

OPCNet: An Optimized Parallel Convolutional Neural Network for Classification of Satellite Imagery



By

Priyanti Paul Tumpa, MSc Eng

20METE008P

A thesis submitted in partial fulfillment of the requirements for the degree of
MASTER of SCIENCE in ELECTRONICS AND TELECOMMUNICATION
ENGINEERING

Department of Electronics and Telecommunication Engineering

CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

FEBRUARY 2024

APPROVAL

The thesis titled “OPCNet: An Optimized Parallel Convolutional Neural Network for Classification of Satellite Imagery” submitted by Priyanti Paul Tumpa, ID 20METE008P, Session: 2020-21 has been accepted as satisfactory in partial fulfillment of the requirement of the degree of Master of Science (M.Sc.) in Electronics and Telecommunication Engineering on 22th February 2024.

Board of Examiners

- | | | |
|----|---|--------------------------|
| 1. | Dr. Md. Saiful Islam
Associate Professor, Dept. of ETE, CUET | Chairman
(Supervisor) |
| 2. | Prof. Dr. Md. Azad Hossain
Head, Dept. of ETE, CUET | Member
(Ex-Officio) |
| 3. | Dr. Md. Azad Hossain
Professor, Dept. of ETE, CUET | Member |
| 4. | Dr. Md. Jahedul Islam
Professor, Dept. of ETE, CUET | Member |
| 5. | Dr. Mohammad Rubaiyat Tanvir Hossain
Professor, Dept. of EEE, CUET | Member
(External) |

Declaration

I hereby declare that the work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the Thesis contains no material previously published or written by another person except where due reference is cited. Furthermore, the Thesis complies with PLAGIARISM and ACADEMIC INTEGRITY regulation of CUET.

Priyanti Paul Tumpa

20METE008P

Department of Electronics and Telecommunication Engineering
Chittagong University of Engineering & Technology (CUET)

Copyright © Priyanti Paul Tumpa, 2024.

This work may not be copied without permission of the author or Chittagong University of Engineering & Technology.

Dedication

Dedicated to

My beloved parents, colleagues and **Dr. Md. Saiful Islam** sir

For their continuous support

List of Publications

1. **P.P. Tumpa, M. S. Islam**, “LPCNN: Lightweight Parallel Convolutional Neural Network with SVM classifier for Satellite Imagery Classification” *IEEE Transactions on Artificial Intelligence* (under review), [Q1 journal, Impact factor: 7.25, Citescore: 4.9].
2. **P.P. Tumpa, M. S. Islam**, J. Uddin, M. A. Samad, and K. Choi, “Machinery Bearing Fault Identification through Parallel Convolution Neural Network and Support Vector Machine Analysis”, *IEEE Access* (under review), [Q1 journal, Impact factor: 3.9, Citescore: 9].
3. **M. S. Islam**, M. A. Mahfuz, **P. P. Tumpa**, J.Uddin, S. G. Villar, E. G. Villena, I. Ashraf, M. A. Samad, “A Neuro-Fuzzy Based Reliability System for Generating Failure Probability to Evaluate Nuclear Basic Events of Power Plant” *Arabian Journal for Science and Engineering (AJSE)*, springer, (under review), [Q1 journal, Impact factor: 2.8, Citescore: 4.4].
4. **P.P. Tumpa, M. S. Islam**, “Forecasting of Nuclear Power Plant Burst Factors using Radial Basis Neural Network throughout Loss of Coolant Accident”, *3rd International Conference on Advancement in Electrical and Electronic Engineering*, (Submitted, 30 January, 2024), [Scopus Index].
5. **P.P. Tumpa, M. S. Islam**, Z. May, M. K. Alam, “Nuclear Power Plant Burst Parameters Prediction During a Loss-of-Coolant Accident Using an Artificial Neural Network”, *Proceedings of the International Conference on Big Data, IoT, and Machine Learning, Lecture Notes on Data Engineering and Communications Technologies*, vol 95. Springer, Singapore, 2022. https://doi.org/10.1007/978-981-16-6636-0_31 [Scops Index].

Approval by the Supervisor

This is to certify that **Priyanti Paul Tumpa** has carried out this research work under my supervision, and that she has fulfilled the relevant Academic Ordinance of the Chittagong University of Engineering and Technology, so that she is qualified to submit the following Thesis in the application for the degree of MASTER of SCIENCE in Electronics and Telecommunication Engineering (ETE). Furthermore, the Thesis complies with the PLAGIARISM and ACADEMIC INTEGRITY regulation of CUET.

Dr. Md. Saiful Islam

Associate Professor

Department of Electronics and Telecommunication Engineering

Chittagong University of Engineering & Technology

Acknowledgement

First and foremost, I'd like to thank my GOD for making things easier than I could ever have imagined. I would like to keep in mind a few very essential people without whom I would be unable to accomplish this thesis. I would like to express my gratitude to my distinguished supervisor, Dr. Md. Saiful Islam, Associate Professor, Department of ETE, CUET, for his contributions, inspiration, and motivation. Throughout my journey, he has provided regular instruction, assistance, and motivation in overcoming various technical challenges. I'd like to thank Prof. Dr. Azad Hossain, Head, ETE, CUET, and my colleagues for their ongoing encouragement and drive to accomplish my work. I'm grateful to my family for inspiring me to pursue higher education in the first place. Their unwavering support and encouragement were the first things that prepared the path for this journey. Last but not least, I would like to acknowledge everyone at the Department of Electronics and Telecommunication Engineering (ETE), CUET, for their technical and administrative assistance with my thesis work.

.

Abstract

Satellite image classification is crucial for various applications, driving advancements in Convolution Neural Networks (CNNs). While CNNs have proven effective, deep models often encounter overfitting issues as the network's depth increases since the model has to learn many parameters. Besides this, traditional CNNs have the inherent difficulty in extracting fine-grained details and broader patterns simultaneously. To overcome these challenges, this research presents a novel approach using an optimized parallel CNN (OPCNet) architecture with an SVM classifier to classify satellite images. Each branch within the parallel network is designed for specific resolution characteristics, spanning from low (emphasizing broader patterns) to high (capturing fine-grained details), enabling the simultaneous extraction of a comprehensive set of features without increasing network depth. The OPCNet incorporates a dilation factor to expand the network's receptive field without increasing parameters, and a dropout layer is introduced to mitigate overfitting. Evaluation of two public datasets (EuroSAT dataset and RSI-CB256 dataset) demonstrates remarkable accuracy rates of 97.91% and 99.8%, surpassing previous state-of-the-art models. Finally, OPCNet, with less than 1 million parameters, outperforms high-parameter models by effectively addressing overfitting issues, showcasing exceptional performance in satellite image classification.

বিমূর্ত

স্যাটেলাইট ইমেজ শ্রেণীবিভাগ বিভিন্ন অ্যাপ্লিকেশনের জন্য গুরুত্বপূর্ণ, কনভোলিউশন নিউরাল নেটওয়ার্কে (সিএনএন) এক্ষেত্রে অগ্রগতি দেখাচ্ছে। যদিও সিএনএন গুলি কার্যকর প্রমাণিত হয়েছে, কিন্তু গভীর মডেলগুলি প্রায়শই ওভারফিটিং সমস্যার সম্মুখীন হয় কারণ নেটওয়ার্কের গভীরতা বৃদ্ধি পায় ফলে মডেলটিকে অনেকগুলি প্যারামিটার শিখতে হয়। এটি ছাড়াও, ঐতিহ্যবাহী সিএনএন-এর একই সাথে সূক্ষ্ম বিবরণ এবং বৃহত্তর নিদর্শনগুলি বের করতে সহজাত অসুবিধা রয়েছে। এই চ্যালেঞ্জগুলি কাটিয়ে উঠতে, এই গবেষণাটি উপগ্রহ চিত্রগুলিকে শ্রেণিবদ্ধ করার জন্য একটি এসভিএম শ্রেণীবিভাগের সাথে একটি লাইটওয়েট সমান্তরাল সিএনএন(এলপিসিএনএন) আর্কিটেকচার ব্যবহার করে একটি অভিনব পদ্ধতি উপস্থাপন করে। সমান্তরাল নেটওয়ার্কের মধ্যে প্রতিটি শাখা নির্দিষ্ট রেজোলিউশন বৈশিষ্ট্যের জন্য ডিজাইন করা হয়েছে, নিম্ন (বিস্তৃত প্যাটার্নের উপর জোর দেওয়া) থেকে উচ্চ (সূক্ষ্ম-দানাযুক্ত বিবরণ ক্যাপচার করা) পর্যন্ত বিস্তৃত, বৈশিষ্ট্যগুলি নেটওয়ার্কের গভীরতা না বাড়িয়ে একযোগে নিষ্কাশন সক্ষম করে। এলপিসিএনএন প্যারামিটার না বাড়িয়ে নেটওয়ার্কের গ্রহণযোগ্য ক্ষেত্রকে প্রসারিত করার জন্য একটি প্রসারণ ফ্যাক্টরকে অন্তর্ভুক্ত করে এবং অতিরিক্ত ফিটিং প্রশমিত করার জন্য একটি ড্রপআউট স্তর প্রবর্তন করা হয়। দুটি পাবলিক ডেটাসেটের মূল্যায়ন (ইউরোস্যাট ডেটাসেট এবং আরএসআই-সিবি ২৫৬ ডেটাসেট) ৯৭.৯১% এবং ৯৯.৮% এর উল্লেখযোগ্য নির্ভুলতার হার প্রদর্শন করে, যা পূর্ববর্তী অত্যাধুনিক মডেলগুলিকে ছাড়িয়ে যায়। অবশেষে, এলপিসিএনএন, ১ মিলিয়নেরও কম প্যারামিটার সহ, ওভারফিটিং সমস্যাগুলিকে কার্যকরভাবে সমাধান করে, স্যাটেলাইট ইমেজ শ্রেণীবিভাগে ব্যতিক্রমী কর্মক্ষমতা প্রদর্শন করে উচ্চ-প্যারামিটার মডেলগুলিকে ছাড়িয়ে যায়।

Table of Contents

Declaration	ii
List of Publications.....	iv
Approval by Supervisor.....	v
Acknowledgement.....	vi
Abstract.....	vii
Table of Contents	ix
List of Figures	xi
List of Tables	xii
Nomenclature	xiii
 Chapter 1: INTRODUCTION.....	1
1.1 Background.....	1
1.2 Problem Statement.....	3
1.3 Aims and Objectives.....	4
1.4 Significance, Scope and Definitions	4
1.5 Thesis Outline.....	5
 Chapter 2: LITERATURE REVIEW	7
2.1 Satellite Images.....	7
2.2 Neural Networks	8
2.2.1 Artificial Neural Network	8
2.2.2 Convolutional Neural Network.....	10
2.3 Dilated Convolutional Neural Ntework.....	16
2.4 Pretrained Models.....	17
2.4.1 VGG 16	17
2.4.2 AlexNet	18
2.4.3 ResNet-50.....	19
2.5 Classifiers	21
2.5.1 SoftMax Classifier.....	21
2.5.2 SVM Classifier	22
2.5.3 KNN Classifier	24

2.5.4 RF Classifier.....	24
2.4.2 Naïve Bayes Classifier.....	25
2.6 Related Works On Satellite Image Classification	25
Chapter 3: MATERIALS AND METHODOLOGY	28
3.1 Overview of Methodology	28
3.2 Dataset Description	30
3.2.1 RSI-CB256 Dataset	31
3.2.2 EuroSAT Dataset.....	31
3.3 Preprocessing	33
3.3.1 Image Resizing	33
3.3.2 Image Sharpening.....	33
3.4 Proposed OPCNet Architecture.....	35
3.4.1 Branch 1 (3x3 Kernel Size): Extracting Detail Feature	33
3.4.2 Branch 2 (5x5 Kernel Size): Capturing Intermediate Structures	37
3.4.3 Branch 3 (7x7 Kernel Size): Grasping Broad Patterns.....	39
3.5 SVM Classifier	43
Chapter 4: RESULT AND ANALYSIS	45
4.1 Evaluation Measurement Of Proposed System.....	45
4.2 Validation Of Lightweight Design	48
4.3 Performance Analysis And Comparison Of Different Classifiers	48
4.4 Reduction Of Overfitting Problem	48
4.5 Parameter Analysis Of The Proposed System	51
4.6 Comparison With Related Works.....	57
Chapter 5: CONCLUSIONS.....	59
5.1 Conclusion	59
5.2 Limitation And Challenges Of The Study	61
5.3 Recommendation For Further Study	61
References	63

List of Figures

Fig. No.	Figure Caption	Page No.
Fig. 2.1	Example of Satellite image.....	8
Fig. 2.2	Basic Structure of ANN.....	9
Fig. 2.3	Basic Structure of CNN.....	10
Fig. 2.4	Visualization of the convolutional operation	11
Fig. 2.5	Visualization of the pooling operation.....	14
Fig. 2.6	(a) Normal Convolutional kernel (b) Dilated Convolutional kernel.....	16
Fig. 2.7	Architecture of the VGG16.	18
Fig. 2.8	Architecture of the AlexNet	19
Fig. 2.9	Architecture of the ResNet-50.....	20
Fig. 2.10	Residual building block of the ResNet50.....	20
Fig. 2.11	Visualization of the SVM classification processes	24
Fig. 3.1	A proposed framework for satellite imagery classification	29
Fig. 3.2	Samples from the Dataset-RSI-CB256 of satellite images.....	31
Fig. 3.3	Samples from the EuroSAT Dataset of satellite images.....	32
Fig. 3.4	Doughnut chart for category distribution of (a)RSI-CB256 (b) EuroSAT Datasets	32
Fig. 3.5	Pre-processed samples of Satellite images	34
Fig. 3.6	The proposed lightweight parallel convolutional neural network.	36
Fig. 3.7	(a) High resolution (finer detailed) features with kernel size 3x3, (b) Mid resolution features with kernel size 5x5, (c) Low resolution (broader pattern) features with kernel size 7x7, and (d) Final feature map after concatenation layer.....	41
Fig. 4.1	Training and testing accuracies for (a) RSI-CB256 dataset, and (b) EuroSAT dataset	50
Fig. 4.2	Training and test accuracy on different training-test data split for (a) RSICB256 dataset (b) EuroSAT dataset.....	53
Fig. 4.3	Training and test accuracy on various learning rate, α for (a) RSICB256 dataset (b) EuroSAT dataset	54
Fig. 4.4	Training and test accuracy on various batch size for (a) RSI-CB256 dataset (b) EuroSAT dataset.	55
Fig. 4.5	Training and test accuracy on various dropout rate for (a) RSI-CB256 dataset (b) EuroSAT dataset.	56

List of Tables

Table No.	Table Caption	Page No.
Table 2.1.	Activation functions in a CNN.....	13
Table 3.1.	Details of the proposed LPCNN model.....	41
Table 3.2.	SVM classifier parameter.	44
Table 4.1.	Comparison result for different image dimensions.	46
Table 4.2.	Simplicity Comparison of the proposed PCNN model to the state of art models.....	47
Table 4.3.	Processing time of the proposed PCNN model to the state of art models.	47
Table 4.4.	Comparative analysis of performance matrices for different classifiersComparison result for different image dimensions.	48
Table 4.5.	Classification metrics on RSI-CB256 Dataset for different training-test data split.	51
Table 4.6.	Classification metrics on EuroSAT Dataset for different training-test data split.	52
Table 4.7.	Hyperparameters of the proposed LPCNN model.....	57
Table 4.8.	Comparison with related works in Satellite Imagery Classification.	58

Nomenclature

ANN	Artificial Neural Network
CNN	Convolution Neural Network
SVM	Support Vector Machine
KNN	K-Nearest Neighbour
NB	Naïve Bayes
RF	Random Forest
FC	Fully Connected
SGD	Stochastic Gradient Descent
BN	Batch Normalization
LBP	Local Binary Patterns

Chapter 1: INTRODUCTION

This chapter explains the overview of recent state of deep learning models in satellite image classification. Also, this chapter outlines the background in section 1.1, problem statement in section 1.2 of the research and its aim and objectives in section 1.3. Next, section 1.4 describes the significance and scope of this research. Finally, section 1.5 includes an outline of the remaining chapters of the Thesis.

1.1 BACKGROUND

Images of Earth gathered by satellites, the unmanned aerial systems are referred to as satellite images [1]. The utility of satellite images extends to various fields, including land planning [2-4], surveillance [5], monitoring, agriculture [6-7], marine studies [8-10] and others. In satellite imagery, experts in computer vision strategies play a pivotal role in tasks such as image classification, interpretation, and object recognition. Satellite image classification, a core element in this domain, involves categorizing diverse land cover types, including forests, croplands, urban areas, and water bodies [11]. This precise classification is indispensable for a wide array of applications, such as natural resource management, land use planning, forestry, agricultural oversight, urban development planning, regional planning, and strategic disaster management. The conventional approaches to satellite image classification rely on traditional methods that necessitate manual feature engineering and the application of standard machine learning classifiers. However, the manual analysis of this data is a costly and labour-intensive process [12], prompting the integration of artificial intelligence to automate and refine the classification of satellite imagery.

In recent years, the fusion of satellite imagery and advanced machine learning techniques, particularly Convolutional Neural Networks (CNNs), has opened avenues for unprecedented insights and discoveries. CNNs, initially

designed for image recognition, exhibit promising capabilities in discerning complex patterns within satellite images. CNNs excel in satellite image classification by automatically extracting intricate features. Their efficiency, adaptability to varying resolutions, and state-of-the-art performance make them superior to traditional machine learning models in handling the complexities of satellite data. Several experiments have demonstrated CNN's capacity to execute land cover classification [13-15] and road extraction from remote sensing imagery [16]. In [9], CNN categorizes the utilized and unutilized terrain. Pretrained CNN models (Resnet50, GoogleNet, VGG19, and AlexNet) effectively classify satellite images [17].

Conventional CNNs often incorporate successive convolutional layers, stacking them to enhance network depth and facilitate learning numerous parameters during training. However, this augmentation in depth leads to overfitting, diminishing the network's classification performance—a significant drawback associated with CNN models. The net dropout approach [18] was employed to minimize overfitting concerns, although it reduces classification accuracy by removing significant information. Pre-trained algorithms with transfer learning were also devised in this regard [19-21]. Another limitation of the deep CNN model is its inability to extract features across multiple resolutions simultaneously. This paper presents difficulties in satellite image classification, where there is a need to capture low-resolution (broader patterns) and high-resolution (fine-grained details) features to enhance the robustness of feature maps and overall performance. The constraint becomes evident as a single deep CNN model struggles to address this diverse requirement effectively.

