

**A MODEL FOR DETECTION AND
CLASSIFICATION OF LANDSLIDES USING
SATELLITE DATA**

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Electronics and Communication Engineering

By

Akanksha Sharma

Registration Number: 42000381

Supervised By

Dr. Shakti Raj Chopra (11636)

**School of Electronics and Electrical
Engineering Department (Professor)**

Supervisor Affiliation

Lovely Professional University

Co-Supervised by

Dr. Suhas Gajanan Sapate

**Department of Computer Science
and Engineering (Professor)**

Supervisor Affiliation

**Dr. Bapuji Salunkhe Institute of
Engineering and Technology
Kolhapur**



**LOVELY PROFESSIONAL UNIVERSITY, PUNJAB
2025**

DECLARATION

I, hereby declared that the presented work in the thesis entitled “**A Model for Detection and Classification of landslide using Satellite data**” in fulfilment of degree of **Doctor of Philosophy (Ph. D.)** is outcome of research work carried out by me under the supervision of Dr. Shakti Raj Chopra, working as Professor, in the School of Electronics and Electrical Engineering of Lovely Professional University, Punjab, India, co-supervision of Dr. Suhas Gajana Sapate, Professor in computer science engineering department of Dr. Bapuji Salunkhe Institute of Engineering and Technology Kolhapur. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

(Signature of Scholar)

Name of the scholar: Akanksha Sharma

Registration No.: 42000381

Department/school: School of Electronic and Electrical Engineering

Lovely Professional University,

Punjab, India

CERTIFICATE

This is to certify that the work reported in the Ph. D. thesis entitled “A Model for Detection and Classification of landslide using Satellite data” submitted in fulfillment of the requirement for the award of degree of **Doctor of Philosophy (Ph.D.)** in the School of Electronic and Electrical Engineering, is a research work carried out by Akanksha Sharma, Registration No. 42000381, is bonafide record of his/her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.



(Signature of Supervisor)

Dr. Shakti Raj Chopra

Designation: Professor

Department/School: School of Electronics
Electrical Engineering

University: Lovely Professional University

(Signature of Co-Supervisor)

Dr. Suhas G Sapate

Designation: Professor

Department/school: Computer and
Science and Engineering.

University: Dr. Bapuji Salunkhe
Institute of Engineering and
Technology Kolhapur

ACKNOWLEDGEMENT

This thesis is a testament to the collective support, encouragement, and inspiration of those who believed in me and walked with me on this journey. I am thankful from the bottom of my heart to the many people who assisted me during this study.

First and foremost, I would like to convey my heartfelt gratitude to my supervisor, Dr. Shakti Raj Chopra, Professor, School of Electronics and Electrical Engineering, Lovely Professional University-Phagwara, India and co-supervisor Dr. Suhas Gajana Sapate, Professor in Computer Science and Engineering department of Dr. Bapuji Salunkhe Institute of Engineering and Technology, Kolhapur, India, for his excellent advice, critical criticism, and patience in every step of this research work during my Ph.D journey. Their thoughts and support guided me through the hurdles of this research and pushed me to persevere in difficult moments. Their intellectual guidance, constant encouragement, and insightful feedback were invaluable in shaping the direction and quality of this research.

I am extremely thankful and would like to express my gratitude to the entire Lovely Professional University family for providing the suitable environment and infrastructure for completing my research work in a timely manner.

Also, I would like to express my gratitude towards the School of Electronics and Electrical Engineering and Research and Development department for their motivation and support in my Ph.D journey.

I am thankful to God Almighty for blessing and grace showered on us to complete this work successfully.

To my family, I owe an immeasurable debt of gratitude. I feel heartfelt thanks to my parents Dr. Vijay Kumar and Mrs. Vidya Sharma, who have been my pillars of strength and have always believed in me. Their love and sacrifices have been the foundation of all my achievements. A special thank you to my husband, Vaibhav Sharma, whose patience, love, and encouragement have been my support through the highs and lows of this journey. His understanding, especially during the times I needed to prioritise my research, and his faith in my dreams have meant the world to me.

I am deeply thankful to my newborn son, Sharvik Sharma for bringing happiness and inspiration into my life during this journey.

A very special thanks to my friends and colleagues for the camaraderie, moral support, and invaluable discussions we shared.

Date: 09/07/2025

Akanksha Sharma

Abstract

Landslides are a periodic natural calamity that poses substantial dangers to human lives, environment and infrastructure, especially in hilly and densely inhabited locations. Developing countries have seen sharp growth in construction. Roads, railway tracks, bridges, tunnels and other transportation networks connect remote locations. Construction in the morphological area disrupts the ecosystem and creates threats such as landslides. Landslides are both natural and man-made disasters that result in fatalities. As a developing country, building cannot be halted, and natural factors that cause landslides cannot be managed. Accurate landslide prediction is vital for disaster preparedness and risk management. Traditional methods for landslide prediction rely on ground survey and statistical models, have limitations in term of data availability, scalability and accuracy. In a catastrophic emergency, reliable and efficient landslide detection can provide logical information to save the life. To meet the requirement of relief operations with accuracy and in time, this research proposes an automatic landslide detection with satellite images instead of the site visiting process in the traditional approach. Convolutional neural networks (CNN) are far more efficient than traditional methods for detecting landslides by lowering the time required to identify relevant features. A convolutional neural network is used to automatically extract information from satellite imagery and enhance the model's ability to detect early indications of landslides.

This research work explore the application of Convolutional Neural Networks, specifically sophisticated CNN backbone networks such as ResNet50, ResNet101, and GoogleNet, to detect and predict landslides using satellite images. This study compares three CNN architectures ResNet50, ResNet101 and GoogleNet, which were chosen for their proven skills in feature extraction and classification tasks. Satellite images provide broad coverage of huge and inaccessible areas, delivering timely data that is crucial for monitoring high-risk environments. However, the vast and complex nature of satellite data requires specialized approaches for finding significant patterns. CNNs have proven useful in a variety of image analysis applications due to their capacity to automatically learn hierarchical features from big datasets. This research work uses CNNs to discern complicated visual patterns indicative of landslides.

ResNet50 and ResNet101 are members of the residual networks family, which was created to address the vanishing gradient problem and enable the training of deeper networks, which is critical for collecting subtle landslide details. GoogleNet, while not as deep, has a distinct inception architecture that allows the network to record multiscale information in a single layer. Each model is trained with and without attention mechanisms to determine how attention layers affect model performance. Attention mechanisms, particularly spatial and channel attention, guide the network's focus to the most relevant portions of an image, significantly improving model interpretability and detection accuracy. The implementation of attention modules into CNN models in remote sensing image processing can improve the model's global context modeling and feature detection.

This work uses a deep learning CNN with attenuation mechanisms and optimization to extract landslides from satellite images to automatically identify landslides. The proposed method is divided into three steps: Pre-processing of the dataset that is an augmentation of labeled datasets; introduce attention module in the decoder to suppress feature map noise with three different backbone networks of CNN (ResNet50, ResNet101, Google Net) for training and performance evaluation of the proposed algorithm on quantitative parameters. The experimental setup includes thorough hyperparameter adjustment to optimize each model. Parameters such as learning rate, batch size, number of epochs, and weight decay are carefully controlled to avoid overfitting and underfitting, ensuring that the model generalizes effectively across data samples. Each network's performance is evaluated based on model training and validation metrics such as training loss, validation loss, accuracy, F1 score, precision and recall. Detailed epoch-wise analysis sheds light on each model's learning dynamics and the efficacy of various architectural options. Furthermore, early stopping criteria and dropout regularization techniques are utilized to prevent overfitting, while cross-validation improves model reliability.

The results show that ResNet-based architectures, notably ResNet101, outperformed all other models tested in terms of detection accuracy. The incorporation of attention mechanisms further increased performance and ResNet models, showing that these

processes assist in focusing the model on critical aspects within each image. The attention-augmented ResNet models demonstrated better accuracy and stability in identifying landslides, implying that attention plays an important role in improving the model's feature recognition capabilities. ResNet 101 obtained 0.9732 accuracy. ResNet 50 obtained 0.9692 accuracy and GoogleNet claimed 0.9648 accuracy. This indicates that the proposed attention module CNN model with ResNet 101 as a backbone network has high accuracy, and can be used as an effective landslide detection method to help in emergency rescue.

A comparative investigation of model performance reveals how architectural depth and attention layers affect the models' capacity to reliably discriminate landslide from non-landslide regions. The findings indicate that deeper networks are better suited to complex classification tasks such as landslide detection, where nuanced patterns are required for reliable predictions. The combined analysis confirms that attention mechanisms improve model performance, especially in complex architectures like ResNet101 and ResNet50, GoogleNet.

This study has great pragmatic consequences. The models used in this study have the potential to be integrated into early warning systems and disaster management frameworks, allowing authorities to monitor landslide-prone areas in near real time and make data-driven decisions about resource allocation and evacuation plans. This technology, which combines satellite data with deep learning algorithms, may produce prediction models that estimate landslide risk based on historical data and ongoing landscape changes. This work provides the framework for future research into landslide prediction utilizing multi-sensor data fusion. Incorporating other variables, such as rainfall, soil moisture, and seismic activity, could improve model accuracy and predictive power, allowing for a more comprehensive evaluation of landslide risk.

This research advances the field of landslide detection and prediction by proving the great accuracy with which satellite-based CNN models can identify landslide-prone locations. The combination of ResNet50, ResNet101, and GoogleNet architectures, together with attention mechanisms, represents a significant step forward in geohazard monitoring using deep learning and remote sensing. This work helps to build safer, more resilient communities in susceptible places by providing a scalable, accurate, and

rapid landslide analysis solution. These findings highlight the potential of deep learning and satellite imaging as transformational tools in the field of catastrophe management.

Key Words : *Landslide prediction , deep learning, convolutional neural network, remote sensing, satellite imagery.*

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
Abstract.....		vi-ix
List of figures.....		xiii-xv
List of Tables.....		xvi
List of Abbreviations.....		xvii-xx
Chapter 1 Introduction.....		1
1.1	Background and History of Landslide	1
1.2	Different Types of Mass Movement.....	5
1.3	Different types of Landslides	8
1.3.1	Slide.....	8
1.3.2	Rotational slide.....	9
1.3.3	Translational slide.....	9
1.3.4	Fall.....	9
1.3.5	Flow	10
1.3.6	Spreads.....	11
1.3.7	Topple.....	11
1.3.8	Avalanches	11
1.4	Reasons of Landslides	12
1.5	Problem Statement	13
1.6	Motivation	14
1.7	Organization of Thesis	15
Chapter 2 Literature survey, Problem definition and objectives		17
2.1	Machine learning based techniques	17
2.1.1	SVM-based landslide classification techniques	17
2.1.2	Bayesian model-based landslide classification techniques	20
2.1.3	Decision tree-based landslide classification techniques	22
2.1.4	Neural network-based landslide classification techniques	24
2.1.5	Fuzzy logic-based landslide classification techniques.....	25
2.2	Deep learning techniques.....	26
2.2.1	CNN-based landslide detection techniques.....	27
2.3	Other classification techniques.....	30
2.4	Findings and research gap.....	34
2.4.1	Challenges which are discussed	34
2.4.2	Challenges which are not discussed.....	35

2.4.3	Observations.....	35
2.4.4	Research gap.....	37
2.5	Problem Statement	38
2.6	Objectives	39
Chapter 3	Pre-processing of satellite images	40
3.1	Bijie landslide dataset.....	40
3.2	Satellite image Pre processing	41
3.2.1	The objective of this work	42
3.2.2	Related work.....	42
3.2.3	Methodology For Satellite Image Pre-processing.....	43
3.2.4	Radiometric correction and Geometric correction	48
3.2.5	Algorithm of Proposed Work.....	50
3.2.6	Results and discussion.....	52
3.3	Preprocessing in database.....	54
3.3.1	Preprocessing techniques	55
3.3.2	Principal Component analysis (PCA) and variants.....	59
3.3.3	Algorithm of Proposed Work.....	60
3.4	Conclusion	62
Chapter 4	Deep Neural Network for Detection of Landslide.....	63
4.1	Architecture of Convolution Neural Network	63
4.2	Deep Learning.....	67
4.2.1	Deep Residual Network	68
4.2.2	Architecture of ResNet 50.....	71
4.2.3	Architecture of ResNet 101.....	73
4.3	GoogleNet.....	74
4.4	Attention Mechanism	76
4.4.1	Spatial Attention	77
4.4.2	Channel Attention.....	78
4.4.3	Squeeze-and-Excitation (SE) Blocks	80
4.4.4	Self-Attention.....	81
4.5	Proposed work	81
4.5.1	Placement of Attention Module	84
4.6	Experimental setup.....	84
4.6.1	Evaluation parameter	84
4.6.2	Computational Complexity	85

4.6.3	Hyperparameters.....	86
4.6.4	Training and Validation	87
4.6.5	Hardware setup	88
4.7	Experimental Result Analysis of Google Net	88
4.7.1	GoogleNet Experimental Result without Attention Mechanism	89
4.7.2	GoogleNet Experimental Result with Attention Mechanism	95
4.7.3	Performance of GoogleNet.....	99
4.8	Result Analysis of ResNet 101.....	100
4.8.1	ResNet 101 Experimental Result without Attention Mechanism	101
4.8.2	ResNet 101 Experimental Result with Attention Mechanism	105
4.8.3	Performance of ResNet101	110
4.9	Result Analysis of ResNet 50	111
4.9.1	ResNet 50 Experimental Result without Attention Mechanism	112
4.9.2	ResNet 50 Experimental Result with Attention Mechanism	116
4.9.3	Performance of ResNet50	118
4.10	Discussion	119
Chapter 5	Conclusion and Future Scope	123
5.1	Future scope	124
References		126-133

LIST OF FIGURES

Figure 1.1 Landslide count in different countries.....	3
Figure 1.2 Fatality count in different countries	3
Figure 1.3 Landslide triggering factors	4
Figure 1.4 Fatality count due to landslide triggering factor from year 2003 to 2023....	5
Figure 1.5 Different types of mass movement based on speed and nature.....	5
Figure 1.6 Labels for different parts of Landslide.....	7
Figure 1.7 Rock fall.....	10
Figure 1.8 Debris flow.....	10
Figure 1.9 Topple landslide.....	11
Figure 2.1 Classification of images used for Landslide Monitoring	36
Figure 2.2 Classification Accuracy of ML/DL based techniques.....	36
Figure 2.3 Number of classifiers and types used in literature review.....	37
Figure 3.1 Examples of landslide instances from Bijie dataset	40
Figure 3.2 Satellite imagery errors	44
Figure 3.3 Rectification Processes	47
Figure 3.4 Geometrically Radiometrically and Atmospherically corrected image	49
Figure 3.5 Histograms of TOA Reflectance and Histograms of DN values.....	50
Figure 3.6 Flow Diagram of Proposed Pre-Process algorithm for Satellite images	51
Figure 3.7 Pre-Processed satellite image	53
Figure 3.8 Histogram analysis of pixel distribution original and processed image....	53
Figure 3.9 Brightness Transformation of image	56
Figure 3.10 Original image alongside five rotated images	57
Figure 3.11 Horizontally flipped version of Original image	58
Figure 3.12 Original and resized image.....	58
Figure 3.13 Proposed Preprocessing.....	61
Figure 4.1 Basic CNN Architecture.....	64
Figure 4.2 The Convolution layer function in CNN.....	65
Figure 4.3 Two Pooling operations with pooling window 2X2.....	66
Figure 4.4 Flattening operation.....	66
Figure 4.5 Skip Connection on ResNet.....	69

Figure 4.6 Bottleneck Residual.....	70
Figure 4.7 ResNet 50 Architecture	73
Figure 4.8 Skip and convolution connection	73
Figure 4.9 ResNet 101 architecture	74
Figure 4.10 GoogleNet Inception Module.....	76
Figure 4.11 Architecture of spatial attention	78
Figure 4.12 The architecture of Channel Attention	80
Figure 4.13 Architecture of Squeeze-and-Excitation (SE)	81
Figure 4.14 Proposed work flow diagram.....	82
Figure 4.15 Performance visualization of GoogleNet architecture without attention mechanism for epoch 9.....	90
Figure 4.16 Performance visualization of GoogleNet architecture without attention mechanism for epoch 5	90
Figure 4.17 Performance visualization of GoogleNet architecture without attention mechanism for epoch 9.....	93
Figure 4.18 Performance visualization of GoogleNet architecture without attention mechanism for epoch 4	92
Figure 4.19 Performance visualization of GoogleNet architecture with attention mechanism for epoch 9	96
Figure 4.20 Performance visualization of GoogleNet architecture with attention mechanism for epoch 9	98
Figure 4.21 Performance visualization of GoogleNet architecture with attention mechanism for epoch 7	99
Figure 4.22 Performance visualization of ResNet101 architecture without attention mechanism for epoch 7	102
Figure 4.23 Performance visualization of ResNet101 architecture without attention mechanism for epoch 9	104
Figure 4.24 Performance visualization of ResNet101 architecture without attention mechanism for epoch 3	105
Figure 4.25 Performance visualization of ResNet101 architecture with attention mechanism for epoch 9	107

Figure 4.26 Performance visualization of ResNet101 architecture with attention mechanism for epoch 5	108
Figure 4.27 Performance visualization of ResNet101 architecture with attention mechanism for epoch 5.....	109
Figure 4.28 Performance visualization of ResNet50 architecture without attention mechanism for epoch 7	113
Figure 4.29 Performance visualization of ResNet50 architecture without attention mechanism for epoch 5.....	114
Figure 4.30 Performance visualization of ResNet50 architecture without attention mechanism for epoch 5	115
Figure 4.31 Performance visualization of ResNet50 architecture with attention mechanism for epoch 7.....	117
Figure 4.30 Performance visualization of ResNet50 architecture with attention mechanism for epoch 8	118

LIST OF TABLES

Table 1.1 Comparision of different landslides.....	12
Table 3.1 Pre-processed image parameters.....	50
Table 3.2 Proposed algorithm Pre-processed image parameters	54
Table 3.3 Compare the result of PCA and Kernel PCA for different images.....	60
Table 4.1 Confusion matrix to classify landslide.....	84
Table 4.2 Training and validation outcomes from the 9 epochs.....	89
Table 4.3 Training and validation outcomes from the 5 epochs.....	91
Table 4.4 Training and validation outcomes from the 9 epochs.....	92
Table 4.5 Training and validation outcomes from the 4 epochs.....	94
Table 4.6 Training and validation outcomes from the 9 epochs.....	95
Table 4.7 Training and validation outcomes from the 9 epochs.....	97
Table 4.8 Training and validation outcomes from the 7 epochs.....	98
Table 4.9 Comparative result of GoogleNet with or without attention module	100
Table 4.10 Training and validation outcomes from the 7 epochs.....	101
Table 4.11 Training and validation outcomes from the 5 epochs.....	103
Table 4.12 Training and validation outcomes from the 3 epochs.....	104
Table 4.13 Training and validation outcomes from the 9 epochs.....	106
Table 4.14 Training and validation outcomes from the 5 epochs.....	108
Table 4.15 Training and validation outcomes from the 5 epochs.....	109
Table 4.16 Comparative result of ResNet101 with or without attention module	110
Table 4.17 Training and validation outcomes from the 7 epochs.....	112
Table 4.18 Training and validation outcomes from the 5 epochs.....	113
Table 4.19 Training and validation outcomes from the 5 epochs.....	114
Table 4.20 Training and validation outcomes from the 7 epochs.....	116
Table 4.21 Training and validation outcomes from the 5 epochs.....	117
Table 4.22 Comparative result of ResNet50 with or without attention module	119
Table 4.23 Comparative result of trained Model	120
Table 4.24 comparative result of proposed model and existing models.....	121

LIST OF ABBREVIATIONS

AI – Artificial Intelligence

ANN- Artificial Neural Network

AUC – Area Under the Curve

ATCOR-Atmospheric and Topographic Correction

ATREM- Atmospheric Correction for Remote Sensing.

BAM - Bottleneck Attention Module

BMPTC- Building Material and Technology Promotion

BGFC- Boosted Genetic Fuzzy Classifier

BRT- Boosted Regression Trees

CBAM - Convolutional Block Attention Module (CBAM)

CNN – Convolutional Neural Network

CRF- Conditional Random Forest

DL – Deep Learning

DT- Decision tree

DCNN-Deep Convolutional Neural Network

DEM – Digital Elevation Model

DN- Digital Number

DOS- Dark Object Subtract

ELBP- Extended Local Binary Pattern

FLDA- Fisher’s Linear Discriminant Analysis

FLAASH - Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes

FPR – False Positive Rate

FP – False Positive

FN – False Negative

F1 – F1 Score (Harmonic Mean of Precision and Recall)

FCN- Fully Convolutional Networks

GIS – Geographic Information System

GA-SVM - Genetic Algorithm Support Vector Machine

GAP-Global Average Pooling

GMP- Global Max Pooling

IoT – Internet of Things

JLD- Joint Landslide Database

KNN- K-Nearest Neighbors

KPCA – Kernel Principal Component Analysis

LMT-Logistic Model Tree

LDM- Landslide Detection Mapping

LKSVM- Linear Kernel base Support Vector Machine

LSZ- landslide susceptibility zone

LR – Learning Rate

LCLU- Land Cover and Land Use

MLP- MultiLayer Perceptron

MSRG- Modified Seeded Region Growing

MLR - Multinomial Logistic Regression

MLC- Maximum likelihood classification

MS- Multispectral

ML – Machine Learning

MOBIA- Multiscale Object-Based Image Analysis

MAE- Mean absolute error

NDVI-Normalized Difference Vegetation Index

OOA- Object-Oriented Analysis (OOA)

PCA – Principal Component Analysis

PSNR-Peak Signal to Noise Ratio

RKSVM- Radial Kernel-based Support Vector Machine

RVM- Relevance Vector Machine

RNN – Recurrent Neural Network

RGB – Red, Green, Blue (Color Model in Imaging)

ResNet – Residual Network

RMSE – Root Mean Square Error

ROC – Receiver Operating Characteristic

RecLD- Recent Landslide Database

ReLU- Rectified Linear Unit

RMS- Root Mean Square

RBF -Radial Basis Function

SRG- Seeded Region Growing

SCAM- Spatial Channel Attention Module

SE-Squeeze-and-Excitation

SVM- Support Vector Machine

SAR- Synthetic Aperture Radar

SPCA – Sparse Principal Component Analysis

SAR – Synthetic Aperture Radar

SNR – Signal-to-Noise Ratio

SSIM- Structural Similarity Index

TPR – True Positive Rate

TP – True Positive

TN – True Negative

TOA- Top of Atmospheric

UNet – U-Net Convolutional Network Architecture

UAV- Unmanned Air Vehicles

Chapter 1 Introduction

1.1 Background and History of Landslide

A landslide is a severe geological phenomenon that can devastate human lives and destroy infrastructure anywhere in the world. Landslides occur regularly during the rainy season when massive volumes of rock, debris, and dirt slide down a slope as a result of natural phenomena and human activity. Landslides, for example, result in irreversible damage to both people and infrastructure. In a catastrophic disaster, dependable and quick action is essential to save lives. In today's world, protecting people and infrastructure against natural disasters such as landslides is critical. As more mountain areas become occupied, there is a rise in governmental initiatives to ensure the safety of living beings in landslide-prone areas. Landslides can cause significant harm to both lives and property [1].

The Geological Survey of India claims 12.6% of covered land except snow-covered areas, is prone to landslides. About 0.32 million sq. km area falls under the Himalayan range, which is further categorized into the Northeast Himalaya and the Northwest Himalaya. Darjeeling and Sikkim fall under the North East Himalayas and cover 0.18 million sq. km area prone to landslides. Northwest Himalaya covers Uttarakhand, Himachal Pradesh and Jammu and Kashmir, comprising 0.14 million sq. Km. Western Ghats cover Tamil Nadu, Kerala, Karnataka, Goa, and Maharashtra, contributing 0.09 million sq. km and Eastern Ghat contributes 0.01 million sq. km of total landslide-prone area. The Himalayan range lies in earthquake Zone IV and V, these areas are susceptible to landslides initiated by earthquakes. The estimated loss of infrastructure due to landslides is 1 to 2% of the gross national product in most developing countries [2]. Estimating and minimizing the damage caused by landslides is a challenging task for the government authorities and technical teams in developing countries as approximately 80% of the casualties due to landslides are reported from these countries [3]. By survey of Building Material and Technology Promotion council (BMPTC) and TARU data landslide hazard probability is divided into three categories: Low, Medium and High. Landslide Hazard zonation Atlas claims that 8% of entire area of Himachal Pradesh is under high risk zone and by revised methodology Expert knowledge 3.2%

area is under high risk and AHP indicate 5.65% area is under high risk zone. In mountain areas landslides are most dangerous geological hazard [4].

According to the International Disaster Database report from 1990 to 2015, landslide events are 4.9% of all natural disaster events and 1.3 % of all natural hazard casualties during this period. Alone in Asia, 54% of landslide events take place. A total of 55997 human fatalities in 4862 distinct landslide events during the period 2004 to 2016[5]. Significant variations in weather lead unrivaled increase in catastrophic hazards worldwide. In Mocoa, Colombia 1 April 2017 a huge mudflow took place due to heavy rainfall, causing 300 fatalities and leaving thousands of people homeless [6]. A series of landslides in Brazil in 2022 claimed over 100 casualties [7]. In 2021 huge landslide blocked the flow of the Chenab river in the Lahul Spiti district of Himachal Pradesh, which created a threat of flood in many nearby villages [8]. Heavy rainfall in Kedarnath, Uttarakhand from 15 to 17 June 2013 caused numerous landslides, including mudflow and caused a high rate of fatalities, approximately 6000 casualties [9]. On 28 July 2021 massive landslide took place in the Chamba district of Himachal Pradesh, burying several villages and killing 14 people [10]. In Himachal Pradesh's Kothipura district, Mandi on 12 August 2017, a debris flow type landslide took place, causing 47 fatalities. In 1977, at the same site, a huge landslide took place and was reactivated again on 13 August 2007. Tension cracks, antecedent rainfall, rock mass, rise in soil moisture and increase in seismic activities were various causes. Still, its recurrence chances are possible, so continuous monitoring from satellite provides us with information or early alarming of the event [11]. Several landslides have occurred in the Kinnaur district of Himachal Pradesh, causing several deaths and property loss due to heavy rainfall, seismic activities and unplanned construction [12].

Using an open dataset available on NASA's data portal, we examined the temporal and spatial patterns of landslides in various countries across time. Figure 1.1 presents the geographic distribution of landslide events in various countries. The chart shows landslide counts, with darker colors indicating countries with more incidents. The visualization displays the global regions most affected by landslides, highlighting the importance of focused monitoring and mitigation measures. Analyzed data reveals that

United State followed United Kingdom, Canada, India and China have had a more significant number of landslides.

Landslide counts in different countries

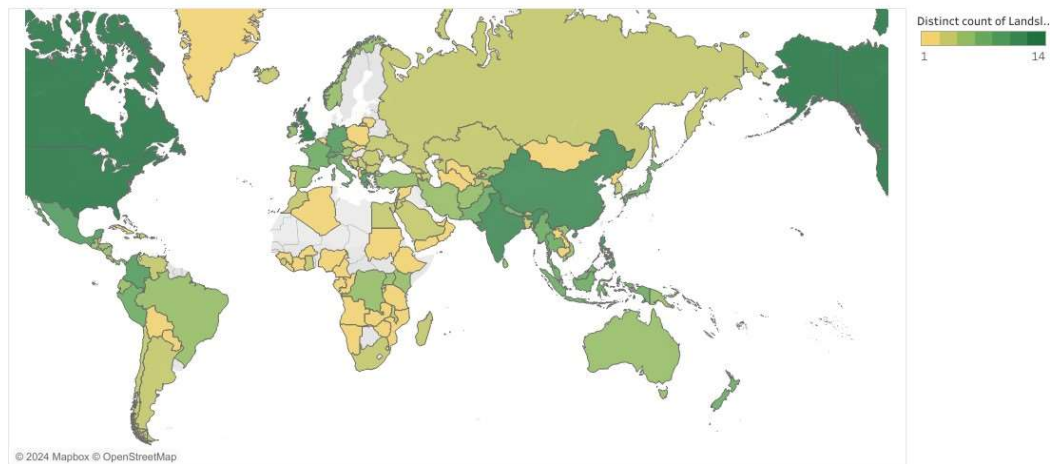


Figure 1.1 Landslide count in different countries

Figure 1.2 depicts the fatalities due to landslides across various countries using an open dataset available on NASA's data portal. In this study, countries are color-coded based on the overall number of fatalities caused by landslides from 1915 to 2023. Dark shades reflect more fatalities, whereas lighter shades suggest fewer fatalities. Analyzes reveal that Colombia has the highest fatality rate followed by India and China.

Fatality Count in Different Countries



Figure 1.2 Fatality count in different countries

In addition to monitoring landslide occurrences and fatalities, a different study was undertaken to identify the primary causes triggering landslides globally. On processing open NASA landslide dataset, landslides are classified on the basis of different triggering factors like: rainfall, snowfall, construction, earthquake, freeze-thaw, tropical cyclon, mining, flooding, etc. Figure 1.3 represents a bar chart that displays count of landslides triggered by different factors. Rainfall, which is a group of heavy rain, monsoon, downpour and continuous rain emerges as the most frequent triggering factor of landslides. It is investigated among various triggering elements rainfall sources of landslides that resulted in large mortality between 2000 and 2023. Following the rainfall is the earthquake, which claimed a large number of fatalities in 2018. Figure 1.4 also shows that the biggest number of fatalities occurred in 2013 as a result of a landslide caused by rain.

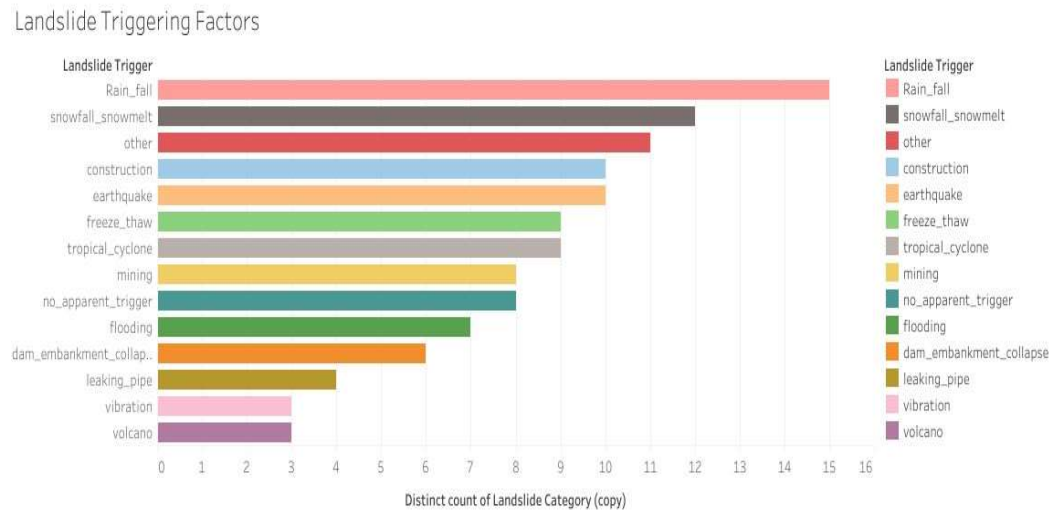


Figure 1.3 Landslide triggering factors

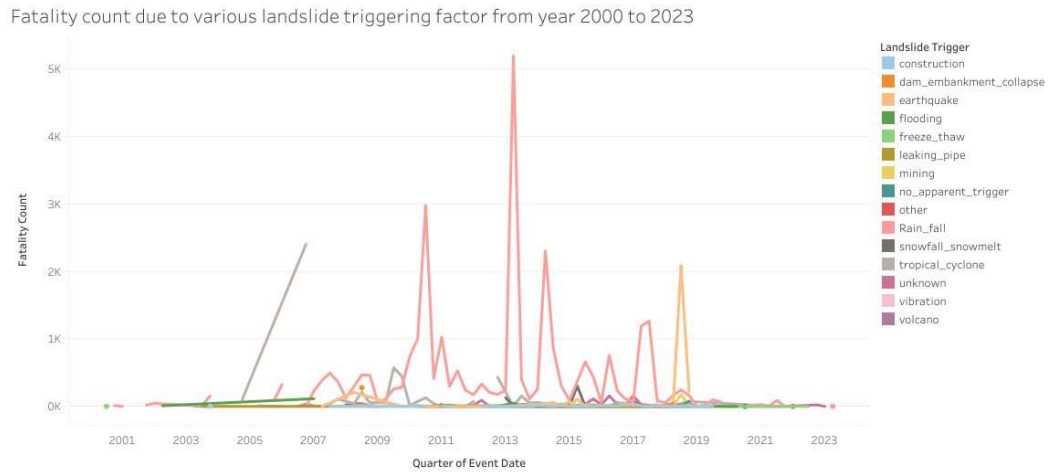


Figure 1.4 Fatality count due to various landslide triggering factor from year 2003 to 2023

1.2 Different Types of Mass Movement

Landslides are natural disasters that occur when rock, soil, and a combination of both slide down a slope due to gravity. A landslide is the movement of a mass of rock, earth, or debris downslope. Mass movement can be grouped under three major categories based on the speed and nature of the movement: Slow movement, Rapid movement and Landslide [13], shown in Figure 1.5 below.

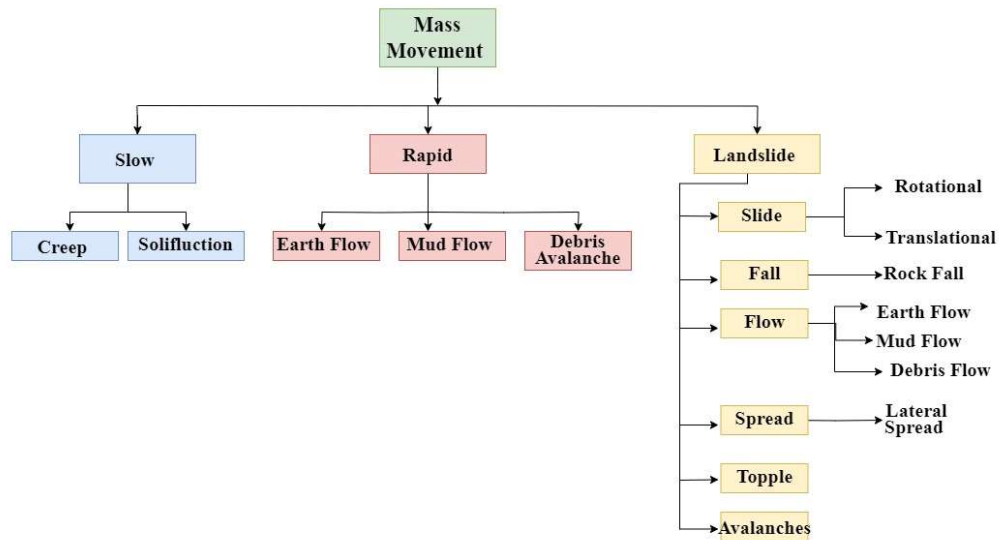


Figure 1.5 Different types of mass movement based on speed and nature of the movement.

1. **Slow movement** is a movement of earth material that is going to be very slow and goes on for years. These are progressive motions of soil or rock down a slope that are typically invisible over short periods of time. The migration occurs gradually over a lengthy period, frequently taking years to demonstrate substantial impacts. Creep and solifluction are two types of slow movement.

Creep is a form of landslide that involves the slow, steady flow of soil, rock, or debris down a slope. This movement is usually negligible in the near term, but over time, it can cause significant material displacement. The creep effect is caused by gravity and is frequently increased by factors such as repeated freeze-thaw cycles, changes in soil moisture and the expansion and contraction of materials owing to temperature fluctuations. Tilted trees, cracks in buildings, bent fences and utility poles are a few examples of creep.

Solifluction is the slow flow of water-saturated soil downslope, which is typical in permafrost locations or where the earth is frozen for a portion of the year. During the warmer months, the thawed soil progressively travels over the frozen layer beneath.

2. **Rapid Movement** is a movement of earth material going to be rapidly fast. Rapid mass movements happen suddenly and without warning, and they can be extremely damaging. These motions often involve loose soil, rocks, and debris falling or sliding down slopes due to gravity and are frequently driven by events such as excessive rains, earthquakes, or volcanic activity. Rapid movements are of three types: Earth flow, Mud Flow, Debris Flow.

Earth Flow: Massive amounts of fine-grained water-saturated soil flow down the slope under the pull of gravity. These flows move quicker than creeps but slower than mudflows or debris flows.

