

Using Machine Learning Techniques to Develop an Effective Weather Prediction Model from Ground-Based Cloud Images



By

Noran Zulfiqar

Fall-2021-MS-SYSE

Supervisor

Dr. Muhammad Tariq Saeed

Department of Engineering

Ms Systems Engineering

School of Interdisciplinary Engineering & Science (SINES)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

May, 2023

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by Mr/Ms Noran Zulfiqar Registration No. 00000362770 of SINES has been vetted by undersigned, found complete in all aspects as per NUST Statutes/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.


Signature with stamp: 
Name of Supervisor: M. Tariq Saeed
Date: 22/05/2023

(DR. MUHAMMAD TARIQ SAEED)
Assistant Professor
Research Centre for Modeling & Simulation
NUST, Sector H-12, Islamabad

Signature of HoD with stamp: 
Date: 22/05/2023

Dr. Mian Ilyas Ahmad
HcD Engineering
Professor
SINES - NUST, Sector H-12
Islamabad

Countersign by

Signature (Dean/Principal): 
Date: 25 MAY 2023

Declaration

I, *NORAN ZULFIQAR* declare that this thesis titled “Using Machine Learning Techniques to Develop an Effective Weather Prediction Model from Ground-Based Cloud Images” and the work presented in it are my own and has been generated by me as a result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in a candidature for Master of Sciences degree at NUST
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at NUST or any other institution, this has been clearly stated
3. Where I have consulted the published work of others, this is always clearly attributed
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work
5. I have acknowledged all main sources of help
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself

NORAN ZULFIQAR,

Reg No. 00000362770-MSSYSE-Fall-2021

Copyright Notice

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instructions given by the author and lodged in the Library of SINES, NUST. Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights which may be described in this thesis is vested in SINES, NUST, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of SINES, which will prescribe the terms and conditions of any such agreement.

I dedicate this thesis to my family, who has supported me throughout my academic journey. Their encouragement and guidance have played a big role in shaping the person I am today. I am grateful for their unwavering faith in me and for always believing in my abilities, even when I didn't.

Acknowledgements

The journey towards completing this research has been a long and winding road, filled with many twists and turns, ups and downs, and unexpected detours. Along the way, I have been blessed with the support and guidance of many wonderful people who have helped me to stay on track and keep moving forward.

First and foremost, I would like to thank my thesis supervisor, Dr. Muhammad Tariq Saeed, for his unwavering support and mentorship throughout this process. His encouragement and willingness to push me out of my comfort zone has been invaluable, and I feel grateful to have had him as my mentor.

I would also like to thank the members of my thesis committee, Dr. Ishrat Jabeen and Dr. Mian Ilyas Ahmad, for their insightful feedback and constructive criticism. Their input has been instrumental in shaping my research and improving the quality of my work.

Throughout this journey, I have been fortunate to have the support of my family and friends. My parents, and siblings, have always been my biggest cheerleaders, encouraging me to pursue my dreams and supporting me through the inevitable setbacks and challenges. My friends have been my pillars of strength, providing me with much-needed laughter, support, and encouragement when I needed it the most.

This thesis is a result of sleepless nights, and it would not have been possible without the support and encouragement of these individuals. Thank you all for making this journey a little bit easier.

Abstract

The measurement, analysis and forecasting of weather has several use cases in climate modelling, agriculture, and farming. Every year, millions of lives are affected by heavy rains and storms. Therefore, it is important to predict the weather before time to avoid any disaster. Different types of sensors are used to collect data from the environment and then analysis is performed for necessary forecasting. The conventional process requires deployment of several sensors and consequently does not remain very cost-effective for small-scale localized deployment scenarios, especially in farming and agriculture. In recent years, computer vision and AI have made significant progress to address similar problems. In this research, we investigate the potential of ground-based images towards the development of weather prediction models. This study focuses on the development of cost-effective methods for weather-prediction in a localized environment that can be deployed using off-the-shelf, low-cost cameras pointed towards the sky. A custom data set is developed by using images from **S**ingapore **W**hole Sky **I**Maging **C**ATegories (SWIMCAT) dataset, Kaggle and images obtained through web-scraping. The conventional machine learning algorithms i.e., Stochastic Gradient Descent (SGD) Classifier, Support Vector Classifier (SVC) and Random Forest Classifier (RFC) achieve 66%, 76% and 72% accuracy by using Histogram of Oriented Gradients (HoG) features. The study uses a Convolutional Neural Network (CNN) model that achieves 95% accuracy. However, the comparison of training and testing accuracy and validation scores show that the model is over-fitting. To overcome this limitation, a second CNN model has been used that shows a better generalization. The presented dataset demonstrates better class balance compared to previous work, as many models trained on the SWIMCAT dataset overlook class-balance issues. Moreover, the proposed CNN model utilizes fewer layers to tackle over-fitting concerns while maintaining similar accuracy levels. This work can be used for the development of an embedded system or a low-cost IoT system interfaced with a ground camera for low-cost weather forecasting.