The motivation behind this research stems from the need to address the challenges associated with overfitting in satellite image classification while avoiding the complexity of deepening neural networks. In satellite images, fine-grained features like intricate architecture, textures, and small-scale elements are

crucial, while broader pattern features contribute to understanding diverse land cover, geographical structures, and overall contextual elements of a region. To extract various resolution characteristics from satellite images, a CNN model with parallel branches and unique parameters needs to be developed. This architecture will allow for the simultaneous extraction of several characteristics, addressing the need for accurate representation of features in the classification process. Furthermore, the reasons of the research intend to improve network efficiency by keeping it lightweight, and to determine the best classifier for satellite image classification. The inspiration for this work aligns with the goal of achieving effective classification performance while addressing overfitting concerns and maintaining a simplified framework.

1.2 PROBLEM STATEMENT

When standard CNN models are utilized to classify satellite images, they encounter by the following limitations, which could impact their effectiveness in extracting relevant features and yielding precise outcomes.

- Conventional CNN models have a sequential layering design, implying layers are stacked on each other. The model's depth and complexity are increased by this stacking technique, which also raises in the number of trainable parameters. This intricacy could lead to overfitting, causing the model to perform severely on new satellite images while obtaining excellent training accuracy.
- Conventional CNNs might encounter difficulty capturing features at several resolutions simultaneously. As a result, important information at various resolutions goes unnoticed, compromising the model's capacity to distinguish tiny features and broader patterns in satellite data.
- Deep CNNs with large architectures need substantial computing resources, processing power, and computational time.

1.3 AIMS AND OBJECTIVES

The main objective of this research is to develop an optimized parallel CNN model that can precisely classify the satellite images while reducing the limitations of traditional CNN models.

The objectives of this research are specified as below:

- i. To develop a OPCNet framework for classifying satellite images.
- ii. To overcome the overfitting problem of conventional neural networks without increasing the network's depth using feature fusion technology of parallel neural networks.
- iii. To select an efficient classifier to classify the satellite images using obtained features.
- iv. To perform a comparative analysis on the proposed system to study the improvements in efficiency and performance on multiple datasets.

1.4 SCOPE OF THE STUDY

Traditional CNNs, which use consecutive convolutional layers, are susceptible of overfitting, which impairs the model's classification accuracy. Another drawback is that it cannot extract features from multiple resolutions at the same time. This difficulty is especially acute in satellite image classification, where capturing both low-resolution (broader patterns) and high-resolution (fine-grained details) features is critical for creating strong feature maps and assuring overall performance. The scope of this research extends to satellite image classification, offering a fresh solution to the issues encountered in deep CNN models. The parallel building of the CNN models enables the extraction of information at various resolutions, addressing the specific requirements of satellite data, where both broader patterns and fine-grained features are required for precise categorization.

The scope of this study extends to an assessment of multiple classifiers with the goal of determining the most effective one for improving the overall performance of satellite image categorization. This research aims to offer novel insights applicable to a variety of computer vision applications and neural network designs, in addition to tackling overfitting concerns. The potential impact extends beyond overcoming specific challenges in satellite image classification, paving the way for greater improvements and developments in related domains.

1.5 THESIS OUTLINE

The topics in this thesis is divided into five chapters. The following is a brief overview of the chapters:

- Chapter 1, the introduction chapter, contains research motivation, problem statement, objective and thesis organization. Motivation for the research, objective for the study and how this thesis paper is organized are concisely described in the chapter.
- Theoretical background in line with literature review associated with the study is organized in chapter 2. It will provide a brief overview of the theories, formulae and literature required for the audience to grow interest in and understand the study.
- Chapter 3 systematically organizes the methodological framework followed in the study. A comprehensive flowchart for the methodological framework is illustrated for providing a concise understanding on how the study is performed. Details on designing and modelling of various algorithms along with design parameters and figures are included in the chapter.
- Chapter 4 includes simulation results, performance analysis, and comparative study. Data and output figures are included in this chapter.

- Finally, chapter 5 summarizes overall study method, results and findings. Challenges and limitations are also discussed. Future recommendations and conclusion finish the thesis paper.

Chapter 2: LITERATURE REVIEW

This chapter provides an extensive theoretical foundation, supported by a significant amount of literature, to help the reader gain a comprehensive grasp. In Section 2.1, the details of satellite images are discussed. Section 2.2 then looks into the operational concepts of neural networks, which include both artificial and convolutional networks. Section 2.3 describes about the Dilated Convolutional Neural Network. The discussion extends to pre-trained models such as VGG16, AlexNet, and ResNet50 in Section 2.4. Finally, Section 2.5 describes the operating principles of classifiers, including the SoftMax, Support Vector Machine (SVM), K-Nearest Neighbour (K-NN), Random Forest (RF), and Naïve Bayes (NB). Section 2.7 discusses significant works in the field of satellite image analysis.

2.1 SATELLITE IMAGES

A satellite image is a photograph or visual depiction of the Earth's surface taken by satellites orbiting around the earth. These photographs are captured utilizing specialized sensors onboard these satellites, which allow them to record information across multiple wavelengths of the electromagnetic spectrum. Satellite photos offer a bird's-eye perspective on landscapes, towns, oceans, and other geographical phenomena. The capacity to capture large geographic areas with high spatial resolution enables landscape monitoring, natural disaster tracking, and environmental impact assessments. Satellite imagery has become an essential tool in the agricultural sector, urban planning, disaster management, and environmental monitoring. Notable advances in satellite technology, sensor capabilities, and data processing have considerably enhanced the accuracy and availability of these images, allowing researchers and officials to derive useful information for sustainable development. [22] and [23] highlight the importance of satellite imagery in environmental study and its numerous applications in

current times. Figure 2.1 shows a typical satellite view of a city. This image was obtained from Sporcle, an online site with high-resolution satellite images [24].



Figure 2.1: Example of Satellite image [24].

2.2 NEURAL NETWORKS

A neural network is a technique used in artificial intelligence that trains computers to process data in a manner modelled after the human brain. Deep learning is a form of machine learning technique that employs interconnected nodes or neurons in a structure with layers similar to the human brain [25]. Neural networks can assist computers in making intelligent decisions with minimal human intervention. There are various types of neural networks used in deep learning, like artificial neural networks (ANN), convolutional neural networks (CNN), and so on. Some important neural networks are listed below.

2.2.1 Artificial neural networks (ANN)

ANNs are a specialized set of algorithms that follow the neuron structure of the human brain and are used in various sorts of classification or pattern recognition tasks. This learning can take a varying amount of time depending on the complexity of the problem and this is basically how humans learn. ANNs try to mimic this high order learning behaviour to find relations in data sets or to model unknown transfer functions. The basic unit of computation in an artificial

neural network is called a neuron. All the neurons in a neural network are arranged in layers. A neural network may contain multiples of different types of layers in them. The first or initial layer is called the input layer. This layer contains input neurons and any inputs to the neural network are applied through this layer. As this is the first layer, no processing is done here and the data is just forwarded to the internal layers. The layers between the first or input layer and the last or output layers are called hidden layers. Each hidden layer may hold varying amounts of neurons in them depending on the application. These neurons take weighted inputs from neurons in the previous layer and apply biases to them. After the weighted sums are taken and biases are accounted for, the outputs from each neuron may pass through an activation function and forwarded to the neurons in the succeeding layer. The final layer of a neural network is called the output layer. They also do similar computations as the neurons in the hidden layers and contain special activation functions that help to map the neural network outputs to the desired output format. Figure 2.2 depicts the basic structure of ANN [26].

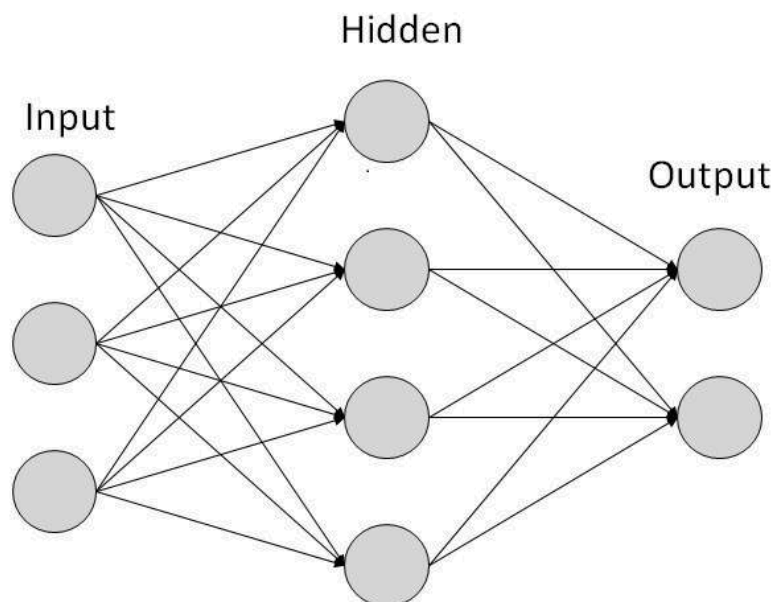


Figure 2.2: Basic Structure of ANN [26].

2.2.2 Convolutional Neural Networks (CNN)

A deep feed-forward neural network model known as a convolutional neural network is designed specifically for processing grid-like data, such as images. They have become a cornerstone in computer vision applications, demonstrating exceptional performance in tasks like image classification, object detection, and facial recognition and was first introduced by Yann LeCun in 1989 [27]. CNN has excellent feature extraction capabilities due to the construction of many filters, which it employs to extract representative features from input data layer by layer. Figure 2.3 depicts the basic structure of CNN [28].

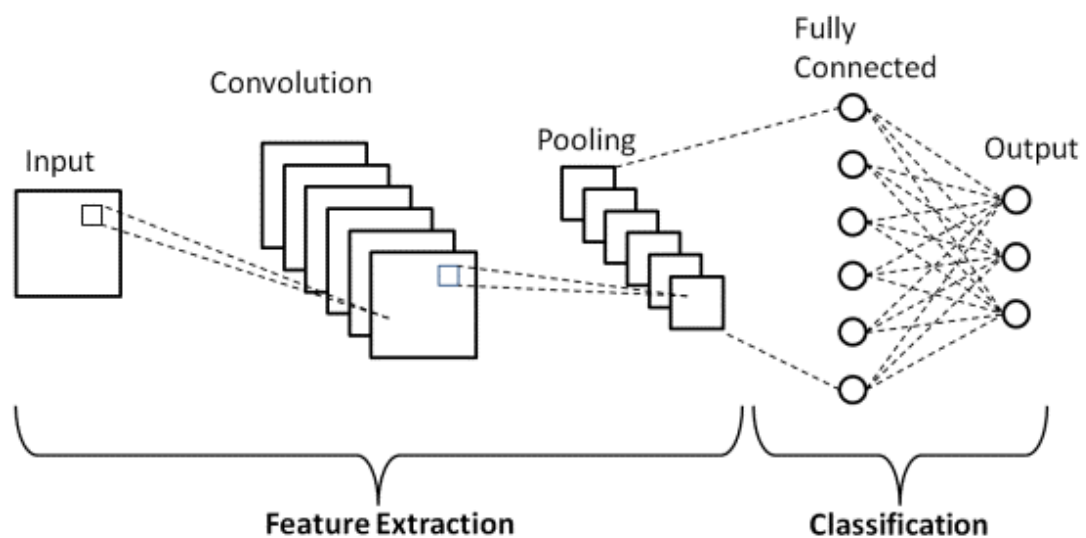


Figure 2.3: Basic Structure of CNN [27].

Convolutional Neural Networks (CNNs) typically consist of two main parts: the feature learning part and the classification part. the feature learning part focuses on extracting and understanding hierarchical features from the input data, while the classification part uses these learned features to make predictions or classifications. The combination of these two parts enables CNNs to excel in tasks such as image classification, object detection, and other computer vision applications.

2.2.2.1 Convolutional Layer

Convolutional layers are the fundamental components of CNNs. These layers execute convolution operations on the input image by sliding kernels across it. The convolution operation consists of element-wise multiplication of the kernel with a local portion of the input, resulting in a feature map that highlights patterns and features [29]. An error back propagation method determines these weights automatically. Figure 2.4 depicts the calculation technique principal schematic diagram of a convolution operation

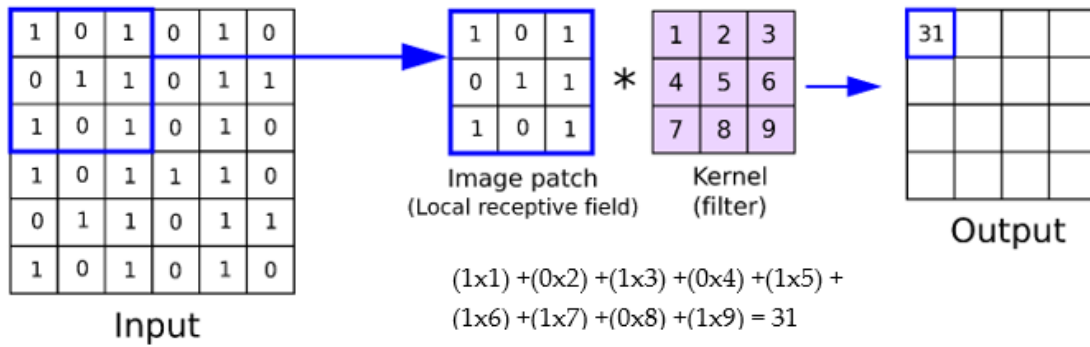


Figure 2.4: Visualization of the convolutional operation [26].

The convolution kernel can be shifted along with the up-down and left-right directions in accordance with the stride on the input feature map. The rectangular convolutional kernel can be used in the feedforward calculation process to traverse every element on the whole input feature map in order to obtain the output feature maps of the convolutional layer. A multichannel input feature map's convolution process is as follows:

$$\mathbf{X}_i^k = \sum_{c=1}^C \mathbf{W}_i^{(c,i)} * \mathbf{X}_{i-1}^c + \mathbf{B}_i^k \quad (2.1)$$

Where $*$ is the convolutional operator; i is the network layer index; k is the convolution layer's index number; the k th is the group convolution kernel; C is the channel index number; the input feature map of channel c is \mathbf{X}_{i-1}^c ; The layer's convolution kernel's weight is $\mathbf{W}_i^{(c,i)}$; \mathbf{X}_i^k is the k th output feature map obtained

following convolution of the k th group convolution kernel with the input feature map, The bias of the i^{th} layer's k th group filter is B_i^k .

2.2.2.2 Activation Functions

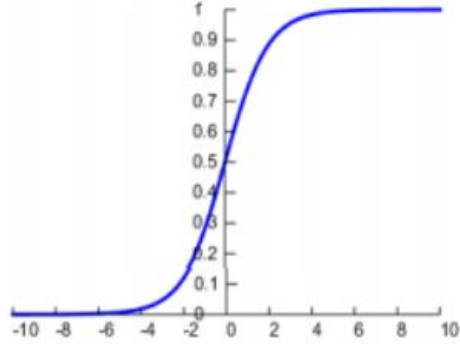
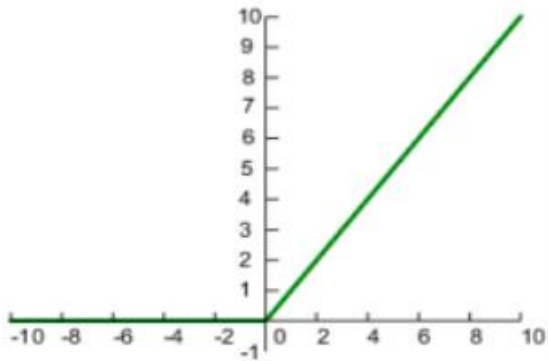
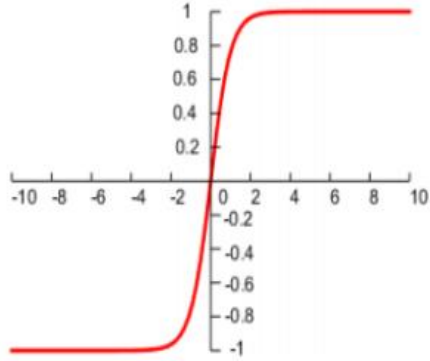
Following the convolution operation, the output of the convolution layer is subjected to a nonlinear transformation activation function [29]. Nonlinear transformations can be obtained using various activation functions. This helps the model learn complex relationships and improves the network's capability to represent intricate patterns. In a CNN model, activation functions decide whether a neuron is activated or not. It determines whether or not the input to the work is relevant enough for prediction using mathematical procedures. After including the activation function, Equation (2.2) may be rewritten as follows:

$$A_i^{(k)} = f(\sum_{c=1}^C W_i^{(c,l)} * X_{i-1}^c + B_i^k) \quad (2.2)$$

Where $f()$ denotes the activation function and $A_i^{(k)}$ denotes the k th output feature map of a nonlinear transformation.

The Sigmoid, hyperbolic tangent, and ReLU are the most often utilized activation functions in CNN. The sigmoid's input value x is $-\infty \sim +\infty$, and the sigmoid's output value $f(x)$ is 0–1. The output value $f(x)$ of tanh functions is $-1-1$. Although both of these functions are better at nonlinear transformation, they also have an issue with gradient fading and saturation. The output change values $f(x)$ of the two functions practically equal zero when the input value x 's absolute value is high. To address this issue, the ReLU activation function only evaluates the forward signal. The ReLU activation function only takes into account the forward signal, eliminating the influence of the negative signal, and it has an excellent fitting ability and sparsity, which considerably improves calculation efficiency [45]. Table 2.1 shows the equations and graphs.

Table 2.1: Activation functions in a CNN

Name of the function	Expressions	Graphs
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	
ReLU	$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$	
Tanh	$f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$	

2.2.2.3 Pooling Layer

The Pooling Layer is a critical component of Convolutional Neural Networks (CNNs), usually implemented after the Convolutional Layer. Its major goal is to downsample the convolved feature maps, thereby reducing their spatial dimensions and lowering computational costs [30]. This downsampling aids in the extraction of vital information from feature maps while removing unnecessary features, hence increasing the network's computational efficiency. It serves as a link between the Convolutional Layer and the Fully Connected Layer in the network design.

The Pooling Layer works independently on each feature map produced by the Convolutional Layer. It consists of various pooling operations, each of which has a different effect on the downsampling process. One frequent method is Max Pooling, which selects the largest element inside a predetermined region of the feature map. This enables the network to focus on the most important elements. Average Pooling calculates the average value of items inside a given segment, resulting in a simpler summation of information. Furthermore, Sum Pooling calculates the total sum of components in a designated region, which contributes to a different downsampling approach.