Mud Flow: Mud flow is similar to earth flow, but much bigger in quantity. A thick layer of mud, rock, and debris that moves down with the water is particularly devastating. Mudflows can be caused by severe rains or volcanic activity and they frequently follow existing routes such as rivers or valleys.

Debris Avalanche: A debris avalanche is a fast-moving and chaotic mass movement that, like a snow avalanche, comprises the rapid flow of a mixture of soil, and other debris down a steep slope.

3. **Landslides** refer to the downslope movement of rock, soil, or debris. They are typically caused by severe rainfall, earthquakes, volcanic activity, or human activities that destabilize slopes. The phrase "landslide" is commonly used to indicate rapid mass movements, but it can also refer to slower movements. Landslide relies on the form of irregularity in rock, the degree of weathering, and the steepness of the slope. Landslides can be explained using two terms: material and movement. The type of movement specifies the exact internal principles of how the landslide mass is displaced such as slide, fall, flow, spread, topple. Understanding the many components of a landslide is critical for studying the mechanics of slope failure and landslide evolution. Figure 1.6 depicts the anatomy of a rotational landslide that has converted into an earthflow, according to Varnes (1978)[13].

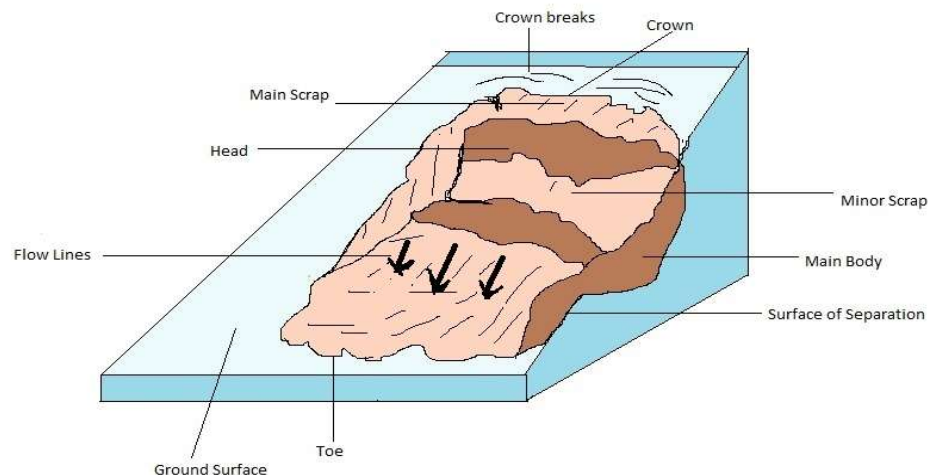


Figure 1.6 Labels for different parts of the Landslide

Crown: The topmost area of the landslide, where the original ground surface remains visible. Crown breaks are frequently seen on the crown, which form as the ground begins to move.

Main Scrap: The steep, exposed area at the head of the landslide where the material has broken away from its original position. This is usually the most visible characteristic of a landslide, indicating the point of early breakdown.

Head: The head is located at the top of the landslide and contains the primary scarp as well as the upper portion of displaced debris. This is where the landslide movement starts.

Flow lines: Arrows or lines indicate the direction of movement inside the landslide mass.

Toe: The lowest point of the landslide is where the dislodged debris comes to rest. It's also where the sliding material goes downslope.

Surface of separation: This is the lowest border of the displaced material, separating the landslide from the underlying stable ground.

Minor Scrap: A tiny break in the surface of the landslide, usually indicating additional movement or deformation within the main body of the landslide.

Main Body: The middle area of the landslide contains the majority of the displaced debris.

1.3 Different types of Landslides

Different types of landslides are listed below:

1.3.1 Slide

Slide type of landslide occurs when a mass of rock, soil, or debris moves downslope along a definite surface of rupture, which is typically a rather well-defined slip plane. The movement happens when the force of gravity overcomes the material's strength, which is commonly caused by saturation with water or other destabilizing forces. A slide's move can vary significantly, ranging from very slow (creeping) to rapid, contingent upon the steepness of the slope, the characteristics of the material, and external triggers. Roads, buildings, and natural areas can sustain serious damage from slides, particularly if they happen quickly and involve a lot of material. A number of factors, including heavy rain, fast melting snow, earthquakes, volcanic eruptions, and human activity that disturbs the slope (such as construction or deforestation), can cause slides. There are two types of slides: Rotational slide and Translational Slide.

1.3.2 Rotational slide

Rotational slide is a landslide characterized by an upwardly spoon-shaped rupture surface and a slide that goes largely down an axis parallel to the slope contour. The displaced material may move nearly vertically downward at its head, leaning backward toward the scarp on its upper surface. A slump is a rotating slide that moves on numerous parallel, curved surfaces. Rotational slides are frequently caused by water saturation or the slope's undercutting and are common in uniform materials like clay or soft rock.

1.3.3 Translational slide

Translational slide is the mass flows outward, or down and outward, along a relatively flat terrain such as a fault line, bedding plane or with weak layers of slope, with little rotational movement or rearward tilt. These slides are very common worldwide and frequently occur in places with fractured rock formations or slopes that have pre-existing fractures. Translational slides may initially be sluggish, causing damage to property or lifelines, but in extreme situations, they can accelerate and become life-threatening.

1.3.4 Fall

Fall is a sudden downhill movement of soil rock and debris from cliffs causing a bounce down the slope as falling material strikes with less angle to the lower slope. Falling material continues to roll on the slope till terrain is flat. These landslides are very rapid and material such as soil, and rock bounce and roll. The velocity depends on the slope of cliff as shown in Figure 1.7. Falling rock and material can cause fatalities, damage to infrastructure and block the highways.

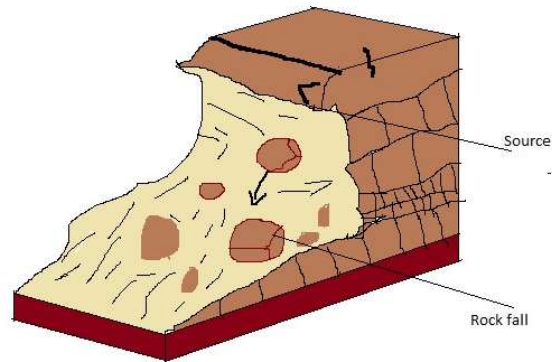


Figure 1.7 Rock Fall

1.3.5 Flow

Flow is a mass movement of loose mud, sand, soil, water and debris downward under the influence of gravity. The substance acts like a fluid, with individual particles moving randomly. Flow rates can vary from sluggish to very fast. Earth flow, mud flow, debris flow and creep are different types of flow. Consistency and slope define the slow and rapid nature of flow. Flows are commonly triggered by heavy rain, sudden snow melt and frequently deployed by other types of landslides that occur on downward slopes, are almost saturated, and contain a high proportion of silt and sand-sized material. They can carry things as large as buildings along the slope flow or quickly fill structures with sediment and organic waste. They can impact the quality of water by depositing enormous amounts of silt and debris. Figure 1.8 is a labelled diagram of debris flow.

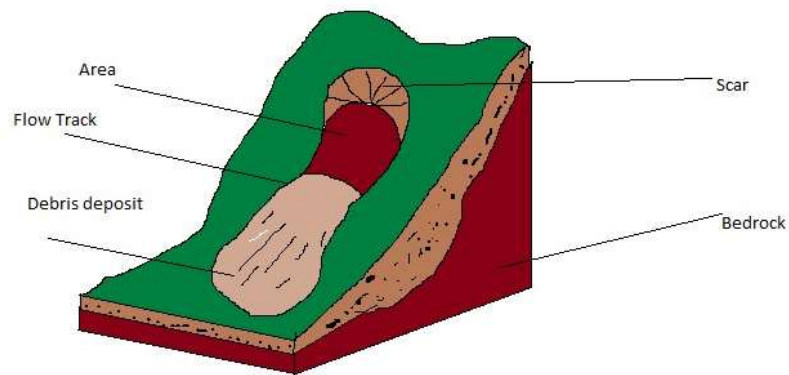


Figure 1.8 Debris flow

1.3.6 Spreads

Spreads are the horizontal movement of big, cohesive masses of material. They typically occur on mild slopes or flat terrain and are often caused by the liquefaction of weak soils during an earthquake. After specific triggering mechanisms such as earthquakes, it can be gradual, moderate, or even quick

1.3.7 Topple

Topple is defined as the forward rotation of a mass of soil or rock along a slope around a point or axis below the displaced mass's center of gravity as shown in Figure 1.9. Toppling can be triggered by gravity acting on the weight of material upslope from the displaced mass. Topples can be made of coarse material such as rock, or debris, and fine-grained material such as soil. Topples can be both complex and composite. Topple can be exceedingly devastating, especially if the fall occurs suddenly and at a high velocity.

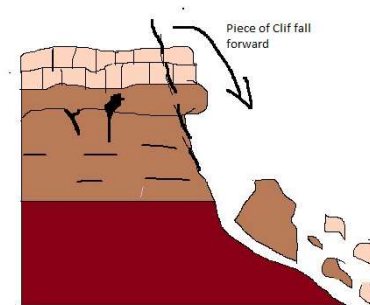


Figure 1.9 Topple Landslide

1.3.8 Avalanches

Avalanches are the rapid downslope flow of snow, ice, boulders, or debris. Snow avalanches are the most prevalent, although rock and debris avalanches can occur, particularly in steep mountain areas. Triggering factors of avalanches are heavy snowfall, rapid snowmelt, volcanic earthquakes and earthquakes.

Different types of landslides are compared in table 1.1 below on the base of material, movement and speed of material

Table 1.1 Comparison of different landslides

Type	Material	Movement	Speed
1. Slide	Soil, Rock, Debris	Coherent movement along a plane	Slow to Rapid
2. Fall	Rock, Soil, Debris	Bouncing or free fall sudden downhill movement	Very Rapid
3. Flow	Mud, Soil, Debris	Erratic, Fluid-like movement, downward under the influence of gravity	Slow to rapid
4. Spread	Soil, Debris, cohesive masses of material	horizontal movement	Slow to rapid
5. Topples	Mass of soil, Rock, Debris	forward rotation	Slow, Rapid and sudden
6. Avalanche	Debris, snow, ice	Rapid downslope flow	Very Fast

1.4 Reasons of Landslides

Landslides are caused by a variety of natural and human-induced events, including excessive rainfall, earthquakes, volcanic activity, erosion, and human activities such as construction, mining and deforestation. They can vary in size, speed and material composition, causing significant damage to people, infrastructure, and the environment.

The following are some of the main causes of landslides:

1. Geological elements that lower slope stability, such as loose or weak soil and rock pieces, are responsible for landslides. Pre-existing cracks and fractures in rock create weakness and lead to landslides. The material is further weakened by high pore water pressure in soils and weathering of rocks.
2. Water related factors, such as heavy rainfall and erosion, can trigger landslides. Heavy rain saturates the soil and reduces its strength to hold material, which causes slope failure. An increase in groundwater level due to rain or snowmelt can erode the base of the slope near rivers can triggering landslides.

3. Human activities like roads, dams, building construction, destabilizing the slope and creating a risk of landslides. Deforestation, improper irrigation and improper drainage systems also saturate the soil and cause landslides
4. Climate change, such as an increased frequency of intense rainfall, sudden glacier retreat in mountains leads slope disability and triggers debris flow.
5. Seismic activity, both natural and human-induced, can trigger landslides by causing the ground to shake and destabilize slopes. Natural earthquakes produce vibrations that weaken the bond between the particles and create cracks can cause landslides. Human activities like large construction, deep drilling and mining can cause vibrations, potentially leading to landslides in areas with unstable material.

1.5 Problem Statement

Landslides are a common and catastrophic natural hazard that mostly affects hilly and mountainous areas. They can seriously harm infrastructure, the environment, and result in fatalities. There are numerous case studies based on field surveys that are used for landslide detection and monitoring. Traditional landslide prediction methods, which depend on topographical, geological, and water cycle data, usually struggle with accuracy due to the complicated relationships between various components such as rainfall, seismic activity, soil composition, and topographical features [5]. All the traditional approaches to detecting, classifying and monitoring landslides such as field-based investigation, topographic and geographical mapping, geo technical methods, remote sensing techniques using aerial photos, GIS-based susceptibility mapping, and rainfall threshold analysis etc. are reliable but very time-consuming. These methods also face challenges in real-time monitoring and large-scale prediction, especially in remote areas. Automatic landslide classification, detection and monitoring are possible with the advancement in satellite image processing.

Recent development in satellite imagery and artificial intelligence technologies provides large data; nevertheless, analyzing and interpreting this data to effectively anticipate landslides is a major difficulty. For analysis of landslide detection aerial images have been widely used and provide good accuracy [14]. Data from the Digital elevation model perform a key role in the detection and prediction of landslides by

giving topographic information. For landslide classification and prediction-number of machine learning algorithms were used. Machine learning classification schemes are categorized as SVM classifiers, clustering-based classifiers, learning-based classifiers, fuzzy classifiers, and Bayesian classifiers [15][16]. These classification machine learning algorithms are based on low-level features which result in poor classification accuracy [17]. Deep learning is a subset of machine learning that has evidenced its efficiency in classification and prediction with satellite images in the past few years. Deep learning algorithms, with their ability to study enormous datasets and identify patterns, provide an intriguing potential solution to the challenges. Different deep learning algorithms integrated with attention mechanisms handle diverse data and deliver accurate predictions but time and other factors like soil moisture level, weather data, and rainfall patterns remain further research challenges. The goal of this research work is to develop an attention-based deep convolutional network with improved accuracy. This model focuses on providing help for disaster management in the timely prediction of landslides.

1.6 Motivation

Landslides pose serious threats to human safety, infrastructure, and the environment, especially in mountainous and hilly areas where they occur frequently. Developing countries follow a steep increase in construction. Remote areas are connected to roads, railway tracks, bridges, tunnels etc. Constructions in the morphological area cause a problem in the ecosystem environment and create hazards like landslides. A landslide is a natural and manmade disaster that causes loss of life. Being a developing country, construction cannot be stopped and natural parameters that trigger landslides cannot be controlled. Although traditional prediction techniques are available, their effectiveness is limited in real-time surveillance across large inaccessible areas. Recent developments in technology, such as satellite imaging, have provided new possibilities to improve landslide prediction. The ability to continuously and remotely monitor vast areas has created opportunities for more flexible and extensive prediction systems. However, the most difficult challenge is to discover how to use this data to make accurate forecasts. Deep learning models present a promising path because of their ability to process enormous datasets and identify intricate patterns from several sources. However,

substantial gaps remain in their application in landslide prediction. This research work addresses these gaps by developing a deep learning-based landslide prediction model with improved accuracy.

The primary goal of this research is to develop a deep learning-based model for landslide prediction by integrating with an attention mechanism along with satellite images. Preprocess the images and create a training and testing landslide dataset with a large number of images. Train the model using landslide and non-landslide datasets and compare the model's performance with traditional prediction methods. Then evaluate the model's accuracy, precision, recall and timeliness in predicting landslides. The focus will be on improving the reliability of predictions. Ensuring the model's applicability for large-scale landslide-prone regions, including remote or inaccessible areas.

1.7 Organization of Thesis

The thesis is organized into 5 chapters, each chapter is a critical component of the research on developing a deep learning-based landslide prediction model using satellite imagery. The arrangement is as follows:

Chapter 1: Introduction

This chapter presents the research problem in perspective with the history of landslides, their effects on society and the causes of various types of landslides. It discusses the problems and limitations of traditional landslide prediction approaches and focuses on deep learning techniques along with satellite imagery. Problem statements, motivation and research objectives are outlined in this chapter.

Chapter 2: Literature Review

This chapter presents a comprehensive study of previous research in traditional approaches as well as latest machine learning, deep learning and remote sensing. Research gaps are identified in the survey of present research in relevant fields. This Chapter defines the problem statement and objective of the work.

Chapter 3: Preprocessing of satellite images

This chapter explains the Bijie landslide dataset and the pretreatment methods, which included radiometric and geometric corrections. It also describes the methodology used for satellite picture preparation and presents an algorithm for the proposed task, along with preliminary findings and discussions.

Chapter 4: Deep Neural Networks for detection of landslides

This chapter discusses the architecture of deep learning models, namely, convolutional neural networks (CNNs), deep residual networks such as ResNet 50 and ResNet 101, GoogleNet, and attention mechanisms (spatial, channel, and self-attention). The results of the proposed work are also discussed in this chapter.

Chapter 5: Conclusion and future scope

The final chapter summarizes the research findings, contributions, and probable future directions.

Chapter 2 Literature survey, Problem definition and objectives

This section takes a comprehensive review of different machine-learning, deep learning algorithms and methodologies for landslide detection and classification using satellite data

2.1 Machine learning based techniques

The literature studied reveals that the entire Machine learning algorithm used for landslide detection or classification can be divided into four main categories: supervised learning-based algorithms, unsupervised learning-based algorithms, Fuzzy classification algorithms and combination or hybrid classification algorithms. Hence, we have grouped the methodologies and their summaries in four different sub-sections as below.

2.1.1 SVM-based landslide classification techniques

This subsection summaries all Machine learning techniques under supervised learning-based algorithms as below:

Utsav Kumar Malviya et al. [18] used learning-based Extended Local Binary Patterns and SVM for the classification of 24 different class satellite images. Two major issues with satellite image processing were discovered in this paper: noise is more noticeable in satellite images and different satellite images have unique properties. The SVM algorithm is used to estimate the noise pattern and Local Binary Pattern used for segmentation. In this research, the researcher considers only four different classes of pictures for training the framework with three algorithms: Radial Kernel-based Support Vector Machine, Linear Kernel base Support Vector Machine, and extended Local Binary Patterns. Extended Local Binary Patterns is preferred which correctly classify all 24 images. The overall 0.94 accurate result was obtained by the ELBP SVM algorithm for satellite image classification. Satellite images have unique features and have varieties in texture and quite difficult to propose one strategy for all images. Still, work needs to be done to design a more accurate algorithm to give improved results for the classification of different classes of satellite images. Only a few images for training cannot guarantee better accuracy. The robustness of technique with more dataset is not attempted which may be the bottleneck in its applicability.

Young Gi Byun et al. [19] proposed a landcover classification multispectral image approach based on the Seeded Region Growing (SRG) approach. Efficient image segmentation techniques and high-resolution pan-sharpened images were used. The modified SRG approach combines the multispectral and gradient information of images for homogeneous image regions with accurate and close boundaries. In the noise removal process of multispectral images multi-valued anisotropic diffusion method was used to collect edge information for extracting seed points local minima. Two datasets Quick Bird image and GeoEye-1 were used for experimental results. At a threshold value of 0.5 and mean square spectral error, the proposed algorithm provided the best result and has an accuracy of 0.9115 and the kappa coefficient is 0.9670. MSRG can use multi-feature information including edge and multi-spectral information. This proposed method uses a threshold value for seed selection which cannot provide the best result of seed section for every image. The work needs to be done in an area that is more efficient in segmentation.

Chanika Sukawattanaavit et al. [20] developed GA SVM algorithm for the classification of multi-frequency images from RADARSAT-2 (RS2), Synthetic Aperture Radar (SAR) and Thaichote (THEOS) MS images. SVM classifier was used for the classification of land cover. To obtain the best input feature GA was used. Function classification accuracy and the number of features in the selected subset were used to define the fitness of the function. Two datasets THEOS & LANDSAT8 of MS images were used for experiments. To convert the intercorrelated MS band into a set of non-correlated components PCA was used. Training sets and testing sets were developed by using the ENVI program GA-SVM algorithm was compared with the grid search algorithm based on parameter searching. GA-SVM algorithm has 85.02% accuracy for THEOS images and 0.95 accuracy with combined RS2 and THEOS images. The genetic algorithm along with SVM provides better results as compared to grid search but the Genetic algorithm can be computationally intensive and time-consuming for large datasets. High classification accuracy was achieved with fused RS2 and THEOS images and performance might be different with other testing datasets.

Xin Huang et al. [21] proposed a multi-feature model-based SVM that combines multiple spatial and spectral features both for object and pixel levels. Differential

morphological profiles Gray-level, co-occurrence matrix and an urban complexity index, are three features that were used. Probabilistic fusion, object-based semantics and certainty voting three algorithms were proposed to add multiple features. Two WorldView-2 datasets and DC Mall dataset were used for training and testing. In DC Mall 50 samples were used in the training process and 19332 in testing. For the classification of high-resolution imagery data, one optimal feature for different images was impossible to select. In the proposed multi-feature, SVM was based on multi classifier system that contain a series of spatial and spectral features for high-resolution image classification. Newly developed SVM has 0.944 accuracy with GLCM on DC Mall dataset. With the Worldview-2 dataset developed SVM has 0.928 accuracy. This work is limited to training sets and knowledge base rules for construction. Two datasets used in the experimental result used a limited number of datasets for training does not provide efficient results. Semantic analysis was used for the post-processing feature system and depended on segmentation quality which can reduce the overall classification accuracy.

Dericks P. Shukla et al. [22] discussed the survey of different LSZ map approaches for preparing landslide susceptibility zonation maps with support vector machine by considering one case study on the area of Garhwal. The datasets were prepared from the survey of the India toposheet. To finalize the tectonic map of the selected area, Landsat satellite images of 30 m resolution were used. Data is pre-processed with ArcGIS software to generate parameters such as soil, aspect ratio, drainages, and elevation of the study area. The vector layer of 30x30 m resolution data set was converted into Raster data and raster to ASCII format to use Matlab for SVM. To test the trained SVM Model Ukhimath river basin data were used which was prepared by the geologic survey of India. The trained proximal SVM model to classify more areas in landslides susceptible zone have a classification higher accuracy of 0.842 and prediction accuracy of 0.8115. Preparing a landslide susceptibility zonation map for an area that is sensitive to landslides is most important. The focus of such kind of map prepared with a Support vector machine is to identify the landslide-prone areas.

Kadir Sabanci et al. [23] compared the results of K-Nearest Neighbor Algorithm and multilayer perceptron (MLP) for the classification of varied forest types to classify the

dataset data mining methods used. A dataset of ASTER satellite images was created and the collected images were processed in three parts: Classification, regression, and clustering along with association rules. To train the model, training sets of ASTER Satellite images were used to classify the sample images into different classes. A total of 524 images were used of which 38% data was used for training and testing 62% data was used. The machine learning algorithm MLP yielded a classification accuracy of 0.9043 and KNN produced 0.8910 accuracy. KNN and MLP have the best classification accuracy. In this research training set is only 38 % and by increasing the ratio of a training set the result can be further improved.

2.1.2 Bayesian model based landslide classification techniques

This subsection area summarizes all Machine learning techniques under Bayesian model-based algorithms as below.

Fereidoun A. Mianji et al. [24] proposed a modified supervised classification method in which the feature reduction technique combined with Bayesian learning-based probabilistic sparse kernel method. To increase the distance between the classes, hyperspectral data was first transferred to low-dimensionality feature space and processed with a multiclass RVM classifier. The proposed method uses a dataset of AVIRIS with a resolution of 10nm and wavelength of 0.4 to 2.5micro m images. This dataset contains two datasets Indian Pine and San Diego dataset. The experiment was performed for both Linear [FLDA+RVM] and nonlinear [GNDA+RVM] and the performance of the proposed methods was evaluated on varying trains to test the sample. The overall accuracy of Linear FLDA+RVM and GNDA+RVM was 0.9801 and 0.9904 when the train-to-test sample ratio is 1:30 respectively. Real Hyperspectral data is used for verifying the effectiveness of this proposed supervise classification method. The result is compared with the SVM algorithm and this proposed method gives better performance over SVM.

Jun Li et al. [25] investigated an active sampling supervised Bayesian approach with active learning for the segmentation of Hyperspectral images. A multinomial logistic regression model based on logic regression was used for class posterior probability distribution learning Unbiased multilevel logistic prior (MLP) was used to encode spatial information and segment the hyperspectral images. Active learning is useful for

reducing number of labelled samples. Gaussian RBF kernel is applied for all experiments to normalize the input hyperspectral data. The LORSAL algorithm was used to learn MLR (multinomial logistic regression). The Multilevel logistic (MLL) prior model was adopted for smooth segmentation. The researcher designed an algorithm that combines LORSAL, MLL and active learning. To evaluate proposed algorithm datasets Indian pines, AVIRIS and ROSIS Pavia were used for experimental results. The proposed algorithm yielded 0.8672 accuracy on 3921 labelled sample. Overall accuracy Based on experimental results MBT approach gives unbiased sampling and better classification. In this paper, the main dominating factor is a limited dataset for the algorithm performance evaluation. The result can be modified with more training samples.

Pablo Ruiz et al. [26] proposed a Remote sensing image classification-based method for nonparametric and interference paradigms. This approach allows dealing with infinite dimension features. For both fine and infinite dimension feature space this method is useful. This scheme provides point-wise class prediction and confidence interval prediction. This method is efficiently used for supervised and active learning. The experimental result of this proposed algorithm was performed over two multispectral images for supervised and active learning classification. Landsat images of Rome city were acquired for supervised classification and ROSIS images of Pavia city were used for active classification. Multispectral and synthetic aperture radar data is used to test this algorithm and Hyperspectral images are used for multiclass land cover classification. The proposed method has 0.9680 overall accuracy in supervised mode. For active learning minimum normalized distance (BAL-3) has 0.9734 accuracy and running time is 9s. In the supervised mode, proposed algorithm provides the same result as compared to SVM but an improvement is observed in active learning. This work can provide pointwise class prediction and confidence intervals. To an extent this work can use multitemporal image segmentation for better results.

Zhaobin Cui et al. [27] suggested a novel classification method for multispectral (MS) images and this approach was based on nonparametric supervised classification. To provide a digital vector number of different class statistic distributions were followed. In MS image high posterior probability was calculated only when an unknown pixel digital

number is the same as this pixel in a training class. To estimate the maximum posterior optimized simulated algorithm was used in the proposed method. Spectral classification of the proposed approach yielded 0.8530 accuracy and 0.799 Kappa coefficients for the first dataset. Spectral Spatial classification of the proposed approach yielded 0.9478 accuracy and 0.92 Kappa coefficient for the first dataset. Three datasets of multispectral images acquired from the SPOT6 satellite have four bands and each band has a spatial resolution of 2m. The proposed Bayesian approach has better results than the traditional approach. This approach uses the Gaussian Mixture model for fitting the training dataset instead traditional single Gaussian model to provide better results.

2.1.3 Decision tree based landslide classification techniques

This subsection area summaries all Machine learning techniques under Decision tree based algorithms as below

Dennis C. Duro et al. [28] explored a multiscale object-based image analysis (MOBIA) approach based on an RF classifier for EO imagery. MOBIA can produce more than a dozen variables for classification as compared to the pixel-based approach. The use of object features to evaluate information from multispectral bands vegetation index and digital elevation model or other input layers is possible with MOBIA. For object-based classification, object features are used for calculating individual image objects and provide a segmentation process. Maximum likelihood classification (MLC) and K-nearest neighbor (k-NN) are traditional classification algorithms used for MOBIA classification. As compared to modern or parametric algorithms MLC gives poor classification results. In nonparametric algorithms, the RF classifier is more faster and reliable for MOBIA. Two datasets from SPOT-5 high-resolution geometrics sensors and LANDSat-5's thematic mapper sensor were used for testing and training. For multisource, multi-sensor data RF classifier accuracy is 0.90. This approach consistently gives 0.85 accuracy with RF algorithm. The data used is of high resolution of 10 m and quite complex to collect data for the training process. This algorithm can be implemented with more datasets to improve the result.

Lena Albert et al. [29] introduced a classification approach for land cover and land use (LCLU). This classification approach focuses on spatial and semantic context for

LCLU classification simultaneously. In land cover land use classification conditional random field was applied. Nodes are used as super-pixels in the land cover layer and nodes represent the object in the land use layer. An iterative inference procedure was introduced to enable inference in high-order Conditional Random Forest (CRF). Aerial images were used as input for this proposed classification approach. Two test sets located in Germany were used for testing the algorithm and all these pictures were of orthophoto with four channels and 0.2m ground sampling distance. The result is homogenous for land cover classification and the classification result is improved for similar land use. The overall accuracy for the first test set is 0.837 and for the second set is 0.825. The size of the super-pixel is very useful for good classification results. As compared to the non-contextual classifier proposed approach gives better experimental results.

Javier A. Montoya Zegarra et al. [30] proposed approach is multi-class semantic segmentation with class-specific for high-resolution aerial images. This research includes prior knowledge about the layout in the CRF model. The first step starts with a Pixel-wise prediction of the class likelihood. For better results, the appearance feature sampled from the neighborhood of each pixel was considered. From object specifies the assumption high-level representation at the level of the object was added. The hypothesis was for road segments and buildings. In the classifier stage, all pixels that belong to the hypothesis were assigned the same level. Experimental results were performed on 1000x1000 pixels file generated from dense matching from Vaihingen dataset. This model consists of three steps: The first step is the input of aerial data, then passes through a multilevel classifier with good appearance feature extraction and the last is the recovery step. In the second step large window of the classifier is used because of this building boundaries get blurred and boundaries get mixed even if buildings are close enough. Overall 0.8242 accuracy was achieved with experimental results. This classifier Accuracy is given by CRF for buildings, roads, grass, tree and background. Classifiers give more than 0.80 accuracy but the boundaries of roads and buildings were blurred. The proposed approach is useful for urban planning and environmental monitoring. The complexity, computational cost, and sensitivity to extreme variations of objects are a few disadvantages that can be improved by improving datasets.

2.1.4 Neural network based landslide classification techniques

This subsection summaries all Machine learning techniques under Neural Network supervised learning based algorithms as below.

Nur Anis Mahmon et al. [31] surveyed different algorithms which were backpropagation and K mean algorithm for the classification of satellite images with different classification methodologies. ANN's classifier approach was compared with convolutional classifier techniques which are Maximum Likelihood (ML) and unsupervised (ISODATA). To cover the different types of area, present work categorized the LU/LC into three different classes. Either output of k means clustering image output or ground truth data samples were used as a training set. The training set was selected randomly in this research. Accuracy and kappa coefficient were used to compare the result of image classification. Overall accuracy is 0.893 and the kappa coefficient is 0.820.

Rachid Sammouda et al. [32] introduced a Hopfield Neural Network for agriculture satellite images. Pixel clustering-based segmentation was performed on Satellite images which is quite difficult due to poor resolution, poor illumination and environmental conditions. Geo Eye satellite images dataset with 0.5m resolution was used for clustering. Hopfield Neural Network is giving good results when using three, four and five clusters in terms of classification sensitivity and accuracy.

Wei Zhao et al. [33] presented a Convolutional Neural Network model for multispectral and panchromatic image classification. The model introduced in this paper was a super pixel-based multiple local CNN. A very high-resolution multispectral and panchromatic images were fused together to achieve results. The introduced CNN model was valid for two datasets one was prepared from the DEIMOS-2 satellite for Vancouver images and the other was prepared from Quick Bird Satellite for China images. Both dataset images were MS remote-sensing and panchromatic images. For the segmentation of MS images and to collect superpixel linear clustering algorithm was used. Super-pixel multiple region joint representation method was introduced to collect all spatial and environmental information of super-pixel. Superpixels were taken as basic units. To enhance classification performance of the proposed algorithm that combines detailed information and semantic information. The overall accuracy for

classification was 0.944 and the kappa coefficient was 0.92. Further, this experiment can be extended to semi-supervised and unsupervised deep learning. The processing time may increase due to the complexity of the SML-CNN model. This work will be more helpful in urban planning, environment monitoring and vegetation.

2.1.5 Fuzzy logic based landslide classification techniques

This subsection area summaries all Machine learning techniques under Fuzzy based algorithms as below:

Tao Lei et al. [34] proposed an unsupervised change detection using fuzzy c mean clustering for landslide mapping. For VHR remote sensing image change detection approach based on image segment was used for landslide mapping. Gaussian pyramid-based fast fuzzy c mean clustering algorithm is used to get better spatial information for landslide regions and for accurate landslide region difference of image structure information. Three datasets of biotemporal images of 0.5 m resolution were prepared from aerial survey system. The result was compared with existing three algorithms in terms of higher accuracy, fewer parameters, and short execution time. The proposed CDFFCM model yields 0.79, 0.80, and 0.62 accuracy for three data sets, respectively. The proposed approach work on spatial information to achieve better difference images and also has better computational time due to Gaussian pyramid method. This algorithm also reduces the sensitivity to a threshold for segmentation and requires fewer parameters. Post-event images have complex information and still, this algorithm needs to be modified for post-event images. More landslide images and ground truth are required to improve the accuracy.

D.G. Stavrakoudis et al. [35] developed a classification approach for VHR multispectral images based on a Boosted Genetic Fuzzy Classifier. The classification procedure followed two stages, one was fuzzy rule-based, which is followed by the genetic tuning stage. The fuzzy rule is useful in local feature selection and it is allowed to select the feature by repeating the Boosted Genetic algorithm. The next stage was the tuning stage used to improve the classification by using an Evolutionary Algorithm. An IKONOS satellite database with 1m spatial resolution was used for experimental results. The testing performance of BGFC is 0.8487. The main aim was to increase the overall

classification performance of the algorithm and the proposed algorithm was good in handling complex multidimensional classification.

Dinh Sinh Mai et al. [36] presented a method that combines the fuzzy probability theory and fuzzy clustering classification algorithm to overcome the disadvantages like low accuracy and instability of other satellite image classification algorithms. This proposed method initially calculates the number and coordinates of cluster-based Fuzzy probability and then for classification applies a fuzzy algorithm. Landsat 7 Satellite datasets were used for experimental results. The experimental results show that the developed fuzzy clustering algorithm gives a Classification entropy of 0.13 and a kappa coefficient of 0.9156 for one dataset and a Classification entropy of 0.14 and a kappa coefficient of 0.8599 for the second dataset. This method yields high classification accuracy on multispectral satellite images as compared to the various developed algorithms.

Long Thanh Ngo et al. [37] developed an Interval Type 2 C-mean clustering scheme for multi-spectral satellite imagery. The dataset for experimental results was taken from LANDSAT7 imagery which includes rivers, rocks, fields, jungles planted forests. To generate NDVI image of the chosen study area, two channels were used: Near Infrared and the other is visible red. NDVI is classified by IT2FCM to define different types of land covers. For some undefined pixels, the IT2FCM algorithm can handle uncertainty. Further, this algorithm can be implemented with a hyperspectral image for better results.

2.2 Deep learning techniques

Many literature surveys and comprehensive reviews on deep learning and its application applications carried out in number of researches are available [21-29]. This research discusses the challenge of high quality datasets, impact of model complexity on computational resources and limitations of model interpretability. This subsection summaries all Deep learning techniques under hybrid algorithms.

2.2.1 CNN-based landslide detection techniques

Shunping Ji et al. [38] designed an attention module to emphasize the different features of a complicated background landslide. The paper was focused on developing an accurate and time-efficient inventory based on the recognition of latent landslides. The work has been done on high-resolution optical satellite images with a CNN model to detect landladies. The attention mechanism which is based on a human visual system was developed with a number of landslides with complex backgrounds. Attention Mechanism was combined with CNN to boost the result of CNN to extract more features from the background of the landslide. Bijie landslide database with 770 images was created. Design and attention module which combine spatial and channel attention map and known 3D spatial channel attention module (3D SCAM) was designed and used in this work. The proposed 3D SCAM was trained with two-thirds of the images and results are compared with other attention modules. A few deep learning architectures such as VGGNet, ResNet, Inception, and DenseNet were evaluated with four attention modules as SE module, BAM Module, CBAM module, 3D SCAM module. Experimental result shows that ResNet50 with the proposed 3D SCAM bossed CNN provide the best result in all the combination. This research claims 0.977 accuracy, 0.97 precision. The accuracy is high with the attention module. In this attention module model can learn the characteristics of landslides very clearly.

Silvia Liberata Ullo et al. [39] presented a landslide detection method that uses Mask R-CNN with pixel-based segmentation to identify object layouts. ResNet50 and ResNet 101 was used as the backbone for the proposed method and the result was evaluated with accuracy, recall, precision and F1 score. The goal of this research was to detect landslides with pertained mask RCNN with a limited data set and Augmentation was used on the dataset to increase the volume. The dataset that was used in training and testing was created from different resources. Images collected were high-resolution digital photographs collected from UAV, search engines. Two data sets A and B were created. Data set A contains a total of 160 images. Data set B contains a total 121 images. Experimental result shows that ResNet101 have better accuracy, recall, precision and F1 score over ResNet 50. ResNet 101 yields 0.97 accuracy for dataset A and 0.90 accuracy for data set B. RestNet 101 has 1 precision, 0.93 recall, and 0.97 F1

score. The main advantage of this research was mask R CNN can provide segmentation and detection of landslide at the same time. The result shows that with the higher number of training samples accuracy was high.