Contents

Abstract	vi
List of Abbreviations	ix
List of Figures.....	x
List of Tables	xi
Introduction.....	1
1.1 Cloud Observations – From ancient Warfare to modern Precision Agriculture	1
1.2 Impact of weather on human civilization	1
1.3 Impact of Extreme Weather Events in Pakistan	2
1.3 Challenges in Cloud Classification	3
1.4 WMO Cloud Classification	4
1.5 Early Techniques for Weather Prediction	5
1.6 Advent of ML and Weather Prediction from Ground-Based Cloud Cameras.....	7
1.7 Problem statement.....	8
1.8 Aims of Study.....	8
1.9 Objectives of study.....	8
1.10 Research questions	8
1.11 Expected outcome.....	9
1.12 Datasets for this study	9
1.13 Thesis structure	9
Literature Review.....	10
2.1 Early work in weather prediction	10
2.2 Sensor based weather prediction	10
2.3 Ground based camera approaches.....	13
2.3.1 Total sky imagers and Whole sky imagers.....	13
2.3.2 Use of Digital Camera for weather prediction	14
2.3.4 Advantages of low-cost solutions	18
2.4 Research gap	18

Research Methodology	19
3.1 Workflow of Methodology	19
3.2 Data Sources	21
3.2.1 SWIMCAT Database.....	21
3.2.2 Kaggle Data	21
3.2.3 Custom Data Set	21
3.3 Exploratory Data Analysis (EDA).....	22
3.4 Features Extraction.....	22
3.5 Features Extraction using Bag of Features (BoF).....	22
3.6 Feature Extraction using Histogram of Oriented Gradients.....	22
3.7 Development of Machine Learning models.....	23
3.7.1 Decision Tree Classifier	23
3.7.2 Support Vector Machine (SVM) Classifier	24
3.7.3 Random Forest Classifier	24
3.7.4 Classification using a Deep CNN model	25
3.8 Hyper-parameters Tuning	26
3.9 Performance Evaluation.....	27
Results and Discussion	30
4.1 Data Visualization	30
4.2 Classification using Stochastic Gradient Descent Gradient (SGD) Algorithm	31
4.3 Hyper-Parameters Tuning for Best Estimator	32
4.4 Classification using Convolutional Neural Networks	37
Conclusion and Future Work.....	42
REFERENCES.....	43

List of Abbreviations

SWIMCAT	Singapore Whole sky Imaging Categories Database
EDA	Exploratory Data Analysis
SVC	Support Vector Classifier
CNN	Convolutional Neural Network
SGD	Stochastic Gradient Descent
HOG	Histogram of Oriented Gradients
SIFT	Scale Invariant Feature Transform
CSI	Clear Sky Index
RELU	Rectified Linear Unit

List of Figures

Figure 1.1: Temperature Rise in Pakistan (1901 - 2021)	13
Figure 1.2: Cumulative Precipitation Anomaly (July 1 to August 31)	14
Figure 1.3: Cloud Classification Types	16
Figure 2.1: Masked Images of four different class samples (Fish, Flower, Sugar, Gravel) from the train dataset. [19]	24
Figure 2.2: Weather station sensors (left) and 3D scheme of a whole-sky camera (right) [32]	27
Figure 2.3: The three predefined classes of sky state shown in an image. [38]	29
Figure 3.1: A generic workflow of machine learning methodology for classification of cloud data	31
Figure 3.2: A schematic diagram to understand the working of Random Forest Models.	34
Figure 3.3: A schematic diagram to understand the working of Support Vector Machine (SVM).	36
Figure 3.4: The confusion matrix is used to depict algorithm performance	38
Figure 3.5: The Area under the Receiver Operating Characteristic curve with various thresholds	40
Figure 4.1: Five Random images from each class visualized using python matplotlib library.	41
Figure 4.2: Relative number of Images for each class in SWIMCAT data set	42
Figure 4.3: Parameter grid for hyper-parameters tuning using Grid-Search CV	44
Figure 4.4: Configuration of the best estimator resulted after Grid Search	44
Figure 4.5: Confusion Matrices of three machine learning models used for classification	45
Figure 4.6: Visualization of a Decision Tree in the Random Forest Model	47
Figure 4.7: Accuracy plot for training and test data using CNN classification without a Dropout layer	49
Figure 4.8: Loss plot for training and test data using CNN classification without a Dropout layer	50
Figure 4.9: Revised CNN model with 20% drop-out layer to address overfitting	50
Figure 4.10: Accuracy plot for training and test data using CNN classification by adding 20 percent dropout.	51
Figure 4.11: Loss plot for training and test data using CNN classification by adding 20 percent dropout	51

List of Tables

Table 3.1: Different categories of Sky/Cloud Images	32
Table 4.1: Performance of Stochastic Gradient Descent (SGD) classifier	47
Table 4.2: Performance of Support Vector Machine (SVM)	47
Table 4.3: Performance of Random Forest Classifier (RFC)	48

Chapter 1

Introduction

1.1 Cloud Observations – From ancient Warfare to modern Precision Agriculture

Humans have an ancient association with clouds and their patterns due to various reasons such as amusement, rain prediction and references in religious scriptures. They have captured the attention of people from diverse cultures due to their dynamic features [1] [2]. In the ancient world, cloud patterns were symbolized as superpowers, and this association of cloud shapes with religious symbols is still present in various civilizations.

The forecasting of cloud and weather conditions has several use cases. During warfare, the information can be of great tactical advantage [3]. Certain weather conditions, such as mist or fog and heavy cloud cover, can be used by armies to carry out an attack or surprise movement of supplies and troops [4]. Cloud cover forecasting was used in World War 2 (WW2) for the planning of bombing raids. These use cases highlight the importance of cloud forecasting in warfare. The assessment of weather conditions has remained a goal of military reconnaissance missions. Weather balloons were used by the US Army to collect information about the movement of troops in WW2.