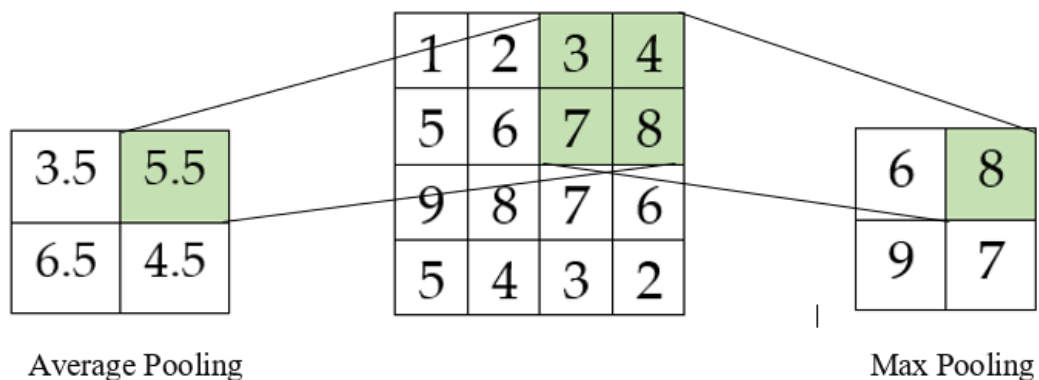


Figure 2.5: Visualization of the pooling operation.

2.2.2.4 Fully Connected Layer

The Fully Connected (FC) Layer is a vital component in recent neural network architectures, considerably improving the model's capacity to learn and predict. The FC Layer, which is typically located after the convolutional and pooling layers, is critical in synthesizing and integrating high-level features derived from the previous layers. This layer connects every neuron in the preceding and following layers, resulting in a firmly interconnected structure. This architecture allows the network to identify complex relationships, interconnections, and patterns in the data.

In the traditional CNN architecture, a fully connected layer is used in the classification step. In a classic CNN, the fully connected layer, which typically consists of two to three layers, makes complete connections with a feedforward neural network. The flatten function in a CNN is used to convert the final output feature map to a one-dimensional array [31]. This layer extracts additional features from the CNN's output data and connects the feature extraction stages to the SoftMax classifier. All neurons inside the layers are intimately coupled, resulting in a completely integrated network, as described below:

$$Y_{FC} = \sigma (W_{FC} \cdot X_{flatten} + B_{FC}) \quad (2.3)$$

Y_{FC} is the output of the Fully Connected Layer, $\sigma(\cdot)$ represents the activation function., W_{FC} is the weight matrix for the FC Layer, determining the impact of each input. $X_{flatten}$ is the flattened input vector from the preceding layer. B_{FC} is the bias vector, introducing an offset to the output.

In this equation (2.3), the FC Layer computes a weighted sum of the input features ($X_{flatten}$), applies an activation function, and adds a bias term to produce the final output (Y_{FC}). This process allows the network to learn complex relationships and representations from the flattened features, facilitating tasks such as classification or regression.

2.3 DILATED CONVOLUTIONAL NEURAL NETWORK

In deep learning, increasing the size and depth of the convolution kernel expands the receptive field while also increasing the number of parameters in the network. Dilated convolution can widen the receptive field by inserting weights of zero in the normal convolution kernel, without increasing the extra network parameters. A dilation rate parameter determines the spacing between values in the convolutional filter. A dilation rate of 1 corresponds to conventional convolution, while rates greater than one cause gaps between kernel values. Figure 2.6(a) shows a 1-dilated convolution kernel as same as the traditional convolution kernel of 3×3. Figure 2.6(b) shows the dilated convolution kernel, which has the same weights as the standard kernel (3×3) in Figure 2.6(b), but the receptive field has been increased to 5×5.

1	2	3
4	5	6
7	8	9

(a)

1	0	2	0	3
0	0	0	0	0
4	0	5	0	6
0	0	0	0	0
7	0	8	0	9

(b)

Figure 2.6: (a) Normal Convolutional kernel (b) Dilated Convolutional kernel.

Equation (2.4) defines * the convolution operator, which uses 1-D dilated convolution with dilation rate $l = 1$ to combine input picture F with kernel k . This one-dimensional convolution is known as a conventional convolutional neural network [32]. When the dilation l increases, the network is called a dilated convolutional neural network.

$$(f * k)(i) = \sum_{s+p=i} f(s)k(p) \quad (2.4)$$

When a dilation factor called l is introduced and by generalizing this factor l that can be defined as,

$$(f * lk)(i) = \sum_{s+lp=i} f(s)k(p) \quad (2.5)$$

Here l is referred to as the dilation rate of the convolutional neural network.

2.4 PRETRAINED MODELS

Pretrained models in deep learning play a pivotal role in leveraging the knowledge gained from extensive datasets to boost the performance of specific tasks. These models, such as VGG16, AlexNet, and ResNet50, are initially trained on large datasets like ImageNet, where they learn to extract high-level features from diverse images. This learned knowledge can then be transferred to new, related tasks with smaller datasets, enabling more efficient training and often yielding better results. The versatility of pretrained models extends across various applications, including image classification, object detection, and face recognition [33]. By harnessing the learned representations from these established models, practitioners can benefit from the wealth of knowledge encapsulated in the pretrained weights, saving computational resources and time in training new models from scratch.

2.4.1 VGG 16

The VGG16 model, developed by the Visual Graphics Group at the University of Oxford, stands out in the field of computer vision due to its deep and uniform design. VGG16 has 16 layers, including convolutional and fully connected layers, and is designed in a sequential and simple manner. VGG16's distinct feature is its persistent usage of small-sized convolutional filters, primarily 3x3, throughout the network. This method enables the model to accurately capture complicated characteristics and patterns in the input data.

The input layer of the VGG-16 is 224×224 in size [34]. The convolutional layers of VGG16 are distinguished by the repetitive use of 3x3 filters, each followed by a rectified linear unit (ReLU) activation function. Max pooling layers with 2x2 filters are implemented to reduce the spatial dimensions of the feature

maps, which improves computational efficiency. The network culminates with three fully linked layers, culminating in the final layer that produces the classification result. Figure 2.7 shows the VGG16 architecture in detail.

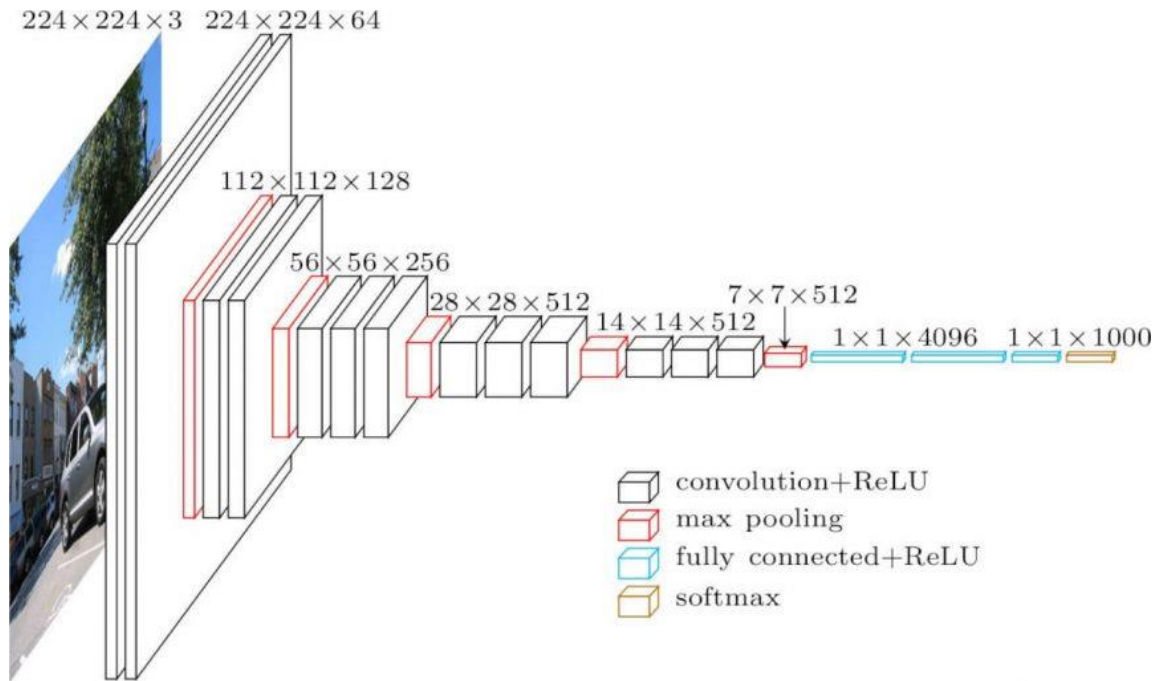


Figure 2.7: Architecture of the VGG16 [34].

2.4.2 AlexNet

AlexNet, an essential deep convolutional neural network, revolutionized the field of computer vision and deep learning. Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton designed this architecture specifically for the ImageNet Large Scale Visual Recognition Challenge in 2012 [35]. Five convolutional layers make up AlexNet: the first, second, third, and fourth layers, which are followed by the pooling layer and the three fully-connected layers in the fifth layer. The last fully connected layer comprises 1,000 neurons, aligning with the number of classes in the ImageNet dataset. During the back-propagation optimization process for the AlexNet architecture, the convolutional kernels are retrieved by using the stochastic gradient descent (SGD) technique to optimize the entire cost function. The input layer of the AlexNet is 227x227x3. To mitigate overfitting,

AlexNet introduces dropout in the fully connected layers during training. Figure 2.8 shows the AlexNet architecture in detail.

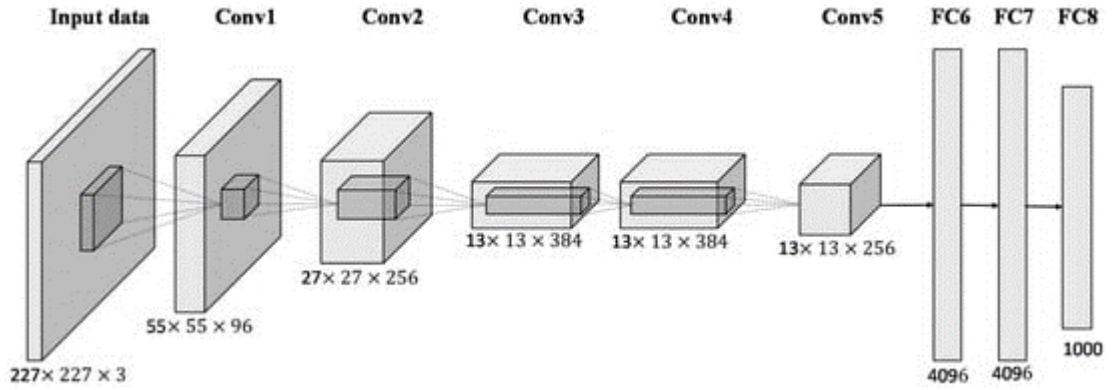


Figure 2.8: Architecture of the AlexNet [35].

2.4.3 ResNet-50

ResNet-50, which stands for Residual Network with 50 layers, is a deep convolutional neural network architecture notable for its unique usage of residual learning blocks. ResNet-50, developed by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, overcomes the challenges of training very deep networks by incorporating residual connections. The architecture is made up of 50 layers, comprising convolutional, pooling, and fully linked layers [36]. ResNet-50's distinguishing feature is the residual block, which has shortcut connections that enable the network to skip one or more layers. These skip connections allow the model to learn the residual, which is the difference between the input and output of a given block. Figure 2.10 depicts the residual block of this architecture. The residual building block consists of two convolutional layers, two batch normalizations (BN), and two ReLU activation functions. The operation of ResNet model is defined as:

$$y = f(x_i, W_i) + x \quad (2.6)$$

In the context where input and output vectors of a layer are denoted as x and y , respectively, the function $f(x_i, W_i)$ represents the residual mapping that the model aims to learn across multiple convolutional layers and operators.

Subsequently, the combination of feature maps occurs in an element-wise manner, proceeding channel by channel. The total number of parameters in this model is 44,611,648.

This method solves the vanishing gradient problem and allows for the training of extremely deep networks. ResNet-50 has excelled at picture classification tasks and has established itself as a benchmark in the field. Its influence extends to a variety of computer vision applications, helping to design deeper and more effective neural network structures.

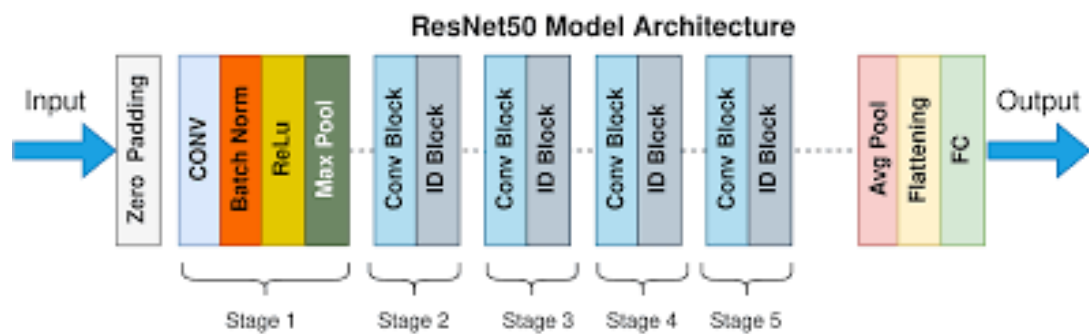


Figure 2.9: Architecture of the ResNet-50 [36].

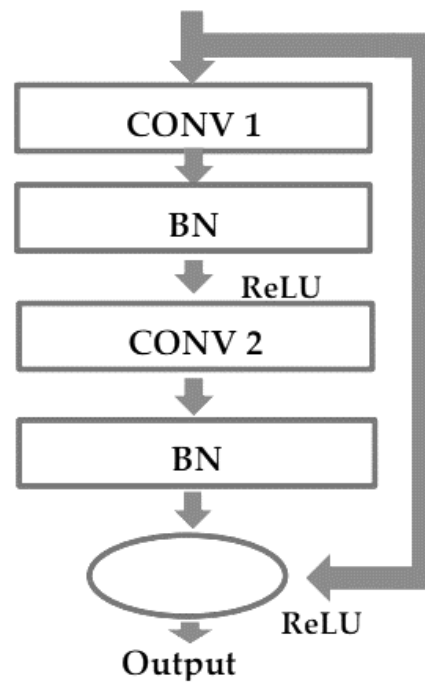


Figure 2.10: Residual building block of the ResNet50.

2.5 CLASSIFIERS

A classifier plays a pivotal role in the realm of data science, serving as a type of machine learning algorithm designed to categorize input data. A classifier is a type of machine learning algorithm that categorizes incoming data and plays an important role in data science. Classifiers are trained with labelled data, and the algorithm learns to correlate certain features with predefined labels. For example, in image recognition, a classifier is trained on a dataset with labels for each image. Once suitably trained, the classifier may assess new, unlabelled images and assign categorization labels to each. Classifier algorithms use complicated mathematical and statistical techniques to predict the probability of a given input being classified in a specific manner [36].

Deep learning includes a variety of classifiers, each adapted to a different purpose. Examples include the K-Nearest Neighbour (K-NN) method, Softmax classifier, Support Vector Machine (SVM), Random Forest (RF), and Naive Bias (NB). These classifiers have a variety of strengths and are selected based on the data's features and the nature of the classification task. The ability of classifiers to make accurate predictions highlights their importance in applications ranging from image identification to natural language processing, leading to the growth of machine learning and artificial intelligence.

2.5.1 SoftMax Classifier

The SoftMax classifier is widely used in machine learning for multiclass classification applications. It is particularly popular in neural networks, where it is used as the final layer to convert raw scores into probability distributions across several classes. The SoftMax function converts input scores to probabilities, making it appropriate for circumstances in which an input can belong to one of several distinct classes. Mathematically, the SoftMax function converts an input vector, often known as scores, into a probability distribution. For an input vector z , the SoftMax function (S) is defined as follows:

$$S(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (2.7)$$

Here, $S(z)_i$ represents the i^{th} element of the SoftMax output vector, e is the base of the natural logarithm, and the sum in the denominator is taken over all elements in the input vector. The SoftMax function exponentiates the input scores, effectively amplifying the differences between them, and then normalizes the results to produce a probability distribution.

The Softmax layer is frequently employed as the output layer in neural networks to solve multiclass classification issues. The network computes initial scores for each class, which are then fed into the Softmax function to provide class probabilities. It takes values between 0 and 1, with 0 representing impossibility and 1 representing certainty. The predicted class is usually the one with the greatest chance. One important feature of the Softmax function is that it assures that the predicted probabilities add to one [37], making it interpretable as a probabilistic distribution. The cross-entropy loss is frequently used together with the Softmax classifier to calculate the difference between predicted and actual distributions during training.

2.5.2 SVM Classifier

Support Vector Machine (SVM) is a robust machine learning method that is commonly used for both binary and multiclass classification applications. The basic idea of SVM is to determine the hyperplane that optimally separates data points from distinct classes while maximizing the margin between them [38]. This hyperplane is defined by support vectors, which are the data points closest to the decision boundary.

Mathematically, the decision function for an SVM with a linear kernel can be expressed as:

$$f(x) = \text{sign}(w \cdot x + b) \quad (2.8)$$

Here, $f(x)$ represents the decision function, \mathbf{w} is the weight vector, \mathbf{x} is the input feature vector, b is the bias term, and $\text{sign}(\cdot)$ is the sign function indicating the predicted class.

The optimization goal of SVM is to find the values of w and b that maximize the margin while satisfying the requirement that the data points are correctly classified. The margin represents the distance between the hyperplane and the nearest data point from each class. For a linearly separable dataset, the objective function of SVM is given by:

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

Subject to the constraints:

$$y_i(w \cdot x_i + b) \geq 1 \text{ for all } i \quad (2.9)$$

Here, y_i is the class label of the i -th data point, and $||w||$ denotes the Euclidean norm of the weight vector. Polynomial and RBF kernels are two of the most often utilized kernel functions for non-linearly separable classes in SVM. The polynomial kernel allows SVM to handle more complex decision boundaries by introducing non-linearity. The decision function becomes:

$$f(x) = \text{sign}((w \cdot x + b)^d) \quad (2.10)$$

Here, d is the degree of the polynomial, controlling the level of non-linearity introduced. The RBF kernel is particularly effective in capturing intricate, nonlinear relationships in the data. The decision function is defined as:

$$f(x) = \text{sign}(\sum_{i=1}^N \alpha_i y_i K(x_{i,x}) + b) \quad (2.11)$$

Here, N is the number of support vectors, α_i are the Lagrange multipliers, y_i are the class labels of the support vectors, and $K(\mathbf{x}_i, \mathbf{x})$ is the RBF kernel function. The RBF kernel is particularly powerful due to its ability to implicitly map the input space into a higher-dimensional space.