Qi Zang et al. [40] undertook a theoretical comparative framework of Artificial intelligence, Machine learning, and Deep learning emphasizing their major component and learning approaches. Filippo Catani [41]. Discussed the implementation of convolutional neural networks (CNNs) for discerning mass movement patterns using transfer learning to attain higher classification than existing architectures. Gang Chang et al. [42] discussed the application of deep learning model in landslide recognition. This work emphasizes the incorporation of a transformer into ResU-Net to improve context modeling utilization of large and different sources of data for better identification. This study shows deep learning with InSAR shows promise for early landslide prediction.

Shun Yang et al. [43] present a semantic segmentation model for automatic landslide detection. Three semantic models: U-Net, DeepLab3+ and PSPNet were combined with different deep learning models (ResNet50, ResNet 101) to evaluate experimental results. Among all combinations, PSPNet with ResNet50 as the backbone network yields 0.9118 mIoU. This paper indicates high accuracy in landslide recognition but further needs to improve landslide boundary segmentation and dataset. DEM data and remote sensing data can be integrated to enhance segmentation accuracy.

Lui T et al. [44] designed a landslide detection mapping (LDM) model based on residual neural networks and Dense convolutional neural networks. ResNet and DenseNet take high spectral resolution data and conditioning factors. To create a database two cities of the chain were taken as steady objects which is China's water conservancy project. CNN, ResNet and DenseNet were trained with nineteen conditional factors and found application in the field of LDM. In all these three algorithms DensNet with remote sensing (RS) images yielded the best result. All three trained algorithms claim accuracy above 0.95 and densNet with RS images and condition factor claims 0.99 accuracy and recall and F1 score for this particular dataset. The learning efficiency of the model was enhanced with conditional factor and yielded good results in landslide identification.

Fu R et al. [45] proposed a study on post-earthquake seismic landslides. To determine the size of post-earthquake few images of post-earthquake seismic landslide satellite

imagery data were used. The database was created from Post-quake images of unmanned air vehicles (UAV) over Wenchuan country of China. The database has an average 2000 m altitude and 0.25m spatial resolution with an image size of 5616x3744 pixels. In pre-processing steps the database images were reduced in size to remove the complexity and textual information was added to these seismic landslide images. To increase the number of images in the dataset data augmentation was done with image rotation and image flip. Mask R CNN framework can scan the image and mark the region of a landslide as a target and propose a mask according to the marked region of interest. For identification and prediction of landslides in the real world requires a large number of datasets and by using transfer learning this requirement can be reduced. . Among three backbone Swis Transformer with Mask R-CNN claims 0.822 accuracy, 0.9328 precision, 0.8741 recall and 0.9025 F1 score. This model needs to improve the accuracy in the identification and prediction of landslides. Accuracy can be enhanced with the quality of post-quake landslide images. Instead of using UAV images satellite images with high spatial and spectral resolution can be used.

Ghorbanzadeh O et al. [46] presented a model that fuses object-based image analysis (OBIA) with a Fully convolution network. ResUnet as the predominate FCN model was trained and tested with the Sentinel-2 database and designed a combination of FCN-OBIA segmentation and classification using knowledge-based rules. In OBIA image difference indices were calculated between pre and post landslide. The data set used was created from Sentinel-2 images of Eastern Iburi Japan. The experimental results show that ResU-Net yield 50.24 mIOu, 0.7615 precision, 0.6001 recall and 0.6662 F1 score whereas ResU Net OBIA yields 72.49 mIOu, 0.855 precision, 0.826 recall and 0.8403 F1 score. In this work, it is observed that the ResU-Net model detects landslides correctly but has high false positive results. The result was more accurate and had fewer false positives by adding rule-based OBIA for a landslide to train ResU-Net.

2.3 Other classification techniques

This subsection area summaries all Machine learning techniques under hybrid algorithms as below:

Tapas R. Martha et al. [47] presented an algorithm uses spectral, contextual and shape information of images for landslide detection. For object-oriented analysis, multispectral images were segmented and objects collected from these images were used as a classifying unit. The main objective was to correctly identify the landslide using OOA. Complex landslides were difficult to segment because of different characteristics, like low contrast and overlapping shadows. To identify the false positive landslides, shape and morphological information were combined. A landslide is categorized by the base material and movement of flow. To identify landslide Resources at 1 and LISS IV multispectral data sets were used. For testing the algorithm, images of the area in Himalayas in India were selected and test the algorithm with 5.8 m MS data from Resources at 1 and 2.5 m Cartosat1. 0.764 recognition is possible with the proposed algorithm and classification accuracy is 0.691. This algorithm yields more efficient and accurate landslide detection by utilizing object-based classification. The main challenge in the work is to distinguish landslides from other objects with similar spectral properties like soil and water.

Thomas Blaschke et al. [48] used a semi-automated object-based image analysis methodology to detect landslides. Object-based image analysis has gained an important role in remote sensing. IRS-ID and SPOT 5 satellite image database were used for the detection process. Digital Elevation and gray-level co-occurrence matrices were used to collect slope and flow directions. In object-based image analysis, multi multi-resolution segmentation was applied for selecting the feature and classifying the object. The segmented object was processed with their spatial, spectral, and textural parameters. The landslide class was defined on the base of its morphological characteristics. The inventory database of 109 landslide events was used as proof to validate the results and according to rule-based classification, the area above 1600m (about 5249.34 ft) with a slope greater than 7% is considered landslide landslide-affected area. The brightness threshold is set for a database created from IRS-ID and SPOT 5 satellite images. The combinations of these parameters indicated that an overall

accuracy of 0.9307 was achieved for landslide detection. This method will be useful to detect landslides even without proper landslide inventory.

Sansar Raj Meena [49] used U-Net and machine learning approaches for automatic detection of landslides by landslide event-based inventory of triggering events and occurrence landslides. The major issue lies in mapping performances among interpretations in the event-based inventory. In this research, two datasets: Dataset 1 from RapidEye satellite imagery and Dataset 2 combine RapidEye and ALOS-PALSAR. 239 data samples were used to evaluate the model. Experiments were performed over a fully convolutional U-Net, Support Vector Machine, K-nearest neighbour, Random Forest. Among all machine learning techniques, U-Net performs best result of 76.59%MCC. The performance of the U-Net model further can be increased by increasing the sample size for training samples.

Haojie Wang et al. [50] presented a 11-layer deep convolutional neural network (DCNN-11) model for landslide identification using ML& deep learning. Promising results from a case study of Hongkong City were achieved on three databases: Recent Landslide Database (RecLD), Relict Landslide, Database (RelLD) and Joint Landslide Database (JLD). Experimental result reveals that DCNN-11 is very effective model among Support Vector Machines, Random Forest and logistic regression. DCNN-11 has the highest area accuracy 0.925 with the RecLD database. Further, it is observed that the performance of DCNN can be improved by considering the inconsistency in terrain, landslide, inaccuracy in the database and the necessity for more complicated CNN's in the future owing to computational restrictions. Soumik Saha et al. [51] also investigate landslide susceptibility in the Garhwal Himalaya using machine learning models, with Deep learning neural network (DLNN) demonstrating good accuracy. Omid Ghorbanzadeh et al. [52] compare Artificial neural networks, Support vector machine, and Random forest Convolutional neural network for landslide detection. Optical data from the Rapid Eye satellite were used for experimental results. CNNs are used for effective feature representation in image recognition and have better accuracy for small window size.

Mahnaz Naemitabar et al. [53] undertook a comparative study on four machine learning techniques: Support vector machine, the boosted regression trees (BRT) model, the

Random Forest (RF), and a Logistic Model Tree (LMT) for the identification of landslide-prone areas. The SVM and RF yield higher reliability in assessing landslide susceptibility, with factors like lithology, slope, and land use identified as crucial. Experimental results show that SVM and RF models have AUC 0.86 and 0.89 respectively.

Jie Dou et al. [54] presented an automatic method for landslide detection. This approach combines three different approaches namely Genetic algorithm, object-oriented analysis, and case-based reasoning. In object-based analysis, segmentation plays a very important role. High resolution of the image provides correct information about the landslide and was helpful in the better result of the segmentation process. To obtain the object of interest in object-oriented analysis multi-segmentation was preferred on collected images. The genetic algorithm was applied for the feature selection. Geographical features classify and enhance the accuracy with case-based reasoning. The case-based reasoning is achieved with different techniques like k nearest neighbor etc. In this paper, Quickbird images of 0.6 m spatial resolution were used for image segmentation and feature selection. Roadside landslides were more exposed to high damage due to landslides and caused difficulties in day-to-day life. SPOT 5 and DEM datasets were also used for experimental results. All data were rectified to remove the distortion and noise. Object-oriented image analysis gives 0.75 accuracy for the detection of landslides and fused Object-oriented image analysis with a case-based reasoning and genetic algorithm (GA) yields 0.87 accuracy in the detection of landslides. The proposed technique provides benefits over a knowledge-based section for the detection of landslides. This technique helps in creating inventory that will be helpful for providing specifications for future landslides.

Tapas R. Martha et al. [55] designed a new approach to detect landslides using bitemporal multispectral images. Multispectral images were used to collect the object from post-landslide images. For the analysis of high-resolution images, a tool is developed which makes input data in a user-defined grid. Multispectral images were collected from the Resourcesat-2 LISS-IV satellite for a defined study area. These two datasets have three bands and are useful for object-based change detection techniques to recognize landslides. For the detection of landslides 10m DEM from Cartosat1 satellite data was used. For good quality images auto-rectified Resourcesat-2 LISS-IV

satellite images are further processed to achieve high pixel match. Top atmospheric reflectance calculations were performed in the preprocessing step of images to overcome weather conditions like sunlight. Pre and post-landslide image reflectance differences identify the landslide. Image segmentation was performed with knowledge-based approach. Object-based change detection was used to detect landslides. The developed graphic user interface (GUI) tool provides overall good accuracy in landslide detection. Combined spectral and morphometric parameters have 0.89 accuracy in the detection of landslides with 10m DEM from Cartosat-1 satellite images. This work can further be modified for the shadow of clouds in pre-landslide images. Some landslides were not identified due to small clouds over the pre-landslide images.

Tapas R. Martha et al. [14] presented a comparison of the pixel-based approach and object-oriented approach for landslide detection. Very high resolutions of 0.5m remotely sensed images were used. An inventory was created with 115 field-based landslides fused with 0.5m spatial resolution for comparative analysis. Unsupervised classification was used in pixel base classification and images were classified in eleven different classes. For non-landslide and landslide pixel binary analysis was used and assigned zero and unity for landslide and non-landslide, respectively. In object-oriented analysis k mean clustering was used to remove regions based on brightness to detect landslides and object properties were used to reduce false positive results. Object-oriented analysis has 0.965 and Pixel-based unsupervised classification has 0.943 accuracy. In this paper, further investigation on challenges associated with OOA needs to be discussed for improvement.

Filippo Vecchiotti et al. [56] presented a semi-automatic image classification technique for landslides caused by rainfall. This approach combines pixel-based classification with remotely sensed images multi multi-parameters for landslide detection. Vegetation change in pre and post-image will identify the landslide event. In the method, bitemporal pixel change detection was applied. It was a double classification technique. Terra ASTER L2 data sets were used for the defined study area. 110 landslides which were recognized accurately with this semiautomatic image classification technique. This double classification workflow gives 0.815 producer accuracy coupled with a more than acceptable 0.689 accuracy and 0.729 kappa coefficient. With its data set

cloud detection was not performed but it can be overcome with SPOT and Sentinel-2 for a better view of scenes.

2.4 Findings and research gap

The primary tendency of this literature survey to observe the performance of classification on landslide detection. In this literature survey, the analysis is based on the accuracy of the classification, satellite-based datasets used for detection and algorithms used for classification. During this study, we found different observations that will be helpful for future research directions, concluded in this section.

2.4.1 Challenges which are discussed

Numerous researches have been done with semi-automatic classification of landslides and a few with automatic detection. The most important stage is the data collection. The high-quality image provides a better result for feature extraction. The review concludes that two different images are fused to give a better feature selection. Earth-observing satellites are two types: active satellites and passive satellites [16]. Satellite data provides images and features are extracted based on the following points:

- Active satellites are microwave remote sensing and have their own source of energy. Active satellites have controlled illumination and have the least effect of weather.
- In Active satellite Day and night operations are possible. ESA satellite, Canada RadarSat, Indian satellite (RISAT) and Japanese satellite (ALSO) is a type of active satellite. ESA's Sentinel-1 is an active microwave remote sensor and is useful in providing data for all types of disasters like floods, earthquakes, and landslides.
- Passive remote sensing is more useful nowadays and does not assign any external source of energy. These types of satellites measure either reflected radiation from the sun or emitted radiation from the earth. Reflected radiation depends on sunlight so it works on the daytime only and suffers various illumination conditions like weather play a major role. LANDSAT, SPOT, IRS, Cartosat, and IKONOS are some examples of passive satellites.

- To image segmentation, compare object-based and pixel-based image classification.
- To classify multispectral images and multi-frequency images.

2.4.2 Challenges which are not discussed

There are few challenges which are not discussed:

- Real-time remote sensing data: Automatic detection of landslide require real-time data which based on satellite imagery need highly efficient algorithms for real-time computation.
- Limited training data set: Acquiring large-scale data sets for training the model in a machine learning algorithm is a challenging task, which can hinder the ability of the trained model to detect landslides accurately.
- Feature extraction: for automatic detection and classification of landslides, need to extract the feature from the database and the feature should be relevant so that model can accurately differentiate between landslide and non-landslide images.
- Environmental factor: landslide depends on various environmental factors like rainfall, soil type, topography etc. By considering these parameters machine learning model requires a more careful feature selection algorithm.

2.4.3 Observations

This research work covers a review of 50 research papers, out of which 70% papers used passive sensor-based satellite databases for training and testing, 22% of papers used active sensor-based satellite databases while 8% of papers used aerial images for experimental results. The different types of satellites used in different research work as shown in Figure 2.1

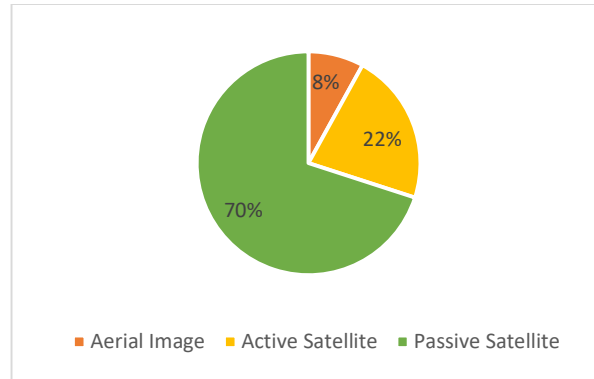


Figure 2.1 Classification of images used for Landslide Monitoring

Analysis carried out on the basis of accuracy shows that an accuracy range between 95%-100% is obtained in three research papers, 90%-94% is obtained in seven research papers, 85%-89% is obtained in seven research papers, 80%-84% is obtained in four research papers and below 80% is obtained in two research papers as shown in Figure 2.2.

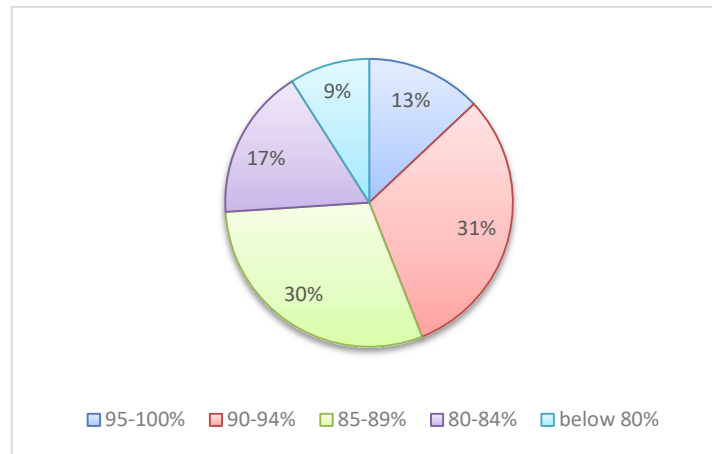


Figure 2.2 Classification Accuracy of ML/DL based techniques

Classification algorithms are categorized into seven different classes. This work reviewed fifty research papers, six papers are based on SVM classifier, four papers are based on Bayesian classifier, three papers are based on a decision tree classifier, three papers are based on neural networks, four papers are based on fuzzy, fourteen papers are based on deep learning technique and the remaining are hybrid algorithms that combine different classifier algorithms. Figure 2.3 shows the different algorithm-based research papers.

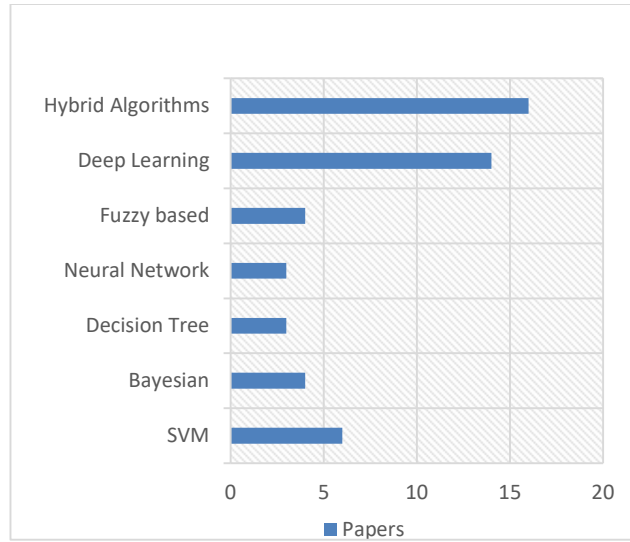


Figure 2.3 Number of classifiers and types used in literature review

2.4.4 Research gap

A landslide is a real-time event and it is very difficult to create a database with a very high number of images. Landslide events depend on a number of factors, like rain, soil and weather conditions. Need to design a model with a Convolutional neural network (CNN) that takes care of different factors of landslide. Deep neural networks are difficult to understand and with a complex architecture of networks, prediction is very difficult. CNN architecture has practical applications in image classification, object detection, and semantic segmentation [38]. Semantic segmentation with pixel-level segmentation provides good results in image segmentation [43].

The articles in this study have very effective results in terms of accuracy, precision, recall and F1 score for landslide detection. There are few potential research gaps as follows:

- **Performance:** Accuracy plays the main role in the performance of an automatic detection landslide model. Selected literature articles were compared with common parameters: accuracy, precision, recall and F1 score. The range of accuracy lies between 0.80 to 0.92. Semantic segmentation model PSPNet with ResNet50 as backbone achieves the highest mIOU at 91.18% , this can be

further improved by exploring false results [43]. The Swis Transformer a backbone network with Mask R-CNN claims 0.822 accuracy [45].

- Method used: Convolutional neural network architectures such as ResNet50, ResNet101, VGG, DensNet, Google Net were used as the backbone with different approaches and provide different results. CNN along with spatial channel attention mechanism and high-resolution optical images yield high accuracy. Need more investigation on the superiority of spatial channel attention mechanism [38].
- Database: Need is a high-resolution remote sensing image dataset for improving accuracy. Models were trained for selected datasets. The Bijie landslide dataset was used, but it is limited to 770 images of landslide and nonlandslide images. Some dataset was created from images of unmanned air vehicles, which contain different images from different sources. Instead of using UAV images, satellite images with high spatial and spectral resolution can be used and will provide good results [39]. Training is limited with the dataset is the biggest research gap.
- Different type of landslide has different characteristics. The model based on Mask R-CNN with pixel-based segmentation and ResNet at the backbone does not specify whether the model tested for different types of landslide. Results are only compared in between ResNet 50 and ResNet 101 rather than the other potential object detector deep learning architecture [39].

2.5 Problem Statement

Landslides are a severe environmental threat, especially in hilly areas. Leveraging satellite data for landslide detection and prediction provides considerable potential, but raw satellite data frequently contains discrepancies due to factors such as sensor noise, cloud cover, changing atmospheric conditions, and topographical distortions. The goal is to create an algorithm for pre-processing landslide data derived from satellite images. For automatic detection need to design and develop a novel machine-learning/deep-learning algorithm and optimize the detection and classification model towards better performance.

2.6 Objectives

The objective of this research work is to design an automatic CAD system for Detection and Classification of Landslide using Artificial Intelligence Techniques. This research work contains the following objectives:

- 2 To design an algorithm for pre-processing the landslide data generated by satellite data.
- 3 To design and develop a novel machine-learning/deep-learning algorithm to detection and classify landslide.
- 4 To optimize the detection and classification model towards better performance.

Chapter 3 Pre-processing of satellite images

3.1 Bijie landslide dataset

Deep learning architectures require a large number of images to train the network and creating a dataset for landslides from remote sensing images is a difficult task. The Bijie landslide dataset is an open-source dataset that was used in this research to train the deep learning architecture. Ji et. al [38] developed the Bijie dataset and the study area is over Bijie city of China about 26853 square km area. Due to the fragile nature of soil and unstable geological conditions make this area highly prone to landslides. The RGB images of 0.8 resolution were captured by the TripleSat satellite over an area located in the transitional slope zone from the Tibet Plateau to the eastern hills with altitude ranges from 457 to 2900 m. This dataset contains two Groups of images: landslide images and non-landslide images. The landslide set of images contains seven hundred seventy images and the Non-Landslide set contains two thousand three images. In our experiment, 70% images from the Bijie dataset are used for training and 30% of images are used for testing the model. The dataset was split in a stratified manner, ensuring that both landslide and non-landslide images were included in the training (70%) and testing (30%) sets. The training set contains 70% landslide and 70% non-landslide images, while the testing set contains 30% landslide and 30% non-landslide images. Figure 3.1 shows some images of landslides in our training set.

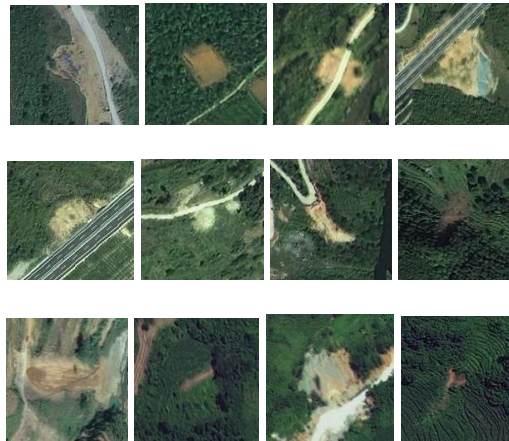


Figure 3.1 Examples of landslide instances from Bijie dataset

3.2 Satellite image Pre processing

The most important task nowadays is to save lives from disasters. During these disasters to save human life is a difficult task for the disaster management team. Satellite images processed with artificial intelligence algorithms are used to save lives during disasters. The contemporary era's remarkable technological advancements make satellite imagery a vital resource for geospatial applications. Analyzing high-resolution satellite images over time using object recognition and image categorization has proven invaluable in areas such as disaster management, agriculture, urban planning, landslide prediction, atmospheric prediction, environmental surveillance, mitigating natural catastrophes and assessing land-based biodiversity [57]. Remote sensing is a method used by satellite imagers to capture data on Earth from orbit. Earth observation satellites are equipped with different sensors like LISS3, Worldview, MODIS, Landsat, Sentinel-2, VIIRS, AVHRR, etc. detect electromagnetic radiation from Earth's surface. Different objects observe and reflect different wavelengths, ranging from radio waves to gamma rays. Information is converted into radio waves and transmitted to base computers which transmit into 0 and 1 binary code and create images from the received information. Sensors are classified into two categories: active and passive sensors. The active sensor provides its energy source for illumination and does not depend on sunlight. These types of sensors can be used for examining wavelengths that are not provided by sunlight, such as microwaves. Synthetic Aperture Radar is an example of an active sensor. Passive sensors measure naturally available energy [16]. Satellite images are quite complex to analyze and it is very important to develop a reliable and accurate system to analyze satellite images. The true difficulty lies in deciphering remote sensing data to obtain precise information. Utilizing this data for automatic decision-making in detection presents challenges. Raw satellite images have many challenges and these challenges arise from the inherent complexities of acquiring and interpreting data from space, where a variety of factors can introduce errors, inconsistencies and distortions. When using satellite data for prediction and classification, the accuracy of any application is reduced by the numerous errors included in raw satellite images. The preprocessing stage uses radiometric correction, atmospheric correction, dark object subtraction, noise removal, random bad pixels,

image enhancement, resizing, augmentation, etc. [58][59]. Understanding these obstacles and errors, as well as the need for preprocessing, is critical for obtaining accurate and dependable results from satellite image analysis.

3.2.1 The objective of the work

The objectives of the work undertaken are as follows

1. To explore different types of systematic and non-systematic errors in raw satellite imagery.
2. Apply different correction techniques on raw satellite landslide images and discuss the results.
3. To analyse and evaluate the results of the proposed pre-processing.

3.2.2 Related work

In the last few years, a number of research studies have taken place with satellite image processing to identify, predict, classify, and detect any real-time event [7][9][60][61]. Satellite image processing has numerous challenges due to the intrinsic challenges of receiving and understanding data from space, where many circumstances might create inaccuracies, inconsistencies, and distortions. There are two main errors in raw satellite imagery: systematic and non-systematic. Systematic errors in satellite images are constant, repeated imperfections caused by several predictable factors relating to the satellite sensor, its orbit, and the environment in which it functions. Geometric distortion, atmospheric distortion, sensor noise, temporal mismatch and orbital error are some systematic errors. Non-systematic errors also called random errors, vary in magnitude and in direction within the same image or different images. Processing of satellite images uses Geometric correction, atmospheric correction, radiometric correction and image enhancement [58][59].

Geometric correction corrects the misleading geographical errors that are present due to errors by Earth's curvature and sensor orientation. James Storey [62] emphasizes the significance of geometric correction as an essential pre-processing step for matching multi-sensor satellite imagery. The study found misregistration difficulties between Landsat-8 and Sentinel-2 imagery, including misalignments of up to 38 m (2σ) due to

discrepancies in geometric control references. It is observed that misalignment varies geographically, but it remains consistent for specific areas. James Storey [63] explains the geometric correction process for Landsat 8 which involves refining the sensor's line-of-sight model through on-orbit calibration activities. The results show excellent geometric performance. Prieto-Amparan et al. [64] compared three radiometric correction methods Atmospheric Correction for Flat Terrain 2, Fast Line-of-Sight analysis of Spectral Hypercubes and Dark Object subtract 1 for estimating grassland biomass using Landsat imagery. The simplest method DOS1, provided the most accurate and consistent result [65]. Atmospheric Correction corrects the surface features that are distorted due to the effect of atmospheric particles and gases. Atmospheric correction transforms at-sensor radiance to surface reflectance using the ATREM algorithm. The real-time results were similar to the standard AVIRIS-NG ground processing, but much faster [66][67]. Radiometric correction was used to convert digital numbers to spectral radiance and reflection. A radiometric correction was done on the Landsat image to improve the quality of the satellite image [68][69]. Image enhancement is a crucial step in image preprocessing and is used to improve the visual quality of an image by emphasizing some specific feature of an image. This process makes images more useful and accurate result-oriented for specific tasks like feature extraction. Anuj Ashokan et al. [70] provide a comprehensive review of image enhancement techniques and categories in two domain methods: spatial and frequency and claim that histogram equalization enhances the overall quality of an image. Shilpa suresh et al. [71] proposed an algorithm for image enhancement by modifying the differential evolution algorithm and combined with the Cuckoo Search. Quality speed distribution and proportionality-spaced distribution are two types of distribution techniques that were used. Yun B et. al. [72] introduce a saliency enhancement method using Fast, Accurate, and Size-Aware to make landslides more prominent in large-scale Landsat8 images.

3.2.3 Methodology For Satellite Image Pre-processing

Satellite sensors' raw images frequently show distortions and irregularities resulting from a variety of causes, including air conditions, absorption, changing scene lighting, satellite orbital motion, and geometric aberrations. Errors in satellite imaging can occur

from several different causes during the collection, transfer, processing, and interpretation of the images. It is essential to comprehend these mistakes and use the proper repair methods to obtain accurate analysis and trustworthy outcomes. This is a thorough analysis of typical mistakes in remote-sensing satellite images. Different errors present in satellite images are shown in Figure 3.2.

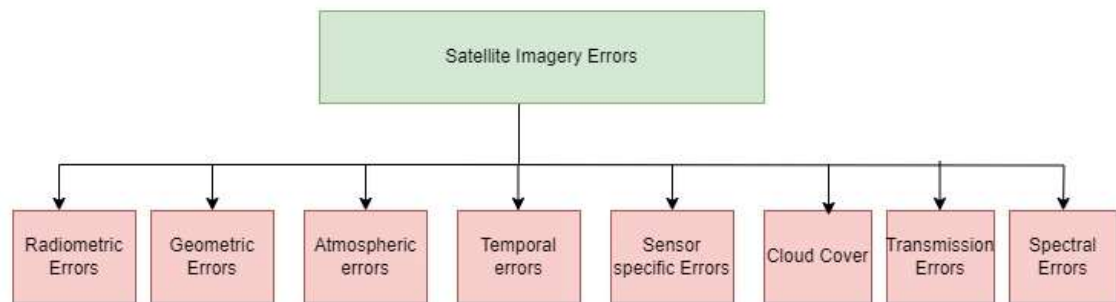


Figure 3.2 Satellite imagery errors

- **Radiometric Error:** Pixel values in an image are impacted by radiometric error, which is usually caused by changes in sensor sensitivity, atmospheric conditions, or problems with equipment calibration. A variety of errors, including noise, banding, and atmospheric distortion, can be found in radiometric errors. Noise occurs when pixel values randomly vary and are unrelated to the desired object. Bands across the photos are caused by uneven sensor performance. Scattering and absorption are the causes of atmospheric distortion. To overcome radiometric, a variety of correction techniques are employed, including noise filtering, calibration, destriping algorithm, and dark object subtraction [73].
- **Geometric Error:** When changes in satellite orbit, sensor geometry, or terrain relief affect the spatial relationship between objects in a image, is known as geometric errors. Geometric errors come in various forms, including perspective, relation, and terrain distortion. Sensor misalignment is the cause of relation inaccuracy. The cause of perspective mistakes is the way the angle of the satellite sensor causes things in higher-elevation areas to seem displaced. Surface feature distortion can result from changes in the terrain. Techniques like as georeferencing, orthorectification, and resampling are employed to mitigate the impact of geometric inaccuracy. Using ground control points, the image is aligned with the coordinate system in georeferencing. Digital

elevation models are used in orthorectification procedures. Bilinear covariance and nearest neighbor are two resampling techniques that are used to manipulate the image with reference coordinates.

- **Atmospheric Errors:** These errors are caused by surface features changing in appearance due to electromagnetic radiation's interaction with gasses and particles in the atmosphere. The image quality is diminished by haze and smog as a result of aerosol scattering. Additionally absorption of specific wavelengths by water vapor, causes spectral aberrations. Atmospheric errors are corrected using radiative transfer models, image-based correction methods like empirical line calibration [66].
- **Temporal Error:** Images taken at different times can exhibit variances due to environmental factors such as seasonal changes, vegetation growth, or urban development, which can lead to temporal errors. Image normalization helps to eliminate variations resulting from acquisition time variances by matching the brightness and contrast of several photos to a reference image. Change Detection Analysis is also used to overcome temporal error. Rather of trying to make direct corrections, temporal errors can be reduced for applications such as land cover change by monitoring and analyzing changes.
- **Sensor-specific error:** Sensor-specific errors are the results of certain satellite sensors producing particular artifacts as a result of their operational parameters and design. Dead pixel and lens distortion are an example of sesor specific errors. In order to correct for these inaccuracies, post-processing adjustments are employed to rectify lens distortion using models based on known sensor characteristics. Image interpolation is utilized to fill in the gap left by dead pixels.
- **Cloud cover:** Clouds have the ability to mask surface characteristics, making parts of satellite imagery useless for analysis. Cloud masking, data fusion and cloud gap filling is used to overcome this error.
- **Transmission error:** Data corruption or missingness due to errors that happen during the transfer of data from the satellite to ground stations is known as transmission error. These types of errors can be overcome by using data reconstruction algorithms and error correction coding.

- **Spectral error:** Spectral error is errors pertaining to the way a sensor records light at various wavelengths, which could lead to an erroneous depiction of surface details. A band alignment algorithm is used to overcome these errors.

Raw satellite data is complex and needs pre-processing to remove the errors. Preprocessing is an important step in satellite image analysis since it prepares the raw data for later processing and interpretation. The primary goals of preprocessing are to rectify distortions, improve image quality, and prepare data for geographic analytics. Remotely sensed digital images are composed of pixels and have a digital number (DN). Each pixel has intensity and address value. Pixel is square and these squares represent an aerial average of the ground. Given the imaging geometry and satellite orbit parameter, the row and column indices of each pixel on the ground must yield its exact graphical location. The resolution of an image plays a very important role in image analysis. Satellite images have five types of resolution: 1. Spatial resolution, 2. Spectral resolution, 3. Temporal resolution, 4. Radiometric resolution, 5. Geometric resolution. Once an image is acquired the spatial resolution is frozen. Raw images captured by satellite sensors often exhibit distortions and inconsistencies caused by various factors, such as atmospheric conditions, absorption, varying scene illumination, satellite orbital motion, and geometric distortions. Systematic and non-systematic error present in satellite images degrades the result in image analysis. Preprocessing is necessary since raw satellite data has inherent problems and limits. To produce an accurate depiction of the actual image, distorted or damaged image data must be corrected using different rectification processes. For accurate and insightful analysis, especially in areas like land-use planning, disaster relief, and environmental monitoring, satellite picture defects must be corrected. The kind of error, the properties of the sensor, and the intended use all influence the correction technique selection. To provide accurate, dependable findings, it is frequently essential to combine many techniques, such as atmospheric correction with geometric and radiometric corrections. To eliminate inaccuracies, pre-processing is necessary for the complex raw satellite data. Since preprocessing sets up the raw data for further processing and interpretation. Resolving distortions, enhancing image quality, and getting data ready for geospatial analytics are the main objectives of preprocessing. Preparing imagery from satellites

for analysis requires a certain level of preprocessing. In order to make the raw data from the satellite sensors more appropriate for later interpretation and modeling, it entails improving and correcting it. Different preprocessing techniques are shown in Figure 3.3.

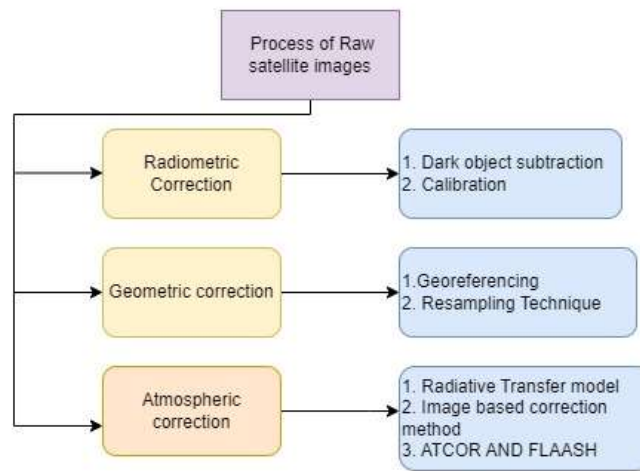


Figure 3.3 Rectification Processes

Radiometric correction: Pixel intensity and radiometric correction are connected. Variations in pixel intensity values are brought on by sensor defects, variations in the sensor's sensitivity, or external circumstances like the illumination during the picture capture. It adjusts the sensor's digital numbers (DNs) to take into account its special characteristics, such as sensitivity and potential deterioration over time. eliminates the influence of the atmosphere and guarantees that, regardless of atmospheric circumstances, the data more accurately depicts the surface reflectance or radiance. Raw satellite photos contain a variety of radiometric aberrations, such as banding, line loss, dropout of a line or column, random poor pixels, and stripe noise etc [74].

Geometric correction: Earth curvature, sensor viewpoint, and satellite motion all cause distortion in raw satellite data that is corrected by geometric correction. Geometric correction seeks to ensure that the satellite image accurately represents the forms and geographic locations of features on the Earth's surface, allowing for precise spatial analysis and mapping [75]. Among the geometric correction algorithms are

ground control points, nearest neighbor, bilinear interpolation, and mathematical transformation resampling.

Atmospheric correction: The effects of the atmosphere, such as light absorption, path radiation, and scattering, are removed from the raw satellite photos using atmospheric correction. Various techniques are employed for atmospheric correction, including Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes, and Atmospheric and Topographic Correction (ATCOR) [64].