In addition to being a part of warfare, religion, and culture, they played an important role in farming and agriculture before the development of any scientific equipment to predict weather [5]. Farmers relied on the different cloud patterns to predict the upcoming weather in their area and used that information in planning their agricultural activities [6]. Certain patterns of clouds were associated with different weather conditions, and in some cases, the movement and shapes of clouds were used to find the location.

1.2 Impact of weather on human civilization

Since the advent of recorded history, weather has affected the human race. Extreme weather events such as hurricanes, floods, and droughts affected human civilization. In recent years, due to climate change, the frequency of extreme weather events has increased [7].

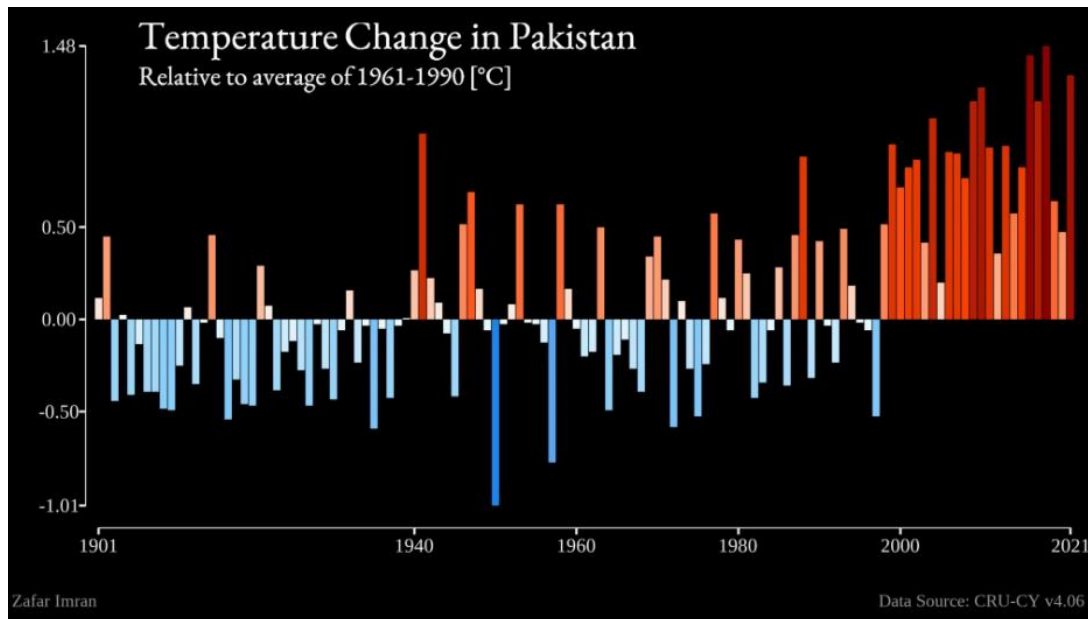


Figure 1.1: Temperature Rise in Pakistan (1901 - 2021)

Each year, hundreds of climate-triggered disasters occur, resulting in the loss of human and animal lives, displacement of hundreds of millions of people, and disruption of social and ecological systems. In 2017, Hurricane Harvey and Hurricane Maria caused several thousands of deaths in the United States and Puerto Rico due to devastating floods. These events also resulted in billions of dollars of damage to infrastructure and businesses. Cyclone Idai hit Zimbabwe in the year 2019 and resulted in widespread damage. The wildfires in California (2017-2020) and bushfires in Australia (2019-2020) resulted in the deforestation of millions of acres and billions of dollars in damage.

1.3 Impact of Extreme Weather Events in Pakistan

Pakistan is among the countries which are most vulnerable to climate change [8]. Due to the increase in average annual temperatures (Figure 1.1), the climate cycles and annual cumulative precipitation have been severely impaired (Figure 1.2). In the last 15 years, there has been a sharp increase in extreme weather events.