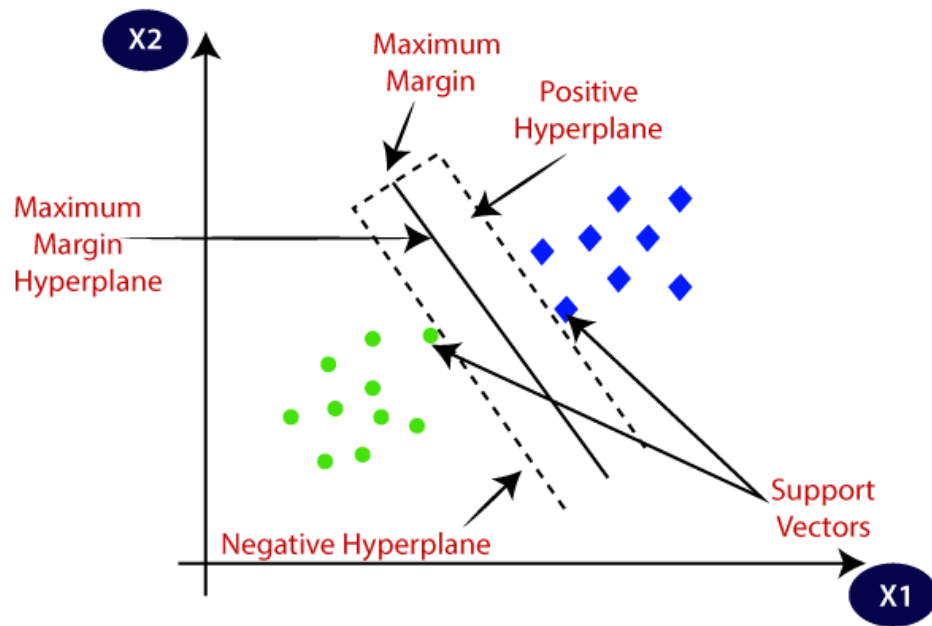


Figure 2.11: Visualization of the SVM classification processes [38].

2.5.3 KNN Classifier

The k-Nearest Neighbours (KNN) classifier is a non-parametric and simple machine learning technique that can be used for classification and regression. The basic idea of KNN is based on the assumption that data points with comparable attributes belong to the same class or exhibit similar behaviour. KNN classifies a data point based on the majority class of its k-nearest neighbours in the feature space. The technique calculates the distances between data points, commonly using Euclidean distance, and then chooses the k-nearest neighbours based on these distances [39]. KNN is simple and flexible, making it straightforward to comprehend and use, and it makes no strong assumptions about the underlying data distribution. KNN is especially effective for datasets with clear class boundaries, and it is widely used in a variety of fields, including image recognition, pattern recognition, and recommendation systems.

2.5.4 RF Classifier

The Random Forest (RF) classifier is a versatile machine learning method well-known for its performance in classification problems. RF, as an ensemble

learning method, generates a large number of decision trees during the training phase. Each tree is created by selecting a random subset of characteristics and data points, resulting in variation among the trees [39]. The ultimate classification is determined by a majority vote or average, with each tree contributing to the decision-making process. This ensemble technique improves the model's generalization capabilities while reducing overfitting, making RF especially adaptable to noisy or complicated datasets. Furthermore, RF gives useful information about feature importance, assisting in the identification of major contributors to the classification process. Because of its versatility, high accuracy, and capacity to handle enormous datasets, the Random Forest classifier is used in a variety of sectors, including finance, healthcare, and image identification.

2.5.5 Naïve Bayes Classifier

The Naive Bayes classifier is a probabilistic machine learning technique that is commonly used for classification applications, including natural language processing and document categorization. It is based on Bayes' theorem, which assesses the likelihood of a hypothesis given prior knowledge [39]. Despite its seemingly "naive" assumption of feature independence, Naive Bayes frequently performs excellently with high-dimensional datasets. Given a collection of features, the classifier estimates the likelihood of each class and assigns the input data to the class with the highest probability. Naïve Bayes is computationally efficient, making it ideal for real-time applications. The Naive Bayes classifier is a common choice for text classification problems due to its simplicity, speed, and success in circumstances when the independence requirement is met.

2.6 RELATED WORKS ON SATELLITE IMAGE CLASSIFICATION

The classification of imagery from satellites according to object attributes or semantic meaning is the process of categorizing them. Different categorization techniques exist, one based on low-resolution characteristics (broader patterns) and the other on high-resolution features (finer details) [41]. As seen in the

method covered in [42], which used texture with LBP for classification, the first category uses low-resolution characteristics and simple texture or shape features. Mid-resolution feature techniques work well for increasingly complex pictures and structures [43]. Remarkably, as compared to other methods, high-resolution feature methods are exceptionally successful at handling the complexity of imagery.

Several deep-learning methods have recently been used to classify satellite photos, and CNN has become a popular algorithm in this field. CNNs were used for ship detection by Mehran et al. [44], who trained Inception-ResNet and other current models on the Image-Net dataset beforehand. With the Dataset-RSI-CB256, Li et al. [45] concentrated on inadequate identifying interpretation in satellite imagery categorization and modified it for deep learning assessment. Gargees et al. [46] used the Dataset-RSI-CB256 to analyze scenery changes via profound visual aspects for change identification. A unique Deep Convolutional Neural Network (CNN) with an attention mechanism designed for scenario categorization in remote sensing was developed by Alhichri et al. [47]. This creative method computes a new feature map by providing the original feature map weights. The pre-trained EfcientNet-B3 CNN was improved using the attention mechanism to create the CNN, known as EfcientNet-B3-Attn-2. SatImNet is an accessible dataset with high compliance standards developed by Syrris et al. [48]. They also used the EuroSAT dataset to build a CNN for satellite imagery categorization. Notable for its flexible design, Yamashkin et al. [49] presented the GeoSystemNet model for high-resolution satellite image categorization. Testing the model on the EuroSAT dataset yielded a 95.30% success rate. The "greedy DropSample" training method was introduced by N. Yang et al. [50] to speed up CNN optimization for image classification. Samples that provide the most significant gradients are given priority using this strategy. However, given the lack of particular training samples, the training process may

display bias in network activations, resulting in the exclusion of samples with more minor losses. Interestingly, the created DropSample method does not consider class similarity.

The stacked layer architecture of traditional and pre-trained CNN models leads to overfitting because of an overwhelming number of learnable parameters, even if they perform exceptionally well in categorizing satellite images. Parallel CNN designs have been developed to solve this problem, enabling the simultaneous extraction of several discriminative features without deepening the network. This is accomplished by using various kernel sizes in each branch of the parallel structure and then fusing the features to create a robust feature vector. Atik [40] developed a transfer learning-based parallel convolutional network for satellite image classification. The study improves feature mapping by utilizing pre-trained knowledge from the transferred network within convolutional branches. The imbalanced class distribution is addressed using an offline augmentation technique, which generally improves network performance. Another study [51] developed REMSNet (Relation-Enhanced Multiscale Convolutional Network) for urban land cover classification. Leveraging DenseNet's connectivity reduces parameters while maintaining performance. Inception modules address fixed receptive fields, and relation-enhanced blocks capture global context, enhancing feature representation. The decoding stage incorporates parallel multi-kernel deconvolution modules and spatial paths to aggregate features at various scales.

Chapter 3: MATERIALS AND METHODOLOGY

This chapter discusses detail methodology and materials used in the research. Section 3.1 gives an overall description of the methodology which will be described in brief in the following sections. Section 3.2 describes the datasets used in the study, while Section 3.3 explains data pre-processing techniques. In Section 3.4, the proposed model architecture as a feature extractor is briefly discussed. Section 3.5 introduces the SVM classifier for categorizing extracted features from satellite images.

3.1 OVERVIEW OF METHODOLOGY

Conventional deep CNN models experience overfitting due to more layers and more learnable parameters. Furthermore, precise satellite imagery categorization requires extracting tiny details and larger patterns from the pictures. The simultaneous extraction, which is difficult for conventional CNNs, is made more accessible using a parallel structure in CNN models. This paper introduces OPCNet (Optimized Parallel CNN), a specialized satellite image categorization model.

The proposed framework developed for categorizing satellite images in this study is illustrated in Figure 3.1. The first stage entails gathering two publicly available satellite image databases. Essential pre-processing operations are then performed to reduce complexity and improve image characteristics. Next, an optimized Parallel Convolutional Neural Network (OPCNet) model is developed to extract several features at once. Three branches, each with a different size convolutional kernel (3×3 , 5×5 , 7×7), comprise this OPCNet model. The proposed OPCNet employs up to three convolutional layers in one branch, with fewer in others. The model has less than 1 million parameters, much less than the prior models. In order to improve nonlinear properties and expedite model convergence, ReLU is used as the activation function following each convolution

layer. After that, the features taken out of the convolution kernels of different sizes are combined to create a robust feature map. A dilatation factor is also included to increase the receptive field without increasing the total number of parameters in the proposed network. Finally, using the final feature map, satellite shots are categorized more accurately using the SVM classifier, alleviating the limitations caused by conventional CNNs.

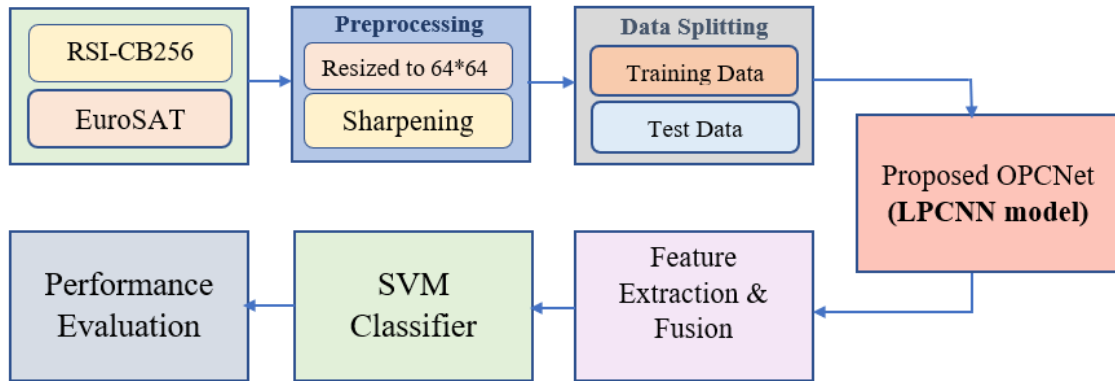


Figure 3.1: A proposed framework for satellite imagery classification.

Algorithm 1 describes the suggested model's sequential flow. Detailed explanations of each component within the framework are provided in the subsequent sections.

Algorithm 1: Proposed OPCNet-SVM model Algorithm

Input: Two different public datasets of Satellite Imagery of size $W \times H$ and class labels Y .

Output: Trained OPCNet-SVM model for satellite imagery classification.

Processing:

1. *for epoch = 1, 2, ..., N_{epochs} do*
2. *for image = 1, 2, ..., K do*
3. $Image_{resized} \leftarrow \text{Resize}(\text{image}, \text{Set } W = 64, H = 64)$
4. $Image_{enhanced} \leftarrow \text{Sharpen}(Image_{enhanced} = \text{imsharpen}(Image_{resized}))$

Lightweight Parallel Convolutional Neural Network:

5. Input image layer takes $Image_{enhanced}$ and send those to the initial convolution layer.

-
6. Divide the network into three parallel branches with convolution kernel size **3*3** for high resolution details, **5*5** for mid resolution details and **7*7** for low resolution details.

 7. Compute Batch Normalization $x = \frac{\gamma(x-\mu)}{\sigma} + \beta$, here x is input, μ is mean, σ is standard deviation of batch and γ and β are the shift and scale parameter.

 8. Compute ReLU activation function, $f(x) = \begin{cases} x & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases}$

 9. Apply max-pool plan, \mathbf{Z}_s to different feature maps $\mathbf{O}_s, \mathbf{O}_{s+1}, \dots, \mathbf{O}_{s+k-1}$ which takes maximum over the \mathbf{O}_s and maps it individually as represented in $\mathbf{Z}_{s,i,j} = \max(\mathbf{O}_{s,i,j}, \mathbf{O}_{s+1,i,j}, \dots, \mathbf{O}_{s+k-1,i,j})$.

 10. Add dilation factor, $\mathbf{out}_{ij} = \sum_{k,l} \mathbf{kernel}_{k,l} * \mathbf{in}_{(i+k*dilation, j+l*dilation)}$ to the low and mid resolution feature path.

 11. Combine feature maps from the three branches.

 12. Apply *sgdm* optimizer to minimize error rate.

 13. Pass the combined feature map through a fully connected layer preceded by dropout layer.

 14. Compute network's output: $\mathbf{z}_{out} = \mathbf{W}_{out} * \mathbf{a}_{fc} + \mathbf{b}_{out}$; $\mathbf{a}_{out} = \sigma(\mathbf{z}_{out})$

 15. Compute the loss between predicted and true class labels:

$$L(\theta) = -\frac{1}{B} \sum_{i=1}^B \sum_{j=1}^C y_{ij} \log(\mathbf{a}_{outij})$$
, where C is the no. of classes, y_{ij} is the true class labels and \mathbf{a}_{outij} is the predicted probability for sample i and class j .

 16. Update weights (\mathbf{W}) and bias (\mathbf{b}) using *back propagation*,

$$\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \alpha \frac{\partial L(\theta)}{\partial \mathbf{W}^{(l)}}, \quad \mathbf{b}^{(l)} \leftarrow \mathbf{b}^{(l)} - \alpha \frac{\partial L(\theta)}{\partial \mathbf{b}^{(l)}}$$

 17. Train the OPCNet model with *SVM* classifier.

 18. Calculate the test accuracy for the OPCNet-SVM model.

 19. Calculate error rate $e(t)$.

 20. *end*

 21. *end*

3.2 DATASET DESCRIPTION

The RSI-CB256 and EuroSAT datasets are the two unique datasets employed in this study to analyze the suggested OPCNet model. Figure 3.4 demonstrates the distribution of categories for both datasets.

3.2.1 RSI-CB256 Dataset

The publicly available Satellite Image Classification dataset RSI-CB256, provided in [52], has four separate types derived from sensors and Google map views. There are four categories titled "cloudy (1500)," "desert (1131)," "green area (1500)," and "water (1500)," with 5631 images in total. The size of every image in the collection is 256 x 256 pixels. Every image has the *.jpg extension. Figure. 3.2 show sample photos from the dataset.

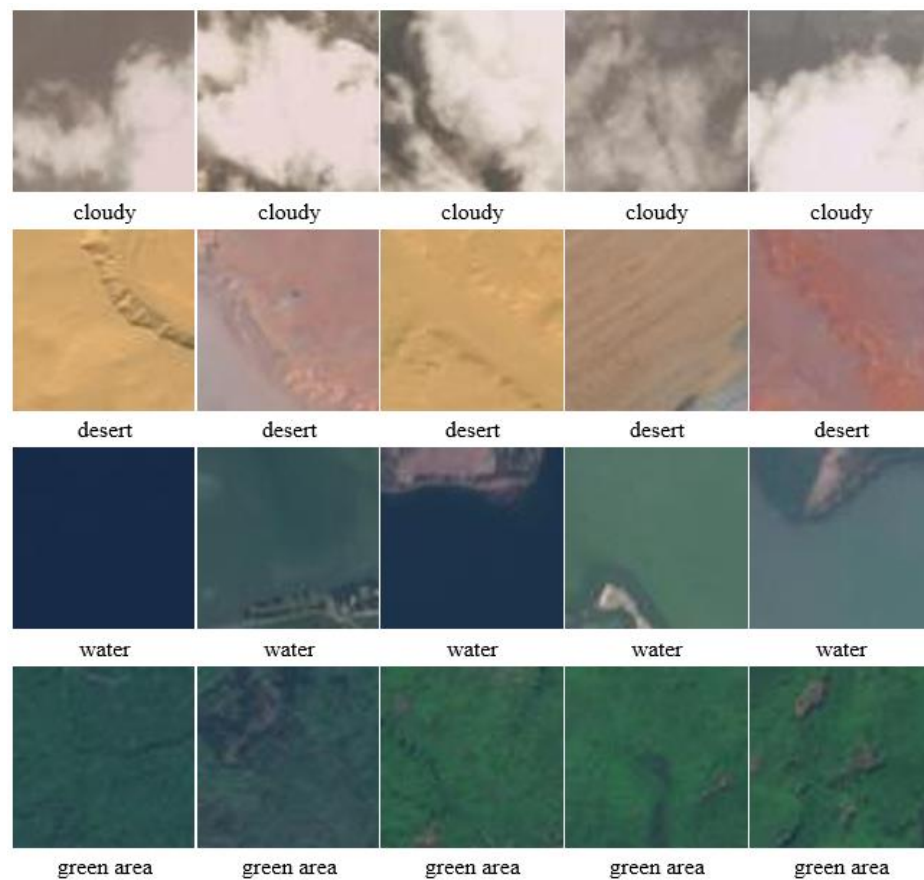


Figure 3.2: Samples from the Dataset-RSI-CB256 of satellite images [52].

3.2.2 EuroSAT Dataset

The German Aerospace Center (DLR) developed the EuroSAT dataset as a portion of the BigEarthNet project, which attempts to offer an ample standard dataset for categorizing satellite imagery. This benchmarking dataset [53], which includes 27,000 shots distributed over ten classes collected from 34 different European states, is utilized in this study. Each class has 2,000-3,000 pictures, each

of which is 64×64 pixels in size. EuroSAT dataset is based on images from the Sentinel-2 satellite. The classifications include industrial, permanent crops, annual crops, pasture, forests, herbaceous vegetation, highways, rivers, residential and sea lakes. Figure 3.3 shows a few illustrations from the collection.



Figure 3.3: Samples from the EuroSAT Dataset of satellite images [53].

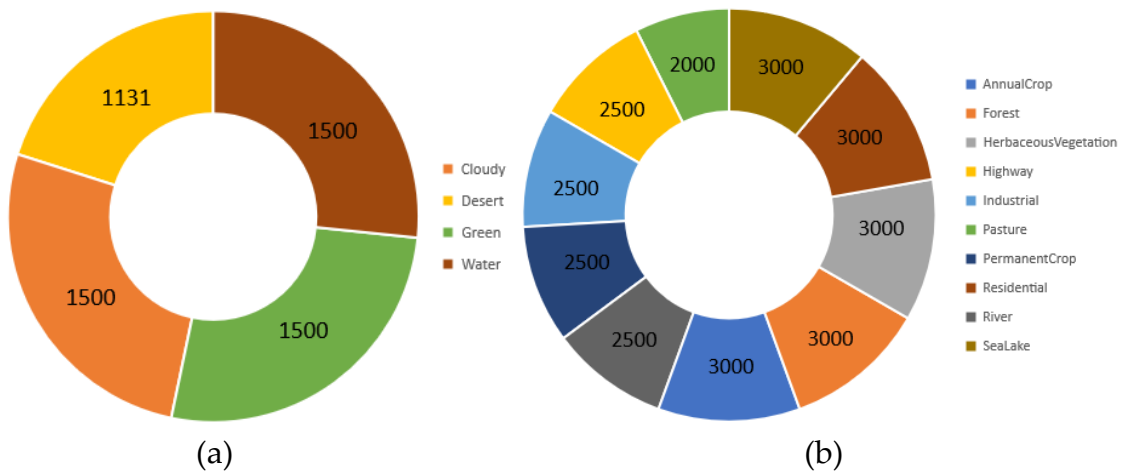


Figure 3.4: Doughnut chart for category distribution of (a)RSI-CB256 (b) EuroSAT Datasets.

