Flevoland 7 across all SENet + GCT variations	63
6.13 Comparison Results on Flevoland 7 across all SEResNet + GCT variations . . . . .	64
6.14 Comparison Results on Flevoland 7 dataset with Data Augmentation across all SENet + GCT variations . . . . .	65
6.15 Comparison Results on Flevoland 7 dataset with Data Augmentation across all SEResNet + GCT variations . . . . .	66
6.16 Comparison Results on Landes across all SENet + GCT variations .	68
6.17 Comparison Results on Landes across all SEResNet + GCT variations	69
6.18 Comparison Results on Landes dataset with Data Augmentation across all SENet + GCT variations . . . . .	70
6.19 Comparison Results on Landes dataset with Data Augmentation across all SEResNet + GCT variations . . . . .	71
6.20 Comparison Results on Mysore dataset with Data Augmentation across all SENet + GCT variations . . . . .	72
6.21 Comparison Results on Mysore dataset with Data Augmentation across all SEResNet + GCT variations . . . . .	74
6.22 Comparison Results on Flevoland 15 dataset across the different state of the art models . . . . .	75



# List of Figures

3.2	Different bandwidths for remote sensing[1] . . . . .	12
3.3	SAR imaging geometry in strip-map mode[2] . . . . .	13
3.4	Different polarization states of scattered waves[3] . . . . .	15
4.1	(a) Pauli RGB of Flevoland 7 dataset (b) Ground Truth of Flevoland 7 dataset . . . . .	20
4.2	(a) Pauli RGB of Flevoland 15 dataset (b) Ground Truth of Flevoland 15 dataset . . . . .	21
4.3	(a) Pauli RGB of Landes dataset (b) Ground Truth of Landes dataset	21
4.4	Mysore data set showing (a) Pseudo-color Freeman-Durden RGB image in slant-range resolution (b) Pseudo-color Freeman-Durden RGB image ground-range resolution and (c) Sentinel-2 optical image	22
5.1	Overview of the proposed Model . . . . .	24
5.2	Parameters of the proposed Model for Flevoland 7 dataset . . . . .	25
5.3	CV-CNN Architecture[4] . . . . .	26
5.4	Squeeze and Excitation Network Block[5] . . . . .	28
5.5	Squeeze and Excitation Network Residual Block[5] . . . . .	29
5.6	Diagram of Gaussian Context Transformer[6] . . . . .	30
5.7	Diagram of Gaussian Context Transformer . . . . .	32
5.8	Resulting image of Flevoland 15 dataset using GCT model (c=2) with the best performance without skip connections . . . . .	34
5.9	Resulting image of Flevoland 15 dataset using GCT model (c=2) with the best performance with skip connections . . . . .	35
5.10	Resulting image of Flevoland 7 dataset using GCT model (c=2) with the best performance without skip connections . . . . .	36
5.11	Resulting image of Flevoland 7 dataset using GCT model (c=2) with the best performance with skip connections . . . . .	37
5.12	Resulting image of Landes dataset using GCT model (c=1) with the best performance without skip connections . . . . .	38

5.13	Resulting image of Landes dataset using GCT model (c=3) with the best performance with skip connections . . . . .	39
5.14	Resulting image of Flevoland 15 dataset with Data Augmentation using GCT model (c=1) with the best performance without skip connections . . . . .	42
5.15	Resulting image of Flevoland 15 dataset with Data Augmentation using GCT model (c=3) with the best performance with skip connections . . . . .	42
5.16	Resulting image of Flevoland 7 dataset with Data Augmentation using GCT model (c=4) with the best performance without skip connections . . . . .	44
5.17	Resulting image of Flevoland 7 dataset with Data Augmentation using GCT model (c=3) with the best performance with skip connections . . . . .	45
5.18	Resulting image of Landes dataset with Data Augmentation using GCT model (c=3) with the best performance without skip connections	46
5.19	Resulting image of Landes dataset with Data Augmentation using GCT model (c=2) with the best performance with skip connections	47
5.20	Resulting image of Mysore dataset with Data Augmentation using GCT model (c=2) with the best performance without skip connections	49
5.21	Resulting image of Mysore dataset with Data Augmentation using GCT model (c=3) with the best performance with skip connections	50
6.1	Resulting images of all models for Flevoland 15 dataset: (a) SENet (b) SEResNet (c) SENet + GCT (c=2) (d) SEResNet + GCT (c=2) (e) SENet + Data Augmentation (f) SEResNet + Data Augmentation (g) SENet + GCT (c=2) + Data Augmentation (h) SEResNet + GCT (c=2) + Data Augmentation . . . . .	53
6.2	Resulting images of all models for Flevoland 7 dataset: (a) SENet (b) SEResNet (c) SENet + GCT (c=2) (d) SEResNet + GCT (c=2) (e) SENet + Data Augmentation (f) SEResNet + Data Augmentation (g) SENet + GCT (c=4) + Data Augmentation (h) SEResNet + GCT (c=3) + Data Augmentation . . . . .	54
6.3	Resulting images of all models for Landes dataset: (a) SENet (b) SEResNet (c) SENet + GCT (c=1) (d) SEResNet + GCT (c=3) (e) SENet + Data Augmentation (f) SEResNet + Data Augmentation (g) SENet + GCT (c=3) + Data Augmentation (h) SEResNet + GCT (c=2) + Data Augmentation . . . . .	56

6.4	Resulting images of all models for Mysore dataset: (a) SENet + Data Augmentation (b) SEResNet + Data Augmentation (c) SENet + GCT (c=2) + Data Augmentation (d) SEResNet + GCT (c=2) + Data Augmentation . . . . .	57
6.5	Image results on Flevoland 15 dataset of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet, (b) SENet + GCT (c=1), (c) SENet + GCT (c=2), (d) SENet + GCT (c=3), (e) SENet + GCT (c=4) . . . . .	59
6.6	Image results on Flevoland 15 dataset of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet, (b) SEResNet + GCT (c=1), (c) SEResNet + GCT (c=2), (d) SEResNet + GCT (c=3), (e) SEResNet + GCT (c=4) . . . . .	60
6.7	Image results on Flevoland 15 dataset with Data Augmentation of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet + Data Augmentation, (b) SENet + GCT (c=1) + Data Augmentation, (c) SENet + GCT (c=2) + Data Augmentation, (d) SENet + GCT (c=3) + Data Augmentation, (e) SENet + GCT (c=4) + Data Augmentation . . . . .	61
6.8	Image results on Flevoland 15 dataset with Data Augmentation of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet + Data Augmentation, (b) SEResNet + GCT (c=1) + Data Augmentation, (c) SEResNet + GCT (c=2) + Data Augmentation, (d) SEResNet + GCT (c=3) + Data Augmentation, (e) SEResNet + GCT (c=4) + Data Augmentation . . . . .	62
6.9	Image results on Flevoland 7 dataset of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet, (b) SENet + GCT (c=1), (c) SENet + GCT (c=2), (d) SENet + GCT (c=3), (e) SENet + GCT (c=4) . . . . .	64
6.10	Image results on Flevoland 7 dataset of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet, (b) SEResNet + GCT (c=1), (c) SEResNet + GCT (c=2), (d) SEResNet + GCT (c=3), (e) SEResNet + GCT (c=4) . . . . .	65

6.11	Image results on Flevoland 7 dataset with Data Augmentation of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet + Data Augmentation, (b) SENet + GCT (c=1) + Data Augmentation, (c) SENet + GCT (c=2) + Data Augmentation, (d) SENet + GCT (c=3) + Data Augmentation, (e) SENet + GCT (c=4) + Data Augmentation . . . . .	66
6.12	Image results on Flevoland 7 dataset with Data Augmentation of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet + Data Augmentation, (b) SEResNet + GCT (c=1) + Data Augmentation, (c) SEResNet + GCT (c=2) + Data Augmentation, (d) SEResNet + GCT (c=3) + Data Augmentation, (e) SEResNet + GCT (c=4) + Data Augmentation . . . . .	67
6.13	Image results on Landes dataset of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet, (b) SENet + GCT (c=1), (c) SENet + GCT (c=2), (d) SENet + GCT (c=3), (e) SENet + GCT (c=4) . . . . .	68
6.14	Image results on Landes dataset of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet, (b) SEResNet + GCT (c=1), (c) SEResNet + GCT (c=2), (d) SEResNet + GCT (c=3), (e) SEResNet + GCT (c=4) . . . . .	69
6.15	Image results on Landes dataset with Data Augmentation of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet + Data Augmentation, (b) SENet + GCT (c=1) + Data Augmentation, (c) SENet + GCT (c=2) + Data Augmentation, (d) SENet + GCT (c=3) + Data Augmentation, (e) SENet + GCT (c=4) + Data Augmentation . . . . .	70
6.16	Image results on Landes dataset with Data Augmentation of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet + Data Augmentation, (b) SEResNet + GCT (c=1) + Data Augmentation, (c) SEResNet + GCT (c=2) + Data Augmentation, (d) SEResNet + GCT (c=3) + Data Augmentation, (e) SEResNet + GCT (c=4) + Data Augmentation . . . . .	71

6.17	Image results on Mysore dataset with Data Augmentation of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet + Data Augmentation, (b) SENet + GCT (c=1) + Data Augmentation, (c) SENet + GCT (c=2) + Data Augmentation, (d) SENet + GCT (c=3) + Data Augmentation, (e) SENet + GCT (c=4) + Data Augmentation . . . . .	73
6.18	Image results on Mysore dataset with Data Augmentation of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet + Data Augmentation, (b) SEResNet + GCT (c=1) + Data Augmentation, (c) SEResNet + GCT (c=2) + Data Augmentation, (d) SEResNet + GCT (c=3) + Data Augmentation, (e) SEResNet + GCT (c=4) + Data Augmentation . . . . .	75

## CHAPTER 1

# Introduction

Remote Sensing has been exceedingly proving its relevance in today's time with the burgeoning application that emerges from it in the fields of agriculture, military, rescue, weather, etc. Data can be accumulated with various types of data-capturing techniques ranging from optical data to microwaves. Synthetic Aperture Radar is one such technique through which remote sensing information can be captured. The advantage of SAR is that the day-night capability and all-weather capability of its sensors make it more reliable for acquiring information in the face of adversities or visual obstacles. SAR provides large-scale two-dimensional images with high spatial resolution. It obtains images from Earth's surface reflectivity of its microwave pulses emitted from its active sensors.

These satellites are of two types, namely, space-borne and air-borne. The data is dependent on the type of signal used by the radar. It is classified into polarization channels. These polarization channels capture various features on the same object. Normally, SAR systems function on only one band from C, L, P, and X but some new projects work on utilizing two bands of frequencies together to capture more information.

The SAR images are mere signal reflectivities returning back from Earth's surface with different polarization at both transmitting and receiving ends making it quad-pol applications with channels being HH, HV, VV, and VH. The first letter indicates whether the transmitted pulse is Horizontally polarized or vertical and the same for the latter for receiving end. A major challenge is also encountered at the time of accumulating SAR reflectivities and resulting in a multiplicative noise that looks like salt noise but it is not. These speckles are very hard to remove since there is no ground truth available for a reference of clean SAR images. Prominent Filters used for the same are Lee filter, Box Filter, etc.

The covariance matrix of polarimetric SAR images is observed to follow a complex Wishart distribution for a homogeneous region. While for heterogeneous terrains, the covariance matrix follows a mixture of multiple Wishart distributions. Chaudhari et al.[7][8] explores the classical methods application of Wishart Classifier on the homogeneous terrain classification and Wishart Mixture Model (WMM) Classifier on the heterogeneous terrain classification tasks. Moreover, Conditional Random Fields are also added to the WMM Classifier to obtain better performance by inculcating spatial information all the while preserving edges and micro-edges.

With the advent of deep learning techniques, several attempts have been made for using Deep Learning techniques for the SAR terrain classification tasks which are entailed in the study. Kussul et al.[9] shows the approach of the Multilayer Perceptron network trained on the SAR data outperforms the classical methods like SVM and Decision Tree. Yet, the phase information remains to be leveraged. In order to overcome this challenge, the work of Zhang et al [4] shows that a Complex Valued CNN can be developed for the purpose of utilizing the phase information as well as the amplitude information of the covariance matrix. This improves the performance of the model providing a 6.6% overall error rate while that of its corresponding RV-CNN was 10.1%.

The local receptive fields' spatial relationships are primarily the subject of CNN's attention. However, the procedure combines spatial information with channel correlation. In [3], Deep Residual Networks (ResNet) have also been used for the classification task. The Squeeze and Excitation Network (SENet) is also investigated for utilizing the covariance matrix's inter-channel dependencies. The work of Ruan et al.[6] shows that the relationship between channels is predetermined in some contexts, A channel attention block called Gaussian Context transformer is implemented in order to further improve the channel interdependencies with respect to the Gaussian distribution. The same has been implemented in the SEResNet model in order to enhance distribution learning. The preliminary results for the same show some improvement in the overall accuracy. Moreover, data augmentation is also done using image processing techniques such as Box Car filter, Lee Sigma Filter, and Mean Shift filters using the software PolSARPro v 6.0 Biomass Edition. This further improves the accuracy of the models.

## 1.1 Objectives

Mathematical or Statistical models can leverage the predisposed mathematical tendencies in a dataset while Deep Learning techniques learn the parameters with the help of deep/hidden layers, CNNs, and interchannel dependencies. The main objectives of the Thesis are as follows:

- Combine the advantages of both mathematical models as well as deep learning techniques by introducing the Gaussian Context Transformer in the deep learning models.
- Exploring the effects of Residual blocks in GCT models.
- Deploying various data augmentation techniques in order to get the advantage in classification rates for small datasets.
- Retrofit the models to work in Indian land samples which have small farm sizes.

## 1.2 Contribution of Thesis

The following are the salient contributions of this Thesis:

- Explore Gaussian Context Transformer techniques in order to leverage the mathematical predispositions in the dataset within the deep learning models.
- Exploring the effect of Residual blocks in the GCT models.
- Speckle filtering techniques are devised in order to reduce speckle noise in the dataset.
- Dataset Augmentation is done for increasing the size of datasets for better training the models.
- Retrofitting models for the Indian scenario of small crop fields.



## 1.3 Organization of Thesis

Following is the organization of the Thesis components:

- Chapter 2 discusses various techniques discovered during the Literature Survey.
- Chapter 3 gives a general overview/background on various technologies such as SAR, and its covariance matrix.
- Chapter 4 contains information on the Datasets used.
- Chapter 5 discusses the Methodologies implemented in the Thesis and their respective results.
- Chapter 6 entails a comparative analysis of results obtained across different methodologies used. It also contains a section for ablation study and a section for comparison with state-of-the-art models.
- Chapter 7 concludes the Thesis
- Chapter 8 points toward possible Future Work.

## CHAPTER 2

# Literature review

Remote sensing involves emitting signals from airborne or spaceborne radars toward the Earth's surface. These signals are in the range of microwaves and are classified into frequency bands of C, L, P, X, K, and Ku, etc. Many Earth Observation Satellite missions have sent these Synthetic Aperture Radars into space such as RADARSAT-2, EN-VISAT, ALOS PALSAR, and TerraSAR-X. These radars get the reflectivities back from Earth's surface and map the topography and telemetry on the basis of the signals obtained. The phase component and amplitude component gives the idea of polarization of signal and thereby, the surface of Earth is mapped along with the dielectric properties of the Surface as well. Since these radars use microwaves, they can be used to get information during bad weather and both day and night. A coherency matrix or a covariance matrix is constructed at last which is a  $3 \times 3$  Complex valued matrix. Each pixel has its covariance ( $C_3$ )/coherency matrix ( $T_3$ ) and based on this, they are classified into various classes such as urban, various crops, water, forest, etc. Various techniques have been developed over time for the same but it can be broadly classified into Statistical methods and Machine Learning or Deep Learning methods.

## 2.1 Classical Methods

Anfinsen et al. [10] explore the use of Support Vector Machines for classification tasks on Polarimetric SAR data. It also analyzes the use of different Kernel functions for the same. The three-component scattering model for polarimetric SAR data is presented in Freeman et al. [11]. The model depicts how electromagnetic waves interact with several types of scatterers present in the environment, including surface, double bounce, and volume scattering. It offers the groundwork for statistical decomposition methods, making it possible to classify polarimetric SAR data by extracting useful information from it.

Lee et al. [2] offers a thorough analysis of the polarimetric radar imaging method. with the use of the Wishart distribution for polarimetric SAR categorization. Polarimetric SAR data's multivariate statistics are modeled by the Wishart distribution, whose parameters are calculated from the observed data. It explores several classification techniques based on the Wishart distribution, including supervised classification algorithms and maximum likelihood estimates. Conradsen et al. [12] propose statistical clustering-based algorithms. It serves as a foundation for clustering-based classification techniques, which group comparable scatterers and categorize various types of land cover using polarimetric SAR data and clustering algorithms. While Van et al. [13] utilizes the Bayesian classification using apriori information which can be helpful while limited data is available.

In Chaudhary et al. [7], the authors state that the homogeneous terrain's covariance matrix has a unimodal complex Wishart distribution. However, for a heterogeneous terrain, there are irregularities, such as some parts of it containing high-density built-ups and other parts containing low-density built-ups. In agricultural land, there can be different heights of the crops or uneven structures, which causes heterogeneity. A mixture model is implemented in this paper, in order to express given data distribution as a weighted summation of multiple component distributions. This is called Wishart Mixture Model (WMM). The parameters of WMM are estimated using the Expectation-Maximization algorithm.

The results show that the WMM classifier performs better in the heterogeneous terrain with an overall accuracy of 99.20% in comparison to the Wishart classifier with that of 90.10% on the Flevoland dataset. Furthermore, it is shown that in EM algorithm initialization gives slightly better results than random initialization.

In Chaudhary et al. [8], the author tries to improve classification results by inculcating spatial-contextual information along with the preservation of finer details such as edges and micro-regions. A Conditional Random Field (CRF) based model is implemented for this purpose along with Wishart and Wishart Mixture Model (WMM) classifiers. In order to implement this, the author classifies polarimetric SAR images using pixel-based statistical classifiers, namely Wishart and WMM, first, and then, computes unary and pairwise potentials. The smaller the Chebyshev distance, the closer or more similar they are considered. Further, the energy functions are computed by adding both potentials and classification la-

bels are updated accordingly. This process is iterated till a threshold is reached. The overall accuracy of Wishart-CRF on the Flevoland dataset is calculated to be 92.19% while that of Wishart-MRF turns out to be 90.29% and for just Wishart it is 84.56%. This shows promising results for the Wishart-CRF model.

## 2.2 Deep-Learning Methods

There have been several approaches to classify PolSAR images through Deep Learning techniques. Zhu et al. [14] state various methods that have so far been explored in SAR image classification. Autoencoder architecture learns the latent representation of the input data points through non-linear mapping with the target. While Deep Belief Network is a stochastic undirected graphical model consisting of a visible layer and a hidden layer and a symmetric connection between two layers. Also, CNNs have achieved massive popularity with their spatial context utilization for image classification, object detection, etc. Alex Net first improved LeNet in a drastic way. It first utilized ReLU in order to reduce training time, introduced dropout layers to avoid overfitting and it was used in GPUs which allowed larger datasets to be trained.

The author further goes on to cite the uses of VGG Net which use filters with small receptive field of 3x3 rather than 5x5 or 7x7. They have the same feature map size and number of filters in convolution layers of the same block and the increased size of deeper layers. Then came ResNet, which, with the use of skip-connections, avoided the problem of overfitting by performing identity mapping.

While classical methods are based mostly on pixel-wise polarimetric target decomposition parameters, the deep learning methods using CNN consider the spatial patterns which convey rich information in high-resolution SAR images.

In Kussul et al.[9], a comparative analysis is done between neural networks, support vector machines, and decision trees for SAR crop classification. The feed-forward neural network or Multi-Layer Perceptron (MLP) was computed with logistic outputs and cross-entropy error function minimized using a quasi-Newton algorithm. The classification was done on a per-pixel basis. The overall accuracy of the MLP classifier was 80.4% compared to that of SVM at 78.6% and Decision Tree at 78.1%. Moreover, adding SAR images to optical images yields better re-

sults for certain crops that register year-round cultivation. In Cozzolino et al. [15], authors explore the use of CNN, in order to detect ships as a use-case of maritime surveillance in the Sentinel-1 dataset. While Wagner et al. [16] provide insight into combining both Support Vector Machine and Convolutional Neural Networks in order to leverage both machine learning and deep learning techniques to extract features of higher dimensions. Chen et al. [17] explore the classification of the target using CNNs and how sparsely connected layers perform better than fully connected layers in ConvNets on the M-STAR benchmark dataset for 10 class classifications. At the same time, Cheng et al. [18] introduces different loss function in order to discriminate while overcoming within-class diversities and between-class similarities.

In Zhang et al.[4], the author suggests that since the SAR images have both amplitude and phase components, both can be taken into account for CNN application rather than just the Real-Valued Components. Thus, the author builds an entire Complex Valued CNN model, including Complex Input-Output layers, Complex convolution layers, Complex activation function, Complex Pooling Layer, and a complex backpropagation algorithm based on stochastic gradient descent.

The covariance matrix is a 3x3 symmetric matrix, hence rendering 6 unique elements, however, the diagonal elements are the sole real elements. Thus, in RV-CNN, 9 channels were fed in order to make the complex numbers as individual real-valued channels. While, in CV-CNN, only 6 channels were used since it can compute complex numbers as a whole, there need not be separate channels for 3 complex elements in the covariance matrix. 12x12 window was taken for feeding patches into the CV-CNN and RV-CNN. The Overall Accuracy of CV-CNN was 96.2% and that of RV-CNN was 95.3%. Thus, phase information is a crucial characteristic of SAR images.

Furthermore, in order to leverage channel inter-dependencies, Squeeze and Excitation network has been implemented in [3]. Thus, spatial as well as channel relationships are utilized together. This, in turn, helps in reducing speckle noise in the images. The additional computational cost does not occur due to content aware mechanism that weighs each channel on a global level adaptively.

One SE-Block consists of three operations which are, namely, Squeeze, Excita-

tion, and Scaling. In the squeeze operation, global pooling is done for each channel. During excitation, the module attenuates different weights for each channel using two fully connected layers. Then ReLU and Sigmoid are there at the end of each Excitation block. In the end, the output of Sigmoid is then scaled by multiplying it with the input feature map of the convolution layer that was added before the SEBlock. Furthermore, skip connections are also introduced in order to tackle the problem of vanishing gradient. The overall accuracy of SENet with skip connections is 98.18% and that without skip connections is 98.78%.

Various works have taken place in data augmentation of PolSAR data as some samples can be scarce in quantity and in order to achieve better classification accuracies, more data is required. Zhang et al. [19] propose Generative Adversarial Networks (GANs) and linear synthesis to generate images from the dataset for any given azimuth in order to increase the quantity and quality of the dataset. While Ding et al [20] suggest the use of different image processing techniques to remove the noise and find different poses for the target classes in order to augment the datasets.

The recent state-of-the-art models include Wishart Deep Belief Network (W-DBN)[21], Self-Paced Convolutional Neural Network (SPCNN) [22], Complex-Valued PDAS (CVPDAS-CNN) [23], and Complex-Valued Multi-Scale CNN (CVMS-CNN) [24].

In Liu et al. [21], Deep Belief Network is created to make extensive use of unlabeled POL-SAR pixels in the modeling of POL-SAR pixels. Furthermore, the coherency matrix is employed directly to represent a POL-SAR pixel. In this research, local spatial information is combined with the confusion matrix to clean the preliminary classification result achieved by the W-DBN approach. In Jiao et al. [22], a self-paced learning-based CNN is developed which learns the easier samples first and then moves on to other samples leading it to converge to better values. While in Dong et al. [23], the authors develop a Complex Valued Pol-SAR tailored Differentiable Architecture Search method which identifies the best possible architecture for the classification model instead of a hand-crafted one. In Zhang et al. [24], the authors have developed a Complex Valued Multi-Scale CNN in order to leverage the local and global features of a multiclass image.

## CHAPTER 3

# Background

The history of SAR and the mathematical foundations of polarimetric SAR are discussed in this chapter. Additionally, it sheds some light on Convolution Neural Networks and the ways that deep learning tackles some of the key roadblocks in PolSAR image classification.

### 3.1 Remote Sensing

Remote sensing is a sophisticated technique that allows for the capture of information about the Earth's surface without direct physical contact. It involves gathering information on the electromagnetic radiation reflected or emitted by objects on the Earth's surface using sensors mounted on aircraft or satellites. Then, from this data, important information on the characteristics of the Earth, such as its land cover, vegetation, water bodies, and atmospheric conditions, is analyzed and processed.

