Towards Automatic Weather Classification Using DCNNs

By

Mattia Tun Nabi

(Registration No: 00000402019)

Department of Robotics & Artificial Intelligence

School of Mechanical & Manufacturing Engineering

National University of Sciences & Technology (NUST)

Islamabad, Pakistan

(2024)

Towards Automatic Weather Classification Using DCNNs

By

Mattia Tun Nabi

(Registration No: 00000402019)

A thesis submitted to the National University of Sciences and Technology, Islamabad,

in partial fulfillment of the requirements for the degree of

Master of Science in
Robotics & Intelligent Machine Engineering

Supervisor: Dr. Sara Baber

Co Supervisor: Dr. Kashif Javed

School of Mechanical & Manufacturing Engineering

National University of Sciences & Technology (NUST)

Islamabad, Pakistan

(2024)

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by **Regn No. 00000402019 Mattia Tun Nabi** of **School of Mechanical & Manufacturing Engineering (SMME)** has been vetted by undersigned, found complete in all respects as per NUST Statues/Regulations, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis titled. **Towards Automatic Weather Classification using DCNNs**

Signature:

Name (Supervisor): Sara Baber Sial

Date: 30 - Jul - 2024

Signature (HOD):

Date: 30 - Jul - 2024

Signature (DEAN):

Date: 30 - Jul - 2024

Form TH-4

National University of Sciences & Technology (NUST)

MASTER'S THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: Mattia Tun Nabi (00000402019) Titled: Towards Automatic Weather Classification using DCNNs be accepted in partial fulfillment of the requirements for the award of MS in Robotics & Intelligent Machine Engineering degree.

Examination Committee Members

1. Name: Kashif Javed Signature: Kashif

2. Name: Muhammad Nabeel Anwar Signature:

3. Name: Khawaja Fahad Iqbal Signature:

4. Name: Sara Baber Sial Signature:

Supervisor: Sara Baber Sial Signature:

Date: 30 - Jul - 2024

Head of Department

30 - Jul - 2024

Date

COUNTERSINGED

30 - Jul - 2024

Date

Dean/Principal

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in this thesis, entitled "Towards Automatic Weather Classification using DCNNs" was conducted by Ms. Mattia Tun Nabi under the supervision of Dr. Sara Ali. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the Department of Robotics and Artificial Intelligence in partial fulfillment of the requirements for the degree of Master of Science in Field of Robotics and Intelligent Machine Engineering.

Department of Robotics and Artificial Intelligence, National University of Sciences and Technology, Islamabad.

Student Name: Mattia Tun Nabi   Signature: ____________

Examination Committee:

a) External Examiner 1: Name   Signature: N/A
(Designation & Office Address)

Signature: N/A

b) External Examiner 2: Name
(Designation & Office Address)

Supervisor Name: Dr. Sara Ali

Signature: ____________

Name of Dean/HOD: Dr. Kunwar Faraz
Ahmed Khan   Signature: ____________

AUTHOR'S DECLARATION

I Mattia Tun Nabi hereby state that my MS thesis titled "Towards Automatic Weather Classification Using DCNNs" is my own work and has not been submitted previously by me for taking any degree from National University of Sciences and Technology, Islamabad or anywhere else in the country/ world.

At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.

Name of Student: Mattia Tun Nabi

Date: 30th July, 2024.

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled "Towards Automatic Weather Classification using DCNNs" is solely my research work with no significant contribution from any other person. Small contribution/ help wherevertaken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of the HEC and National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the University reserves the rights to withdraw/revoke my MS degree and that HEC and NUST, Islamabad has the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Student Signature:

Name: Mattia Tun Nabi

Proposed Certificate for Plagiarism

It is certified that PhD/MPhil/MS Thesis Titled "**Towards Automatic Weather Classification using DCNNs**" by **Ms. Mattia Tun Nabi** has been examined by us. We undertake the follows:

a. Thesis has significant new work/knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.

b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.

c. There is no fabrication of data or results which have been compiled/analyzed.

d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.

e. The thesis has been checked using TURNITIN (copy of originality report attached) and found within limits as per HEC plagiarism Policy and instructions issued from time to time.

Name & Signature of Supervisor

Dr Sara Ali

Signature :

DEDICATION

With boundless love and eternal gratitude, I dedicate this thesis to the two remarkable souls whose unwavering support has been the catalyst for every endeavor — my parents.

This achievement is yours as much as it is mine

ACKNOWLEDGEMENTS

I extend my heartfelt gratitude to my peers, Saad Ali Babar for his invaluable assistance during the preliminary phase of the research. The process would have been quite daunting without his support.

I also express my gratitude to Sabih ul Hassan for his valuable insights; his willingness to contribute his time and effort is greatly appreciated.

Additionally, I express my sincere thanks to Dr. Ihsan Ullah from CUI, Abbottabad, for his guidance and support throughout this research. This milestone is as much yours as it is mine.

TABLE OF CONTENTS

LIST OF TABLES

Page No.

LIST OF FIGURES

Page No.

LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

CNN	Convolutional Neural Networks
LSCIDMR	Large Scale Cloud Images Dataset for Meteorology Research
RSIC	Remote Sensing Image Classification
AI	Artificial Intelligence
ResNet	Residual Network
NWP	Numerical Weather Prediction
RS	Remote Sensing
ML/DL	Machine Learning/Deep Learning
SIFT	Scale Invariant Feature Transform
HOG	Histogram of Gradients
SGD	Stochastic Gradient Descent
FCL	Fully Connected Layers
GAP	Global Average Pooling
OA	Overall Accuracy
GPU	Graphic Processing Unit
RE	Random Erasing

ABSTRACT

The utilization of remote sensing (RS) technology has resulted in the extensive accessibility of a significant amount of satellite image data. In order to ensure the successful execution of the RS in real-life scenarios, it is imperative to create effective and adaptable solutions that can be utilized across different transdisciplinary domains. Deep Convolutional Neural Networks (CNNs) are frequently used to accomplish the goal of fast analysis and precise categorization in RS imaging. This study introduces a unique residual network known as ResNet101. The network comprises FC-1024 fully connected layers, dropout layers, a thick layer, and data augmentation algorithms. To resolve the issue of similarity between different classes, architectural enhancements are implemented. On the other hand, imbalanced classes are dealt with by employing data augmentation techniques. The ResNet101 model use the rigorous Large-Scale Cloud pictures Dataset for Meteorology Research (LSCIDMR), which has 10 classes and a multitude of high-resolution photos. The goal of the model is to precisely classify these photos into their respective categories. The model we have created outperforms numerous previously published deep learning algorithms in terms of Precision, Accuracy, and F1 scores. The accuracy reaches up to 99% and approximately 92%, respectively.

Keywords: Deep Learning, Image Classification, Satellite Imagery, Weather Forecasting, Convolutional Neural Networks.

CHAPTER 1: INTRODUCTION

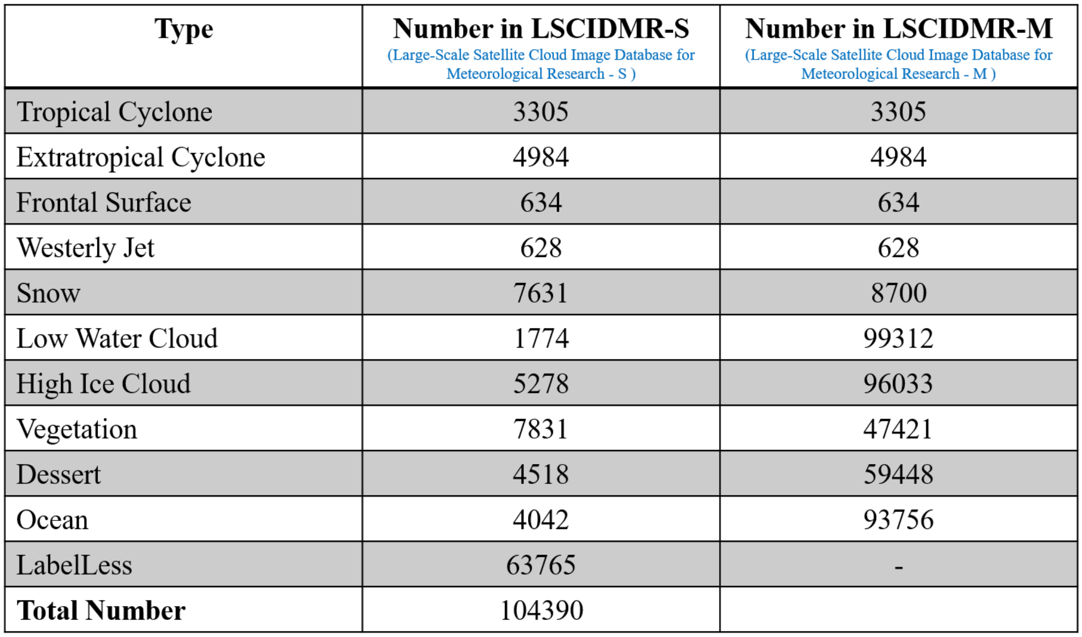
Accurate weather forecasts are crucial for individuals' everyday routines. Tracking the weather can be a challenging task due to its intricate nature, comprising numerous features and variables. The task of precise prediction is unique. Over time, there has been substantial progress in the creation of theories, observation systems, and prediction tools. Weather forecasting requires the collection of a diverse array of meteorological data from many sources such as satellites, observation systems, automated and human stations, aircraft, ships, weather balloons, and other methods. Research indicates that forecast models can be classified into two main categories: global models, which encompass the entire planet, and local models, which concentrate on specific regions such as continents, nations, or mountain ranges. Now it is effortless to generate weather forecasts that are superior and highly precise. The weather is intricate and fluctuating. These alterations might have various impacts on individuals. The prediction of typhoons and other forms of extreme weather remains challenging and is being considered. Currently, there exist numerous forecast models, each specifically tailored to serve a certain purpose.

Simultaneously, numerous research communities have shown a growing interest in the rapid advancement of artificial intelligence, including deep learning. Deep learning can be used more effectively in weather forecasting by learning from how it has been used successfully in computer vision tasks like face recognition, visual extraction, segmentation by semantics, and prominence identification. Additionally, deep learning has also proven to be effective in remote sensing image processing, specifically in tasks like image classification, scene classification, and image prediction.

In order to implement deep learning for image classification in practical applications, a substantial quantity of photographs is required to ensure the system is effective and can be expanded as needed. Satellite images must be quickly analyzed and correctly categorized in many areas, such as maritime, agriculture, the business sector, the military, air traffic, severe weather alerts, and architectural design. So, most of the study in satellite image analysis has been on algorithms for categorizing satellite images, and these performance indicators are used to judge how well they work.

Currently, there is no existing cloud pictures database of that nature. It is recommended to utilize a comprehensive cloud picture database on a broad scale for meteorological research (LSCIDMR). From what I know, this is the first public benchmark database for wind and weather studies. It specifically focuses on satellite cloud photos and aims to establish connections between weather systems and these images. Weather satellites collect data about clouds, the atmosphere, and the seas through satellite cloud photographs. Weather satellites play a crucial role in analyzing weather patterns and issuing alerts for meteorological disasters. Their main purpose is to observe and track Earth's weather conditions and climate. The Himawari-8 satellites, which is the inaugural weather satellite with the capability to capture color photographs, observes genuine cloud images. The LSCIDMR dataset contains 104390 high-resolution pictures that represent 11 different classes. The dataset has descriptions of four different systems: 1) Weather system: A weather system is made up of different parts, like a tropical cyclone, an extratropical cyclone, a frontal surface, a westerly jet, and rain. 2) Cloud system: There are ice clouds high in the sky and water clouds low in the sky. 3) Terrestrial system: The ocean, the desert, and plants are all in this group. 4. Label only what you need to. The dataset has both single-label annotation, which is called LSCIDMR-S, and multiple-label annotation, which is called LSCIDMR-M. The overall number of solo labels is 40,625, whereas the number of distinct labels is 414,221. This is due to the meticulous marking of the labels. Details are shown in table 1.1.

Table 1.1: Details of 11 classes of LSCIDMR



Type	Number in LSCIDMR-S (Large-Scale Satellite Cloud Image Database for Meteorological Research - S)	Number in LSCIDMR-M (Large-Scale Satellite Cloud Image Database for Meteorological Research - M)
Tropical Cyclone	3305	3305
Extratropical Cyclone	4984	4984
Frontal Surface	634	634
Westerly Jet	628	628
Snow	7631	8700
Low Water Cloud	1774	99312
High Ice Cloud	5278	96033
Vegetation	7831	47421
Dessert	4518	59448
Ocean	4042	93756
LabelLess	63765	-
Total Number	104390	

I. UC Merced Land-Use Dataset: The raw data of the UC Merced Land-Use dataset comprises RGB images sourced from the United States Geological Survey National Map. These photos were gathered from 20 specific cities in the US, including Birmingham and Boston. The dataset has 21 land-use scenario types that have been annotated with single labels. Each category contains 100 photos, each of which has dimensions of 256x256 pixels. The dataset has been augmented to provide a multilabel annotation from eight years subsequent to its initial creation. The multilabel annotation method assigns labels to each individual component of the picture, while the single-label annotation method only assigns a label to the core material of the image. Figure 1 exhibits three photos from this collection as illustrations. In Fig. 1.1 (a), an aero plane is shown as a single-label annotation. However, its multi-label annotation comprises an aero plane, bare earth, and other items.

II. NWPU-RESISC45 Dataset: After the UC Merced Land-Use dataset was made available, multiple datasets utilizing Google Earth as the data source have been suggested. Wuhan University developed the WHU-RS19 dataset, which is a collection of remote sensing (RS) images categorized into 19 distinct classes. RSSCN7 utilizes RS scene photos to represent seven prevalent scene categories. Northwestern Polytechnical University developed the remote dataset SIRI-WHU to conduct a study on metropolitan regions in China. Meanwhile, Northwestern Polytechnical University created NWPU-

RESISC45 for the classification of remote sensing image scenes. We utilize NWPU-RESISC45 as an illustrative instance. The total number of scene classes in NWPU-RESISC45 is 45. Inside the RGB color system, every category inside this collection consists of 700 photos, each measuring 256x256 pixels. The annotation for each data entry in this database consists of a single label, which is identical to the labels used in the UC Merced Land-Use dataset. Several illustrations are displayed in Figure 1.2.

III. Brazilian Coffee Scene Dataset: The Brazilian Coffee Scene dataset focuses specifically on classifying coffee and non-coffee crop tiles, in contrast to the datasets discussed before. There are just two resulting categories: There are two categories of pixels: coffee pixels, which make up at least 85% of them, and non-coffee pixels, which make up less than 10% of them. The dataset comprises 36,577 photos captured in 2005 by the SPOT Sensor. Each image has a resolution of 64 by 64 pixels and covers four counties in the Brazilian State of Minas Gerais. The dataset only considers the green, red, and near-infrared channels as they are the most informative and indicative for detecting the existence of vegetation. The labeling was conducted by agricultural research professionals. This dataset comprises landscapes exhibiting spectrum distortions resulting from different plant ages and shadows, as coffee is an evergreen plant and the southern area of Minas Gerais has hilly topography.

IV. BigEarthNet Dataset: The BigEarthNet dataset consists of 59,0326 Sentinel-2 picture patches, each with different image sizes depending on the channels. A cloud cover percentage of less than 1% is associated with tiles that are scattered over 10 European countries. The multilabel database consists of photos, of which 95% have no more than five labels. Additionally, there are 43 labels that are not evenly distributed, indicating an imbalance.

V. AID Dataset: The number of images in the aerial image dataset (AID), which is similar in size to the previously stated databases, has grown. There are a total of 10,000 photos, which have been divided into 30 different classifications.

VI. HAINAN Dataset: A comprehensive time-series remote sensing dataset with a specific focus on Hainan Island was recently unveiled. The HAINAN dataset consists of cloud-free photos from the Landsat series satellites, covering the period from 1970 to 2017. HAINAN specializes in the final processing of satellite data, as opposed to the datasets that were previously mentioned. The deep processing service system offers the opportunity to obtain five value-added goods, including projection conversion, picture registration, band fusion, and image mosaics.

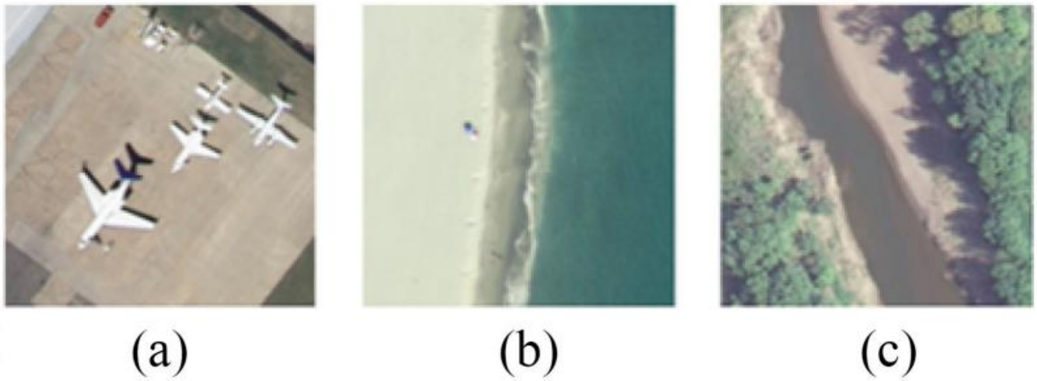


Figure 1.1: Three labeled examples of the UC Merced Land-Use dataset. (a) Airplanes, bare soil, grass, cars. (b) Sand, sea. (c) Water, trees, bare soil.

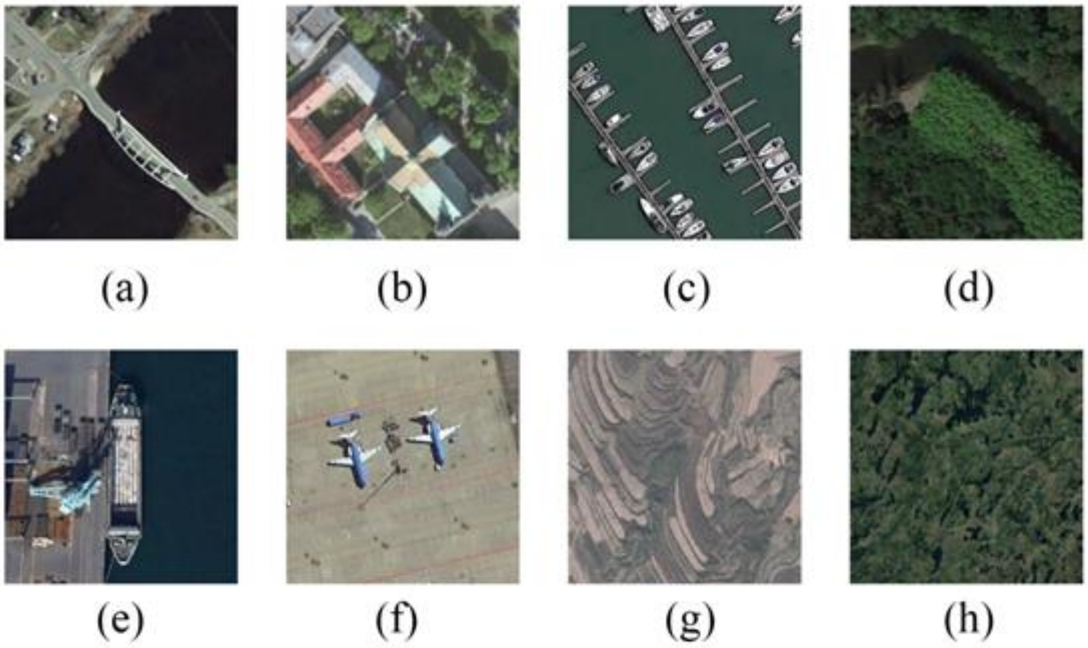


Figure 1.2: Eight scene images from the popular NWPU-RESISC45 dataset.