3.3 PREPROCESSING

Image pre-processing is essential in various domains, such as computer vision and medical imaging, to assure accuracy, improve pattern identification, and enhance subsequent image analysis. In satellite image classification, image pre-processing is an essential step that improves the precision and efficiency of subsequent research. In this case, resizing and sharpening are two vital steps. They are described below:

3.3.1 Image Resizing

Resizing images in satellite image pre-processing is an essential first step that reduces computing complexity, standardizes input dimensions, and maintains critical spatial characteristics. A compromise between computing efficiency and information preservation is achieved by rescaling the satellite images to 64x64 pixels. This method enhances the discernible clarity of distinct features, which makes it very useful for analysing satellite images where slight variations might represent significant facts. The following equation (3.1) is used to resize the satellite images for further analysis.

$$X_{resized}(i,j) = X\left(\frac{i}{4}, \frac{j}{4}\right) \quad (3.1)$$

Here, $X_{resized}(i,j)$ is the resized satellite image, $X(i,j)$ is the original satellite image, and s_i, s_j are the scaling factors for the horizontal and vertical dimensions, respectively.

3.3.2 Image Sharpening

In satellite image analysis, sharpening is an essential pre-processing method. In satellite imaging, even the most minor appearances (buildings, vehicles, or individual trees) could indicate significant information. Sharpening the satellite images makes distinguishing small changes in the landscape, buildings, or natural features easier. This is important because it can highlight

essential structures and patterns, which can later help with the categorization process. The Laplacian operator is one often used for image sharpening. The resized satellite images from the previous stage are sharpened using the following formula (3.2) with the Laplacian operator.

$$X_{sharpened}(i,j) = X_{resized}(i,j) + \alpha * \nabla^2 X_{resized}(i,j) \quad (3.2)$$

Here, $X_{sharpened}(i,j)$ is the sharpened satellite image. α is the enhancement factor. $\nabla^2 X_{resized}(i,j)$ represents the Laplacian of the resized satellite image, which highlights rapid intensity changes (edges). Figure 3.5 demonstrates the samples of preprocessed satellite images.

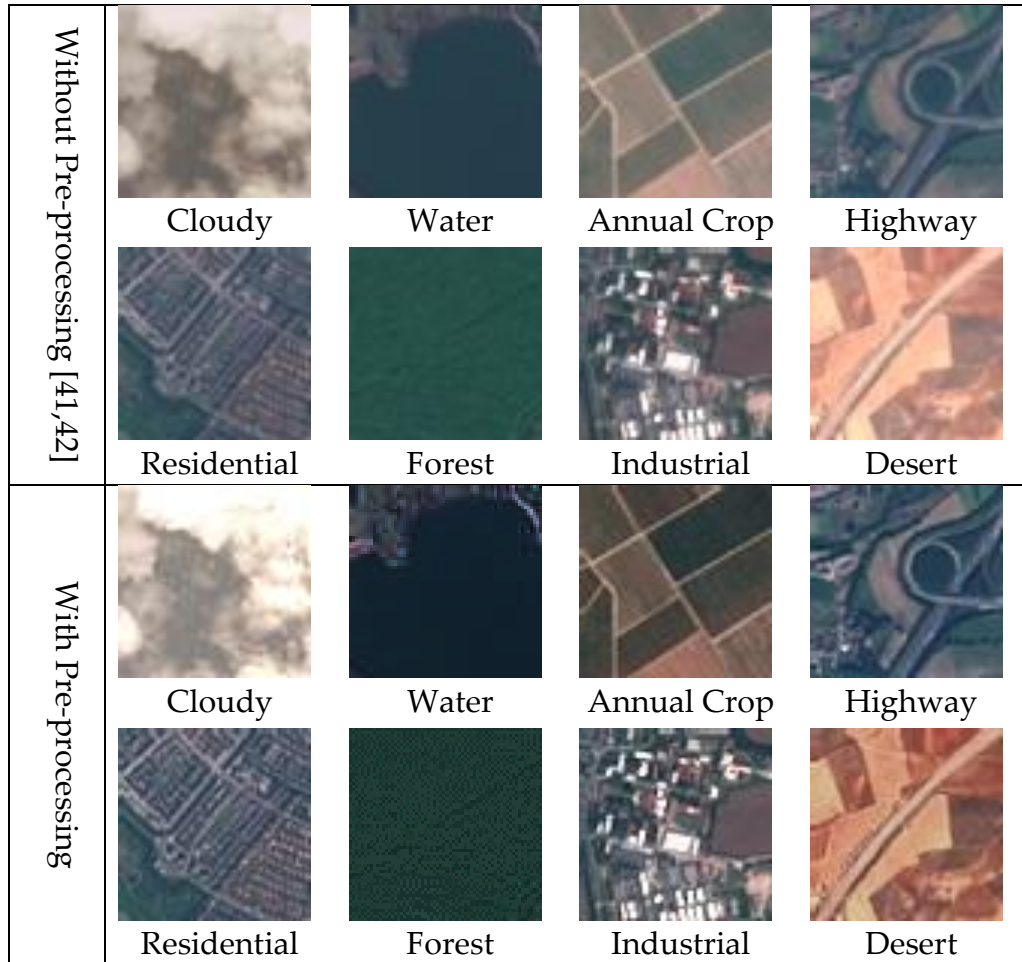


Figure 3.5: Pre-processed samples of Satellite images.

Following image processing, we divide our datasets into training and test sets using an 8:2 ratio. This ratio is critical for model training and assessment to

be effective. Allocating 80% of the data to the training set allows the model to learn complicated patterns, while the remaining 20% in the test set serves as an independent standard for evaluating the model's ability to generalize to new, previously unknown data.

3.4 PROPOSED OPCNET ARCHITECTURE

Satellite images exhibit a wide range of characteristics, including broad patterns, global spatial features, as well as complex texture patterns and intricate details. For precise classification of these satellite images, different kernel sizes, notably 3x3, 5x5, and 7x7, are necessary to extract those features efficiently from the images. However, typical deep CNN models with a single branch face considerable challenges in obtaining these various properties simultaneously. To address these challenges, this study proposed OPCNet model that aims to extract features simultaneously over a range of resolutions (low to high) while decreasing layers and parameters, boosting processing speed for detailed and broader pattern extraction. Smaller kernels (3x3) are better at revealing fine-grained details and high-resolution characteristics, but larger kernels (7x7) are better at acquiring broader patterns in low-resolution features. The key concept is that each branch of a parallel network concentrates on extracting information at distinct levels of abstraction, allowing the network to understand tiny details as well as broad patterns at the same time, which can increase classification accuracy. This model, distinguished by its lightweight design compared to existing state-of-the-art models, will be known to as the Lightweight Parallel CNN (LPCNN) for subsequent study.

The architecture of the LPCNN model begins with an input layer for 3-channel input from 64x64 pixels imagery in which the sharpened image is taken as input. The first convolution (ConV) layer has 16 output filters, each with a 5x5 kernel, followed by Batch normalization and ReLU function. The following equations (3.3) to (3.5) can be used to compute the output of these layers.

$$X_{conv}(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X_{sharpened}(i + m, j + n) \cdot W_{conv}(m, n) + b_{conv} \quad (3.3)$$

$$X_{bn} = \gamma_{bn} \cdot \frac{X_{conv} - \mu_{bn}}{\sqrt{\sigma_{bn}^2 + \epsilon}} + \beta_{bn} \quad (3.4)$$

$$X_{relu} = \max(0, X_{bn}) \quad (3.5)$$

Where, X_{conv} , X_{bn} , X_{relu} are the outputs of the convolutional layer, batch normalization layer, ReLU function. $W_{conv}(m, n)$ is the kernel element at position (m, n) , M and N are the dimensions of the kernel where $M = 5$, $N = 5$. b_{conv} is the bias term. γ_{bn} , β_{bn} are the scaling factor, shifting factor, μ_{bn} , σ_{bn} are mean and variance of input and ϵ is the stability constant.

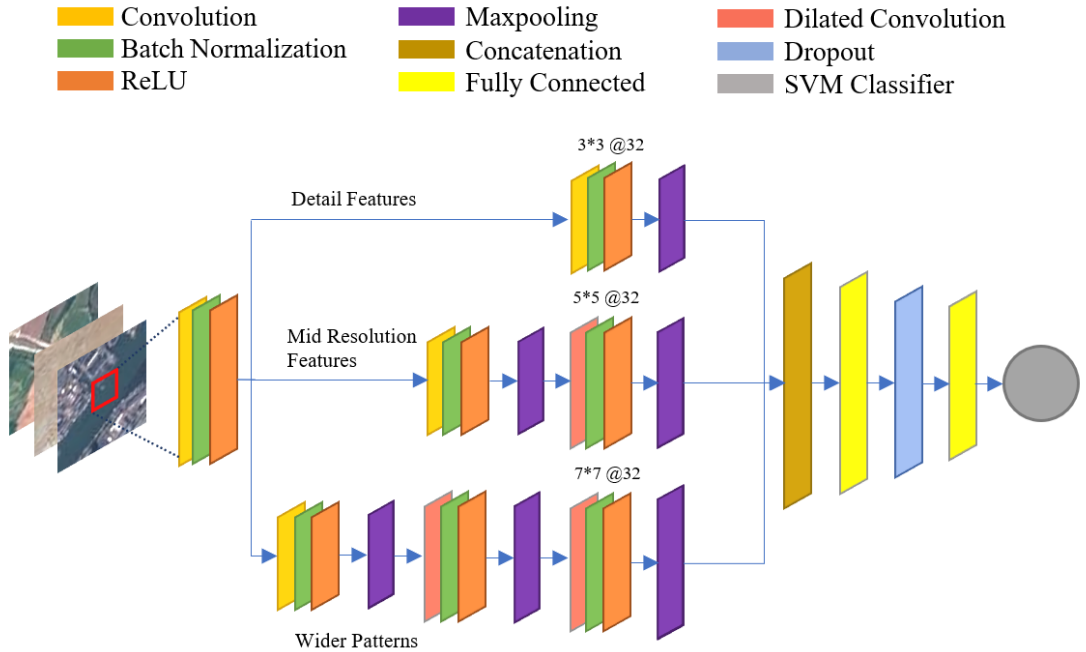


Figure 3.6: The proposed lightweight parallel convolutional neural network.

Following that, the model splits into three branches, each dedicated to extracting features from different resolutions. The details are given below:

3.4.1 Branch 1 (3x3 Kernel Size): Extracting Detail Feature

Satellite images have textured patterns, tiny objects like residences and roads, and edges of land covers. With a 3x3 kernel size, the network's first parallel

branch aims to extract those more detailed features from satellite images. This kernel size is good at capturing tiny structures (buildings, vehicles or even individual trees), subtle characteristics, and complex patterns in the images. This branch can efficiently acquire intricate urban infrastructure, tiny structures, fine-scale land use changes, and texture patterns. This adds to an in-depth understanding of the satellite-captured sceneries. This branch takes the X_{relu} as input and performs the convolution operation with 3x3 kernel size on it as per following equation (3.6):

$$Y_{conv1}(i, j) = \sum_{m=0}^2 \sum_{n=0}^2 X_{relu}(i + m, j + n) \cdot W_{conv1}(m, n) + b_{conv1} \quad (3.6)$$

In the proposed LPCNN model, a Batch Normalization (BN) layer and a ReLU layer are introduced after each convolution layer. ReLU is utilized to augment non-linear features, while BN standardizes each layer's input. The outputs of these layers are calculated through the following equations (3.7) and (3.8):

$$Y_{bn1} = \gamma_{bn1} \cdot \frac{Y_{conv1} - \mu_{bn1}}{\sqrt{\sigma_{bn1}^2 + \epsilon}} + \beta_{bn1} \quad (3.7)$$

$$Y_{relu1} = \max(0, Y_{bn1}) \quad (3.8)$$

Where the symbols have the same meaning as previously stated for the respective layer. A max-pooling layer is immediately applied after the activated convolution layer in each branch. This process effectively reduces parameter volume and extracts the local maximum value from the input feature. The output of the pooling layer is then determined by:

$$Y_{pool1}(i, j) = \max_{m,n} Y_{relu1}(poolsize.i + m, poolsize.j + n) \quad (3.9)$$

3.4.2 Branch 2 (5x5 Kernel Size): Capturing Intermediate Structures

Satellite images also yield moderate-sized objects (for example, buildings and agricultural fields), terrain patterns, wider vegetation regions, and

significant textural characteristics. With a 5x5 kernel size, the second parallel branch extracts those intermediate-level structures and characteristics from satellite images. This branch is good at classifying medium-sized structures in the context of satellite images, such as more significant buildings, fields of crops, and bodies of water. It is possible to extract features, including intricate geometric forms, medium-sized vegetation clusters, and road types. With an intermediate level of detail, this branch improves the model's capacity to identify various aspects of land cover. Similarly to Branch 1, the outcomes of each layer in this branch are determined using the following consecutive equations (3.10) to (3.13):

$$Y_{dilated_{conv2}}(i, j) = \sum_{m=0}^4 \sum_{n=0}^4 X_{relu}(i + r.m, j + r.n) . W_{conv2}(m, n) + b_{conv2} \quad (3.10)$$

$$Y_{bn2} = \gamma_{bn2} . \frac{Y_{dilated_{conv2}} - \mu_{bn2}}{\sqrt{\sigma_{bn2}^2 + \epsilon}} + \beta_{bn2} \quad (3.11)$$

$$Y_{relu2} = \max(0, Y_{bn2}) \quad (3.12)$$

$$Y_{pool2}(i, j) = \max_{m, n} Y_{relu2}(poolsize.i + m, poolsize.j + n) \quad (3.13)$$

$Y_{dilated_{conv2}}$, Y_{bn2} , Y_{relu2} , and Y_{pool2} represent the outputs of the last dilated convolution layer, batch normalization layer, max pooling layer, and ReLU activation function, respectively, while other symbols indicate their standard meanings.

To broaden the receptive field of kernels, a dilation factor has been incorporated into the convolution layer of this branch of the parallel network. Dilated convolution is a technique that expands the convolution field without increasing the number of network parameters by adding zero-weight components to the standard convolution kernel.

3.4.3 Branch 3 (7x7 Kernel Size): Grasping Broad Patterns

In order to extract broad patterns and large-scale characteristics found in satellite pictures, the third parallel branch uses a bigger 7x7 kernel size. This branch best identifies global spatial relationships and vast land cover features. This branch of the model allows it to effectively represent elements such as vegetation density, significant metropolitan areas, important water systems, large-scale landform changes, and climate patterns. Similarly to the prior branches, the outcomes of each layer in this branch are determined using the following consecutive equations (3.14) to (3.17):

$$Y_{dilated_{conv3}}(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X_{relu}(i+r.m, j+r.n) \cdot W_{conv3}(m,n) + b_{conv3} \quad (3.14)$$

$$Y_{bn3} = \gamma_{bn3} \cdot \frac{Y_{dilated_{conv3}} - \mu_{bn3}}{\sqrt{\sigma_{bn3}^2 + \epsilon}} + \beta_{bn3} \quad (3.15)$$

$$Y_{relu3} = \max(0, Y_{bn3}) \quad (3.16)$$

$$Y_{pool3}(i,j) = \max_{m,n} Y_{relu3}(poolsize.i + m, poolsize.j + n) \quad (3.17)$$

$Y_{dilated_{conv3}}$, Y_{bn3} , Y_{relu3} , and Y_{pool3} represent the outputs of the last dilated convolution layer, batch normalization layer, max pooling layer, and ReLU activation function, respectively, while other symbols indicate their standard meanings. The extracted features from these three branches are visualized in Figure 3.7(a), 3.7(b), and 3.7(c) respectively.

Following the extraction of more minor details and broader patterns via parallel branches, these features are combined using equation (3.18) in the concatenation layer, also shown in Figure 3.7(d).

$$Y_{concat} = \text{Concatenate}(Y_{pool1}, Y_{pool2}, Y_{pool3}) \quad (3.18)$$

The values of stride are adjusted to match the dimensions of the features from the three branches. The formula for the output size (Y_{pool}) of a pooling layer is:

$$Y_{pool} = \frac{Y_{relu}}{S} \quad (3.19)$$

For branch 1, the input size = 32x32; stride, $S = [4, 4]$. The following equation can be designed as:

$$Y_{pool1} = \frac{32}{4} \quad (3.20)$$

For branch 2, the input size = 16x16; stride, $S = [2, 2]$. The equation can be designed as:

$$Y_{pool2} = \frac{16}{2} \quad (3.21)$$

For branch 3, the input size = 8x8; stride, $S = [1, 1]$. The equation can be designed as:

$$Y_{pool2} = \frac{8}{1} \quad (3.22)$$

Now, all three branches have same dimension of feature map which is 8x8. Following that, two fully connected (FC) layers are used, and the features are gathered from the final FC layer.

$$Y_{fc} = \text{ReLU} (W_{fc} \cdot \text{Flatten}(Y_{concat}) + b_{fc}) \quad (3.23)$$

A dropout layer with a 0.5 probability is deliberately positioned between these two FC layers to reduce overfitting and speed up the training process. Dropout skips 50% of all nodes randomly, contributing to a more lightweight model. Table 3.1 provides a complete overview of the LPCNN model.