Satellite remote sensing technologies acquire images and data across the electromagnetic spectrum, including visible, infrared, and microwave wavelengths. Numerous industries, including agriculture, forestry, environmental monitoring, urban planning, disaster management, and studies of climate change, use remote sensing. Its capabilities have been further increased by the integration of remote sensing with geographic information systems (GIS) and other tools for spatial analysis. Through this integration, remote sensing data can be combined with other geospatial data, such as topographic maps and land-use information, to provide a thorough picture of the Earth's surface and promote defensible decision-making. Monitoring, comprehending, and managing the Earth's resources and ecosystem are all made possible via remote sensing. It offers a practical and affordable method for large-scale data collection.

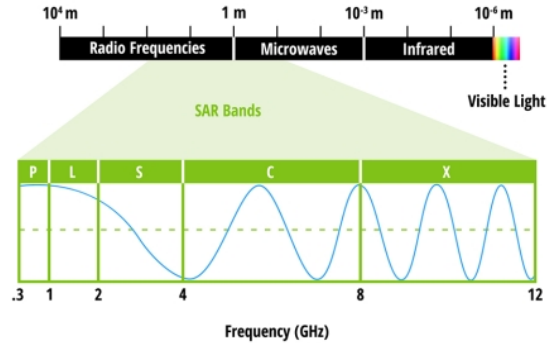


Figure 3.1: Different bandwidths for remote sensing[1]

Microwave imaging is widely used for this purpose since its wavelength falls into a bracket of 1mm-1m which can enable it to bypass atmospheric obstructions such as clouds, storms, fog, etc. It can also penetrate into the ground to some extent depending on its wavelength and give accurate information about the topography and ground cover from the way it bounces off from the surface and the dielectric properties of the surface. Microwave radars also fall into two categories of active radars and passive radars. Passive radars need natural signals and observations are carried out based on that as optical radars and active radars need a source of signal for transmission which is retrieved back and then observations are carried out.

## 3.2 Synthetic Aperture Radar

Synthetic Aperture Radar (SAR) is an active remote sensing technique that makes use of radar emissions to gather detailed information about the surface of the Earth. SAR systems generate microwave pulses and measure the signals that are reflected back from the objects on the ground as opposed to optical remote sensing, which relies on sunlight to illuminate the environment. SAR is a useful instrument for earth observation because of this property, which enables it to function in any weather, day or night.

By mixing the echoes picked up across a sequence of consecutive radar pulses, a moving radar antenna can create a huge aperture or "virtual antenna" in the con-



text of SAR. Real aperture antennas' physical restrictions are addressed by this method, allowing SAR to attain high-resolution imaging capabilities. SAR constantly sends pulses and records the backscattered signals while the radar platform travels along its flight path or the satellite orbits the Earth. SAR systems produce a detailed image of the Earth's surface by examining the phase and amplitude data of the received signals.

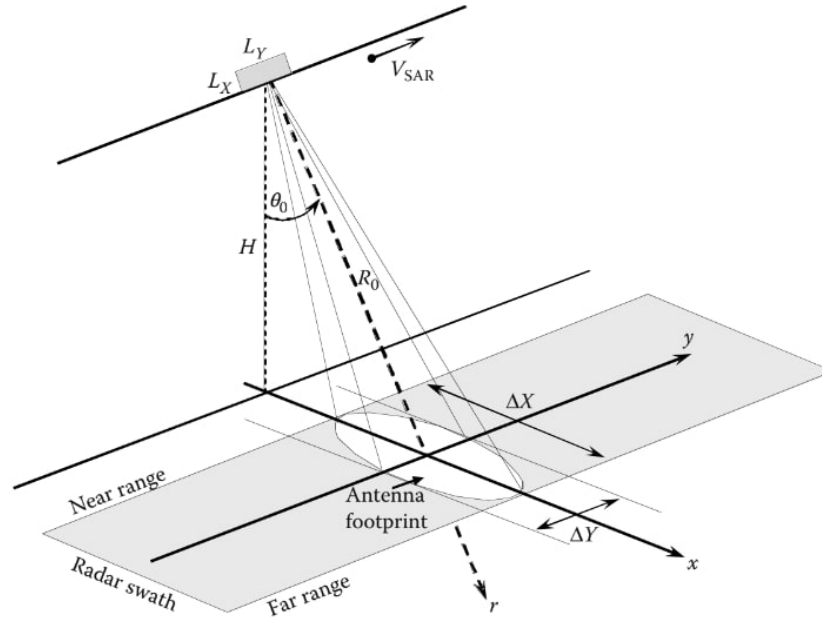


Figure 3.2: SAR imaging geometry in strip-map mode[2]

As shown in the above figure, SAR is a side-looking radar, so as to avoid the confusion of retrieval of backscattered signals coinciding from both directions. This, however, gives rise to slant mapping of the region and hence needs to be corrected later on. It moves in the direction perpendicular to the direction the radar is pointing towards. It is located at a height  $H$  and travels with velocity  $V_{sar}$ . The direction perpendicular to the one it is moving in is called "Azimuth". The radar is adjusted with an incidence angle  $\theta$ . The radial axis or Radar Line of Sight (RLOS) is considered "slant-range". The area it covers under both axis is called "antenna footprint" and the slant range it covers is known as "Radar swath"[2].

SAR has a number of benefits for applications using remote sensing. First off, because of its microwave frequency range, it can see through vegetation, some building materials, and clouds, giving it significant information about features

and things that might be hidden from view in optical images. Additionally, SAR can identify minute variations and changes in the surface by measuring the radar backscatter intensity, which is sensitive to the physical characteristics and structure of the objects. The capacity of SAR to produce high-resolution imaging is one of its primary features. The detection of minute features on the Earth's surface is made possible by SAR's fine spatial resolution, which is made possible by the synthesized aperture approach. Due to this, SAR is especially useful for tasks like mapping land cover, monitoring natural disasters, and urban planning.

Various methods of processing and analysis can be used with SAR data to extract information about the Earth's surface. By providing more details about the polarisation and phase of the radar signals, respectively, different techniques like polarimetry and interferometry improve SAR's capabilities. Applications such as topography mapping, forest monitoring, and deformation analysis are made possible by these techniques. SAR data from numerous satellite projects, including Sentinel-1 from the European Space Agency, RADARSAT from NASA, and ALOS from Japan, have recently become openly available. This has fueled an increase in the number of SAR applications and studies. SAR can be used to combine data from many remote sensing sources, such as optical images, LiDAR, and others, to better comprehend complex Earth systems.

### **3.3 Polarimetric SAR**

Polarimetric Synthetic Aperture Radar is a type of radar remote sensing that captures and analyses radar signal polarization. PolSAR systems transmit and receive signals in multiple polarisation states (e.g., horizontal, vertical, and cross-polarizations) to provide a more thorough understanding of the scattering behavior of objects on the Earth's surface than conventional single-polarization radar systems, which only measure the intensity of the backscattered signal.

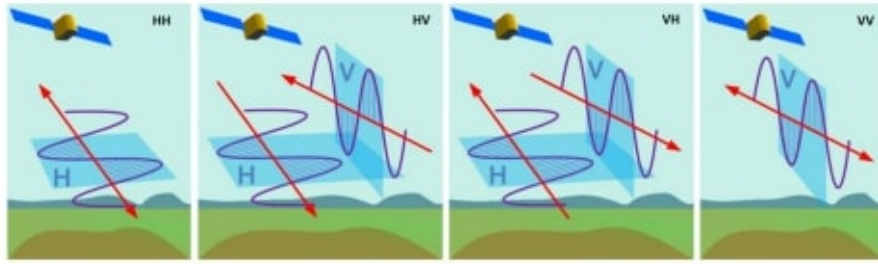


Figure 3.3: Different polarization states of scattered waves[3]

A scattering matrix or coherency matrix, which contains details about the amplitude and phase interactions between the various polarisation channels, is used to represent PolSAR data. This matrix makes it possible to examine the observed targets' polarimetric scattering characteristics. PolSAR data has a number of benefits for applications using remote sensing. It offers improved target-discriminating abilities that makes it possible to distinguish between different kinds of objects and materials according to their polarimetric characteristics. Additionally, it makes it possible to estimate significant structural and physical features of the targets that have been examined, such as surface roughness, vegetational traits, and geometrical qualities.

Radar signals in Full Polarimetric SAR (F-PolSAR) systems can have any combination of polarisations, including horizontal (H), vertical (V), and both cross-polarizations (HV and VH). The most thorough polarisation data is provided by F-PolSAR, which also delivers the maximum amount of detail when examining target scattering patterns. TerraSAR-X which is a German satellite mission operating in X-band is a Full PolSAR system. Dual Polarimetric SAR (D-PolSAR) devices typically send and receive signals in only two polarisation states, horizontal (H) and vertical (V) or cross-polarizations (HV and VH). Although D-PolSAR captures fewer polarisation data than F-PolSAR, it nevertheless allows for accurate target separation and scattering property analysis. ALOS-2 by JAXA is a D-PolSAR system that operates in L-band and is used for forest monitoring.

Compact Polarimetric SAR (C-PolSAR) devices receive signals in both horizontal and vertical polarisations (HV and VH) and broadcast signals in one polarisation state (either H or V). With this setup, a small amount of polarimetric data can be acquired while still keeping a compact system design. RADARSAT-2 which is a C-band Canadian satellite mission is a C-PolSAR system used for agricultural land classification. Signals are sent and received in all four polarisation

states—H, V, HV, and VH—by quadric metric SAR (Q-PolSAR) devices. In comparison to D-PolSAR systems, this configuration offers more polarisation data, improving target differentiation and investigation of the scattering mechanisms. Sentinel-1 by ESA is a C-band Q-PolSAR system used for maritime surveillance. Polarimetric Interferometric SAR (PolInSAR) combines polarimetric and interferometric SAR techniques to deliver elevation and polarimetric information. It entails analyzing the phase differences between two or more PolSAR pictures obtained from marginally various places. Elevation estimate and terrain mapping are two areas where PolInSAR is extremely helpful. TanDEM-X is a private venture of Airbus that generates digital elevation models.

### 3.4 Mathematical concepts for PolSAR data

When the radar emits horizontally or vertically polarized waves the retrieved or backscattered signal can be of either of two polarization which may or may not be different from the polarization of the transmitted wave signal. These backscattering properties are contained in the 2x2 Sinclair matrix:

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

Here  $S_{HH}$  is the horizontally transmitted and horizontally received signal while  $S_{VH}$  is the vertically transmitted and horizontally received signal and so on. In the case of monostatic backscattering,  $S_{VH} = S_{HV}$ , thus becoming target vector  $\Omega$  as follows:

$$\Omega = \begin{bmatrix} S_{HH} \\ \sqrt{2}S_{HV} \\ S_{VV} \end{bmatrix} \quad (3.2)$$

Now, cross product is done with the conjugate transpose  $\Omega^{*T}$  of the above matrix with itself in order to obtain the covariance matrix as follows:

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

Now replacing values of equation 3.2 in equation 3.3 and solving gives the  $C_3$

matrix as follows:

$$C_3 = \begin{bmatrix} \langle S_{HH}S_{HH}^* \rangle & \langle S_{HH}S_{HV}^* \rangle & \langle S_{HH}S_{VV}^* \rangle \\ \langle S_{HV}S_{HH}^* \rangle & \langle S_{HV}S_{HV}^* \rangle & \langle S_{HV}S_{VV}^* \rangle \\ \langle S_{VV}S_{HH}^* \rangle & \langle S_{VV}S_{HV}^* \rangle & \langle S_{VV}S_{VV}^* \rangle \end{bmatrix} \quad (3.4)$$

### 3.5 CNNs in PolSAR Image Classification

Convolutional Neural Networks (CNNs) process and analyse picture data by combining a number of essential elements. The principal elements of a typical CNN are as follows:

1. **Convolutional Layers** The foundational units of CNNs are convolutional layers. They employ convolutional operations on the input image while scanning it with a collection of learnable filters, also referred to as kernels. By computing dot products between the filter weights and the relevant picture regions, these procedures are able to capture local characteristics and patterns, such as edges and textures. In CNNs, feature extraction is carried out via convolutional layers.
2. **Activation Functions** Non-linearity is introduced into the CNN model using activation functions. Rectified Linear Unit (ReLU), which sets negative values to zero while keeping positive values unaffected, and variants like as Leaky ReLU and Parametric ReLU (PReLU) are common activation functions used in CNNs. Activation functions assist in the introduction of non-linear interactions, allowing the model to learn more complicated patterns and representations.
3. **Pooling Layers** Pooling layers are used to reduce the spatial dimensionality of feature maps generated by convolutional layers. Pooling operations that are often utilized are max pooling and average pooling. They minimize spatial resolution while keeping the most significant properties, lowering processing needs and offering some translational invariance.
4. **Fully Connected Layers** Fully connected layers are highly connected layers in which each neuron is linked to every neuron in the layer before it. These layers take the previous layers' flattened output and map it to the required output classes. The CNN can learn high-level representations and produce final predictions thanks to fully connected layers.

5. **Regularization techniques** Dropout is a regularisation technique used in CNNs to prevent overfitting. During training, it randomly picks a subset of neurons in a layer and sets their outputs to zero. This reduces the network's reliance on specific neurons and pushes it to learn more robust and generalizable properties.
6. **Loss Functions** The loss function computes the difference between the CNN's predicted and true outputs. Categorical cross-entropy, binary cross-entropy, and softmax with cross-entropy are all common loss functions used in classification problems. The loss function to be used is determined by the problem at hand.
7. **Optimization Algorithm** During the training process, optimization methods such as Stochastic Gradient Descent (SGD), Adam, or RMSprop are used to update the weights and biases of the CNN. These techniques try to minimize the loss function by modifying the parameters based on backpropagation gradients.

## CHAPTER 4

# Dataset

This section gives information about the 4 datasets that are used to conduct various experiments in this Paper. Following is the table which gives an overview of various datasets:

Name	Region	Sensor	Acquired on	Size	Number of Classes
Flevoland 15	Flevoland, Netherlands	AIRSAR (L-Band)	16 <sup>th</sup> August, 1989	750 x 1024	15
Flevoland 7	Flevoland, Netherlands	AIRSAR (L-Band)	16 <sup>th</sup> June, 1991	750 x 700	7
Landes	Landes, France	AIRSAR (L-Band)	19 <sup>th</sup> June, 1991	1050 x 1000	6
Mysore	Mysore, India	RADARSAT-2 (L-Band)	29 <sup>th</sup> August, 2017	6235 x 3248 (Effective size 3489 x 3352)	11

Table 4.1: Details of Datasets used for experimentation

The first two datasets are of Flevoland, Netherlands which is taken by ESA during different periods of time and has different numbers of classes. Flevoland 15 has 15 classes namely, Water, Forest, Lucerne, Grasses, Peas, Barley, Bare soil, Beet, Wheat 2, Wheat 3, Stembeans, Rapeseed, Wheat, Buildings, and Potatoes. Flevoland 7 has seven classes, namely, Wheat, Rapeseed, Barley, Lucerne, Potatoes, Beet, and Peas. Landes dataset is also taken using AIRSAR by ESA and has 6 classes which are aptly named C1, C2, C3, C4, C5, and C6. The first two are agricultural datasets over Northern Europe while the third is the one in Central Western Europe with unspecified classes but with available ground truth for the agricultural fields. Mysore, however, is taken by RADARSAT-2, is provided by ISRO, and is the newest in the long list of PolSAR datasets studied in this Thesis. It has 11 classes: Ragi, Ginger, Rice, Urban, Magnesite Mine, Water, Arecanut, Banana, Sugarcane, Coconut, and Fallow. A peculiar thing about Indian fields is that they are very small in size and hence need some changes in sample sizes to substantiate the dataset size for training purposes.

The datasets are available in the form of  $C_3$  matrices as discussed in the Background chapter, however, there are also other forms of representation which is  $T_3$  coherency matrix which has a 3D Pauli feature vector component  $k$  instead of  $\Omega$  as in  $C_3$  matrix. The formula for Pauli feature vector  $k$  is as follows:

$$k = \frac{1}{\sqrt{2}} \begin{bmatrix} S_{XX} + S_{YY} & S_{XX} + S_{YY} & 2 \cdot S_{YY} \end{bmatrix}^T \quad (4.1)$$

The Pauli RGB image of the Flevoland 7 dataset and its ground truth is shown in Figure 4.1. The Pauli RGB image of the Flevoland 15 dataset and its ground truth is shown in Figure 4.2. The Pauli RGB image of the Landes dataset and its ground truth are shown in Figure 4.3. For the Mysore dataset, the pseudo-color Freeman-Durden RGB image in the slant range and ground range as well as the Sentinel-1 optical image of the same region over a nearby time frame is shown in Figure 4.4.

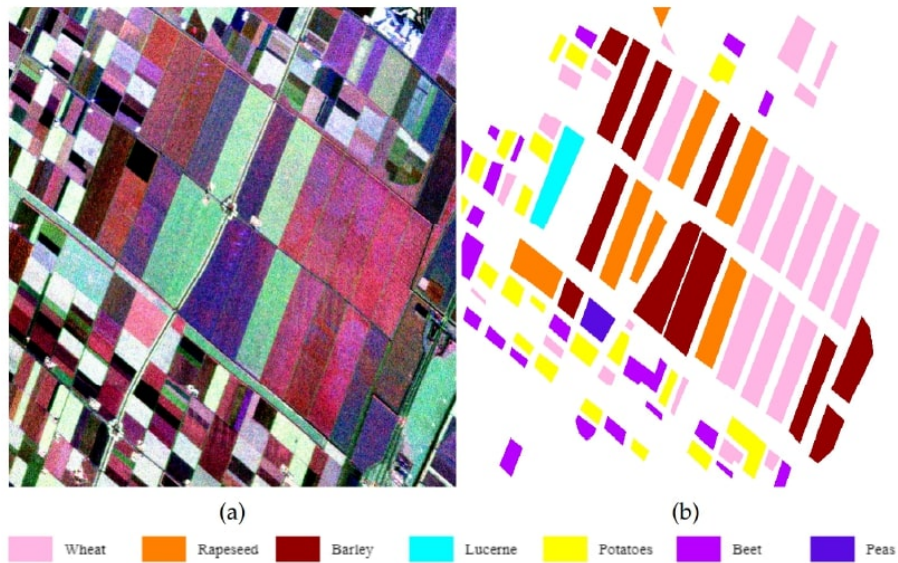


Figure 4.1: (a) Pauli RGB of Flevoland 7 dataset (b) Ground Truth of Flevoland 7 dataset



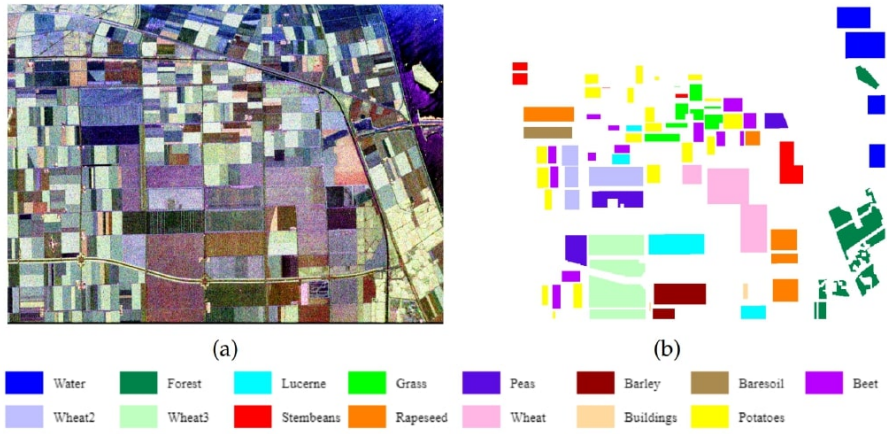


Figure 4.2: (a) Pauli RGB of Flevoland 15 dataset (b) Ground Truth of Flevoland 15 dataset

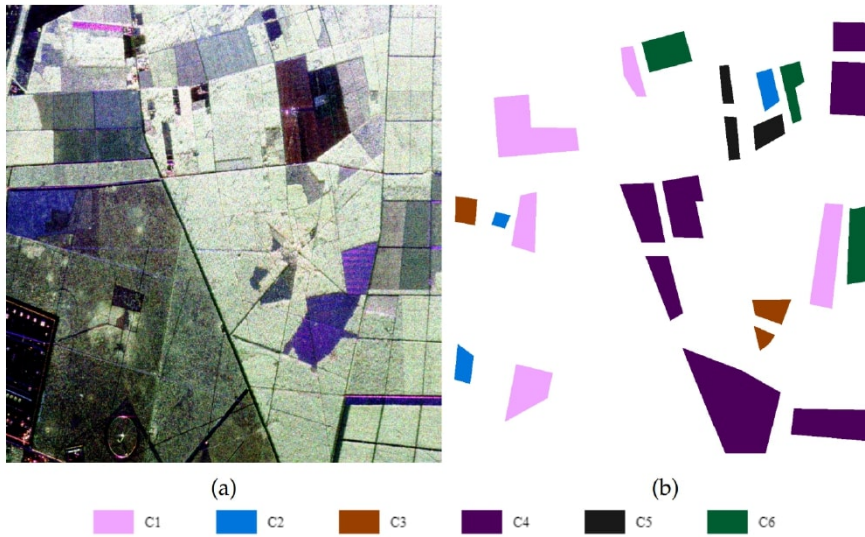


Figure 4.3: (a) Pauli RGB of Landes dataset (b) Ground Truth of Landes dataset

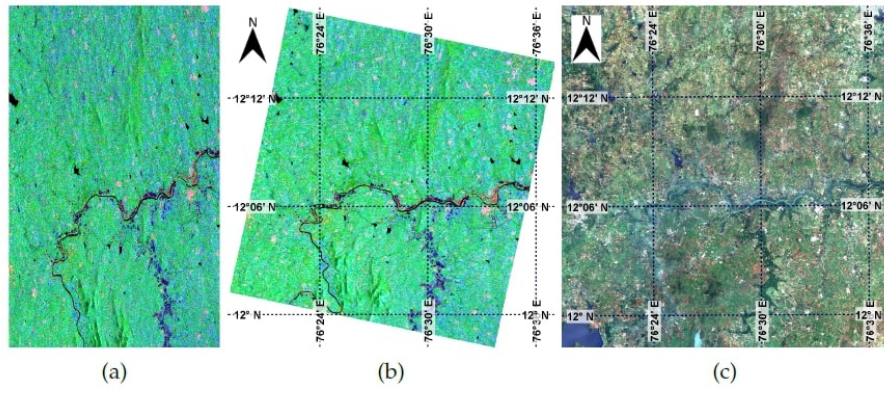


Figure 4.4: Mysore data set showing (a) Pseudo-color Freeman-Durden RGB image in slant-range resolution (b) Pseudo-color Freeman-Durden RGB image ground-range resolution and (c) Sentinel-2 optical image

## CHAPTER 5

# Methodology

The proposed methodology contains several components. Since the given data contains Complex Values, a Complex-Valued Convolutional Neural Network is incorporated in order to leverage the Complex valued nature of the data. Also, the Squeeze and Excitation Network is incorporated in order to utilize the inter-channel dependency that the  $C_3$  matrix values might have. Then, Residual Blocks are added in order to enhance the training by solving the vanishing gradient issues that might be occurring which can cause the model training to be deemed futile after some point. That can be avoided with these residual block which includes the Squeeze and Excitation Blocks. At last, the Gaussian Context Transformer is implemented in order to leverage mathematical normalization for channel attenuation of the data in the squeeze and excitation block by using Gaussian normalization and Gaussian Excitation instead of convolutional components while the squeeze component remains the same. An over-arching view of the model can be seen in the figure below:

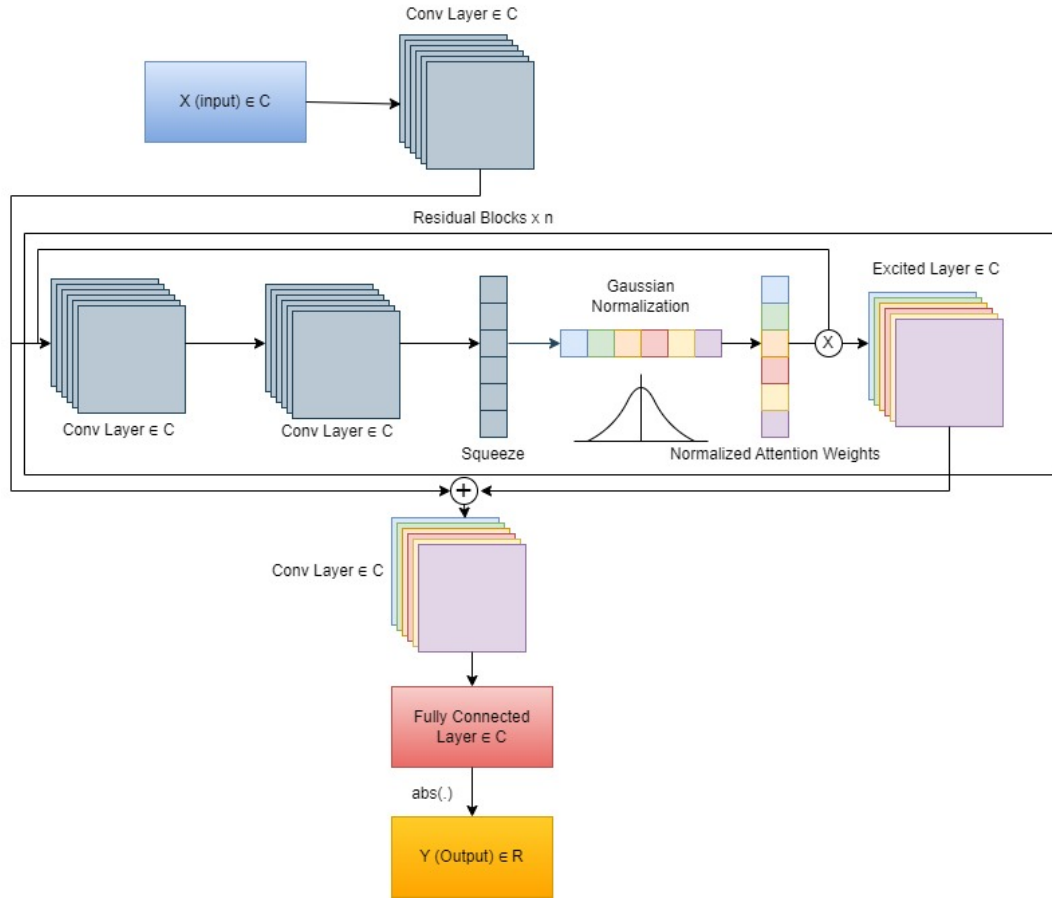


Figure 5.1: Overview of the proposed Model

The parameters of the model for the Flevoland 7 dataset is 649,518. Since the output layer has 7 nodes for the 7 classes in Flevoland 7 dataset, the number of parameters may slightly vary from one dataset to another. The hyperparameters of the proposed model include the learning rate being 0.0001. This is tweaked a little bit from 0.007 to 0.0007 depending on fine-tuning requirements for different datasets. The number of epochs it takes to converge is around 90 to 120 for different scenarios. The batch size is kept to 38 for Flevoland 7 dataset.