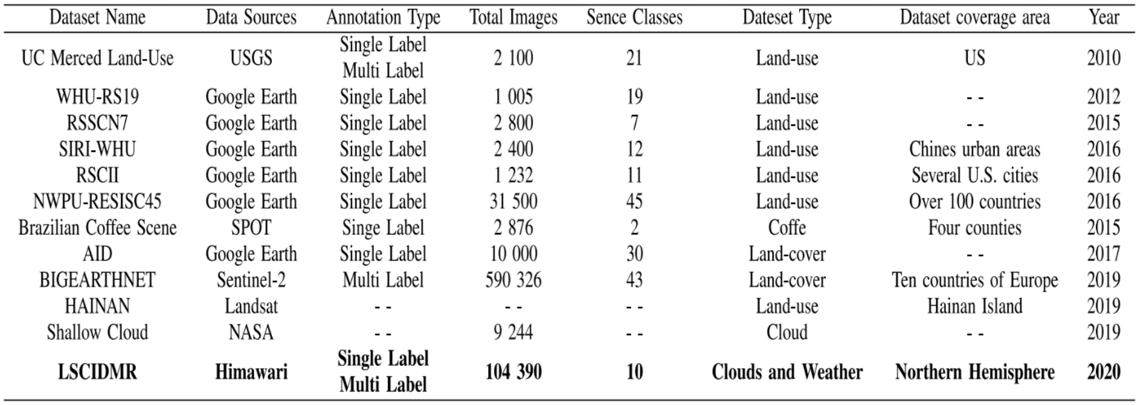
(a) Bridge. (b) Church. (c) Harbor. (d) River. (e) Ship. (f) Airplane. (g) Terrace. (h) Wetland.

Over the past four decades, there has been a decrease in the impact of clouds. Moreover, it can function as a vital asset for informing government macro decision-making, agricultural planning, urban development, environmental conservation, and monitoring industrial applications.

I. *Shallow Cloud:* The dataset named Shallow Cloud, consisting of satellite cloud photos, is accessible on Kaggle. It focuses on analyzing the mesoscale characteristics of shallow clouds in the subtropical regions. The photos in this collection were obtained from NASA Worldview and cover three places situated at latitudes of 21 and 14. Four additional labels have been incorporated into this dataset through the utilization of crowdsourcing and deep learning techniques. These labels include "sugar," "flowers," "fish," and "gravel." Although it consists of satellite cloud photos from the RS, this collection exclusively emphasizes cloud formation and does not establish a clear correlation between the clouds and weather systems.

Current remote sensing databases only contain data from Earth's surface and do not include information from the sky. Additionally, clouds are considered to be undesirable interference in these databases. Nevertheless, clouds exert a substantial influence on the field of meteorology. Shallow Cloud engages in cloud observation but does not instantly correlate the cloud images with meteorological trends. Table 1.2 presents a concise overview of the characteristics of both the individuals and LSCIDMR.

Table 1.2: Comparison between the proposed LSCIDMR and other publicly available satellite image databases



Dataset Name	Data Sources	Annotation Type	Total Images	Sence Classes	Dateset Type	Dataset coverage area	Year
UC Merced Land-Use	USGS	Single Label Multi Label	2 100	21	Land-use	US	2010
WHU-RS19	Google Earth	Single Label	1 005	19	Land-use	- -	2012
RSSCN7	Google Earth	Single Label	2 800	7	Land-use	- -	2015
SIRI-WHU	Google Earth	Single Label	2 400	12	Land-use	Chines urban areas	2016
RSCII	Google Earth	Single Label	1 232	11	Land-use	Several U.S. cities	2016
NWPU-RESISC45	Google Earth	Single Label	31 500	45	Land-use	Over 100 countries	2016
Brazilian Coffee Scene	SPOT	Singe Label	2 876	2	Coffe	Four counties	2015
AID	Google Earth	Single Label	10 000	30	Land-cover	- -	2017
BIGEARTHNET	Sentinel-2	Multi Label	590 326	43	Land-cover	Ten countries of Europe	2019
HAINAN	Landsat	- -	- -	- -	Land-use	Hainan Island	2019
Shallow Cloud	NASA	- -	9 244	- -	Cloud	- -	2019
LSCIDMR	**Himawari**	**Single Label** **Multi Label**	**104 390**	**10**	**Clouds and Weather**	**Northern Hemisphere**	**2020**

The table presents a thorough summary of different datasets utilized for the analysis of land-use, land-cover, coffee sceneries, and clouds and weather. Each row in the table corresponds to a distinct dataset, and several attributes of these datasets are presented in various columns.

The initial column, labeled "Dataset Name," enumerates the titles of the datasets, denoting either the particular emphasis or the organization responsible for curating the dataset. For example, the dataset called "UC Merced Land-Use" is named after UC Merced, whereas the dataset titled "Brazilian Coffee Scene" primarily concentrates on coffee farms.

The second column, labeled as "Data Sources," indicates the source or origin of the data. The sources mentioned encompass well-known platforms such as Google Earth, USGS (United States Geological Survey), NASA, SPOT, Sentinel-2, and Himawari. The variety of data sources is a result of the many procedures and technology employed to gather the data, which might include satellite photography as well as targeted field surveys.

The "Annotation Type" field specifies whether the data is annotated with single labels or multiple labels. This classification aids in comprehending the intricacy and level of detail of the dataset annotations, which is essential for diverse applications in machine learning and data analysis.

The "Total Images" column measures the magnitude of each dataset by indicating the quantity of images it contains. The dataset sizes vary greatly, with smaller datasets like "WHU-RS19" including 1,005 photos, while the bigger "BIGEARTHNET" dataset consists of 590,326 images. The dataset's size frequently corresponds to its usability in developing resilient machine learning models.

"Scene Classes" refers to the number of distinct classes or categories included in the dataset. As an illustration, the "AID" dataset consists of 30 different scene classes, while the "NWPU-RESISC45" dataset contains 45 classes. The presence of a wide range of classes is crucial for carrying out intricate classification operations.

The term "Dataset Type" refers to the main area of interest or subject matter that the dataset primarily focuses on. The majority of datasets are classified as either "Land-use" or "Land-cover," indicating their purpose in studying the utilization or coverage of land. Distinct categories such as "Coffee" and "Clouds and Weather" exhibit specialized datasets tailored for certain purposes, such as agricultural studies and weather forecasting.

The "Dataset coverage area" column offers geographical context, specifying the location where the data was gathered. Certain datasets exhibit extensive coverage, such as the "NWPU-RESISC45" dataset, which encompasses more than 100 countries. In contrast, other datasets are more limited in scope, such the "Brazilian Coffee Scene" dataset, which only covers four counties.

Finally, the "Year" column denotes the publication date or the most recent update of the dataset. This analysis offers valuable information regarding the dataset's pertinence and the timeliness of the data. The datasets vary in age, with the oldest being "UC Merced Land-Use" from 2010 and the most recent being "LSCIDMR" from 2020.

Essentially, this table provides a comprehensive inventory of diverse datasets utilized for the examination of land-use, land-cover, and specific scene assessments. The document contains essential details regarding the data sources, types of annotations, dataset size, variety of scene classes, dataset types, geographical coverage, and publication year. These details are of utmost importance for researchers and practitioners working in the field of remote sensing and geographic information systems (GIS).

The LSCIDMR dataset presents challenges in its use due to the extensive range of classes it offers, however it is substantial in size due to the abundance of images it encompasses.

Regrettably, the utilization of deep convolutional neural network (CNN) structures for remote sensing image classification is currently limited due to the frequent occurrence of disparities in class composition and inter class overlap in raw data, which poses challenges for deeper networks. This occurs because to a class imbalance in the dataset, where the model learns efficiently for classes that include a large number of photos but performs poorly for classes with a smaller number of images. Consequently, our model may exhibit a bias towards specific classes and produce inaccurate predictions. A model with a deep architecture can effectively address the difficulty of inter-class similarity. The researchers have been unable to fully achieve the potential of deep networks for categorizing remote sensing images due to this dilemma.

1.1 Problem Statement

The recent publication of the LSCIDMR dataset has demonstrated the significant capabilities of deep learning architectures in the field of MR. This dataset is particularly hard due to its high inter-class resemblance and disparities in class composition. The primary obstacle to utilizing deep CNN architectures for the LSCIDMR dataset is the inter-class phenomena caused by the similarity between the various classes in the dataset. Therefore, in order to fully use the capabilities of deeper networks for the LSCIDMR dataset, it is essential to employ appropriate strategies or learning procedures to address the inter-class issue. Furthermore, the aim is to enhance the accuracy of image classification for satellite photos by investigating the maximum capabilities of deep CNN architectures combined with network assembling methodologies.

ResNet is a highly popular deep Convolutional Neural Network (CNN) architecture that surpasses its competitors in terms of performance. It is characterized by its considerably greater depth compared to other networks. By operating at a deeper level, the system is able to extract and learn more intricate characteristics. This allows it to function effectively, even when faced with significant similarities across different classes. To effectively categorize the LSCIDMR database employing neural network topologies like ResNets, it is necessary to thoroughly examine an optimization/training mechanism. The degree to which it surpasses the mentioned constraints should be documented.

1.2 Motivation

Accurate weather forecasting has been of utmost importance for those who are committed to ensuring the safety of aviation and human beings. As an avionics engineer, I am deeply passionate about the scientific disciplines related to aviation. The weather presents a hazard for all groups focused on aircraft safety. The aviation sector aims to develop efficient technologies for accurate forecasting, as the problem remains unresolved. Among various factors, adverse weather conditions provide the most significant threat to both aircraft and passengers, as they have the potential to cause injury to both the aircraft itself and its occupants. During my literature review, I found that although there has been notable advancement in predicting, there are still unresolved challenges, such as interclass similarities and limitations in spatial and temporal precision. Both the dataset and a published study on weather forecasting were accessible. In recently published research, the focus is on interclass differences rather than similarities, regardless of whether the accuracy of weather prediction is 90% lower or higher. This motivates me to address the problem of categorization. The primary hindrance to achieving satisfactory accuracy in weather forecasting is the absence of an appropriate algorithm capable of accommodating the diverse lighting and weather conditions.

1.3 Objectives

Employ state-of-the-art AI methods in the domain of computer vision to accurately classify cloud photos obtained from satellites. The primary goal is to develop a deep learning framework capable of accurately classifying intricate scenarios from classes within the dataset of satellite imagery, which are semantically associated. The objective is to examine different training and optimization techniques in order to achieve the highest classification accuracy. This includes exploring transfer learning and implementing countermeasures for inter-class phenomena, particularly when working tackling uneven data for training. The analysis sector now regards the accurate categorization of circumstances that share significant characteristics across several classes as a difficulty. By harnessing the full capability of distinct deep CNN models and combining them via assembling techniques, we anticipate that the network's ability to distinguish between different classes will be enhanced.

1.4 Scope of the Thesis

The scope of the thesis can be summarized under following headers:

- The primary objective of the project is to enhance the weather forecasting system.
- Comprehension of Convolutional Neural Network (CNN) pipeline.
- Comprehension of techniques used in weather forecasting systems
- Developing a weather forecasting system utilizing several architectural approaches.
- Our objective is to train the best appropriate algorithm to identify subtle variations in photos within the given dataset. Additionally, we aim to design our own method that enhances the overall accuracy of image classification by minimizing similarities between satellite images from different classes, with a specific focus on weather forecasting.
- Various data augmentation strategies are employed to address the issue of class imbalance.
- The goal is to integrate an ensembling method that incorporates optimized ResNet topologies in order to surpass the performance of individual models.
- Assessment of the computational expense incurred during various stages of implementation.
- The performance of the implemented architecture will be compared to current work in the literature using conventional evaluation measures, including overall accuracy, precision, f1-score, and statistics provided by confusion matrices.

1.5 Thesis Layout

The thesis is structured as follows: Chapter 1 provides an introduction to satellite image classification. Chapter 2 focuses on the primary obstacles in the field of MR and then examines relevant literature. Chapter 3 provides a detailed explanation of the planned network architecture for LSCIDMR. In Chapter 4, the results produced from the

LSCIDMR dataset are presented. These results are based on several ResNet variations and are then compared with state-of-the-art approaches. Chapter 5 provides a conclusion and briefly outlines the future research directions.

CHAPTER 2: LITERATURE REVIEW

Image categorization has been a focal point of research in the field of satellite and aerial picture analysis for several decades. Aviation places great importance on weather predictions. Weather reports are crucial for ensuring the safety of many lives, particularly those on aircraft. The obstacles in picture classification include the limited availability of extensive datasets, the need for sufficient computational resources, and the disparity between the significant advancements in deep learning theory. The primary obstacles will be addressed subsequently.

2.1 Major Challenges in RSIC

Classifying images based on satellite data presents a significant level of difficulty, unlike other forms of classification. Satellite-based image classification utilizes intricate spatial patterns, resulting in similarities between different classes. As depicted in Figure 2.1, these photos belong to distinct categories yet exhibited similar characteristics. Several significant obstacles are worth mentioning.

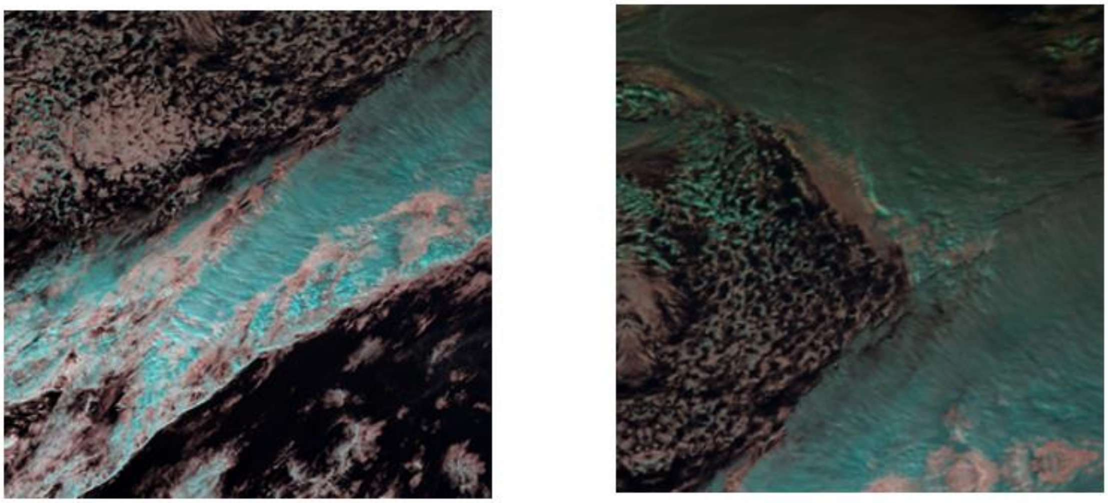
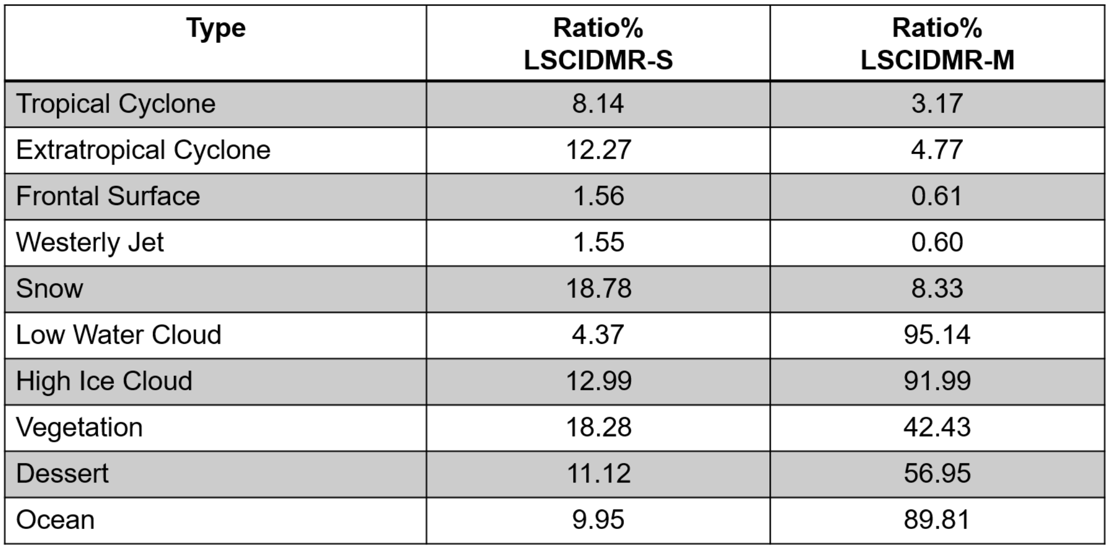


Figure 2. 1: Inter-class similarity between (a) Frontal surface and (b) westerly jet

2.1.1 Inter Class Similarity

The LSCIDMR, which stands for Large-Scale Satellite Cloud Image Database for Meteorological Research, comprises a total of 40,625 pictures and is categorized into 10 distinct classifications. The classes in the dataset have varying quantities of photos, indicating an imbalance in the dataset. The presence of class imbalance poses challenges in both the training and testing of models. As a result, the model exhibits bias towards classes with a high image count and does not perform effectively on classes with a lower image count. This occasionally results in model overfitting, which diminishes the overall accuracy of the model.

Table 2. 1: Detailed summary of class-imbalance



Type	Ratio% LSCIDMR-S	Ratio% LSCIDMR-M
Tropical Cyclone	8.14	3.17
Extratropical Cyclone	12.27	4.77
Frontal Surface	1.56	0.61
Westerly Jet	1.55	0.60
Snow	18.78	8.33
Low Water Cloud	4.37	95.14
High Ice Cloud	12.99	91.99
Vegetation	18.28	42.43
Dessert	11.12	56.95
Ocean	9.95	89.81

The table displays a comparison of the distribution ratios for several meteorological and surface categories between two datasets, identified as LSCIDMR-S and LSCIDMR-M. Each row in the datasets represents a unique meteorological phenomenon or surface feature. The columns display the ratios as percentages for each type within the datasets.

The initial category mentioned is the "Tropical Cyclone," which accounts for 8.14% in the LSCIDMR-S dataset and 3.17% in the LSCIDMR-M dataset. This suggests that the occurrence of Tropical Cyclones is more common in the LSCIDMR-S dataset than in the LSCIDMR-M dataset.

The occurrence of "Extratropical Cyclone" follows a similar pattern, with a higher proportion in the LSCIDMR-S dataset (12.27%) compared to the LSCIDMR-M dataset (4.77%). This indicates that Extratropical Cyclones are considerably more common in the LSCIDMR-S dataset.

The occurrence of the "Frontal Surface" and "Westerly Jet" types in these datasets is quite uncommon. In LSCIDMR-S, the ratio of the "Frontal Surface" type is 1.56% and the ratio of the "Westerly Jet" type is 1.55%. In LSCIDMR-M, the ratios are significantly lower, with the "Frontal Surface" type at 0.61% and the "Westerly Jet" type at 0.60%. This suggests that both occurrences are less frequent in the LSCIDMR-M dataset.

The occurrence of "Snow" in the LSCIDMR-S dataset is noticeable, with a ratio of 18.78%, which is much greater than the 8.33% seen in the LSCIDMR-M dataset. This indicates that there is a higher frequency of data points connected to snow in LSCIDMR-S.