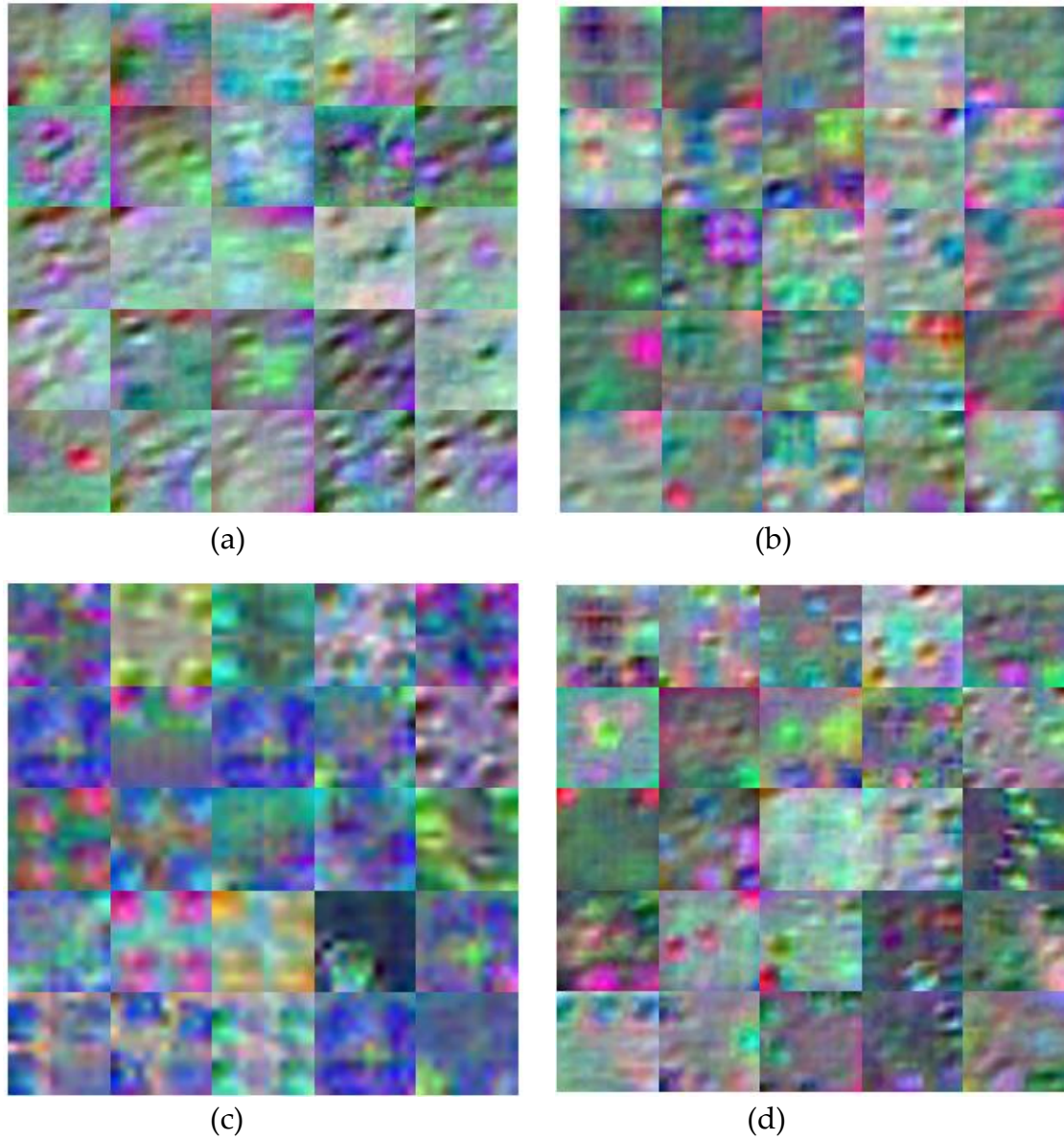


Figure 3.7: (a) High resolution (finer detailed) features with kernel size 3x3, (b) Mid resolution features with kernel size 5x5, (c) Low resolution (broader pattern) features with kernel size 7x7, and (d) Final feature map after concatenation layer.

Table 3.1: Details of the proposed LPCNN model.

Layer No.	Layer Name	No. of Kernel	Kernel Size	Stride	Dilation Factor	Activation	Learnable
1.	Image Input	-	-	-	-	64*64*3	0
2.	Conv_1	16	5×5	[1 1]	[1 1]	64*64*16	1216
3.	Batchnorm_1	-	-	-	-	64*64*16	32
4.	Relu_1	-	-	-	-	64*64*16	0

5.	Conv_1_1	128	3×3	[2 2]	[1 1]	32*32*128	18560
6.	Batchnorm_2	-	-	-	-	32*32*128	256
7.	Relu_2	-	-	-	-	32*32*128	0
8.	Maxpool_1_1	-	3×3	[2 2]	-	16*16*128	0
9.	Conv_1_2	64	5×5	[1 1]	[2 2]	16*16*64	204864
10.	Batchnorm_3	-	-	-	-	16*16*64	128
11.	Relu_3	-	-	-	-	16*16*64	0
12.	Maxpool_1_2	-	3×3	[2 2]	-	8*8*64	0
13.	Conv_1_3	32	7×7	[1 1]	[4 4]	8*8*32	100384
14.	Batchnorm_4	-	-	-	-	8*8*32	64
15.	Relu_4	-	-	-	-	8*8*32	0
16.	Maxpool_1_3	-	2×2	[1 1]	-	8*8*32	0
17.	Conv_2_1	64	3×3	[2 2]	[1 1]	32*32*64	9280
18.	Batchnorm_5	-	-	-	-	32*32*64	128
19.	Relu_5	-	-	-	-	32*32*64	0
20.	Maxpool_1_4	-	3×3	[2 2]	-	16*16*64	0
21.	Conv_2_2	32	5×5	[1 1]	[2 2]	16*16*32	51232
22.	Batchnorm_6	-	-	-	-	16*16*32	64
23.	Relu_6	-	-	-	-	16*16*32	0
24.	Maxpool_1_5	-	2×2	[2 2]	-	8*8*32	0
25.	Conv_3_1	32	3×3	[2 2]	[1 1]	32*32*32	4640
26.	Batchnorm_7	-	-	-	-	32*32*32	64
27.	Relu_7	-	-	-	-	32*32*32	0
28.	Maxpool1_6	-	2×2	[4 4]	-	8*8*32	0
29.	concat	-	-	-	-	8*24*32	0
30.	fc_1	-	-	-	-	1*1*100	614500
31.	dropout	-	-	-	-	1*1*100	0
32.	fc	-	-	-	-	1*1*4	404
							Total = 834,404

3.5 SVM CLASSIFIER

Integrating the OPCNet with a Support Vector Machine (SVM) is an advanced method for classifying satellite images. The parallel architecture greatly improves the model's ability to discern tiny details to identify more general patterns within images, which is intended to extract features simultaneously at different resolutions. The SVM classifier then uses the effectively extracted features to improve its resilience in navigating concatenated feature space. By combining the advantages of both models, this dynamic union offers a potent method for comprehensive and accurate satellite image categorization.

In this work, the SoftMax layer of the OPCNet model is replaced with the SVM classifier. SVM works by determining which hyperplane best divides various classes of satellite images in the feature space. The aim is to maximize the margin between different classes of satellite imagery. The MATLAB multiclass SVM classifier is used in this proposed approach. The SVM parameters list for the suggested classification is in Table 3.2. A linear kernel is utilized in the SVM model, which is presented as follows:

$$K(Y_{fc}, Y_{SVi}) = Y_{SVi}^T Y_{fc} \quad (3.20)$$

The optimal hyperplane in SVM is calculated as:

$$SVM(Y_{fc}) = \sum_{i=1}^S \theta_i y_i K(Y_{fc}, Y_{SVi}) + b_{svm} \quad (3.21)$$

So, for linear SVM's decision function for satellite imagery classification may be simplified as follows:

$$SVM(Y_{fc}) = \text{sign}(\sum_{i=1}^S \theta_i y_i Y_{SVi}^T Y_{fc} + b_{svm}) \quad (3.22)$$

Here, $SVM(Y_{fc})$ is the decision function which returns +1 if the input Y_{fc} belongs to the labeled satellite image category and -1 if it belongs to other

category, S is the number of support vectors, x_i are the support vectors, θ_i are the Lagrange multipliers, y_i is the class label, $K(Y_{fc}, Y_{svi})$ is the kernel function that computes the similarity between Y_{fc} and Y_{svi} , b_{svm} is the bias term for the SVM classifier.

Table 3.2: SVM classifier parameter

PARAMETERS	VALUES
SVM Type	Multiclass SVM classifier
SVM Model	One-vs-all
Solver	Linear Kernel

Chapter 4: RESULT AND ANALYSIS

Section 4.1 specifies the evaluation parameters for the proposed system, and Section 4.2 validates its lightweight design. The performance of several classification algorithms is compared in Section 4.3. Section 4.4 demonstrates how this method reduces overfitting in satellite image classification. Section 4.5 provides a simple parameter analysis for the proposed system. Finally, Section 4.6 compares the proposed system to similar works in the field.

4.1 EVALUATION MEASUREMENT OF PROPOSED SYSTEM

Several performance metrics, such as accuracy, precision, recall, and F1 score were used to evaluate the performance of the proposed framework. Equations (4.1) to (4.4) can be used to define the metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

$$Recall = \frac{TP}{TN + FP} \quad (4.3)$$

$$F1\ score = \frac{2 * (precision * recall)}{precision + recall} \quad (4.4)$$

Where the true positive, true negatives, false positives, and false negatives are symbolized as TP, TN, FP, FN respectively. True positive indicated that actual classes are predicted as those ones.

The proposed system's performance was assessed using various image dimensions for the RSI-CB256 dataset. Table 4.1 presents the results of the proposed method with different image dimensions. When the image dimensions are small, the proposed model's accuracy is also minimal. As image size increased, accuracy gradually improved. The proposed method in 64×64 image achieved 99.8% accuracy on the RSI-CB256 dataset. However, as the matrix size

increases, the accuracy decreases below the previous value. To record specialized features, the matrix size must be appropriate.

Table 4.1: Comparison result for different image dimensions.

Image Dimension	Training Accuracy	Test Accuracy
24 x 24	96.40 %	94.87 %
32 x 32	98.45 %	97.20 %
64 x 64	99.89 %	99.8 %
128 x 128	99.20 %	98.89 %
256 x 256	97.45 %	96.65 %

The proposed OPCNet model was developed and executed using MATLAB 2021 software. The training and testing processes occurred on a computer running a 64-bit Windows 10 Pro operating system. The hardware specifications include an Intel(R) Core (TM) i7-8550U CPU @ 1.80GHz and 8 GB of RAM.

4.2 VALIDATION OF LIGHTWEIGHT DESIGN

In satellite image categorization, the most widely used models frequently use transfer learning and intricate architectures like DenseNet121, which has 121 layers, or VGG16, which has 138.3 million parameters. On the other hand, our proposed model takes a more minimalistic approach. It has seven convolution layers, of which six are parallelized in two separate steps, making four convolution layers. The model's lightweight architecture is highlighted because its total parameter count is less than one million. This model works well with low-resolution shots and requires a 64x64 image size. Table 4.2 displays that the proposed model has the fewest layers and parameters. Table 4.3 shows that the proposed model requires less time than some existing pretrained models. These findings justify the use of the term "lightweight" for the suggested model.

Table 4.2: Simplicity Comparison of the proposed model to the state of art models.

Model Name	No. of Layers	No. of Parameters (million)
EfficientNet [12]	-	66
AlexNet [54]	8	61
VGG16 [39]	16	138.3
InceptionV3 [49]	48	23.8
ResNet50 [55]	50	25.6
ResNet101 [49]	101	44.7
DenseNet121 [49]	121	8
Proposed model	4	≤ 1

Table 4.3: Processing time of the proposed PCNN model to the state of art models.

Model Name	RSI-CB256 Dataset	EuroSAT Dataset
AlexNet	324.51 s	3774.58 s
VGG16	488.25 s	5209.28 s
ResNet50	1227.200 s	15639.18 s
Proposed Model	408.82 s	2458.95 s

4.3 PERFORMANCE ANALYSIS AND COMPARISON OF DIFFERENT CLASSIFIERS

The OPCNet model was utilized to classify satellite images as a compelling feature extractor. The process involved utilizing a Support Vector Machine (SVM) classifier to identify and classify the extracted features from the satellite images. The SVM classifier was used in place of the SoftMax layer, a significant modification that improved the classification process' accuracy. A comparison

study using several important Machine Learning (ML) classifiers, including SVM, K-Nearest Neighbor (KNN), Random Forest (RF), and Naïve- Baise (NB), is presented in Table 4.4 to ensure a complete evaluation of classification performance. SVM was discovered to be the outstanding performer among these classifiers, displaying superior results and supporting its usefulness in dealing with the complexities of satellite image information. This highlights the importance of the hybrid OPCNet-SVM strategy, which not only acts as an effective feature extraction mechanism but also integrates with SVM to obtain excellent outcomes for classification.

Table 4.4: Comparative analysis of performance matrices for different classifiers.

	Model	Accuracy	Error	Precision	Recall	F1-Score
RSI-CB256 Dataset	OPCNet	99.02	0.0098	99.06	99	99.03
	OPCNet-SVM	99.8	0.002	99.67	99.64	99.655
	OPCNet-KNN	99.29	0.0071	99.34	99.25	99.295
	OPCNet-RF	99.23	0.0077	99.17	99.2	99.185
	OPCNet-NB	98.85	0.0115	98.93	98.75	98.84
EuroSAT Dataset	OPCNet	94.6	0.054	92.19	88.86	90.494
	OPCNet-SVM	97.91	0.0209	95.81	94.76	95.282
	OPCNet-KNN	94.79	0.0521	93.75	93.7	93.725
	OPCNet-RF	92.3	0.077	90.79	89.88	90.333
	OPCNet-NB	91.85	0.0815	90.39	89.75	90.069

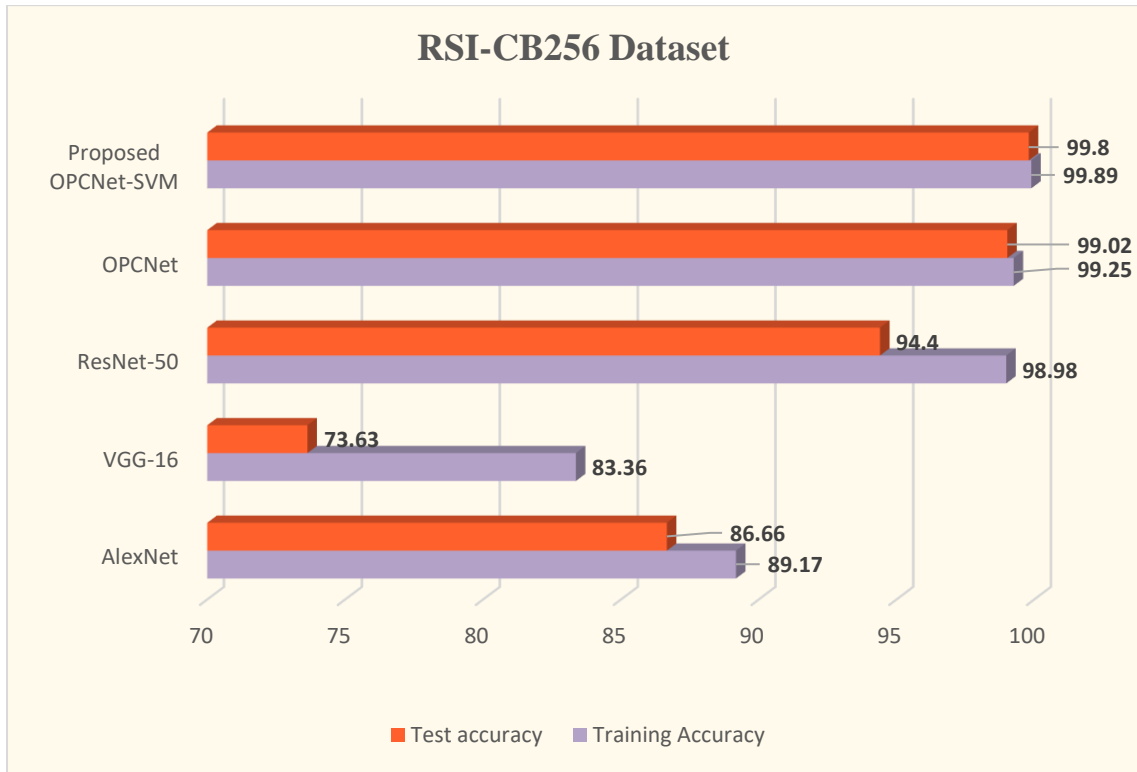
4.4 REDUCTION OF OVERFITTING PROBLEM

Overfitting occurs when a model gets highly good at capturing the intricate details of the training data, limiting its capacity to generalize efficiently to new,

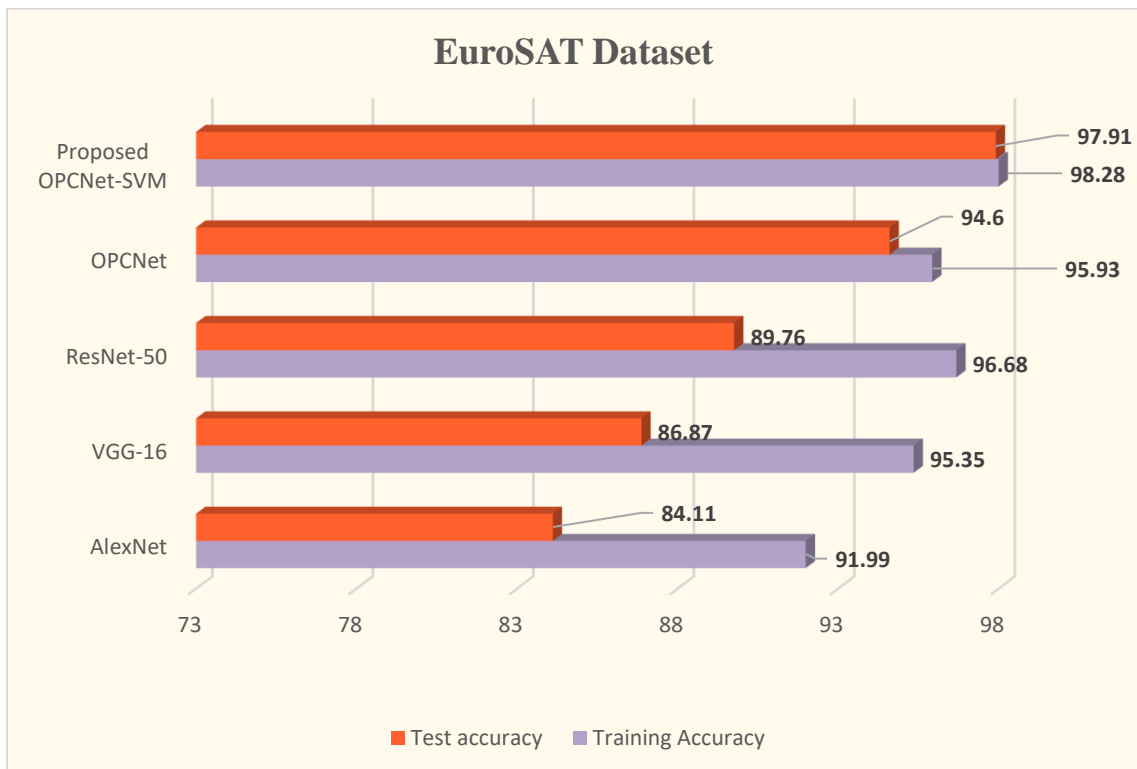
previously unknown data. This problem becomes more evident as the network's depth increases as layers increase, resulting in outstanding performance on the training set but a significant loss on the testing set. The leading cause of overfitting is the significant difference between training and testing accuracy, where training one is far more than testing one and vice versa in terms of error.

$$\textit{Overfitting} = \textit{Training Accuracy} \gg \textit{Test Accuracy} \quad (4.5)$$

The proposed OPCNet presented in this study is highlighted for its minimal number of layers and parameters. The model has included a dropout layer to handle the overfitting issues. A comparison of the training and testing accuracies of Deep CNN models (AlexNet, VGG-16, and ResNet-50), OPCNet, and OPCNet with SVM classifiers (OPCNet-SVM) is shown in Figure 4.1. Notably, for both datasets, the Deep CNN model shows a significant difference in test and training accuracies, indicating the overfitting problem. On the other hand, the OPCNet model's use leads to a significant decrease in the difference between test and training accuracy. Because the OPCNet model has fewer parameters and can merge various resolution features to build a robust feature map. The accuracy has increased when the OPCNet model and SVM classifier are combined for both datasets.