Mod name	Parameters Listed
conv5.conv_r.weight	6912
conv5.conv_i.weight	6912
bn3.weight	384
bn3.bias	256
res_blocks.0.conv3.conv_r.weight	147456
res_blocks.0.conv3.conv_i.weight	147456
res_blocks.0.bn1.weight	384
res_blocks.0.bn1.bias	256
res_blocks.0.conv4.conv_r.weight	147456
res_blocks.0.conv4.conv_i.weight	147456
res_blocks.0.bn2.weight	384
res_blocks.0.bn2.bias	256
res_blocks.0.se_block1.conv1.fc_r.weight	1024
res_blocks.0.se_block1.conv1.fc_r.bias	8
res_blocks.0.se_block1.conv1.fc_i.weight	1024
res_blocks.0.se_block1.conv1.fc_i.bias	8
res_blocks.0.se_block1.conv2.fc_r.weight	1024
res_blocks.0.se_block1.conv2.fc_r.bias	128
res_blocks.0.se_block1.conv2.fc_i.weight	1024
res_blocks.0.se_block1.conv2.fc_i.bias	128
res_blocks.0.se_block2.conv1.fc_r.weight	1024
res_blocks.0.se_block2.conv1.fc_r.bias	8
res_blocks.0.se_block2.conv1.fc_i.weight	1024
res_blocks.0.se_block2.conv1.fc_i.bias	8
res_blocks.0.se_block2.conv2.fc_r.weight	1024
res_blocks.0.se_block2.conv2.fc_r.bias	128
res_blocks.0.se_block2.conv2.fc_i.weight	1024
res_blocks.0.se_block2.conv2.fc_i.bias	128
conv6.conv_r.weight	16384
conv6.conv_i.weight	16384
bn4.weight	384
bn4.bias	256
fc.fc_r.weight	896
fc.fc_r.bias	7
fc.fc_i.weight	896
fc.fc_i.bias	7

Sum of trained paramters: 649518  
649518

Figure 5.2: Parameters of the proposed Model for Flevoland 7 dataset

## 5.1 Complex-Valued Convolutional Neural Network (CV-CNN)

In Zhang et al.[4], the author suggests that since the SAR images have both amplitude and phase components, both can be taken into account for CNN application rather than just the Real-Valued Components. Thus, the author builds an entire Complex Valued CNN model, including Complex Input-Output layers, Complex convolution layers, Complex activation function, Complex Pooling Layer, and a complex backpropagation algorithm based on stochastic gradient descent. The architecture of CV-CNN is shown in Figure 2.

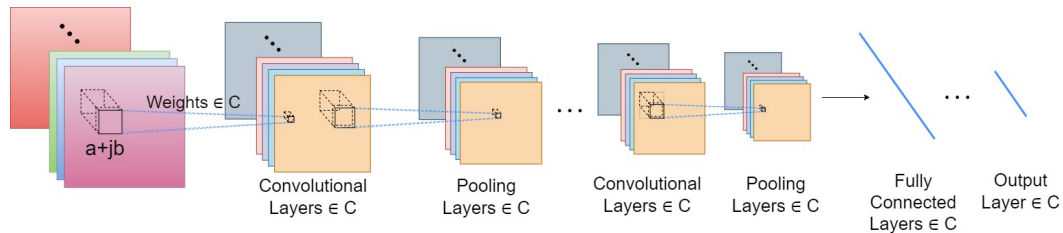


Figure 5.3: CV-CNN Architecture[4]

The covariance matrix is a  $3 \times 3$  symmetric matrix, hence rendering 6 unique elements, however, the diagonal elements are the sole real elements. Thus, in RV-CNN, 9 channels were fed to make the complex numbers as individual real-valued channels. While, in CV-CNN, only 6 channels were used since it can compute complex numbers as a whole, there need not be separate channels for 3 complex elements in the covariance matrix.  $12 \times 12$  window was taken for feeding patches into the CV-CNN and RV-CNN.

The Overall Accuracy of CV-CNN was 96.2% and that of RV-CNN was 95.3%. Thus, phase information is a crucial characteristic of SAR images.

We compute the covariance matrix  $C_3$  from these patches. The data in  $C_3$  matrix is distributed over 6 channels, namely,  $C_{11}, C_{12}, C_{13}, C_{22}, C_{23}, C_{33}$ . This is because the  $C_3$  matrix, which is derived in equation 3.3, is a Hermitian semi-definite positive matrix, hence, the values on only one side of the main diagonal and the diagonal values are useful since the diagonally opposite values are complex conjugates of each other and possess same real and imaginary values. Thus, the aforementioned position elements for each pixel are rendered most useful.

Thus, for an entire image of 9 matrix element channels, these 6 channels are important for feature extraction. Now, it can be seen in equation 3.3 that the diagonal values are real values and the non-diagonal values are imaginary values. The  $C_3$  matrices are passed through the proposed architecture that consists of complex-valued CNN layers, complex batch normalization, complex ReLU, Gaussian Context Transformer comprising of SE-Net/SEResNet, adaptive pooling, fully connected layer, and softmax.

The complex-valued CNN takes care of the complex values of the PolSAR image. The complex-valued feature output of  $l^{th}$  layer  $M_k^{(l)} \in \mathbb{C}^{F \times F \times K \times I}$  is convolved with complex-valued kernel  $w_{ik}^{(l+1)} \in \mathbb{C}^{B_1 \times H_1 \times K}$  along with an addition of bias to produce a feature  $F_i^{(l+1)}$  at  $(l+1)^{th}$  layer.

$$\begin{aligned}
F_i^{(l+1)} &= \sum_{k=1}^K w_{ik}^{(l+1)} * M_k^{(l)} + b_i^{(l+1)} \\
&= \sum_{k=1}^K (\Re\{w_{ik}^{(l+1)}\} \cdot \Re\{M_k^{(l)}\}) - \Im\{w_{ik}^{(l+1)}\} \cdot \Im\{M_k^{(l)}\} \\
&\quad + j \sum_{k=1}^K (\Re\{w_{ik}^{(l+1)}\} \cdot \Im\{M_k^{(l)}\}) + \Im\{w_{ik}^{(l+1)}\} \cdot \Re\{M_k^{(l)}\} \\
&\quad + b_i^{l+1}
\end{aligned} \tag{5.1}$$

The produced feature is passed through batch normalization, followed by non-linear activation. Here, we use complex-ReLU for activation as

$$M_i^{l+1} = ReLU(\Re\{F_i^{l+1}\}) + jReLU(\Im\{F_i^{l+1}\}) \tag{5.2}$$

$M_i^{l+1}$  is passed through the squeeze-excitation network to produce a feature that is enriched with inter-channel correlation.

## 5.2 Complex-Valued Squeeze and Excitation Network

Squeeze and Excitation Network introduced by [5] involves two basic operations: squeezing and stimulating. By using global average pooling, the squeezing process aggregates the spatial dimensions of the input feature map into a global descriptor. This decreases spatial size while keeping channel-specific information.

The exciting operation focuses on capturing channel-specific dependencies and generating channel-specific significance weights. It employs a minimal number of fully connected layers to simulate channel interdependencies and generates a channel-wise weighting vector. This vector is then applied to the original feature map element by element to emphasize or downplay specific channels. The additional computational cost does not occur due to a content-aware mechanism that weighs each channel on a global level adaptively.

In the squeeze operation, global pooling is done for each channel. During excitation, the module attenuates different weights for each channel using two fully connected layers. Then ReLU and Sigmoid are there at the end of each Excitation block. In the end, the output of Sigmoid is then scaled by multiplying it with the input feature map of the convolution layer that was added before the SEBlock.

The figure below gives an overview of the architecture explained above:

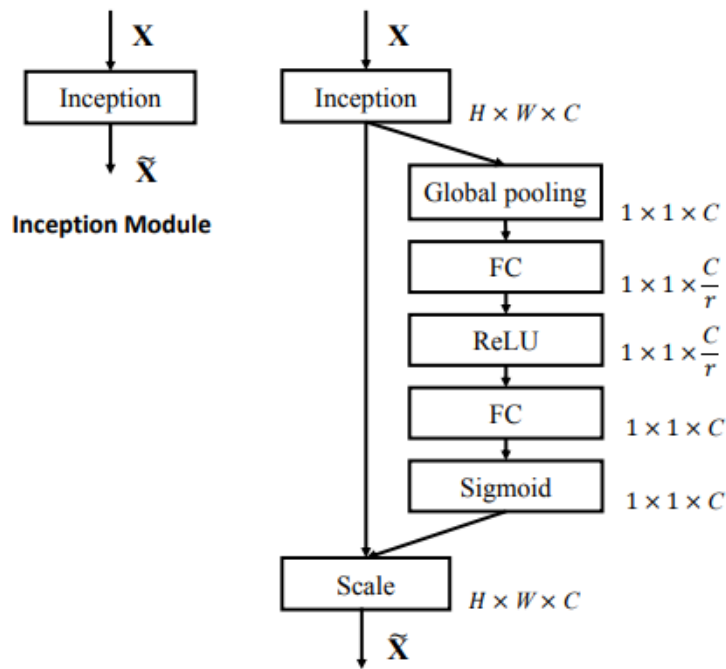


Figure 5.4: Squeeze and Excitation Network Block[5]

Here,  $H$  and  $W$  are the height and width of the data respectively.  $C$  is the number of channels and  $r$  is the ratio using which the channels are calibrated using fully connected layers.



## 5.3 Complex-Valued Squeeze and Excitation Residual Network

ResNet is a well-known deep learning architecture that overcomes the difficulty of training very deep neural networks. It introduces residual connections, also known as skip connections, which allow the network to learn residual mappings rather than the intended underlying mapping directly. This aids in minimizing the degradation issue that emerges as network depth increases.

Combining ResNet and SENet approaches is conceivable by incorporating the squeeze and excitation mechanisms within the residual blocks of a ResNet design. This integration seeks to improve the ResNet model's representational capability by including channel-wise feature recalibration.

The figure below shows the overview of the SEResNet model:

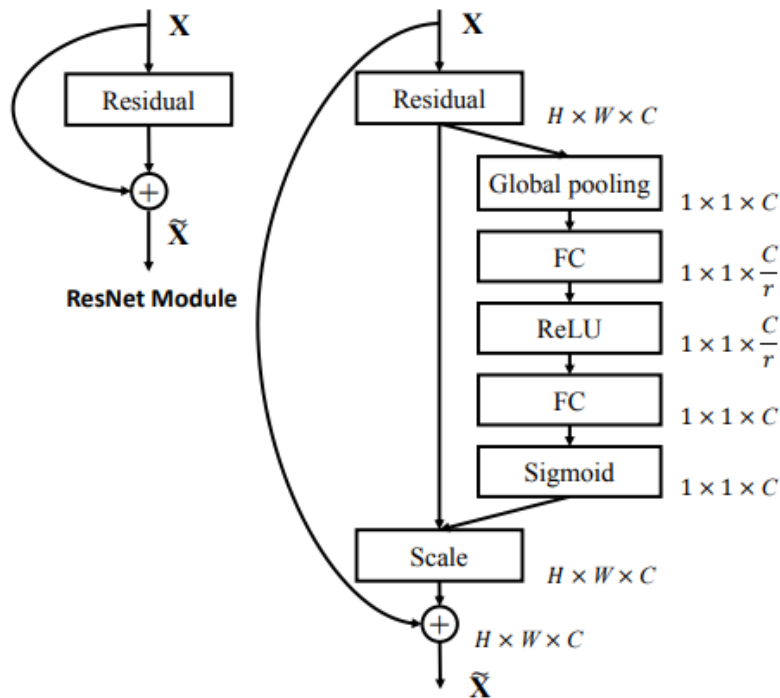


Figure 5.5: Squeeze and Excitation Network Residual Block[5]

## 5.4 Complex-Valued Gaussian Context Transformer

Ruan et al.[6] state that the channel attention blocks normally learn the relationship between global contexts and attention activations by fully connected layers or linear transformation. The authors suggest that the relationship between channels can be a mathematical distribution. Hence, they arrive at exploiting the Gaussian Distribution of the data in its classification. The method does not introduce many parameters into the fold unlike its predecessors, rather it has two variants, one, GCT-B0 which is parameter-free, and the GCT-B1 which has one parameter. It consists of three stages. One is Global Context Aggregation. Adaptive average pooling is used for the purpose in this case. Then normalization is done using the global average values and then at last global context excitation is done. Thus, in practice, attention learning is changed to attention distribution with a determined hunch upon the distribution of data. Thus, the main goal of the Gaussian Context Transformer is to remove the learning component of the channel attention mechanism and introduce a hard set of mathematical learning for channel attention weights in the squeeze and excitation part of any model. This signifies that the channel weightage can be set mathematically rather than learned through a neural network if the inclination of the overall distribution of the data is known. The architecture of GCT is shown in Figure 5.6.

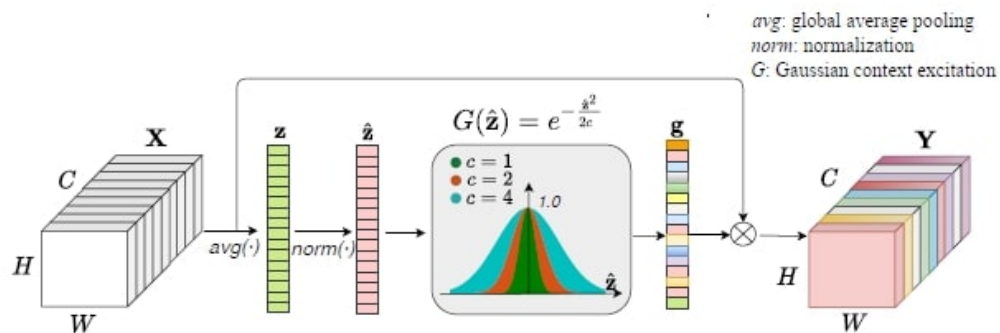


Figure 5.6: Diagram of Gaussian Context Transformer[6]

Here,  $c$  stands for standard deviation. This can be set manually ranging from  $c=1$  to  $c=4$  for the GCT-B0 model while for GCT-B1, it is a learnable parameter. The highest top-5 of GCT-Resnet50-B0 is 93.86% on the Imagenet validation set for  $c=2$  and that of GCT-Resnet50-B1 is 93.81% for the range  $c=2$  to  $c=4$ .

For the current application of SAR images, the covariance matrix of the SAR images follows Wishart Distribution but just to test the hypothesis whether or not the relations between the channels are improved by adding mathematical distribution or not in SAR images, the Gaussian Context Transformer has been implemented in the SEResNet and SENet models in order to observe if the accuracies are comparable to that of those.

First of all, the SE Blocks were modified with the complex-valued normalization and customized excitation blocks in order to retrofit into the complex-valued models. First, the global averages were taken through Adaptive average pooling. Then complex normalization was done using taking amplitude while calculating the standard deviation. Mean was calculated entirely in Complex Domain.

The squeeze and excitation blocks of the SENet and the SEResNet models mentioned above are changed in order to attenuating channel weights according to Gaussian distribution rather than learning through neural network as suggested in the above hypothesis. After the global average pooling, the complex Gaussian normalization and complex Gaussian excitation are performed instead of the convolutional layers learning the entire relations between the existing six channels' global complex values. The values for a and b are kept in the above equation in Figure 3 as to be 1 and 0 respectively while the values of c are kept in integer terms in the range of 1 to 4.

The Complex Gaussian normalization is done as follows:

$$Z' = \frac{Z - \bar{Z}}{Z_{std}} \quad (5.3)$$

where,

$$\bar{Z} = \Re\{\bar{Z}\} + j \cdot \Im\{\bar{Z}\} \quad (5.4)$$

$$Z_{std} = \sqrt{\frac{\sum |Z - \bar{Z}|^2}{c}} \quad (5.5)$$

Now, this  $Z'$  is the normalized complex value of the channel attention weights. Now in order to test if SAR data performs better with the Gaussian distribution

or not channel channel normalization, the distribution is applied to  $Z'$  as follows:

$$G(Z') = e^{-\frac{(Z'-b)^2}{2c}} \quad (5.6)$$

The above process is called Gaussian Excitation since the normalized channel complex weights are fed into Gaussian distribution in order to obtain Complex Gaussian excited values. The values of the coefficient of  $G(Z')$  term are kept as 1, while that of  $b$  is kept as 0 and that of  $c$  kept from 1 to 4.

The model can be seen as follows:

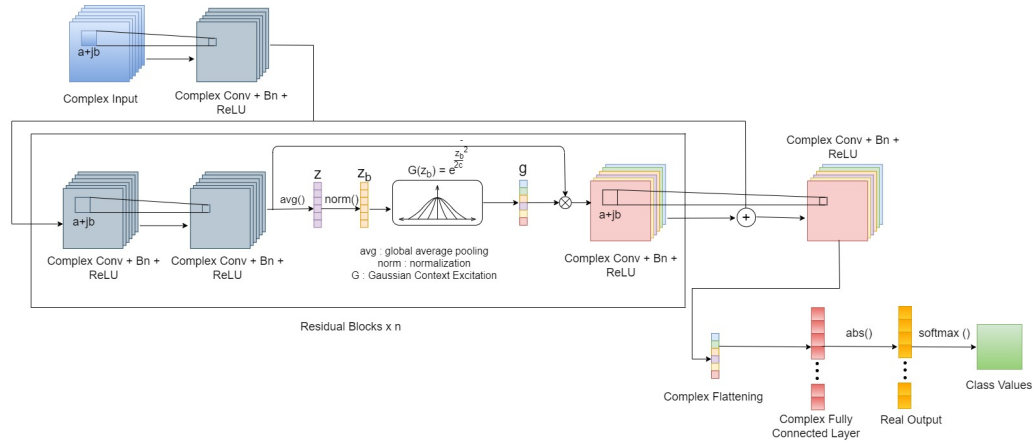


Figure 5.7: Diagram of Gaussian Context Transformer

## 5.5 Loss Function and Optimization

The loss function used here is the Categorical Cross Entropy function. This is because the activation function for the output layer is chosen to be softmax and since there is a multiclass classification problem, the best function here is Categorical Cross Entropy in measuring the dissimilarity between the predicted probability distribution and the true probability distribution of the target classes. The formula for the same is as follows:

$$CE = \sum_{i=1}^{outputsize} y_i \cdot \log \hat{y}_i \quad (5.7)$$

Optimizer chosen for this model is Adam which stands for Adaptive Momentum Optimizer. Adam's fundamental idea is to adaptively alter the learning rate for each parameter in a neural network based on estimates of the gradient's first and second momentums. This variable learning rate enables Adam to outperform

traditional SGD on a wide range of optimization tasks and converges faster.

For each parameter in the network, the method maintains an exponentially decaying average of past gradients (first momentum) and an exponentially decaying average of past squared gradients (second momentum). Weighted moving averages are used to compute these averages, with higher weights applied to recent gradients.

During each training cycle, Adam changes the model's parameters by mixing current gradient information with historical gradient information. The update rule has a gradient term as well as a term that accounts for historical gradient information. Based on the magnitudes of these averaged gradients, the learning rate is adaptively scaled.

Adam also features bias correction algorithms to help reduce the effects of initialization bias at the start of training. This bias adjustment aids the optimizer's performance during the initial iterations.

## Experimental Results

The experimental results with the complete model as shown in Figure 5.6 or in Figure 5.1 implemented on all the datasets mentioned in Chapter 4 of Dataset. The best results out of all the variations of  $c$  from 1 to 4 are mentioned in the respective tables for both the SENet with GCT as well as SEResNet with GCT model variants. Please find the findings below:

In Flevoland 15 dataset, it is clearly seen that the accuracy scores of the SENet model are clearly more than that of the SEResNet model. This is because neither the depth of the architecture is very large nor the dataset is significantly large and hence the SEResNet variant leads to simpler weights weighing over the finer feature learning and leads to reduced accuracy results. Following are the accuracies found on the Flevoland 15 dataset for the proposed model :

Class Label	GCT + Without Skip Connections (c=2)	GCT + With Skip Connections (c=2)
Water	100.00	100.00
Forest	100.00	100.00
Lucerne	100.00	90.00
Grasses	90.00	94.44
Peas	100.00	95.24
Barley	100.00	90.00
BareSoil	100.00	100.00
Beet	95.24	95.24
Wheat2	94.74	91.67
Wheat3	100.00	100.00
Stembeans	100.00	100.00
Rapeseed	100.00	88.89
Wheat	91.67	96.43
Buildings	100.00	100.00
Potatoes	100.00	100.00
OA	<b>98.92</b>	<b>96.37</b>

Table 5.1: Comparison Results on Flevoland 15 across GCT model with the best accuracy with and without skip connections

The results images of the same are shown below:



Figure 5.8: Resulting image of Flevoland 15 dataset using GCT model (c=2) with the best performance without skip connections



Figure 5.9: Resulting image of Flevoland 15 dataset using GCT model ( $c=2$ ) with the best performance with skip connections

As can be seen from the figure, the class of Lucerne is misclassified later in the top-left corner of the image. There are also other classes such as Grasses and Wheat which are correctly classified in the latter rather than in the first result.