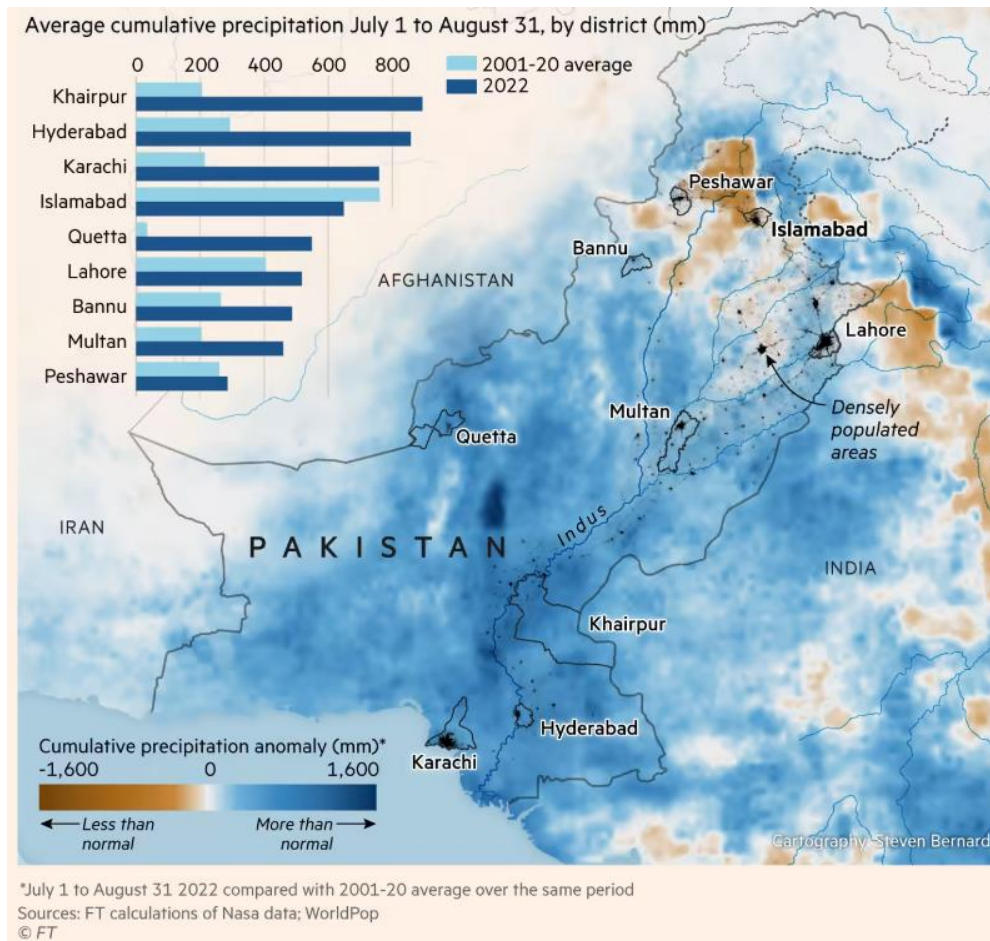


Figure 1.2: Cumulative Precipitation Anomaly (July 1 to August 31),

<https://www.ft.com/content/116f34fc-b44d-487d-822b-d3f1926eaca2>

This includes devastating floods in 2010, heat waves in 2015, 2020 and 2022, drought in 2018, increased events of glaciers melting and landslides. In 2010, due to heavy rainfall in the provinces of Khyber Pakhtunkhwa, Punjab and Balochistan, approximately 20 million people were displaced. It resulted in around 2000 death and nearly 20% of the area was affected.

1.3 Challenges in Cloud Classification

The classification of clouds involves the analysis of cloud data (such as their structural appearance, thickness, color, altitude, etc.) and their categorization into various types. Due to several variations in the aforementioned features, the task of cloud classification becomes very challenging.

One challenge involved in cloud classification is the lack of image data sets about cloud types. Many of the publicly available data sets comprise satellite imagery. However, very little data is available that can be used for ground-camera-based cloud classification. The two data sets that are available and have been widely used are HYTA and SWIMCAT.

Another challenge involving cloud classification is the lack of standardization and benchmarks on cloud types. Scientists have different viewpoints on cloud types as they have overlapping sets of features. To overcome this challenge, the World Meteorological Organization (WMO) has introduced a standard that takes into account the cloud altitude, cloud structure, and moisture contents (Figure 1.3). The WMO standard divides clouds into low-altitude clouds, mid-altitude clouds, and high-altitude clouds. Clouds having a height of about 2000 meters are classified as low-altitude clouds, clouds with a height of more than 6000 feet are categorized as high-altitude clouds, and clouds having a height between the range of 2000 to 6000 meters come under the category of mid-altitude clouds. The WMO standard further divides the low-altitude clouds into four more cloud types, i.e., Stratus, Cumulus, Cumulonimbus, and Stratocumulus. The mid-altitude cloud types are further segregated into Nimbostratus, Altostratus, and Altopcumulus. The high-altitude cloud types are further divided into Cirrostratus, Cirrus, and Cirrocumulus. The cloud types are further explained in the next section.

1.4 WMO Cloud Classification

The WMO cloud classification system further divides the low-altitude, mid-altitude and high-altitude clouds into the following types:

Stratus Clouds: These are low-altitude clouds that are white or gray in color and these are often associated with light rain. The appearance of stratus-clouds is like a white or gray layer that resembles a blanket in the sky [9].

Cumulus Clouds: These are also low-altitude clouds that are white and fluffy in shape and associated with rise of warm air and condensation of water vapors into clouds.

Cirrus Clouds: Cirrus clouds are present at a height of 6000 meters and are categorized as high-altitude clouds. These clouds are often irregular in shape and made up of ice. Storms are often associated with Cirrus clouds.