(a)



(b)

Figure 4.1: Training and testing accuracies for (a) RSI-CB256 dataset, and (b) EuroSAT dataset.

4.5 PARAMETER ANALYSIS OF THE PROPOSED SYSTEM

This subsection explores the effects of the proposed OPCNet-SVM model's parameters—dataset training-testing ratio, learning rate and batch size, dropout rate—during SGDM optimizer. Both the RSI-CB256 and EuroSAT datasets are utilized for the analysis of the parameters.

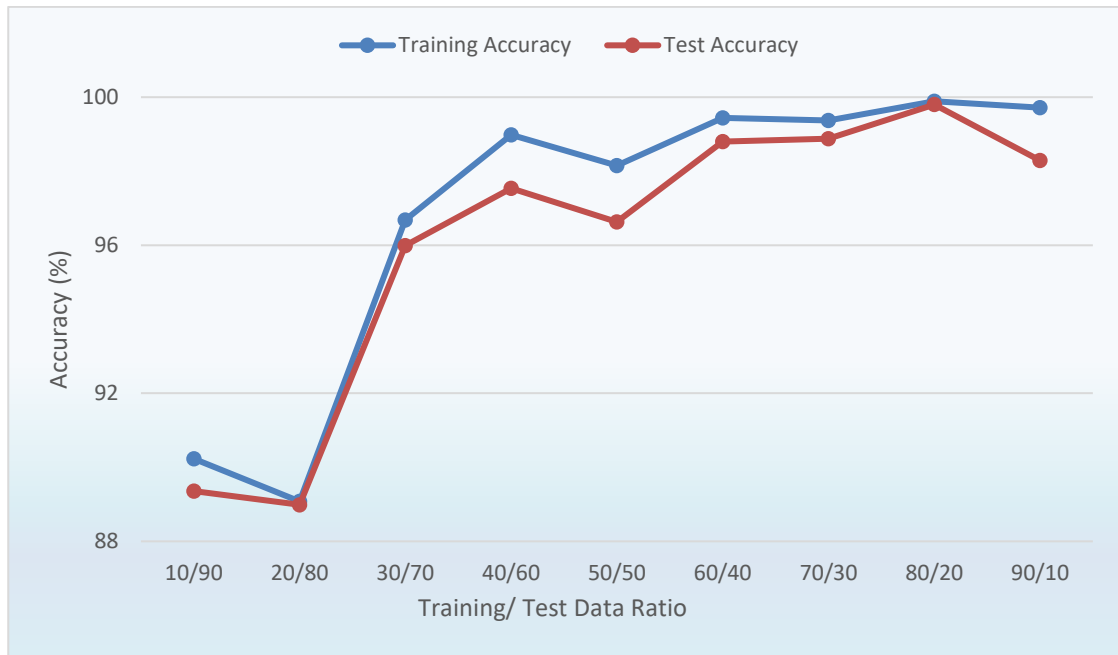
Tables 4.5 and 4.6 show the classification results of the proposed model over various training and test data splits for the RSI-CB256 and EuroSAT datasets, respectively. The model's learning performance deteriorates when trained with minimal data, as seen by the tables. Notably, training and test accuracy improve as the amount of training data increases. This is due to the model's better ability to detect varied patterns and variances in data with a more extensive training dataset, resulting in higher prediction accuracy on unknown test data. However, a massive amount of training data and a scarcity of test data might result in overfitting or introduce bias in the results. Figure 4.2 depicts the effect of the varying data ratios on accuracies for both datasets. A training-test data split of an 80-20 ratio has shown to be best in optimizing overall accuracy. This split offers a reasonable trade-off that guarantees efficient learning while preserving processing time on both datasets.

Table 4.5: Classification metrics on RSI-CB256 Dataset for different training-test data split.

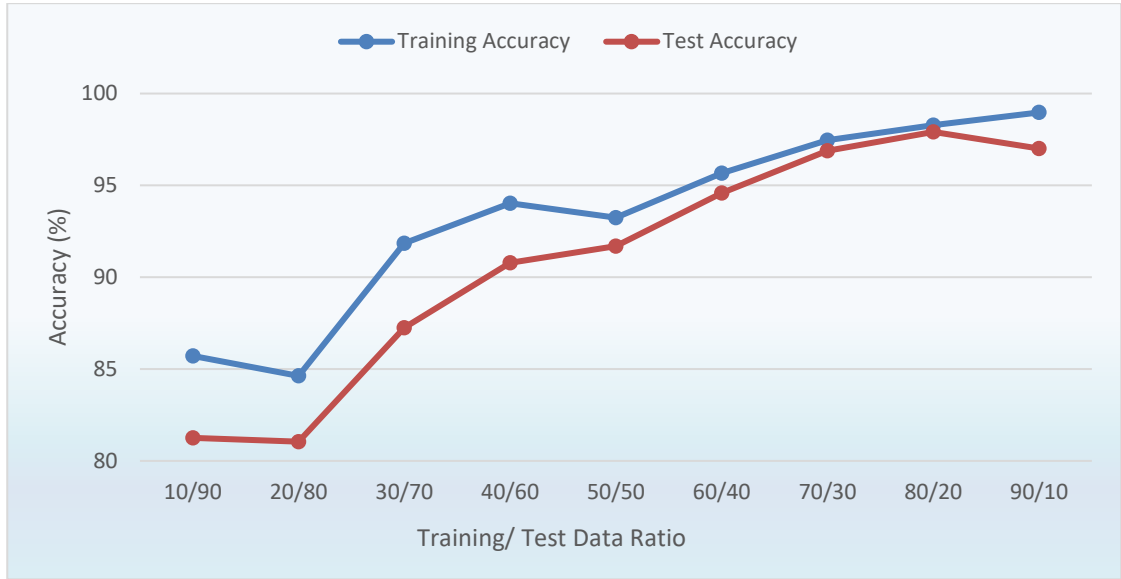
Parameter	10/90	20/80	30/70	40/60	50/50	60/40	70/30	80/20	90/10
Overall Accuracy	89.36	88.99	95.99	97.54	96.63	98.8	98.88	99.8	98.29
Training Accuracy	90.23	89.08	96.68	98.98	98.15	99.44	99.37	99.89	99.72
Precision	90.120	89.349	95.682	97.63	97.062	98.88	98.952	99.67	97.345
Recall	89.637	89.394	95.842	97.372	96.242	98.765	98.82	99.64	99.28
F1-score	89.486	88.575	95.751	97.475	96.52	98.82	98.882	99.654	98.307
Time (s)	137.26	284.21	388.87	405.25	940.38	632.78	529.23	408.82	364.59

Table 4.6: Classification metrics on EuroSAT Dataset for different training-test data split.

Parameter	10/90	20/80	30/70	40/60	50/50	60/40	70/30	80/20	90/10
Overall Accuracy	81.25	81.047	87.24	90.78	91.698	94.58	96.88	97.91	97.01
Training Accuracy	85.718	84.626	91.846	94.031	93.242	95.66	97.468	98.28	98.97
Precision	83.43	83.11	89.57	92.485	92.55	93.22	94.58	95.81	95.42
Recall	84.25	83.42	88.55	92.44	92.99	91.25	95.64	94.76	92.58
F1-score	83.838	83.265	89.057	92.462	92.769	92.224	95.107	95.282	93.979
Time (s)	823.57	1705.31	2333.26	2431.54	5642.31	3796.72	3175.43	2458.95	2187.58



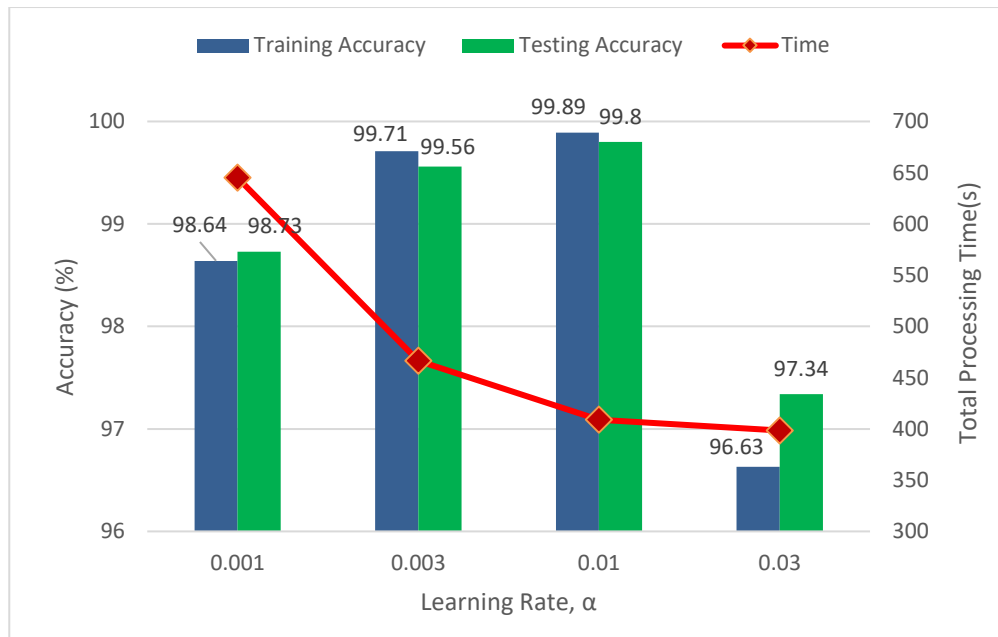
(a)



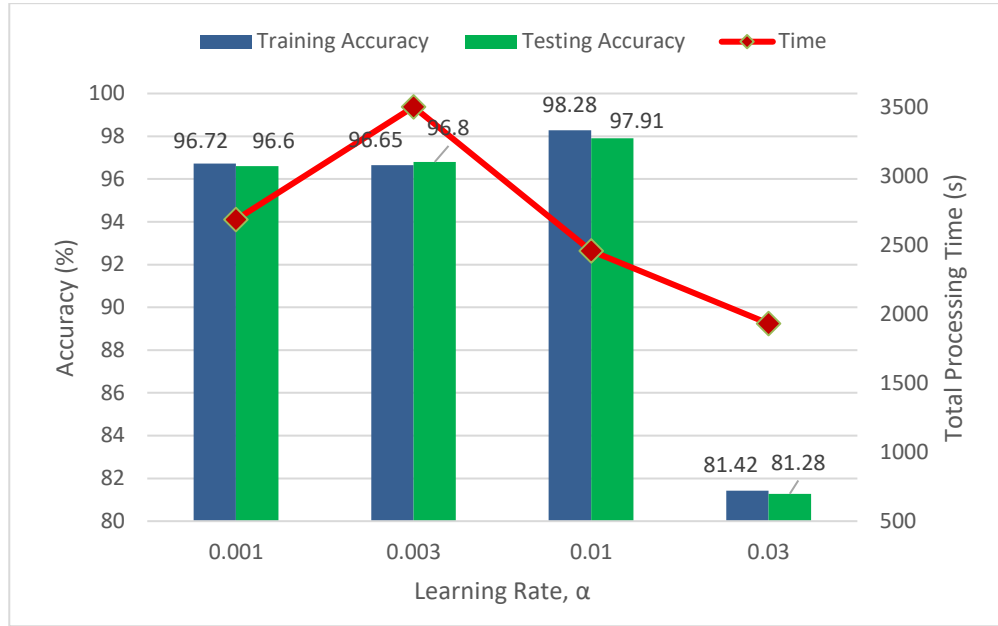
(b)

Figure 4.2: Training and test accuracy on different training-test data split for (a) RSI-CB256 dataset (b) EuroSAT dataset.

Figures 4.3(a) and 4.3(b) illustrate the effects of learning rate α on training, test accuracy, and also on time. It is evident that an excessively high or low value of α can result in a decline in performance, while a slightly higher learning rate can successfully shorten the training time. When was set to 0.01, the suggested OPCNet model produced the best outcomes in the shortest amount of time.



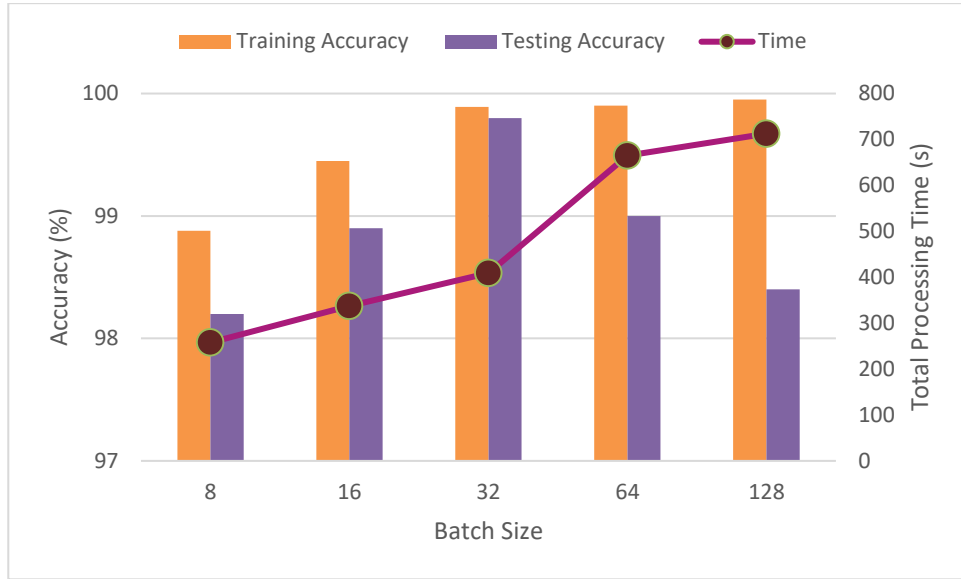
(a)



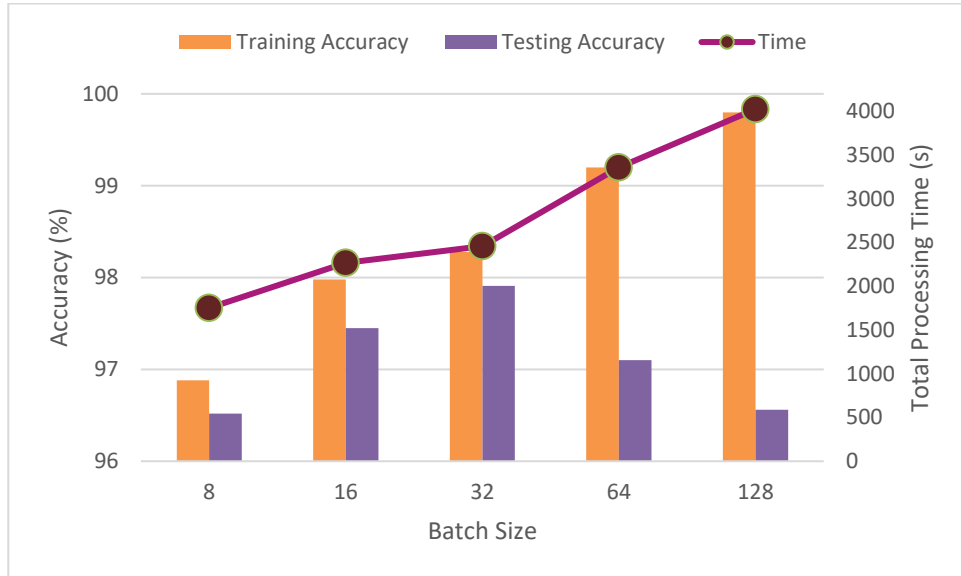
(b)

Figure 4.3: Training and test accuracy on various learning rate, α for (a) RSI-CB256 dataset (b) EuroSAT dataset.

The effect of batch size on training and test accuracy, as well as processing time, is represented in Figures 4.4(a) and 4.4(b). Smaller batches make more frequent weight updates possible, which hastens convergence and enhances generalization. Larger batches, on the other hand, provide a more precise gradient estimate but can impede convergence and lead to overfitting. With larger batch sizes, there is a noticeable rise in overall processing time. Also, it shows better training accuracy but frequently leads to poorer test accuracy, indicating potential overfitting. For the suggested model, an ideal batch size of 32 is chosen in order to attain a balance between processing time and accuracy.



(a)

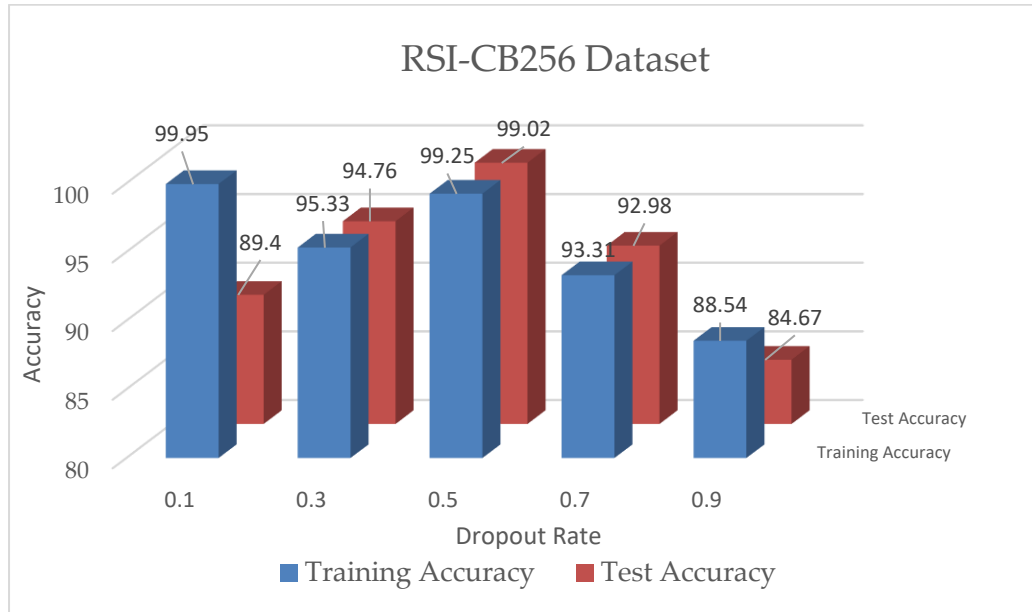


(b)

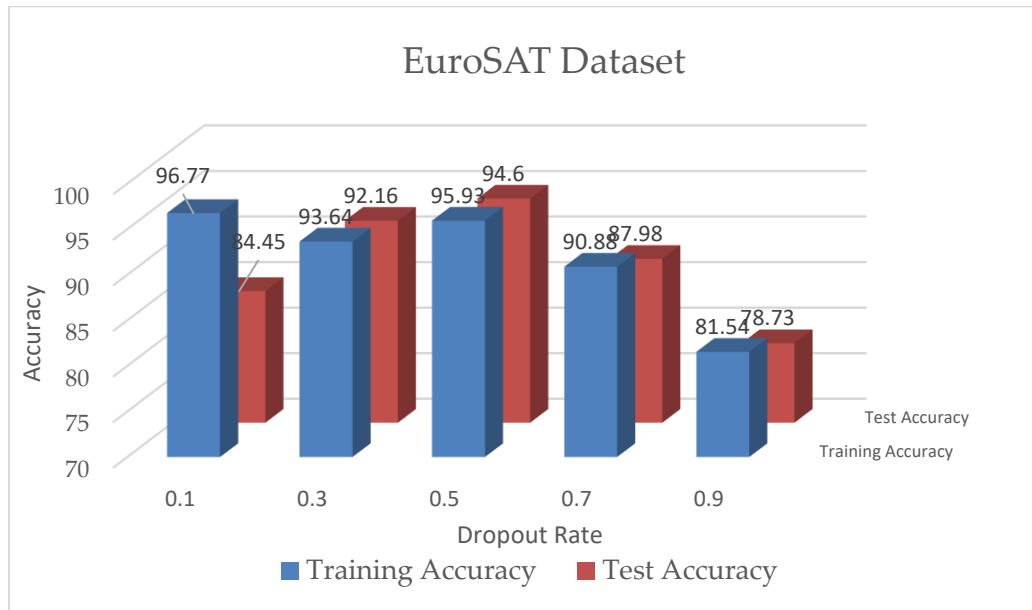
Figure 4.4: Training and test accuracy on various batch size for (a) RSI-CB256 dataset
(b) EuroSAT dataset.