The datasets for "Low Water Cloud" exhibit a significant disparity, with LSCIDMR-S displaying a comparatively low ratio of 4.37%, while LSCIDMR-M demonstrates an unusually high ratio of 95.14%. The LSCIDMR-M dataset places significant importance on low water cloud data.

The "High Ice Cloud" category exhibits a substantial disparity, as the LSCIDMR-S dataset has a ratio of 12.99% compared to 91.99% in the LSCIDMR-M dataset. This indicates a significant amount of high ice cloud data in LSCIDMR-M.

The proportion of "Vegetation" is higher in the LSCIDMR-S dataset (18.28%) than in the LSCIDMR-M dataset (42.43%). Vegetation is abundant in both datasets; however, it is more prominent in the LSCIDMR-M dataset.

The dataset LSCIDMR-S indicates a ratio of 11.12% for the category "Dessert", while the LSCIDMR-M dataset displays a significantly larger ratio of 56.95%. This suggests that the LSCIDMR-M dataset contains a higher number of data points connected to deserts.

Finally, the "Ocean" category comprises 9.95% of the LSCIDMR-S dataset and a notably greater proportion of 89.81% in the LSCIDMR-M dataset. The LSCIDMR-M dataset has a significant emphasis on ocean data compared to the LSCIDMR-S dataset.

The table demonstrates notable disparities in the occurrence of different meteorological occurrences and surface types across the LSCIDMR-S and LSCIDMR-M datasets, with certain types being significantly more common in one dataset compared to the other.

2.2 Numerical weather prediction methods

Now, let us examine the historical background of numerical weather prediction (NWP). Lewis Fry Richardson pioneered manual numerical weather prediction (NWP) in Britain in 1922, and by 1950, early computer-assisted weather forecasts were being generated. Sweden, the USA, and Japan quickly adopted operational weather forecasting.

Numerical Weather Prediction (NWP) refers to a concise set of equations, known as the fundamental equation, that is utilized to compute alterations in weather conditions. Contemporary weather forecasting heavily relies on numerical weather prediction. Lutgens and TarBuck (1989) contend that the term "numerical" is inappropriate when referring to all techniques of weather forecasting, as they are all reliant on quantitative data. The adherence of atmospheric gases to established scientific principles forms the basis for numerical weather prediction. Considering the present conditions, it would be optimal to employ these basic principles to predict the future state of the atmosphere. This situation can be likened to predicting the future position of the moon by using physical laws and utilizing knowledge about its current location. Nevertheless, this procedure is quite demanding because of the multitude of variables that need to be considered while accounting for the changing environment. To do weather prediction at a resolution that produces useable results, it is necessary to use some of the most powerful supercomputers available. These computers are essential for manipulating large datasets and performing complex calculations. Numerical modelling, as described by Houghton (1986), involves the manipulation of equations and boundary conditions to prepare them for solving on high-performance digital computers. The efficacy of the model's processes is assessed by comparing its behaviour to that of the real atmosphere. The development of efficient and dependable systems for regular weather forecasting is the primary application of numerical modelling. This work employs many equations to develop predictive models. Multiple advancements are required to prevent NWP techniques from becoming outdated.

2.3 Feature-based vs Network-based Methods

Prior to the real implementation of deep learning algorithms, feature-based approaches [1] were used to address RSIC. Feature-based approaches involve the extraction of low-level characteristics, such as corners and edges. These features are then subjected to rigorous post-processing and used to classify images using a statistical classifier. Descriptors such as Scale Invariant Feature Transform (SIFT) [2] and Histogram of Gradients (HoG) [3], which are created by humans, are employed to extract features from images. These descriptors primarily rely on local neighborhoods to describe points of interest within the image. Extracting high-level graphical features poses a problem due to the specific nature of these descriptors.

Network-based approaches effectively address these limitations. In order to clarify the meaning behind the picture content network-based methods, these techniques integrate basic features to generate more advanced and intricate features in each consecutive layer. This greatly enhances the capacity of network-based algorithms to learn features for picture categorization. This has a downside in terms of the time it takes to process, the computational resources needed, and the demand for training data. Convolutional neural networks (CNN) are the most effective method for extracting and learning features in the field of network-based approaches. The CNN design, as depicted in section 2.2, primarily comprises a series of convolution operations. These operations effectively combine low-level features to generate higher-level features in each successive layer. Additionally, the architecture includes classification layers that are responsible for making predictions.

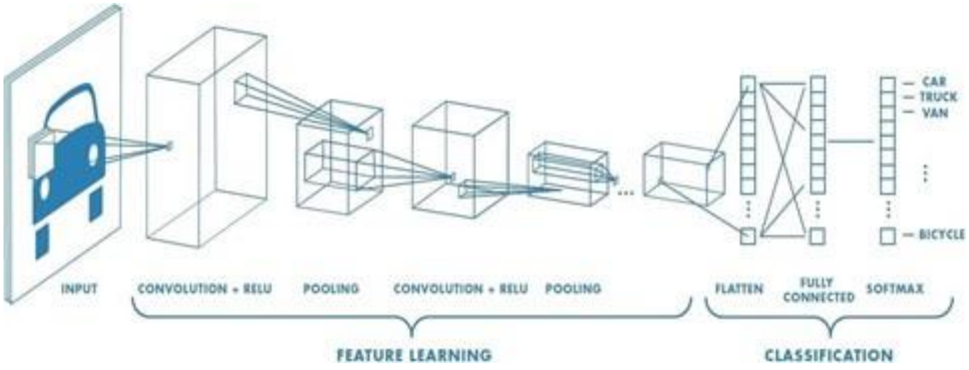


Figure 2. 2: Base CNN architecture

The graphic depicts a fundamental design of a Convolutional Neural Network (CNN), commonly employed for tasks involving the classification of images. The design is partitioned into two primary components: Feature Learning and Classification.

The method in the **Feature Learning** phase commences with an input image. The image undergoes a sequence of convolutional layers, which are then followed by pooling layers. The initial layer employs a convolution technique, utilizing a collection of filters to extract diverse characteristics from the input image. These filters are applied to the image by performing element-wise multiplications and summations. They are designed to capture local patterns such as edges, textures, and simple forms. Following the convolution operation, an activation function, commonly ReLU (Rectified Linear Unit), is utilized to incorporate non-linearity into the model, hence facilitating its ability to comprehend more intricate patterns.

After performing the convolution and ReLU processes, a pooling layer is applied. Pooling layers minimize the spatial dimensions of the feature maps, resulting in reduced computational effort and increased invariance to tiny translations of the input image. The pooling technique, typically max pooling, identifies the highest value within a window of values in the feature map, resulting in a reduction in image size while preserving the most important information. This sequence of convolution, activation, and pooling can be iterated multiple times, with each iteration collecting more complex characteristics and decreasing the spatial dimensions.

The subsequent phase is **Classification**. The result obtained from the previous pooling layer is transformed into a single-dimensional vector, which is subsequently inputted into one or many fully linked (dense) layers. The fully connected layers are responsible for classifying the data by integrating the features acquired throughout the convolution and pooling processes. Every neuron in these layers is linked to every neuron in the preceding layer, enabling the network to acquire intricate combinations of the information extracted by the convolutional layers. Following the completely connected layers, it is customary to apply a softmax activation function. The softmax function transforms the output of the last dense layer into a probability distribution over the potential classes.

Within this particular architectural framework, the ultimate outcome comprises probabilities assigned to distinct categories, namely vehicles, trucks, vans, and bicycles. The predicted label for the input image is determined by selecting the class with the highest likelihood. The CNN architecture efficiently acquires the ability to categorize images by systematically extracting features and integrating them to generate conclusive predictions. The integration of convolutional and pooling layers enables the network to efficiently process the spatial arrangement of images, while the fully linked layers and softmax function offer a reliable means of categorization.

2.4 CNN-Based Methods – A Birds Eye View

In view of the remarkable performance of CNN-based networks in the domain of large-scale visual classification, the researchers regarded CNNs as a viable choice for tackling RS applications [4], [5]. There are three parallel sub-branches of CNN algorithms that are used for picture categorization.

2.4.1 Utilizing Pretrained CNNs for Feature Extraction

Utilizing pre-trained convolutional neural networks (CNNs) as feature extractors is the most direct and uncomplicated option. To categorize remote sensing images, the feature vectors that are created are then inputted into pre-existing statistical classifiers [6], [7]. While the classification accuracy of this method is frequently higher than that of feature-based approaches, there is still ample room for improvement. It is crucial to highlight the outstanding feature description capabilities of CNNs. They outperform hand-engineered descriptors in terms of efficiency and domain expertise, even without being trained on a specific dataset.

2.4.2 Fine-tuning Pretrained CNNs on Target Dataset

This technique involves fine-tuning pre-trained Convolutional Neural Networks (CNNs) using target datasets to improve their performance. After being pre-trained on large-scale semantically related datasets like ImageNet [8], networks are further optimized on small-scale target datasets. This strategy utilizes the pre-trained network's ability to infer scenes, resulting in improved classification accuracy on the target dataset, in proportion to the improvement. During fine-tuning, just a subset of the pre-trained network is utilized to optimize the hyperparameters for the target dataset. The rest of the network remains unchanged, resulting in lower computing costs compared to training the network from the beginning. In addition, the scientists in [9] determined that fine-tuning pre-trained networks leads to superior classification performance for small datasets compared to the technique of training from scratch. This will be addressed in the following section.

2.4.3 Training CNN from Scratch

Although pre-trained networks can be fine-tuned to achieve impressive results, they may not fully capture the complex aspects specific to the target dataset. In addition, when using pre-trained networks, the ability to modify a CNN architecture is somewhat restricted, unlike when employing a training-from-scratch approach. The disadvantage of constructing a network from the beginning is the substantial computational load it necessitates, together with the requirement for enough training data to optimize the network. Therefore, this method is most appropriate for datasets of significant size. To summarize, there are compromises involved in selecting a certain CNN-based approach, given that the condition of having sufficient training data has already been fulfilled. Alternatively, the fine-tuning technique is the optimal choice for training networks when working with datasets of limited size.

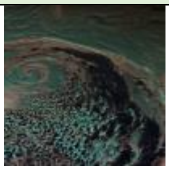
2.5 Remote Sensing Image Classification Benchmarks

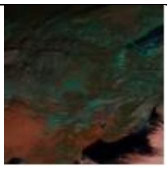
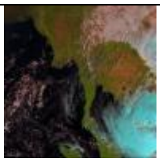
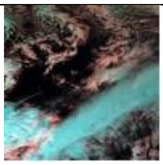
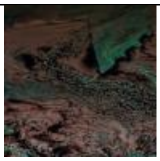
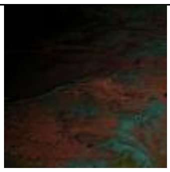
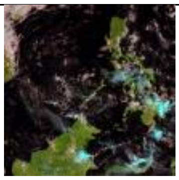
The assessment of overall picture classification performance is greatly dependent on the dataset used for benchmarking. Despite substantial advancements in remote sensing technology, the RS research community still need a benchmark image classification dataset of similar size to ImageNet [8], which is currently only accessible for conventional image classification tasks. The benchmark datasets commonly used in the field of Remote Sensing (RS) include Brazilian Coffee Scene, Big Earth Net, HAINAN Dat, RSCII, SIRI-WHU, RSSCN7, and WHU-RS19. The LSCIDMR data is widely regarded as the most difficult because to its satellite-based cloud images and its notable characteristics of high interclass similarity, class imbalance, resolution, and illumination changes. The publication of the LSCIDMR dataset has enabled academics to effectively utilize deep learning models, significantly enhancing their application in remote sensing research and weather forecasting.

This study's performance analysis utilizes the identical dataset. For testing and evaluating the model's performance, 30% of the dataset is allocated, while the remaining 70% is dedicated to training.

The LSCIDMR has certain guidelines for selecting categories. The rules are formulated by meteorology experts and presented in table 2.2.

Error! Reference source not found. Rules of categories selection in LSCIDMR in the field of meteorology

Type	Image	Classification Criteria
Extra Tropical Cyclone		The center of the extra tropical cyclone should be located within the designated area.
Frontal Surface		Frontal surface on the slide

Snow		Snow is in the slice
Tropical Cyclone		The center of a tropical cyclone should be located inside the designated area.
Westerly Jet		Westerly jet is in the slice
Low Water Cloud		Area (Low Water Cloud) > 50% and Area (else) < 20%
High Ice Cloud		Area (High Ice Cloud) > 50% and Area (else) < 20%
Ocean		Area (Ocean) > 80% and Area (else) < 20%
Desert		Area (Desert) > 50% and Area (else) < 20%
Vegetation		Area (Vegetation) > 50% and Area (else) < 20%

2.6 Related Work

Clearly, for a weather forecast to be possible, there must be established ways by which it is conducted. Numerical Weather Prediction (NWP) is a long-standing approach for weather forecasting. It involves a simplified set of equations, referred to as the fundamental equation, which is utilized to calculate changes in meteorological conditions. Contemporary weather forecasting heavily relies on numerical weather prediction. Nevertheless, this procedure is highly complex because it requires considering multiple variables while accounting for the ever-changing environment. To do weather prediction at a resolution that produces meaningful output, it is necessary to use some of the most powerful supercomputers to handle large data sets and perform complex calculations. Numerical modelling, as described by Houghton (1986), involves the formulation of equations and boundary conditions in a suitable manner prior to their solution on high-performance digital computers. The efficacy of the model's processes is evaluated by comparing its behaviour to that of the real atmosphere.

Lutgens and TarBuck (1989) contend that the term "numerical" is misleading when used to all techniques of weather forecasting, as they are all reliant on quantitative data. The adherence of atmospheric gases to established scientific principles forms the basis for numerical weather prediction. Considering the present conditions, it would be optimal to utilize these basic principles to predict the future condition of the atmosphere.

Other historical methods of weather prediction include:

1) Persistence forecasting is the most straightforward method of prediction, based on the assumption that current conditions would persist in the future. In the tropics, where the weather remains consistent over the summer months, the ability to forecast the weather relies on the current conditions of the day.

2) Climatology forecasting: Climatology forecasts are based on the observation that the weather conditions on a particular day at a certain area remain relatively stable from one year to another.

3) Observing the sky: One of the key meteorological characteristics that can be used to forecast weather in mountainous areas is the state of the sky, together with pressure tendency. Over time, the use of sky cover in weather forecasting has led to the development of several weather legends.

4) Utilizing a barometer: Forecasting has depended on the measurement of barometric pressure and its trend since the late 19th century. The magnitude of the pressure differential, particularly if it exceeds 2.54mmHg, directly correlates with the magnitude of the anticipated weather shift.

5) Nowcasting: This phrase denotes the act of forecasting the weather conditions for the upcoming six-hour period. In this time range, smaller features like as isolated showers and thunderstorms, as well as other minute factors that cannot be accurately depicted by a computer model, can be adequately forecasted. By utilizing the latest radar, satellite, and observational data, individuals may effectively examine the minute details and features, thereby generating a more precise forecast for the next few hours.

6) Analogue Forecasting: This intricate method necessitates the forecaster to remember a previous weather incident that is expected to be replicated by an upcoming event.

7) Ensemble Forecasting: To address the uncertainty in the starting state of the atmosphere, ensemble forecasting involves generating multiple forecasts to account for mistakes in observations and limited sampling.

During the past decade, substantial works have appeared in literature that address the weather classification problem through various methods developed. In this section, we briefly discuss recent developments in the weather classification domain. The authors in [10] introduced a new and efficient weather forecasting model by studying the time-dependent modelling approaches. The complex model comprises two types of correlations. This model demonstrates exceptional accuracy in predicting weather conditions for a duration of up to 12 hours, surpassing the performance of the climate science predicting algorithm. Booz et al. [11] introduced a weather forecast system based on deep learning, which utilizes re-al-world data for analysing data volume and recency.

In this approach, [12] Pushpa et al. tells the dimensionality of the data was reduced using SOM, followed by a reduction in the amount of data used for climate forecasting. This method demonstrates a significant increase in accuracy, ranging from 7% to 23%, when compared to existing methods for predicting weather and agricultural outcomes. In addition, R. Meenal et al. [13] found that AI-based techniques.

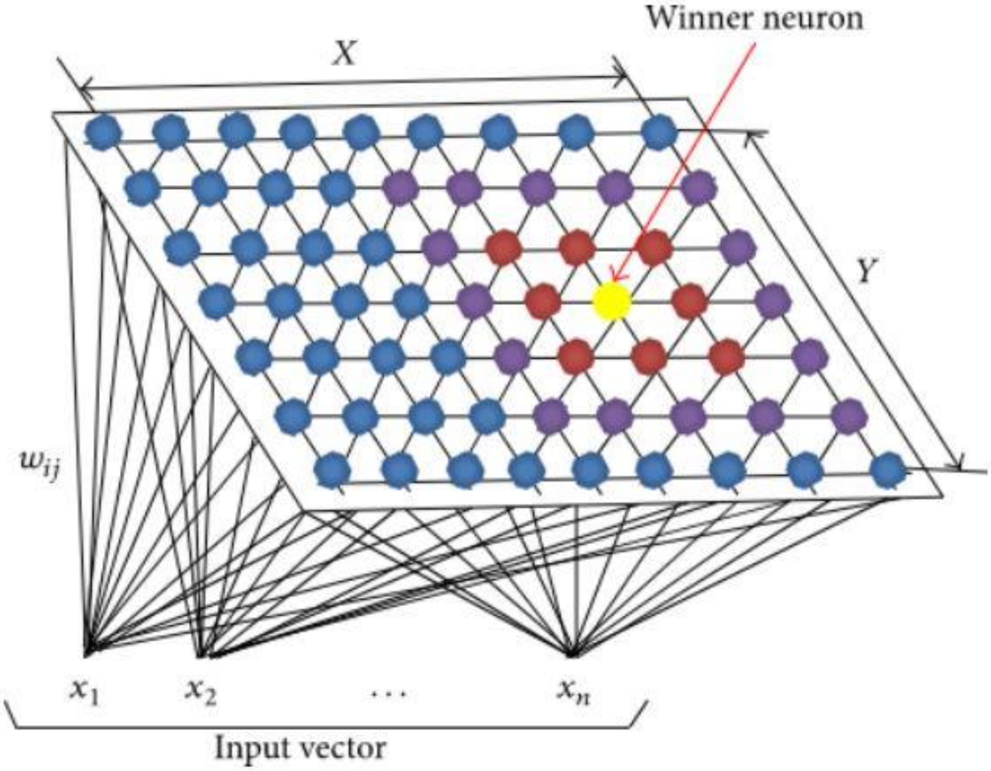


Figure 2. 3: Self-Organizing Maps for Anamoly Detection

An LSTM model based on deep learning was developed by Anil Utku et al. [14]. The study determined that the established LSTM model performed much better than the benchmarks that were considered. In [15], authors proposed an ensemble machine learning approach, utilizing artificial neural networks, the SVR, and Gaussian processes for classification and prediction. This method incorporates wind generation forecasts generated by numerical weather prediction models, as well as meteorological observation data collected from weather stations. In [16], researchers demonstrates that the UNET model, which is based on residual learning, can effectively uncover the underlying physical connections to precipitation. This enables the incorporation of physical restrictions into operational models, leading to enhanced accuracy in precipitation forecasts. Their findings are crucial as they have paved the path for the advancement of online, hybrid prediction algorithms.

In [17], researchers proposed that deep learning architectures might be effectively designed to handle the non-linear nature. This work provides a comprehensive overview of the latest research in deep learning-based weather forecasting. The acquired discoveries are subsequently emphasized, with a focus on the asserted precision and the extent to which the model may be applied to new situations. Ultimately, the writers list the factors involved. that are influenced by other factors and those that are not influenced by any other factors in the context of weather forecasting. They then proceed to evaluate the most effective timing strategies for training the dataset. Kang et al. [18] provide in their work an environmental detection system, utilizing deep learning techniques to account for atmospheric conditions. The technique employs an automated process to categorize an input image into its appropriate categories. To assess the effectiveness of the suggested approach, the GoogLeNet and AlexNet models were applied to an open weather image dataset, and the viability of the strategy was confirmed.