In Flevoland 7 dataset, it is clearly seen that the accuracy scores of both the SENet model and the SEResNet model are the same. This is because some classes such as Wheat which is found in abundance in the dataset are classified better with the skip connections in the later model while in classes like Beet where the dataset is scarce and contributed the finer feature involvement due to skip connections causing trouble in order to train better weights for the latter model and hence the first model performs better in this case. Following are the accuracies found on the Flevoland 7 dataset for the proposed model :

Class Label	GCT + Without Skip Connections (c=2)	GCT + With Skip Connections (c=2)
Wheat	99.09	100.00
Rapeseed	100.00	100.00
Barley	100.00	100.00
Lucerne	100.00	100.00
Potatoes	100.00	100.00
Beet	100.00	96.00
Peas	100.00	100.00
OA	<b>99.69</b>	<b>99.69</b>

Table 5.2: Comparison Results on Flevoland 7 across GCT model with the best accuracy with and without skip connections

The results images of the same are shown below:

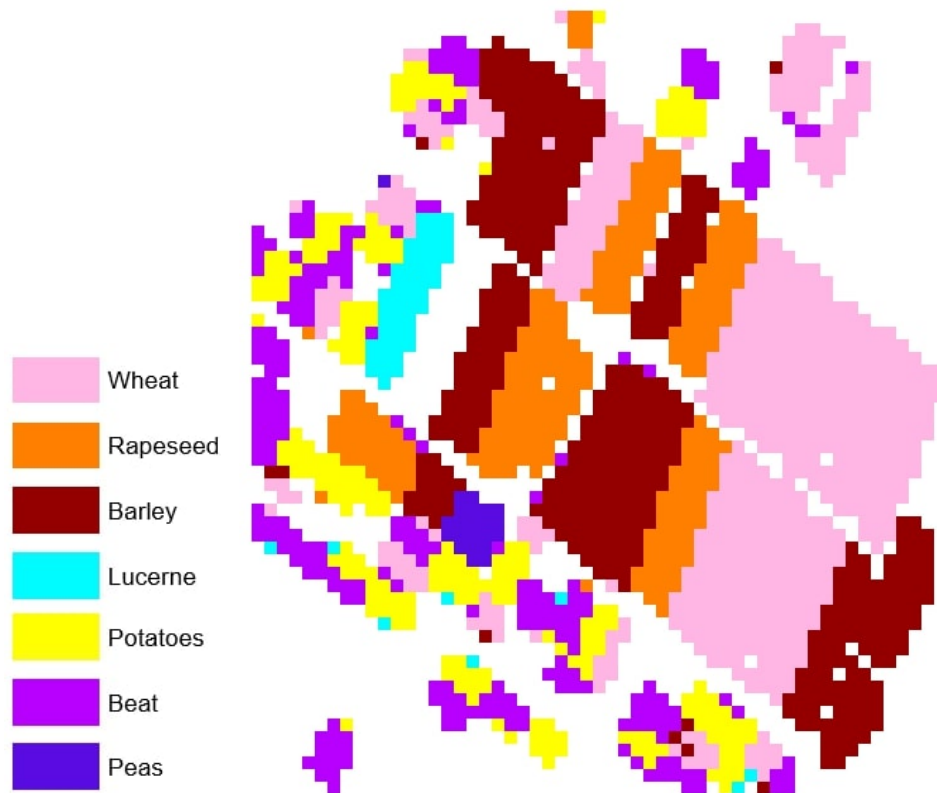


Figure 5.10: Resulting image of Flevoland 7 dataset using GCT model (c=2) with the best performance without skip connections



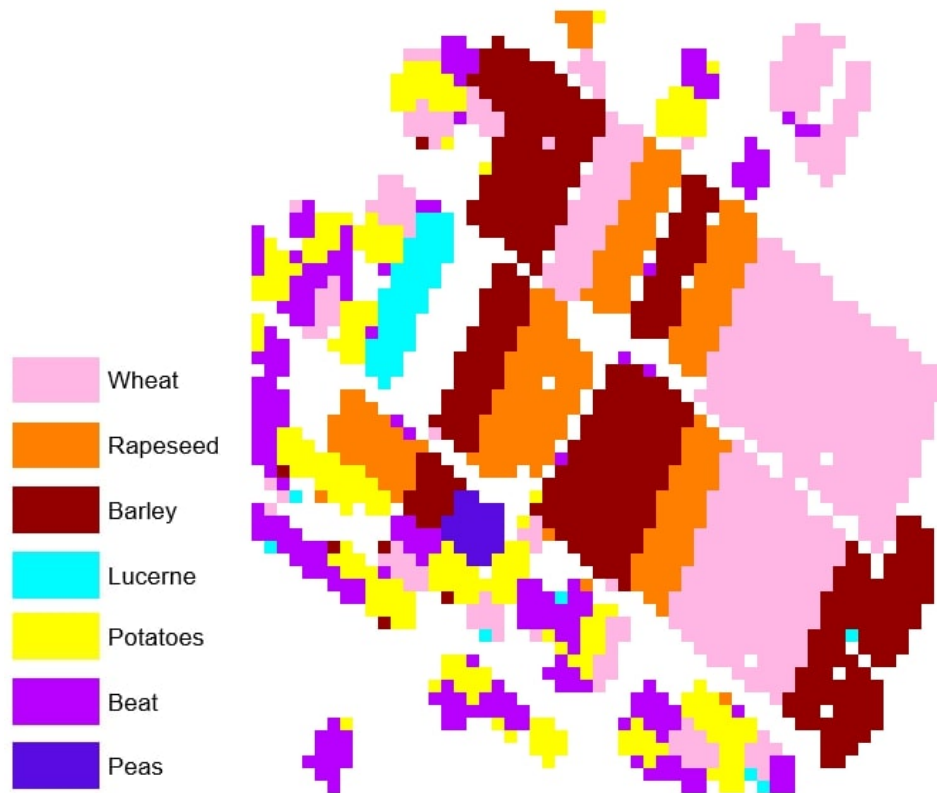


Figure 5.11: Resulting image of Flevoland 7 dataset using GCT model ( $c=2$ ) with the best performance with skip connections

As can be seen in the figures above, the Wheat class is misclassified in several regions in the first model rather than the 2nd and in the Beet class the position is vice versa. Hence, the accuracy table can be verified with the learning through these visual proofs and comparing it with the ground truths of the Datasets in Chapter 4.

In the Landes dataset, the results are similar in nature as in the F15 dataset. This is partly because the size of both these datasets is larger than the F7 dataset. It can be said that since the F7 dataset is smaller in comparison to these Datasets, the effect of Residual blocks is just slightly seen in some exceptional classes. In this case, as well, it can be seen that the accuracy of classes C2, C4, and C6 decreases in the latter model since the residual learning weights interfere with proper learning in scarce classes such as C2 and C6. Following are the accuracies found on the Landes dataset for the proposed model :

Class Label	GCT + Without Skip Connections (c=1)	GCT + With Skip Connections (c=3)
C1	98.91	98.91
C2	100.00	92.31
C3	100.00	100.00
C4	100.00	99.35
C5	100.00	100.00
C6	100.00	93.75
OA	<b>99.08</b>	<b>98.47</b>

Table 5.3: Comparison Results on Landes across GCT model with the best accuracy with and without skip connections

The results images of the same are shown below:

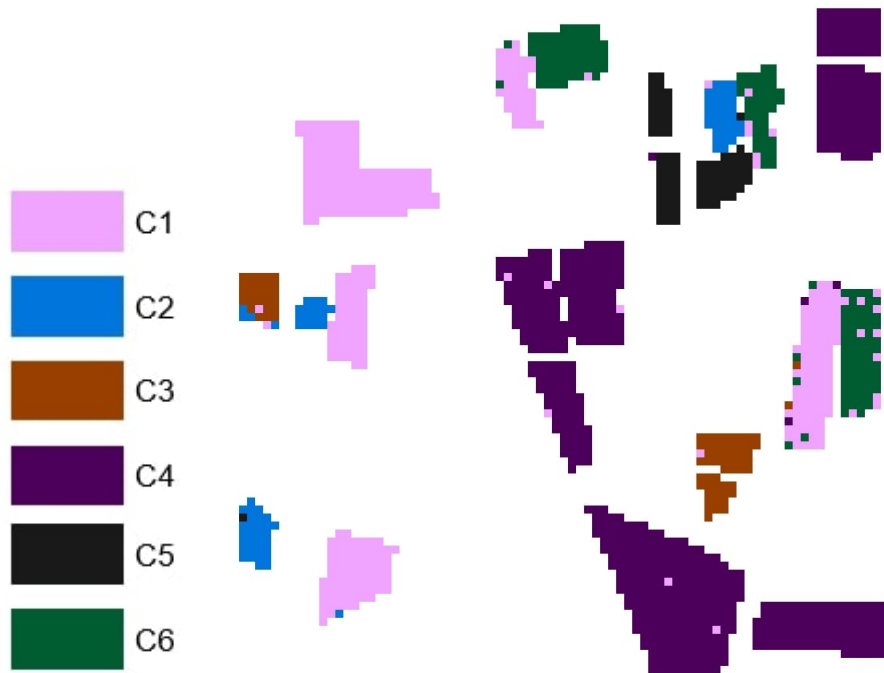


Figure 5.12: Resulting image of Landes dataset using GCT model (c=1) with the best performance without skip connections

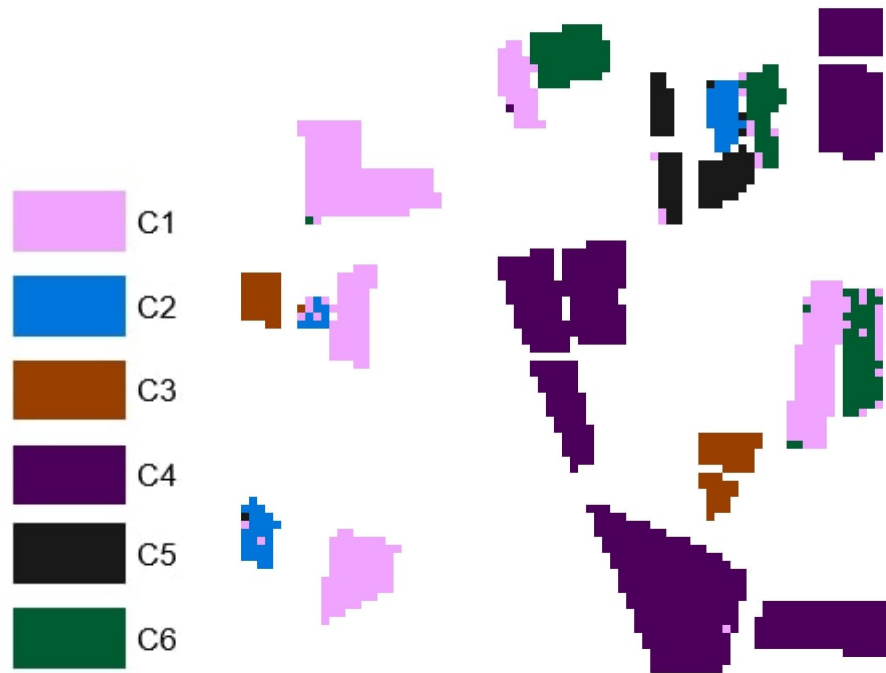


Figure 5.13: Resulting image of Landes dataset using GCT model ( $c=3$ ) with the best performance with skip connections

As can be seen from the figure the classes of C2 and C6 are scarcely distributed and hence the Residual blocks incorporate the simpler learnings producing the lower degree of polynomial in order to discriminate between the classes and hence resulting in misclassification.

Here, the experiment has not been able to be conducted with the Mysore dataset since the Indian dataset has a smaller size of fields and the  $8 \times 8$  size of patches have been used to sample the lands rather than the regular  $12 \times 12$  size patches due to which as a smaller number of patches have been generated for some classes with the number of patches in some classes as low as 1. Due to this reason, the patches cannot be properly distributed in the training and testing samples without the use of data augmentation. Hence, the use of Data Augmentation becomes necessary.

## 5.6 Data Augmentation

Data Augmentation is done in order to see the effects of significantly more data with the models on different datasets of Flevoland 15, Flevoland 7, Landes, and Mysore. The data augmentation is performed using the tool of PolSARPro v 6.0

Biomass edition which is available publicly. The augmentation is done using the  $C_3$  filtering techniques on the original dataset. The techniques of the Box Car filter, Lee Sigma Filter, and the Mean Shift Filter have been proven to be very effective in order to achieve excellent results. For this purpose, the software of PolSARPro v 6.0 Biomass Edition[25] is used in order to augment the  $C_3$  matrices with the aforementioned image processing techniques.

The Boxcar filter[26] and the Lee Sigma filter are two popular image-filtering algorithms, each with its own set of benefits. Because it requires averaging pixel values within a given neighborhood, the Boxcar filter is simple and fast to compute. Its simple implementation makes it computationally efficient, particularly in small neighborhoods. Furthermore, the Boxcar filter is noted for its ability to preserve edges without excessive blurring, making it ideal for applications requiring precise transitions. Furthermore, by averaging pixel values, it efficiently eliminates random noise in photos.

The Lee Sigma filter[27], on the other hand, has several notable advantages. As an adaptive filter, it adjusts the filtering strength based on the image's local statistics. Because of this adaptability, it can better maintain image features while successfully decreasing noise. The Lee Sigma filter excels in reducing speckle noise in radar and synthetic aperture radar (SAR) images. In contrast to typical filters, which can blur texture and fine details, the Lee Sigma filter is designed to preserve texture and fine details while lowering noise.

The Mean Shift filter[7] is an adaptive filter that automatically modifies its spatial and range kernels based on the local image structure. Because of this adaptability, it can effectively handle images with different textures, structures, and noise levels. The Mean Shift filter, unlike linear filters, conducts non-linear filtering by taking into account the local distribution of pixel values and assigning weights appropriately. This non-linear filtering property aids in the preservation of crucial visual features and details while decreasing noise. Another benefit of the Mean Shift filter is its ability to maintain image borders and bounds. It keeps sharp transitions between regions by taking into account spatial and range information, minimizing excessive blurring, and preserving edge clarity.

## Experimental Results

Due to the above-mentioned advantages of Box Car, Lee Sigma, and Mean Shift filters, these filters are used in carrying out data augmentation using the Pol-SARPro v6.0 Biomass Edition [25]. The process to download it is very simple and it is an open-source software developed by European Space Agency. The way to download and use can be found on the website in the reference attached. Due to the data augmentation, a little edge is obtained in classifying the classes in datasets other than Mysore in which some classes have less sample size.

As can be seen in the table below for the Flevoland 15 dataset, the accuracy of the SEResNet model is more than that of the SENet. This is because the dataset size is increased and previously misclassified classes due to less number of sample sizes are no longer at a disadvantage and leverage the use of residual blocks. Following are the accuracies found on the Flevoland 15 dataset with the data augmentation for the proposed model :

<b>Class Label</b>	<b>GCT + Data Augmentation + Without Skip Connections (c=1)</b>	<b>GCT + Data Augmentation + With Skip Connections (c=3)</b>
<b>Water</b>	100.00	99.21
<b>Forest</b>	99.36	100.00
<b>Lucerne</b>	100.00	98.78
<b>Grasses</b>	98.59	100.00
<b>Peas</b>	100.00	100.00
<b>Barley</b>	98.39	98.39
<b>BareSoil</b>	100.00	100.00
<b>Beet</b>	100.00	100.00
<b>Wheat2</b>	96.74	95.65
<b>Wheat3</b>	98.97	99.49
<b>Stembeans</b>	100.00	100.00
<b>Rapeseed</b>	100.00	100.00
<b>Wheat</b>	99.35	100.00
<b>Buildings</b>	100.00	100.00
<b>Potatoes</b>	100.00	100.00
<b>OA</b>	<b>99.38</b>	<b>99.45</b>

Table 5.4: Comparison Results on Flevoland 15 with Data Augmentation across GCT model with the best accuracy with and without skip connections

The results images of the same are shown below:

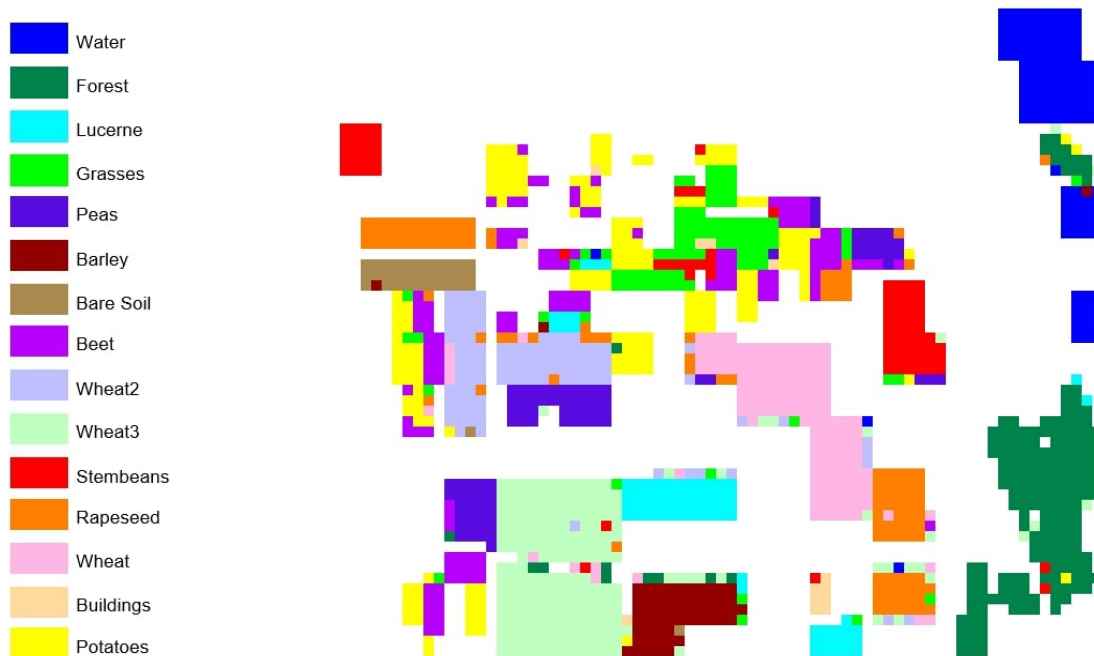


Figure 5.14: Resulting image of Flevoland 15 dataset with Data Augmentation using GCT model ( $c=1$ ) with the best performance without skip connections



Figure 5.15: Resulting image of Flevoland 15 dataset with Data Augmentation using GCT model ( $c=3$ ) with the best performance with skip connections

As can be seen in the images above, the previously misclassified classes of

Wheat and Grass are now classified correctly in the SEResNet model and are better than the first variant.

The same can be seen in the Flevoland 7 dataset, in which the Beet class is classified better by SEResNet than the SENet with data augmentation as the Residual block learning is leveraged more if the size of the dataset increases. Following are the accuracies found on the Flevoland 7 dataset with Data Augmentation for the proposed model :

<b>Class Label</b>	<b>GCT + Data Augmentation + Without Skip Connections (c=4)</b>	<b>GCT + Data Augmentation + With Skip Connections (c=3)</b>
<b>Wheat</b>	100.00	100.00
<b>Rapeseed</b>	100.00	100.00
<b>Barley</b>	100.00	100.00
<b>Lucerne</b>	100.00	100.00
<b>Potatoes</b>	100.00	100.00
<b>Beet</b>	98.62	99.31
<b>Peas</b>	100.00	100.00
<b>OA</b>	<b>99.87</b>	<b>99.94</b>

Table 5.5: Comparison Results on Flevoland 7 with Data Augmentation across GCT model with the best accuracy with and without skip connections

The results images of the same are shown below:

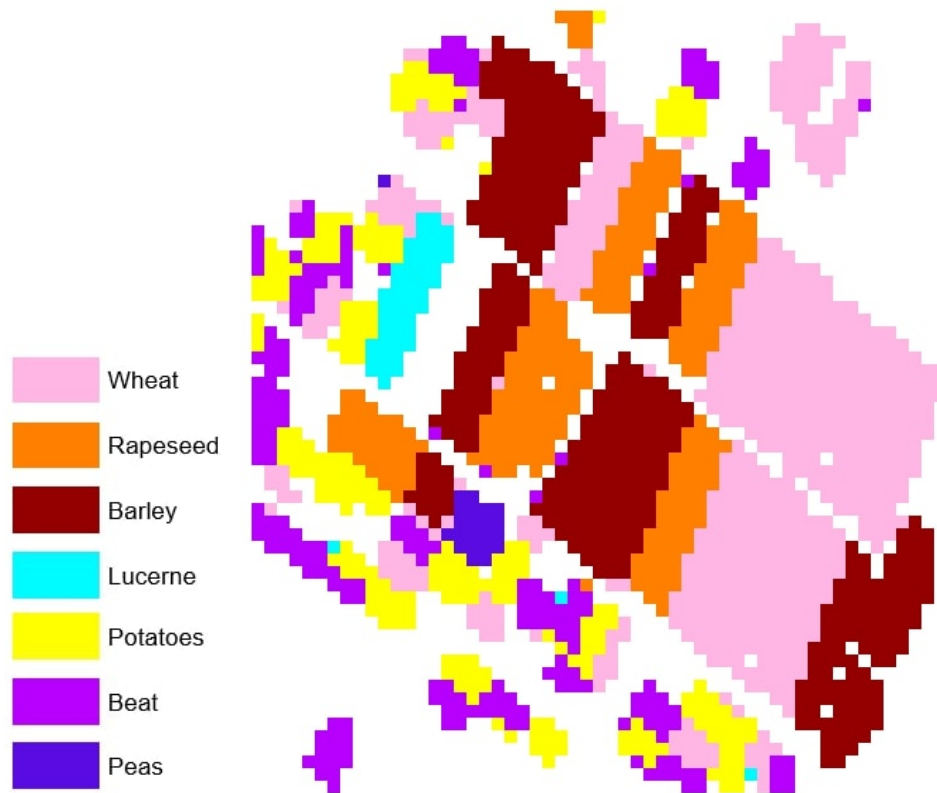


Figure 5.16: Resulting image of Flevoland 7 dataset with Data Augmentation using GCT model ( $c=4$ ) with the best performance without skip connections



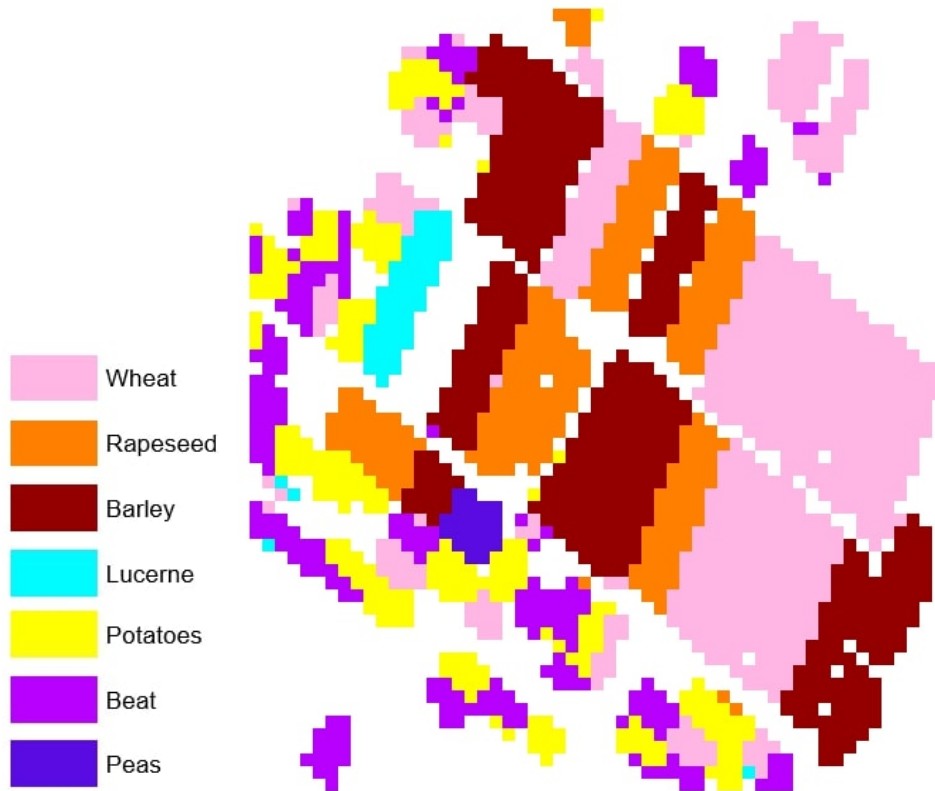


Figure 5.17: Resulting image of Flevoland 7 dataset with Data Augmentation using GCT model ( $c=3$ ) with the best performance with skip connections

As can be seen in the figures, the accuracy table depicts the closest accuracy to the ground truth for the SEResNet model for the  $c=3$  variant which is 99.94%, and gaps in between are filled as these classes are classified as nothing while spatial learning compels them to classify as features of their nearby positions. Hence, it looks slightly different than the ground truth. However, the misclassification rate is very low for the same.