3.2.4 Radiometric correction and Geometric correction

From the study area select one raw satellite image and perform geometric radiometric and atmospheric correction. A digital number (DN) represents the brightness value of each pixel in a satellite image, which is obtained from reflected or emitted energy. It is the sensor's raw digital output that is utilized to adjust radiometric and atmospheric data. All preprocessing stages rely on DN values, which are the raw data output by satellite sensors. They are used to quantify physical parameters like radiance and reflectance using radiometric correction, allowing for accurate analysis. These values are significant for several remote sensing applications, including land cover classification, change detection, and thematic mapping, as they allow quantitative interpretation of surface properties. Initially set constants Multiplicative rescaling factor, additive rescaling factor and sun elevation. Then define the coordinate reference system for radiometric correction. Extract the digital number value of the image and calculate transformation parameters. Rejoin DN band to target coordinate reference system. Then apply atmospheric correction. Atmospheric correction modifies the DN and this DN is converted into Top of Atmospheric (TOP) reflectance using radiometric correction.

DN is converted into Top of Atmospheric is done using linear transformation equation as shown below :

$$L_{\lambda} = \text{Gain} \times \text{DN} + \text{Offset} \quad \text{-----(1)}$$

Where:

- L_{λ} is the TOA radiance (in W/m²/sr/μm)

- Gain and Offset are radiometric calibration coefficients specific to the satellite sensor (usually provided in metadata)
- DN is the raw digital number value of a pixel

This process adjusts for sensor calibration and translates raw DN into radiance units, suitable for further atmospheric correction and surface reflectance analysis.

Figure 3.4 shows the processed image. Figure (a) shows Geometrically corrected image and Figure (b) shows Radiometrically and Atmospherically corrected image. Figure 3.5 shows the histogram result of the pre-processed image. Table 2 shows the parameters of the processed image.

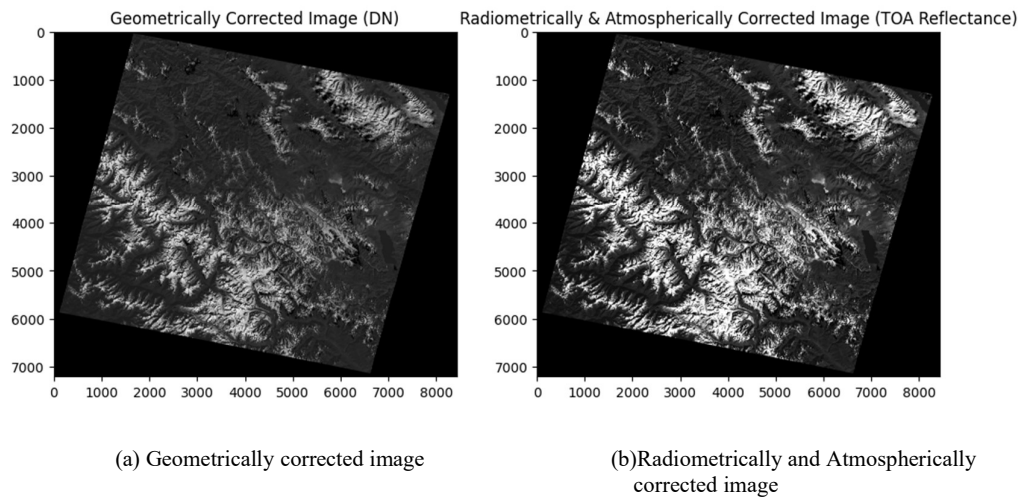
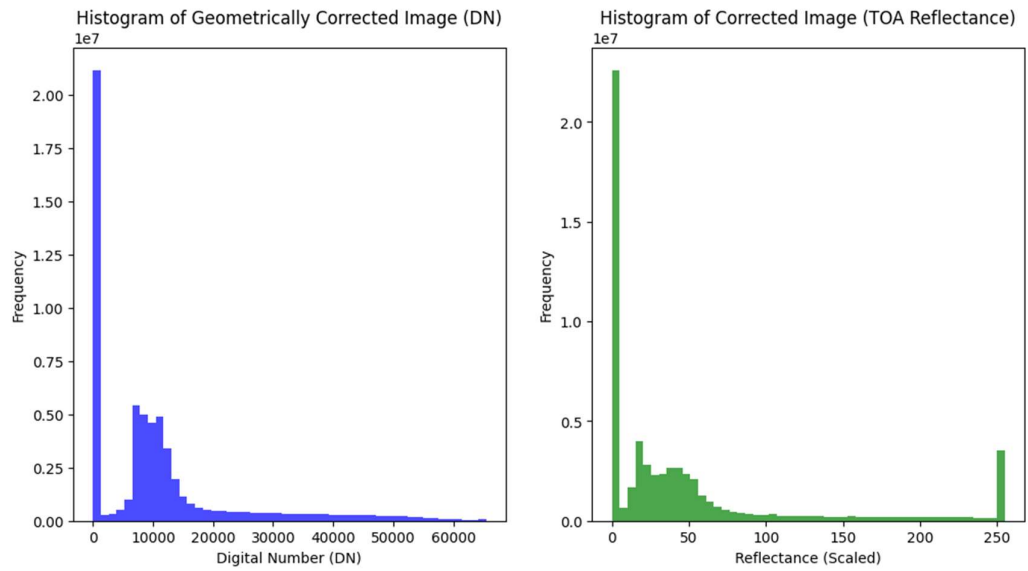


Figure 3.4 Geometrically Radiometrically and Atmospherically corrected image



(a) : Histograms of DN values

(b): Histograms of TOA Reflectance

Figure 3.5 Histograms of TOA Reflectance and Histograms of DN values

Table 3.1 Pre-processed image parameters

Geometrically Corrected Image (DN)	Min: 0, Max: 65455
	Mean: 10654.471086060137
	Std Dev: 12895.613843682024
Corrected Image (TOA Reflectance)	Min: 0, Max: 255
	Mean: 49.59613890229188
	Std Dev: 71.85929504994736

3.2.5 Algorithm of Proposed Work

The preprocessing workflow contains several steps including radiometric correction, dark object subtraction, random bad pixel and Gaussian smoothing. The flow diagram of the proposed work is shown in Figure 3.6.

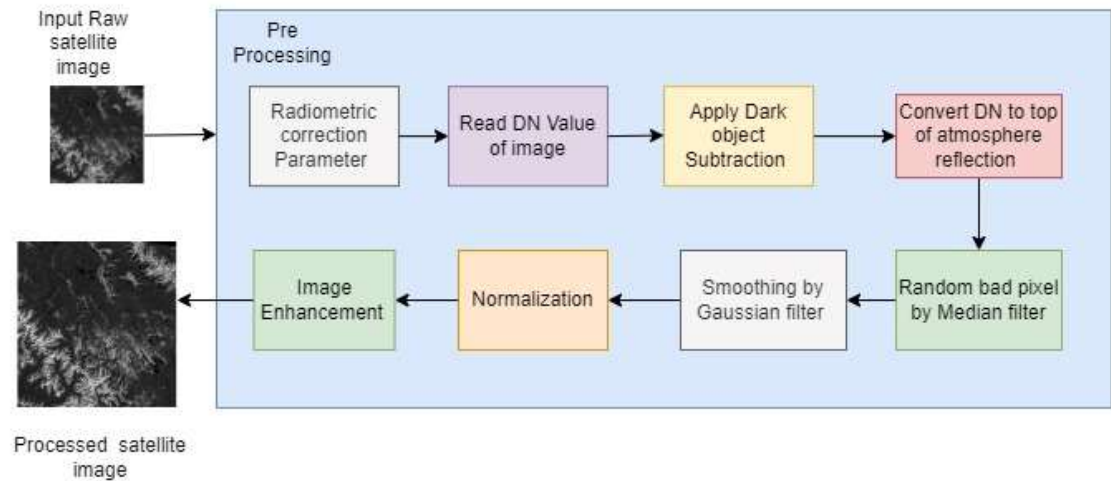


Figure 3.6 Flow Diagram of Proposed Pre-Process algorithm for Satellite images

Algorithm of Proposed Work

Step 1: Download the satellite image for the study area. And input the satellite image file.

Step 2: Set Radiometric Parameters: Sun elevation, additive rescaling, Multiplicative rescaling correction

Step3: Read the digital number (DN) value.

Step 4: Perform atmospheric correction using Dark object subtraction (DOS)- identify minimum DN value and subtract dark object value from DN.

Step 5 : Perform radiometric correction – convert atmospherically corrected DN to Top of atmospheric (TOA). Scale TOA reflected value for visualization.

$$\text{Reflectance} = \text{DN} \times \text{Multiplicative rescaling factor} \quad \text{-----}(2)$$

Step 6: Apply a Median filter to correct bad random pixels.

Step 7: Gaussian filter is used to smoothing of image.

Step 8: In normalization, image is clipped to range [0,1] and scale the clipped reflectance to [0,255]

Step 9: Display the result using a Histogram

Step 10: Convert DN to quantitative Metrics to compute : Peak Signal to Noise Ratio (PSNR), Root Mean Square error ,Mean absolute error and Structural Similarity Index.

$$PSNR = 10 \log_{10} \left(\frac{(Maximum\ pixel\ value)^2}{MSE} \right) \text{-----}(3)$$

$$RMSE = \left(\frac{1}{N} \sum_{i=1}^N (DN\ in\ 8bit - normalized\ reflection)^2 \right)^{1/2} \text{-----}(4)$$

3.2.6 Results and discussion

In this experiment, we perform satellite images processing to improve the quality of the image and prepare dataset for landside analysis. This proposed preprocessing workflow contain few key steps: bad pixel correction, radiometric correction, darkobject subtraction, normalization, gaussian smoothing, resizing and augmentation. The result of this experiment is evaluated based on histogram comparison and qualitative metrics.

First step start with reading the Digital number of satellite image from study area. We set the radiometric correction constants: sun elevation, additive rescaling factor and multiplicative rescaling factor. To reduce the impact or atmospheric scattering, apply Dark object subtraction, find the minimum value of digital number band and subtract from all the present pixel in the band to correct the band. Now, this corrected band DN number is converted to top-of-atmosphere reflectance.

To correct random bad pixel, a median filter is applied atmospheric reflectance image. To reduce noise gaussian smoothing was applied. The resulting images are scaled in the range [0, 255].

Figure 3.7 (a) and (b) shows original DN image and shows proposed pre-processing algorithm processed image. Figure 3.8 shows histogram analysis of pixel distribution original image and processed image.

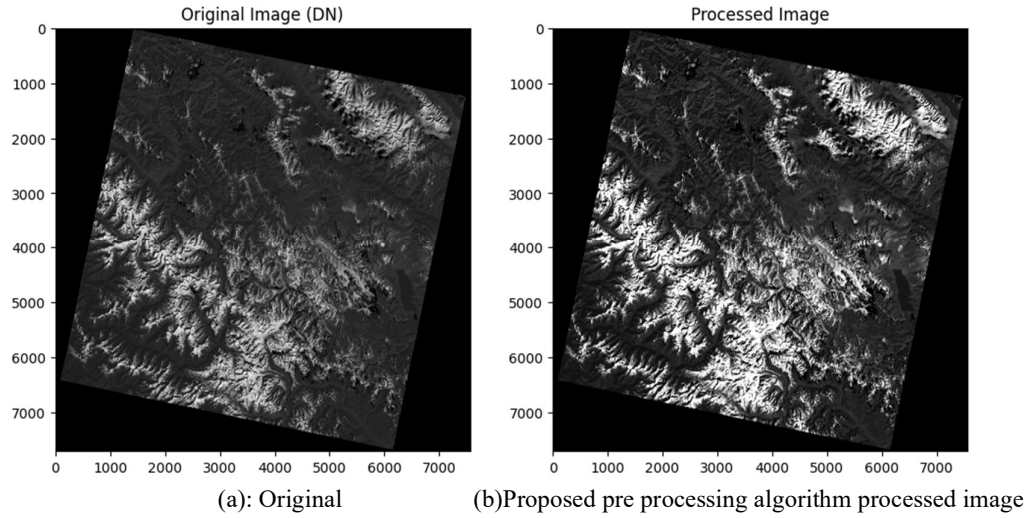


Figure 3.7 Pre-Processed satellite image

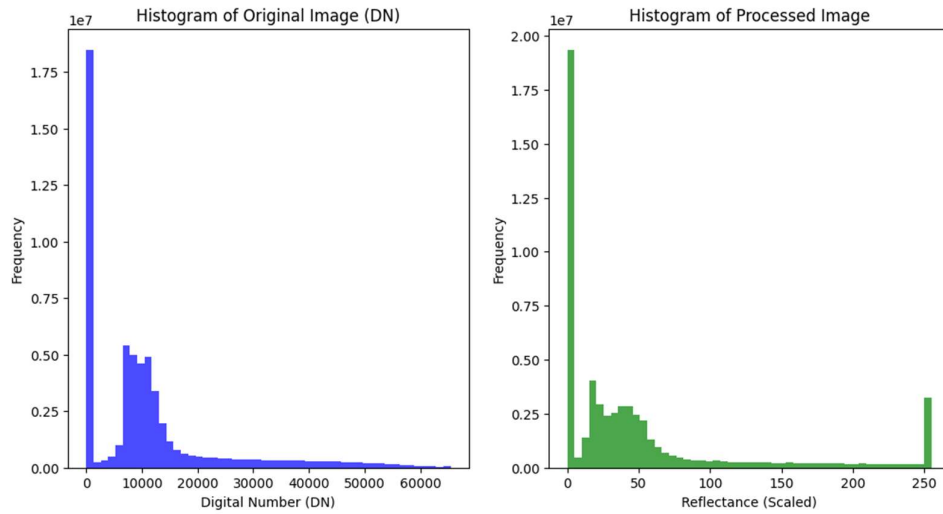


Figure 3.8 Histogram analysis of pixel distribution original image and processed image.

Table 3.2 summarize the quantitative analysis of proposed algorithm. Processed image have Mean value 52.05141, standard deviation 71.41093. To check the effect of pre processed image we calculate quantitative metrics: mean absolute error is 82.42, RMSE is 7.188, SSIM is 0.87 and Peak signal to noise ratio is 19.4512.

Table 3.2 Proposed algorithm Pre-processed image parameters

Parameters	value
DN	Min: 0, Max: 255, Mean: 52.05141686125336, Std Dev: 71.41093891140993
Mean Absolute Error (MAE):	82.4215676571151
Root Mean Square Error (RMSE):	7.188411504145282
Structural Similarity Index (SSIM):	0.875150638861635
Peak Signal-to-Noise Ratio (PSNR):	19.45124544207347 dB

The preprocessed proposed algorithm effectively normalizes raw satellite images, enhances its quality, reduces noise and removes radiometric errors, as shown in the result in terms of PSNR. The pre-processed image has mean and standard deviation values significantly lower than the original DN satellite image which indicates the normalized process was successful. In the original image mean pixel is 11166.08175 and the standard deviation is 12991.40667. Images processed only with radiometric and geometric correction have a mean of 49.59613 and the standard deviation is 71.85929. Satellite images processed with the proposed algorithm have a mean of 52.05141 and a standard deviation of 71.41093. This indicates a uniform distribution of pixel value and will help use this data in the training of artificial intelligence models with deep learning for the prediction of landslide events.

3.3 Preprocessing in database

Preprocessing is an essential step in deep learning, ensuring that raw data is processed into an appropriate format for model training. The quality of the preprocessing can have a substantial impact on the model's performance, training time, and capacity to generalize to new data. Here, we explore various preprocessing approaches typically employed in deep learning, with a special focus on their applications in landslide prediction. Deep learning models require large number of data and sensitive to the quality and format of their input data. Raw data frequently contains inconsistency, noise, which may disrupt the learning process. Preprocessing help to resolve these challenges by increasing data diversity through augmentation, normalizing the data distribution, reducing dimensionality for more effective calculation. Proper

preprocessing allows for faster convergence during training, greater model generalization and higher overall performance.

3.3.1 Preprocessing techniques

1. Data Augmentation

Data augmentation artificially improves the diversity of the training dataset by adding random transformations such as rotation, flip, translation, and scaling. This reduces overfitting while increasing the model's capacity to generalize to fresh data. Augmentation is very effective when there is a limited amount of data. Different augmentation techniques as color jitter, resize, random flip, random horizontal can be applied to dataset to increase number of images in dataset [43]. Augmentation was important because the supplied dataset would be small which could have resulted in overfitting.

- **ColorJitter** : The use of ColorJitter augmentation increased the model's robustness by mimicking various real-world circumstances. This simple augmentation strategy exposed the model to images of variable brightness levels, allowing it to generalize more effectively to previously unknown data taken under various lighting situations. Outdoor footage, such as satellite pictures, frequently features a variety of lighting conditions. By modifying images with brightness variations, the model can be trained to be more resilient to lighting shifts. This ensures that the model does not rely on unique lighting conditions to detect landslides, increasing its generalizability. Color jitter transformation randomly adjusts the image's brightness by up to 20%. ColorJitter is a strong enhancement tool that can change contrast, saturation, and hue. Figure 3.9 shows the input image and brightness transformation 5 times, generating 5 augmented images. Each transformed image will have a slightly different brightness.



Figure 3.9 Brightness Transformation of image

- Rotation:** Rotation is a crucial transformation for landslide images since landslides can occur in a variety of orientations. To imitate diverse viewpoints, we used random rotations within a different angle range. This was accomplished with the Rotate function from the Albumentations library, a powerful tool for picture augmentation in deep learning pipelines. To create numerous augmented images, we performed the rotation transformation to each image five times. This technique generated five new versions of the original image, each rotated at a different random angle. The inclusion of rotation augmentation is predicted to increase the model's generalization by making it more invariant to diverse landslide orientations observed in test data. Figure 3.10 shows the original image alongside five rotated images with $\pm 50^\circ$ generated through augmentation. When applying rotation as a data augmentation technique, choosing an appropriate rotation limit can significantly impact the diversity of the dataset and improve the model's ability to generalize.



(a) original image



Figure 3.10 Original image alongside five rotated images

- Horizontal Flip :** Horizontal flipping is a transformation that reverses an image along its vertical axis. For landslide imagery, this technique simulates circumstances where satellite imaging or ground-level photography may catch landslides from different camera angles. Introducing this type of variation allows the model to become less sensitive to the precise orientation of features in the images, boosting the model's robustness and generalizability to new data sets. Argumentation is a useful tool for enhancing images in computer vision applications, with transformations like rotation, cropping, scaling, and flipping. In our augmentation pipeline, we used a random horizontal flip with a 50% probability. This means that each image has a 50% probability of being flipped, therefore both the original and inverted versions of the landslide are used in the training process. The horizontal flip augmentation added variation to the training dataset by increasing the variability of image orientations. This led in increased model generalization, allowing the model to perform better on previously unknown test data, especially when images were given from various perspectives. Figure 3.11 shows an example of an original image and its horizontally flipped version.

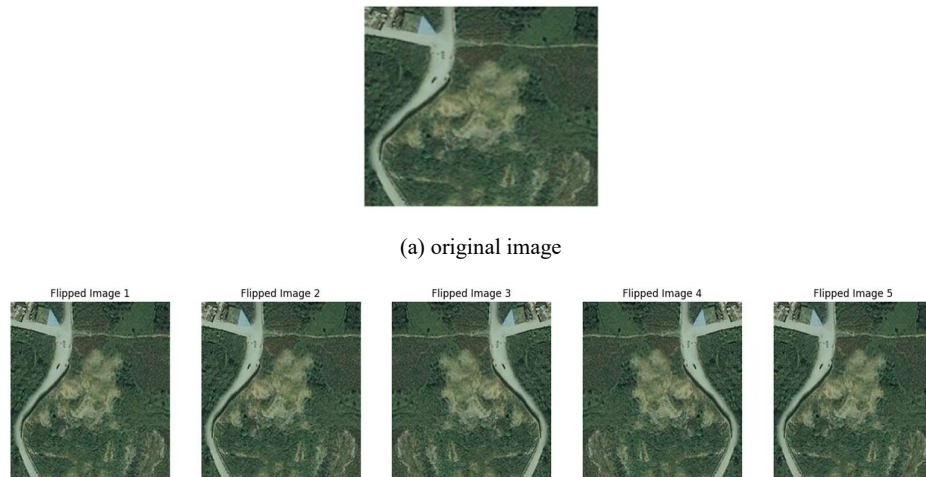


Figure 3.11 Horizontally flipped version of Original image

- **Resize:** The resizing step ensured that all of the images in the dataset were compatible with the deep learning model's input layer, which requires a fixed input size. Resizing photos to 150 x 150 pixels reduces computational cost, allowing for faster training while keeping sufficient quality to detect landslide-related features. By standardizing the input dimensions, the model was able to focus on learning important patterns throughout the dataset without being impacted by image size or resolution differences. This preprocessing step made the training process more efficient, allowing the model to converge faster. Figure 3.12 displays the original landslide image and the scaled image, with the original image shrunk to 150 by 150 pixels. As observed, the image's fundamental qualities are preserved, ensuring that resizing does not compromise the essential attributes required for model training.



Figure 3.12 Original and resized image

3.3.2 Principal Component analysis (PCA) and variants

PCA is a popular dimensionality reduction preprocessing technique that converts high-dimensional data into a smaller collection of components that represent the majority of the variance [76] [77]. It converts the input data into a set of linearly uncorrelated variables called as principal components, which are sorted by the amount of variance they extract from the original dataset. This method works exceptionally well when dealing with high-dimensional data, where redundancy and multicollinearity can degrade model performance. This is especially effective for large datasets, as fewer features can greatly reduce the model's computational complexity. In deep learning, PCA is frequently used to eliminate duplicate information from image data and compress it while retaining critical features. This can help reduce the training time of convolutional neural networks (CNNs) while preserving essential patterns in images [78]. Kernel PCA is a non-linear data version that uses a kernel function to map input data into a higher-dimensional space, allowing for the capture of more complicated patterns.

In landslide detection research, PCA is used to minimize the dimensionality of image data. Given that high-resolution satellite images frequently contain a significant number of correlated features (e.g., pixel intensities across different bands), PCA enabled to minimize data complexity while keeping the most relevant elements required for categorization. Dimensionality reduction proved critical in speeding up the training of deep learning models like ResNet and GoogleNet while preserving important visual information.

PCA has various variations that address specific constraints of the traditional PCA method. These versions provide greater flexibility and efficiency with high-dimensional or non-linear data. Kernel PCA is one variant of PCA is non linear extension of PCA that allows for non-linear dimensionality reduction using kernel functions [79]. Common Kernel functions are Gaussian (RBF) kernel, Polynomial kernel, and Sigmoid kernel.

Table 3.3 Compare the result of PCA and Kernel PCA for different images.

	Mean Square Error for PCA	Structure Similarity Index for PCA	Mean Square Error for Kernel PCA	Structure Similarity Index for Kernel PCA
Sample 1	30.97	0.97	0.43	0.99
Sample 2	27.40	0.97	0.48	0.99
Sample 3	19.17	0.98	0.46	0.99
Sample 4	19.13	0.98	0.49	0.99
Sample 5	12.02	0.98	0.49	0.99
Sample 6	17.17	0.98	0.49	0.99
Sample 7	37.18	0.93	0.47	0.99
Sample 8	15.38	0.98	0.49	0.99
Sample 9	21.76	0.99	0.51	0.99
Sample10	14.70	0.99	0.49	0.99

Kernel PCA uses the RBF kernel, which maps data into a high-dimensional space where nonlinear patterns are captured more effectively. As a result, it preserves image structures, textures, and edges more faithfully than linear PCA, leading to consistently high SSIM values.

3.3.3 Algorithm of Proposed Work

The preprocessing workflow contains several steps including Data cleaning, color jitter, random rotational, random horizontal, resize and feature and dimensionality reduction. Figure 3.13 presents a flow diagram of the proposed work.

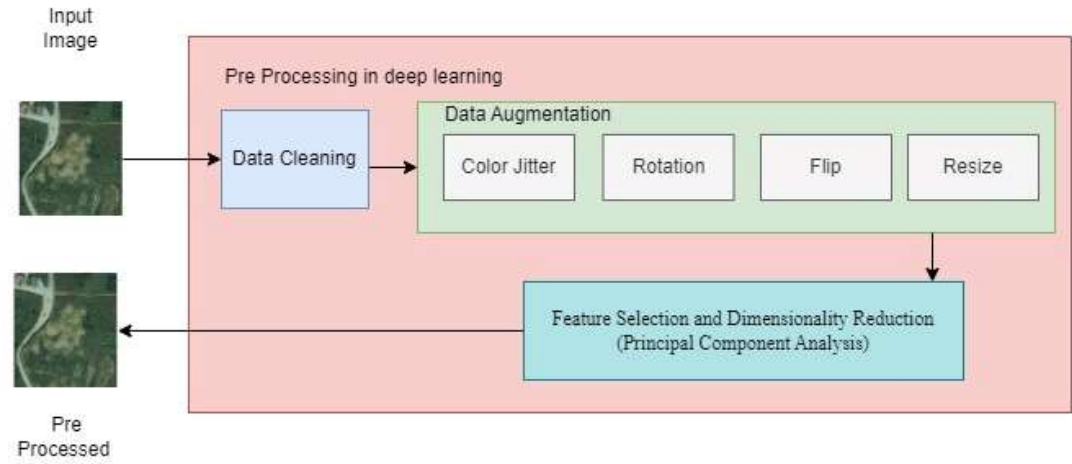


Figure 3.13 Proposed Preprocessing

Algorithm of Preprocessing of Dataset

Step1: Upload the dataset. Load images and labels from the dataset.

Step 2: Clean the dataset by removing the duplicate images in the dataset to avoid redundancy.

Step3: Define Augmentation techniques: Color Jitter, rotation, horizontal flip and resize. Set brightness of 0.2, maximum rotation is 5 degree in clockwise and anticlockwise direction, flip the image horizontally with 50% probability and resize image to 150x150. Append the augmented images to the training set.

Step 4: Normalize pixel values, scale pixel values to the range [0, 1].

Step 5 : Flatten the images for Principal component analysis.

Step 6 : Apply Kernel Principal Component Analysis. Choose a kernel function RBF for the Kernel PCA. Initialize Kernel PCA with the desired number of components and the selected kernel function. Fit Kernel PCA on the combined matrix of flattened images. Transform the combined matrix to reduce dimensionality based on Kernel PCA.

$$K(x,y) = \exp(-\|x-y\|^2 / 2\sigma^2) \quad \text{-----}(5)$$

Where $\|x-y\|^2$ is squared Euclidean distance between the two feature vectors.

σ is the variance and our hyperparameter.

Step 7: Split the dataset into training and testing datasets.

3.4 Conclusion

The preprocessing techniques applied to raw satellite images include radiometric, geometric, atmospheric corrections, noise removal, normalization and filtering, etc. towards improving the quality of the image for further processing. In this work, we studied and interpreted different types of errors present in raw satellite images of landslide events and their impact on the accuracy of further landslide detection tasks. A strong preprocessing algorithm containing radiometric, geometric, and atmospheric corrections is then applied to improve the quality of satellite images. The results of preprocessing yielded in the experimental work are promising and the performance of the techniques is evaluated using quantitative analysis parameters such as PSNR, RMS, and SSIM. Further, these preprocessed images can be used in training and testing data sets of deep learning and machine learning models. Good-quality images can help to improve the training of the artificial model and helpful in the prediction of atmospheric conditions, and disasters like earthquakes, landslides and floods.

Integrating data augmentation and Kernel Principal Component Analysis (PCA) into the preprocessing workflow provides considerable advantages for increasing model performance in tasks such as image classification using Convolutional Neural Networks. Data augmentation adds diversity to the training set by producing versions of the original dataset using techniques such as rotation, scaling, and flipping. This allows the model to generalize better to new data, lowering the likelihood of overfitting. These strategies combine to generate a more robust model that can reliably predict classes in complex datasets, eventually leading to enhanced performance of deep learning models. Kernel PCA reduces the dataset's dimensionality while keeping its fundamental properties. This not only accelerates the training process, but also helps to alleviate the curse of dimensionality, making the model more efficient and effective. Kernel PCA can help filter out noise and less significant features by reducing dimensionality, resulting in a cleaner dataset for better training and model accuracy. The combination of data augmentation with Kernel PCA in the preprocessing step considerably increases the quality and efficacy of the dataset used for training.

Chapter 4 Deep Neural Network for Detection of Landslide

Convolutional Neural Networks have transformed the field of computer vision, with exceptional results in picture classification, object recognition, and segmentation applications. Convolutional Neural Networks have emerged as a strong and crucial tool in the field of deep learning, particularly for applications like image processing and computer vision. Inspired by the structure and function of the human visual system, CNNs are meant to automatically and adaptively learn spatial hierarchies of characteristics from input images. CNNs' capacity to extract useful characteristics at many levels of abstraction has allowed them to achieve cutting-edge performance in a number of applications, including object detection, image classification, medical image analysis and facial recognition.

4.1 Architecture of Convolution Neural Network

A convolutional Neural Network is a feed-forward deep neural network that contains three layers: input layer, hidden layer and output layer. CNN uses images as input and, through iterative training, automatically modifies the network's parameters based on the discrepancy between the network's output and the provided ground truth. CNN is used in computer vision which is a branch of Artificial intelligence. The basic architecture of CNN has multiple layers except for input and output, this network contains a convolutional Layer, Pooling Layer, and Dense layer as shown in Figure 4.1. CNN architecture have two main parts: feature extractor and classifier. Feature extractor also known as encoder has convolution layer, activation layer and pooling layer. Classifier also known as a decoder have fully connected or dense layer [80], [81].

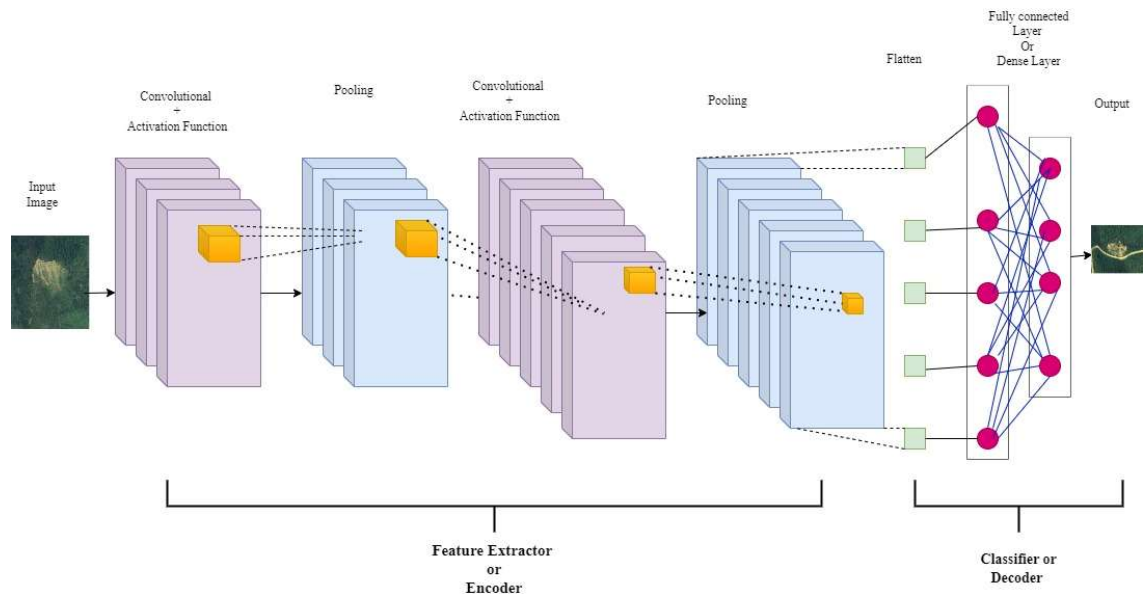


Figure 4.1 Basic CNN Architecture

1. **Input Layer:** The input layer receives the image as input and passes it to the next layer.
2. **The Convolutional layer:** The Convolutional layer is core of CNN and the majority of computation occur in the CNN's core convolutional layer. CNNs are built on convolutional layers, which perform convolution on the input image. This layer requires a few components: input data, a filter, and a feature map. Each convolution process uses a filter or kernel to slide across the input image, conducting element-wise multiplications and adding the results to create a feature map. The convolutional layer applies a set of filters to the input image to extract the features such as edges, shapes and texture. The filter size is generally a 3x3 matrix applied to the image and dot product of input pixels and the filter is fed into the output array is known as a feature map. This procedure enables the network to identify local patterns like edges, textures, and forms. Multiple filters can be used to capture different features, resulting in a diverse set of feature maps that reflect various characteristics of the input data. Different filters like sobel filter is used to detect edges and the average filter is used for enhancement.

3. **Activation Function** : A linear convolutional layer followed by nonlinear activation function f . Activation functions can be binary, sigmoid, and Rectified linear unit (ReLU) functions. The Rectified Linear Unit (ReLU) is one of the most popular activation functions due to its simplicity and effectiveness. It enables the model to learn complicated patterns by representing non-linear correlations between features.

$$\text{Output feature map} = f(\text{input} \times \text{weights} + \text{bias}) \text{ -----(6)}$$

Where f is the activation function. Figure 4.2 shows the output feature map obtained by pixel-by-pixel scan of input image with the filter of size $(n \times n \times c)$ where c is a number of channels. The output is processed by numerous convolutional layers, and activation functions are given to the next pooling layer. The pooling layer follows the convolution and ReLU activation functions.

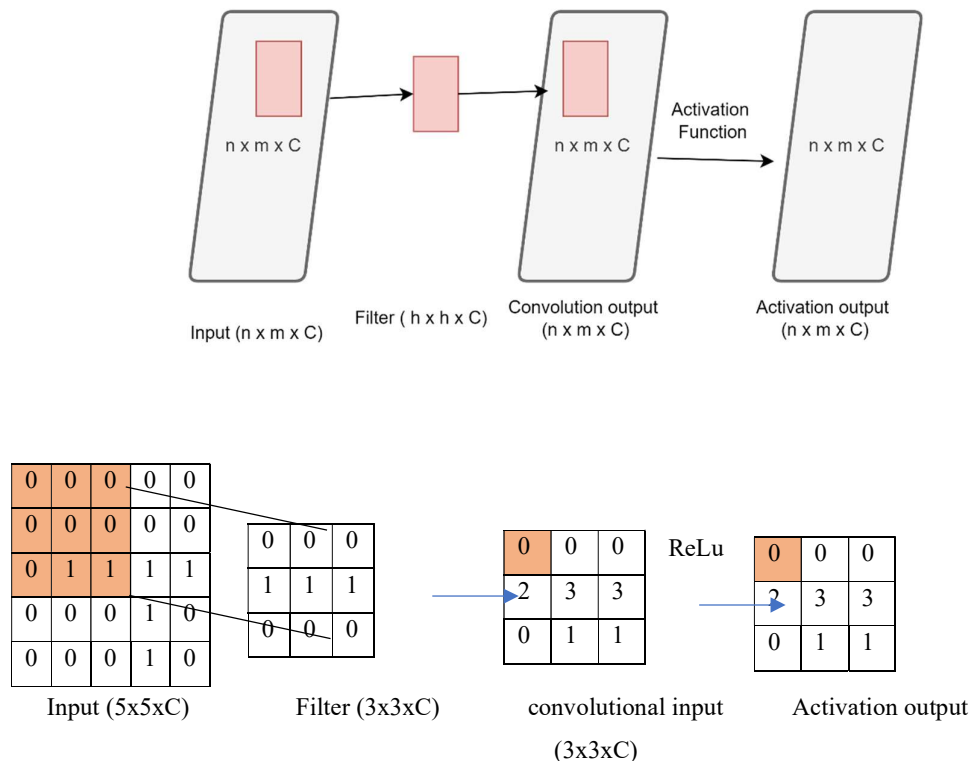


Figure 4.2 The Convolution layer function in CNN

4. **Pooling Layer** : Pooling layers are used to minimize the spatial dimensions of feature maps while still keeping the most important information. It is the process to downsampling and reducing the size of the matrix and increase the computational speed. It is used to select the prominent features from the output image. This layer consists of a filter with a defined stride size that moves across the feature map area. Maximal and average are two common pooling operations. The maximum value is stored in feature maps with the maximal pooling operation, while the average value is stored in feature maps with the average operation as shown in Figure 4.3. Max pooling, which selects the highest value from a given window, is widely employed. This downsampling strategy not only reduces computing cost, but it also helps to avoid overfitting by introducing translation invariance. Which type of pooling is used depends on the type of information required. The flatter function converts the multi-dimensional matrix into a single-dimension vector. This vector is then provided to the fully connected layer shown in Figure 4.4.