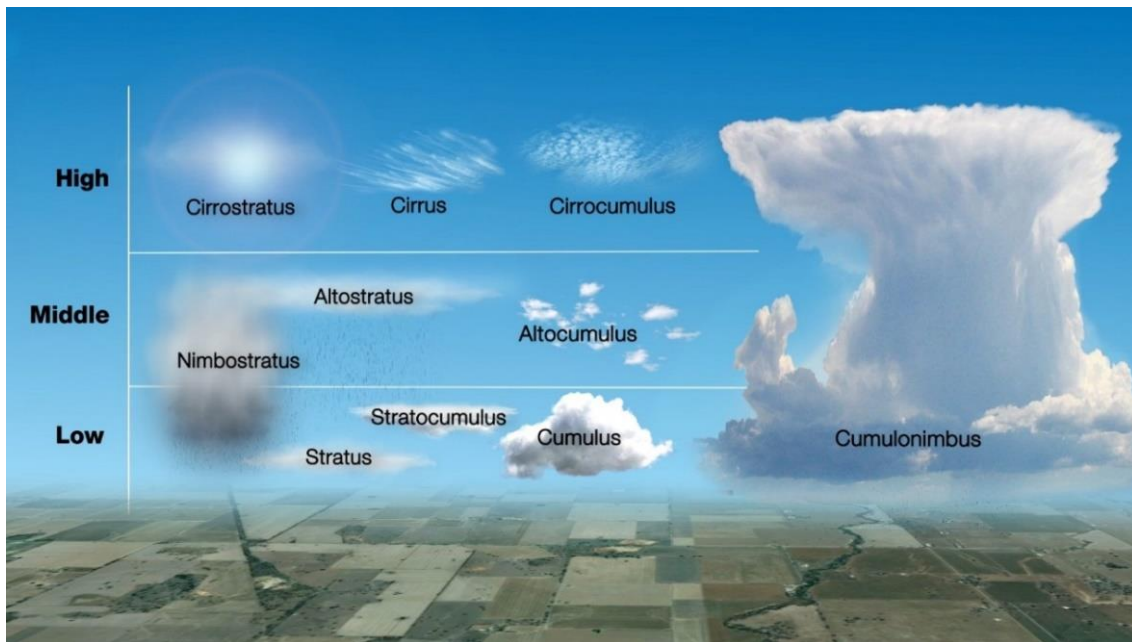


Figure 1.3: Cloud Classification Types according to World Meteorological Organization (Image Taken from International Cloud Atlas <https://cloudatlas.wmo.int/>)

Cumulonimbus Clouds: These are towering clouds that start from low-altitude and can go up till high-altitude (above 6000 meters). These large vertical clouds are formed when warm air rises and then cools suddenly.

Altocumulus Clouds: These are categorized as mid-altitude clouds and appear white or gray with round edges and are associated with good weather or an approaching storm.

Stratocumulus Clouds: These are like Altocumulus clouds in structure and color, but they are often associated with light rain.

Cirrostratus Clouds: These are high-altitude thin clouds with white color often associated with light rain or an approaching storm.

1.5 Early Techniques for Weather Prediction

Initially, weather prediction was done manually using various instruments and observations. Experts in weather prediction used to predict the weather based on their observations and using manual techniques. Weather parameters such as temperature, humidity, and atmospheric pressure were measured using equipment such as ‘thermometers’, ‘hygrometers’, and ‘barometers’. In the manual weather prediction, the movement and appearance of clouds, wind direction, speed, and other natural phenomena were observed to

predict the weather. However, manual weather prediction had several limitations. It was time-consuming and required a lot of data to make accurate predictions. Additionally, it was prone to errors due to subjective interpretation and human biases. Lastly, it lacked the ability to make long-term forecasts, which are critical for industries such as agriculture and transportation.

With the evolution of technology, scientists started using satellites in the 1970s for computing complex data for weather prediction [10]. Weather prediction through satellite images involves analyzing the cloud patterns, temperatures, and other atmospheric conditions observed from space to forecast weather patterns. Although satellite images have been an essential tool in weather prediction for several decades, they have some drawbacks. The data coming from GIS satellites provide data after a certain period of time, making it unsuitable for frequent weather prediction. Moreover, satellite data is mostly not available for public use, which limits its accessibility to relevant people. Furthermore, satellite images do not have localized data, making them unsuitable for certain regions. Although weather prediction through satellite images is a useful resource, we cannot solely rely on their outcome. Better and more accurate results can be produced by combining their results with other resources such as radars and complex computational models.

Using sky imagers is another way to predict the weather. There are two types of sky imagers, namely total sky imagers and whole sky imagers. A total sky imager captures the hemisphere of the sky in a single shot. On the other hand, a whole sky imager captures the image of the whole sky. Their data can be used for studying various cloud patterns and atmospheric phenomena [11]. The data generated from imagers can be used for determining the cloud types based on their shapes and movements. Although sky imagers cover a good area of sky, we cannot solely rely on their results because they only cover the sky that is exactly above them. Additionally, these cameras are sensitive to certain conditions (glare and shadows) and do not provide any information about important parameters of weather which makes their results doubtful. They can be used as complementary tools in conjunction with other techniques, such as satellite imagery, radar data, and ground-based weather stations, to improve the accuracy and reliability of weather forecasts.

Ground-based cameras are a suitable method for collecting cloud images for weather prediction for several reasons. They can capture high-resolution images of the sky and clouds with high accuracy and detail, which is useful for small-scale weather patterns and changes in

cloud formations. The feasible handling of these cameras makes them suitable for installation in remote areas, which provides localized data and improves the data quality. Ground-based cameras are low-cost and can be used by local people easily. They can be easily integrated with machine learning models for accurate weather prediction.