For the Landes dataset, it can be seen that the SEResNet has an edge with better classification rates in the C2 class which was scarcely scattered without the data augmentation. However, with the data augmentation, it can leverage the residual block learning mechanism in order to learn more complex features as the size of the sample is abundant and yet the vanishing gradient problem is avoided. Following are the accuracies found on the Landes dataset with Data Augmentation for the proposed model :

Class Label	GCT + Data Augmentation + Without Skip Connections	GCT + Data Augmentation + With Skip Connections
	(c=3)	(c=2)
C1	98.81	98.57
C2	95.65	100.00
C3	100.00	100.00
C4	100.00	100.00
C5	100.00	100.00
C6	96.39	96.99
OA	<b>99.10</b>	<b>99.30</b>

Table 5.6: Comparison Results on Landes with Data Augmentation across GCT model with the best accuracy with and without skip connections

The results images of the same are shown below:

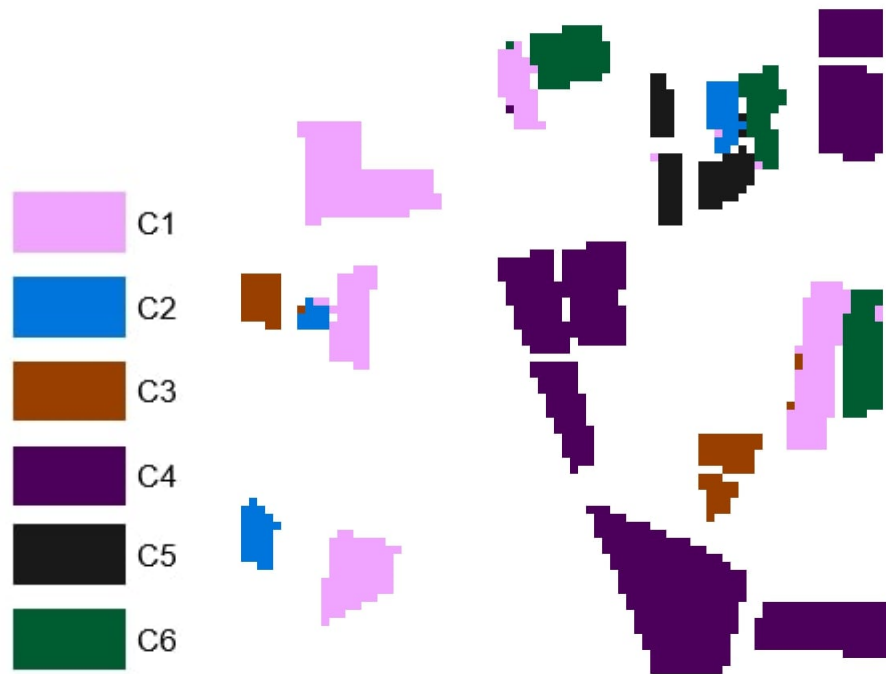


Figure 5.18: Resulting image of Landes dataset with Data Augmentation using GCT model (c=3) with the best performance without skip connections

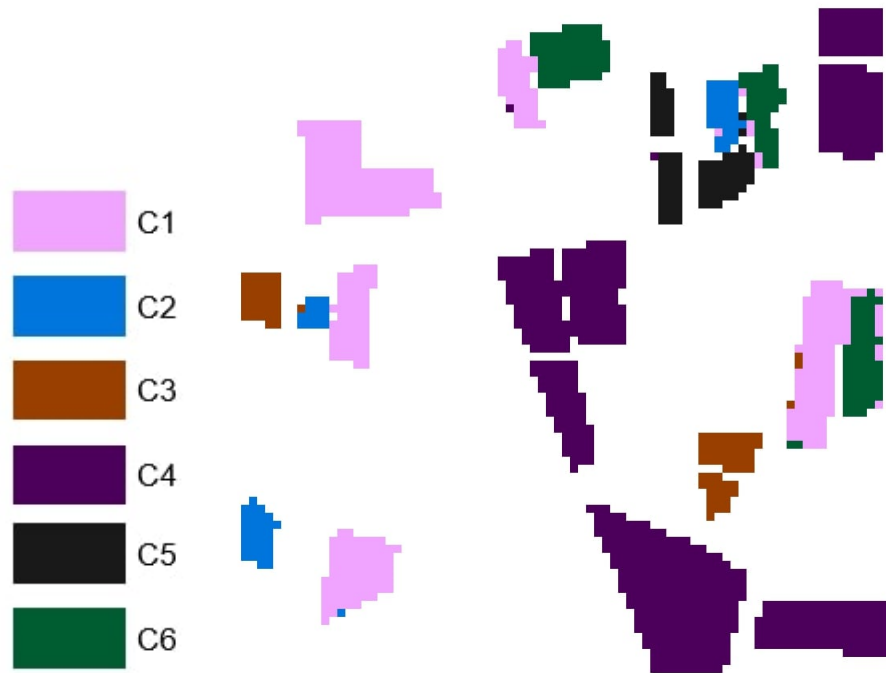


Figure 5.19: Resulting image of Landes dataset with Data Augmentation using GCT model ( $c=2$ ) with the best performance with skip connections

As can be seen in the figures above that the C2 class is now better classified in the SEResNet model rather than in the SENet model while all the other classes are more or less equally classified and the latter is able to catch up with the former after the data augmentation.

For the Mysore dataset, the accuracy factor without data augmentation is null, hence, the only data available is with the data augmentation. Here, it can be seen that the SEResNet has a better overall accuracy score than the SENet. This is because it is better able to classify the scarcely sampled Magnesite Mine class than its counterpart which can be due to the residual block learning advantage or skip connections. Following are the accuracies found on the Mysore dataset with Data Augmentation for the proposed model :

<b>Class Label</b>	<b>GCT + Data Augmentation + Without Skip Connections (c=3)</b>	<b>GCT + Data Augmentation + With Skip Connections (c=2)</b>
<b>Ragi</b>	100.00	100.00
<b>Ginger</b>	100.00	100.00
<b>Rice</b>	100.00	100.00
<b>Urban</b>	100.00	100.00
<b>Water</b>	100.00	98.57
<b>Arecanut</b>	100.00	100.00
<b>Banana</b>	100.00	100.00
<b>Sugarcane</b>	100.00	100.00
<b>Coconut</b>	100.00	100.00
<b>Fallow</b>	100.00	100.00
<b>Magnesite Mine</b>	81.82	100.00
<b>OA</b>	<b>99.20</b>	<b>99.60</b>

Table 5.7: Comparison Results on Mysore with Data Augmentation across GCT model with the best accuracy with and without skip connections

The results images of the same are shown below:

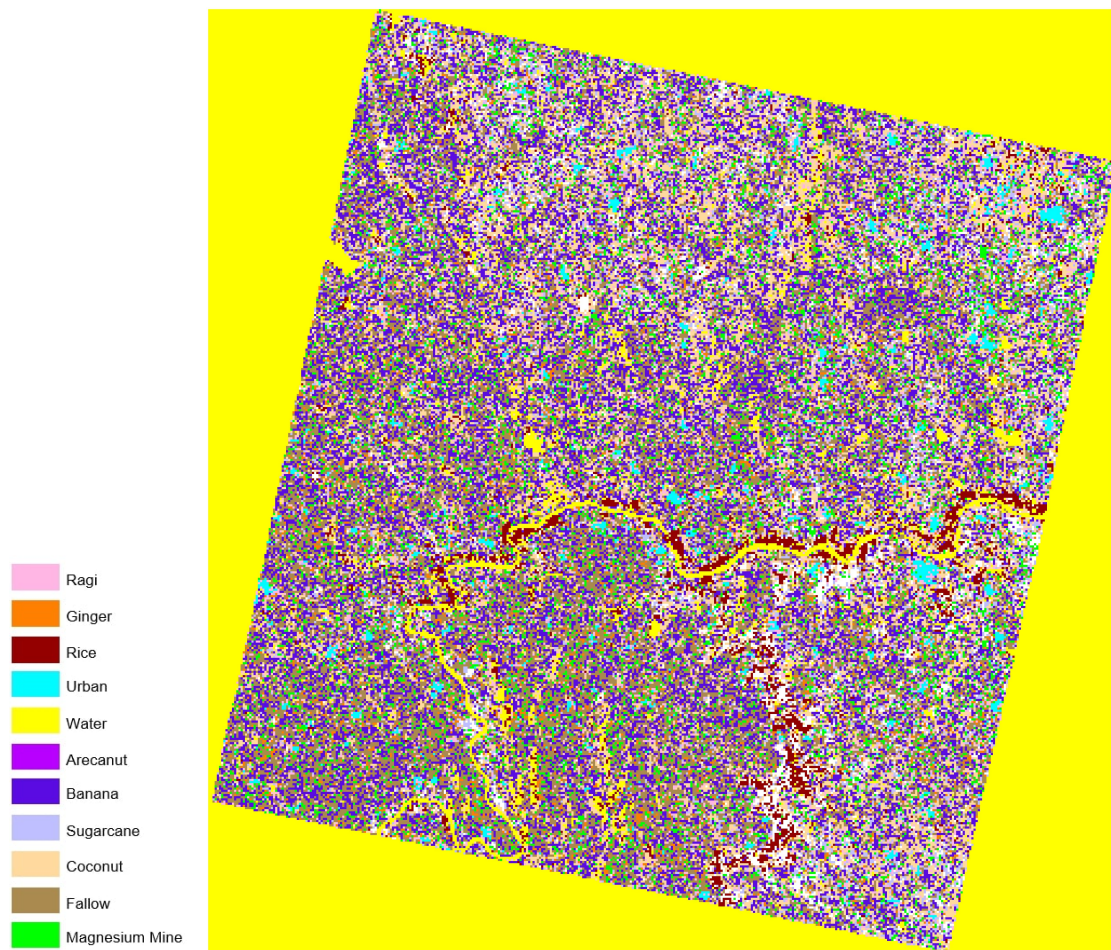


Figure 5.20: Resulting image of Mysore dataset with Data Augmentation using GCT model ( $c=2$ ) with the best performance without skip connections

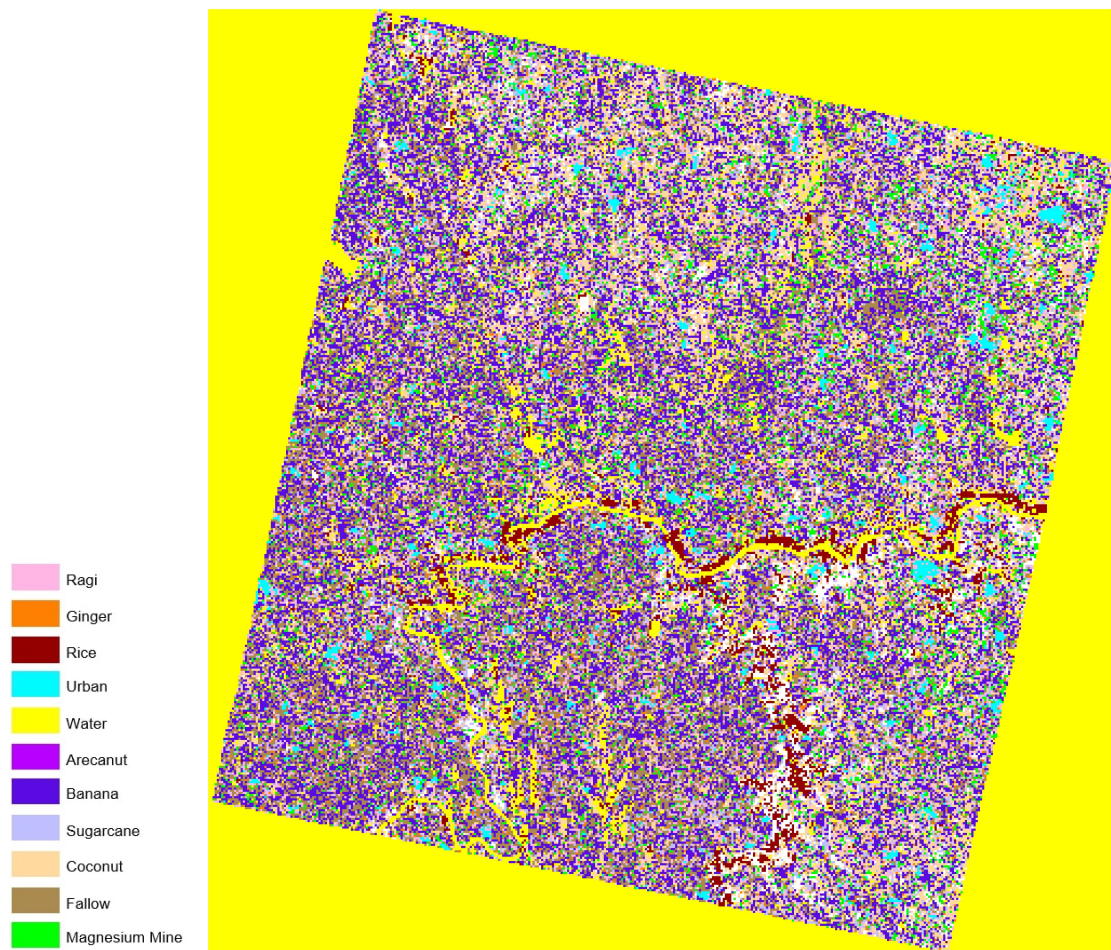


Figure 5.21: Resulting image of Mysore dataset with Data Augmentation using GCT model ( $c=3$ ) with the best performance with skip connections

As can be seen in the figure, the fields are very small in size in comparison to other datasets due to which the patch size was reduced to  $8 \times 8$  for this dataset causing the smaller sample size. However, with data augmentation, a satisfactory accuracy score can be reached with the proposed model.

## CHAPTER 6

# Summary and Analysis

In this chapter, the Cumulative Analysis is taken with consideration to other models such as Wishart Mixture Model and previous deep learning methods without the Gaussian Context Transformer. However, in order to better understand the changes in learning, an Ablation study with changes in  $c$  in Formula 5.7 is also carried out. The value of  $c$  is kept to be an integer in the range of 1 through 4.

### 6.1 Cumulative Analysis

In this section, a cumulative analysis is carried out in order to see if the proposed model is comparable to other models discussed in the Literature Survey and some components of the proposed model with the entirety of it. The main purpose of this section is to compare all the variations of components of the model and also compare it with the Machine Learning method such as WMM Classifier.

Here, in the Flevoland 15 dataset, the overall accuracies show that the variants with the dataset augmentation perform better than their counterparts without data augmentation. This is intuitive because the more the sample size, the better the learning. However, in the models without data augmentation, the ones with the skip connection perform worse than the ones without the skip connections. While this effect is reversed in the variants with the data augmentation. Also, the small class such as buildings are accurately classified in all the deep learning methods except the one without skip connections and without data augmentation however other models, perform better for that class than the WMM Classifier. Following are the accuracies of different models for the Flevoland 15 dataset:

Class Label	Speckled Filtered Data	Unfiltered Data			
	WMM Classifier	Without Skip Connections		With Skip Connections	
	Global K-Means	Without Data Augmentation	With Data Augmentation	Without Data Augmentation	With Data Augmentation
Water	99.21	100.00	99.21	88.14	99.21
Forest	91.95	100.00	100.00	100.00	100.00
Lucerne	96.82	100.00	97.56	90.91	98.78
Grasses	92.50	77.78	98.59	88.24	97.18
Peas	97.61	100.00	100.00	95.45	98.81
Barley	97.18	66.67	100.00	69.57	98.39
BareSoil	96.66	100.00	100.00	57.14	100.00
Beet	94.73	100.00	100.00	97.56	100.00
Wheat2	87.20	100.00	96.74	97.96	96.74
Wheat3	95.45	100.00	99.49	100.00	99.49
Stembeans	97.19	100.00	100.00	95.24	100.00
Rapeseed	87.46	95.83	100.00	97.14	99.09
Wheat	93.58	97.44	99.35	98.18	100.00
Buildings	80.80	0	100.00	100.00	100.00
Potatoes	90.61	100.00	100.00	98.51	100.00
OA	<b>93.72</b>	<b>97.12</b>	<b>99.18</b>	<b>94.39</b>	<b>98.83</b>

Table 6.1: Comparison Results on Flevoland 15 across all models without GCT

Class Label	Unfiltered Data			
	GCT + Without Skip Connections		GCT + With Skip Connections	
	Without Data Augmentation (c=2)	With Data Augmentation (c=2)	Without Data Augmentation (c=2)	With Data Augmentation (c=2)
Water	100.00	100.00	100.00	99.21
Forest	100.00	99.36	100.00	100.00
Lucerne	100.00	100.00	90.00	98.78
Grasses	90.00	98.59	94.44	100.00
Peas	100.00	100.00	95.24	100.00
Barley	100.00	98.39	90.00	98.39
BareSoil	100.00	100.00	100.00	100.00
Beet	95.24	100.00	95.24	100.00
Wheat2	94.74	96.74	91.67	95.65
Wheat3	100.00	98.97	100.00	99.49
Stembeans	100.00	100.00	100.00	100.00
Rapeseed	100.00	100.00	88.89	100.00
Wheat	91.67	99.35	96.43	100.00
Buildings	100.00	100.00	100.00	100.00
Potatoes	100.00	100.00	100.00	100.00
OA	<b>98.92</b>	<b>99.38</b>	<b>96.37</b>	<b>99.45</b>

Table 6.2: Comparison Results on Flevoland 15 across all models with GCT

Following are the image results of the same:



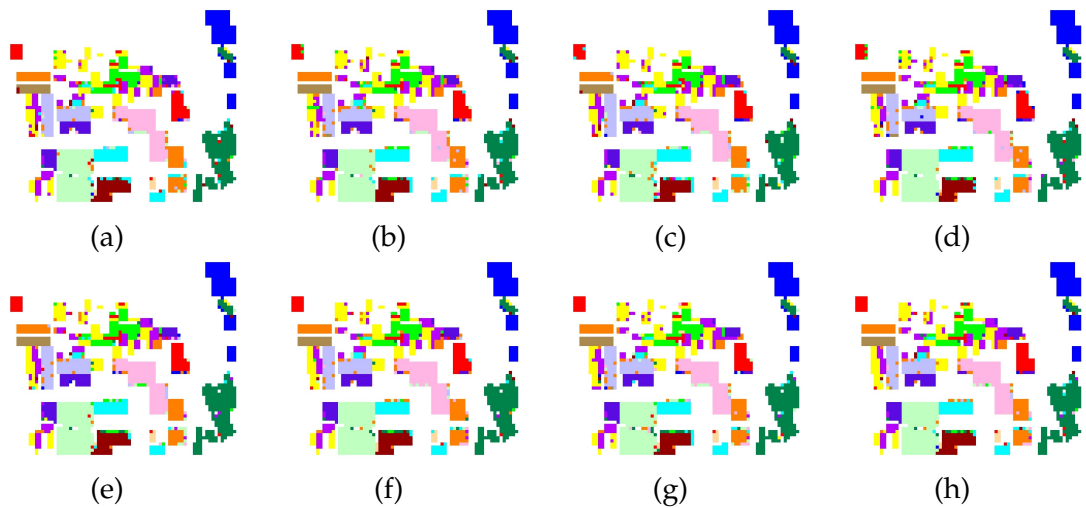


Figure 6.1: Resulting images of all models for Flevoland 15 dataset: (a) SENet (b) SEResNet (c) SENet + GCT (c=2) (d) SEResNet + GCT (c=2) (e) SENet + Data Augmentation (f) SEResNet + Data Augmentation (g) SENet + GCT (c=2) + Data Augmentation (h) SEResNet + GCT (c=2) + Data Augmentation

As can be seen in the image above, the classes such as Grasses and Wheat are better classified in the SEResNet models with the data augmentation rather than the one without. The implementation of the Gaussian Context Transformer can also be seen to be giving an edge to all the variants than the ones without.

For the Flevoland 7 dataset, the results of overall accuracies show that the data augmentation and Gaussian Context Transformer both individually have a better effect on the model rather than the one with their absence. However, the trend with SENet and SEResNet follows the same path of the Residual network performing better with the data augmentation. Following are the accuracies of different models for the Flevoland 7 dataset:

Class Label	Speckled Filtered Data	Unfiltered Data			
	WMM Classifier	Without Skip Connections		With Skip Connections	
	Global K-Means	Without Data Augmentation	With Data Augmentation	Without Data Augmentation	With Data Augmentation
Wheat	98.98	99.12	100.00	98.51	100.00
Rapeseed	99.91	100.00	100.00	100.00	100.00
Barley	99.39	100.00	100.00	98.31	99.29
Lucerne	94.97	85.71	100.00	100.00	100.00
Potatoes	99.15	98.18	100.00	98.85	100.00
Beet	96.99	93.94	97.93	98.18	96.55
Peas	92.87	100.00	100.00	100.00	100.00
OA	98.88	98.48	99.81	98.78	99.75

Table 6.3: Comparison Results on Flevoland 7 across all models without GCT

Class Label	Unfiltered Data			
	GCT + Without Skip Connections		GCT + With Skip Connections	
	Without Data Augmentation (c=2)	With Data Augmentation (c=4)	Without Data Augmentation (c=2)	With Data Augmentation (c=1,3)
	Wheat	99.09	100.00	100.00
Rapeseed	100.00	100.00	100.00	100.00
Barley	100.00	100.00	100.00	100.00
Lucerne	100.00	100.00	100.00	100.00
Potatoes	100.00	100.00	100.00	100.00
Beet	100.00	98.62	96.00	99.31
Peas	100.00	100.00	100.00	100.00
OA	99.69	99.87	99.69	99.94

Table 6.4: Comparison Results on Flevoland 7 across all models with GCT

Following are the image results of the same:

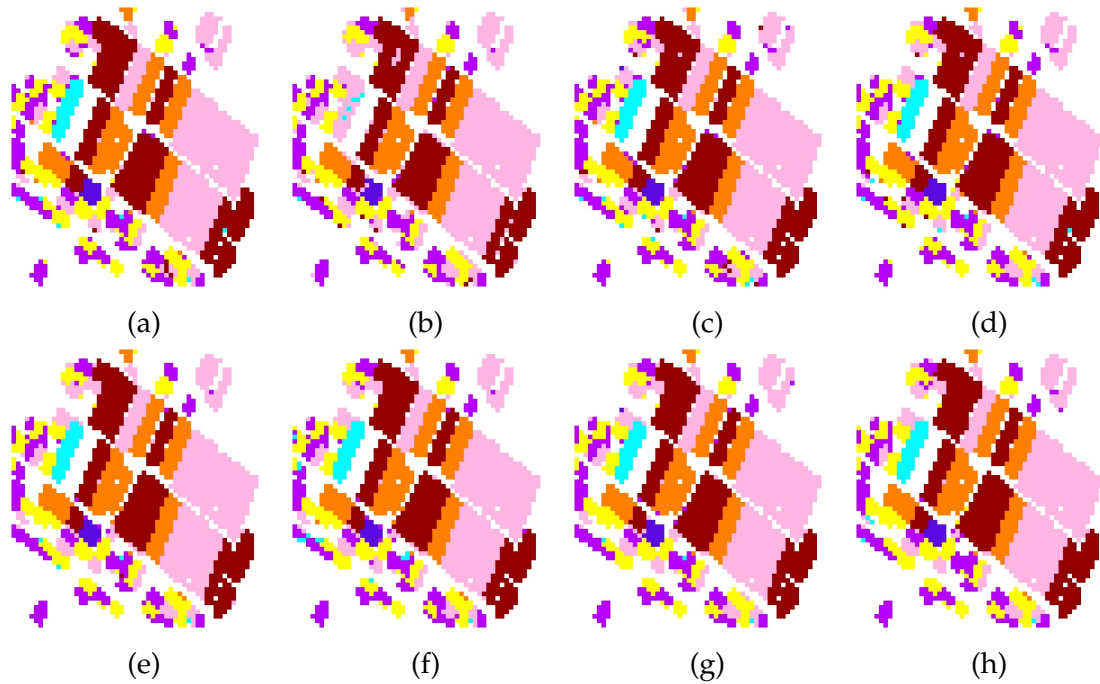


Figure 6.2: Resulting images of all models for Flevoland 7 dataset: (a) SENet (b) SEResNet (c) SENet + GCT (c=2) (d) SEResNet + GCT (c=2) (e) SENet + Data Augmentation (f) SEResNet + Data Augmentation (g) SENet + GCT (c=4) + Data Augmentation (h) SEResNet + GCT (c=3) + Data Augmentation

As can be seen in the figures above, a slightly better change in the classification is observed throughout the variants of models due to the presence of the Gaussian Context Transformer rather than their counterparts. However, data augmentation also seems to help the classes with scarce distribution in the sample.