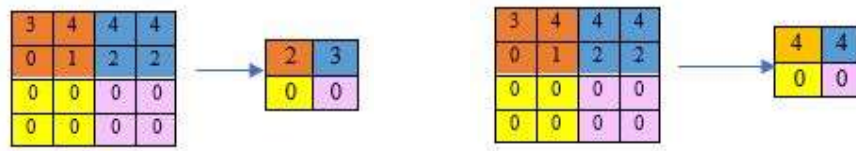


Figure 4.3 Two Pooling operations with pooling window 2x2.

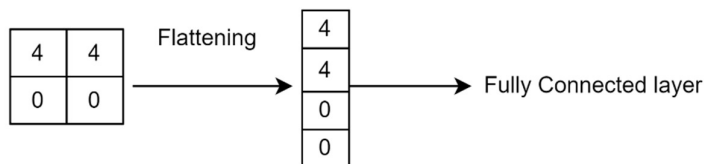


Figure 4.4 Flattening operation

5. **Fully Connected Layer**: At the network's end, fully connected layers use the features learned from previous levels to produce predictions. These layers connect every neuron in the previous layer with every neuron in the current

layer, eventually leading to an output layer that computes the final categorization scores.

Since the topic of artificial intelligence has grown, CNNs continue to play an important role in the creation of intelligent systems capable of comprehending and interpreting visual data. Their architecture, inspired by biological systems, allows them to acquire hierarchical representations, making them important for sophisticated visual identification tasks.

4.2 Deep Learning

Deep learning is a specialized subset of machine learning that utilizes neural networks with multiple layers—often referred to as deep neural networks—to model complex patterns in data. Traditional machine learning methods are simple and typically depend on manual feature extraction whereas in deep learning feature extraction is an automatic process that allows the network to learn directly from raw data. In this system, lower network layers collect basic data from images, such as edges and textures, while higher layers combine these features to capture more abstract ideas, such as shapes or individual objects. This ability to learn progressively complicated representations makes deep learning especially useful for applications where feature engineering is difficult or impossible [82].

Deep learning has drawn tremendous amounts of attention for its exceptional ability to handle unstructured data like pictures, audio recordings, and text. One of its key advantages is its ability to handle large amounts of data efficiently. As big data has become more widely available, deep learning models have demonstrated the capacity to significantly increase their performance by using enormous datasets. Additionally, in the field of computer vision, deep learning models have outperformed standard algorithms in image categorization tasks, identifying complicated patterns with more accuracy. CNN models have revolutionized object identification and picture segmentation in computer vision, enabling precise real-time applications in industries such as autonomous driving and monitoring. Another significant benefit of deep learning is its adaptability and scalability. Deep learning models can be easily customized to perform a variety of tasks by fine-tuning pre-trained networks on specific

datasets. This adaptability enables practitioners to use current models to achieve high performance in a variety of areas, including healthcare where deep learning aids in disease diagnosis from medical images, finance, where it improves fraud detection systems and disaster management where it save life and property by prediction and detection of disasters. Deep learning stands out for its capacity to automatically learn rich representations from data, particularly unstructured data, while delivering higher performance across a variety of tasks. Its adaptability and effectiveness make it an essential framework for the improvement of modern artificial intelligence [83].

4.2.1 Deep Residual Network

Deep learning models have significantly advanced in recent years, enabling the building of deep neural networks with multiple layers. Deep neural networks have several hidden layers and are difficult to train. Training deep models can be difficult as the number of layers in a neural network rises. Residual Networks, or ResNets, are a type of deep neural network that tackled these most significant difficulties in deep learning. Deep networks are harder to train because of vanishing gradient problems. Deeper networks survived worse than shallower networks, not because of overfitting, but due to optimization challenges. As networks traveled deeper, the gradients of the loss function decreased, making it impossible to propagate error back to earlier layers—this is known as the vanishing gradient problem. The Residue Network is used to avoid the vanishing gradient problem that happens while employing backpropagation to update the weights. The residual block is shown in Figure 4.5 below, input is represented by X , Output I is represented by Y , $F(x)$ is output after the operation function of two convolution layers. These blocks incorporate a shortcut pathway, effectively functioning as an identity function, thereby indirectly bypassing the training process for one or more layers. A residual network adds input directly to the calculated value or loss function and tries to make the loss function equal to zero, so that input must be equal to output. Feedforward neural networks with "shortcut connections" can implement $Y = F(x) + x$. Residual Networks are deeper neural networks but are easy to train and easy to optimize. This network provides highly accurate results [84], [85]

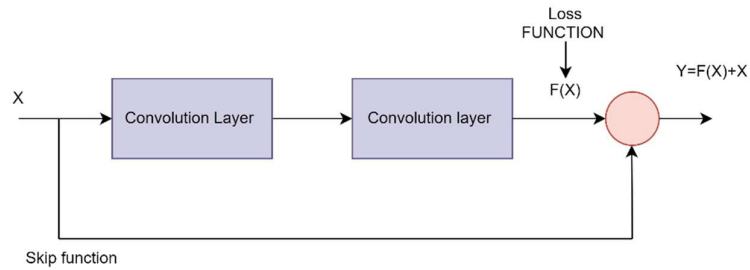


Figure 4.5 Skip Connection on ResNet

There are two types of residual blocks in ResNet: Identity block and convolutional block.

Identity Block: In this block, the input and output dimensions remain unchanged, and the shortcut connection adds the input to the output without any modification.

Convolutional Block: If the dimensions of the input and output differ due to downsampling, the shortcut connection uses a convolution operation to match the dimensions before adding the input to the output.

Bottleneck : The bottleneck design is an important aspect of deeper ResNet structures. It optimizes the architecture by lowering computational costs while retaining excellent performance. The word "bottleneck" alludes to the architecture's method of reducing and then expanding the dimensionality in each residual block, which makes the network more efficient while maintaining its capacity to learn rich, complicated representations. Each residual block in the bottleneck design consists of three layers of convolutions: 1x1 convolution layer for Dimensionality Reduction, 3x3 Convolution for Feature Extraction, 1x1 Convolution for Dimensionality Restoration. As shown in Figure 4.6

1. **1x1 convolution:** The first layer in the bottleneck block is a 1x1 convolution, which minimizes the number of channels and thereby compresses the data. This lowers the computational cost by reducing the amount of feature maps that must be processed in subsequent layers.
2. **3x3 Convolution:** Following dimensionality reduction, the second layer is a 3x3 convolution, which performs the fundamental feature extraction operation. This layer gathers spatial information while reducing computing overhead by acting on compressed feature maps.

3. **1x1 convolution** :A second 1x1 convolution layer is used to restore the original number of channels. This ensures that the network's depth and width are retained while taking use of the efficiency gained during the intermediate steps.

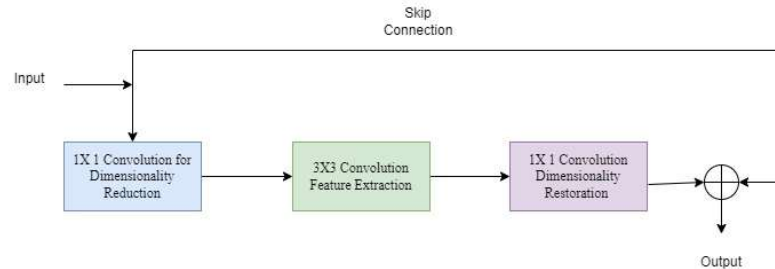


Figure 4.6 Bottleneck Residual

Residual Networks introduced different architectures with varied depths. The deeper variants, such as ResNet-50 and beyond, use bottleneck layers to lower the computational cost of deeper networks while preserving performance.

Benefits of Residual network

1. **Correction of the Vanishing Gradient Problem:**ResNet's skip connections enable gradients to flow directly through the network's layers, avoiding the vanishing gradient problem that affects deep networks. This leads to more reliable training with very deep architectures.
2. **Scalability and Flexibility:** ResNet architectures have been able to scale much deeper than was previously possible with conventional deep networks. Because ResNet can train networks with more than 100 or even 1,000 layers, it is a very adaptable model for a wide range of applications
3. **Easy Optimization:** Instead of forcing the network to learn the full transformation between layers, ResNet simplifies the learning process by asking the model to focus on learning the residuals. This leads to more efficient training and quicker convergence.
4. **Better Generalization:** Residual learning improves both training performance and the model's ability to generalize to new inputs. Networks with residual connections achieve improved test performance across many applications. Skip connections simplify optimization and reduce overfitting, allowing the model

to focus on learning key residuals and boosting generalization to real-world problems.

5. **Enhanced Accuracy:** Residual learning has dramatically increased accuracy in different applications with ResNet models setting new standards. It permits the capture of both low- and high-level characteristics, which improves precision in a variety of computer vision tasks. Beyond vision, residual connections have been modified for speech recognition and natural language processing, improving performance in a variety of applications.

ResNet has different architectures and ResNet101 and ResNet50 are used in this study along with Inception GoogNet

4.2.2 Architecture of ResNet 50

ResNet50 is a deep convolutional neural network with 50 layers that incorporates the main concept of residual blocks with skip connections. These skip connections allow the model to learn residual functions rather than entire transformations, making it easier to train the network even with multiple layers. ResNet50 comprises of 16 residual blocks, each with three convolutional layers. The residual blocks use skip connections, which bypass one or more levels and connect a block's input to its output immediately. ResNet50 comprises of 16 residual blocks, each with three convolutional layers. The residual blocks use skip connections, which bypass one or more levels and connect a block's input to its output immediately. These connections are critical for reducing the vanishing gradient problem, which arises in very deep networks when gradients decrease as they propagate back through several layers during training. These skip connections allow gradients to flow more easily, resulting in smoother and more efficient deep network training.

ResNet50 consists of 50 weight layers, including convolutional, pooling, and fully connected layers. The piled remnant blocks, on the other hand, represent a significant architectural innovation. These deep layers enable the network to extract high-level information that are required for more difficult tasks like picture categorization and object detection.

ResNet 50 designs differ only in terms of the number of layers. Figure 4.7 shows ResNet 50 architecture [86][87]. Different layers perform different roles, and the machine can learn from data and extract information that is useful in categorization. Convolutional layers extract features from an input image using a succession of convolutional filters. Batch normalization used before the activation function helps to improve generalization, Faster convergence and Regularization. The activation function is an excellent approach to create nonlinearity in the output of a neuron. $\text{ReLU}(x) = \max(x, 0)$ where x is input, is a piecewise linear function, making it computationally efficient to compute and differentiate while avoiding the vanishing gradient problem. Maximum pooling is used to keep the essential information and reduce the dimensions of feature maps that are generated from the first convolution layer. It overcomes the overfitting issue, computational cost and reduces the noise. There are two blocks in ResNet: identity block and convolution block. An identity block or skip connection maintains the same dimensions of the input and output image. The convolutional or projection block is used when the dimensions to input and output are different as shown in Figure 4.8. ResNet50 has 3 layers in second convolution layer, and 4 layers in the third convolution layer. In fourth convolution layer have 6 layers in ResNet 50. Networks have 3 layers of filter size 512, 512, 2048 followed by average pooling layer. Average pooling takes average value of specified area and reduce the size of image. Flattening uses a flatter function to convert matrix in to single dimension vector whose output is fed to fully connected layer also named as dense layer. Each neuron in a fully connected layer receives input from all neurons in the preceding layer. Fully connected layer are often employed as the last layers in a neural network and make the final predictions. The output is then calculated as a weighted sum of its inputs, followed by an activation function shown in equation (7)

$$Y(\text{out}) = \sigma(\sum X_i W_i + b_i) \text{-----}(7)$$

Where σ is activation function, Y is the output of the neuron, X is the input neuron, W is the weight vector and b is Bias

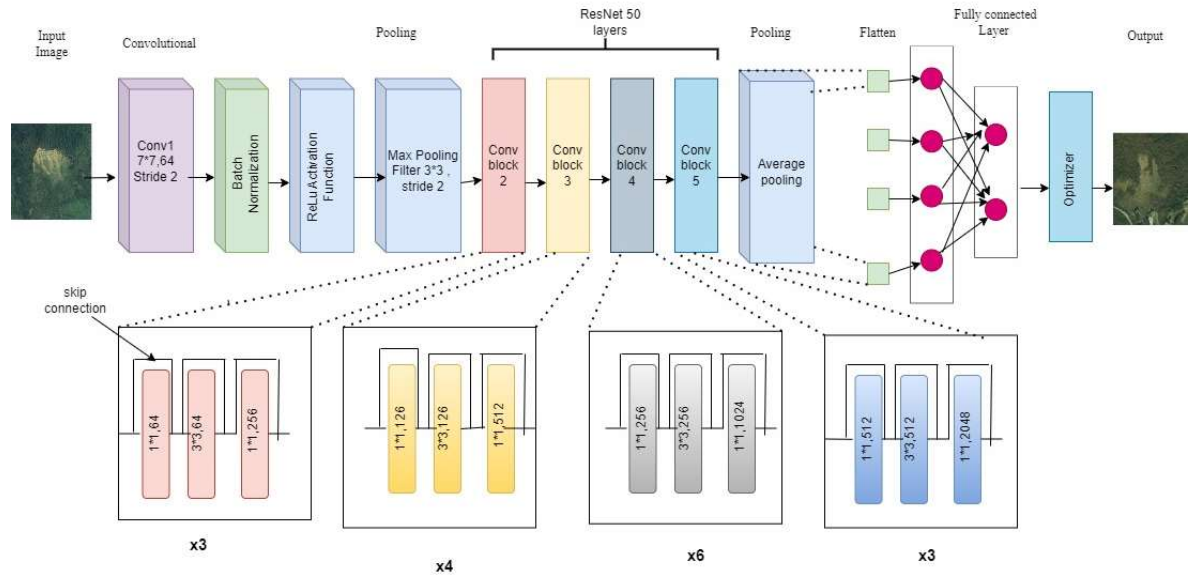


Figure 4.7 ResNet 50 Architecture

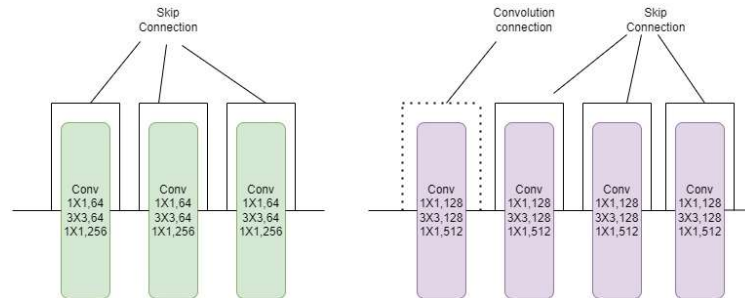


Figure 4.8 Skip and convolution connection

4.2.3 Architecture of ResNet 101

ResNet 50 and ResNet 101 designs differ only in terms of the number of layers. ResNet101 expands ResNet50's design by 101 layers, allowing it to learn more complicated data representations. ResNet101, like ResNet50, combines residual blocks with skip connections to provide efficient training and avoid the vanishing gradient problem. However, it has 33 residual blocks instead of 16, therefore it is substantially deeper. This increased depth allows ResNet101 to extract more detailed and hierarchical features from the data, potentially improving performance on applications

requiring finer-grained feature identification, such as picture classification at high resolutions or object detection in congested environments. ResNet101's additional layers allow it to capture more complex patterns and relationships in the data. This is particularly useful for large datasets or highly detailed tasks. Figure 4.9 shows ResNet 101 architecture [86][87].

Both ResNet50 and ResNet 101 has 3 layers in second convolution layer, and 4 layers in the third convolution layer. In fourth convolution layer have 6 layers in ResNet 50 and 23 layers in ResNet 101. Both networks have 3 layers of filter size 512, 512, 2048 followed by average pooling layer.

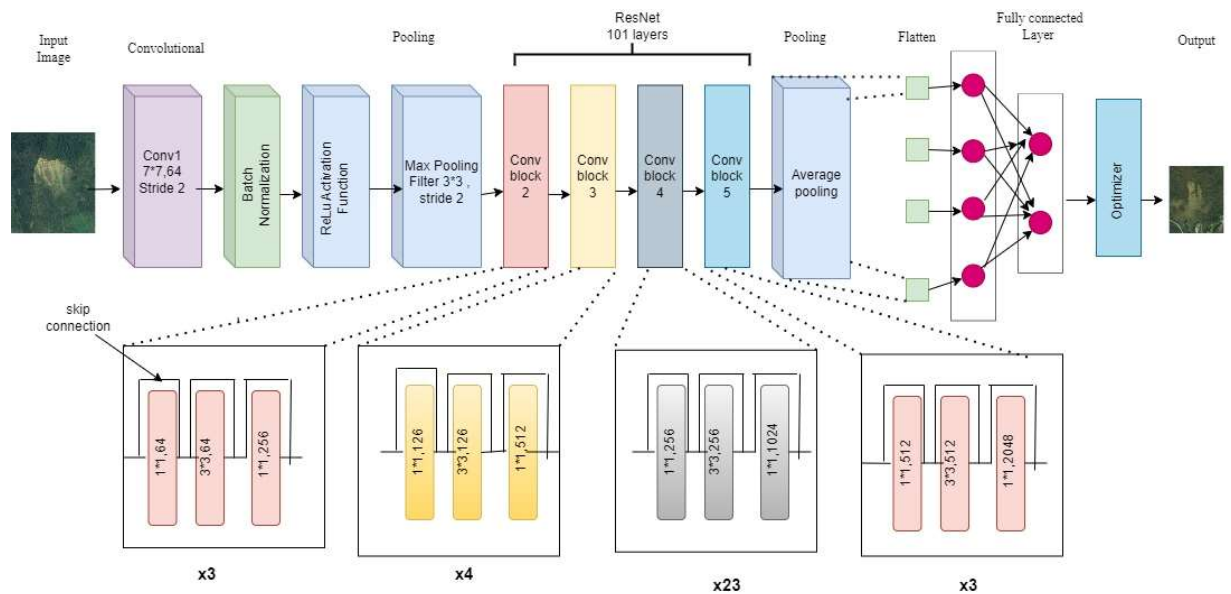


Figure 4.9 ResNet 101 architecture

4.3 GoogleNet

GoogleNet, developed as part of the Inception architecture, revolutionized deep learning by delivering a computationally efficient and powerful network structure. GoogleNet, unlike typical convolutional neural networks (CNNs) that rely on sequential layers, employs a new Inception module that enables the network to perform convolutions of varying filter sizes in parallel. GoogleNet's primary breakthrough is its capacity to execute feature extraction with improved computational efficiency, allowing for the training of deep networks without requiring excessive resources. GoogleNet has 22 layers, which is substantially deeper than previous systems. Despite

its complexity, GoogleNet is more computationally efficient thanks to the creative utilization of Inception modules [88] [89]. Rather than using completely linked layers, GoogleNet uses global average pooling at the final layer. Global average pooling substitutes fully linked layers, which typically introduce a high number of parameters, lowering the danger of overfitting. This pooling method averages the spatial dimensions of the feature maps, acting as a regularizer and increasing the model's robustness, particularly when working with larger datasets.

The Inception module, which is at the center of GoogleNet, presents a new approach of stacking layers. Rather than applying the same convolution to the whole network, the Inception module runs convolutions with filters of varying sizes in parallel. Each inception module has 1x1 convolution, 3x3 convolution, 5x5 convolution as shown in Figure 4.10

1. **1x1 convolution:** This layer performs dimensionality reduction, which helps to save computational costs by lowering the number of channels before performing larger convolutions. It catches fine-grained characteristics.
2. **3x3 convolution:** This filter is intended to capture medium-sized spatial patterns while maintaining a balance between spatial resolution and computational cost.
3. **5x5 convolution:** The 5x5 filter extracts broader features over a larger receptive field, helping the network capture more complex and wider spatial patterns.
4. **3x3 max pooling :** Pooling activities are performed concurrently to provide more context while reducing geographical dimensions.

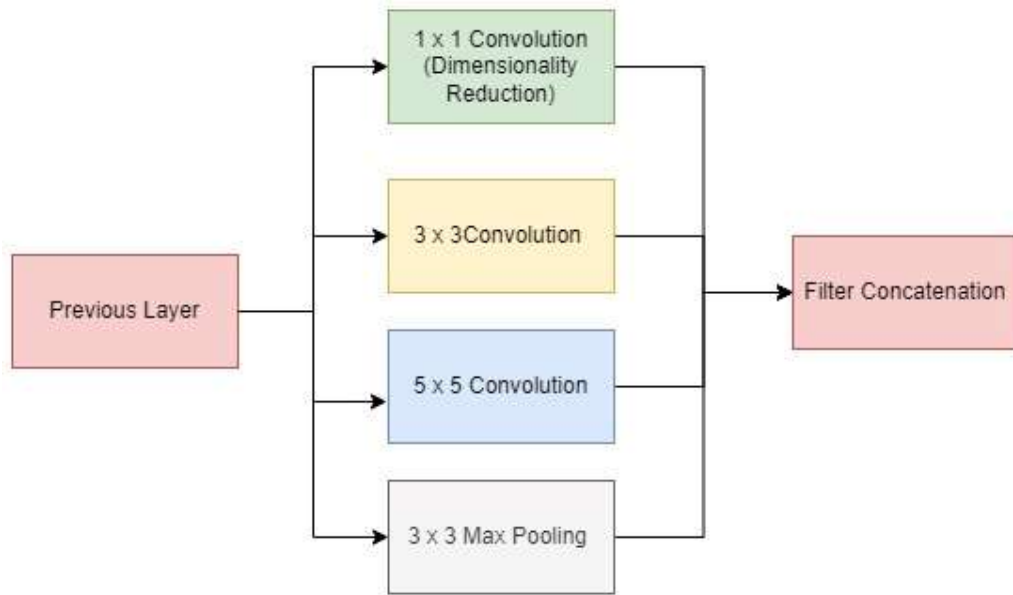


Figure 4.10 GoogleNet Inception Module

4.4 Attention Mechanism

Deep learning models have made considerable advances in image processing, but some issues remain, such as detecting meaningful characteristics from large images. The attention mechanism, modeled after human visual perception, provides a strong solution by allowing models to focus on the most informative aspects of a picture. Attention mechanisms improve the model's ability to process visual data by altering the relevance of different regions or characteristics. The attention mechanism utilized in CNN focuses on specific parts of the input in the input feature map and dominates other regions in the background, improving the performance of CNN in prediction. Various types of attention model are spatial attention, channel attention, self-attention, and multi-scale attention.[90]. To introduce an effective attention module for classification and prediction convolution layer, activation function, pooling layer and fully connected layers are combined in different ways. Convolution block attention module has two sub-modules: Spatial attention module, the Channel Module. The input image is in 3D matrix($w \times h \times c$) where w and h are the width and height of the feature map respectively and c is the total number of channels. C is decided by the total number of filters used in that layer.

4.4.1 Spatial Attention

Spatial attention focuses on important regions that is useful for the task of classification. Spatial attention is concerned with where the model should concentrate inside an image, emphasizing crucial spatial regions that contribute the most to the prediction. This method is very beneficial in applications like image segmentation and object recognition, which need the model to recognize specific regions of interest, such as objects or edges. Spatial attention in satellite imagery used for landslide detection assists the model in prioritizing areas of instability or movement, hence improving the detection of high-risk zones.

The architecture of Spatial Attention includes the production of an attention map that focuses on the most relevant spatial regions of a feature map. It is designed to help a neural network attention to the most significant sections of an image while ignoring less informative regions. It is designed to help a neural network attention to the most significant sections of an image while ignoring less informative regions [91].

Spatial Attention architecture as shown in the Figure 4.11 have three layers: input feature map, pooling, and convolution.

1. Input Feature Map

The spatial attention module operates on the feature map generated by a CNN's convolutional layer. Input of the spatial attention module is a feature map $X \in \mathbb{R}^{C \times H \times W}$ where C is number of channel, H , W is height and width of feature map.

2. Pooling

Two types of pooling operations are performed in this step: **max pooling** and **average pooling**. Max pooling captures the most significant features in each spatial region across all channels. Average Pooling measures the average feature value at each spatial location across all channels.

$$X_{\text{maxpooling}} = \text{MaxPool}(X) \quad \text{-----}(8)$$

$$X_{\text{averagepooling}} = \text{AveragePool}(X) \quad \text{-----}(9)$$

3. Concatenation

The results of the max pooling and average pooling processes are then concatenated along the channel axis to form a combined feature representation that captures both the maximum and average context. The combined feature

map subsequently passes through a convolution layer to produce the spatial attention map.

$$X_{\text{concatenate}} = \text{Concat}(X_{\text{maxpooling}}, X_{\text{averagepooling}}) \quad \text{-----}(10)$$

$$X_{\text{spatialattention}} = \sigma(\text{Conv}_{7 \times 7}(X_{\text{concatenate}})) \quad \text{-----}(11)$$

Where σ is sigmoidal activation function, $\text{Conv}_{7 \times 7}$ is convolution operator with kernel 7×7 .

$$X_{\text{spatialattention}} = \sigma(\text{Conv}_{7 \times 7}(X_{\text{maxpooling}}, X_{\text{averagepooling}})) \quad \text{-----}(12)$$

4. Fully Connected layer

Combined output is applied to convolution layer. This layer generates a mask between 0&1 with sigmoid function and a single filter. The batch norm layer is used to normalize the output of convolution

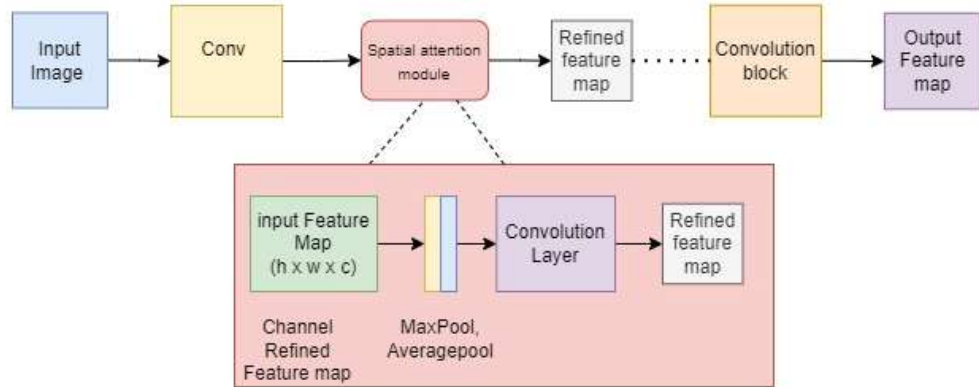


Figure 4.11 Architecture of spatial attention

4.4.2 Channel Attention

Channel attention is a technique used to concentrate attention on the most significant information across many channels. In terms of channel attention, the network learns to prioritize some feature channels over others. This helps the model to prioritize channels containing more relevant information. Channel attention map by taking advantage of the features' inter-channel relationships. Every channel in a feature map functions as a feature detector; its attention is directed towards determining "what" is significant in relation to the input image. This is accomplished by first performing a global pooling operation, then applying two fully connected layers (or convolutional layers) and a

sigmoid activation function. Number of neuron in MLP is decided by reduction ratio. MLP output is the provided to sigmoid layer [92]. A channel attention map was created by leveraging feature relationships across channels. In Channel attention weights are assigned to each channel, enhancing those channels that contribute to learning and improving overall model performance [93].

Channel Attention Module

Channel Attention architecture as shown in the Figure 4.12 has three layers: input feature map, Globalpooling and convolution.

1. Input Feature Map

The input to a channel attention mechanism is a feature map as $X \in \mathbb{R}^{C \times H \times W}$, where C is number of channel, H, W is height and width of feature map.

2. Global Pooling

To summarize the information from each channel, a global pooling procedure across spatial dimensions is used. To extract global information from each channel, two types of pooling are : **Global Average Pooling (GAP)**, **Global Max Pooling(GMP)**.Global Average Pooling computes the average value for each channel over all spatial positions. Global Max Pooling selects the maximum value across all spatial positions for each channel. Both GAP and GMP reduce the spatial dimension of the feature map to two $C \times 1$ vector. Resulting in a descriptor $Y \in \mathbb{R}^C$ where C is number of channel.

$$Y = \text{GlobalPooling}(x) = (\text{AvgPool}(X) + (\text{MaxPool}(X))) \text{ -----(13)}$$

3. Fully Connected layer

Following global pooling, the two vectors are passed through shared completely connected layers, producing a weight for each channel. This descriptor is then sent through two fully connected layers with a nonlinear activation (ReLU), to introduce nonlinearity. This stage generates a collection of channel weights, with each channel assigned a different priority score. The fully connected layers' outputs are then sent through a sigmoid activation function, producing a weight value between 0 and 1 for each channel.

$$\begin{aligned}
M_L(Z) &= \sigma(\text{MLP}(\text{Relu}(\text{GlobalPooling}(x)))) \\
M_L(Z) &= \sigma(\text{MLP}(\text{Relu}((\text{AvgPool}(X) + (\text{MaxPool}(X))))) \\
M_L(Z) &= \sigma(\text{MLP}(\text{Relu}(\text{AvgPool}(X) + \text{Relu}(\text{MaxPool}(X)))) \\
M_L(Z) &= \sigma(W_2(\text{Relu}(W_1(X^c_{\text{avg}}) + W_2(\text{Relu}(W_1(X^c_{\text{max}})))) \quad \text{-----}(14)
\end{aligned}$$

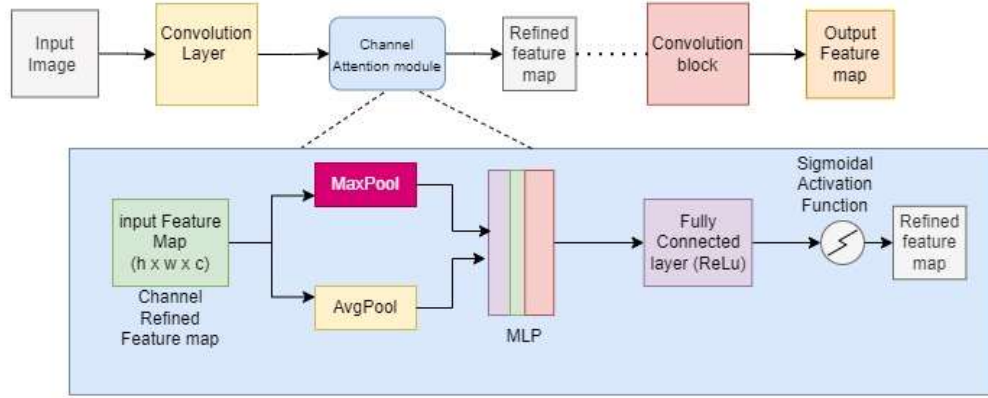


Figure 4.12 The architecture of Channel Attention

4.4.3 Squeeze-and-Excitation (SE) Blocks

SE blocks are a prominent method of integrating attention with convolutional networks. These blocks use channel attention, which "squeezes" global information and "excites" important feature channels by dynamically scaling them based on their relevance [92]. This SE block is categorized in three parts: Squeeze, Computation, Excitation as shown in Figure 4.13.

- **Squeeze** : Global average pooling is applied to the CNN layer's output feature map. This effectively takes the average value of all activations in the spatial dimension ($H \times W$), resulting in one activation per channel. This produces a vector of shape $(1 \times 1 \times C)$.
- **Computation**: The previous operation's vector is sent through two fully connected layers in succession. This achieves the goal of fully capturing channel-wise dependencies derived from the spatial mappings. After the first FC layer, a ReLU activation is applied, followed by a sigmoid activation after the second. The report also mentions a reduction ratio, which means that the

intermediate output of the first FC layer has a reduced dimension. The end result of this phase is also shaped $(1 \times 1 \times C)$.

- **Excitation:** The computing step's output is utilized to create a weight modulation vector for each channel.

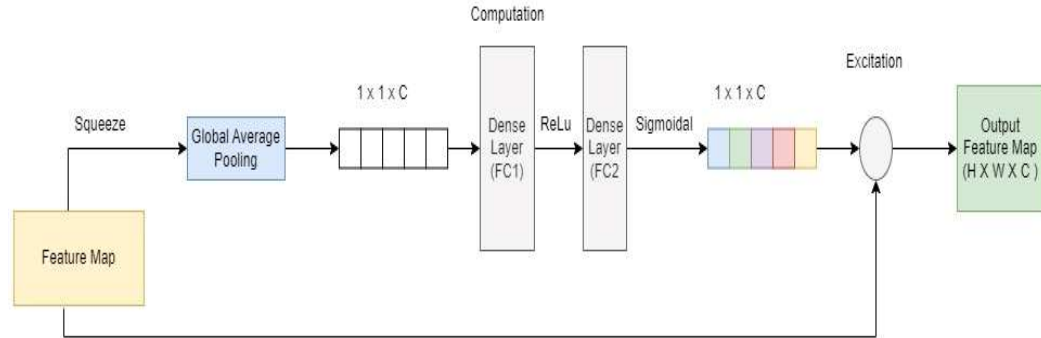


Figure 4.13 Architecture of Squeeze-and-Excitation (SE)

4.4.4 Self-Attention

Self-attention techniques, as employed in Transformer models, can also be applied to convolutional networks. Self-attention enables the network to represent links between various parts of the image by computing attention ratings across spatial locations or channels [94] [95]. This method can be used in deeper layers of ResNet and GoogleNet to record long-range dependencies within an image, allowing the network to handle more difficult tasks like object detection and semantic segmentation.

4.5 Proposed work

In this research work we have explored open source Biji's landslide database for landslide detection method using deep learning network with ResNet50, ResNet 101, GoogleNet and Attention module. Landslide detection flowchart consist: data preprocessing (augmentation and resizing of images) , data-based labelling for training and testing(Lnadslide and non landslide), training backbone networks (ResNet50, ResNet 101, GoogleNet), training and testing attention module, evaluate the result and optimize the result.

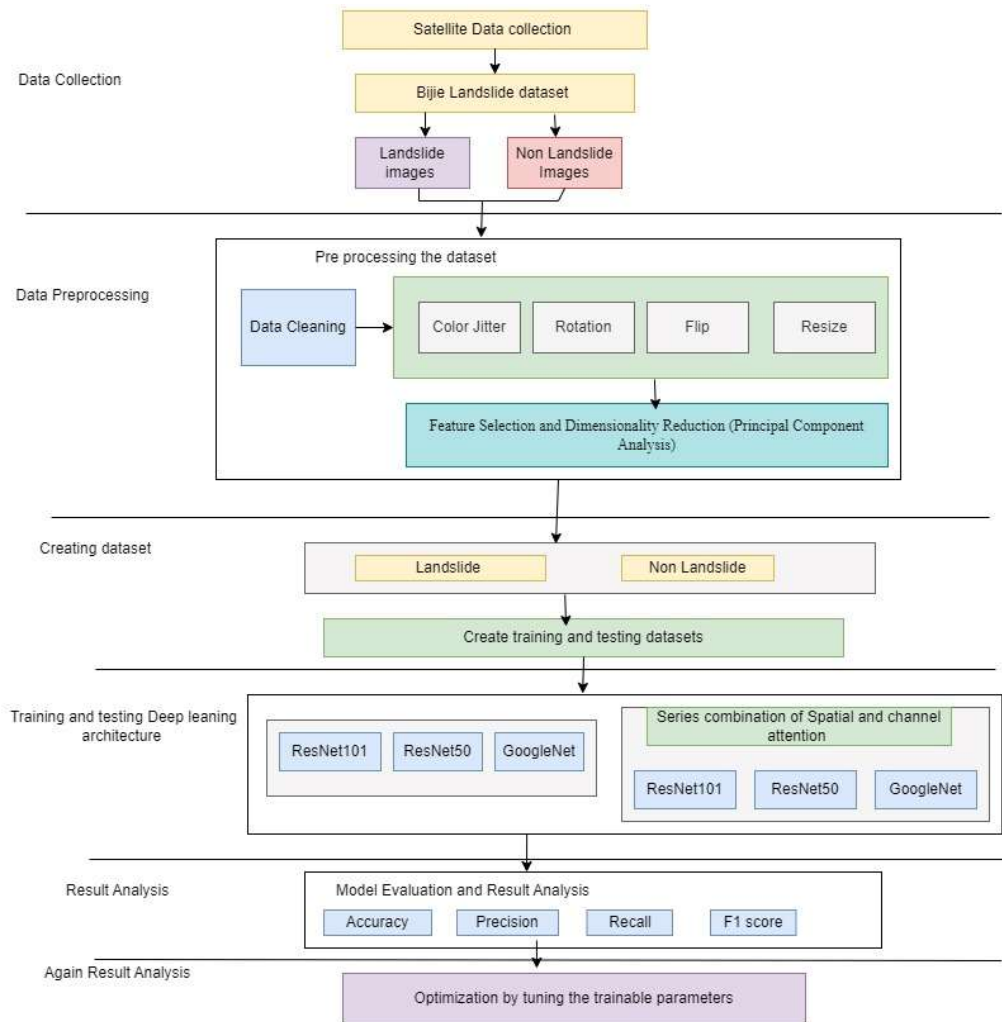


Figure 4.14 Proposed Flow diagram

Algorithm

Step 1: Create a database from satellite images.