Cong et al. [19] suggested a precipitation picture database for study on meteorology called the Large-Scale Cloud Image Database (LSCIDMR). Researchers assert that this is the very initial telescopic weather image database that is available to the public and may be used for meteorological research. The LSCIDMR dataset consists of two types of notes, both individual and numerous, and it has a total of 104,390 images that cover 11 different classes. Every picture class defines different kinds of characteristics. A grand total of 414,221 labels categorized as several labels (LSCIDMR-M) and 40,625 labels categorized as single labels (LSCIDMR-S) were gathered. The LSCIDMR-S indicates that each image is assigned to only one class and possesses distinct characteristics of that class. On the other hand, the LSCIDMR-M indicates that images can be assigned to one or more classes as they possess characteristics of multiple classes. Several advanced deep learning techniques, including VGGNet-19, Res-Net-101, AlexNet, and EfficientNet, are utilized to analyze these photos. The outcomes obtained from these approaches serve as a reference point for future research.

For weather classification, the method reported in [12] uses CNN with Transfer Learning to classify weather images, including cloudy, rainy, shine, sunrise, snowy, and foggy. Four CNN architectures (MobileNetV2, VGG16, DenseNet201, and Xception) were used for classification. The proposed method has an average accuracy of 90.21% and a training time of 2,438 seconds, with Xception showing the best performance. Researchers devised an inexpensive, nearly instantaneous rainfall monitoring system to reduce the risk of urban flooding. The method employs Transfer Learning, utilizing a Convolutional Neural Network, to accurately identify rainfall in individual photos captured under diverse situations. The model attained an accuracy of 85.28% and an F1 score of 0.86. It can be incorporated into warning systems for the automated surveillance of rain-related incidents. Nevertheless, the model's ability to detect is restricted in the absence of measurement.

In [20], authors present a novel training approach for meteorological identification by combining the strengths of EVA02 and Linear Support Vector Classification through an improved fusion technique. The model refines the pre-trained EVA02 network, improves the process of extracting features, and combines the Nystroem approach with Linear SVC for classification. The results of simulation studies demonstrate that the model surpasses existing approaches by 0.75% and 9.89% on the publicly available datasets MWD and WEAPD.

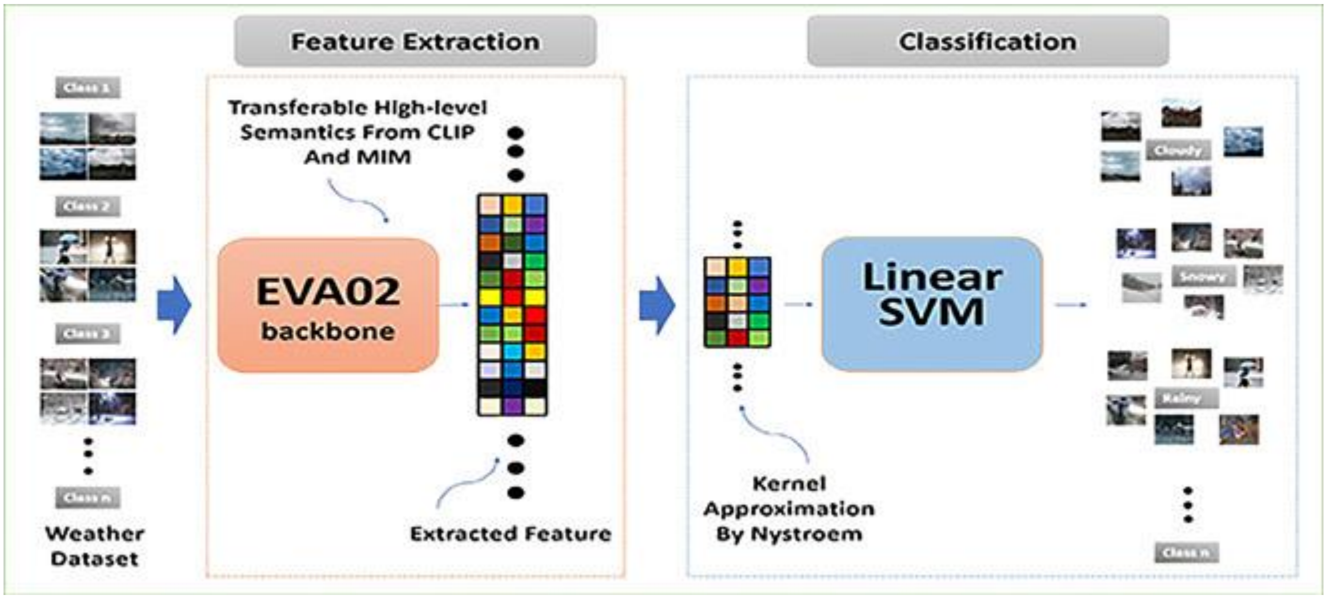


Figure 2. 4: The Framework of the Fusion Enhanced Model Based on EVA02 and LinearSVC

Jun et al. [21] suggests utilizing a vision transformer to achieve precise weather image identification, with the objective of precisely detecting and categorizing weather patterns and situations. The vision transformer model outperformed previous methods with a remarkable accuracy of 99.58%, demonstrating the potential of deep learning techniques in accurately recognizing weather patterns. Another study [22] shows data collection comprises 9 distinct labels for each weather condition, light intensity, and street type. The dataset was captured using a front camera in an industrial setting and then reduced to 1.1k photos. The dataset is currently unavailable for autonomous driving. However, the baseline ResNet18 network demonstrates exceptional performance in two datasets that are not related to automotive applications. Scientists cooperated with NASA to create machine learning algorithms for identifying TCB clouds in satellite imagery. By employing CNN models and data preprocessing techniques, researchers discovered that training CNNs on the original-colored satellite photos produced the most optimal outcomes. Additional research is required to ascertain the optimal edge detection threshold for various satellite images [23].

The few examples described above provide a strong case for considering satellite image classification as a crucial step in the development of an intelligent classification system. The aforementioned approaches represent a limited selection of successful endeavors, which are aimed at tackling the issue of picture classification in various scenarios. Our method is one of the latest additions in this research domain. In next section, we describe our developed method in detail.

Since the 1950s, there has been significant progress in numerical weather forecasting. Due to limited processing resources, equations had to be computed on a coarse spatial grid until recently. Unresolved small-scale processes are represented using simplified methodologies known as 'parameterization schemes', which might restrict the accuracy of forecast predictions. Advancements in computer power have enabled the development of weather forecasting models that can provide highly detailed predictions, with grid resolution at scales of over one kilometre. Although numerous processes are still parameterized, the use of more refined grids allows for the precise calculation of storm trajectories and deep convective events.

As technology advances, enhancements are made to current methods. Currently, weather forecasting is transitioning towards the utilization of Artificial Intelligence, machine learning, and deep learning algorithms, either individually or in combination. This transition is currently ongoing. ML and DL based techniques have led to the development of many successful algorithms that excel in scene classification. Prior to forward with the recommended plan, we will now examine some of the notable RSISC methodologies.

ML approaches have recently gained a lot of excitement due to their frequent usage of deep neural networks and graphical frameworks to identify correlations in non-linear data. Deep learning techniques make use of a directed graph. The data is entered at the lower end of the graph, undergoes modifications through hidden layers, and is subsequently outputted at the same location. In the context of a graph, the connections between nodes are referred to as "weights," whereas the individual nodes themselves are referred to as "biases." The weights determine the degree of connectivity between neurons in different layers, while the bias serves as an adjustment that regulates the responsiveness of the neuron. Each neuron's signal is multiplied by an activation function, which is determined by the weighted input from the previous layer. The weights and biases are adjusted according to the error tolerances present in the training data sets. Once trained, these directed graphs can generate predictions on test data that is not part of the training sets. The ability of a model to accurately represent the features of recent severe events (referred to as "out of sample" events) in climate research, like Gaussian processes, enhances the reliability of predicting future anomalies that may become more common. Deep learning and neural network approaches avoid the need for process-based models, such as those required for Sequential Monte Carlo and Bayesian methods. This data-driven approach allows for a more comprehensive understanding of the interconnected relationships in nonlinear systems. The text encompasses a wide range of characteristics exhibited by contemporary neural network applications in the field of climate studies.

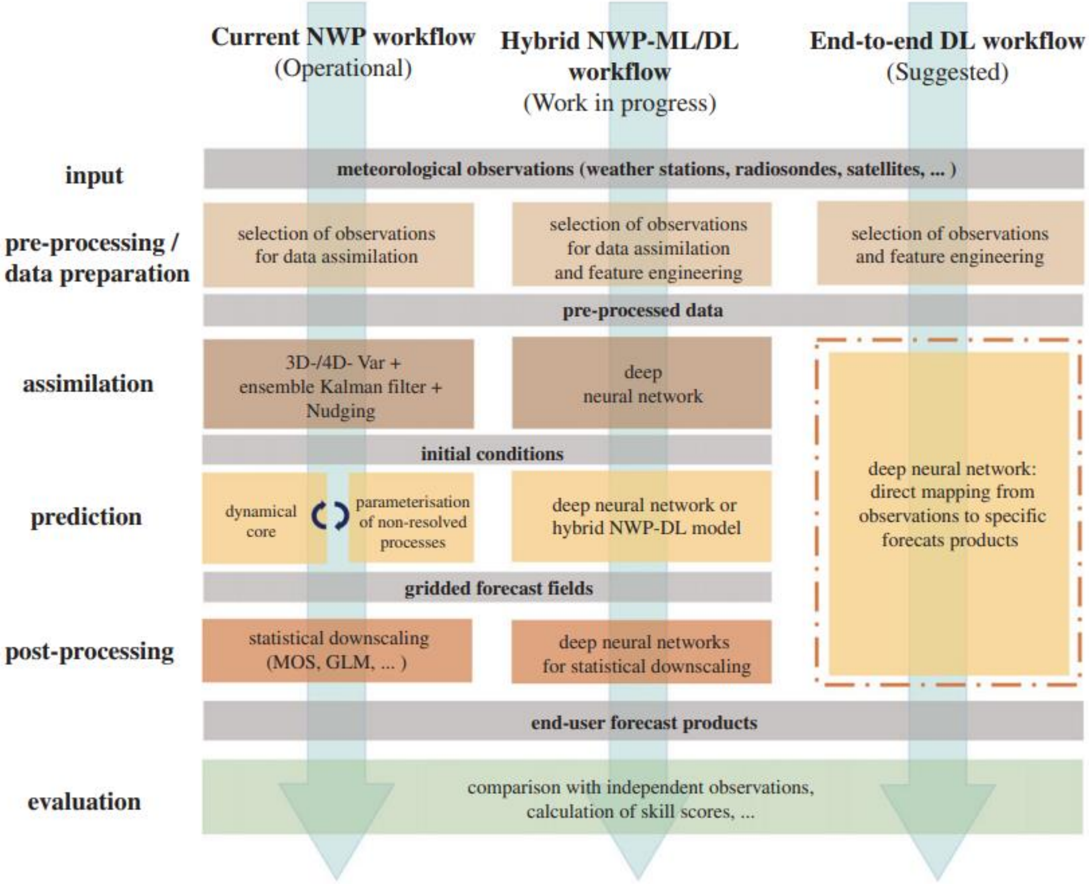

Current NWP workflow
(Operational)
Hybrid NWP-ML/DL workflow
(Work in progress)
End-to-end DL workflow
(Suggested)
input
meteorological observations (weather stations, radiosondes, satellites, ...)
pre-processing / data preparation
selection of observations for data assimilation
selection of observations for data assimilation and feature engineering
selection of observations and feature engineering
pre-processed data
assimilation
3D-/4D- Var + ensemble Kalman filter + Nudging
deep neural network
initial conditions
prediction
dynamical core
parameterisation of non-resolved processes
deep neural network or hybrid NWP-DL model
deep neural network: direct mapping from observations to specific forecats products
gridded forecast fields
post-processing
statistical downscaling (MOS, GLM, ...)
deep neural networks for statistical downscaling
end-user forecast products
evaluation
comparison with independent observations, calculation of skill scores, ...

Figure 2. 5: Flow chart Current NWP, Hybrid NWP-ML/DL and End-to-end DL

As mentioned in section 2.4, it is widely recognized in the field of deep learning that ensembles of CNNs surpass their solo counterparts in performance. The research in [24] and [25] examines several CNN assembling methods that utilize established CNN models as backbone architectures. It has been observed that RSIC experiences a significant improvement in classification accuracy when applied to network ensembles.

It should be emphasized that RSIC has limited applicability for more complex CNN architectures like as ResNets. Most of the research contributions mentioned above utilize the fundamental architectures of the moderately less deep neural networks, such AlexNet, VGGNet, or EfficientNet. The absence of extensive scene datasets, time-consuming training processes, and the requirement for substantial computational resources are the primary obstacles that hinder the broad adoption of deeper networks in the remote sensing industry. Compared to the vast ImageNet dataset, which includes over a million annotated images, the LSCIDMR satellite imaging dataset, currently considered the benchmark for the RSIC community, is rather small. Residual neural networks (ResNets) with little data are susceptible to overfitting, a phenomenon that restricts their ability to reach their maximum potential. Hence, our study aims to surmount the performance limitation that has been placed on the RSIC problem when working with deep CNN architectures like as ResNets. To achieve this objective, our strategy involves initially implementing the necessary precautions to prevent overfitting. Subsequently, we will incorporate a suitable network ensembling technique, along with the required logical modifications to the general design of the deep CNN baseline models.

2.7 Contribution

2.7.1 Enhancing Model Performance

Self-learning and semi-supervised learning work well together to handle big sets of unlabeled data. A model is first trained on a small set of named data. This is called self-learning, which is also called self-training. Then, this model is used to guess what names should go on the large set of unlabeled data. The model is retrained over and over again with results that have a high chance of being true. This cycle will keep going as long as there aren't any big changes in how well the plan works. Self-learning uses the model's ability to make accurate fake labels, which successfully increases the labeled data set without requiring any manual annotation.

When training with semi-supervised learning, on the other hand, both identified and unlabeled data are used. Some techniques used in semi-supervised learning are pseudo-labeling and consistency regularization. In consistency regularization, the model is told to produce similar outputs for the same input when things are changed. In pseudo-labeling, the model's estimates on data that hasn't been labeled are used as if they were real labels. These methods take advantage of the way the data is structured to get the most out of the information in both named and unlabeled data. A small amount of named data and a large amount of unlabeled data are used to train semi-supervised learning models to learn representations that work well in a wide range of situations.

When you combine these two methods, the first step is for self-learning to bootstrap the original model with fake labels, which makes the labeled data set bigger. Then, this bigger set of data is put into a semi-supervised learning framework. This framework improves the model by learning from the bigger set of labeled data while also using regularization methods on the unlabeled data. Self-learning can quickly create a large set of labeled data, and semi-supervised learning can improve the model's performance by combining both types of data. This combined approach takes the best parts of both approaches.

This mix works especially well in fields like natural language processing, computer vision, and bioinformatics where named data is hard to come by but unlabeled data is plentiful. The model can become very accurate and reliable by using pseudo-labels and semi-supervised methods to be improved over and over again. This method cuts down on the need for significant labeling by hand, which saves time and resources while keeping or even improving the accuracy of the model's predictions.

2.7.2 Iterative Self-Learning with Pseudo-Labels for Continuous Model Refinement

Iterative self-learning is a technique used in machine learning and artificial intelligence to continuously improve a model's performance over time. This method requires first training the model on a labelled dataset, where it learns to predict or categorize data points depending on the attributes provided. Following the initial training phase, the model is exposed to a bigger pool of unlabelled data. The model makes predictions on the unlabelled data, resulting in pseudo-labels. These pseudo-labels are essentially the model's best estimate at what labels these additional data points should have.

This self-learning process relies heavily on the use of pseudo-labels. After creating pseudo-labels, the model uses these new, self-generated labels to retrain itself. This is done iteratively, which means that the model undergoes repeated cycles of prediction and retraining. Each iteration tries to increase the model's accuracy and generalizability. The model successfully expands its training dataset using pseudo-labels, eliminating the need for extra manual labelling, which may be time-consuming and costly. This iterative retraining allows the model to better understand the underlying patterns in the data, resulting in more accurate predictions over time.

The continuous refining method ensures that the model evolves in response to fresh data inputs rather than remaining static. This adaptability is especially useful in dynamic contexts where data distributions may alter, a process known as notion drift. The model may adapt to these changes by continuously adding new pseudo-labels and retraining, hence keeping its performance and relevance. Furthermore, this strategy can aid in harnessing vast amounts of unlabelled data, which are frequently more readily available than labelled data, increasing the model's robustness and usefulness in real-world scenarios.

Overall, iterative self-learning with pseudo-labels is an effective machine learning technique. It enables models to continuously develop, adapt to new data, and maximize the use of existing resources, resulting in more effective and efficient learning systems.

2.7.3 Enhancing Semi-Supervised Learning with Confidence-Based Pseudo-Label Selection

In semi-supervised learning situations, confidence-based picking of pseudo-labels is a way to improve the quality and usefulness of labels. Labelled data is hard to find and costs a lot of money, but it's what standard supervised learning uses to train models. The goal of semi-supervised learning is to improve model performance by using both named and unlabelled data. Making fake labels for the data that doesn't have labels is a popular way to use this framework. The model can then use these labels as if they were real ones. However, it is hard to make sure that these fake labels are correct and useful, since bad labels can hurt model performance.

To deal with this problem, confidence-based selection of pseudo-labels checks how sure the model is that its estimates on the unlabelled data are correct. There are many ways to measure confidence, such as the softmax output of neural networks, which says that more confidence means a higher chance of belonging to a certain class. As pseudo-labels, only guesses that meet a certain level of confidence are chosen. This process makes sure that only the most reliable fake labels are used for further training. This lowers the chance of adding wrong labels that could lead the model astray.

There are two good things about this method. First, it keeps the purity of the training data by adding noisy labels as little as possible. Since estimates with high confidence are more likely to be right, the fake labels that were added to the training set are better. In turn, this helps the model learn better because it isn't getting confused by wrong information. Second, confidence-based selection helps put the most important data pieces at the top of the list. Models often have too much faith in their predictions when the examples are easy to classify and not so much when the examples are less clear. By focusing on predictions with a high level of confidence, the model can learn from cases where it is more certain, which makes it more robust and improves its overall performance.

This method can also be changed on the fly while it is being trained. As the model improves and changes, the confidence level can be raised, which lets the pseudo-labels be chosen with more care. Instead, a smaller threshold could be used in the beginning of training to include a wider range of data points, which would make learning more enjoyable. This flexibility makes sure that the model always has access to the best and most relevant data, which improves its ability to generalize and work with data it hasn't seen before.

To sum up, confidence-based picking of pseudo-labels is an important part of semi-supervised learning. It raises the quality and usefulness of pseudo-labels by using the model's faith in its predictions. It helps keep data accurate and useful by getting rid of estimates with low confidence. This makes model training and performance better. The flexibility of this method during training makes even more sure that the model always learns from the best data, which leads to more accurate and trustworthy results.

CHAPTER 3: PROPOSED METHODOLOGY

Our objective is to employ deep learning techniques for classifying satellite remote sensing images.

- We will utilize ResNet variations as our baseline models due to their outstanding performance in traditional image classification tasks, as discussed in chapters 1 and 2.
- A refinement technique considering a pre-existing deep Convolutional Neural Network (CNN) model.
- The LSCIDMR satellite imaging dataset serves as a benchmark for evaluating performance.