1.6 Advent of ML and Weather Prediction from Ground-Based Cloud Cameras

In the past, several approaches have been used for classification of clouds and weather prediction [12]. The details of these approaches are presented in Chapter 2, but these can be divided into numerical or mathematical approaches, data driven machine learning based approaches and hybrid approaches. In numerical weather prediction based approaches, the atmospheric region of interest is divided into a grid-like structure where each cell covers a certain area. Then numerical or mathematical modelling techniques are used to study and simulate the flow of important features (humidity, air speed, cloud shapes and structure etc.) across the grid of cells. Accurate weather prediction of large regions of interest requires lot of computational power.

On the other hand, machine learning based approaches have widely used in the recent years. These are approaches that use labelled data sets in case of supervised learning approaches to identify various patterns in data received from weather stations. In many cases, these approaches show better results because the modern machine learning algorithms are good at extracting hidden features present in raw data. The advancement in machine learning has revolutionized everything. The complex algorithms of machine learning have been used to solve big problems in the domains of weather, economy and agriculture. These techniques can be used in weather prediction, leveraging the availability of large datasets of weather-related information. These techniques involve using statistical algorithms to identify patterns in the data and make predictions about future weather conditions. Ground-based cloud images have become increasingly popular as a source of data for weather prediction using machine learning techniques, as they provide a visual representation of cloud formation and movement. Using machine learning for weather prediction is an effective way for weather prediction due to the improved accuracy, real-time updates, increased efficiency, scalability and integration with other systems. These benefits can lead to more reliable and accurate weather predictions, which can have significant benefits for management authorities. In this thesis, we aimed to develop a weather prediction model using ML techniques from ground-based cloud images.

1.7 Problem statement

Weather prediction is important since it has an impact on human lives. The traditional approaches used for weather prediction are expensive and sometimes do not provide localized data. Pakistan, being a developing country, cannot afford expensive weather prediction software and suffers every year because of natural disasters that occur due to severe weather. This study explores the use of ground based camera images to develop a low-cost weather prediction model using ML/DL techniques.

1.8 Aims

The aim of this study is to understand the hidden patterns in the cloud images and classify them in to different classes. These classes can then be used to predict different types of weather conditions. The results generated from this research can be integrated in bigger weather prediction systems and used for accurate weather prediction of the area of interest. Authorities can utilize this data to take precautionary measures before a natural disaster due to bad weather occurs.

1.9 Objectives of study

The objective of this research is to develop an effective weather prediction model that can use ground-based cloud images to predict weather. Given below are some major objectives of this study:

- 1) Build a custom dataset by fusion of existing public datasets and the dataset collected through web-scraping.
- 2) Development of multiple ML/DL models on our custom dataset.
- 3) Performance evaluation and comparison of the developed ML and DL techniques

1.10 Research questions

Based on its objectives, this research is focused on the following research questions:

- 1) Can we use ground camera to accurately predict local weather conditions?
- 2) Can we use cloud images for weather prediction?
- 3) How much accuracy can AI models achieve for this study?
- 4) How much data will be required for development of AI models for this research?

5) Which AI algorithm is more suitable for this research?

1.11 Expected outcome

The intended outcome of this study is to develop a cost-effective weather prediction model that takes a cloud image as an input and predicts weather effectively. The results generated from this system may not be sufficient themselves but can be used as an integral part of a huge weather prediction system. The predictions can be useful for higher authorities linked to flood management and agriculture to make better decisions.

1.12 Datasets for this study

The publically available datasets for this study were insufficient and un-balanced. Therefore, a fusion of the available datasets from “Kaggle” and “Singapore whole sky cloud images” dataset with web-scraped cloud images were used to create a customized dataset.

1.13 Thesis structure

The contents of this thesis are divided in to five main chapters namely introduction, literature review, methodology, results and discussion and finally conclusion. Chapter 1 contains a detailed background, objectives and intended outcome of this research. Chapter 2 provides a review of the previous studies conducted in this area and important contributions by different researchers and the limitations associated with their work. Chapter 3 discusses the workflow of our proposed methodology. It contains a detailed overview of the techniques and algorithms that were used for conducting experimentations for this study. Chapter 4 highlights results obtained from the experiments and important discussion. chapter 5 concludes the overall outcome of this study along with a future roadmap. Figures and tables are separately mentioned at the start of this thesis and references are mentioned at the end.

Chapter 2

Literature Review

2.1 Early work in weather prediction

Weather prediction has several applications and therefore predicting it accurately is an important problem. As discussed in previous chapter, classification of cloud types is an important step towards weather prediction. An extensive research has been carried out in the domain of weather forecasting from sensor –based models to ground-based camera approaches. Different researchers have proposed different techniques to improve the performance of weather prediction models. This chapter discusses a detailed overview of the past work done to improve the prediction capability of weather prediction models.

2.2 Sensor based weather prediction

Ohring, G., et al. [13] highlighted how the satellites can be utilized in providing useful information for the measurement of Earth's climate and weather. This work focused on the numerical prediction models of the Environmental Modelling Center of the U.S. The study, further discusses the National Center for Environmental Prediction forecast model, assimilation system. Finally, it was figured out that the weather prediction models of NCEP use 83% of satellite data which is only a fraction of the available data. If more data is used then it will not only increase the accuracy and precision but will also improve the timelessness of the weather prediction models.