Collect satellite images of landslide events and non-landslide events .

Step 2: Preprocessing

To increase the diversity of the images perform Augmentation and principal component analysis on collected images. Resize the image to match symmetry.

Step 3 : Create database: Divide input data in to two class Landslide and non landslide.

- Training dataset (Landslide and Non landslide): Landslide(X train) is a set of images for training.
- Testing data set (Landslide and Non landslide): Landslide (X test) is a set of images for testing.

Step 4: Build deep learning model

- Define a CNN architecture with convolution layer, pooling layer and fully connected layer .
- Use the activation function as ReLU.
- Use a dropout layer for regularization
- Select loss function
- Select an optimizer like Adam and learning rate.
- Use a series combination of channel and spatial attention mechanism.
- Take different backbone network : ResNet 50, ResNet101, GoogleNet, Attention Model.

Step 5: Compile the Model

- Compile the model with selected parameters like an optimizer, loss function and evaluation parameters .

Step 6: Train the Model

- Input the images training dataset (landslide and non landslide)
- Select the batch size and epoch.
- Train the model with the different batch size and epochs to minimize loss.
- Select the number of Epoch based on convergence.

Step 7 : Evaluate the model

- Evaluate the model for test database.
- Calculate accuracy, F1 score and precision.
- Analysis the result for better performance.

Step 8: Improve the model

- If the result is not good then need to adjust the hyperparameters, backbone Architectures and transfer learning.

Step 9: Predication landslide

- Use a trained model to predict a landslide and non-landslide image that is not in the database.

4.5.1 Placement of Attention Module

Attention Module can be arranged in different ways in any CNN network. Individual or a combination of attention modules can be placed in a residue block in the residual network or after inception block in googleNet. The attention module can be placed in the last residue block to refine the hidden layer or can be placed in the output of the residue network to refine the output map in the Residual network.

Channel and Spatial module can be arranged in three submodules: Sequential channel spatial module, sequential spatial channel module, paraller spatial and channel module. In our proposed work we place a sequential combination of channel spatial modules in the hidden layer of the residue network and after Inception block in googleNet. It refines the hidden layer for better results.

4.6 Experimental setup

4.6.1 Evaluation parameter

In proposed deep learning classification algorithms confusion matrix is used to evaluate the result. To evaluate the result we select four common metrics which are precision, recall, accuracy and F1-score. The confusion matrix presents a table as shown in Table 4.1 where T predicts the true value, and F predicts False value. P and N represent positive and negative type of prediction. To classify the result TP(True Positive), TN (True Negative), FN(False Negative), FP(False Positive) are used .

Table 4.1 Confusion matrix to classify landslide

Actual Value	Predicted Positive	Predicted Negative
Ground Positive	TP	TN
Ground Negative	FP	FN

Precision count the fraction of genuine positive prediction of landslide among all instances predicted as positive by the model, as shown in equation (15)

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False positive}} \quad \text{-----} \quad (15)$$

Recall assesses the model's ability to properly identify all positive instances among all actual positive instances, as shown in equation (15)

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad \text{-----} \quad (16)$$

F1 score combines both precision and recall into a single value and can better explain the result of classification. F1 score is calculated as the harmonic mean of precision and recall , as shown in equation (15)

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{\text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \quad \text{-----} \quad (17)$$

Accuracy is basic matric used to evaluate the overall performance of classification. It measures the proportion of truly classified instances among all instances in the classification dataset.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total number of instances}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad \text{-----} \quad (18)$$

4.6.2 Computational Complexity

In the experimental setup, the computational complexity of Convolutional Neural Networks (CNNs) is an important parameter that affects both the training and testing processes. Assessing computational complexity confirms that the models used are efficient, scalable, and appropriate for the given hardware resources. This computational complexity is based on the following important factors:

- **Floating point operations (FLOPS):** The number of floating-point operations required for a single forward transit through the network is a common measure of computational cost. FLOPs measure the number of operations performed

during convolutions, activations, pooling, and fully linked layers. Higher FLOPs typically indicate a more computationally intensive model, which might influence processing performance and resource requirements.

$$\text{FLOPS for Convolution layer} = 2(C_{\text{in}} \times C_{\text{out}} \times H \times W \times K_h \times K_w) \text{ --- (19)}$$

Where $C_{\text{in}}, C_{\text{out}}$: Number of input and output channel,

H, W : Height and width of the output feature map

$K_h \times K_w$: Height and width of the convolution kernel.

$$\text{FLOPS for Fully Connected Layer} = 2 (\text{Input size} \times \text{output size}) \text{ ----(20)}$$

- **Number of Parameters:** The total number of trainable parameters in a CNN reflects the model's size and memory requirements. Networks with a higher parameter count require large storage memory and typically require longer training times. This study compares the parameter efficiency of various architectures to understand their scalability and computational burden.
- **Inference Time:** Inference time is the time required for the model to process a single input sample and make a prediction. It is especially important in real-time applications, where minimal latency is required. Efficient models with shorter inference times are chosen for scenarios with stringent timing constraints.
- **Input Size:** Larger input dimensions increase the number of computations because convolutions are done on all pixels.
- **Number of Layers:** Deeper designs include more layers and thus more operations, which increases computing needs.

4.6.3 Hyperparameters

- **Learning Rate:** The initial learning rate was chosen to guarantee that the model takes tiny, controlled steps towards minimizing the loss function. A learning rate scheduler was used to gradually reduce the learning rate between epochs. The learning rate was near to zero, therefore the model converged to an optimal solution without overshooting.
- **Batch Size:** The batch size of 64 and 32 were chosen to strike a compromise between memory utilization and training stability. For large datasets, greater batch sizes aid in faster convergence but demand more memory.

- **Epoch:** The number of epochs was chosen to balance computational costs and model performance. Training for additional epochs can improve performance, but it also increases the risk of overfitting, therefore early-ending conditions could be investigated in future studies.
- **Weight Decay:** A weight decay was employed to regularize the model, preventing the weights from growing too big and thereby improving generalization on the validation dataset.
- **Gradient Clipping:** Gradient clipping with threshold value was used to keep gradients from inflating, which could cause instability during training, particularly when employing sophisticated models such as ResNet.
- **Optimizer:** The Adam optimizer was chosen because of its ability to dynamically modify learning rates during training, resulting in faster convergence than stochastic gradient descent (SGD).

4.6.4 Training and Validation

- **Dataset splitting :** Dataset was split in two training and testing datasets. No test data was used in training dataset as the aim was to optimize the models during the training process.
- **Overfitting Prevention**
 - **Dropout:** To prevent overfitting, a dropout layer was utilized during training, with half of the activations set to zero at random.
 - **Data Augmentation:** To improve the generality of the models and simulate differences in real-world circumstances, the augmentations such as : color jitter, resize, random horizontal, random rotation were applied.
 - **PCA and Kernel PCA:** These strategies were tested for dimensionality reduction to improve model performance by removing redundancy in features.

4.6.5 Hardware setup

Device : The experiments were carried out on a desktop computer (DESKTOP-35FIQ2R) outfitted with an Intel(R) Core(TM) i9-9900K CPU @ 3.60 GHz. This 8-core CPU provided ample computational capability for the data preparation and model training requirements.

RAM : 32 GB of RAM was allotted to handle huge batch processing and real-time data augmentations without creating memory constraints.

DISK : The dataset and models were kept on a 1 TB SSD, which allowed for fast read/write performance while loading data batches and model checkpoints.

The DESKTOP-35FIQ2R enabled the training of the model in a realistic timescale, while the RAM and SSD configuration ensured smooth data handling during training, particularly for big datasets and real-time augmentations.

4.7 Experimental Result Analysis of Google Net

For landslide detection and prediction initial stage is data collection. We consider the Bijie dataset for experimental results. This set of images contains two classes of data one is landslide and the other is non-landslide. Google Collaboratory is used for Python code. We began our study by assessing the performance of GoogleNet, a popular convolutional neural network design, for landslide detection. The model was trained on a dataset of landslide and non-landslide images. First, we apply GoogleNet as the backbone in the proposed model and then also assess the performance of the proposed model with and without a series combination of attention mechanism. The performance was evaluated using a variety of hyperparameter combinations which includes:

- Learning rate= 0.00001 to 0.00005
- Epoch= 3 to 9
- Batch Size = 32 and 64
- Weight Decay= $1e-4$
- Optimizer = Adam

These decisions were made based on preliminary testing, which revealed that a moderate learning rate and a reasonably high batch size were optimal for balancing computing efficiency with model convergence.

4.7.1 GoogleNet Experimental result without Attention Mechanism

During the training process, the model was evaluated using two critical metrics: training loss and validation loss. In addition, the Validation Accuracy was recorded to determine the model's capacity to generalize to previously unseen data.

Initially, we considered batch size 64, learning rate 0.00005, and epoch 9 for GoogleNet. Table 4.2 presents the training and validation outcomes from the 9 epochs. The training loss, validation loss, and validation accuracy all consistently improve as the number of epochs increases.

Table 4.2 Training and validation outcomes from the 9 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00002	0.6953	0.6760	0.6501
2	0.00004	0.6787	0.6563	0.7132
3	0.00005	0.6471	0.6008	0.8047
4	0.00004	0.5978	0.5491	0.8990
5	0.00004	0.5440	0.5128	0.8979
6	0.00002	0.4851	0.4881	0.9040
7	0.00001	0.4767	0.4772	0.8934
8	0.00000	0.4506	0.4741	0.9062
9	0.00000	0.4479	0.4751	0.9062

With reference to the accuracy progression, the accuracy begins at 0.6501 in the first epoch and improves significantly over the following few epochs, reaching 0.8990 by the fourth epoch. After epoch 4, the accuracy varies slightly but stays close to 0.90, with very minor improvements between epochs 5 and 9. This indicates that the model has converged and no longer learns meaningfully from the data. The model's final accuracy of 0.9062 for epochs 8 and 9 indicates that it is performing consistently and is unlikely to benefit from additional epochs. This implies that the model successfully

learned the underlying patterns in the data during the early phases of training. Change in accuracy with respect to epoch is shown in Figure 4.15(a)

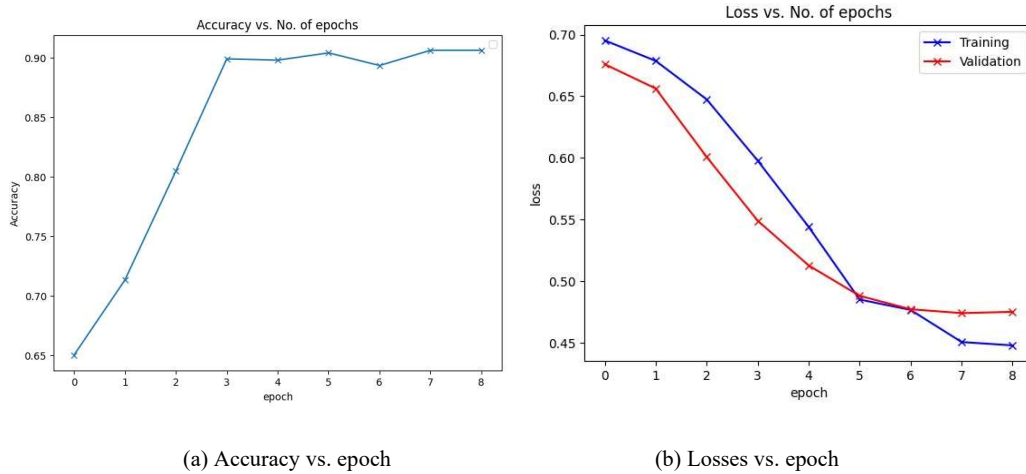


Figure 4.15 Performance visualization of GoogleNet architecture without attention for epoch 9

Figure 4.15(b) presents the losses throughout nine epochs. The training loss steadily declines, as does the validation loss, until about the 5th epoch, when both losses plateau. Training loss starts at 0.6953 and gradually decreases with increasing epoch and at epoch 9 it is 0.4479. Validation loss is 0.6760 at epoch 1 and gradually decreases with increasing epochs and at epoch 9 it is 0.4751. After the 5th epoch, both training and validation losses plateau. This shows that the model has learned the majority of the data's patterns, and more training yields declining returns. After this point, more training epochs are unlikely to appreciably enhance the model.

Overfitting happens when a model performs well in training but poorly in validation. This usually results in high accuracy on the training set but poor generalization to the validation or test set. Underfitting happens when a model fails to capture the underlying patterns in the data, resulting in significant training and validation losses and low accuracy.

According to the loss curve, the model exhibits no evidence of overfitting or underfitting, and the hyperparameters selected (learning rate 0.00005, batch size 64, 9 epochs, Adam optimizer) enable efficient training. However, the model's performance plateaus after the fifth epoch, implying that early halting could be used in future training

sessions to save needless computation and cut training time while maintaining accuracy.

We now consider epoch 5, batch size 64 and learning rate 0.00005, for GoogleNet. Table 4.3 presents the training and validation outcomes from the 5 epochs.

Table 4.3 Training and validation outcomes from the 5 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00004	0.6958	0.6849	0.5938
2	0.00005	0.6391	0.5848	0.8195
3	0.00003	0.5470	0.4924	0.9184
4	0.00001	0.4707	0.4698	0.9324
5	0.00000	0.4558	0.4740	0.9330

The accuracy increased from 0.5938 in 1st epoch to 0.9330 in last epoch. The steep increase from epoch 1 to epoch 3 indicates that the model quickly learned to classify the validation set more accurately. The accuracy plateaued between epochs 4 and 5, suggesting that the model may have attained its peak performance under the current configuration. Change in accuracy with respect to epoch is shown in Figure 4.16(a)

The training loss decreased consistently with each epoch, demonstrating that the model was successfully learning from the data. The validation loss initially showed a decreasing trend, indicating that the model's performance increased on the validation set until epoch 4. However, a minor increase in validation loss at epoch 5 indicates that the model may have begun to overfit slightly, as validation loss increased while training loss decreased. Figure 4.16(b) represents the losses throughout five epochs.

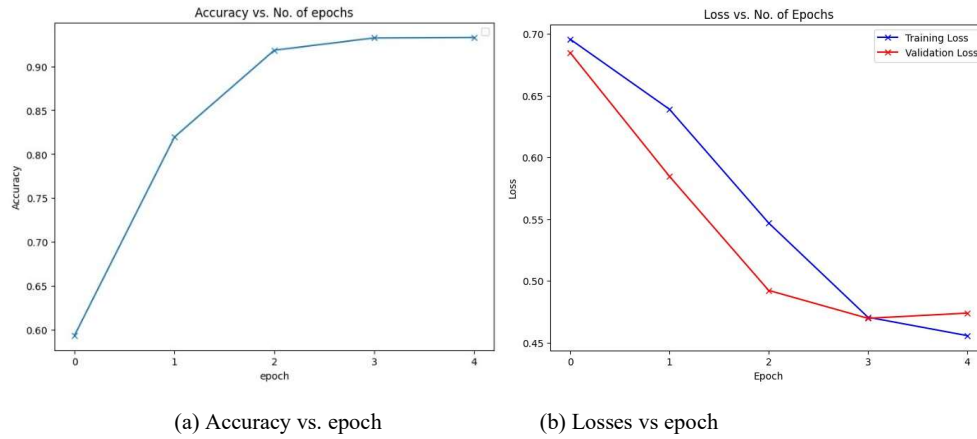


Figure 4.16 Performance visualization of GoogleNet architecture without attention for epoch 5

Although the model performed well, with a validation accuracy of more than 0.93 at the end of training, the minor rise in validation loss after epoch 4 indicates a risk of overfitting. This could mean that the model is learning to fit the training data too closely, limiting its capacity to generalize.

We change the hyperparameters for fine tuning, increase the learning rate 0.0001, epoch 9 and batch size 64. Table 4.4 presents the training and validation outcomes from the 9 epochs.

Table 4.4 Training and validation outcomes from the 9 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00003	0.6916	0.6918	0.4710
2	0.00008	0.6708	0.6520	0.6378
3	0.00010	0.5911	0.5367	0.8270
4	0.00009	0.4896	0.4502	0.9224
5	0.00007	0.4410	0.4177	0.9403
6	0.00005	0.4063	0.4083	0.9414
7	0.00002	0.3689	0.4052	0.9453
8	0.00001	0.4006	0.4096	0.9336
9	0.00000	0.3652	0.4081	0.9453

The Validation Accuracy improved significantly during the training process, starting at 0.4710 in epoch 1 and reaching a high accuracy of 0.9453 in epochs 7 and 9 as shown in Figure 4.17(a). The sharp improvement between epochs 1 and 4 indicates that the

model learned efficiently during the early phases of training. Beyond epoch 7, the validation accuracy plateaued, indicating that the model has probably converged.

The Training Loss dropped continuously during the epochs, from 0.6916 in epoch 1 to 0.3652 in epoch 9, but after epoch 6 there is fluctuation in training loss. Training loss drops at epoch 7 and increases in epoch 8. The Validation Loss declined consistently, with only slight changes after epoch 6. The lowest validation loss achieved was 0.4052 in epoch 7. The decreasing loss values indicate that the model learned to match the data effectively, although the tiny oscillations in the validation loss at later epochs may indicate that the model was achieving peak performance. Figure 4.17(b) shows the losses throughout nine epochs.

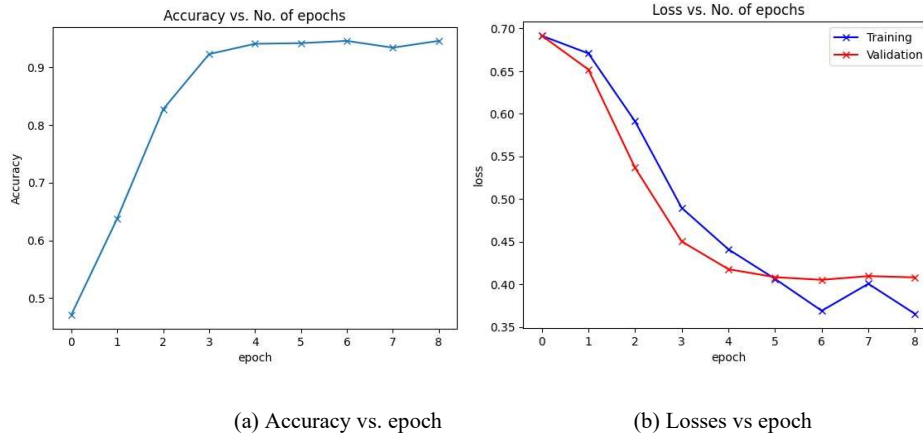


Figure 4.17 Performance visualization of GoogleNet architecture without attention for epoch 9

Despite a general downward tendency early in the training loss, the training loss increased at epoch 8 (0.4006) and fluctuated between epochs 6 to epoch 9. Although the training loss grew, the validation accuracy continued to improve, peaking at 0.9453 at epoch 9. This shows that the model was still able to generalize effectively to the validation set. A minor rise in training loss as validation accuracy improves may suggest that the model is overfitting to specific features of the training data. After epoch 6, the model may have begun to memorize specific patterns in the training data that are ineffective for generalization. This can result in higher losses without a corresponding fall in validation accuracy.

The increase in training loss after epoch 6 could indicate a few potential points of concern. We reduce the batch size from 64 to 32 and train model for different learning rate and epochs.

Now we consider the Learning rate 0.00005, batch size 32 and epoch 4. Table 4.5 presents the training and validation outcomes from the 4 epochs.

Table 4.5 Training and validation outcomes from the 4 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00005	0.6813	0.6580	0.6301
2	0.00004	0.6188	0.5504	0.8469
3	0.00001	0.5329	0.5019	0.8948
4	0.00000	0.5000	0.4966	0.8999

The model's validation accuracy improves dramatically throughout training. Starting at 0.6301 in epoch 1, accuracy gradually rises to 0.8999 by epoch 4. The significant increase in validation accuracy between epochs 1 and 2 indicates that the model quickly learns key patterns in the dataset. Change in accuracy with respect to epoch is shown in Figure 4.18(a)

The training loss begins at 0.6813 in epoch 1 and rapidly drops to 0.5000 in epoch 4. This constant drop in training loss suggests that the model is learning from the data and capturing relevant features. The validation loss decreases from 0.6580 in epoch 1 to 0.4966 at epoch 4. The validation loss follows the same declining pattern as the training loss, this indicates that the model applies well to fresh data. Figure 4.18(b) shows the losses throughout four epochs.

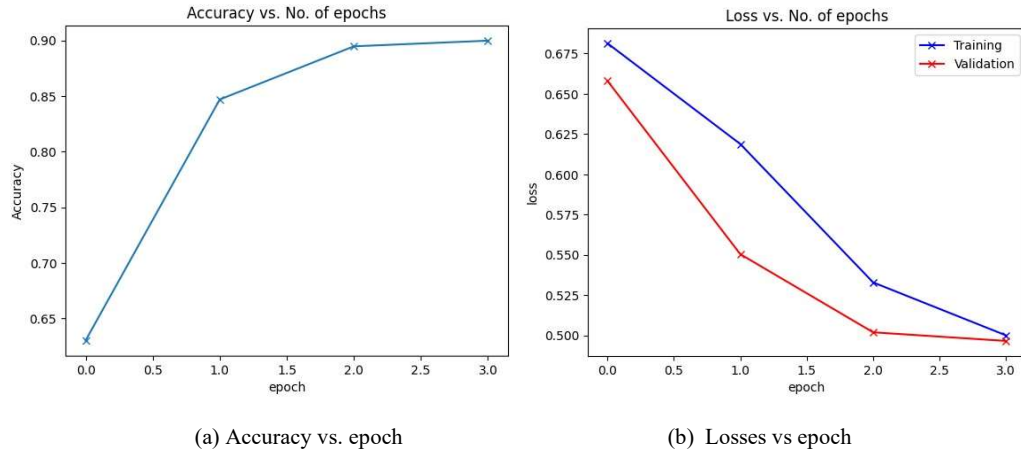


Figure 4.18 Performance visualization of GoogleNet architecture without attention for epoch 4

The close alignment of training and validation losses shows that the model is not overfitting. The validation loss stabilizes in the last two epochs, indicating that the model has reached a nearly optimal solution.

4.7.2 GoogleNet Experimental Result with Attention Mechanism

In our proposed model, we use a series combination of channel attention and spatial attention in the inception block. Initially, the training was carried out across 9 epochs with a learning rate of 0.0001 and batch size 64, for the proposed GoogleNet. Table 4.6 presents the training and validation outcomes from the 9 epochs.

Table 4.6 Training and validation outcomes from the 9 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00003	0.6959	0.6973	0.4280
2	0.00008	0.6660	0.6560	0.6161
3	0.00010	0.5956	0.5591	0.7812
4	0.00009	0.4767	0.4573	0.8968
5	0.00007	0.4334	0.4074	0.9481
6	0.00005	0.4180	0.3941	0.9648
7	0.00002	0.4019	0.4002	0.9336
8	0.00001	0.3972	0.3989	0.9492
9	0.00000	0.3894	0.3895	0.9648

The model's validation accuracy increases dramatically, from 0.4280 in epoch 1 to 0.9648 in epoch 9 as shown in Figure 4.19(a). The model is expected to develop quickly in the early epochs as it collects essential properties from the training dataset. The highest validation accuracy is attained at epoch 6, with a value of 0.9648, which remains constant until the final epoch. After epoch 5, there are modest swings in accuracy, indicating that the model may have hit peak performance with the current settings.

The training and validation losses reduce gradually and consistently during the training procedure as shown in Figure 4.19(b). At epoch 1, the training loss is 0.6959, and the validation loss is somewhat greater at 0.6973. However, as training advances, both losses considerably decrease, with values of 0.3894 (training) and 0.3895 (validation) by epoch 9. Around epoch 6, there is a little rise in training loss while validation loss decreases. However, this is not a strong evidence of overfitting.

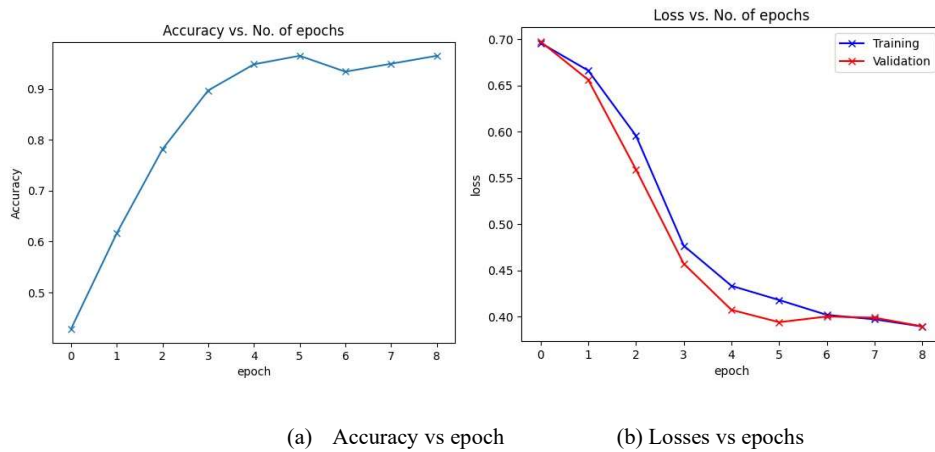


Figure 4.19 Performance visualization of GoogleNet architecture with attention for epoch 9

The modest drop after epoch 5 could indicate slight overfitting, but overall accuracy remains excellent. This implies that, while the model is doing well, it may benefit from modifying hyperparameters such as the learning rate, batch size and epochs, approaches to reduce overfitting tendencies in further experiments.

The training was carried out across 9 epochs with a learning rate of 0.0001. The results show a constant improvement in the training and validation performance. We now consider epoch 9, batch size 32 and learning rate 0.00005, for the proposed GoogleNet. Table 4.7 presents the training and validation outcomes from the 9 epochs.

Table 4.7 Training and validation outcomes from the 9 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00002	0.7223	0.7200	0.4727
2	0.00004	0.6707	0.6240	0.6669
3	0.00005	0.5270	0.4826	0.8320
4	0.00004	0.4038	0.4074	0.8477
5	0.00004	0.3297	0.3320	0.8555
6	0.00002	0.2559	0.2905	0.8873
7	0.00001	0.2681	0.2667	0.9129
8	0.00000	0.2530	0.2610	0.9129
9	0.00000	0.2576	0.2643	0.9258

The validation accuracy increased from 0.4727 in the first epoch to 0.9258 in epoch 9. This increase indicates that the model's predictions improved over time, particularly after epoch 5, when it passed the 0.85 threshold. Change in accuracy with respect to epoch is shown in Figure 4.20(a)

The training loss began at 0.7223 in epoch 1 and decreased dramatically to 0.2576 in epoch 9. This continuous drop indicates that the model was successfully minimizing loss and learning patterns from the training data. The validation loss decreased significantly from 0.7200 in epoch 1 to 0.2643 by epoch 9, indicating the model's improved ability to generalize well on the validation data. Lower validation loss, particularly after epoch 6, indicates high generalization. Figure 4.20(b) shows the Training and validation loss throughout nine epochs.

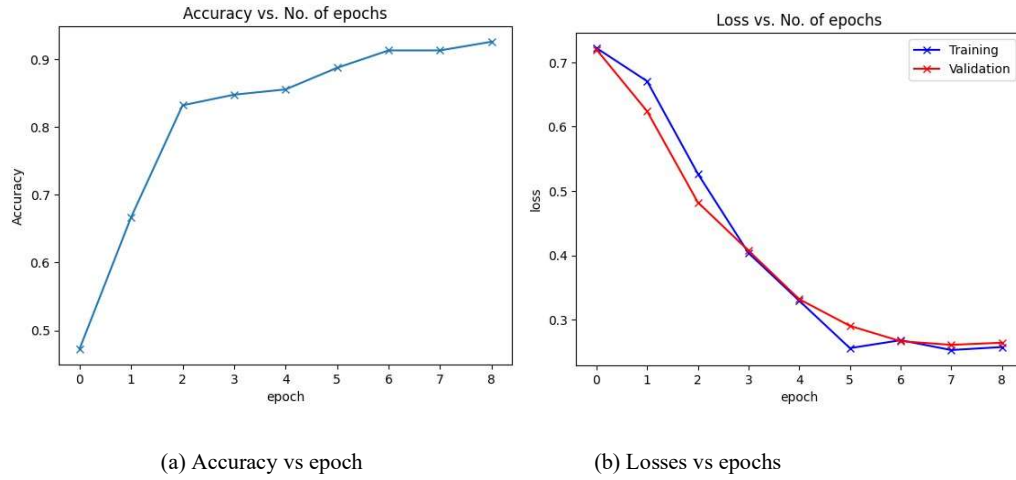


Figure 4.20 Performance visualization of GoogleNet architecture with attention for epoch 9

The progressive modification in learning rate has a good effect on the model's training dynamics. The model achieved an acceptable accuracy of 0.8555 by epoch 5 and improved further to 0.9258 by epoch 9. The Adam optimizer with a batch size of 32 and the gradual adaptation of the learning rate allowed for consistent training without evidence of overfitting.

Now we consider Batch size 32, learning rate 0.0005 and 7 epochs to train the proposed model with googleNet as backbone. Table 4.8 presents the training and validation outcomes from the 7 epochs.

Table 4.8 presents the training and validation outcomes from the 7 epochs.

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00002	0.6901	0.6758	0.5179
2	0.00005	0.6581	0.6011	0.8960
3	0.00005	0.5718	0.5101	0.8999
4	0.00003	0.4885	0.4666	0.8935
5	0.00002	0.4498	0.4364	0.9369
6	0.00000	0.4202	0.4323	0.9413
7	0.00000	0.4248	0.4301	0.9592

The validation accuracy increased significantly, from 0.5179 at epoch 1 to 0.9592 at epoch 7. This significant rise in accuracy shows that the model is generalizing

effectively to previously unseen validation data, with performance steadily rising as training progressed as the Figure 4.21(a)

Both training and validation loss reduce steadily over time. The training loss started at 0.6901 in the first epoch and gradually decreased to 0.4248 by epoch seven. Similarly, the validation loss decreased from 0.6758 to 0.4301 from epoch one to epoch seven. This suggests that the model is boosting its learning capacity while avoiding overfitting or considerable underfitting. Figure 4.21(b) shows losses throughout seven epochs.

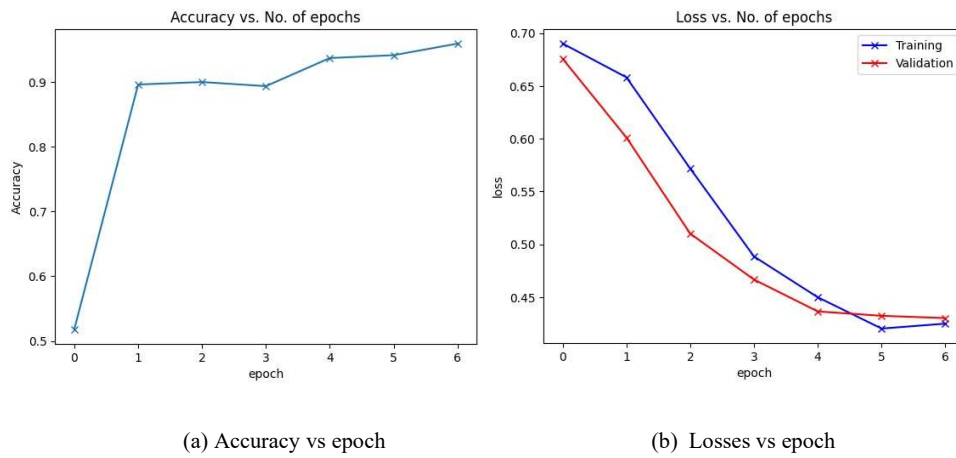


Figure 4.21 Performance visualization of GoogleNet architecture with attention for epoch 7

Overall, the model has good learning ability and performance. The decrease in training and validation loss, combined with the constant increase in validation accuracy, demonstrates that the hyperparameters: learning rate, batch size, and optimizer, are properly set. The results imply that the model is neither overfitting nor underfitting, as it retains good accuracy while minimizing losses.

4.7.3 Performance of GoogleNet

During the experiment we conducted, it is observed that, GoogleNet without attention performed reasonably well; however, with a consistent decrease in both training and validation losses. Although accuracy values improved across epochs, they did not reach the levels seen when attention mechanisms were implemented. The model's learning was consistent, although there were times when the validation loss plateaued, indicating that the model may have struggled to grasp intricate spatial dependencies in the data.

When attention mechanisms were introduced, the model demonstrated a faster reduction in both training and validation loss. This improvement can be due to attention mechanisms, which assist the model in focusing on the most relevant elements of the input data. As a result, the model improves its generalization capabilities, resulting in increased validation accuracy and lower loss.

A comparative result of both with attention or without attention GoogleNet is shown in table 4.9

Table 4.9 Comparative result of GoogleNet with or without attention mechanism

Sr. No	With or without attention	Learning rate	Number of Epoch	Training Loss	Validation Loss	Accuracy
1	No	0.00005	9	0.4479	0.4781	0.9062
2	No	0.00005	5	0.4558	0.4740	0.9330
3	No	0.0001	9	0.3652	0.4081	0.9453
4	No	0.00005	4	0.5000	0.4966	0.8990
5	Yes	0.0001	9	0.3894	0.3895	0.9648
6	Yes	0.00005	9	0.2576	0.2643	0.9258
7.	Yes	0.00005	7	0.4248	0.4301	0.9592

The addition of a series combination of channel and spatial attention mechanisms considerably enhanced GoogleNet's performance in terms of both accuracy and loss. The computational complexity is measured in terms of total floating point operations (FLOPS). The computational complexity of GoogleNet is 52712300. The improved models were able to generalize better to validation data while avoiding overfitting and underfitting, suggesting that attention mechanisms offer significant advantages when working with complex geographical information.

4.8 Result Analysis of ResNet 101

Now we continue our study by assessing the performance of ResNet 101, a popular convolutional neural network design, for landslide detection. The model was trained on a dataset of landslide and non-landslide images. Initially we use ResNet101 as the backbone network in the proposed model and then also assess the performance of

the proposed model with and without a series combination of attention mechanism. The performance was evaluated using a variety of hyperparameter combinations

- Learning rate= 0.00001 to 0.00005
- Epoch= 3 to 9
- Batch Size = 32 and 64
- Weight Decay= $1e-4$
- Optimizer = Adam

These decisions were made based on preliminary testing, which revealed that a moderate learning rate and a reasonably high batch size were optimal for balancing computing efficiency with model convergence.

4.8.1 ResNet 101 Experimental Result without AttentionMechanism

During the training process, the model was evaluated using training loss and validation loss. In addition, the Validation Accuracy was recorded to determine the model's capacity to generalize to previously unseen data.

Initially, we considered batch size 64, learning rate 0.00005, and epoch 7 for ResNet101. Table 4.10 presents the training and validation outcomes from the 7 epochs. The training loss, validation loss, and validation accuracy all consistently improve as the number of epochs increases.

Table 4:10 Training and validation outcomes from the 7 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00002	0.6883	0.6834	0.4799
2	0.00005	0.5971	0.5411	0.8108
3	0.00005	0.4236	0.4010	0.9676
4	0.00003	0.3520	0.3598	0.9676
5	0.00002	0.3561	0.3651	0.9648
6	0.00000	0.3567	0.3678	0.9609
7	0.00000	0.3598	0.3768	0.9492

The validation accuracy improved significantly, starting at 0.4799 and rising to 0.9676 by the third epoch. This increase indicates that the model was successfully learning to classify or predict unknown data. The accuracy stabilized around 0.9648 to 0.9676 during the middle epochs (4th and 5th), indicating the model reached a good level of generalization. In the later epochs 6th and 7th, validation accuracy decreased slightly, reaching 0.9492 by the end. This reduction, together with the increasing validation loss, shows possible overfitting, in which the model's performance on unseen data deteriorated marginally despite continuing training. Figure 4.22(a) shows accuracy curve with respect to epoch.