To tackle the challenges in the research community, it is essential to examine efficient optimization strategies and logical architecture modifications to fully utilize the potential of highly complex CNN networks such as ResNets for the task of classifying remote sensing images (section 2.1). In pursuit of this goal, we will now examine the sequence of tests and inquiries that led to the proposed methodology discussed later in this chapter.

3.1 Introduction to ResNets

Before we start our methodological approach, it is crucial to have a concise overview of ResNets. To mitigate the issue of the vanishing/exploding gradient problem [26], [27] that gets more pronounced as we delve further into neural networks, the authors in [28] introduced a residual block to be incorporated into the simple CNN architecture. This would facilitate the utilization of the inherent discriminatory powers of deep Convolutional Neural Networks (CNNs). The inclusion of a shortcut or skip connection, as depicted in Figure 3.1, allows the residual block to link the output of a previous layer to the feature map (before ReLu activation) of a subsequent layer. Although they enable far greater network penetration, these identity mappings do not have a negative impact on network performance. As the neural network's layers become more complex, they are able to learn more intricate information, resulting in improved accuracy when classifying images. This greatly enhances the network's ability to acquire and process features. Table 3.1 provides a concise summary of the structural details of various proposed versions of ResNet. ResNets demonstrated remarkable picture classification capability on the extensive ImageNet dataset. The pioneering methodology that enabled a more profound exploration of the network surpassed the superficial competing networks. This promotes the utilization of Residual Neural Networks (ResNets) for the classification of satellite Remote Sensing (RS) images.

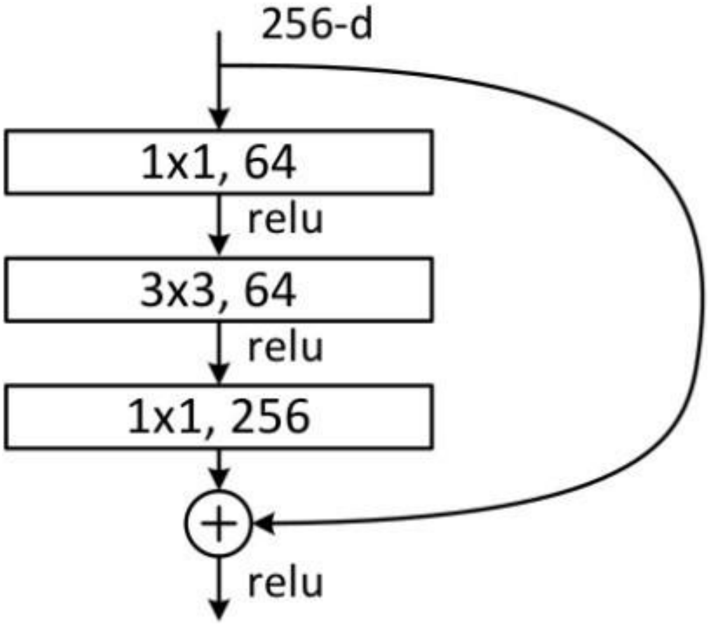
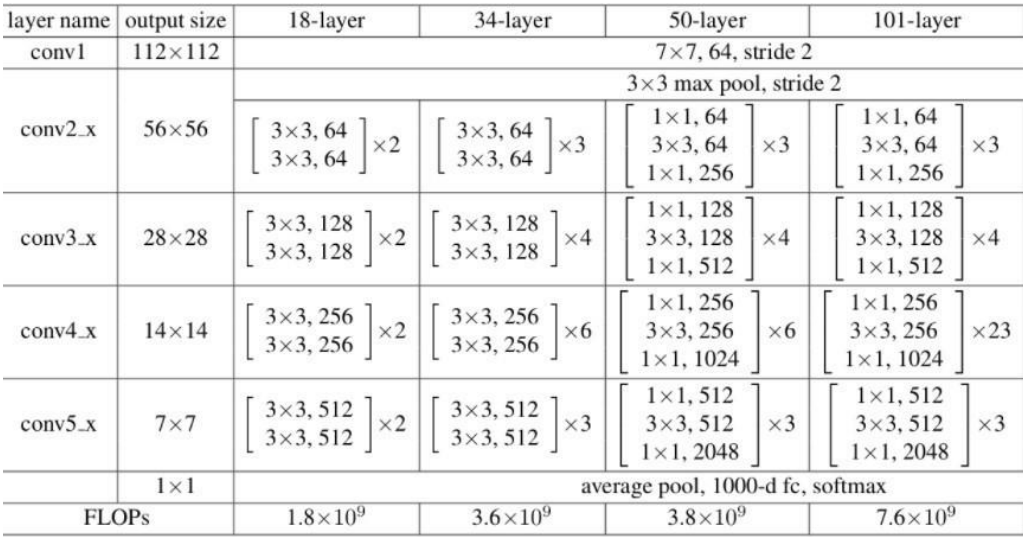


Figure 3. 1: Residual block responsible for identity mappings in ResNets.

Table 3. 1: Structural details of different ResNet architectures



layer name	output size	18-layer	34-layer	50-layer	101-layer
conv1	112×112	7×7, 64, stride 2			
conv2_x	56×56	3×3 max pool, stride 2			
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax			
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9

3.2 Evaluation Metrics

The commonly used performance measures in satellite Remote Sensing Image Classification (RSIC) are:

- The Overall Accuracy (OA) measures the classifier's overall performance over the entire test dataset.
- The Confusion Matrix (CM) is a visual representation that provides a comprehensive summary of categorization outcomes for a specific class. It aids in identifying the specific errors made by the classifier when categorizing a certain group.
- Precision (P) is a metric that quantifies the accuracy of positive forecasts by measuring the proportion of true positives.
- Recall (R) is a metric that quantifies the classifier's ability to accurately predict positive cases out of all the positive examples in the dataset. It is occasionally known as Sensitivity as well.
- The F1-Score is a metric that combines precision and recall. The term "harmonic mean" is commonly used to refer to the average of two values. The harmonic mean is an alternative method for calculating the "average" of numbers.

3.3 Proposed Training Regime

Our method is inspired by deep learning domain as recently these methods have proved to be successful in handling various weather classification related tasks. The classification of cloud images using remote sensing satellites is still a difficult task because of the similarities between different classes and the imbalance in the number of instances in each class. To tackle this problem, we employ a ResNet based model. Our method comprises several interconnected steps. Below we briefly explain each step. Figure 3.3 shows the flow of our developed method.

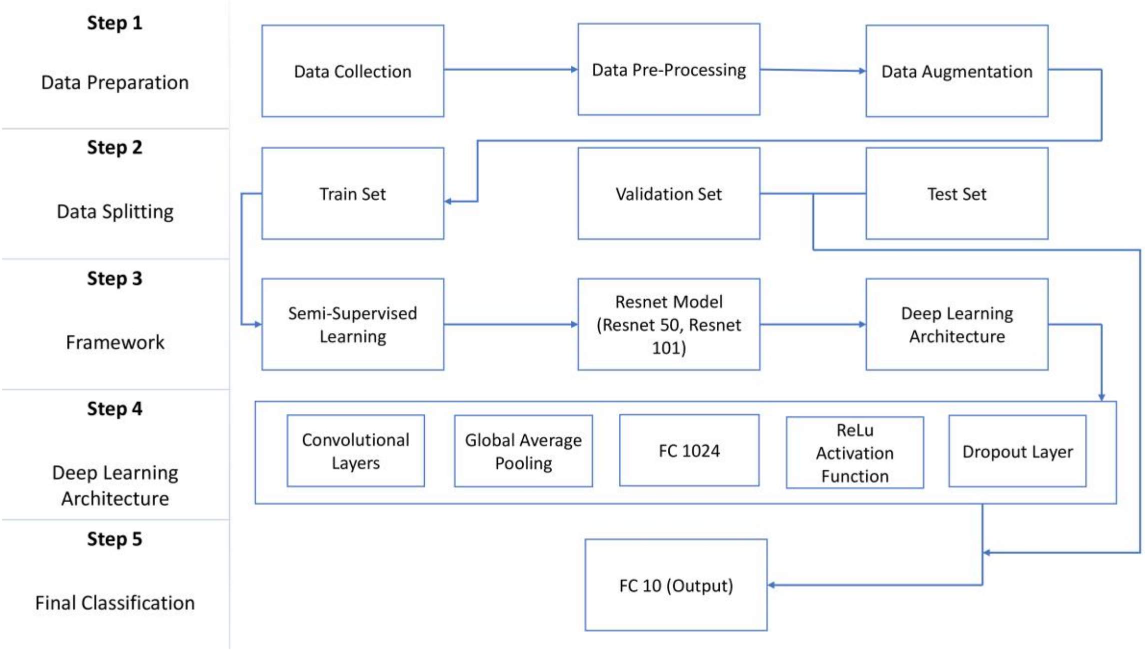


Figure 3. 2: Proposed System Flow

The flow diagram illustrates a thorough workflow for a deep learning project, organized into five primary stages: Data Preparation, Data Splitting, Framework, Deep Learning Architecture, and Final Classification.

The first step, Data Preparation, commences with Data Collection, which entails the acquisition of unprocessed data that is relevant to the current activity. Subsequently, the unprocessed data undergoes Data Pre-Processing, a pivotal stage aimed at purifying and structuring the data, guaranteeing its appropriateness for subsequent analysis.

Subsequently, Data Augmentation is executed to artificially amplify the magnitude and heterogeneity of the training dataset, hence enhancing the model's resilience and efficacy by producing alterations of the current data through techniques such as rotation, translation, and scaling.

Step 2, Data Splitting, partitions the pre-processed and enriched data into three separate sets: the Train Set, Validation Set, and Test Set. The Train Set is employed for training the deep learning models, whereas the Validation Set is used to fine-tune the hyperparameters and validate the model during the training process. The Test Set is not observed during the training phase and is employed to assess the ultimate model's performance, guaranteeing its ability to generalize effectively to unfamiliar, unobserved data.

During Step 3, known as the Framework stage, the workflow diverges into two separate branches. Semi-Supervised Learning is a branch that utilizes both labeled and unlabeled data to enhance the efficiency and accuracy of learning. The alternative branch of study is around the ResNet Model, specifically Resnet 50 and Resnet 101. This deep learning architecture is renowned for its proficiency in image recognition tasks, thanks to its residual learning framework. This branch also encompasses the wider Deep Learning Architecture, emphasizing the adaptability to incorporate more deep learning models as needed.

Step 4, Deep Learning Architecture, explores the individual components of the model architecture in further detail. Convolutional layers serve as the essential components of the model, responsible for extracting features from the input images. Afterwards, Global Average Pooling is applied to decrease the spatial dimensions of the feature maps. This results in a reduced number of parameters and helps prevent overfitting. An FC 1024 layer is added to further process the extracted features. The Rectified Linear Unit (ReLU) Activation Function incorporates non-linearity into the model, hence improving its capacity to learn intricate patterns. A Dropout Layer is included in order to mitigate overfitting by stochastically changing a portion of input units to zero throughout the training process.

The procedure concludes in Step 5, which is the Final Classification. The extracted features are then fed into a Fully Connected Layer (FC 10), which generates the ultimate output, often indicating the probabilities of different classes in a classification problem. The ultimate layer of the model delivers the predictions, so finalizing the deep learning pipeline from data preprocessing to model inference.

In summary, this diagram presents a methodical strategy for constructing a deep learning model, highlighting the significance of each stage in guaranteeing the model's efficacy and dependability.

3.3.1 Data Collection

The training and testing of the model were conducted utilizing a Low Spatial and Temporal Resolution Cloud and Image Data Record collected from the Himawari is-8 observatory [29]. This is the inaugural weather satellite with the capability to capture colour photographs and monitor genuine cloud formations. The LSCIDMR dataset contains 104,390 excellent quality pictures that represent the 11 classes. The time interval between measurements is 10 minutes, and the distance between each measurement point is 2.0 kilometres. Further details of this dataset are given in Section 4.

The LSCIDMR Dataset: The collection of data consists of characteristics of four networks, as shown in Table 1.

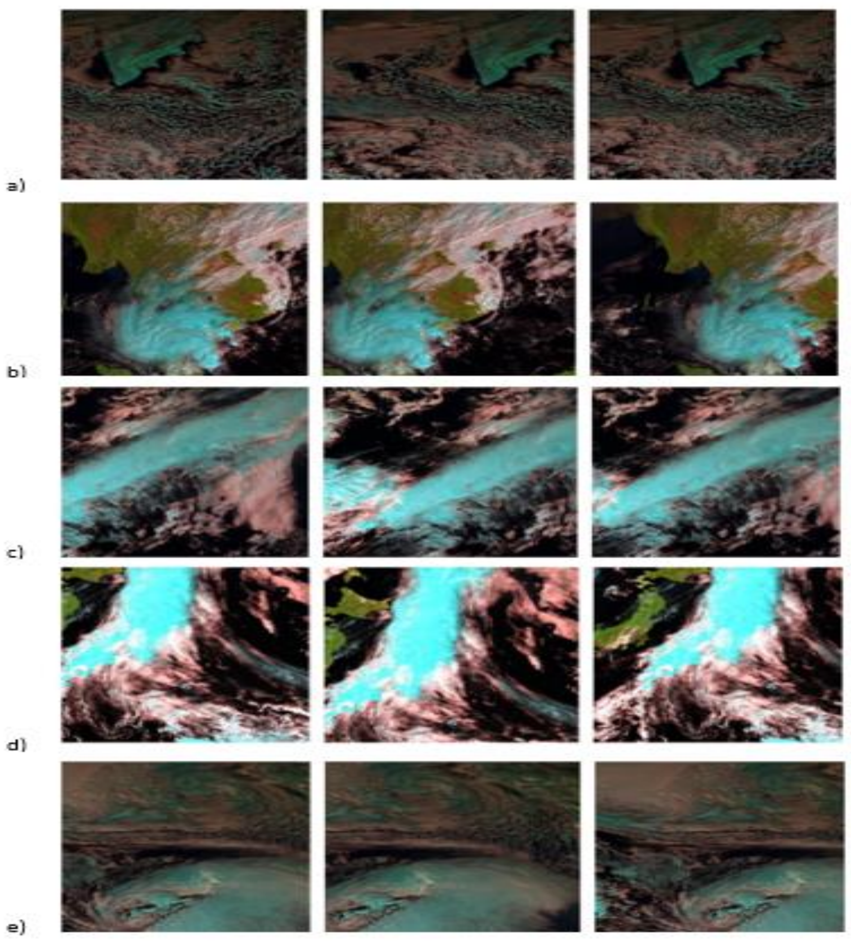


Figure 3. 3: (a) High Ice Cloud (b) Tropical Cyclone (c) Westerly Jet (d) Frontal Surface (e) Extra Tropical Cyclone

3.3.2 Data Augmentation

To mitigate the issue of overfitting, we have focused exclusively on exploring the techniques which may be integrated into the network's planning stage. Amplification of data solutions are crucial in addressing network overfitting difficulties by targeting the underlying cause, which is a lack of training data. These tactics have proven to be successful in mitigating the issue. The objective of these tactics is to enhance the data in a manner that allows the model to discover new explanatory characteristics. Despite the presence of differences in perspective, size, illumination, occlusion, colour, etc. within the target dataset, Various data augmentation strategies are accessible for vision-based systems with the aim of improving model performance. [30]. The LSCIDMR dataset is considered the standard for evaluating the effectiveness of categorization of cloud images using satellite-based technology. The dataset is often recognized as the most challenging due to its 10 distinct classes and the significant overlap between these classes., as well as imbalanced class distribution.

As previously stated, the dataset is very imbalanced. Certain datasets contain thousands of photographs for certain classes, whereas other have only several pictures. This also leads to model bias to categories with many images and hinders the model via showcasing its whole capabilities. In order to tackle the issue of class imbalance, the underrepresented classes undergo specific augmentation techniques to ensure that they have a similar number of photos as the classes with a greater number of images.

Instead of immediately using commonly used data augmentation approaches that include geometric alterations in photographs, we choose to begin with the recently published random erasing data augmentation strategy [31]. The concept of random erasing aims to tackle the issue of image obstructions. The term "occlusion" refers to any indistinct objects or elements in a photograph. This methodology promotes the network to give priority to the overall composition of the image, instead of excessively focusing on specific local characteristics. It achieves this by randomly eliminating distinct regions of the image and substituting them with randomly generated pixel values. By augmenting the data using this technique, the network will acquire more distinctive features that are unique to each category in the dataset, hence boosting the accuracy of scene categorization. Furthermore, it will partially resolve the problem of similarity across different classes in the satellite RSIC scenario by preventing the network from solely depending on scene details.

Afterwards, we continued to train the FC for review-1024 ResNet50 model by implementing the random erasing strategy to take use of its benefits. We then compared the outcomes with the results obtained only from the utilization of methods integrated into the network architecture. The random wiping data augmentation approach alone greatly improved the accuracy of scene categorization (Figure 3.4), as described earlier.

Subsequently, our attention was directed towards the prevailing techniques of data augmentation, which encompass geometric and colour space changes the enumerated strategies are as follows:

- Vertical and horizontal flipping of images
- Rotating images
- Shifting the height and breadth of images
- Varying the brightness of images
- Zooming in and out of photographs
- Applying shear to images
- Applying different scales to pictures
- Varying the contrast of images
- Performing several augmentations on images

The distinguishing features of the image being analysed are determined by each of the above data augmentation methods, rendering them all easy to understand. The act of simultaneously applying two or more of the data augmentations is referred to as the multiple augmentation technique. To properly utilize the enhanced data, we have employed a diverse range of random permutations.

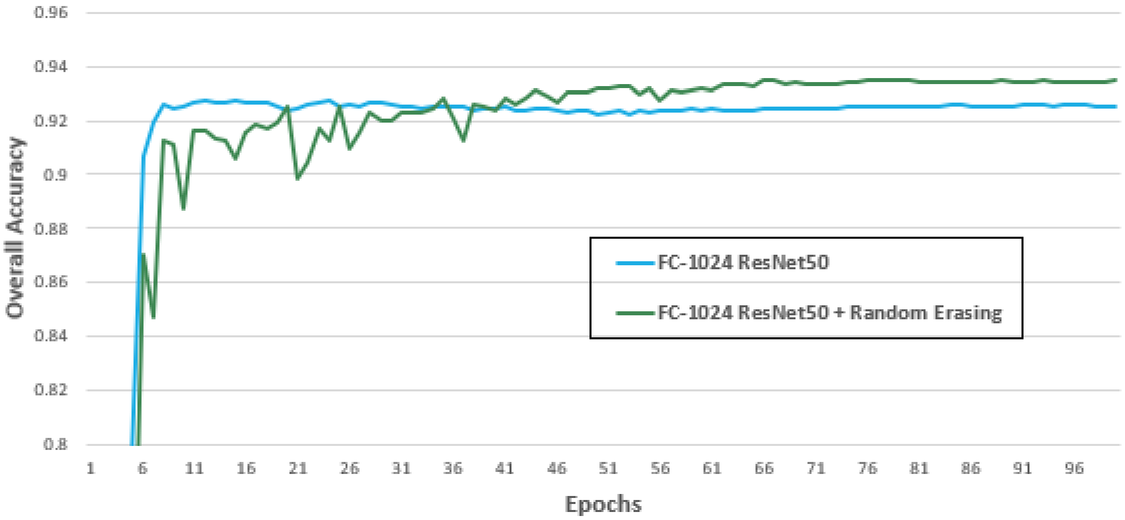


Figure 3. 4: FC-1024 ResNet50 vs FC-1024 ResNet50 + Random Erasing

After implementing data augmentation tactics, including random wiping, we observed a decrease in the network's scene classification performance following the training of the FC-1024 ResNet50 architecture. This contrasted with our previous approach of solely implementing random wiping on the training data. In our study, the term "all data augmentations" encompasses each of the methods previously stated. Surprisingly, employing all data augmentation techniques resulted in a modification of the input data distribution, which had an adverse effect on the model's ability to generalize on the test set (Figure 3.5).