For the Landes dataset, the trends seem to follow the same path that Gaussian Context Transformer and data augmentation helps individually and collectively

to the proposed model and the other trends seem to be true in this dataset as well. Following are the accuracies of different models for the Landes dataset:

Class Label	Speckled Filtered Data	Unfiltered Data			
	WMM Classifier	Without Skip Connections		With Skip Connections	
	Global K-Means	Without Data Augmentation	With Data Augmentation	Without Data Augmentation	With Data Augmentation
C1	84.41	96.74	98.33	95.74	99.29
C2	79.98	100.00	94.20	93.33	95.65
C3	97.62	100.00	100.00	100.00	100.00
C4	97.31	100.00	100.00	99.65	100.00
C5	95.41	100.00	100.00	96.77	100.00
C6	78.17	90.63	96.39	91.89	96.39
OA	<b>90.73</b>	<b>98.16</b>	<b>98.85</b>	<b>97.24</b>	<b>99.10</b>

Table 6.5: Comparison Results on Landes dataset across all models without GCT

Class Label	Unfiltered Data			
	GCT + Without Skip Connections		GCT + With Skip Connections	
	Without Data Augmentation (c=1)	With Data Augmentation (c=3)	Without Data Augmentation (c=3)	With Data Augmentation (c=2)
C1	98.91	98.81	98.91	98.57
C2	100.00	95.65	92.31	100.00
C3	100.00	100.00	100.00	100.00
C4	100.00	100.00	99.35	100.00
C5	100.00	100.00	100.00	100.00
C6	93.75	96.39	93.75	96.99
OA	<b>99.08</b>	<b>99.10</b>	<b>98.47</b>	<b>99.30</b>

Table 6.6: Comparison Results on Landes dataset across all models with GCT

Following are the image results of the same:

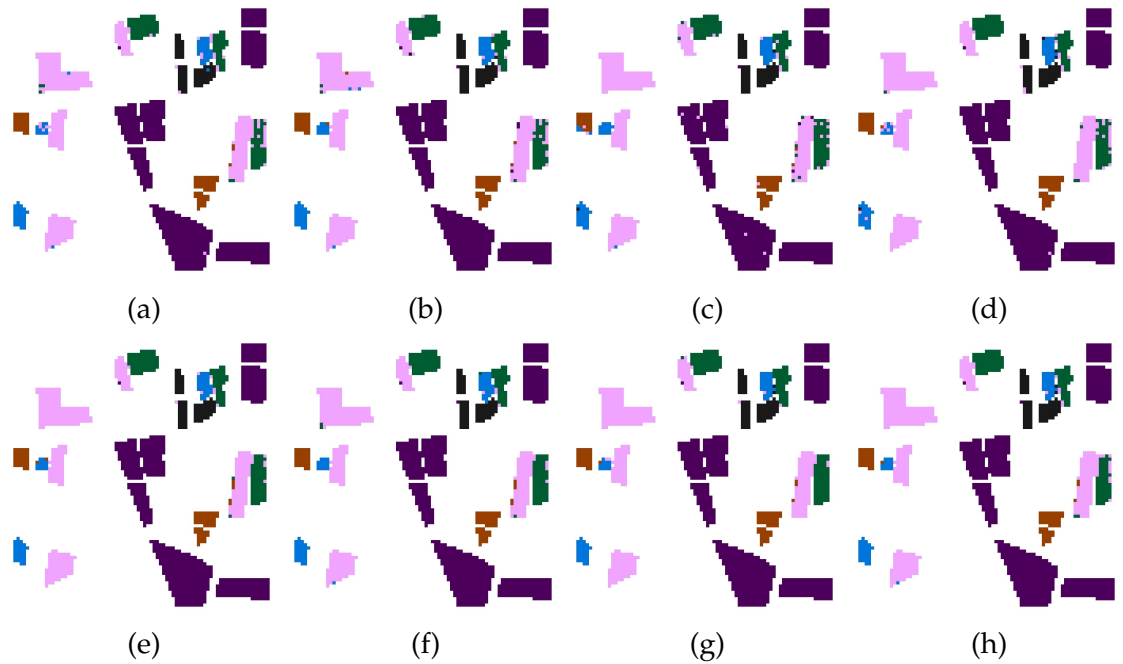


Figure 6.3: Resulting images of all models for Landes dataset: (a) SENet (b) SEResNet (c) SENet + GCT (c=1) (d) SEResNet + GCT (c=3) (e) SENet + Data Augmentation (f) SEResNet + Data Augmentation (g) SENet + GCT (c=3) + Data Augmentation (h) SEResNet + GCT (c=2) + Data Augmentation

As can be seen above, the classes with scarce sample sizes perform better with data augmentation but the performance of almost all classes is enhanced with the Gaussian Context Transformer apart from the slight give or take of accuracies in the classes that are already perfectly classified.

For the Mysore dataset, the Gaussian Context Transformer is the major contributing factor as all the models have data augmentation. The performance enhancement due to Gaussian Context Transformer is also visible in this dataset. Following are the accuracies of different models for the Mysore dataset:

Class Label	Speckled Filtered Data	Unfiltered Data			
	WMM Classifier	Without Skip Connections	With Skip Connections	GCT + Without Skip Connections	GCT + With Skip Connections
	Global	With Data	With Data	With Data	With Data
	K-Means	Augmentation	Augmentation	Augmentation (c=2)	Augmentation (c=2)
Ragi	67.54	100.00	100.00	100.00	100.00
Ginger	93.20	100.00	100.00	100.00	100.00
Rice	95.97	100.00	98.33	100.00	100.00
Urban	99.77	100.00	100.00	100.00	100.00
Water	91.19	98.44	98.44	100.00	98.57
Arecanut	81.42	100.00	100.00	100.00	100.00
Banana	73.13	100.00	100.00	100.00	100.00
Sugarcane	65.83	100.00	100.00	100.00	100.00
Coconut	76.80	100.00	100.00	100.00	100.00
Fallow	78.41	100.00	100.00	100.00	100.00
Magnesite Mine	97.88	72.73	81.82	81.82	100.00
OA	92.72	98.41	98.01	99.20	99.60

Table 6.7: Comparison Results on Mysore dataset across all models

Following are the image results of the same:

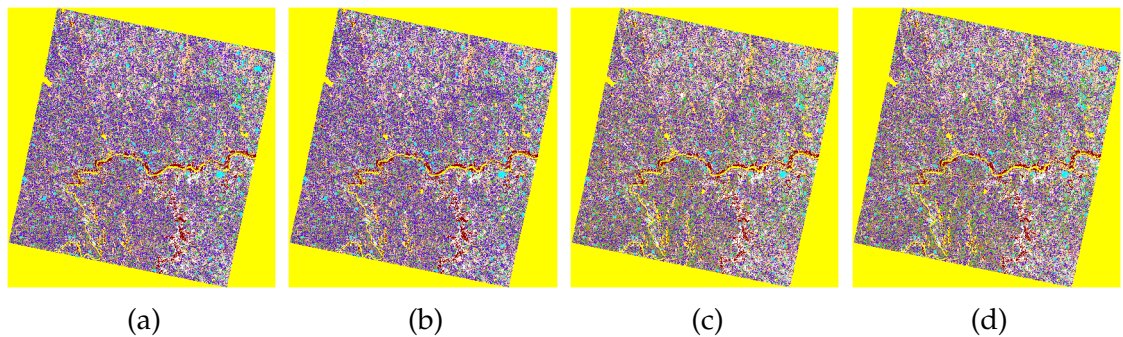


Figure 6.4: Resulting images of all models for Mysore dataset: (a) SENet + Data Augmentation (b) SEResNet + Data Augmentation (c) SENet + GCT (c=2) + Data Augmentation (d) SEResNet + GCT (c=2) + Data Augmentation

As can be seen above in the figure, the images have very small patches scattered across a large plain, and hence the reason for reducing the patch size in the sampling of the data. However, with the data augmentation, the effect of the Gaussian Context Transformer can be seen and it seems to have brought about a positive impact on the overall accuracy of both variants.

## 6.2 Ablation Study

In this section, the effect of changing the standard deviation in the complex normalization part of the Gaussian Context Transformer is observed in different models. This changes the Gaussian distribution's expanse and hence the channels are weighted accordingly in the squeeze and excitation part of the models.

In the Flevoland 15 dataset, the effect of changing the constant  $c$  in the formula 5.7 for the Gaussian Context Transformer seems that almost all of the values of  $c$  perform better than the one without the GCT. However, it is observed that in  $c=2$  and  $c=3$  the overall accuracies are almost always better than the one without GCT but in the other two variants with  $c=1$  and  $c=4$  the case might become slightly worse than the one without the GCT. This shows that the perfect values lie around the range of 2 to 3 with some exceptions.

The following Table shows the effect of changing standard deviation in the Flevoland 15 dataset for the SENet + GCT model:

Class Label	SENet	SENet + GCT			
		c=1	c=2	c=3	c=4
<b>Water</b>	100.00	100.00	100.00	100.00	100.00
<b>Forest</b>	100.00	100.00	100.00	100.00	100.00
<b>Lucerne</b>	100.00	100.00	100.00	95.24	100.00
<b>Grasses</b>	77.78	90.00	90.00	90.00	90.00
<b>Peas</b>	100.00	95.45	100.00	100.00	100.00
<b>Barley</b>	66.67	100.00	100.00	100.00	100.00
<b>BareSoil</b>	100.00	100.00	100.00	100.00	100.00
<b>Beet</b>	100.00	95.24	95.24	95.24	95.24
<b>Wheat2</b>	100.00	89.47	89.47	100.00	94.74
<b>Wheat3</b>	100.00	100.00	100.00	100.00	100.00
<b>Stembeans</b>	100.00	100.00	100.00	100.00	100.00
<b>Rapeseed</b>	95.83	100.00	100.00	100.00	100.00
<b>Wheat</b>	97.44	87.50	95.83	91.67	91.67
<b>Buildings</b>	0.00	100.00	100.00	100.00	100.00
<b>Potatoes</b>	100.00	100.00	100.00	100.00	100.00
<b>OA</b>	<b>97.12</b>	<b>97.12</b>	<b>98.20</b>	<b>98.20</b>	<b>98.20</b>

Table 6.8: Comparison Results on Flevoland 15 across all SENet + GCT variations

Following are the image results for the same:

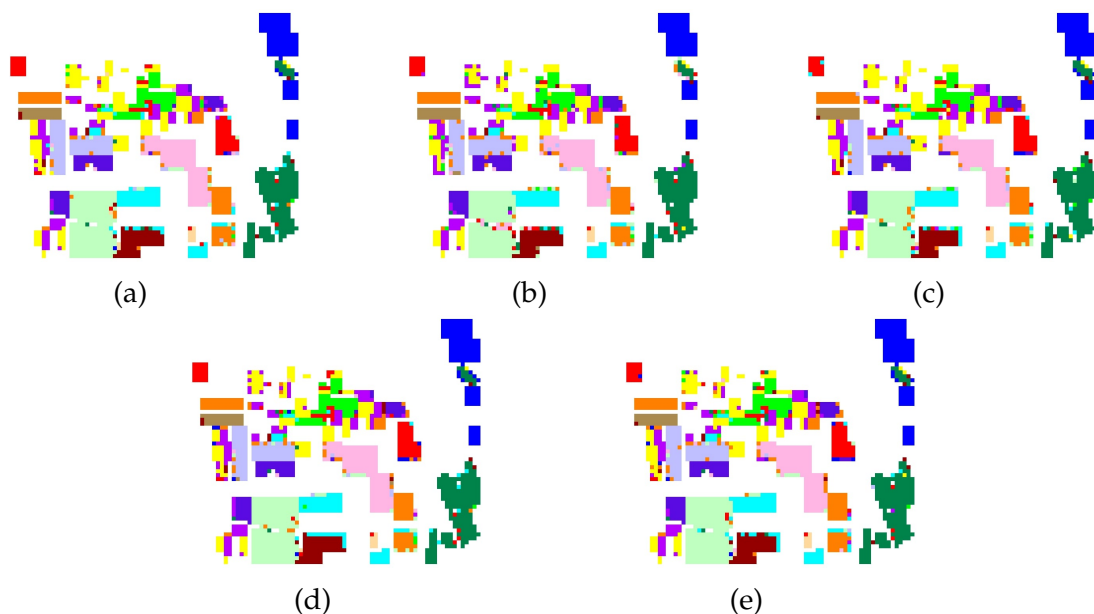


Figure 6.5: Image results on Flevoland 15 dataset of SENet + GCT with variations in standard deviation ( $c$ ) in Gaussian Context Transformer: (a) SENet, (b) SENet + GCT ( $c=1$ ), (c) SENet + GCT ( $c=2$ ), (d) SENet + GCT ( $c=3$ ), (e) SENet + GCT ( $c=4$ )

The following Table shows the effect of changing standard deviation in the Flevoland 15 dataset for the SEResNet + GCT model:

Class Label	SEResNet	SEResNet + GCT			
		$c=1$	$c=2$	$c=3$	$c=4$
<b>Water</b>	100.00	100.00	100.00	100.00	100.00
<b>Forest</b>	100.00	100.00	100.00	100.00	100.00
<b>Lucerne</b>	100.00	100.00	100.00	100.00	100.00
<b>Grasses</b>	100.00	100.00	100.00	100.00	100.00
<b>Peas</b>	94.12	94.12	94.12	94.12	94.12
<b>Barley</b>	100.00	100.00	100.00	100.00	100.00
<b>BareSoil</b>	100.00	100.00	100.00	100.00	100.00
<b>Beet</b>	94.74	94.74	100.00	100.00	100.00
<b>Wheat2</b>	94.74	100.00	94.74	100.00	94.74
<b>Wheat3</b>	95.12	100.00	100.00	92.68	97.56
<b>Stembeans</b>	100.00	100.00	100.00	100.00	100.00
<b>Rapeseed</b>	95.65	95.65	95.65	95.65	91.30
<b>Wheat</b>	96.88	96.88	96.88	93.75	100.00
<b>Buildings</b>	100.00	100.00	100.00	100.00	100.00
<b>Potatoes</b>	100.00	97.00	100.00	100.00	100.00
<b>OA</b>	<b>97.69</b>	<b>98.35</b>	<b>98.68</b>	<b>97.69</b>	<b>98.35</b>

Table 6.9: Comparison Results on Flevoland 15 across all SEResNet + GCT variations

Following are the image results for the same:

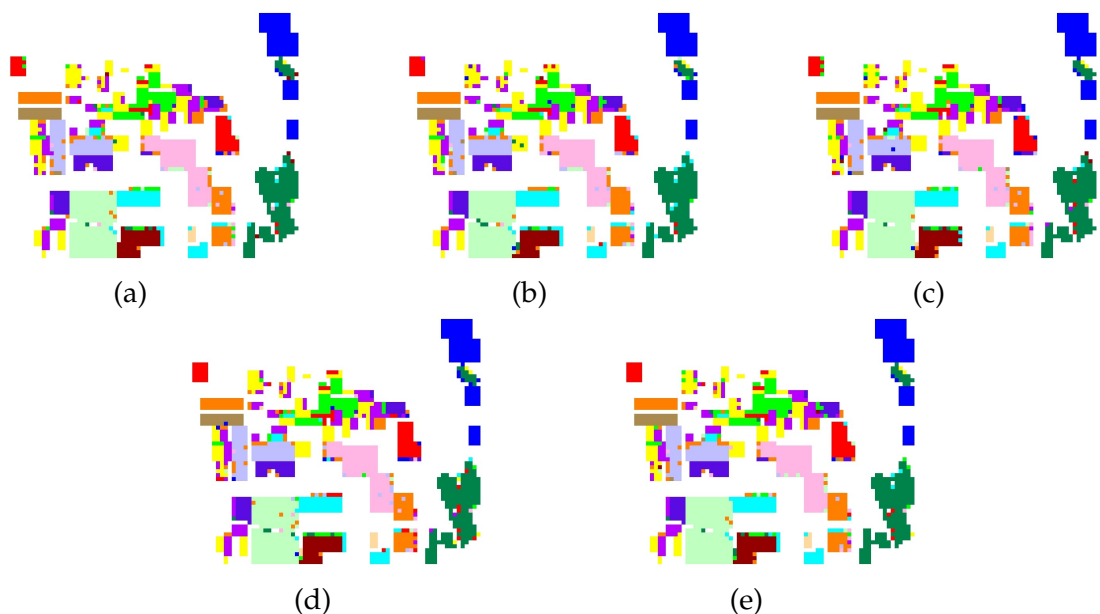


Figure 6.6: Image results on Flevoland 15 dataset of SEResNet + GCT with variations in standard deviation ( $c$ ) in Gaussian Context Transformer: (a) SEResNet, (b) SEResNet + GCT ( $c=1$ ), (c) SEResNet + GCT ( $c=2$ ), (d) SEResNet + GCT ( $c=3$ ), (e) SEResNet + GCT ( $c=4$ )

The following Table shows the effect of changing standard deviation in the Flevoland 15 dataset with Data Augmentation for the SENet + GCT model:

Class Label	SENet	SENet + GCT + Data Augmentation			
		$c=1$	$c=2$	$c=3$	$c=4$
Water	99.21	100.00	99.21	99.21	99.21
Forest	100.00	99.36	100.00	100.00	100.00
Lucerne	97.56	100.00	98.78	97.56	97.56
Grasses	98.59	98.59	97.18	98.59	97.18
Peas	100.00	100.00	100.00	100.00	100.00
Barley	100.00	98.39	98.39	98.39	100.00
BareSoil	100.00	100.00	100.00	100.00	100.00
Beet	100.00	100.00	100.00	100.00	100.00
Wheat2	96.74	96.74	96.74	96.74	96.74
Wheat3	99.49	98.97	99.49	99.49	99.49
Stembeans	100.00	100.00	100.00	98.11	100.00
Rapeseed	100.00	100.00	100.00	100.00	97.27
Wheat	99.35	99.35	99.35	99.35	99.35
Buildings	100.00	100.00	100.00	100.00	100.00
Potatoes	100.00	100.00	99.32	100.00	100.00
<b>OA</b>	<b>99.18</b>	<b>99.38</b>	<b>99.24</b>	<b>99.24</b>	<b>99.11</b>

Table 6.10: Comparison Results on Flevoland 15 dataset with Data Augmentation across all SENet + GCT variations



Following are the image results for the same:

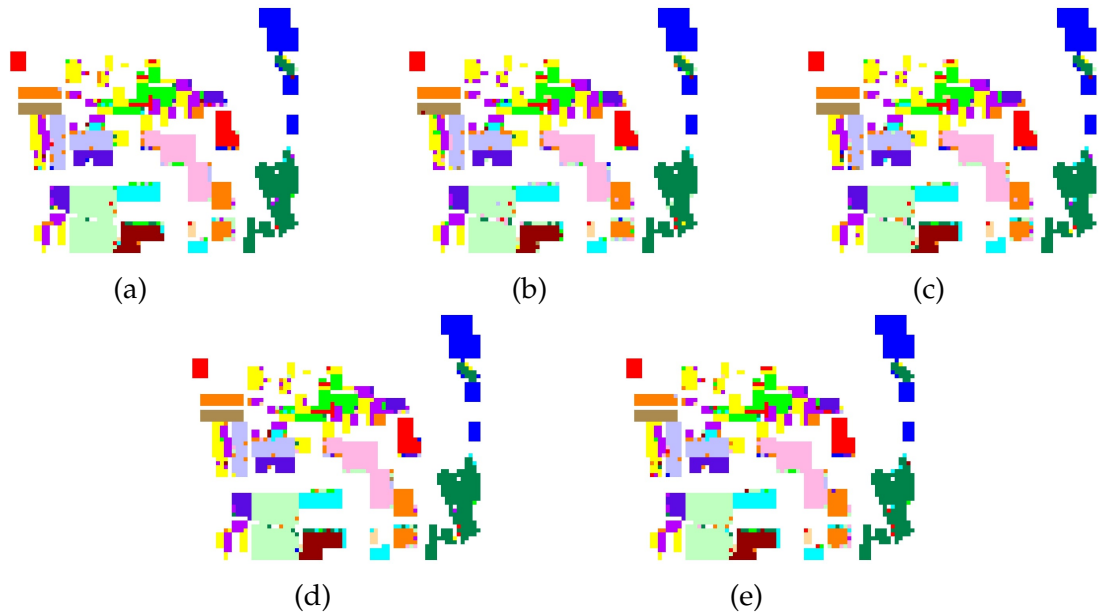


Figure 6.7: Image results on Flevoland 15 dataset with Data Augmentation of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet + Data Augmentation, (b) SENet + GCT (c=1) + Data Augmentation, (c) SENet + GCT (c=2) + Data Augmentation, (d) SENet + GCT (c=3) + Data Augmentation, (e) SENet + GCT (c=4) + Data Augmentation

The following Table shows the effect of changing standard deviation in the Flevoland 15 dataset with Data Augmentation for the SEResNet + GCT model:

Class Label	SEResNet	SEResNet + GCT + Data Augmentation			
		c=1	c=2	c=3	c=4
Water	99.21	99.21	99.21	99.21	99.21
Forest	100.00	100.00	100.00	100.00	100.00
Lucerne	98.78	98.78	98.78	98.78	97.56
Grasses	97.18	98.59	97.18	100.00	98.59
Peas	98.81	100.00	100.00	100.00	98.81
Barley	98.39	100.00	98.39	98.39	98.39
BareSoil	100.00	100.00	100.00	100.00	100.00
Beet	100.00	100.00	100.00	100.00	100.00
Wheat2	96.74	96.74	96.74	95.65	97.83
Wheat3	99.49	98.97	99.49	99.49	99.49
Stembeans	100.00	100.00	100.00	100.00	100.00
Rapeseed	99.09	99.09	100.00	100.00	98.18
Wheat	100.00	100.00	100.00	100.00	100.00
Buildings	100.00	100.00	100.00	100.00	100.00
Potatoes	100.00	100.00	100.00	100.00	100.00
<b>OA</b>	<b>98.83</b>	<b>99.38</b>	<b>99.38</b>	<b>99.45</b>	<b>99.24</b>

Table 6.11: Comparison Results on Flevoland 15 dataset with Data Augmentation across all SEResNet + GCT variations

Following are the image results for the same:

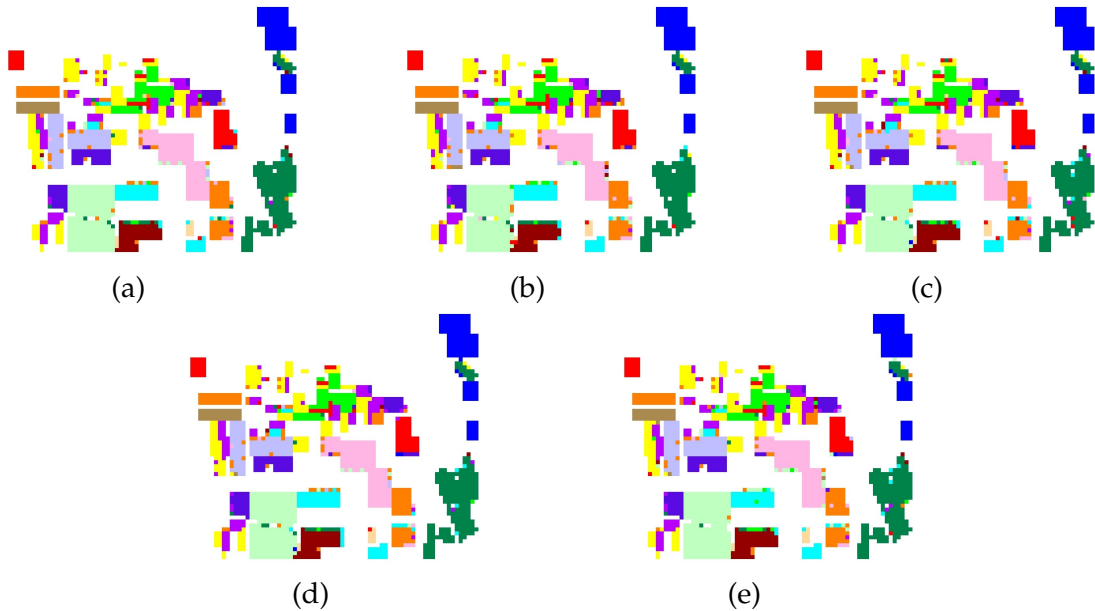


Figure 6.8: Image results on Flevoland 15 dataset with Data Augmentation of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet + Data Augmentation, (b) SEResNet + GCT (c=1) + Data Augmentation, (c) SEResNet + GCT (c=2) + Data Augmentation, (d) SEResNet + GCT (c=3) + Data Augmentation, (e) SEResNet + GCT (c=4) + Data Augmentation

In the Flevoland 7 dataset, the effect of changing the constant  $c$  in the formula 5.7 for the Gaussian Context Transformer seems that almost all of the values of  $c$  perform better than the one without the GCT. However, sometimes the values of some variants of  $c$  are the same which goes to show that the change in the standard deviation stretch of the interchannel dependencies does not change that much after a point of accuracy, since some accuracy values are already high enough. Hence, some anomalies can also be seen in the trends.