The training loss reduced consistently in the early stages, beginning at 0.6883 and reaching a low of 0.3520 by the 4th epoch. This pattern suggests that the model learnt well and minimized error on the training data. Similarly, the validation loss decreased significantly from 0.6834 in the initial stage to 0.3598 in the fourth epoch. This consistent decrease in both training and validation losses indicates that the model generalized well to the validation data, with no obvious symptoms of overfitting or underfitting. Starting with the 5th epoch, the validation loss began to climb slightly, whereas the training loss remained modest. By the last epoch, the validation loss had increased to 0.3768, indicating the possibility of overfitting, in which the model was slightly overtuned to the training data patterns. Figure 4.22(b) present training and validation losses

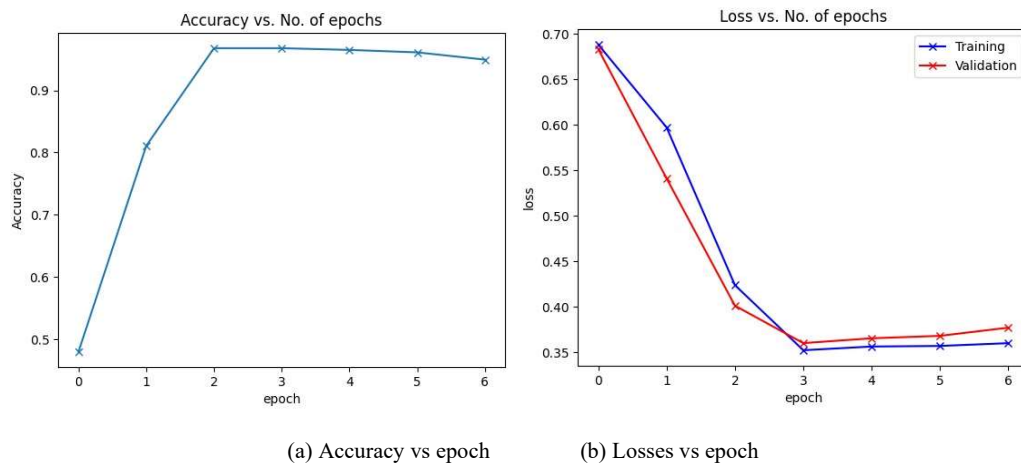


Figure 4.22 Performance visualization of ResNet101 architecture without attention for epoch 7

Overall, the model displayed effective learning and generalization from the early to middle epochs. However, the increased validation loss and a minor decline in accuracy near the conclusion indicate early indicators of overfitting. These findings suggest that future training iterations should explore using early stopping or regularization procedures to maintain high performance and avoid overfitting.

Now we change a number of epochs to 5 and the learning rate 0.00005 with batch size 32. Table 4.11 presents the training and validation outcomes from the 5 epochs

Table 4:11 Training and validation outcomes from the 5 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00004	0.6842	0.6237	0.6741
2	0.00005	0.4928	0.3872	0.9362
3	0.00003	0.3868	0.3694	0.9541
4	0.00001	0.3703	0.3668	0.9585
5	0.00000	0.3431	0.3642	0.9585

Validation accuracy increased significantly from 0.6741 in the 1st epoch to 0.9362 in the 2nd epoch. This shows that the model was able to learn and adapt fast, as evidenced by its good performance on validation data early in training. The accuracy increased further, reaching 0.9541 in the 3rd epoch before stabilizing at 0.9585 in the 4th and 5th epochs. This stability shows that the model's generalization skills were intact, resulting in consistent and accurate predictions across the validation dataset. Figure 4.23(a) present accuracy with respect to epoch

The training loss decreased consistently over the epochs, starting at 0.6842 in the 1st epoch and dropping to 0.3431 by the 5th epoch. This reduction suggests that the model was gradually learning to minimize errors on the training dataset. The validation loss fell significantly from 0.6237 in the 1st epoch to 0.3668 in the 4th epoch. This downward trend, combined with the decrease in training loss, indicates that the model successfully generalized to the validation dataset throughout the early and middle epochs. By the 5th epoch, the validation loss was 0.3642, which was slightly lower than the previous epoch. This stabilization indicates that the model has attained a near-optimal level of

performance without further significant improvements. Figure 4.23(b) presents training and validation losses.

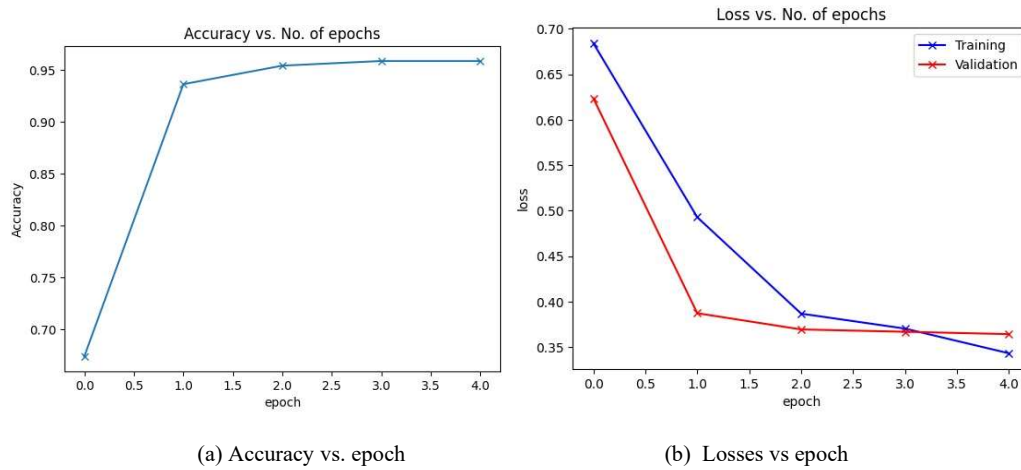


Figure 4.23 Performance visualization of ResNet101 architecture without attention for epoch 5

The training loss gradually declines from 0.6842 to 0.3431, and the validation loss follows a similar downward pattern. The rapid improvement in validation accuracy from 0.6741 to over 0.9362 by the 2nd epoch and its stabilization at around 0.9585, indicate that the model efficiently learned the patterns from the data without underfitting. The stability of the validation loss from 0.3694 to 0.3642 across the last epochs and the consistently high validation accuracy indicate that the model generalizes well, without showing a trend of memorizing the training data at the expense of performance on unseen data. The use of a gradually reducing learning rate helped the model to converge effectively. The lack of substantial variations in validation metrics indicates that the model did not show evidence of overfitting, and future training could use similar tactics for optimal performance.

Now we change number of epochs to 3 and the learning rate of 0.00005 with batch size 32. Table 4.12 presents the training and validation outcomes from the 3 epochs

Table 4:12 Training and validation outcomes from the 3 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00005	0.6431	0.5452	0.8119
2	0.00002	0.4427	0.4217	0.9062
3	0.00000	0.3872	0.3887	0.9413

Initially, the model had a somewhat high training loss of 0.6431 and a validation loss of 0.5452, with a validation accuracy of 0.8119. This shows that the model was still learning how to identify patterns in the data. The training loss dropped further to 0.3872, while the validation loss decreased marginally to 0.3887. The validation accuracy increased to 0.9413. The continuous decrease in both losses as accuracy increases implies that the model is learning efficiently and generalizing successfully. Figure 4.24(a) show accuracy with respect to epochs and Figure 4.24(b) shows the Training and validation loss throughout three epochs.

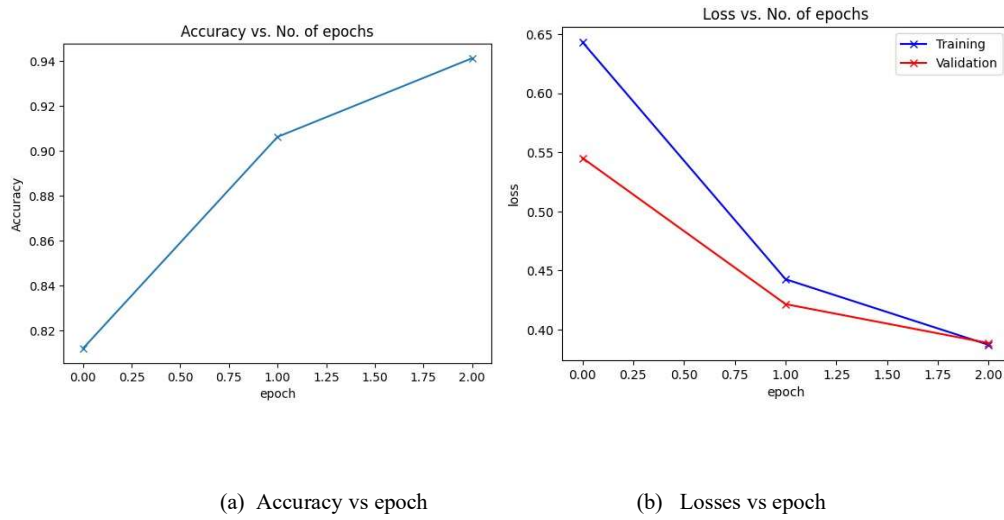


Figure 4.24 Performance visualization of ResNet101 architecture without attention for epoch 3

The training findings show that effective learning occurs over all three epochs. Both training and validation losses dropped, but validation accuracy climbed from 0.8119 to 0.9413, indicating a constant improvement in the model's performance. The decrease in both losses with no substantial divergence indicates that the model has not yet overfitted the training data. Consistent gains in validation accuracy show that the model learns and generalizes well.

4.8.2 ResNet 101 Experimental Result with Attention Mechanism

In our proposed model we use a series combination of channel attention and spatial attention in the Residue block. Initially, we start with batch size 32, a learning rate of 0.00005 for epoch 9. Table shows the training and validation outcomes from the 9 epochs. Table 4.13 presents the training and validation outcomes from the 9 epochs.

Table 4:13 Training and validation outcomes from the 9 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00002	0.3672	0.3767	0.9413
2	0.00004	0.3519	0.3930	0.9145
3	0.00005	0.3623	0.3895	0.9286
4	0.00004	0.3575	0.3560	0.9598
5	0.00004	0.3458	0.3687	0.9509
6	0.00002	0.3424	0.3546	0.9643
7	0.00001	0.3323	0.3618	0.9554
8	0.00000	0.3197	0.3385	0.9777
9	0.00000	0.3198	0.3476	0.9732

The model begins with a high validation accuracy of 0.9413, demonstrating great initial learning capabilities in epoch 1. A slight decrease to 0.9145 in epoch 2 suggests possible overfitting or noisy data, but it recovers to 0.9286 in epoch 3. From epoch 4 onward, the model steadily improves, reaching a peak accuracy of 0.9777 in epoch 8 before dropping to 0.9732 in epoch 9. This shows a robust learning process, with minor improvements even in later epochs. Figure 4.25(a) shows accuracy with respect to epoch.

The training loss decreases from 0.3672 to around 0.3519, with modest variability at epoch 3. Validation loss behaves similarly, initially falling but then slightly increasing in epochs 2 and 3, indicating some instability. After epoch three, both training and validation losses stabilize and gradually improve. By the last epoch, the training loss is 0.3197, while the validation loss is 0.3476, showing effective learning without severe overfitting. shows the Training and validation loss throughout nine epochs. Figure 4.25(b) shows the Training and validation loss throughout nine epochs.

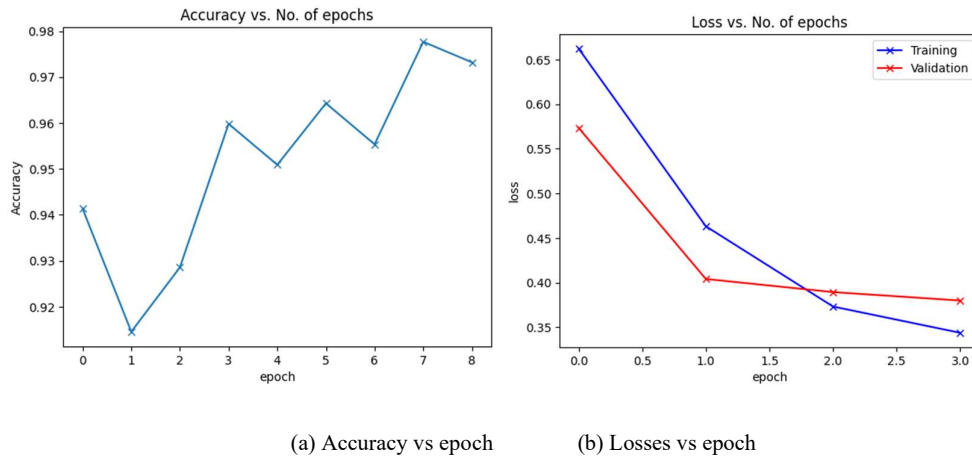


Figure 4.25 Performance visualization of ResNet101 architecture with attention for epoch 9

The modest change in validation accuracy between epochs 2 and 3 suggests that the model encountered tough patterns in the data but reacted well. The overall trend indicates effective generalization, which is critical for consistent performance on unknown data.

From epochs 1 to 3, the training loss decreases, while the validation loss increases slightly from 0.3767 to 0.3930 in epoch 2 and 0.3895 in epoch 3. This could indicate that the model is beginning to overfit, as it performs well on the training set but struggles significantly on the validation set. After epoch 4, both training and validation losses normalize, and validation accuracy stays consistently high, more than 0.95. This shows that the model can generalize effectively, maybe due to changes in the learning rate and other regularization strategies. There is no strong evidence of underfitting, since the validation accuracy starts at 0.9413 in the first epoch and improves further, it indicates that the model is learning efficiently and is not underfitting.

To mitigate overfitting we change the hyperparameters, Now we set the Learning rate to 0.0005 batch size 64 and epoch 5. Table 4.14 presents the training and validation outcomes from the 5 epochs.

Table 4:14 Training and validation outcomes from the 5 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00004	0.6775	0.6652	0.7048
2	0.00005	0.5702	0.4595	0.7208
3	0.00003	0.4176	0.3807	0.9531
4	0.00001	0.3824	0.3653	0.9570
5	0.00000	0.3478	0.3695	0.9609

In epoch 1 the training loss is large is 0.6775 , as is the validation loss is 0.6652, indicating initial underfitting. The validation accuracy is low 0.7048, indicating that the model has not yet learned patterns efficiently. The training and validation losses decrease significantly between Epochs 1 and 3 . The validation accuracy increases dramatically from 0.7208 to 0.9531 at Epoch 3, shows that the model is starting to generalize effectively after few changes. The learning rate adjustment is beneficial in directing the model to improved performance, lowering losses, and increasing accuracy. Figure 4.26(a) shows accuracy with respect to epoch. And Figure 4.26(b) shows the Training and validation loss throughout nine epochs.

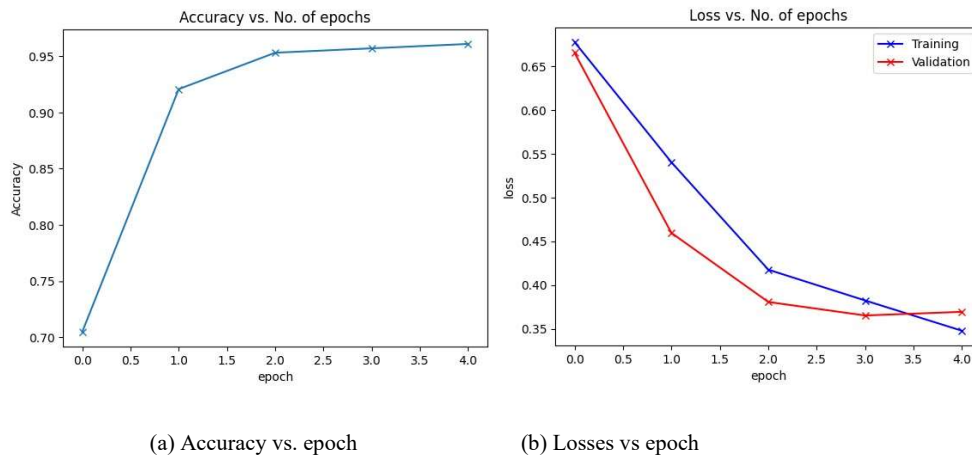


Figure 4.26 Performance visualization of ResNet101 architecture with attention for epoch 5

Now we set the Learning rate to 0.0001 batch size 32 and epoch 5 and analys the result of model. Table 4.15 presents the training and validation outcomes from the 5 epochs.

Table 4:15 Training and validation outcomes from the 5 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00007	0.6512	0.4907	0.9177
2	0.00010	0.4050	0.3491	0.9624
3	0.00006	0.3494	0.3823	0.9636
4	0.00002	0.3670	0.3495	0.9643
5	0.00000	0.3552	0.3409	0.9770

The validation accuracy improves dramatically, starting at 0.9177 and rising to 0.9770 after the last learning rate adjustment. This shows that the learning rate tweaks improved the model's accuracy, allowing for more effective optimization. The steady accuracy of 0.96 suggests stable performance, with a peak of 0.9770 indicating strong generalization ability. Figure 4.27(a) shows accuracy with respect to epoch.

The statistics reveal a significant reduction in both training and validation losses from the beginning, indicating excellent model training. The training loss decreases from 0.6512 to around 0.3552. Although the training loss fluctuated slightly at epoch 3, it generally trended downward, indicating that the model is still learning successfully. The validation loss exhibits a pattern of small variations before decreasing at the conclusion, indicating successful generalization. Figure 4.27(b) shows the Training and validation loss throughout five epochs.

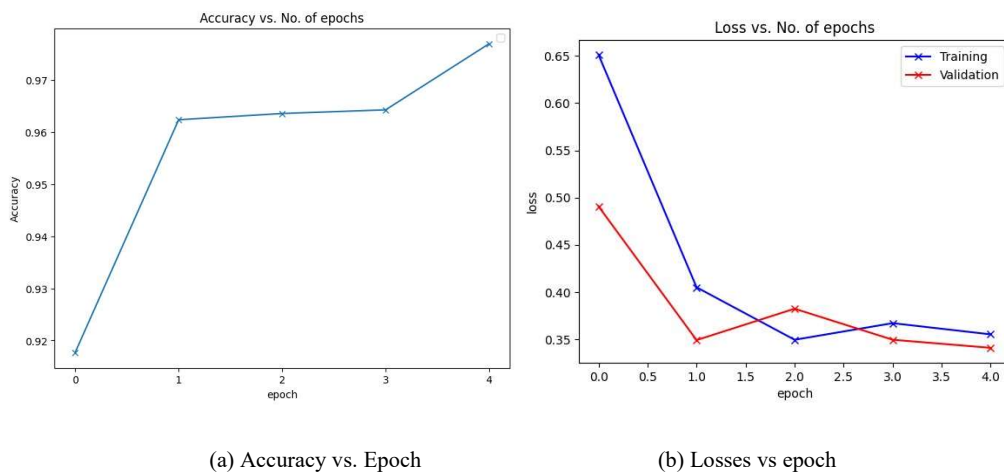


Figure 4.27 Performance visualization of ResNet101 architecture with attention for epoch 5

The training loss starts high at 0.6512 and steadily lowers, as does the validation loss. There is no evident sign of overfitting, as the validation accuracy stays good and even increases near the end. The training and validation losses are generally near to each other, with no significant differences, showing that the model generalizes effectively to new data. Minor changes in validation loss in epochs 2 to 3, do not indicate significant overfitting. Instead, these could represent expected fluctuations during training.

4.8.3 Performance of ResNet101

The models without attention attained a maximum accuracy of 0.9585 over five epochs, with validation losses ranging from 0.3642 to 0.3887. The training losses showed small oscillations, indicating steady learning. However, the validation loss for the model trained for three epochs was slightly larger than the others, indicating that performance is unstable with fewer epochs. This mismatch may highlight the risk of overfitting, as evidenced by the three-epoch model's lower accuracy of 0.9413, which may not have properly captured the underlying data patterns.

The models that included attention processes performed better overall. After five epochs, the best-performing model had 0.9770 accuracy with a validation loss of 0.3409. Notably, the validation loss fell consistently across all epochs, demonstrating that the attention modules helped the model focus on critical features for landslide identification. The model trained for nine epochs also performed well, with an accuracy of 0.9732, highlighting the value of attention in improving model robustness.

As comparative result of both with attention or without attention, ResNet101 is shown in the table 4.16

Table 4:16 comparative result of ResNet101 with attention or without attention

Sr. No	With or without attention	Learning rate	Number of Epochs	Training Loss	Validation Loss	Accuracy
1	No	0.00005	7	0.3598	0.3768	0.9492
2	No	0.00005	5	0.3431	0.3642	0.9585
3	No	0.00005	3	0.3872	0.3887	0.9413
4	Yes	0.00005	9	0.3198	0.3476	0.9732

5	Yes	0.00005	5	0.3478	0.3695	0.9609
6	Yes	0.0001	5	0.3552	0.3409	0.9770

The results clearly demonstrate that introducing attention mechanisms to ResNet101 improves its learning capacities and performance in landslide detection tasks. The computation complexity is assessed in terms of total FLOPS. The computational complexity of ResNet101 is 51653608. Attention mechanisms are a crucial addition to the model, allowing it to focus on essential spatial elements and improve overall resilience.

4.9 Result Analysis of ResNet 50

We now proceed with our investigation by evaluating ResNet50, a well-known convolutional neural network architecture to performance identification of landslide identification. The model was trained on a dataset of landslide and non-landslide images. Initially, we use ResNet50 as the backbone network in the proposed model and then also assess the performance of the proposed model with and without a series combination of attention mechanism. The performance was evaluated using a variety of hyperparameter combinations

- Learning rate= 0.00001 to 0.00005
- Epoch= 3 to 9
- Batch Size = 32 , 64,256
- Weight Decay= 1e-4
- Optimizer = Adam

These decisions were made based on preliminary testing, which revealed that a moderate learning rate and a reasonably high batch size were optimal for balancing computing efficiency with model convergence.

Initially we consider Batch size 32 , learning rate 0.00005 and epoch 7 and analys the result of model.

4.9.1 ResNet 50 Experimental Result without Attention Mechanism

During the training process, the model was evaluated using training loss and validation loss. In addition, the Validation Accuracy was recorded to determine the model's capacity to generalize to previously unseen data.

Initially, we considered batch size 64, learning rate 0.00005, and epoch 7 for ResNet50. Table 4.17 presents the training and validation outcomes from the 7 epochs.

Table 4:17 Training and validation outcomes from the 7 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00002	0.6757	0.6275	0.6710
2	0.00005	0.5002	0.5944	0.9338
3	0.00005	0.3675	0.3684	0.9504
4	0.00003	0.3411	0.3871	0.9338
5	0.00002	0.3325	0.3696	0.9485
6	0.00000	0.3315	0.3931	0.9228
7	0.00000	0.3364	0.3815	0.9320

Training loss consistently decreases from 0.6757 in epoch 1 to 0.3325 in epoch 5. However, it grows slightly from epoch 6th to 7th. Validation loss drops dramatically from 0.6275 to 0.3684 by epoch 3 but then increases in epochs 4, 6, and 7. This inconsistency may imply modest overfitting, in which the model's performance on the validation set does not improve consistently. Validation accuracy begins at 0.6710 in 1st epoch, peaks at 0.9504 in 3rd epoch, and then dipping to 0.9338 before settling around 0.9320 in 7th epoch. Epoch 3 has the best performance 0.9504 accuracy as well as the lowest validation loss 0.3684. However, the variations in later epochs indicate that the model is struggling to maintain stability, which could be related to overfitting. Figure 4.28(a) shows accuracy with respect to epochs and Figure 4.28(b) presents training and validation loss.

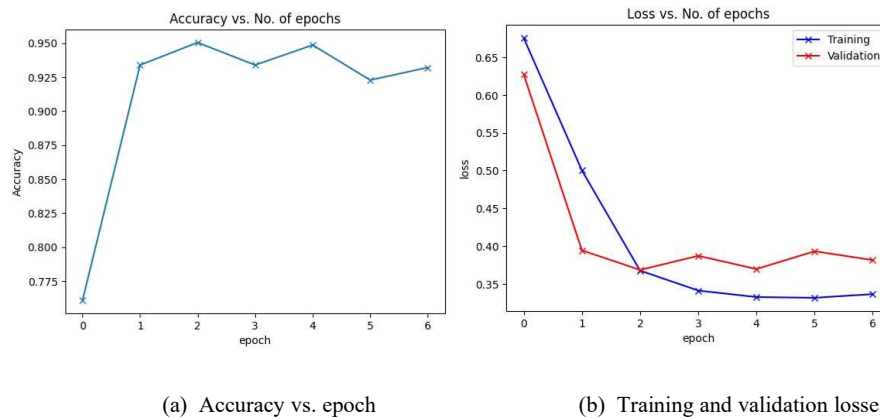


Figure 4.28 Performance visualization of ResNet50 architecture without attention for epoch 7

The results show an early period of effective learning in which the model rapidly improves, peaking at epoch 3 with a validation accuracy of 0.9504 and the lowest validation loss (0.3684). However, succeeding epochs exhibit swings in validation performance, including higher loss and slightly worse accuracy. This signals possible overfitting, in which the model begins to acquire patterns specific to the training data rather than generalizable features.

Now we consider batch size 32, a learning rate of 0.00002 and epoch 5. Table 4.18 presents the training and validation loss for epoch7.

Table 4:18 Training and validation outcomes from the 5 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00001	0.6856	0.6614	0.6224
2	0.00002	0.5861	0.4904	0.9037
3	0.00001	0.4360	0.4089	0.9592
4	0.00000	0.3878	0.4087	0.9330
5	0.00000	0.3780	0.4426	0.9107

The validation accuracy improved sharply, from 1st to 3rd epoch from 0.6224 to 0.9592, indicating effective learning and generalization. Validation accuracy peaked at 0.9592 in 3rd epoch, began to decline, falling to 0.9107 in 5th epoch. This shows that the model may be overfitting, as it continues to improve on the training data while failing to generalize on the validation set. Figure 4.29(a) presents accuracy with respect to epochs.

The model demonstrated a strong pattern of progress in the early epochs. Training loss fell dramatically from 0.6856 in 1st epoch to 0.4360 in 3rd epoch. Validation loss also dropped, from 0.6614 in epoch 1 to 0.4089 by epoch 3. The model's performance did not improve regularly from the 3rd epoch onwards. Although training loss decreased marginally, validation loss started to fluctuate, rising from 0.4087 in epoch 4 to 0.4426 in epoch 5. Figure 4.29(b) present the training and validation losses.

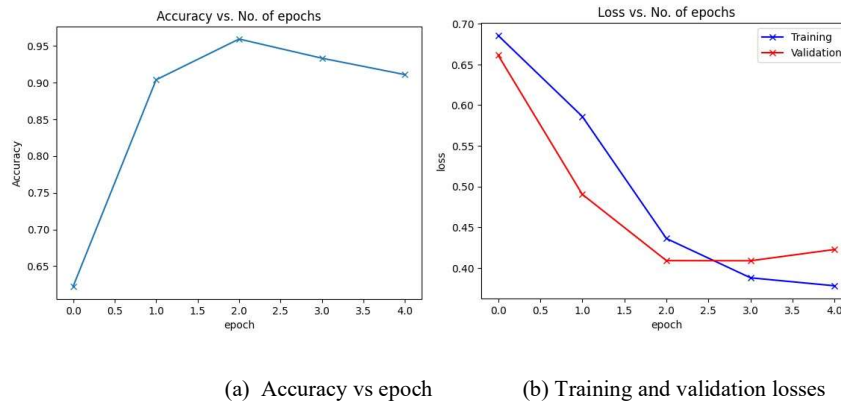


Figure 4.29 Performance visualization of ResNet50 architecture without attention for epoch 5

The model's performance analysis shows effective initial learning, with a peak validation accuracy of 0.9592 at epoch 3. However, the subsequent decrease in validation performance indicates overfitting. Future experiments should aim to validate these approaches to achieve more consistent and reliable model performance by fine-tuning the hyperparameters.

Then we consider Batch size 64, the learning rate of 0.00005 and epoch 5. Table 4.19 presents the training and validation outcomes from the 5 epochs.

Table 4:19 Training and validation outcomes from the 5 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00004	0.6551	0.5726	0.8508
2	0.00005	0.4445	0.3880	0.9449
3	0.00003	0.3608	0.3858	0.9265
4	0.00001	0.3451	0.3835	0.9228
5	0.00000	0.3494	0.3608	0.9577

Validation accuracy continuously rises from 0.8508 to 0.9577, indicating an improvement in the model's capacity to generalize to new data.

At epoch 5, the model has the highest validation accuracy (0.9577) and the lowest validation loss (0.3608), indicating that it has reached a well-balanced state between learning and generalization. The Figure 4.30(b) shows accuracy with respect to epoch.

The training loss steadily drops from 0.6551 to 0.3494, indicating that the model is picking up new information efficiently. Validation loss begins at 0.5726 and gradually drops to 0.3608, with a little increase between epochs 3 and 4. This implies slight variations, but the overall tendency is lower, indicating better generality. The Figure 4.30(a) shows the Training and validation loss throughout nine epochs

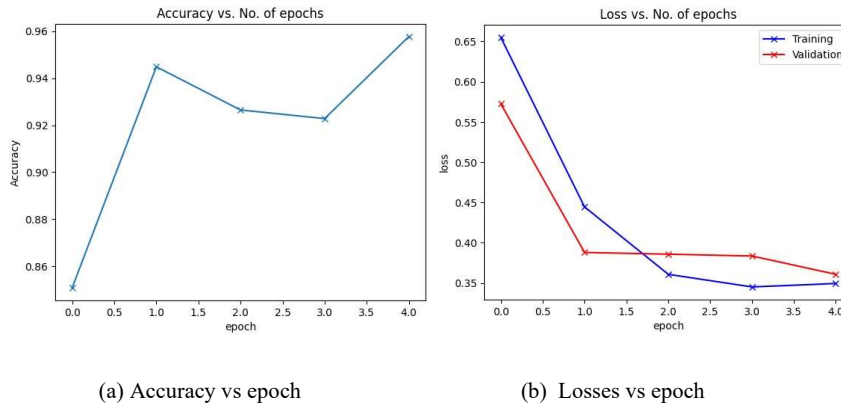


Figure 4.30 Performance visualization of ResNet50 architecture without attention for epoch5

The model's training and validation performance improved consistently across epochs. The gradual reduction in the learning rate played a crucial role in fine-tuning the model, leading to a significant decrease in training and validation loss and a steady increase in validation accuracy. Starting with an initial accuracy of 0.8508, the model improved to a high accuracy of 0.9577 by epoch 5, which coincided with the lowest validation loss observed. The slight fluctuation in validation loss around epochs 3 and 4 could be a sign of the model encountering slight overfitting, but it is mitigated as the learning rate decreases further. Since the final validation loss and accuracy are both optimal at epoch 5, it suggests that the model was able to overcome this potential overfitting and learned to generalize better.

4.9.2 ResNet 50 Experimental Result with AttentionMechanism

In our proposed model we use a series combination of channel attention and spatial attention in the inception block. Then we consider Batch size 64, the learning rate of 0.00005 and epoch 7. Table 4.20 presents the training and validation outcomes from the 7 epochs.

Table 4:20 Training and validation outcomes from the 7 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00002	0.6556	0.5451	0.8110
2	0.00004	0.5502	0.4812	0.9348
3	0.00005	0.4271	0.3364	0.9387
4	0.00003	0.3511	0.3210	0.9485
5	0.00001	0.3401	0.3241	0.9450
6	0.00000	0.3215	0.3159	0.9671
7	0.00000	0.3047	0.2891	0.9671

Validation accuracy improves from 0. 8110 in epoch 1 to 0.9671 in epoch 7, indicating a considerable improvement in the model's performance on previously unknown data. The accuracy stabilizes at 0.9671 in both epochs 6 and 7, equal to the lowest validation loss, demonstrating that the model is well-optimized. Figure 4.31(a) presents accuracy with respect to epochs.

Training loss steadily falls from 0.6556 in epoch 1 to 0.3047 in epoch 7, indicating efficient learning over the epochs. Validation loss has the same decreasing trend, beginning at 0.5451 in epoch 1 and falling to 0.2891 by epoch 7. Figure 4.31(b) shows training and validation loss.

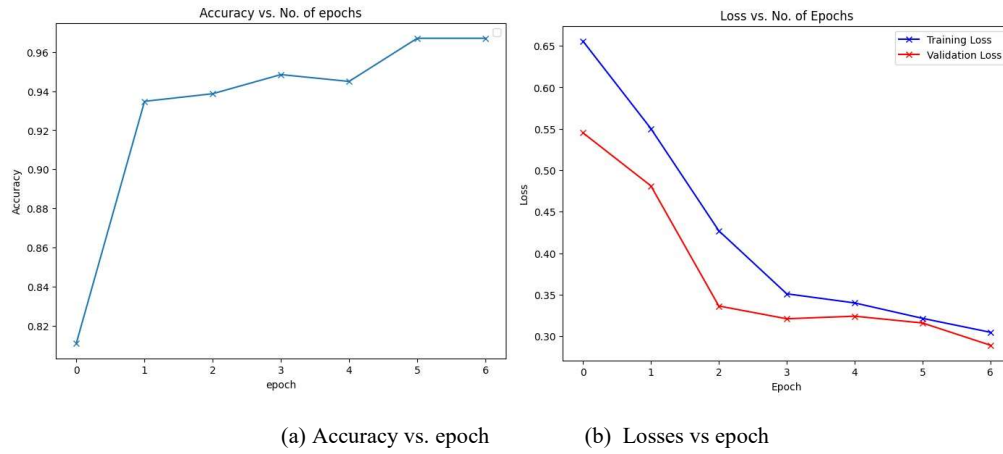


Figure 4.31 Performance visualization of ResNet50 architecture with attention for epoch 7

The training results show a well-optimized model with strategic use of learning rate modifications, resulting in increased performance over epochs. Starting with a lesser accuracy of 0.8110, the model promptly improved to a high accuracy of 0.9671 by epoch 6, and maintained this level in the subsequent epochs. This improvement is supported by a consistent decrease in both training and validation losses, indicating that the model was successfully trained to generalize without overfitting.

Then we consider Batch size 32, the learning rate of 0.00002 and epoch 5. Table 4.21 presents the training and validation outcomes from the 5 epochs.

Table 4:21 Training and validation outcomes from the 5 epochs

Epoch	Learning Rate	Training Loss	Validity Loss	Validity Accuracy
1	0.00001	0.6860	0.6087	0.8516
2	0.00002	0.4878	0.4578	0.9107
3	0.00001	0.4085	0.3985	0.9187
4	0.00000	0.3678	0.3380	0.9330
5	0.00000	0.3280	0.3020	0.9692

Validation accuracy increases from 0.8516 to 0.9692, indicating better generalization and performance on unknown data. The maximum accuracy 0.9692 at epoch 5 correlates to the lowest validation loss 0.3020, indicating robust model performance.

Training loss gradually falls from 0.6860 to 0.3280, suggesting that the model learns effectively over epochs. Validation loss follows a similar pattern, beginning at 0.6087

and decreasing to 0.3020, indicating a constant improvement in the model's capacity to generalize. The consistent drop in validation loss, particularly in later epochs, indicates that the model is not overfitting and is learning well. Figure 4.32(a) and Figure 4.32(b) shows accuracy and losses.

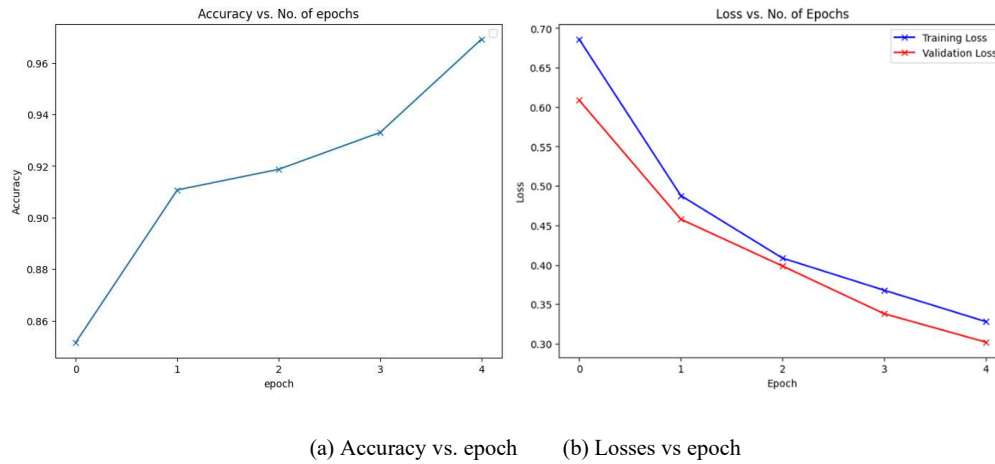


Figure 4.32 Performance visualization of ResNet50 architecture with attention for epoch 5

The results show a well-tuned training process in which changes in the learning rate successfully directed the model to an optimal state. Beginning with a validation accuracy of 0.8516, the model reached a peak accuracy of 0.9692 by epoch 5, with a consistent reduction in validation loss. This improvement indicates that the model was able to train well without overfitting, as evidenced by a gradual decline in the learning rate, particularly in the last epochs. The last epoch's performance reveals the model's strong generalization capacity, with low loss and excellent accuracy.