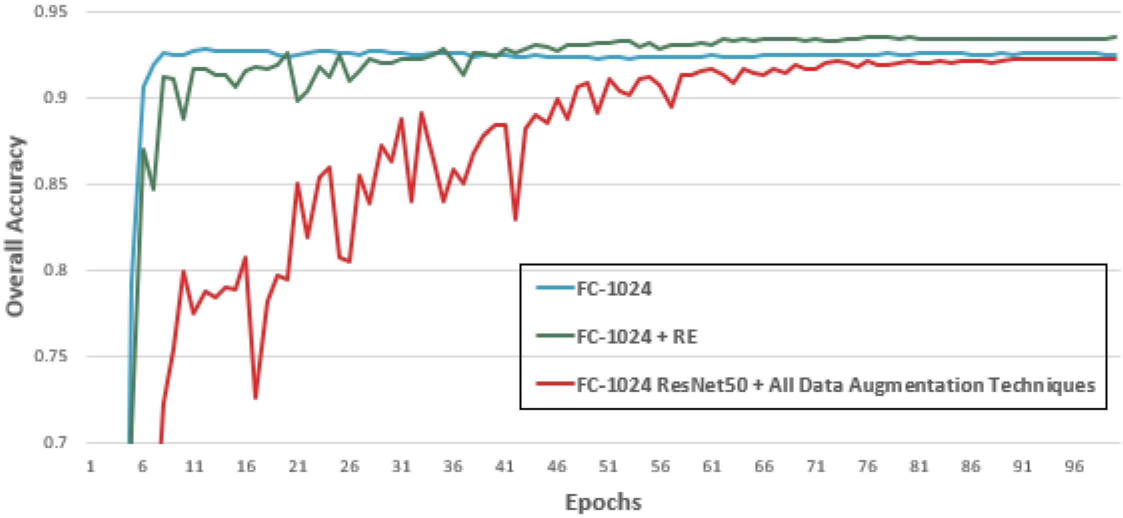


Figure 3. 5: Overall accuracy utilizing all data augmentation techniques

Sequentially, we incorporated one method at a time into the dataset, while simultaneously identifying and addressing this specific issue. Through this strategy, we effectively identified the data augmentation techniques, such as picture zooming, compression, and dimension shift, that led to the alteration in the data domain. Subsequently, we retrained the model by employing alternative methodologies, while deliberately eliminating these three data augmentation techniques. Figure 3.6 demonstrates that the test accuracy of the trained model has increased in comparison to the previously mentioned training techniques.

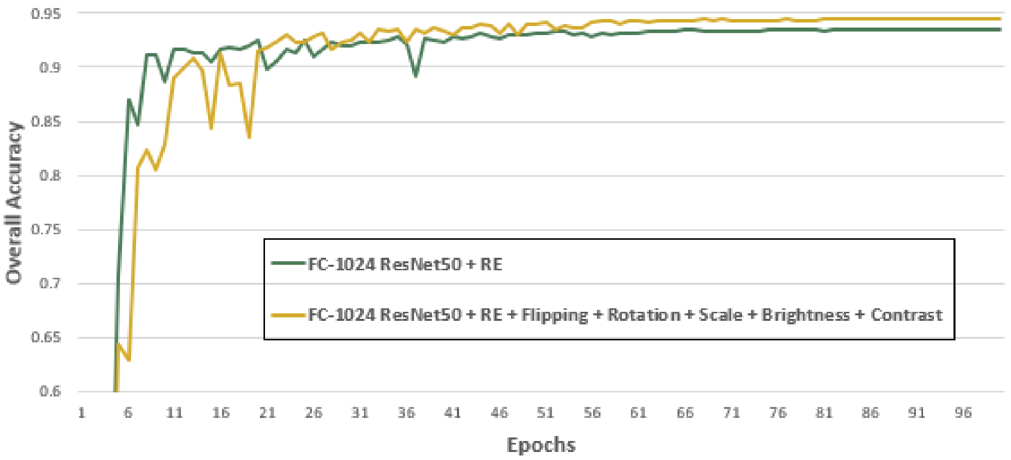


Figure 3. 6: Image classification accuracy using short-listed data augmentations

3.3.3 Semi Supervised Learning

This research had the task of utilizing a substantial dataset comprising of 63,765 photos without labels, in addition to 40,625 images with labels. The main objective was to improve the model's performance by leveraging both labelled and unlabelled data using semi-supervised learning techniques. The model was initially trained on the annotated images, which enabled it to learn and identify distinctive features and patterns. However, the genuine improvement in our methodology was our ability to integrate a substantial quantity of unannotated data to enhance the model's adaptability and accuracy.

When there is a lot of unlabelled data available—as exists in our dataset—semi-supervised learning is quite successful. We applied consistency regularization and pseudo-labelling among other semi-supervised learning methods [32]. Consistent regularization is the idea that the predictions of the model should stay constant even in cases of somewhat changed or transformed input data. Using this method, we motivated the model to produce consistent and reliable projections, so effectively leveraging the unlabelled data to acquire resilient traits. Pseudo-labelling, a fundamental element of our strategy, consisted of training the model on the labelled dataset and subsequently utilizing this trained model to produce pseudo-labels for the unlabelled images. The pseudo-labelled images were subsequently integrated into the training process, treating them as if they were actual labelled data. The iterative technique enabled the model to gradually enhance its predictions by using the extra data, without the need for manual annotation.

Apart from the above-described semi-supervised learning techniques, we also included self-learning strategies to improve the training process. Self-learning is a repeated process whereby the model trains, generates predictions on unlabelled data, then uses most confident predictions back into the training loop as fresh labelled data. This method uses the own predictions of the model to progressively improve its accuracy. We have built a robust system that efficiently uses the large unlabelled dataset by means of the merger of self-learning and semi-supervised learning. As shown in Fig. 3.2 that after semi-supervising, the data is now fed to the ResNet architectures for further processing as described below.

3.3.4 The Strategy We Chose towards ResNets

For the goal of classifying remote sensing photos, our research focused on examining the ResNet50 and ResNet101 models. Initially, we choose ResNet50 as our fundamental model for constructing the methodological framework. The ResNet50 layered structure is depicted in Figure 3.2 as a block diagram. It is important to emphasize some fundamental characteristics of the ResNet architecture:

- To minimize the number of trainable parameters and, consequently, reduce training time, the usage of fully connected layers (FCL) is minimized, with only a single softmax based FCL being employed.
- Due to the influence of the VGGNet framework, the maximum kernel size is restricted to 3x3.
- In contrast to traditional CNN architectures such as AlexNet and VGGNet, all convolution blocks, save for the one following the initial convolution layer, do not incorporate a pooling operation.
- Right after the last convolution block, a Global Average Pooling (GAP) layer is included.
- Every convolution block in the network undergoes training with batch normalization, along with global normalization of the data.

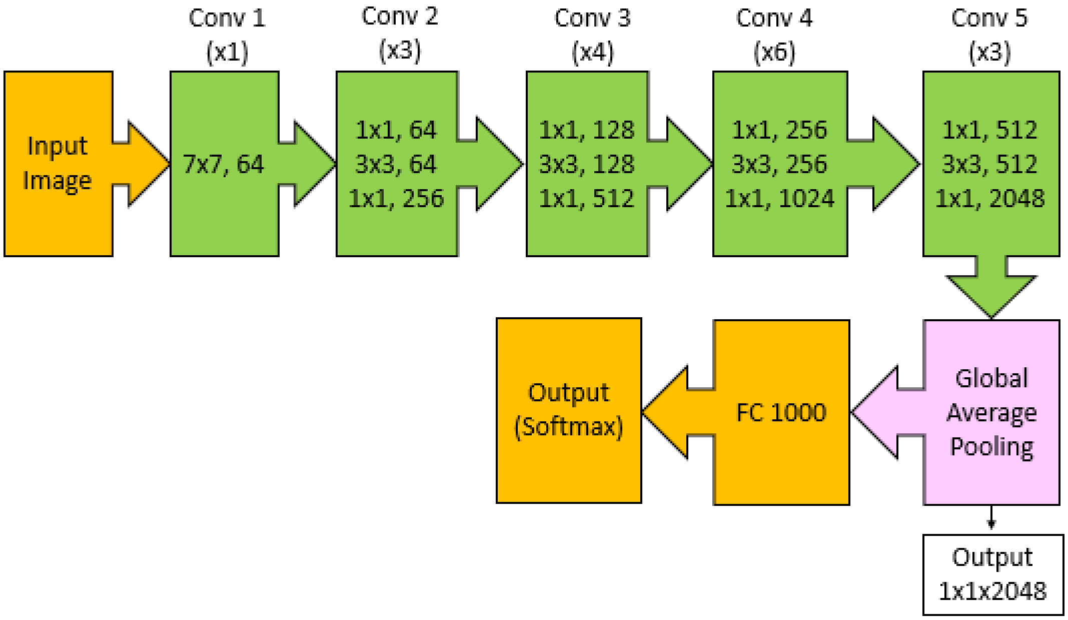


Figure 3. 7: ResNet50 architecture

The characteristics of the target dataset significantly influence the option to utilize fully connected layers in convolutional neural networks (CNNs). The Fully Connected Layer (FCL) encompasses the complete spatial receptive field of the picture, unlike the feature maps obtained from convolution layers, which have a defined receptive field based on the size of the kernel. An inherent drawback of utilizing FCL is the escalation in computational expenses that accompanies an augmentation in the quantity of training parameters. As an example, AlexNet consists of a total of 60 million parameters, with 58 million of these being specifically associated with fully connected layers (FCL). The VGGNet architecture contains a total of 138 million parameters, with FCL being responsible for 123 million of them [33]. As a result, the expensive computing expenses needed to include FCL have compelled the researchers to create alternate methods. Based on the information provided, the ResNet design solely comprises the Fully Connected Layer (FCL) of the network. The importance of this layer lies in its role of generating classification scores in the form of probabilities, which is accomplished by the utilization of softmax activation.

The size of the convolution kernel in CNNs is chosen solely by actual observations or by widely accepted conventions within the academic community. Researchers typically choose a kernel size that has demonstrated effectiveness in past studies for a dataset that is semantically like the data being studied. The creators of ResNet drew inspiration from the accomplishments of VGGNet in the task of image classification. They choose to utilize the identical 3x3 convolution kernel in multiple consecutive convolution blocks.

Pooling layers in convolutional neural networks (CNNs) are commonly used to reduce dimensionality and confine calculations within acceptable boundaries. ResNets have accomplished this goal by employing 1x1 convolutional kernels. The architecture utilizes a 1x1 bottleneck layer in each convolution block to limit the size of the input activation. This is followed by the application of a 3x3 convolution kernel to raise the dimensions again as the last step. 1x1 bottleneck layers have the advantage of learning important features, like a convolution layer, it offers the advantage of being adaptable in controlling the size of the output.

A Global Average Pooling (GAP) layer is added after the final convolution block in ResNets. This layer serves a dual function of decreasing the spatial dimensions and executing the flattening operation, which is typically used to determine the input dimensions for Fully Connected Layers (FCL). The primary objective of our work is to investigate an alternative viewpoint about the utilization of the GAP layer. The ultimate result of the last convolution block in the ResNet50 architecture, as shown in Figure 3.2, consists of 2048 activation maps. These activation maps correspond to the number of 1x1 kernels that constitute a constant component of the architecture. Irrespective of the initial input size to the network, when the Global Average Pooling (GAP) layer is applied to these 2048 activation maps, it results in a column vector with 2048 dimensions. In this case, there is no requirement to adjust the input image to match the FCL proportions, as demonstrated in VGGNet and AlexNet. The addition of the GAP (Global Average Pooling) layer in the ResNet architecture has rendered it independent of the input image size. The architecture first trained on a 224x224 input image can be further trained using input images of any size, such as 256x256 for LSCIDMR. This aids in the prevention of the loss of crucial feature data that is essential for the process of categorizing pictures.

Normalizing the input data to the network significantly accelerates the training of Convolutional Neural Networks (CNNs). Batch normalization is a technique that is used to normalize the input of the hidden layers of a network. When dealing with extremely deep networks, the input data distribution per mini batch may change as the weights are updated deeper into the network. As a result, the learning algorithm is compelled to adhere to a constantly changing objective, leading to issues with convergence. The term "internal covariate shift" refers to this phenomenon [34], [35].

3.3.5 Fine-tuning Pretrained Network vs Training from Scratch

All experiments in this study were conducted using Windows 11 Pro and trained on a PRO Z690-P DDR4 (MS-7D36) (16 GB) GPU. The software environment utilized for the experiment is Jupyter Notebook. The neural network was tested using input images from the LSCIDMR dataset, with dimensions of 256 × 256 pixels. Additionally, picture flip (both horizontally and vertically) and rotation augmentation techniques were implemented at the beginning of each training epoch. The system parameters and specifications used for training and testing are outlined in Table 1. The initial phase was choosing the suitable dimensions of images for the suggested model.

Table 3. 2: Model training parameters and system specifications.

Sr. No	Parameters	Value
1	Train/Test Ration	0.7/0.3
2	Train/Test Images	73,074/31,316
3	Epochs	100
4	Iterations/Epoch	2284
	Development Tools/Platform	
1	PRO Z690-P DDR4 (MS-7D36) (16 GB) GPU	
2	Jupyter Notebook	

3.4 Countering Overfitting Phenomenon

In the context of neural networks, overfitting is a common problem in which the model retains too many patterns and details from the training data, including noise and outliers. When the model is exposed to unexpected, unfamiliar data, it fails to generalize adequately. To address this issue, different regularization strategies are utilized, one of which is the implementation of the dropout layer. The dropout layer mitigates overfitting by stochastically discarding a portion of the neurons during each training iteration. This hinders the model from excessively depending on neurons and motivates it to acquire more resilient and universal characteristics.

Our neural network architecture incorporates a fully connected (fc) layer consisting of 1024 units. A fully connected layer, sometimes referred to as a dense layer, is characterized by each neuron being linked to every neuron in the preceding layer. This setting enables the model to acquire intricate representations of the input data. Nevertheless, because of its high level of interconnections, it is susceptible to overfitting, particularly when the number of units is substantial, such as 1024.

To tackle this issue, we incorporate a dropout layer within our network. During the training process, the dropout layer selectively deactivates a portion of the input units, setting them to zero. This thus keeps them out of the network during every learning cycle. Usually defined between the range of 0.2 to 0.5, this fraction is a hyperparameter. This suggests that every training iteration ignores between 20% to 50% of the neurons. This method forces the network to obtain duplicate representations of the data, hence improving its capacity to resist and lower its vulnerability to idiosyncrasies in the training set.

Within our fully connected layer, we incorporate the ReLU (Rectified Linear Unit) activation function alongside the dropout layer. ReLU is defined as the maximum of 0, x from its input: f(x) = This non-linear activation function gives the model non-linearity thereby enabling it to learn intricate patterns in the data. ReLU guarantees faster convergence during training and helps to reduce the vanishing gradient problem, so it is chosen over other activation functions including sigmoid or tanh.

ReLU activation function combined with a dropout layer with a fully connected layer generates a strong method to raise the generalization capacity of the neural network. While the dropout layer helps to minimize overfitting by making sure the model does not become too dependent on any one collection of neurons, the ReLU activation function guarantees that the model can capture complicated, non-linear interactions. This balanced approach makes the network a strong choice for different applications in deep learning since it helps it to function on fresh, unknown data.

Unlike the combination of GAP and dropout, including dropout with FCL has proven to be effective. This is because each neuron in FCL encompasses the core functionality of the GAP layer. Consequently, when training the network, the loss of specific neurons in FCL does not lead to a permanent loss of crucial feature information. Figure 3.8 displays the test set validation results for each of the designs.

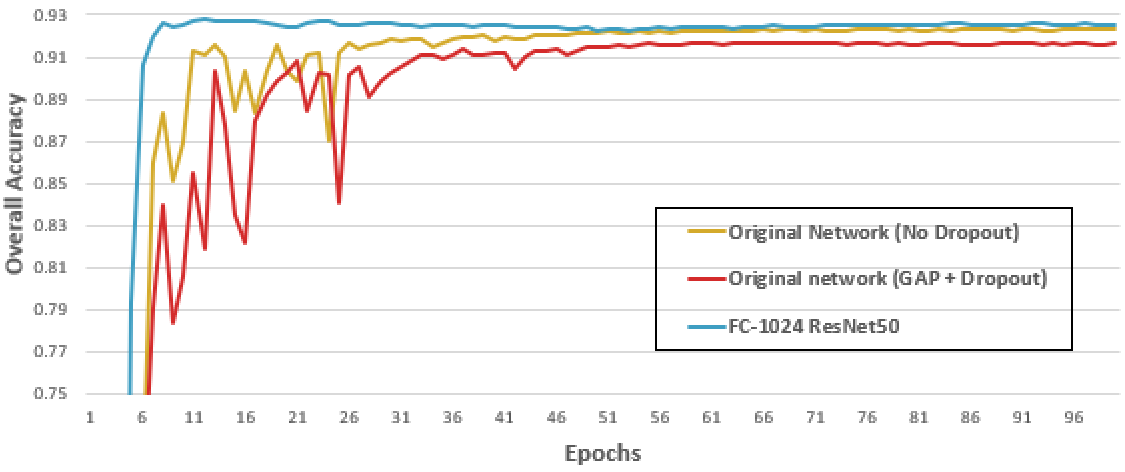


Figure 3. 8: Original network vs original network (GAP + dropout) vs FC-1024 ResNet50

CHAPTER 4: RESULTS AND COMPARISON

The statistical study of the fraction of photographs representing distinct weather systems during various seasons, as defined by the northern hemisphere, uncovers significant trends. of the winter months, there is a notable surge of photos depicting snowstorms and blizzards, which accurately represent the customary frigid and snowy circumstances of this season. In contrast, the summer months primarily exhibit visuals of thunderstorms and bright skies, which correspond to the hotter and more changeable climate encountered during this time. Spring imagery often features precipitation and emerging scenery, symbolizing the transitory character of this season. Autumn is distinguished by the sight of leaves descending and cloudy skies, representing the transition to colder weather and shorter daylight hours. The arrangement of photographs across the seasons emphasizes the clear and anticipated weather patterns linked to each specific time of the year in the northern hemisphere.

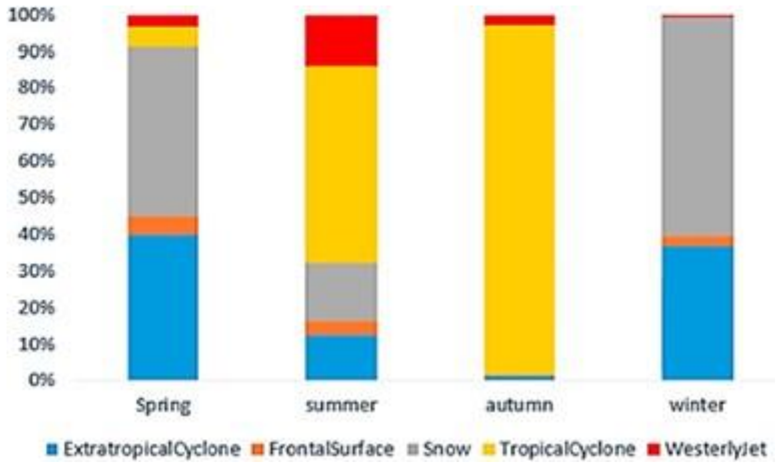


Figure 4. 1: A statistical analysis of the proportion of images of each weather system in different seasons is shown.