The following Table shows the effect of changing standard deviation in the Flevoland 7 dataset for the SENet + GCT model:

Class Label	SENet	SENet + GCT			
		c=1	c=2	c=3	c=4
<b>Wheat</b>	99.12	98.19	99.09	98.19	99.09
<b>Rapeseed</b>	100.00	100.00	100.00	100.00	100.00
<b>Barley</b>	100.00	100.00	100.00	100.00	100.00
<b>Lucerne</b>	85.71	100.00	100.00	100.00	100.00
<b>Potatoes</b>	98.18	100.00	100.00	100.00	100.00
<b>Beet</b>	93.94	93.55	100.00	100.00	93.55
<b>Peas</b>	100.00	100.00	100.00	100.00	100.00
<b>OA</b>	<b>98.48</b>	<b>98.78</b>	<b>99.69</b>	<b>99.39</b>	<b>99.09</b>

Table 6.12: Comparison Results on Flevoland 7 across all SENet + GCT variations

Following are the image results for the same:

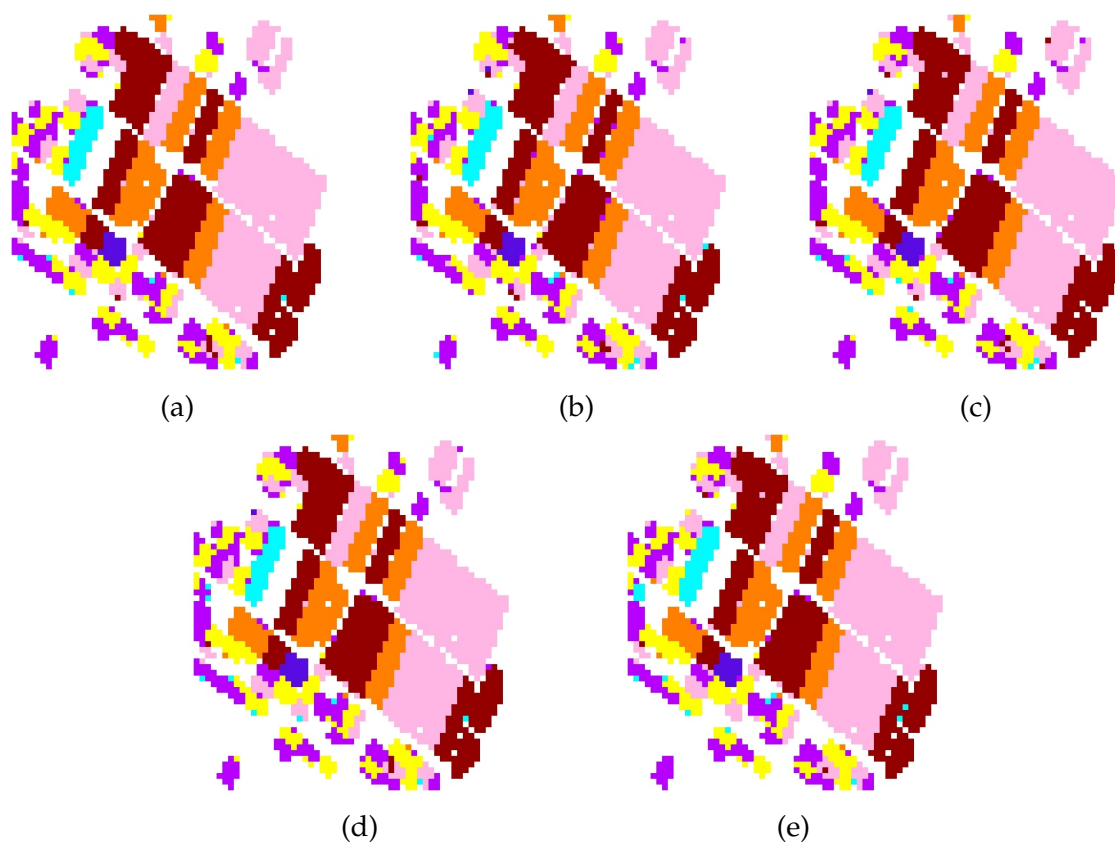


Figure 6.9: Image results on Flevoland 7 dataset of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet, (b) SENet + GCT (c=1), (c) SENet + GCT (c=2), (d) SENet + GCT (c=3), (e) SENet + GCT (c=4)

The following Table shows the effect of changing standard deviation in the Flevoland 7 dataset for the SEResNet + GCT model:

Class Label	SEResNet	SEResNet + GCT			
		c=1	c=2	c=3	c=4
<b>Wheat</b>	98.51	100.00	100.00	99.08	100.00
<b>Rapeseed</b>	100.00	100.00	100.00	100.00	100.00
<b>Barley</b>	98.31	89.69	100.00	100.00	98.84
<b>Lucerne</b>	100.00	100.00	100.00	100.00	100.00
<b>Potatoes</b>	98.85	93.94	100.00	100.00	100.00
<b>Beet</b>	98.18	96.00	96.00	96.00	96.77
<b>Peas</b>	100.00	100.00	100.00	100.00	100.00
<b>OA</b>	<b>98.78</b>	<b>96.05</b>	<b>99.69</b>	<b>99.39</b>	<b>99.39</b>

Table 6.13: Comparison Results on Flevoland 7 across all SEResNet + GCT variations

Following are the image results for the same:

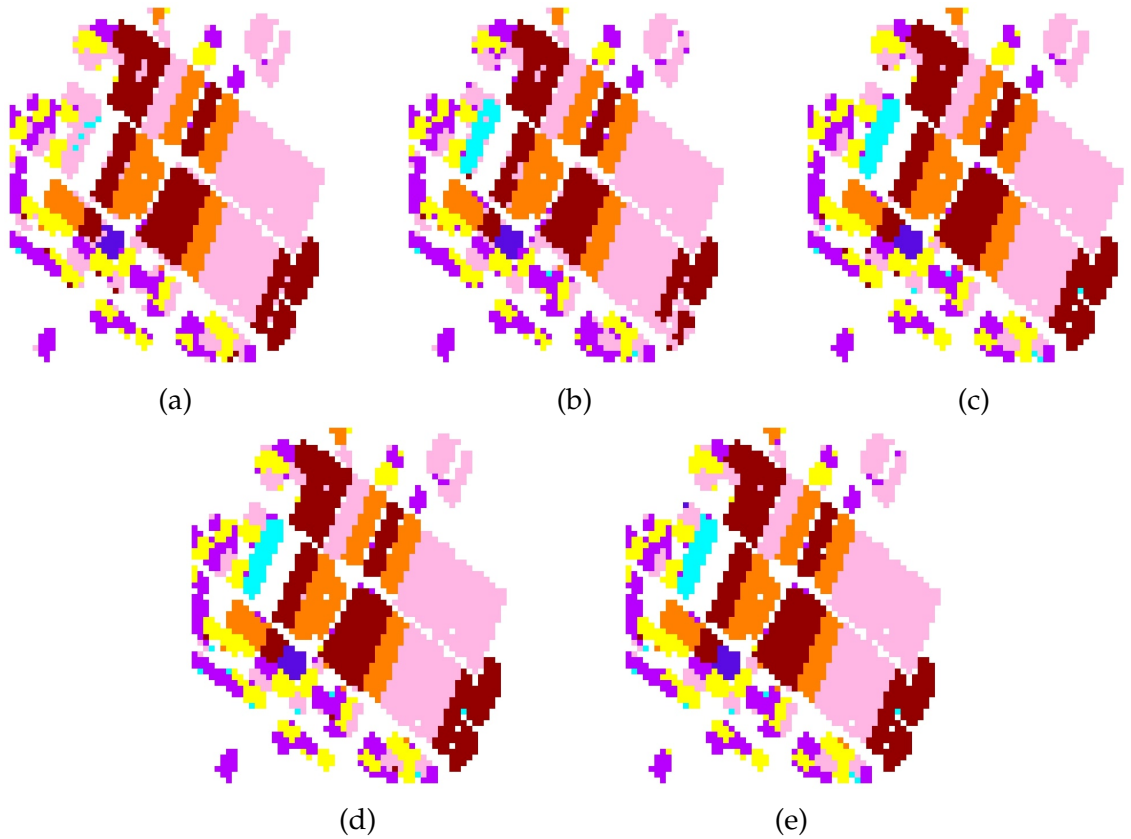


Figure 6.10: Image results on Flevoland 7 dataset of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet, (b) SEResNet + GCT (c=1), (c) SEResNet + GCT (c=2), (d) SEResNet + GCT (c=3), (e) SEResNet + GCT (c=4)

The following Table shows the effect of changing standard deviation in the Flevoland 7 dataset with Data Augmentation for the SENet + GCT model:

Class Label	SENet	SENet + GCT + Data Augmentation			
		c=1	c=2	c=3	c=4
<b>Wheat</b>	100.00	100.00	100.00	100.00	100.00
<b>Rapeseed</b>	100.00	100.00	100.00	100.00	100.00
<b>Barley</b>	100.00	99.76	100.00	100.00	100.00
<b>Lucerne</b>	100.00	100.00	100.00	100.00	100.00
<b>Potatoes</b>	100.00	100.00	100.00	100.00	100.00
<b>Beet</b>	97.93	95.17	97.24	97.93	98.62
<b>Peas</b>	100.00	100.00	100.00	100.00	100.00
<b>OA</b>	<b>99.81</b>	<b>99.49</b>	<b>99.75</b>	<b>99.81</b>	<b>99.87</b>

Table 6.14: Comparison Results on Flevoland 7 dataset with Data Augmentation across all SENet + GCT variations

Following are the image results for the same:

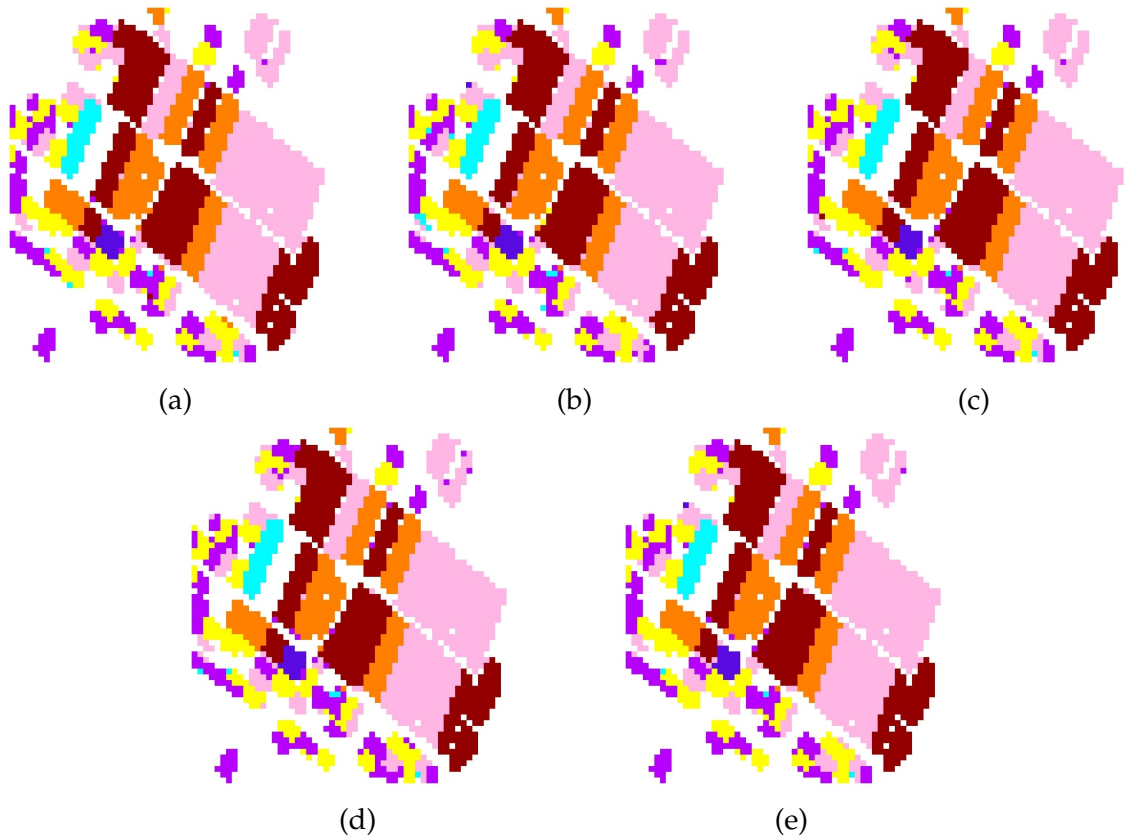


Figure 6.11: Image results on Flevoland 7 dataset with Data Augmentation of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet + Data Augmentation, (b) SENet + GCT (c=1) + Data Augmentation, (c) SENet + GCT (c=2) + Data Augmentation, (d) SENet + GCT (c=3) + Data Augmentation, (e) SENet + GCT (c=4) + Data Augmentation

The following Table shows the effect of changing standard deviation in the Flevoland 7 dataset with Data Augmentation for the SEResNet + GCT model:

Class Label	SEResNet	SEResNet + GCT + Data Augmentation			
		c=1	c=2	c=3	c=4
<b>Wheat</b>	100.00	100.00	99.81	100.00	99.81
<b>Rapeseed</b>	100.00	100.00	100.00	100.00	100.00
<b>Barley</b>	99.29	100.00	100.00	100.00	99.52
<b>Lucerne</b>	100.00	100.00	100.00	100.00	100.00
<b>Potatoes</b>	100.00	100.00	100.00	100.00	100.00
<b>Beet</b>	96.55	99.31	99.31	99.31	97.93
<b>Peas</b>	100.00	100.00	100.00	100.00	100.00
<b>OA</b>	<b>99.75</b>	<b>99.94</b>	<b>99.87</b>	<b>99.94</b>	<b>99.62</b>

Table 6.15: Comparison Results on Flevoland 7 dataset with Data Augmentation across all SEResNet + GCT variations

Following are the image results for the same:

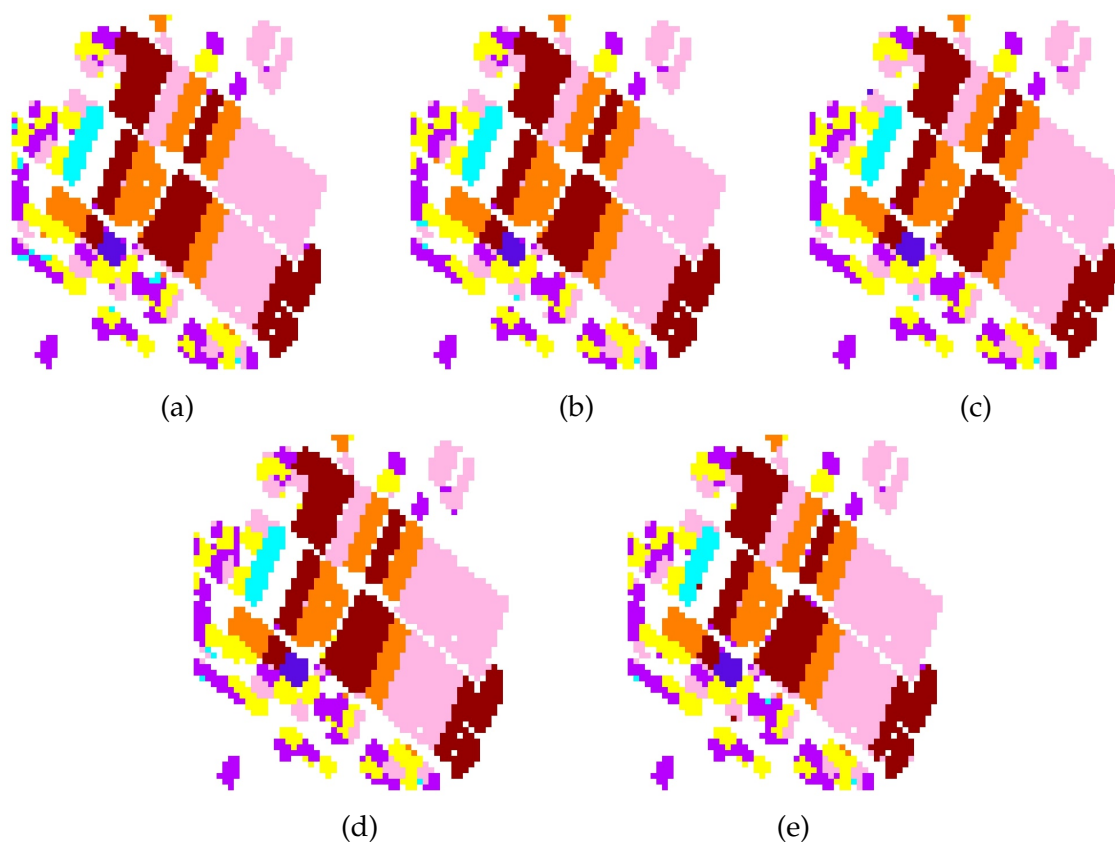


Figure 6.12: Image results on Flevoland 7 dataset with Data Augmentation of SEResNet + GCT with variations in standard deviation ( $c$ ) in Gaussian Context Transformer: (a) SEResNet + Data Augmentation, (b) SEResNet + GCT ( $c=1$ ) + Data Augmentation, (c) SEResNet + GCT ( $c=2$ ) + Data Augmentation, (d) SEResNet + GCT ( $c=3$ ) + Data Augmentation, (e) SEResNet + GCT ( $c=4$ ) + Data Augmentation

For the Landes dataset, the same trend follows as that for the Flevoland 15 dataset, however, sometimes the values of accuracies turn out to be the same due to the possibility of overfitting in some cases since the classes are very large compared to the other two datasets yet the number of classes are less.

The following Table shows the effect of changing standard deviation in the Landes dataset for the SENet + GCT model:

Class Label	SENet	SENet + GCT			
		c=1	c=2	c=3	c=4
C1	96.74	98.91	98.91	100.00	100.00
C2	100.00	100.00	100.00	76.92	92.31
C3	100.00	100.00	100.00	100.00	100.00
C4	100.00	100.00	99.35	99.35	99.35
C5	100.00	100.00	100.00	100.00	100.00
C6	90.63	93.75	87.50	93.75	96.88
OA	<b>98.16</b>	<b>99.08</b>	<b>98.16</b>	<b>98.16</b>	<b>99.08</b>

Table 6.16: Comparison Results on Landes across all SENet + GCT variations

Following are the image results for the same:

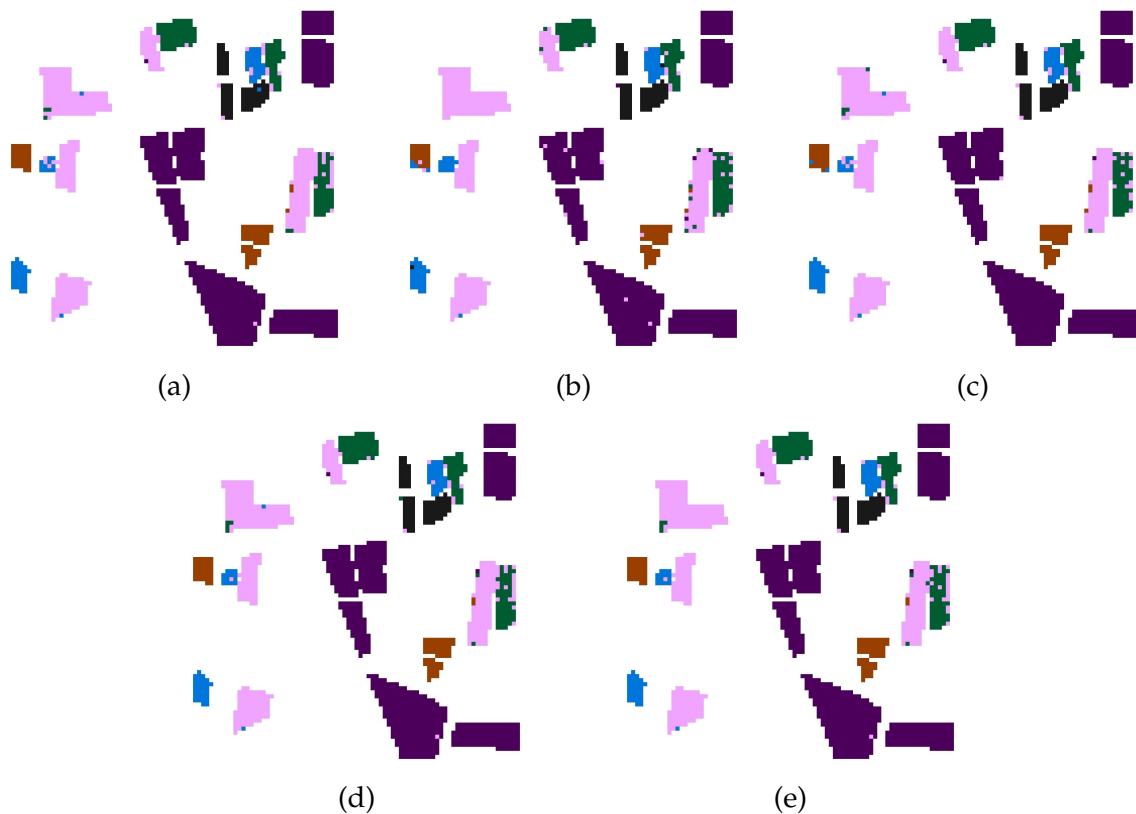


Figure 6.13: Image results on Landes dataset of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet, (b) SENet + GCT (c=1), (c) SENet + GCT (c=2), (d) SENet + GCT (c=3), (e) SENet + GCT (c=4)

The following Table shows the effect of changing standard deviation in the Landes dataset for the SEResNet + GCT model:



Class Label	SEResNet	SEResNet + GCT			
		c=1	c=2	c=3	c=4
C1	95.74	95.65	96.74	98.91	98.91
C2	93.33	100.00	84.62	92.31	92.31
C3	100.00	84.21	94.74	100.00	100.00
C4	99.65	97.39	99.35	99.35	99.35
C5	96.77	100.00	100.00	100.00	100.00
C6	91.89	90.63	96.88	93.75	93.75
OA	<b>97.24</b>	<b>95.71</b>	<b>97.55</b>	<b>98.47</b>	<b>98.47</b>

Table 6.17: Comparison Results on Landes across all SEResNet + GCT variations

Following are the image results for the same:

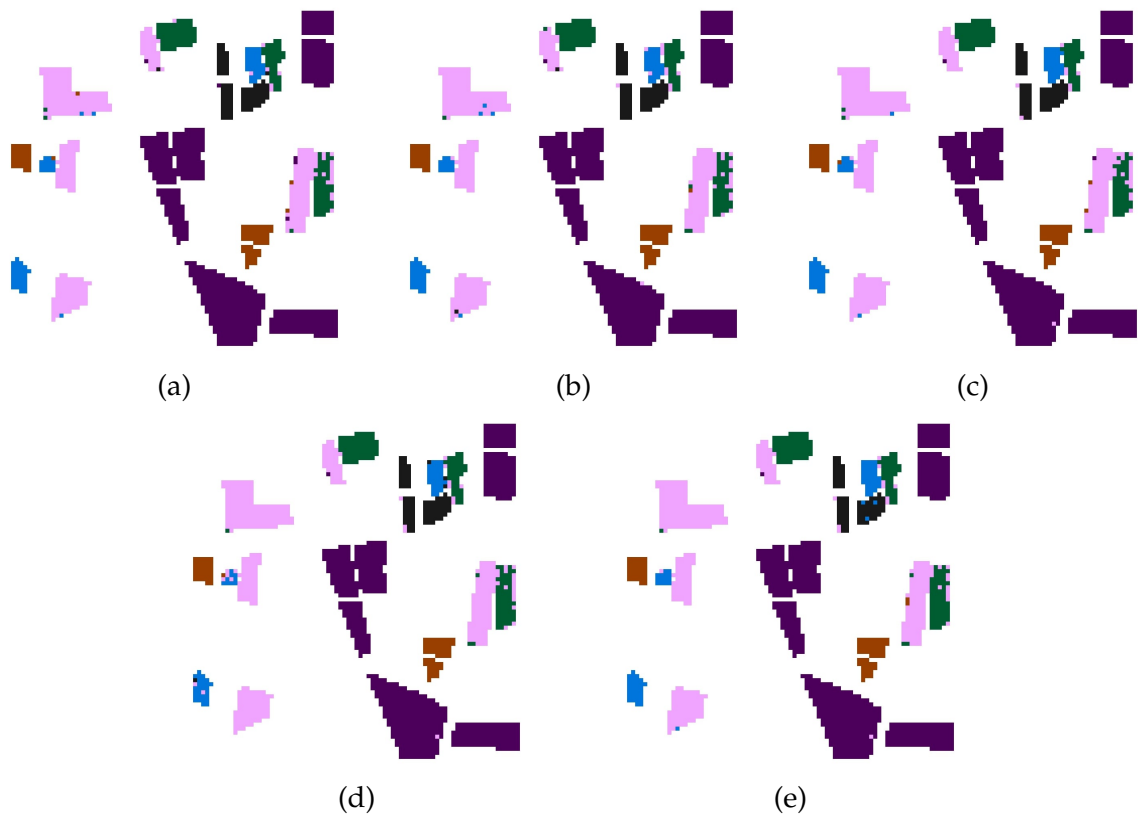


Figure 6.14: Image results on Landes dataset of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet, (b) SEResNet + GCT (c=1), (c) SEResNet + GCT (c=2), (d) SEResNet + GCT (c=3), (e) SEResNet + GCT (c=4)

The following Table shows the effect of changing standard deviation in the Landes dataset with Data Augmentation for the SENet + GCT model:

Class Label	SENet	SENet + GCT + Data Augmentation			
		c=1	c=2	c=3	c=4
C1	98.33	98.33	97.86	98.81	98.33
C2	94.20	94.20	95.65	95.65	97.10
C3	100.00	100.00	100.00	100.00	100.00
C4	100.00	99.87	99.87	100.00	100.00
C5	100.00	100.00	100.00	100.00	100.00
C6	96.39	97.59	95.78	96.39	92.77
OA	<b>98.85</b>	<b>98.15</b>	<b>98.72</b>	<b>99.10</b>	<b>98.66</b>

Table 6.18: Comparison Results on Landes dataset with Data Augmentation across all SENet + GCT variations

Following are the image results for the same:

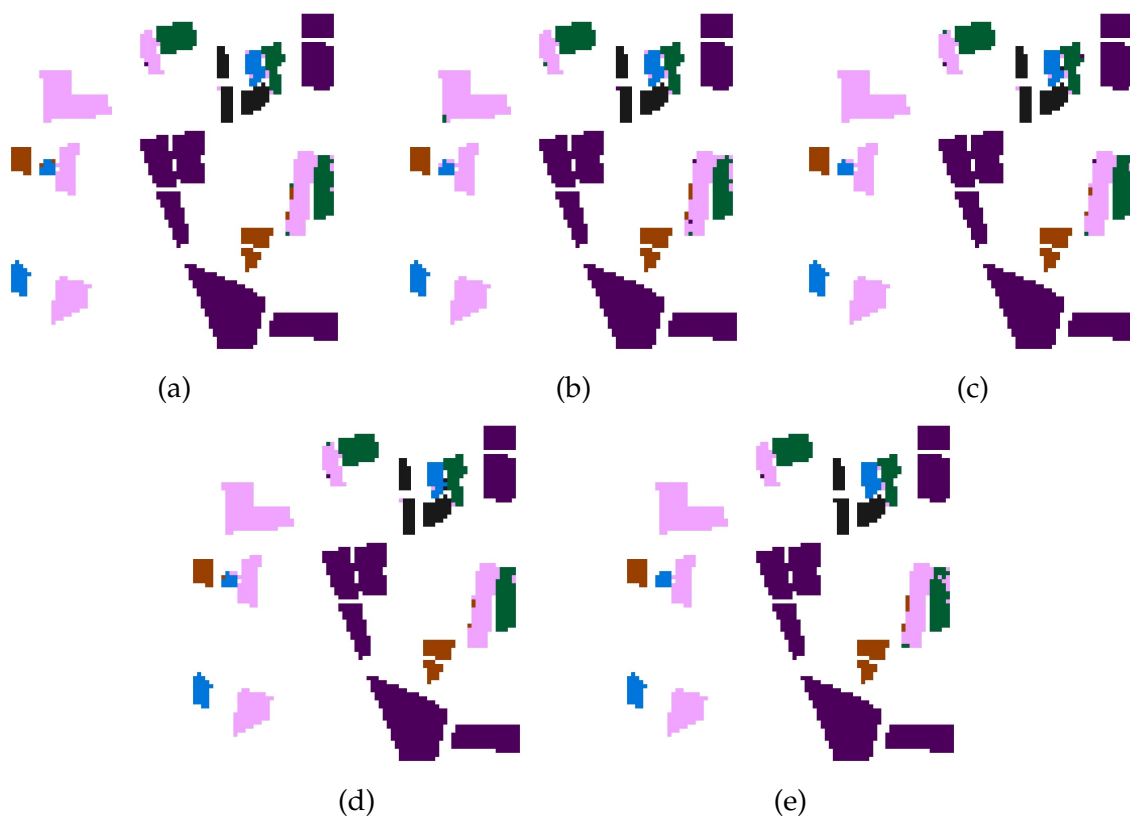


Figure 6.15: Image results on Landes dataset with Data Augmentation of SENet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SENet + Data Augmentation, (b) SENet + GCT (c=1) + Data Augmentation, (c) SENet + GCT (c=2) + Data Augmentation, (d) SENet + GCT (c=3) + Data Augmentation, (e) SENet + GCT (c=4) + Data Augmentation

The following Table shows the effect of changing standard deviation in the Landes dataset with Data Augmentation for the SEResNet + GCT model:

Class Label	SEResNet	SEResNet + GCT + Data Augmentation			
		c=1	c=2	c=3	c=4
C1	99.29	97.38	98.57	99.05	99.29
C2	95.65	94.20	100.00	95.65	98.55
C3	100.00	100.00	100.00	100.00	100.00
C4	100.00	100.00	100.00	99.87	100.00
C5	100.00	100.00	100.00	100.00	100.00
C6	96.39	98.19	96.99	96.99	94.58
OA	<b>99.10</b>	<b>98.85</b>	<b>99.30</b>	<b>99.17</b>	<b>99.17</b>

Table 6.19: Comparison Results on Landes dataset with Data Augmentation across all SEResNet + GCT variations

Following are the image results for the same:

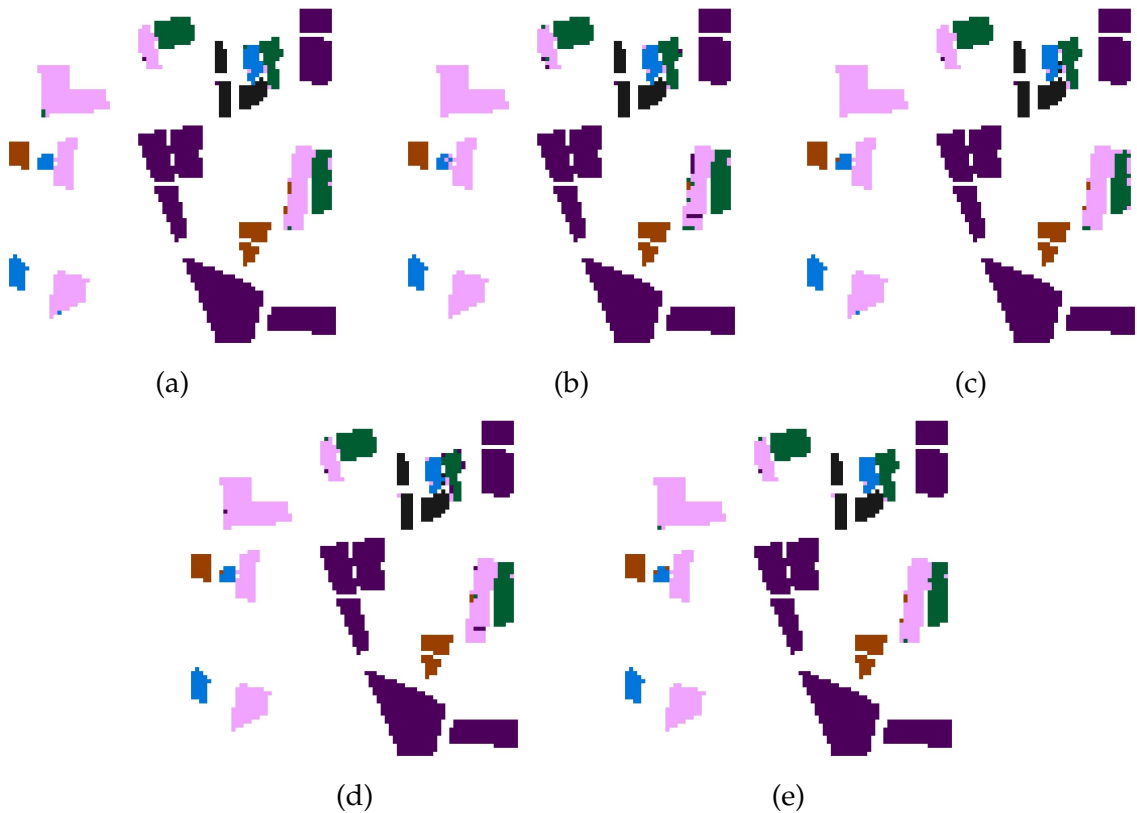


Figure 6.16: Image results on Landes dataset with Data Augmentation of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet + Data Augmentation, (b) SEResNet + GCT (c=1) + Data Augmentation, (c) SEResNet + GCT (c=2) + Data Augmentation, (d) SEResNet + GCT (c=3) + Data Augmentation, (e) SEResNet + GCT (c=4) + Data Augmentation

For the Mysore dataset, the case is different since the patch size is smaller. Here, the accuracies across the variants of c are similar as they reach high enough values. This may show that once the accuracy of a model reaches high enough

values the contribution of variation in  $c$  decreases.

The following Table shows the effect of changing standard deviation in the Mysore dataset with Data Augmentation for the SENet + GCT model:

Class Label	SENet	SENet + GCT + Data Augmentation			
		c=1	c=2	c=3	c=4
<b>Ragi</b>	100.00	100.00	100.00	100.00	100.00
<b>Ginger</b>	100.00	100.00	100.00	100.00	100.00
<b>Rice</b>	100.00	100.00	100.00	100.00	100.00
<b>Urban</b>	100.00	100.00	100.00	100.00	100.00
<b>Water</b>	98.44	96.88	100.00	96.88	98.44
<b>Arecanut</b>	100.00	100.00	100.00	100.00	100.00
<b>Banana</b>	100.00	100.00	100.00	100.00	100.00
<b>Sugarcane</b>	100.00	100.00	100.00	100.00	100.00
<b>Coconut</b>	100.00	100.00	100.00	100.00	100.00
<b>Fallow</b>	100.00	100.00	100.00	100.00	100.00
<b>Magnesite Mine</b>	72.73	81.82	81.82	81.82	81.82
<b>OA</b>	<b>98.41</b>	<b>98.41</b>	<b>99.20</b>	<b>98.41</b>	<b>98.80</b>

Table 6.20: Comparison Results on Mysore dataset with Data Augmentation across all SENet + GCT variations

Following are the image results for the same:

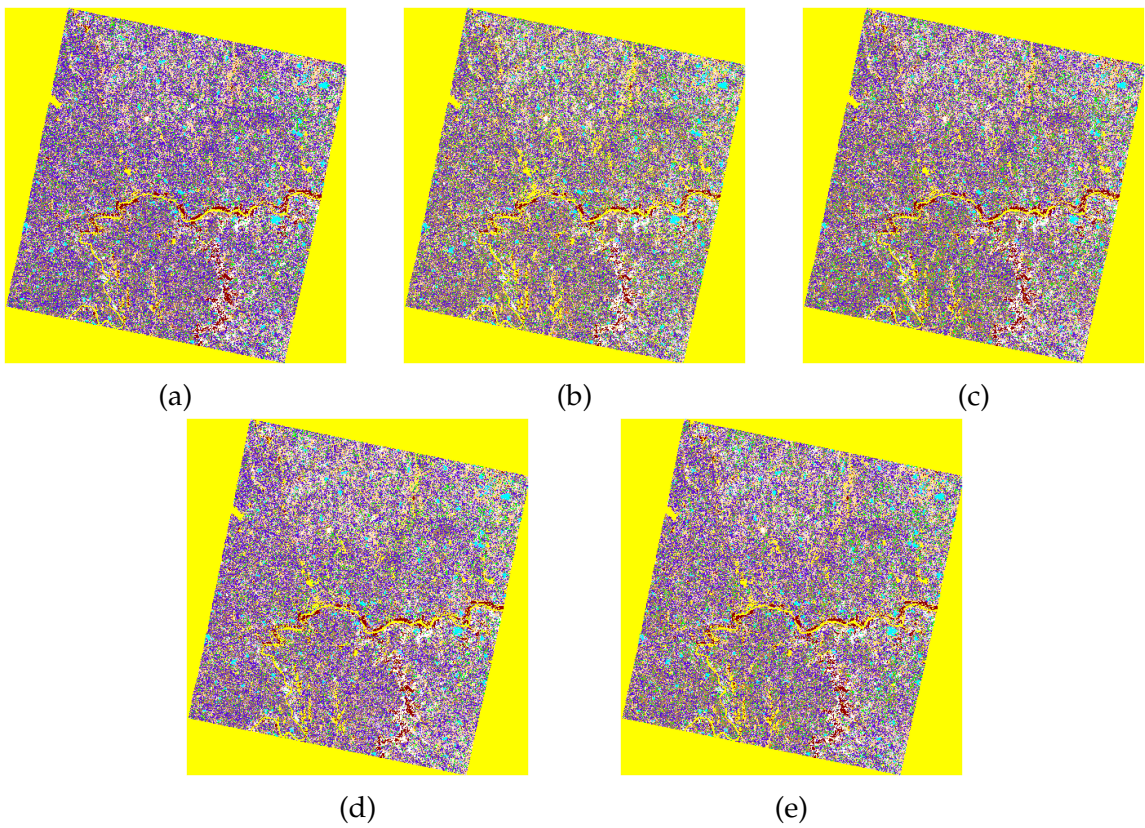


Figure 6.17: Image results on Mysore dataset with Data Augmentation of SENet + GCT with variations in standard deviation ( $c$ ) in Gaussian Context Transformer: (a) SENet + Data Augmentation, (b) SENet + GCT ( $c=1$ ) + Data Augmentation, (c) SENet + GCT ( $c=2$ ) + Data Augmentation, (d) SENet + GCT ( $c=3$ ) + Data Augmentation, (e) SENet + GCT ( $c=4$ ) + Data Augmentation

The following Table shows the effect of changing standard deviation in the Mysore dataset with Data Augmentation for the SEResNet + GCT model:

Class Label	SEResNet	SEResNet + GCT + Data Augmentation			
		c=1	c=2	c=3	c=4
<b>Ragi</b>	100.00	100.00	100.00	100.00	100.00
<b>Ginger</b>	100.00	100.00	100.00	100.00	100.00
<b>Rice</b>	98.33	100.00	100.00	100.00	100.00
<b>Urban</b>	100.00	100.00	100.00	100.00	100.00
<b>Water</b>	98.44	98.57	98.57	98.57	98.57
<b>Arecanut</b>	100.00	100.00	100.00	100.00	100.00
<b>Banana</b>	100.00	100.00	100.00	100.00	100.00
<b>Sugarcane</b>	100.00	100.00	100.00	100.00	100.00
<b>Coconut</b>	100.00	100.00	100.00	100.00	100.00
<b>Fallow</b>	100.00	100.00	100.00	100.00	100.00
<b>Magnesite Mine</b>	81.82	100.00	100.00	100.00	100.00
<b>OA</b>	<b>98.01</b>	<b>99.60</b>	<b>99.60</b>	<b>99.60</b>	<b>99.60</b>

Table 6.21: Comparison Results on Mysore dataset with Data Augmentation across all SEResNet + GCT variations

Following are the image results for the same:

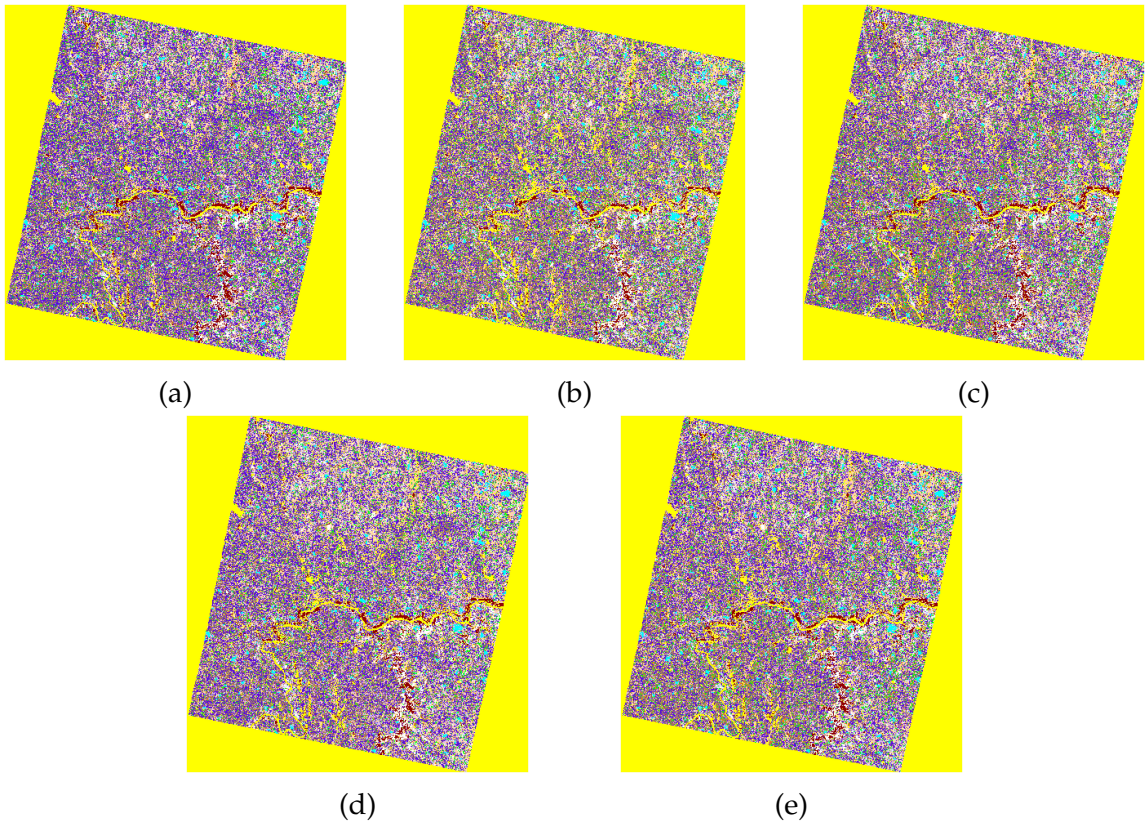


Figure 6.18: Image results on Mysore dataset with Data Augmentation of SEResNet + GCT with variations in standard deviation (c) in Gaussian Context Transformer: (a) SEResNet + Data Augmentation, (b) SEResNet + GCT (c=1) + Data Augmentation, (c) SEResNet + GCT (c=2) + Data Augmentation, (d) SEResNet + GCT (c=3) + Data Augmentation, (e) SEResNet + GCT (c=4) + Data Augmentation

### 6.3 Comparison with State of the art Models

Following is the comparison of the current proposed model with the state-of-the-art models that are published before this work. Following is the table which shows the comparison of these models in the Flevoland 15 dataset:

State of the Art Models	Flevoland 15(OA)
W-DBN	97.57
SPCNN	96.90
CVPDAS-CNN	98.32
CVMS-CNN	97.74
<b>Proposed Model (SEResNet+GCT+Data Augmentation)</b>	<b>99.45</b>

Table 6.22: Comparison Results on Flevoland 15 dataset across the different state of the art models

The recent state-of-the-art models include Wishart Deep Belief Network (W-DBN)[21], Self-Paced Convolutional Neural Network (SPCNN) [22], Complex-Valued PDAS (CVPDAS-CNN) [23], and Complex-Valued Multi-Scale CNN (CVMS-CNN) [24]. The table above shows that the Proposed model performs better than the other state-of-the-art models. In W-DBN, a cleaning algorithm is proposed based on the confusion matrix and local spatial information. While others use CNN in order to leverage the spatial information in order to classify the data. The proposed model, however, beats those accuracies because there is mathematical normalization of inter-channel dependencies that are leveraged in this case, giving it an edge in classification.



## CHAPTER 7

# Conclusion

The introduction of the Gaussian Context Transformer in the Squeeze and Excitation block does improve the accuracy of all the pre-existing models. This suggests that the deep learning part itself can learn better if some channel attenuations are learned with statistical means. However, the Gaussian Context Transformer does not completely interfere with the learning parameters; it only weighs them accordingly with the Gaussian distribution of specified standard deviations and provides channel attention coefficients. This process further enhances the model training which can be seen in the results across all models with the addition of GCT. The Data Augmentation part yields better results due to the addition of the training data, however, the image processing techniques chosen also play a small role in the quality of training. The speckle reduction techniques used are Box Car filter, Lee Sigma Filter, and Mean Shift filter.

The SEResNet models across all datasets performed worse than SENet models, however, with the introduction of Gaussian Context Transformers, the SEResNet blocks starts to learn better than their SENet counterparts without loss of performance in the SENet models. This happens due to the generalization of learning polynomial weights due to channel attention normalization during the Squeeze and Excitation block. This removes some hampering factors in the better learning obtained by the Residual blocks.

Thus, the main goal of the Gaussian Context Transformer is to remove the learning component of the channel attention mechanism and introduce hard-set mathematical learning for channel attention weights in the SENet part of any model. This signifies that the channel weights can be set mathematically rather than learned through a neural network if the data's overall distribution's inclination is known.

## CHAPTER 8

# Future Work

Moreover, this Thesis contributes towards balancing both statistical learning methods as well as deep learning methods into a single model. However, in the future, two separate models can be balanced in the classification of PolSAR images.

1. In the future, the standard deviation parameter which is decided manually can also be left to the neural network by the introduction of a sigmoid activation layer.
2. Ablation study on various data augmentations can be done to see which image processing technique contributes more towards better learning for PolSAR data.
3. Current study is performed only on L-band data. In the future, when dual-band data is available, these models can be tested on them in order to see how they fare in those conditions.

## References

- [1] NASA Earth Science Data Systems. What is synthetic aperture radar? <https://www.earthdata.nasa.gov/learn/backgrounders/what-is-sar>, Apr 2020.
- [2] Jong-Sen Lee and Eric Pottier. *Polarimetric radar imaging: from basics to applications*. CRC press, 2017.
- [3] Shradha Makhija. *PolSAR Image Classification using Deep Learning*. PhD thesis, 2022.
- [4] Zhimian Zhang, Haipeng Wang, Feng Xu, and Ya-Qiu 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.
- [5] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [6] Dongsheng Ruan, Daiyin Wang, Yuan Zheng, Nenggan Zheng, and Min Zheng. Gaussian context transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15129–15138, 2021.
- [7] Nilam Chaudhari, Suman K Mitra, Sanid Chirakkal, Srimanta Mandal, Deepak Putrevu, and Arundhati Misra. Discrimination of multi-crop scenarios with polarimetric sar data using wishart mixture model. *Journal of Applied Remote Sensing*, 15(3):034514, 2021.
- [8] Nilam Chaudhari, Suman K Mitra, Srimanta Mandal, Sanid Chirakkal, Deepak Putrevu, and Arundhati Misra. Edge-preserving classification of polarimetric sar images using wishart distribution and conditional random field. *International Journal of Remote Sensing*, 43(6):2134–2155, 2022.
- [9] Nataliia Kussul, Sergii Skakun, Andrii Shelestov, and Olga Kussul. The use of satellite sar imagery to crop classification in ukraine within jecam project. In

- 2014 *IEEE Geoscience and Remote Sensing Symposium*, pages 1497–1500. IEEE, 2014.
- [10] Gustavo Camps-Valls and Lorenzo Bruzzone. Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 43(6):1351–1362, 2005.
- [11] Anthony Freeman and Stephen L Durden. A three-component scattering model for polarimetric sar data. *IEEE transactions on geoscience and remote sensing*, 36(3):963–973, 1998.
- [12] Knut Conradsen, Allan Aasbjerg Nielsen, Jesper Schou, and Henning Skriver. A test statistic in the complex wishart distribution and its application to change detection in polarimetric sar data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(1):4–19, 2003.
- [13] JJ Van Zyl and CF Burnette. Bayesian classification of polarimetric sar images using adaptive a priori probabilities. *International Journal of Remote Sensing*, 13(5):835–840, 1992.
- [14] Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017.
- [15] Davide Cozzolino, Gerardo Di Martino, Giovanni Poggi, and Luisa 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. IEEE, 2017.
- [16] Simon 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.
- [17] Sizhe Chen, Haipeng Wang, Feng Xu, and Ya-Qiu Jin. Target classification using the deep convolutional networks for sar images. *IEEE transactions on geoscience and remote sensing*, 54(8):4806–4817, 2016.
- [18] Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo, and Junwei Han. When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns. *IEEE transactions on geoscience and remote sensing*, 56(5):2811–2821, 2018.

- [19] Mingrui Zhang, Zongyong Cui, Xianyuan Wang, and Zongjie Cao. Data augmentation method of sar image dataset. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 5292–5295. IEEE, 2018.
- [20] Jun Ding, Bo Chen, Hongwei Liu, and Mengyuan Huang. Convolutional neural network with data augmentation for sar target recognition. *IEEE Geoscience and remote sensing letters*, 13(3):364–368, 2016.
- [21] Fang Liu, Licheng Jiao, Biao Hou, and Shuyuan Yang. Pol-sar image classification based on wishart dbn and local spatial information. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6):3292–3308, 2016.
- [22] Changzhe Jiao, Xinlin Wang, Shuiping Gou, Wenshuai Chen, Debo Li, Chao Chen, and Xiaofeng Li. Self-paced convolutional neural network for polsar images classification. *Remote Sensing*, 11(4):424, 2019.
- [23] Hongwei Dong, Bin Zou, Lamei Zhang, and Siyu 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.
- [24] Lamei Zhang, Siyu Zhang, Hongwei Dong, and Da Lu. Polsar image classification via complex-valued multi-scale convolutional neural network. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 200–203. IEEE, 2020.
- [25] Polsarpro v 6.0 biomass edition. <https://step.esa.int/main/download/polsarpro-v6-0-biomass-edition-toolbox-download/>.
- [26] Jong-Sen Lee, Shane R Cloude, Konstantinos P Papathanassiou, Mitchell R Grunes, and Iain H Woodhouse. Speckle filtering and coherence estimation of polarimetric sar interferometry data for forest applications. *IEEE Transactions on Geoscience and Remote Sensing*, 41(10):2254–2263, 2003.
- [27] Asli Ozdarici and Zuhail Akyurek. A comparison of sar filtering techniques on agricultural area identification. In *ASPRS 2010 Annual Conference*, pages 26–30, 2010.