Figures 4.5(a) and 4.5(b) illustrate the effects of learning rate α on training, test accuracy. If the dropout rate is too low, the network may still overfit to the training data. If the dropout rate is too high (close to 1), the network might struggle to learn meaningful representations. High dropout rates can lead to underfitting. Dropout rate is in the range of 0.2 to 0.5, strikes a balance between

introducing regularization to prevent overfitting and allowing the network to learn useful representations



(a)



(b)

Figure 4.5: Training and test accuracy on various dropout rate, α for (a) RSI-CB256 dataset (b) EuroSAT dataset.

The hyperparameters of the OPCNet model have been carefully adjusted to optimize its performance. The specific values for these hyperparameters can be found in Table 4.7.

Table 4.7: Hyperparameters of the proposed OPCNet model.

Parameter	Value
Mini Batch Size	32
Initial learning rate	0.001
Optimization	sgdm
Max Epoch	50

4.6 COMPARISON WITH RELATED WORKS

Efficiency in the suggested classification approach is demonstrated through comparison with state-of-the-art techniques. In order to classify high-resolution satellite images, Yamashkin et al. [49] developed the GeoSystemNet model, which demonstrated an impressive 95.30% success rate on the EuroSAT dataset. M. Ye et al. [56] employed a VGG-16-based transfer learning model for satellite image categorization on a similar dataset, with notable results. However, because of too many layers and parameters in their models, both techniques suffer from overfitting. The RSI-CB256 dataset has gained significant popularity for satellite imagery classification in recent years. Researchers such as Atik [40] and Bihari [57] utilized this dataset, employing Parallel CNN and other Deep CNN models to categorize satellite images and achieve noteworthy results. The comparison with various methodologies in Table 4.8 demonstrates the proposed technique's improved classification accuracy. This not only confirms its effectiveness but also mitigates the drawbacks associated with traditional CNN models.

Table 4.8: Comparison with related works in Satellite Imagery Classification

Reference	Model Name	Year	Dataset	Accuracy
Yamashkin et al. [49]	GeosystemNet	2020	EuroSAT	95.30%
S. Akshay et al. [13]	CNN	2020	EuroSAT	89.00%
M. Ye et al. [56]	VGG-16	2021	EuroSAT	95%
Proposed model	Optimized Parallel CNN- SVM	2024	EuroSAT	97.91%
Atik [40]	Parallel CNN	2023	RSI-CB256	97.84%
S. Tehsin et al. [12]	EfficientNet B7	2023	RSI-CB256	99.64%
Bihari et al. [57]	Deep CNN	2023	RSI-CB256	98.96%
Proposed model	Optimized Parallel CNN- SVM	2024	RSI-CB256	99.8%

Chapter 5: CONCLUSIONS

This chapter explores the conclusion and suggests directions for further research. Section 5.1 provides an overview of the thesis and interprets the results leading to a general conclusion. Section 5.2 highlights the difficulties associated with the development of a method for classifying satellite images. Section 5.3 provides suggestions for further research in this field.

5.1 CONCLUSION

This paper introduces a novel approach employing a Optimized Parallel CNN (OPCNet) architecture with an SVM classifier to tackle challenges in satellite image classification, particularly addressing overfitting and enabling the simultaneous extraction of fine-grained details and broader patterns. In the proposed framework, the OPCNet algorithm acts as an effective feature extractor, accompanied by the SVM classifier for those feature categorization. It provides a significant advantage in the classification of satellite images. The OPCNET model's unique strengths, along with SVM's generalization capabilities, have a synergistic effect that significantly improves the accuracy and efficacy of satellite image classification tasks.

Satellite images have a wide variety of resolution features, which are important for precise classification. Traditional deep CNN models, on the contrary, struggle to extract these fine and broader details simultaneously. To resolve this issue, our proposed lightweight parallel CNN model is effectively designed to capture and integrate features at a variety of resolutions, from low to high, into a unified single feature map across all network branches. This novel approach allows the model to fully capture the fine details and broader patterns present in satellite images. This method has resulted in much better accuracy. The model gets a high accuracy rate of 97.91% on the EuroSAT

database, demonstrating its ability to distinguish between various satellite scenes. Furthermore, it achieves an amazing 99.8% accuracy rate on the RSI-CB256 database, demonstrating its strong performance across a wide range of satellite imagery. Its capacity to integrate low- to high-resolution characteristics demonstrates its ability to capture complex information, highlighting its potential for future advances in satellite image classification.

The suggested model's lightweight architecture provides a significant advantage, establishing it as a distinct model in the field of satellite image classification. The Optimized Parallel CNN (OPCNet) model is highly efficient, with only four layers and fewer than one million parameters. This parameter and layer count is significantly reduced as compared to current state-of-the-art models. The model's lightweight architecture results in faster processing and training times, representing a major improvement in computational efficiency. This reduced design not only helps with resource optimization, but it also emphasizes the model's practical feasibility, making it an appealing option for situations where computing efficiency is critical.

A significant problem often observed in classic CNNs models is overfitting, which occurs due to the enormous number of parameters to be trained in deep learning models. Overfitting occurs when a model performs extraordinarily well on training data but fails to generalize effectively to unknown data, hence affecting overall performance. Our suggested approach strategically addresses the overfitting problem by creating a framework with fewer layers and a significantly lower parameter count. This approach shows a significant resistance to overfitting, demonstrating the model's improved capacity to generalize beyond the training dataset. Furthermore, the use of a dilation factor aids in increasing the network's receptive field without introducing additional parameters. Furthermore, the inclusion of a dropout mechanism plays an important role in addressing overfitting concerns. As a whole, these factors

contribute to the robustness of the proposed technique, demonstrating its success in overcoming the overfitting issues inherent to classic CNNs models.

5.2 LIMITATION AND CHALLENGES OF THE STUDY

Deep learning methods with parallel structures have been widely employed for satellite image classification due to their significant advantages. However, these algorithms have some limitations and obstacles for classifying satellite images, as described below:

- MATLAB does not accept TIFF format images by default, limiting program compatibility. Therefore, it was necessary to convert photos from TIFF to JPG format.
- Training deep learning models is a time-consuming and expensive procedure. The most advanced models need weeks of training on high-performance GPU workstations. In addition, training the model requires a significant amount of memory.
- The model's performance gets influenced by the different resolutions of satellite imagery. Lower resolution images (EuroSAT dataset) may lack fine features, influencing the classification performance of satellite images.

5.3 RECOMMENDATION FOR FURTHER STUDY

- This research can be expanded to more datasets and real-world scenarios, contributing to a better understanding of the performance of satellite image classification algorithms.
- This research can be extended to investigate advanced deep learning architectures such as transformer models and attention mechanisms, which can improve feature extraction and pattern recognition in satellite images for better image classification.

- A potential way of future research for this research would be to concentrate on the identification of specific objects in satellite images. The current thesis focused on broader aspects of satellite image analysis such as classification. Addressing specific object identification could have been beyond the scope of the current research. However, as a potential avenue for future research, the identification of specific objects in satellite images presents an intriguing direction. To address this, efforts could involve gathering datasets tailored to the objects of interest, developing specialized algorithms, and integrating multimodal data sources. By systematically addressing these aspects, future research can advance the field's understanding of object identification in satellite imagery.

References

- [1] S. Voigt et al., "Satellite image analysis for disaster and crisis-management support," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, pp. 1520–1528, 2007.
- [2] T. W. Timberlynn, "Deep convolutional neural networks for remote sensing investigation of looting of the archaeological site of Al-Lisht, Egypt," PhD dissertation, University of Southern California, 2018.
- [3] R. Padmanaban et al., "Satellite image fusion to detect changing surface permeability and emerging urban heat islands in a fast-growing city," *PLoS ONE*, vol. 14, p. e0208949, 2019.
- [4] V. Khryashev and L. Ivanovsky, "Urban areas analysis using satellite image segmentation and deep neural network," in *E3S Web of Conferences*, Les Ulis, France, 2019, vol. 135, p. 01064.
- [5] S. Tripodi et al., "Brightearth: Pipeline for on-the-fly 3D reconstruction of urban and rural scenes from one satellite image," *ISPRS Annals of the Photogrammetry. Remote Sens. Spat. Inf. Sci.*, vol. 3, pp. 263–270, 2022.
- [6] G. Soldi et al., "Space-based global maritime surveillance," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 36, pp. 8–28, 2021.
- [7] S. R. Phinn et al., "Assessing the potential for satellite image monitoring of seagrass thermal dynamics: For inter-and shallow sub-tidal seagrasses in the inshore great barrier reef world heritage area, Australia," *Int. J. Digit. Earth*, vol. 11, pp. 803–824, 2018.
- [8] R. H. Dehkordi et al., "High spatio-temporal monitoring of century-old biochar effects on evapotranspiration through the etlook model: A case study with UAV and satellite image fusion based on additive wavelet transform," *GIScience Remote Sens.*, vol. 59, pp. 111–141, 2022.
- [9] K. Sasaki et al., "Coastal marine debris detection and density mapping with very high-resolution satellite imagery," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 6391–6401, 2022.
- [10] M. K. Pichugin et al., "Severe marine weather systems during freeze-up in the Chukchi Sea: Cold-air outbreak and mesocyclone case studies from satellite multisensor measurements and reanalysis datasets," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 12, pp. 3208–3218, 2019.

- [11] A. T. Kokal et al., "Detection of mucilage phenomenon in the Sea of Marmara by using multi-scale satellite data," *Environ. Monit. Assess.*, vol. 194, p. 585, 2022.
- [12] S. Tehsin et al., "Satellite Image Categorization Using Scalable Deep Learning," *Applied Sciences*, vol. 13, no. 8, p. 5108, Apr. 2023, doi: 10.3390/app13085108.
- [13] S. Akshay et al., "Satellite Image Classification for Detecting Unused Landscape using CNN," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 215-222.
- [14] C. Zhang et al., "A multi-level context-guided classification method with object-based convolutional neural network for land cover classification using very high-resolution remote sensing images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 88, 2020, 102086.
- [15] D. Konstantinidis et al., "A modular CNN-based building detector for remote sensing images," *Computer Networks*, vol. 168, 2020.
- [16] F. Bastani et al., "RoadTracer: Automatic Extraction of Road Networks from Aerial Images," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4720–28.
- [17] M.A. Kadhim and M.H. Abed, "Convolutional Neural Network for Satellite Image Classification," in *Intelligent Information and Database Systems: Recent Developments*, vol. 830, p. 165, 2019.
- [18] J. Jiang et al., "Multi-spectral RGB-NIR image classification using double-channel CNN," *IEEE Access*, vol. 7, pp. 20607–20613, 2019.
- [19] S. Deepak et al., "Brain tumor classification using deep CNN features via transfer learning," *Comput. Biol. Med.*, vol. 111, p. 103345, 2019.
- [20] Z.N.K. Swati et al., "Brain tumor classification for MR images using transfer learning and fine-tuning," *Comput. Med. Imaging Graph.*, vol. 75, pp. 34–46, 2019.
- [21] M. Talo et al., "Application of deep transfer learning for automated brain abnormality classification using MR images," *Cogn. Syst. Res.*, vol. 54, pp. 176–188, 2019.
- [22] Z. Li, Q. Du, S. Wang, and Y. Hu, "The Applications of Satellite Remote Sensing in the Monitoring of Global Lands," *Remote Sensing*, vol. 12, no. 11, p. 1808, 2020.
- [23] M. A. Wulder, N. C. Coops, and D. P. Roy, "Satellites: Make Earth Observations Open Access," *Nature*, vol. 566, no. 7742, pp. 315-317, 2019.
- [24] <https://www.sporcle.com/games/palmtree/arial-views-of-us-cities> [Accessed: 20-January-2024].

- [25] <https://aws.amazon.com/what-is/neural-network> [Accessed: 20-January-2024].
- [26] Structure of Artificial Neural Network, <https://data-flair.training/blogs/artificial-neural-network> [Accessed: 20-January-2024].
- [27] Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* 2015, 521, 436–444.
- [28] <https://www.upgrad.com/blog/basic-cnn-architecture> [Accessed: 13- January-2024]
- [29] F. Diker and I. Erkan, "Classification of Satellite Images with Deep Convolutional Neural Networks and Its Effect on Architecture," *Eskişehir Technical University Journal of Science and Technology A - Applied Sciences and Engineering*, vol. 23, pp. 31-41, 2022.
- [30] Schmidhuber, J. "Deep learning in neural networks: An overview", *Neural Netw.* 2015, 61, 85–117.
- [31] Z. H. Jarrallah and M. A. A. Khodher, "Satellite Images Classification Using CNN :A Survey," 2022 International Conference on Data Science and Intelligent Computing (ICDSIC), Karbala, Iraq, 2022, pp. 111-116.
- [32] B. Rekabdar and C. Mousas, "Dilated Convolutional Neural Network for Predicting Driver's Activity," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 2018, pp. 3245-3250
- [33] <https://www.linkedin.com/advice/0/what-some-most-popular-widely-used-pretrained>. [Accessed: 15-January-2024]
- [34] <https://neurohive.io/en/popular-networks/vgg16> [Accessed: 15-January-2024]
- [35] Han, X.; Zhong, Y.; Cao, L.; Zhang, L. Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision for High Spatial Resolution Remote Sensing Image Scene Classification. *Remote Sens.* 2017, 9, 848.
- [36] <https://c3.ai/glossary/data-science/classifier> [Accessed: 15-July-2023]
- [37] Muhammad Asad Iqbal Khan, "Introduction to Softmax Classifier in PyTorch", *Deep Learning with PyTorch*, [Accessed: 05-December-2023]
- [38] <https://www.javatpoint.com/machine-learning-support-vector-machinealgorithm> [Accessed: 13-August-2023]
- [39] P. Shelke, A. Gole, P. Kanherkar, S. Singh, and P. Abhishek, "Comparative Analysis of Machine Learning Algorithms: Random Forest Algorithm, Naive Bayes Classifier, and KNN - A Survey," 2023.
- [40] I. Atik, "Parallel Convolutional Neural Networks and Transfer Learning for

Classifying Landforms in Satellite Images," *Information Technology and Control*, vol. 52, no. 1, pp. 228-244, 2023.

- [41] Y. Yu and F. Liu, "Dense connectivity based two-stream deep feature fusion framework for aerial scene classification," Available: www.mdpi.com/journal/remotesensing, 2018.
- [42] T. Ojala et al., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 971–987, 2002.
- [43] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, 2015.
- [44] A. Mehran, S. Tehsin, and M. Hamza, "An effective deep learning model for ship detection from satellite images," *Spat. Inf. Res.*, vol. 31, pp. 61–72, 2023.
- [45] Y. Li, Y. Zhang, and Z. Zhu, "Learning deep networks under noisy labels for remote sensing image scene classification," in *Proceedings of the IGARSS, Yokohama, Japan, 28 July–2 August 2019*, pp. 3025–3028.
- [46] R. S. Gargees and G. J. Scott, "Large-scale, multiple level-of-detail change detection from remote sensing imagery using deep visual feature clustering," *Remote Sens.*, vol. 13, p. 1661, 2021.
- [47] H. Alhichri et al., "Classification of remote sensing images using EfficientNet-B3 CNN model with attention," *IEEE Access*, vol. 9, pp. 14078–94, 2021.
- [48] V. Syrris, O. Pesek, and P. Soille, "SatImNet: Structured and Harmonised Training Data for Enhanced Satellite Imagery Classification," *Remote Sens.*, vol. 12, pp. 3358, 2020.
- [49] S. A. Yamashkin et al., "Improving the Efficiency of Deep Learning Methods in Remote Sensing Data Analysis: Geosystem Approach," *IEEE Access*, vol. 8, pp. 179516-179529, 2020.
- [50] N. Yang et al., "Accelerating the Training Process of Convolutional Neural Networks for Image Classification by Dropping Training Samples Out," *IEEE Access*, vol. 8, pp. 142393–142403, 2020.
- [51] C. Liu et al., "Urban Land Cover Classification of High-Resolution Aerial Imagery Using a Relation-Enhanced Multiscale Convolutional Network," *Remote Sensing*, vol. 12, no. 2, p. 311, 2020.
- [52] Available online: <https://www.kaggle.com/datasets/mahmoudreda55/satellite->

image-classification [accessed on 20 October 2023].

- [53] Available online: <https://www.kaggle.com/datasets/apollo2506/eurosat-dataset> [accessed on 11 June 2023].
- [54] E. Charou et al., "Deep Learning for Agricultural Land Detection in Insular Areas," in 10th International Conference on Information, Intelligence, Systems and Applications (IISA), 2019, pp. 1-4.
- [55] P. Helber et al., "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2217-2226, July 2019.
- [56] M. Ye et al., "A Lightweight Model of VGG-16 for Remote Sensing Image Classification," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 6916-6922, 2021.
- [57] B. N. Pandey et al., "Deep learning-based ensemble model for satellite image classification," Authorea, May 24, 2023.