The graph shows a comparison of the seasonal distribution of distinct weather events, demonstrating how different atmospheric conditions are dominant at different times of the year. Spring, summer, fall, and winter are shown on the x-axis, with the percentage contribution of each meteorological phenomena within each season shown on the y-axis.

Partitioning each bar into its own category shows how each of six distinct meteorological events—a tropical cyclone, snow, frontal surface, westerly jet, and an unnamed sixth—contributes to the overall picture.

Extratropical cyclones account for approximately half of the seasonal meteorological conditions in the spring, making them the most significant weather occurrence of the season. These massive weather systems, common in the mid-latitudes and a major climatic factor in the area, appear to predominate in the spring. Indicative of continued winter conditions, snow is also a major component, making up about 35% of the meteorological events. During this transitional season, tropical cyclones and the westerly jet have limited influences, since frontal surfaces contribute minimally and the westerly jet appears small.

The prominent weather phenomena undergo a dramatic change during summer. The season's most common weather event, tropical cyclones, make up around 45% of all weather events, mirroring the average hurricane season in many parts of the world. The effect of the snow is little as its presence is greatly reduced. As a result, extratropical cyclones take up less space in the summer than they do in the spring. Their spring levels are affected in a similar way by frontal surfaces and the westerly jet, suggesting that these phenomena are seasonal.

Tropical storms account for almost 80% of the weather events that occur in the fall. The hurricane season is when most severe weather occurs, dominating the weather forecast for the rest of the year. Frontal surfaces and the westerly jet contribute very little to the general weather pattern, whereas extratropical cyclones and snow have minor impact. The enormous climatic impact of tropical cyclones throughout fall is highlighted by their dominance.

Around 60% of the meteorological events that occur in winter are snowfalls. With the season's typical cold weather and heavy snowfall, this is to be expected. The role of extratropical cyclones, which make up around 35% of the weather, is still substantial. Winter weather patterns are mostly unaffected by frontal surfaces, tropical cyclones, and the westerly jet, which are practically nonexistent.

In general, the graph does a good job of showcasing how the most common weather events change from season to season. When the weather gets colder, extratropical cyclones and snow tend to occur more often, while tropical cyclones tend to be more common in the warmer months. This distribution highlights how weather conditions are constantly changing and how they are affected by the seasons.

The trials in this study were conducted using Windows 11 Pro and trained on a PRO Z690-P DDR4 (MS-7D36) (16 GB) GPU. The software environment used for the experiment is Jupyter Notebook. The neural network was tested using input images from the LSCIDMR dataset, which had a size of 256 × 256 pixels. The training epochs were capped at a maximum of 100, and the batch size for training was fixed at 32. In addition, image flipping (both horizontally and vertically) and rotation augmentation techniques were applied at the start of each training epoch. The system characteristics and specifications utilized for both training and testing are clearly stated in Table 4.1. The first step was selecting the appropriate dimensions of the photos for the proposed model.

Table 4. 1: Model training parameters and system specifications.

Sr. No	Parameters	Value
1	Train/Test Ration	0.7/0.3
2	Train/Test Images	73,074/31,316
3	Epochs	100
4	Iterations/Epoch	2284
5	Batch Size	32
6	Dropout Layer	0.2
	Development Tools/Platform	
7	PRO Z690-P DDR4 (MS-7D36) (16 GB) GPU	
8	Jupyter Notebook	

The table provides a comprehensive overview of the parameters and technologies employed in a machine learning project. The table contains entries that describe various aspects of the setup and configuration. These entries provide valuable information on the environment and methods used.

The initial row displays the Train/Test Ratio, which has been established as 0.7/0.3. This indicates that 70% of the available data is utilized for training the model, while the remaining 30% is set aside for testing. The aforementioned ratio is crucial for assessing the model's efficacy on unfamiliar data and guaranteeing its ability to generalize effectively.

The second row displays the quantity of photos utilized for both training and testing purposes. More precisely, there are a total of 73,074 photos available for training purposes and 31,316 images designated for testing. The significant quantity of this dataset indicates that the research intends to utilize a huge amount of data in order to build a strong and precise model.

The third row indicates the number of epochs, which is set at 100. An epoch refers to a single iteration over the full training dataset. Training the model for 100 epochs involves iterating the dataset through the network 100 times, enabling the model to learn and adapt its parameters incrementally.

The fourth row specifies the number of iterations per epoch, which is fixed at 2284. Each epoch has 2284 iterations, with each iteration representing a single update of the model's parameters using a batch of training data. The batch size, as specified in the fifth row, is 32. Consequently, in each iteration, the model handles 32 images simultaneously, achieving a harmonious compromise between computing efficiency and convergence speed.

The sixth row specifies the presence of a dropout layer with a dropout rate of 0.2. Dropout is a regularization method employed to mitigate overfitting by randomly deactivating a portion of input units to zero throughout the training process. A dropout rate of 0.2 indicates that 20% of the neurons are excluded after each update cycle in the training phase, which enhances the model's resilience and ability to generalize.

The seventh row enumerates the development tools and platform utilized. It specifically refers to the PRO Z690-P DDR4 (MS-7D36) motherboard, which includes a 16 GB graphics processing unit (GPU). This denotes the hardware setup used, emphasizing the utilization of a high-performance graphics processing unit (GPU) that is essential for speeding up the computationally demanding operations associated with training deep learning models.

The eighth and last row specifies that Jupyter Notebook is the utilized development environment. Jupyter Notebook is a widely used open-source web tool that enables users to generate and distribute documents including executable code, equations, visualizations, and descriptive prose. The utilization of this tool signifies a dynamic and adaptable setting for the creation and exploration of machine learning models.

In summary, this table presents a concise summary of the essential components and parameters of the machine learning project, offering a thorough outline of the dataset, training setup, and software tools utilized.

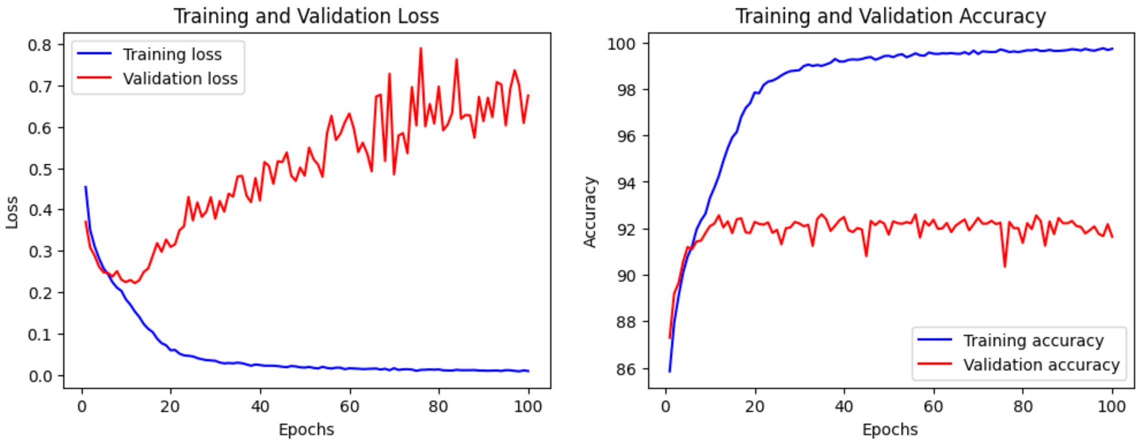


Figure 4. 2: After 100 epochs. (a) Training and Validation Loss (b) Training and Validation Accuracy

During the training and validation stages of a machine learning model, the performance metrics are shown in the graphs. The two main measures shown are accuracy and loss, which are tracked over 100 epochs.

The training and confirmation loss is shown in the first graph on the left. Loss is a way to figure out how close the model's predictions are to the real goal values. A smaller loss means that the model works better. At the beginning, both the training and validation losses are high. This is normal because the model hasn't learned from the data yet. The training loss goes down quickly as training goes on, which means the model is learning to fit the training data better. This decrease keeps going smoothly and hits a very low value near the end of the epochs. This shows that the model correctly matched the training data.

But the approval loss acts in a different way. At first, it goes down, which suggests that the model is learning and getting better at using the test data as well. However, once it reaches a certain point, the validity loss starts to rise and changes a lot over the next few epochs. This trend shows that the model is overfitting, which means that it works well on the training data but not on the validation data. The fact that the training loss is going down while the confirmation loss is going up is a clear sign of this problem. Overfitting usually happens when the model is too complicated and picks up noise and features in the training data that don't show up in new data.

The accuracy for training and confirmation is shown in the second graph on the right. Accuracy is a measure of how many of the model's guesses were right out of all of them. Like the loss graph, the training accuracy starts out low and quickly rises until it reaches a peak close to 100%. This shows that as training goes on, the model gets much better at guessing the right labels for the training data. The sharp rise and following stabilization show that the training set was learned and adapted well.

Confirmation accuracy, on the other hand, shows a different trend. It goes up at first, which means that the model is getting better at handling data it hasn't seen yet. But after a certain number of epochs, the validation accuracy levels off and even goes down a little, ending up at a value that is much lower than the training accuracy. This difference in accuracy between training and confirmation adds to the evidence of overfitting. The model gets better at guessing the training data, but it doesn't stay as good at it on the validation data, which shows that it can't be used in other situations.

These graphs give us useful information about how the model learns. The model is successfully fitting the training data to its parameters as shown by the sharp drop in training loss and the associated rise in training accuracy. But the fact that validation loss is going up while validation accuracy stays the same shows that the model is remembering the training data, even the noise and oddities in it, instead of learning the trends that would help it do better with new data.

Several options can be thought of to deal with the overfitting that was seen. One popular method is to use regularization techniques, like L1 or L2 regularization, which make the loss function less accurate when weights are too high. This makes models simpler and less likely to overfit. Dropout is another way. It randomly sets a portion of the input units to zero at each update during training time. This keeps the model from relying too much on any one node, which helps it generalize.

Overfitting can also be avoided by making the model simpler by cutting down on the number of levels or units. Another important factor is making sure there is enough training data. More data can give a better picture of the problem space and help the model learn trends that can be used in more situations. Data augmentation methods, which use random transformations to add more training examples, can also be helpful in this case.

Another good method is to stop the training process as soon as the validation loss stops getting better. This stops the model from learning from the training data any further and stops it from becoming too well-fitted. Cross-validation, in which the training process is done more than once with different groups of data, can help you understand how well the model works and make sure it works well in other situations.

A hyperparameter called the model's learning rate, which controls the step size during optimization, is also very important. If the learning rate is too high, the model might find a less-than-ideal solution too fast. On the other hand, if the learning rate is too low, the training process could take too long and the model could become too well-fitted. Because of this, the learning rate needs to be carefully tuned.

Finally, it's important to make sure that the problem space is reflected in both the training and evaluation data. Overfitting can happen if these datasets are biased in any way. If the training data has too many examples that are too similar, for example, the model might learn how to do well on those examples but not be able to apply what it has learned to other examples. Making sure that the training data is diverse can help you build a strong model.

To sum up, the plots give a thorough look at how well the model did in both training and testing. The model is learning to guess the training data because the training loss is going down and the training accuracy is going up. But the fact that validation loss is going up while accuracy stays the same shows a big problem called overfitting, which happens when the model doesn't work as well on new data as it did on training data. To fix this overfitting problem, you need to use a number of methods, such as dropout, regularization, model simplification, data augmentation, early stopping, cross-validation, learning rate tuning, and making sure the training data is varied. These techniques can help the model do better on new data it hasn't seen before and at generalizing what it knows.

4.1 Model Analysis

For the proposed network design, we considered ResNet50 and ResNet101 as the initial models. The feasibility of this was achieved by the shared underlying structure of the ResNet variants, with the only difference being the number of convolution blocks, which determines the depth of the network. Moreover, 70% of the LSCIDMR was utilized for training purposes. An increase in accuracy was noticed after switching the baseline design from ResNet50 to ResNet101. This can be attributed to the greater capacity of the deeper network to acquire more detailed and informative features. The accuracy statistics acquired for a given target dataset may not necessarily have an inverse relationship with the depth of the network. Learning the hidden complex features has a saturation threshold. Going beyond this threshold by increasing the depth of the network may result in reduced accuracy. This occurs when the network becomes excessively intelligent and begins to overfit on the training dataset rather than acquiring intricate features. The main catalyst for a substantial improvement in performance for Resnet101 was this.

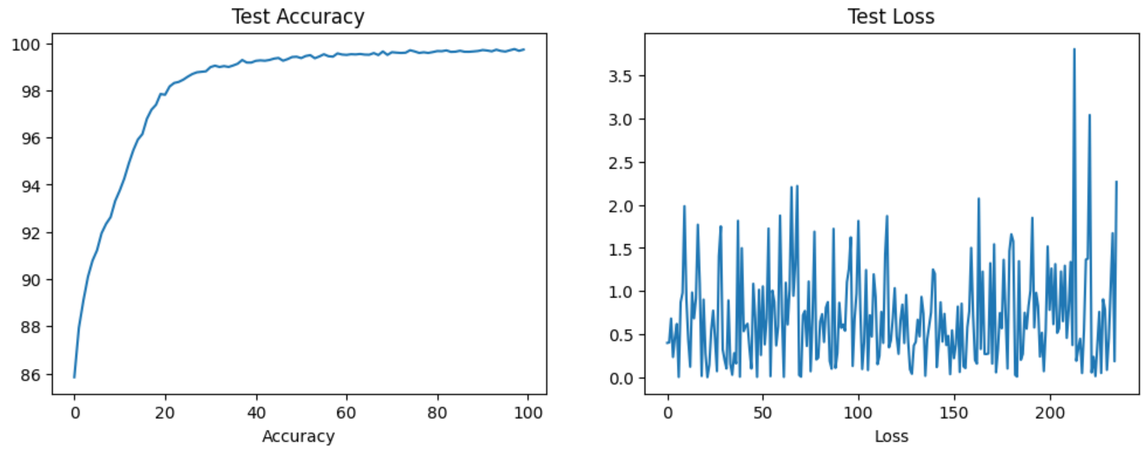


Figure 4. 3: Output

Two plots side by side make up the graph shown: the one on the left is marked "Test Accuracy," and the one on the right "Test Loss." Presumably over several iterations or epochs, both graphs show the performance of a machine learning model.

Starting first on the "Test Accuracy" graph, it shows accuracy against some arbitrary metric running down the x-axis from 0 to 100. This x-axis most certainly shows epochs, repetitions, or some kind of cumulative count. As shown, the accuracy begins the training process at about 86% and rises quickly in the first phase. This fast ascent implies that during the early epochs the model rapidly learns the basic trends in the data. As one moves forward, the curve gets more gradual, suggesting a slowing down in accuracy. Typical in training machine learning models, this trend shows early on significant performance gains but makes incremental improvements more difficult as the model adjusts its knowledge of the data. With just slight changes or oscillations in accuracy while training goes, the graph plateaus eventually indicate that the model's accuracy reaches a high point near 100%, therefore implying very great performance. This plateau implies the model has learnt the data sufficiently well; any more training might not appreciably improve its performance.

Turning now to the "Test Loss" graph on the right, it charts loss values against presumably the same x-axis criteria as the accuracy graph. In machine learning, loss usually describes the inaccuracy in the predictions of the model; lower values indicate improved performance. The accuracy graph shows a smooth, rising trend; the loss graph shows a far more random pattern. Over the course of the training, the values vary greatly, spikes and troughs mark times of greater and less loss. Several factors might cause this fluctuation in loss, including the model running distinct batches of data that might be more or less difficult or the optimization method making more or less significant changes during particular rounds. Notwithstanding these variations, the overall pattern seems to point to numerous possible problems since the loss does not regularly drop with time. The model can be overfitting the training data, performing quite well on it while failing to generalize as successfully on the test data. Given the great accuracy and irregular loss, this scenario is reasonable where the model learns to predict the training data too well but struggles with unseen data.

Another reading of the fluctuating loss graph could be connected to the type of data or the training procedure itself. Should the data be particularly noisy or erratic, the loss can represent this natural variance. On the other hand, if the learning rate of the model is set too high, the optimization process may be making too significant changes that result in the loss jumping erratically instead of settling gently to a lower number. This theory fits the occasional spikes in the loss graph. Changing the learning rate or using learning rate annealing could help to reduce such behavior.

Furthermore, the differences between the erratic loss curve and the smooth accuracy curve could suggest possible problems with the evaluation criteria or the loss function applied. While connected, accuracy and loss highlight various facets of model performance. While loss offers a more complex picture of prediction errors, accuracy is a straight indication of the proportion of accurate forecasts. Under particular conditions, the model can attain great accuracy with sporadic large errors—that would show up as spikes in the loss graph. This could imply the need to check the loss function to make sure it fits the targeted performance goals.

These two graphs taken together offer a thorough understanding of the performance of the model. Suggesting the model efficiently learns and forecasts the data patterns, the "Test Accuracy" graph clearly demonstrates a tendency of progress and stabilization at a high level of accuracy. Conversely, the "Test Loss" graph points to problems such overfitting, learning rate settings, or natural data variability, therefore suggesting possible volatility in model performance. Refining the model further and guaranteeing it generalizes effectively to new, unknown data depends on an awareness of and resolution of these differences between accuracy and loss.

4.2 Comparison with Existing Approaches in Literature

We assessed the precision outcomes of the suggested model utilizing the ResNet framework, in contrast to cutting-edge deep learning techniques for RSIC tasks. The proposed architecture has surpassed all existing optimal methods, setting a new standard with an overall accuracy of 91.78%.

Table 4. 2: Accuracy comparison with recent methods

Methods	Precision	Accuracy	F1 Score
Xception [36]	0.9022	0.9022	0.8351
MobileNetV2 [36]	0.8351	0.8351	0.8351
VGG16 [37]	0.86	0.87	0.90
AlexNet [38]	0.8874	0.88	0.87
Proposed	0.917	0.917	0.93

The table provides a comparative study of various approaches employed in a specific application, possibly pertaining to machine learning or computer vision, based on the context of the listed methods. The table has five methods: Xception, MobileNetV2, VGG16, AlexNet, and a novel technique. The evaluation of these methods is conducted using three metrics: Precision, Accuracy, and F1 Score.

The initial technique mentioned is Xception, which exhibits a precision of 0.9022, an accuracy of 0.9022, and an F1 score of 0.8351. The values demonstrate that Xception performs admirably in all parameters, exhibiting a notable level of precision and accuracy. However, it has a slightly lower F1 score, indicating a small compromise in maintaining a balance between precision and recall.

The MobileNetV2 model achieved a precision and F1 score of 0.8351, along with an accuracy of 0.8351. The uniformity of these metrics indicates that MobileNetV2 exhibits a well-rounded performance, albeit slightly inferior to Xception in terms of precision and accuracy. This suggests that MobileNetV2 may be a more versatile model, but with a lower maximum performance level.

VGG16, an alternative approach, demonstrates a precision of 0.86, an accuracy of 0.87, and an F1 score of 0.90. This approach exhibits strong performance, demonstrating a slightly superior F1 score in comparison to its precision and accuracy. The higher F1 score suggests that VGG16 is more proficient in managing imbalanced datasets by effectively maintaining a balance between precision and recall.

AlexNet achieves a precision of 0.8874, an accuracy of 0.88, and an F1 score of 0.87. The performance metrics of AlexNet indicate that it exhibits a relatively high level of precision, slightly surpassing its accuracy and F1 score. This suggests that it may have a greater ability to minimize false positives.