Pierro, Marco, et al. [14] published a work “Data-driven up-scaling methods for regional Photovoltaic power station”. PV prediction is an important element in a number of systems such as Energy Traders and Transmission System Operators etc. He conducted a study in the region of Italy for prediction and calculation of Photovoltaic electricity. The research used neural networks and clustering technique for the prediction of Photovoltaic energy. Satellite data and numerical weather data were used as an input for the model. It was observed that less computational power was required for the simple technique. Additionally, the prediction intervals for the PV energy are computed using probabilistic correction technique. These results can be used in energy reserve evaluation, energy trading optimization and transmission scheduling to lower the fine expenses and maintain energy.

Kalsi, S. R. [15] did a comprehensive analysis of weather forecasts for Indian agriculture and demonstrated how satellite data can be utilized in forecasting of weather conditions. Accurate weather prediction is extremely important for crop yields and socioeconomics growth. Satellite technology plays an important role in weather monitoring systems and dynamic modelling due to its high resolution and multi-spectral bands. In agriculture, the data from both domestic and foreign satellites is crucial and the images data generated can be used to locate convective precipitations in weather. To enhance the results of convectional forecasting and analysis that can be beneficial to Indian agriculture, using a hybrid version of satellite data along with the traditional meteorological approach is recommended. Furthermore, the dynamic space-based observing system must stay up-to-date with user requests and developments in numerical weather prediction.

Saunders, Roger. [16] emphasized on the use of satellite data for numerical weather prediction and discussed various ways to use satellites for the weather forecasting. Satellites offer a multitude of observations of the Earth's surface and atmosphere, providing a real-time global coverage. Advanced satellites have the ability to measure upwelling radiation from the surface, atmosphere, aerosols, and clouds to deduce profiles of atmospheric temperature, water vapors, and other gases. The speed and direction of these radiations from the sea can be used to forecast cyclones. Satellite sounders that operate in the microwave and infrared regions of the spectrum are crucial to measure the atmosphere accurately.

Neiburger, Morris, and Harry Wexler. [17] discussed the effects of weather satellites on meteorology and weather forecasting approaches. Tiros I and II provided the first perspective of large-scale weather patterns, taking 14,000 and 24,000 images and measuring visible and infrared radiation exiting the Earth's atmosphere. The use of satellites with advanced technology, for data collection has revolutionized the domain of meteorology and the accuracy of weather prediction models. The shapes of clouds in satellite images can tell a lot about the upcoming weather in that area. They show large-scale low-pressure areas and cyclonic wind patterns associated with circular cloud patterns up to 1,000 miles in diameter. These observations can be used to support the ground-based systems and improve energy budgets of the atmosphere.

Li, Zhiwei, et al. [18] suggested MSCFF, a deep learning based cloud detection approach that uses a symmetric encoder-decoder module to extract multi-scale spatial characteristics from feature maps. The images from different satellites were used to test this approach and it was

observed that MSCFF achieves greater accuracy than conventional rule-based cloud identification approaches and deep learning models. It has potential to be used in real-world environments for cloud detection in images of different resolutions which is important for precise application in remote sensing. According to the researcher, cloud detection needs an all-encompassing technique to handle the rise in satellite image sources instead of digital number values for radiometric calibration. Although this approach can accurately detect clouds it can be inaccurate sometimes. This can be prevented by increasing the receptive pitch of the model but it complicates the model. Another issue is the insufficient dataset for high-resolution cloud dataset. However, the accuracy of MSCFF model can be improved by generalizing it for a range of datasets

Ahmed, Tashin, and Noor Hossain Nuri Sabab. [19] presented a technique to classify the clouds in to four main. Climate change is a problem on global level and the study of clouds at low-altitude play an important role in understanding the climate. Clouds were classified in to four main types namely “Sugar”, “Gravel”, “Flower”, and “Fish” using the satellite images. EfficientNet was used for experimentation and comparison of results. EfficientNet is a technique provided by Google AI research. It is based on convolutional neural network and has 8 different types. It works on the dataset depth-wise and point-wise, in inverse ResNet blocks and in linear bottleneck. Kaggle dataset “Understanding clouds from satellite images” was used in this research. The technique presented in this research can be changed for other remote sensing applications.