4.9.3 Performance of ResNet50

Models without attention mechanisms, although capable of learning effectively, may overfit to the training data. The larger validation losses indicate that, while the models are highly accurate, they may struggle with generalization, resulting in performance decreases when exposed to new, previously unseen data. This pattern highlights the possible need for further regularization strategies, such as dropout or data augmentation, to reduce overfitting in subsequent tests.

The models trained with attention mechanism showed consistent and strong improvement in all important measures. The attention mechanism helped the models focus more effectively on relevant features, resulting in improved learning and generalization.

As comparative result of both with attention or without attention, ResNet50 is shown in the table 4.22

Table 4.22 comparative result of ResNet50 with attention or without attention module

Sr. No	With or without attention	Learning rate	Number of Epoch	Training Loss	Validation Loss	Accuracy
1	No	0.00005	7	0.3364	0.3815	0.9504
2	No	0.00002	5	0.3780	0.4426	0.9592
3	No	0.00005	5	0.3494	0.3608	0.9577
4	Yes	0.00005	7	0.3041	0.2891	0.9671
5	Yes	0.00002	5	0.3280	0.3020	0.9692

The comparison clearly shows that attention mechanisms enhance the performance of deep learning models, leading to lower training and validation losses and higher validation accuracy. The computation complexity is assessed in terms of total FLOPS. The computational complexity of ResNet50 is 55647616. Models without attention mechanisms, while still capable of achieving high accuracy, showed less stability and a tendency to overfit. Integrating attention mechanisms appears to help the model focus on essential features, resulting in more reliable and generalized performance.

4.10 Discussion

A comparison of GoogleNet, ResNet 101, and ResNet 50 models, both with and without attention methods, provides some key insights into their performance in landslide detection. Each model was tested using key measures like as accuracy, training and validation loss, F1 score, precision, and recall to better understand their strengths and the impact of attention mechanisms. Table 4.23 present the comparative result of GoogleNet, ResNet 101, and ResNet 50 models.

Table 4:23 Comparative result of trained Model

Model	Attention Mechanism	Learning Rate	Training Loss	Validity Loss	Accuracy	F1 Score	Precision	Recall
GoogelNet	No	0.0001	0.3652	0.4081	0.9453	0.9361	0.9435	0.9363
GoogelNet	Yes	0.0001	0.3894	0.3895	0.9648	0.9590	0.9610	0.9590
ResNet101	No	0.00005	0.3431	0.3642	0.9585	0.9590	0.9591	0.9590
ResNet101	Yes	0.00005	0.3198	0.3476	0.9732	0.9584	0.9427	0.9509
ResNet50	No	0.00002	0.3789	0.4426	0.9592	0.9083	0.9230	0.9090
ResNet50	Yes	0.00002	0.3280	0.3020	0.9692	0.9576	0.9600	0.9576

- **Model Comparison and Performance Evaluation**

ResNet101 with the Attention mechanism achieved the maximum accuracy 0.9732 and performed well across all measures. Its capacity to perform complex feature extraction tasks makes it suitable for detecting landslides, particularly when high accuracy is required. However, this paradigm necessitated additional computational resources, which could be a disadvantage for real-time or resource-limited applications. GoogelNet showed significant improvement when attention was applied, with accuracy rising to 0.9648. The model's lightweight architecture makes it a feasible option in situations where speed and cheap computing costs are more important than peak accuracy. ResNet50 with Attention, despite having a slightly lower accuracy of 0.9692 than ResNet101, this model had a balanced performance in precision of 0.9600 and recall of 0.9576 with a validation loss of 0.3020. ResNet50 with attention is also a good choice due to its high accuracy, minimal validation loss, and balanced metrics, especially when computational efficiency and generalization are important concerns.

- **Impact of Attention Mechanism**

Attention mechanism have consistently positively impacted the performance of all three models. The accuracy of each model is improved with an attention mechanism. ResNet101 achieves 0.9732 accuracy indicating that this model focuses on relevant features which leads to accurate prediction. Provide better generalization by reducing validation loss in all three models which minimizes overfitting and performs well on unseen data. These results show that attention mechanisms can significantly improve feature extraction, which is required for reliably detecting complicated patterns in landslide imagery.

The trained model is also compared with the existing models which are trained with the same and datasets. Table 4.24 shows the comparative result.

Table 4.24 Comparative results of the proposed model and existing models.

Model	DataSet	Recall	Precision	F1 Score	Accuracy
Proposed ResNet 50	Bijie Landslide Dataset	0.9576	0.9600	0.9576	0.9692
Proposed ResNet 101	Bijie Landslide Dataset	0.9509	0.9427	0.9584	0.9732
Proposed GoogleNet	Bijie Landslide Dataset	0.9590	0.9610	0.9590	0.9648
REF[45] ResNet50+Mask R-CNN	UAV dataset	0.8032	0.8615	0.8313	0.7391
REF[43] ResNet101+Mask R-CNN	UAV dataset	0.8392	0.8955	0.8664	0.7643
REF[39] ResNet101-RCNN	UAV dataset	0.87	0.93	0.90	0.90

REF [96] TransferU Net with CBAM	Bijie Landslide Dataset	0.8823	0.8724	0.8773	0.8791
REF [82] TransferU Net with CBAM	Iburi dataset	0.8170	0.7709	0.7933	0.8023

The combined analysis confirms that attention mechanisms improve model performance, especially in complex architectures like ResNet101 and ResNet50, GoogleNet. Future research could explore experimentation with different types of attention mechanisms to see if further improvements can be made.

Chapter 5 Conclusion and Future Scope

In this research work, we proposed deep learning based model for the automatic detection of landslide with three different CNN backbone models. This study describes an innovative and comprehensive technique to landslide identification and prediction that combines satellite data with sophisticated convolutional neural network (CNN) architectures, notably ResNet50, ResNet101, and GoogleNet. This study highlights the potential of remote sensing data to provide reliable, scalable landslide monitoring solutions by using high-resolution satellite images and CNNs' deep feature extraction capabilities. The findings reveal that ResNet-based models, ResNet101, outperform in landslide detection tasks, especially when supplemented with spatial and channel attention methods. These attention layers enable the models to prioritize crucial portions of images, resulting in higher detection accuracy and interpretability.

The purpose of this research work is to design an automatic CAD system for detecting and classifying landslide using Artificial Intelligence Techniques. This research provided the following contributions:

- Preprocessing is a critical step in deep learning that converts raw data into an acceptable format for model training. The quality of the preprocessing can significantly affect the model's performance, training time, and ability to generalize to new data. To process the images and increase the number of images for deep learning training and testing of the model we apply data augmentation methodologies, dimensionality reduction using PCA and its variants, and rigorous model evaluation to provide the groundwork for constructing more accurate and adaptive landslide monitoring systems.
- In this research work, a deep-learning algorithm with an attention mechanism is proposed to detect and classify landslides. Train the proposed model with a training set by using ResNet 101, ResNet 50 and GooglNet. Prestained model, used to evaluate the experimental results based on training loss, validation loss, precision, recall, F1 score and accuracy in landslide prediction. According to experimental results, ResNet 101 obtains 0.9427 precision, 0.9509 recall, 0.9584 F1 score and 0.9732 accuracy. ResNet 50 obtains 0.9600 precision, 0.9576 recall, 0.9576 F1 score and 0.9692 accuracy. GoogleNet obtains 0.9610

precision, 0.9590 recall, 0.9590 F1 score and 0.9648 accuracy. These experimental results indicate that the proposed attention module CNN model with ResNet 101 as a backbone network has high accuracy, and can be used as an effective landslide detection method to help in emergency rescue.

- In next step, adjust the hyperparameters to optimize each model. Parameters such as learning rate, batch size and number of epochs are carefully adjusted to prevent overfitting and underfitting, ensuring that the model generalizes well across data samples. Each network's performance is measured using model training and validation parameters such as training loss, validation loss, accuracy, F1 score, precision, and recall. A detailed epoch-by-epoch study reveals the learning dynamics of each model as well as the usefulness of alternative architectural solutions. Furthermore, early stopping criteria and dropout regularization techniques are used to prevent overfitting.

A comparative analysis of model performance indicates that architectural depth and attention layers influence the models' ability to reliably distinguish landslide from non-landslide regions. The findings suggest that deeper networks are better suited to challenging classification tasks like landslide detection, which require detailed patterns for accurate predictions. The combined analysis reveals that attention mechanisms boost model performance, especially in complicated designs like ResNet101, ResNet50, and GoogleNet. These approaches and findings show that incorporating satellite-based deep learning models into early warning systems has the potential to revolutionize landslide risk management by allowing for prompt and data-driven mitigation decisions.

5.1 Future Scope

The societal contribution of automatic landslide detection stems from its ability to save lives by providing early warnings and allowing for prompt evacuations in disaster-prone locations. It promotes catastrophe preparedness through effective resource allocation and community knowledge, resulting in safer living conditions. It contributes to regional stability by preserving important infrastructure and minimizing economic damage. The technology also helps to preserve ecosystems by identifying high-risk zones and minimizing environmental damage. Governments can use these tools to

create more effective land use and disaster management policies. This work of landslide detection is critical for improving public safety, lowering hazards, and boosting long-term development.

Building on the study's findings, numerous areas present intriguing possibilities for future research to improve landslide detection and prediction systems. Future research can improve and expand the capabilities of CNN-based landslide models by building on the current technique and incorporating additional data sources, methods, and technology. There are a few limitations in this work. To alleviate the restrictions future research could combine satellite image processing with meteorological data and provide more accurate understanding of landslide detection and prediction.

1. This research focuses on visual features from satellite imagery. Images used in this work provide extensive and precise spatial coverage but it does not directly incorporate crucial environmental characteristics and parameters such as soil moisture, precipitation, and seismic activity which have a substantial impact on landslide susceptibility. Soil moisture has an impact on soil stability and has valuable insights into landslides but can not be detected only through image analysis. The image-based analysis is also not able to detect precipitation and seismic activity as they change over time and space. Hence, these environmental parameters are not incorporated in the proposed model. However, the accuracy of the proposed technique can increase substantially if these parameters are incorporated into the feature vector.
2. Implementing trained models in real-time monitoring systems could provide continuous landslide surveillance. Deploying these models on operational platforms such as Geographic Information Systems (GIS) and connecting them to IoT devices in high-risk areas will give authorities with fast alerts and actionable insights, shortening emergency response times.
3. The satellite imagery is a strong tool, cross-validating model predictions using ground truth data and field studies may improve the model's robustness and reliability in real-world scenarios.

REFERENCE

- [1] A. K. Turner, “Social and environmental impacts of landslides,” Dec. 01, 2018, *Springer*. doi: 10.1007/s41062-018-0175-y.
- [2] W. Pollock and J. Wartman, “Human Vulnerability to Landslides,” *Geohealth*, vol. 4, no. 10, Oct. 2020, doi: 10.1029/2020GH000287.
- [3] Landslide atlas, Geological survey of India ,.Accessed: Dec 12, 2023 [Online]. Available: <https://www.gsi.gov.in>
- [4] “Preparation of Hazard, Vulnerability & Risk Analysis atlas and report for the state of Himachal Pradesh Building Vulnerability and Risk Assessment Composite Final Draft Report (T6) Prepared for Disaster Management Cell, Department of Revenue Government of Himachal Pradesh, Shimla.”2015
- [5] M. J. Froude and D. N. Petley, “Global fatal landslide occurrence from 2004 to 2016,” *Natural Hazards and Earth System Sciences*, vol. 18, no. 8, pp. 2161–2181, Aug. 2018, doi: 10.5194/nhess-18-2161-2018.
- [6] D. Cheng, Y. Cui, F. Su, Y. Jia, and C. E. Choi, “The characteristics of the Mocoa compound disaster event, Colombia,” *Landslides*, vol. 15, no. 6, pp. 1223–1232, Jun. 2018, doi: 10.1007/s10346-018-0969-1.
- [7] J. A. Marengo *et al.*, “Flash floods and landslides in the city of Recife, Northeast Brazil after heavy rain on May 25–28, 2022: Causes, impacts, and disaster preparedness,” *Weather Clim Extrem*, vol. 39, Mar. 2023, doi: 10.1016/j.wace.2022.100545.
- [8] D. K. Dwivedi, A. K. Saraf, and J. D. Das, “Geoinformatics-based investigation of slope failure and landslide damming of Chenab River, Lahaul-Spiti, Himachal Pradesh, India,” *Natural Hazards Research*, vol. 3, no. 2, pp. 186–195, Jun. 2023, doi: 10.1016/j.nhres.2023.02.008.
- [9] T. R. Martha, P. Roy, K. B. Govindharaj, K. V. Kumar, P. G. Diwakar, and V. K. Dadhwal, “Landslides triggered by the June 2013 extreme rainfall event in parts of Uttarakhand state, India,” Feb. 01, 2015, *Springer Verlag*. doi: 10.1007/s10346-014-0540-7.
- [10] D. D. Pandey, Banshtu, Singh Kanwarpreet, and Versain, “landslide hazard assessment using probabilistic and statistical approaches: a 1 case study of chambha region, himachal pradesh, india 2.”
- [11] N. Singh, S. K. Gupta, and D. P. Shukla, “Analysis of landslide reactivation using satellite data: a case study of kotrupi landslide, mandi, himachal pradesh, india,” in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, International Society for Photogrammetry and Remote Sensing, Feb. 2020, pp. 137–142. doi: 10.5194/isprs-archives-XLII-3-W11-137-2020.
- [12] V. Kumar, V. Gupta, I. Jamir, and S. L. Chatteraj, “Evaluation of potential landslide damming: Case study of Urni landslide, Kinnaur, Satluj valley, India,” *Geoscience Frontiers*, vol. 10, no. 2, pp. 753–767, Mar. 2019, doi: 10.1016/j.gsf.2018.05.004.

- [13] L. Highland, P. T. Bobrowsky, "The Landslide Handbook-A Guide to Understanding Landslides," US Geological Survey, 2008
- [14] R. N. Keyport, T. Oommen, T. R. Martha, K. S. Sajinkumar, and J. S. Gierke, "A comparative analysis of pixel- and object-based detection of landslides from very high-resolution images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 64, pp. 1–11, 2018, doi: 10.1016/j.jag.2017.08.015.
- [15] A. B. Gavade and V. S. Rajpurohit, "Systematic analysis of satellite image-based land cover classification techniques: literature review and challenges," *International Journal of Computers and Applications*, vol. 43, no. 6, pp. 514–523, 2021, doi: 10.1080/1206212X.2019.1573946.
- [16] A. Sharma and K. K. Sharma, "A Review on Satellite Image Processing for Landslides Detection," in *Artificial Intelligence and Machine Learning in Satellite Data Processing and Services: Proceedings of the International Conference on Small Satellites*, Singapore: Springer Nature, Jan. 2023, pp. 123–129.
- [17] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN computer science*, 2021, Springer. doi: 10.1007/s42979-021-00815-1.
- [18] U. K. Malviya and R. Gupta, "Satellite image classification method using ELBP and SVM classifier," in *Proceedings of the 2021 1st International Conference on Advances in Electrical, Computing, Communications and Sustainable Technologies, ICAECT 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICAECT49130.2021.9392509.
- [19] Y. G. Byun, Y. K. Han, and T. B. Chae, "A multispectral image segmentation approach for object-based image classification of high resolution satellite imagery," *KSCE Journal of Civil Engineering*, vol. 17, no. 2, pp. 486–497, Mar. 2013, doi: 10.1007/s12205-013-1800-0.
- [20] C. Sukawattanavijit, J. Chen, and H. Zhang, "GA-SVM Algorithm for Improving Land-Cover Classification Using SAR and Optical Remote Sensing Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 3, pp. 284–288, Mar. 2017, doi: 10.1109/LGRS.2016.2628406.
- [21] X. Huang and L. Zhang, "An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 257–272, 2013, doi: 10.1109/TGRS.2012.2202912.
- [22] D. P. Shukla, S. Gupta, C. S. Dubey, and M. Thakur, "Geo-spatial Technology for Landslide Hazard Zonation and Prediction," in *Environmental Applications of Remote Sensing*, InTech, 2016. doi: 10.5772/62667.
- [23] K. Sabanci, M. Fahri Ünlerşen, and K. Polat, "CLASSIFICATION OF DIFFERENT FOREST TYPES WITH MACHINE LEARNING ALGORITHMS", *Research for Rural Development*, vol 1, pp 254-260, 2016.

- [24] F. A. Mianji and Y. Zhang, "Robust hyperspectral classification using relevance vector machine," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6 PART 1, pp. 2100–2112, Jun. 2011, doi: 10.1109/TGRS.2010.2103381.
- [25] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Hyperspectral image segmentation using a new bayesian approach with active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10 PART 2, pp. 3947–3960, Oct. 2011, doi: 10.1109/TGRS.2011.2128330.
- [26] P. Ruiz, J. Mateos, G. Camps-Valls, R. Molina, and A. K. Katsaggelos, "Bayesian active remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 4, pp. 2186–2196, 2014, doi: 10.1109/TGRS.2013.2258468.
- [27] Z. Cui, Y. Wang, X. Gao, J. Li, and Y. Zheng, "Multispectral image classification based on improved weighted MRF Bayesian," *Neurocomputing*, vol. 212, pp. 75–87, Nov. 2016, doi: 10.1016/j.neucom.2016.03.097.
- [28] D. C. Duro, S. E. Franklin, and M. G. Dubé, "Multi-scale object-based image analysis and feature selection of multi-sensor earth observation imagery using random forests," *Int J Remote Sens*, vol. 33, no. 14, pp. 4502–4526, 2012, doi: 10.1080/01431161.2011.649864.
- [29] L. Albert, F. Rottensteiner, and C. Heipke, "A higher order conditional random field model for simultaneous classification of land cover and land use," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 130, pp. 63–80, Aug. 2017, doi: 10.1016/j.isprsjprs.2017.04.006.
- [30] J. A. Montoya-Zegarra, J. D. Wegner, L. Ladický, and K. Schindler, "Semantic segmentation of aerial images in urban areas with class-specific higher-order cliques," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Copernicus GmbH, Mar. 2015, pp. 127–133. doi: 10.5194/isprannals-II-3-W4-127-2015.
- [31] N. A. Mahmon and N. Ya'acob, "A review on classification of satellite image using Artificial Neural Network (ANN)," in *2014 IEEE 5th Control and System Graduate Research Colloquium*, IEEE, Aug. 2014, pp. 153–157. doi: 10.1109/ICSGRC.2014.6908713.
- [32] R. Sammouda, N. Adgaba, A. Touir, and A. Al-Ghamdi, "Agriculture satellite image segmentation using a modified artificial Hopfield neural network," *Comput Human Behav*, vol. 30, pp. 436–441, Jan. 2014, doi: 10.1016/j.chb.2013.06.025.
- [33] W. Zhao *et al.*, "Superpixel-based multiple local CNN for panchromatic and multispectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 4141–4156, Jul. 2017, doi: 10.1109/TGRS.2017.2689018.
- [34] T. Lei, D. Xue, Z. Lv, S. Li, Y. Zhang, and A. K. Nandi, "Unsupervised change detection using fast fuzzy clustering for landslide mapping from very high-resolution images," *Remote Sens (Basel)*, vol. 10, no. 9, Sep. 2018, doi: 10.3390/rs10091381.
- [35] D. G. Stavrakoudis and J. B. Theocharis, "An evolutionary fuzzy classifier for satellite image classification," in *2009 17th Mediterranean Conference on Control and Automation*, IEEE, Jun. 2009, pp. 383–388. doi: 10.1109/MED.2009.5164571.

- [36] M. D. Sinh, L. H. Trinh, and N. T. Long, “COMBINING FUZZY PROBABILITY AND FUZZY CLUSTERING FOR MULTISPECTRAL SATELLITE IMAGERY CLASSIFICATION,” *Vietnam J Sci Technol*, vol. 54, no. 3, p. 300, Jun. 2016, doi: 10.15625/0866-708x/54/3/6463.
- [37] L. T. Ngo and D. D. Nguyen, “Land cover classification using interval type-2 fuzzy clustering for multi-spectral satellite imagery,” in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, Oct. 2012, pp. 2371–2376. doi: 10.1109/ICSMC.2012.6378097.
- [38] S. Ji, D. Yu, C. Shen, W. Li, and Q. Xu, “Landslide detection from an open satellite imagery and digital elevation model dataset using attention boosted convolutional neural networks,” *Landslides*, vol. 17, no. 6, pp. 1337–1352, Jun. 2020, doi: 10.1007/s10346-020-01353-2.
- [39] S. Ullo *et al.*, “A New Mask R-CNN-Based Method for Improved Landslide Detection,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3799–3810, 2021, doi: 10.1109/JSTARS.2021.3064981.
- [40] Q. Zhang and T. Wang, “Deep Learning for Exploring Landslides with Remote Sensing and Geo-Environmental Data: Frameworks, Progress, Challenges, and Opportunities,” Apr. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/rs16081344.
- [41] F. Catani, “Landslide detection by deep learning of non-nadir and crowdsourced optical images,” *Landslides*, vol. 18, no. 3, pp. 1025–1044, Mar. 2021, doi: 10.1007/s10346-020-01513-4.
- [42] G. Cheng *et al.*, “Advances in Deep Learning Recognition of Landslides Based on Remote Sensing Images,” *Remote Sens (Basel)*, vol. 16, no. 10, May 2024, doi: 10.3390/rs16101787.
- [43] S. Yang *et al.*, “Automatic Identification of Landslides Based on Deep Learning,” *Applied Sciences (Switzerland)*, vol. 12, no. 16, Aug. 2022, doi: 10.3390/app12168153.
- [44] T. Liu, T. Chen, R. Niu, and A. Plaza, “Landslide Detection Mapping Employing CNN, ResNet, and DenseNet in the Three Gorges Reservoir, China,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 11417–11428, 2021, doi: 10.1109/JSTARS.2021.3117975.
- [45] R. Fu *et al.*, “Fast Seismic Landslide Detection Based on Improved Mask R-CNN,” *Remote Sens (Basel)*, vol. 14, no. 16, Aug. 2022, doi: 10.3390/rs14163928.
- [46] O. Ghorbanzadeh, K. Gholamnia, and P. Ghamisi, “The application of ResU-net and OBIA for landslide detection from multi-temporal sentinel-2 images,” *Big Earth Data*, 2022, doi: 10.1080/20964471.2022.2031544.
- [47] T. R. Martha, N. Kerle, V. Jetten, C. J. van Westen, and K. V. Kumar, “Characterising spectral, spatial and morphometric properties of landslides for semi-automatic detection using object-oriented methods,” *Geomorphology*, vol. 116, no. 1–2, pp. 24–36, Mar. 2010, doi: 10.1016/j.geomorph.2009.10.004.

- [48] T. Blaschke, B. Feizizadeh, and D. Hölbling, "Object-based image analysis and digital terrain analysis for locating landslides in the Urmia Lake basin, Iran," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 7, no. 12, pp. 4806–4817, Dec. 2014, doi: 10.1109/JSTARS.2014.2350036.
- [49] S. R. Meena *et al.*, "Landslide detection in the Himalayas using machine learning algorithms and U-Net," *Landslides*, vol. 19, no. 5, pp. 1209–1229, May 2022, doi: 10.1007/s10346-022-01861-3.
- [50] H. Wang, L. Zhang, K. Yin, H. Luo, and J. Li, "Landslide identification using machine learning," *Geoscience Frontiers*, vol. 12, no. 1, pp. 351–364, Jan. 2021, doi: 10.1016/j.gsf.2020.02.012.
- [51] S. Saha, P. Majumdar, and B. Bera, "Deep learning and benchmark machine learning based landslide susceptibility investigation, Garhwal Himalaya (India)," *Quaternary Science Advances*, vol. 10, Apr. 2023, doi: 10.1016/j.qsa.2023.100075.
- [52] O. Ghorbanzadeh, T. Blaschke, K. Gholamnia, S. R. Meena, D. Tiede, and J. Aryal, "Evaluation of different machine learning methods and deep-learning convolutional neural networks for landslide detection," *Remote Sens (Basel)*, vol. 11, no. 2, Jan. 2019, doi: 10.3390/rs11020196.
- [53] M. Naemitabar and M. Zanganeh Asadi, "Landslide zonation and assessment of Farizi watershed in northeastern Iran using data mining techniques," *Springer Science and Business Media B.V.* Sep. 01, 2021 doi: 10.1007/s11069-021-04805-7.
- [54] J. Dou *et al.*, "Automatic case-based reasoning approach for landslide detection: Integration of object-oriented image analysis and a genetic algorithm," *Remote Sensing (Basel)*, vol. 7, no. 4, pp. 4318–4342, 2015, doi: 10.3390/rs70404318.
- [55] T. R. Martha, P. Kamala, J. Jose, K. Vinod Kumar, and G. Jai Sankar, "Identification of new Landslides from High Resolution Satellite Data Covering a Large Area Using Object-Based Change Detection Methods," *Journal of the Indian Society of Remote Sensing*, vol. 44, no. 4, pp. 515–524, Aug. 2016, doi: 10.1007/s12524-015-0532-7.
- [56] F. Vecchiotti, N. Tilch, and A. Kociu, "The use of Terra-Aster satellite for landslide detection," *Geosciences (Switzerland)*, vol. 11, no. 6, Jun. 2021, doi: 10.3390/geosciences11060258.
- [57] D. Phiri, M. Simwanda, S. Salekin, V. R. Nyirenda, Y. Murayama, and M. Ranagalage, "Sentinel-2 data for land cover/use mapping: A review," *remote sensing*, Jul. 01, 2020, *MDPI AG*. doi: 10.3390/rs12142291.
- [58] U. Shrawankar and C. Shrawankar, "An Algorithm for High-Resolution Satellite Imagery Pre-processing," 2022, doi: 10.20944/preprints202203.0095.v1.
- [59] S. D., P. Deepa, and V. K., "Remote Sensing Satellite Image Processing Techniques for Image Classification: A Comprehensive Survey," *International Journal Computer Application*, vol. 161, no. 11, pp. 24–37, Mar. 2017, doi: 10.5120/ijca2017913306.
- [60] Y. Yi and W. Zhang, "A New Deep-Learning-Based Approach for Earthquake-Triggered Landslide Detection from Singleoral RapidEye Satellite Imagery," *IEEE J*

Sel Top Appl Earth Obs Remote Sensing, vol. 13, pp. 6166–6176, 2020, doi: 10.1109/JSTARS.2020.3028855.

- [61] A. Sharma *et al.*, “Artificial Intelligence Techniques for Landslides Prediction using Satellite Imagery,” *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3446037.
- [62] J. Storey, D. P. Roy, J. Masek, F. Gascon, J. Dwyer, and M. Choate, “A note on the temporary misregistration of Landsat-8 Operational Land Imager (OLI) and Sentinel-2 Multi Spectral Instrument (MSI) imagery,” *Remote Sensing Environment*, vol. 186, pp. 121–122, Dec. 2016, doi: 10.1016/j.rse.2016.08.025.
- [63] J. Storey, M. Choate, and K. Lee, “Landsat 8 operational land imager on-orbit geometric calibration and performance,” *Remote Sensing (Basel)*, vol. 6, no. 11, pp. 11127–11152, 2014, doi: 10.3390/rs6111127.
- [64] J. A. Prieto-Amparan, F. Villarreal-Guerrero, M. Martinez-Salvador, C. Manjarrez-Domínguez, E. Santellano-Estrada, and A. Pinedo-Alvarez, “Atmospheric and radiometric correction algorithms for the multitemporal assessment of grasslands productivity,” *Remote Sensing (Basel)*, vol. 10, no. 2, Feb. 2018, doi: 10.3390/rs10020219.
- [65] S. D. Pawar, " Kamal, and K. Sharma, “Machine Learning Approach towards Mammographic Breast Density Measurement for Breast Cancer Risk Prediction: An overview.” [Online]. Available: <https://ssrn.com/abstract=3599187>
- [66] B. D. Bue *et al.*, “Real-Time Atmospheric Correction of AVIRIS-NG Imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 12, pp. 6419–6428, Dec. 2015, doi: 10.1109/TGRS.2015.2439215.
- [67] S. D. Pawar *et al.*, “Multichannel DenseNet Architecture for Classification of Mammographic Breast Density for Breast Cancer Detection,” *Front Public Health*, vol. 10, Apr. 2022, doi: 10.3389/fpubh.2022.885212.
- [68] S. Suyarso, M. D. Setiawati, I. H. Supriyadi, and B. Prayudha, “Climate change indicator, impact, adaptation, and innovation at the local level: learn from the peoples’ experience of the coastal plain of Probolinggo, East Java, Indonesia,” in *Climate Change, Community Response and Resilience*, Elsevier, 2023, pp. 93–118. doi: 10.1016/B978-0-443-18707-0.00005-9.
- [69] B. L. Markham *et al.*, “Landsat Program,” in *Comprehensive Remote Sensing*, Elsevier, 2018, pp. 27–90. doi: 10.1016/B978-0-12-409548-9.10313-6.
- [70] A. Asokan, J. Anitha, M. Ciobanu, A. Gabor, A. Naaji, and D. J. Hemanth, “Image processing techniques for analysis of satellite images for historical maps classification- An overview,” *Applied Sciences*, Jun. 01, 2020, MDPI AG. doi: 10.3390/app10124207.
- [71] S. Suresh and S. Lal, “Modified differential evolution algorithm for contrast and brightness enhancement of satellite images,” *Applied Soft Computing Journal*, vol. 61, pp. 622–641, Dec. 2017, doi: 10.1016/j.asoc.2017.08.019.
- [72] B. Yu and F. Chen, “A new technique for landslide mapping from a large-scale remote sensed image: A case study of Central Nepal,” *Computer Geoscience*, vol. 100, pp. 115–124, Mar. 2017, doi: 10.1016/j.cageo.2016.12.007.

- [73] X. Xie, Y. Zhang, H. Wang, and W. Zhang, "Analysis and modeling of radiometric error caused by imaging blur in optical remote sensing systems," *Infrared Phys Technol*, vol. 77, pp. 51–57, Jul. 2016, doi: 10.1016/j.infrared.2016.05.010.
- [74] A. M. Deshpande, S. R. Patale, and S. Roy, "Removal of line striping and shot noise from remote sensing imagery using a deep neural network with post-processing for improved restoration quality," *International Journal Remote Sensing*, vol. 42, no. 19, pp. 7357–7380, Oct. 2021, doi: 10.1080/01431161.2021.1957512.
- [75] N. Panigrahi, B. K. Mohan, and G. Athithan, "Pre-processing algorithm for rectification of geometric distortions in Satellite images," *Defence Science Journal*, vol. 61, no. 2, pp. 174–179, 2011, doi: 10.14429/dsj.61.421.
- [76] J. Lever, M. Krzywinski, and N. Altman, "Points of Significance: Principal component analysis," Jun. 29, 2017, *Nature Publishing Group*. doi: 10.1038/nmeth.4346.
- [77] M. Jambu, "Principal Components Analysis," in *Exploratory and Multivariate Data Analysis*, Elsevier, 1991, pp. 125–167. doi: 10.1016/B978-0-08-092367-3.50011-3.
- [78] Y. Sun, S. Zhou, S. Meng, M. Wang, and H. Mu, "Principal component analysis–artificial neural network-based model for predicting the static strength of seasonally frozen soils," *Scientific Report*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-43462-7.
- [79] K. Ezukwoke and S. Zareian, "KERNEL METHODS FOR PRINCIPAL COMPONENT ANALYSIS (PCA) A comparative study of classical and kernel PCA", doi: 10.13140/RG.2.2.17763.09760.
- [80] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," Aug. 01, 2018, *Springer Verlag*. doi: 10.1007/s13244-018-0639-9.
- [81] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," March 01, 2023, *MDPI*. doi: 10.3390/computation11030052.
- [82] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.
- [83] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Computer Science*, vol. 2, no. 6, p. 420, Nov. 2021, doi: 10.1007/s42979-021-00815-1.
- [84] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, [Online]. Available: <http://arxiv.org/abs/1512.03385>
- [85] Y. Hu, A. Huber, J. Anumula, and S.-C. Liu, "Overcoming the vanishing gradient problem in plain recurrent networks," Jan. 2018, [Online]. Available: <http://arxiv.org/abs/1801.06105>
- [86] S. A. Hasanah, A. A. Pravitasari, A. S. Abdullah, I. N. Yulita, and M. H. Asnawi, "A Deep Learning Review of ResNet Architecture for Lung Disease Identification in CXR

- Image,” *Applied Sciences*, vol. 13, no. 24, p. 13111, Dec. 2023, doi: 10.3390/app132413111.
- [87] Q. Zhang, “A novel ResNet101 model based on dense dilated convolution for image classification,” *SN Applied Science*, vol. 4, no. 1, Jan. 2022, doi: 10.1007/s42452-021-04897-7.
 - [88] C. Szegedy *et al.*, “Going Deeper with Convolutions.” *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, doi: [10.1109/CVPR.2015.7298594](https://doi.org/10.1109/CVPR.2015.7298594)
 - [89] S. Sa'idah, A. Fany, and I. P. Y. N. Suparta, “Convolutional Neural Network GoogleNet Architecture for Detecting the Defect Tire,” in *2022 International Conference on Computer Science and Software Engineering (CSASE)*, IEEE, Mar. 2022, pp. 331–336. doi: 10.1109/CSASE51777.2022.9759694.
 - [90] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional Block Attention Module,” Jul. 2018, [Online]. Available: <http://arxiv.org/abs/1807.06521>
 - [91] L. Chen *et al.*, “SCA-CNN: Spatial and Channel-wise Attention in Convolutional Networks for Image Captioning,” Nov. 2016, [Online]. Available: <http://arxiv.org/abs/1611.05594>
 - [92] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-Excitation Networks,” Sep. 2017, [Online]. Available: <http://arxiv.org/abs/1709.01507>
 - [93] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” Nov. 2013, [Online]. Available: <http://arxiv.org/abs/1311.2901>
 - [94] H. Zhao, C. Jiaya, J. Cuhk, and V. Koltun, “Exploring Self-attention for Image Recognition.” *Computer vision and pattern recognition*, vol. 1, April 2020, doi <https://doi.org/10.48550/arXiv.2004.13621>
 - [95] Z. Shen, I. Bello, R. Vemulapalli, X. Jia, and C.-H. Chen, “Global Self-Attention Networks for Image Recognition,” *Computer Vision and Pattern Recognition* Oct. 2020, [Online]. Available: <http://arxiv.org/abs/2010.03019>
 - [96] Z. Yang, C. Xu, and L. Li, “Landslide Detection Based on ResU-Net with Transformer and CBAM Embedded: Two Examples with Geologically Different Environments,” *Remote Sens (Basel)*, vol. 14, no. 12, Jun. 2022, doi: 10.3390/rs14122885.

LIST OF PUBLICATION

1. Journal Publication

- [1] Akanksha Sharma, Shakti Raj Chopra, Suhas Gajanan Sapate, Krishan Arora, Mohammad Khalid, Sudan Jha, “Artificial Intelligence Techniques for Landslides Prediction Using Satellite Imagery”, IEEE Access12,2024, pp117318 – 117334. [10.1109/ACCESS.2024.3446037](https://doi.org/10.1109/ACCESS.2024.3446037)

2. Conference Publication

- [1] Akanksha Sharma, Kamal. Sharma, and S. G. Sapate, “A prototype model for detection and classification of landslides using satellite data,” in *Journal of Physics: Conference Series*, Institute of Physics, 2022. doi: 10.1088/1742-6596/2327/1/012029
- [2] Akanksha Sharma, Shakti Raj Chopra, Suhas G. Sapate, and P. B. Bhagawati, “Recent Trends and advances in deep learning techniques for the classification of landslides using satellite images: comprehensive survey,” in *IOP Conference Series: Earth and Environmental Science*, Institute of Physics, 2024. doi: 10.1088/1755-1315/1285/1/012024
- [3] Akanksha Sharma, Shakti Raj Chopra, Suhas G. Sapate, and P. B. Bhagawati, “Algorithm For Preprocessing Satellite Imagery That Uses Geometric, Atmospheric, And Radiometric Correction” *International Conference on Emerging Smart Computing & Informatics 2025*. Doi: 10.1109/ESCI63694.2025.10988359

3. Book chapter

- [1] Akanksha Sharma and Kamal Kishor Sharma, “A Review on Satellite Image Processing for Landslides Detection,” in *Artificial Intelligence and Machine Learning in Satellite Data Processing and Services: Proceedings of the International Conference on Small Satellites*, Singapore: Springer Nature, Jan. 2023, pp. 123–129.