The proposed approach demonstrates superior performance compared to the other methods described, achieving a precision of 0.917, an accuracy of 0.917, and an F1 score of 0.93. The results demonstrate the exceptional performance of the suggested approach, especially in relation to the F1 score, indicating its high efficacy in achieving a balance between precision and recall. This makes it a reliable and suitable choice for the current application.

To summarize, the table presents a concise comparison of several methods, indicating that both Xception and the proposed method exhibit the highest levels of accuracy and precision. However, the suggested method outperforms the others with the highest F1 score, suggesting its superior performance and reliability in real-world scenarios.

The below graph illustrates the comparative performance of various algorithms.

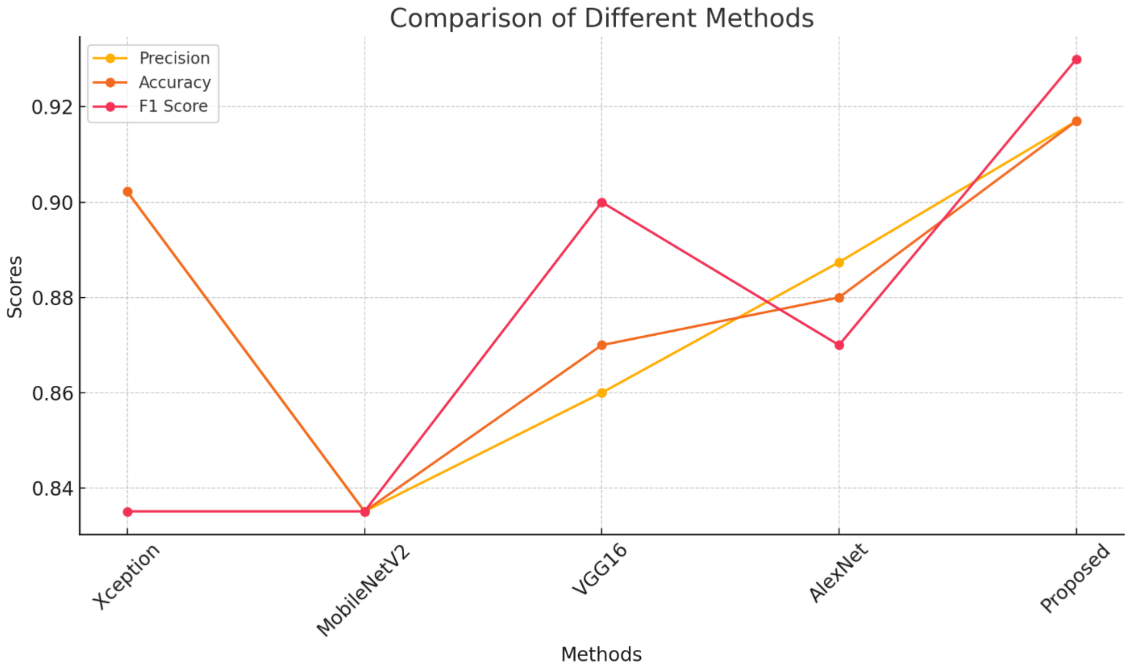


Figure 4. 4: Comparison with Existing Techniques

The graph provides a comparative comparison of various methodologies using three performance metrics: Precision, Accuracy, and F1 Score. The evaluated methods include Xception, MobileNetV2, VGG16, AlexNet, and a Proposed approach. The performance of each approach is represented by three separate lines, color-coded orange for Precision, yellow for Accuracy, and pink for F1 Score.

The Xception approach exhibits a Precision score of roughly 0.90, with lower values for Accuracy and F1 Score, both around 0.84. This implies that although Xception has a high level of accuracy in precisely recognizing pertinent occurrences, it may not consistently maintain a satisfactory level of performance and equilibrium between precision and recall.

However, MobileNetV2 exhibits a substantial decrease in Precision and F1 Score, with both metrics declining to approximately 0.84, suggesting a notable deficiency in its performance. The Accuracy of this method is significantly poor, nearly approaching 0, indicating significant shortcomings compared to other methods.

VGG16 outperforms MobileNetV2, exhibiting a significant enhancement across all three criteria. The alignment between Precision and Accuracy is strong, with a value slightly above 0.86, although the F1 Score has a higher value, approximately 0.89. This suggests a more equitable and efficient performance, especially in maintaining a favorable balance between accuracy and completeness.

AlexNet demonstrates a continued rising trajectory, including additional enhancements. The Precision and Accuracy ratings of this method both exceed 0.88, while its F1 Score, which is nearly 0.90, is the highest among these methods. This indicates that AlexNet demonstrates a high level of effectiveness in classification tasks, effectively achieving a balance between precision and recall.

Ultimately, the Proposed technique surpasses all other methods in terms of all three measures. The precision and accuracy values are approximately 0.91, but the F1 score surpasses 0.92, making it the highest. This suggests that the Proposed technique not only preserves a high level of precision and accuracy, but also attains the optimal equilibrium between precision and recall, leading to the greatest F1 Score.

In summary, the graph clearly illustrates that the Proposed approach outperforms the other established ways, indicating its effectiveness in achieving high levels of precision, accuracy, and a well-balanced F1 Score.

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This work proposes the use of a novel ResNet101 model for the classification of remote sensing satellite photos. To achieve the best performance in RSIC, it is recommended to use ResNets in combination with a network called FC-1024. This network should feature ensembling and augmentation techniques. The primary obstacle to the use of deep neural networks in situations when there is insufficient training data is the issue of overfitting. Given this, the proposed network was carefully constructed through a sequence of experiments to provide the highest level of protection against overfitting and, as a result, improve the network's ability to generalize. The model effectively resolved the issue of inter-class similarity by making necessary modifications to the architectural design. This allowed it to gain a deeper understanding of the specific details and global attributes associated with each class. By improving and training on the demanding LSCIDMR satellite imaging dataset, the proposed ResNet model obtained picture classification outcomes that were on par with the most cutting-edge techniques currently available.

The suggested study presents a novel method that utilizes a ResNet101 model specifically tailored for classifying remote sensing satellite images. ResNet101 is a deep convolutional neural network that excels at image classification tasks by efficiently addressing the challenges associated with complex images. It achieves this by employing residual learning to mitigate the vanishing gradient problem commonly observed in deep networks. The innovation of this strategy is not only the utilization of ResNet101 but also its strategic integration with another network known as FC-1024. This network plays a significant role in improving the model's capability by using ensembling and augmentation approaches. These techniques are essential for enhancing the performance of remote sensing image classification (RSIC).

One of the main obstacles in using deep neural networks, especially when there is a little amount of training data, is the possibility of overfitting. Overfitting is the phenomenon when a model becomes too focused on the noise and specific aspects in the training dataset, causing its performance to suffer when applied to new, unknown data. In order to address this problem, the suggested network was carefully constructed through a sequence of experiments with the goal of optimizing its ability to resist overfitting. This research focused on optimizing the architectural design of the network to ensure its ability to effectively apply learned knowledge to various data samples. The meticulous development procedure enhances the model's capacity to generate precise predictions on fresh data, hence improving its overall ability to generalize.

Furthermore, the suggested model tackles the considerable obstacle of inter-class similarity, a prevalent problem in the classification of remote sensing images where distinct classes may exhibit comparable visual characteristics. In order to address this issue, precise alterations were implemented to the architectural structure of the network, enabling it to acquire a deeper comprehension of the intricate particulars and overall characteristics linked to each category. These alterations enhance the model's ability to discern more efficiently across classes that exhibit nuanced distinctions, hence enhancing classification precision.

The efficacy of the suggested ResNet model was further showcased by its successful implementation on the demanding LSCIDMR satellite imaging dataset. By integrating sophisticated methodologies and enhancements in both the network architecture and the training procedure, the model attained picture classification outcomes that are on par with the most cutting-edge strategies presently accessible in the domain. The suggested approach has the ability to establish new standards in classifying remote sensing satellite photographs. This is achieved by utilizing the strengths of ResNet101 and FC-1024 within a well-optimized and experimentally verified framework.

2.8 Recommendations for Future Work

This paper proposes the utilization of the ResNet model for the classification of remote sensing photos. The subsequent recommendations are proposed to delineate the trajectory of future research in view of this study:

- When dealing with deep networks, it is possible to explore different tactics, learning methods, or combinations to address the problem of overfitting.
- The issue of inter-class similarity in RSIC remains unresolved and requires attention. Further exploration can be done to enhance the feature representations by making appropriate alterations to the architecture and parameters.
- This analysis was carried out using a solitary GPU configuration. The computational cost study can be conducted by utilizing multiple GPU setups.
- One can undertake thorough hyperparameter optimization by utilizing methods, such as grid search, random search, or Bayesian optimization to finely adjust the model for optimal performance.
- It is possible to look into cloud-based or edge computing deployment methods to make the model usable and effective in a range of settings, for instance as those with low latency applications.

REFERENCES

[1] &. A. A. Igor Ševo, "Convolutional Neural Network Based Automatic Object Detection on Aerial Images," *IEEE Geoscience and Remote Sensing Letters* , vol. 13, no. 5, pp. 740 - 744, 2016.

[2] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.

[3] &. B. T. N. Dalal, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA , 2005.

[4] L. Z. &. B. D. Liangpei Zhang, "Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art," *IEEE Geoscience and Remote Sensing Magazine* , vol. 4, no. 2, pp. 22 - 40, 2016.

[5] D. T. L. M. G.-S. X. L. Z. F. X. &. F. F. Xiao Xiang Zhu, "Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8 - 36, 2017.

[6] K. N. J. A. d. S. Otávio A. B. Penatti, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Boston, MA, USA , 2015.

[7] G.-S. X. J. H. &. L. Z. Fan Hu, "Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery," *Remote Sensing,* vol. 7, no. 11, pp. 14680-14707, 2015.

[8] W. D. R. S. L.-J. L. K. L. &. L. F.-F. Jia Deng, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA , 2009.

[9] G. P. C. S. &. L. V. Marco Castelluccio, "Land Use Classification in Remote Sensing Images by Convolutional Neural Networks," *arXiv:1508.00092,* 2015.

[10] M. T. E. P. &. A. B. Pradeep Hewage, "Deep learning-based effective fine-grained weather forecasting model," *Pattern Analysis and Applications* , vol. 24, p. 343–366, 2020.

[11] J. Booz, W. Yu, G. Xu, D. Griffith and N. Golmie, "A Deep Learning-Based Weather Forecast System for Data Volume and Recency Analysis," in *2019 International*

Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 2019.

[12] K. K. P. Pushpa Mohan, "Deep Learning Based Weighted SOM to Forecast Weather and Crop Prediction," *Internation Journal of Intelligent Enineering & Systems,* vol. 11, no. 4, 2018.

[13] D. B. K. C. R. P. A. M. K. V. K. E. R. &. B. S. R. Meenal, "Weather Forecasting for Renewable Energy System: A Review," *Archives of Computational Methods in Engineering ,* vol. 29, p. 2875–2891, 2022.

[14] A. Utku and Ü. Can, "Deep Learning Based Effective Weather Prediction Model for Tunceli City," in *2021 6th International Conference on Computer Science and Engineering (UBMK)*, Ankara, Turkey, 2021.

[15] P. Du, "Ensemble Machine Learning-Based Wind Forecasting to Combine NWP Output With Data From Weather Station," *IEEE Transactions on Sustainable Energy* , vol. 10, no. 4, pp. 2133 - 2141, 2019.

[16] B. K. S. R. S. S. G. R. C. R. S. N. D. N. Manmeet Singh, "Deep learning for improved global precipitation in numerical weather prediction systems," *Atmospheric and Oceanic Physics,* p. arXiv:2106.12045, 2021.

[17] A. M. Abdalla, I. H. Ghaith and A. A. Tamimi, "Deep Learning Weather Forecasting Techniques: Literature Survey," in *2021 International Conference on Information Technology (ICIT)*, Amman, Jordan, 2021.

[18] L.-W. Kang, K.-L. Chou and R.-H. Fu, "Deep Learning-Based Weather Image Recognition," in *2018 International Symposium on Computer, Consumer and Control (IS3C)*, Taichung, Taiwan, 2018.

[19] C. Bai, M. Zhang, J. Zhang, J. Zheng and S. Chen, "LSCIDMR: Large-Scale Satellite Cloud Image Database for Meteorological Research," *IEEE Transactions on Cybernetics,* vol. 52, no. 11, pp. 12538 - 12550, 2021.

[20] C. Cheng, Y. Cao, H. Yu, Q. Zhang and X. Cheng, "A Study on Meteorological Recognition With a Fusion Enhanced Model Based on EVA02 and LinearSVC," *IEEE Access* , vol. 12, pp. 28784 - 28797, 2024.

[21] J. Z. Tan, J. Y. Lim, K. M. Lim and C. P. Lee, "Weather Image Recognition Using Vision Transformer," in *2023 IEEE 11th Conference on Systems, Process & Control (ICSPC)*, Malacca, Malaysia, 2024.

[22] M. M. Dhananjaya, V. R. Kumar and S. Yogamani, "Weather and Light Level Classification for Autonomous Driving: Dataset, Baseline and Active Learning," in

2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 2021.

[23] G. Saini, T. Munasinghe and K. Pan, "Image Classification to identify Transverse Cirrus Band Clouds using Convolutional Neural Networks," in *2021 IEEE International Conference on Big Data (Big Data),* Orlando, FL, USA, 2021.

[24] E. A. &. Y. G. Muhammet Ali Dede, "Deep Network Ensembles for Aerial Scene Classification," *IEEE Geoscience and Remote Sensing Letters,* vol. 16, no. 5, pp. 732 - 735, 2019.

[25] R. Minetto and &. S. S. Maurício Pamplona Segundo, "Hydra: An Ensemble of Convolutional Neural Networks for Geospatial Land Classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 57, no. 9, pp. 6530 - 6541, 2019.

[26] P. S. &. P. F. Y. Bengio, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks,* vol. 5, no. 2, pp. 157 - 166, 1994.

[27] X. G. &. Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings., 2010.

[28] X. Z. S. R. &. J. S. Kaiming He, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA , 2016.

[29] Y. B. a. A. C. I. Goodfellow, "Deep Learning (Adaptive Computation and Machine Learning series)," ed: e MIT Press, Cambridge, England, 2016.

[30] &. T. M. K. Connor Shorten, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data* , vol. 6, no. 1, pp. 1-48, 2019.

[31] L. Z. G. K. S. L. Y. Y. Zhun Zhong, "Random Erasing Data Augmentation," *Computer Vision and Pattern Recognition*, vol. 34, no. 07, pp. 13001-13008, 2017.

[32] L. I. Kuncheva, Combining pattern classifiers: methods and algorithms, John Wiley & Sons, 2014.

[33] S. R. D. V. P. S. M. S.H. Shabbeer Basha, "Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification," *arXiv:1902.02771*, vol. 378, pp. 112-119, 2020.

[34] M. Awais, M. T. B. Iqbal and S.-H. Bae, "Revisiting Internal Covariate Shift for Batch Normalization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 11, pp. 5082 - 5092, 2021.

[35] C. S. Sergey Ioffe, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning*, PMLR, 2015.

[36] M. Z. J. Z. J. Z. S. C. Cong Bai, "LSCIDMR: Large-Scale Satellite Cloud Image Database for Meteorological Research," *IEEE Transactions on Cybernetics* , vol. 52, no. 11, pp. 12538 - 12550, 2021.

[37] S. F. K. Mohammad Farid Naufal, "Weather image classification using convolutional neural network with transfer learning," in *INTERNATIONAL CONFERENCE ON INFORMATICS, TECHNOLOGY, AND ENGINEERING 2021 (InCITE 2021): Leveraging Smart Engineering*, Surabaya, Indonesia, 2022.

[38] P. B, K. Maharajan and R. Srikanteswara, "Deep Learning Based Image Classification Using Small VGG Net Architecture," in *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, Mysuru, India , 2022.

[39] M. T. E. P. A. B. Pradeep Hewage, "Deep learning-based effective fine-grained weather forecasting model," *Pattern Analysis & Applications,* vol. 24, no. 1, pp. 343 - 366, 2021.

[40] J. Booz, W. Yu, G. Xu, D. Griffith and N. Golmie, "A Deep Learning-Based Weather Forecast System for Data Volume and Recency Analysis," in *2019 International*

Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 2019.

[41] K. P. Pushpa Mohan, "Deep Learning Based Weighted SOM to Forecast Weather and Crop Prediction for Agriculture Application," *International Journal of Intelligent Engineering & Systems,* vol. 11, no. 4, 2018.

[42] D. B. K. C. R. P. A. M. K. V. K. E. R. &. B. S. R. Meenal, "Weather Forecasting for Renewable Energy System: A Review," *Archives of Computational Methods in Engineering,* vol. 29, p. 2875–2891, 2022.

[43] A. Utku and Ü. Can, "Deep Learning Based Effective Weather Prediction Model for Tunceli City," in *2021 6th International Conference on Computer Science and Engineering (UBMK)*, Ankara, Turkey, 2021.

[44] P. Du, "Ensemble Machine Learning-Based Wind Forecasting to Combine NWP Output With Data From Weather Station," *IEEE Transactions on Sustainable Energy,* vol. 10, no. 4, pp. 2133 - 2141, 2019.

[45] B. K. S. R. S. S. G. R. C. R. S. N. D. N. Manmeet Singh, "Deep learning for improved global precipitation in numerical weather prediction systems," *arXiv:2106.12045,* 2021.

[46] A. M. Abdalla, I. H. Ghaith and A. A. Tamimi, "Deep Learning Weather Forecasting Techniques: Literature Survey," in *2021 International Conference on Information Technology (ICIT)*, Amman, Jordan, 2021.

[47] L.-W. Kang, K.-L. Chou and R.-H. Fu, "Deep Learning-Based Weather Image Recognition," in *2018 International Symposium on Computer, Consumer and Control (IS3C)*, Taichung, Taiwan, 2018.

[48] C. Bai, M. Zhang, J. Zhang, J. Zheng and S. Chen, "LSCIDMR: Large-Scale Satellite Cloud Image Database for Meteorological Research," *IEEE Transactions on Cybernetics,* vol. 52, no. 11, pp. 12538 - 12550, 2021.

[49] M. F. N. &. S. F. Kusuma, "Weather image classification using convolutional neural network with transfer learning," in *INTERNATIONAL CONFERENCE ON INFORMATICS, TECHNOLOGY, AND ENGINEERING 2021 (InCITE 2021): Leveraging Smart Engineering*, Surabaya, Indonesia, 2021.

[50] K. H. R. A. &. A. S. Nicla Maria Notarangelo, "Transfer Learning with Convolutional Neural Networks for Rainfall Detection in Single Images," *Water,* vol. 13, no. 5, p. 588, 2021.

[51] C. Cheng, Y. Cao, H. Yu, Q. Zhang and X. Cheng, "A Study on Meteorological Recognition With a Fusion Enhanced Model Based on EVA02 and LinearSVC," *IEEE Access*, vol. 12, pp. 28784 - 28797, 2024.

[52] J. Z. Tan, J. Y. Lim, K. M. Lim and C. P. Lee, "Weather Image Recognition Using Vision Transformer," in *2023 IEEE 11th Conference on Systems, Process & Control (ICSPC)*, Malacca, Malaysia, 2023.

[53] M. M. Dhananjaya, V. R. Kumar and S. Yogamani, "Weather and Light Level Classification for Autonomous Driving: Dataset, Baseline and Active Learning," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Indianapolis, IN, USA, 2021.

[54] G. Saini, T. Munasinghe and K. Pan, "Image Classification to identify Transverse Cirrus Band Clouds using Convolutional Neural Networks," in *2021 IEEE International Conference on Big Data (Big Data)*, Orlando, FL, USA, 2021.