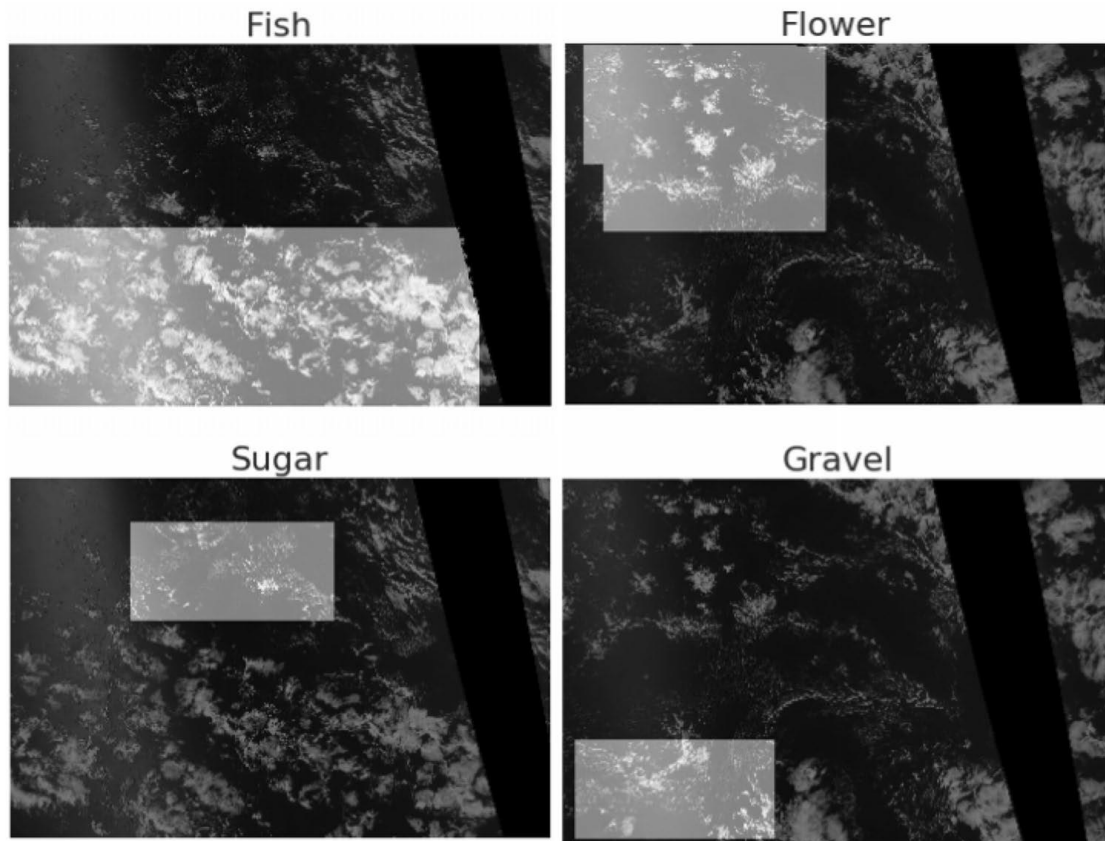


Figure 2.1: Masked Images of four different class samples (Fish, Flower, Sugar, Gravel) from the train dataset.

[19]

2.3 Ground based camera approaches

2.3.1 Total sky imagers and Whole sky imagers

Soumyabrata et al. [20] developed a supervised segmentation framework for identification of colour spaces in sky images. Cloud classification plays a vital part in metrological applications such as correct prediction of upcoming weather, wind, condensation trail, rain, and snow etc. [21] [22] [23]. Classifying clouds based on their shapes is not recommended due to the ever-changing shapes of clouds [24]. They used partial least square regression and probabilistic segmentation methods for segmentation of Singapore Whole Sky Images Segmentation Database.

Calbo, Josep, and Jeff Sabburg. [25] used cloud images captured from digital cameras for feature extraction such as statistical measurements of image texture, and Fourier transform of the image. DIP technique was highlighted as a classification method in this study. Clouds are crucial to meteorology, our daily activities [26] and affect energy balance on both local and global scales [27]. The traditional cloud research requires expert human observers to determine the direction and quantity of cloud cover [28] [29] [30]. Although the available

systems provide details at a local level, they are costly. Therefore, sky imaging devices are used. The presented technique can obtain fractional sky cover and other sky characteristics from any Whole-sky camera image. The classification gives 62% accuracy results when 8 sky conditions are considered. The results deduced from this study can be useful for research on cloud effects on radiative transfer.

Zhuo, Wen, Zhiguo Cao, and Yang Xiao [31] proposed an algorithm to capture both texture and structure information from a colour sky image by converting RGB values red, green, and blue values to different colour spaces and applying census transform. Total-sky imagers were used in this study. This research uses ground-based camera because lately ground-based cameras such as WSI [32] [33], TSI [34] and ASI [35] [36] [37] have been widely used due to their detailed sky coverage. A comparison of results deduced from this study with previous approaches show that it is better than the rest of approaches used for classification. Moreover, the model can differentiate images captured at different times of the day.

2.3.2 Use of Digital Camera for weather prediction

Calvin et al. [38] presented a work to demonstrate how an image dataset, obtained using webcams can be used to build cloud maps [39]. Cloud data can be regarded as an important element for predicting the weather condition of an area. However, building cloud maps with the traditional approach of using satellite imagery is a costly approach. The author demonstrated how ground-based webcams can be used instead of costly geo-stationary satellites to build cloud maps. They used 4388 geographically distributed cameras, across USA. The webcams were installed at different locations of the continental USA and DynTex database was used for the experimental purpose. To train the model, a year-long data of satellite images was collected along with the cloud images collected through geographically-installed webcams. The data collected is used for training the dynamic model and regression models in order to relate ground-up data with the satellite data at that location. The output of these models is provided to a new algorithm based on hierarchically regularized dynamic texture calculation. Traditionally, dynamic models work in two phases [40] [41] [42]. The hierarchical model presented in this paper is able to construct the cloud maps more accurately than the standard models. It was also demonstrated that this technique can be useful in other applications.

Martin et al. [43] developed a new system to collect data necessary for short-term weather prediction. Accurate weather prediction is critical in managing solar power plants [44]. In

recent years, All-Sky cameras installed on ground have become popular to predict short term weather [45] [46]. Efficient methods for solar energy generation and mitigating the emission of carbon are essential for improving sustainability. Therefore, a small weather station is suggested in this research study. The meteorological information is stored in the form of a database by the system using weather sensors. A ground-based camera with fish-eye lens was installed to capture sky images. These cameras are suitable for short-term and local weather forecasting. The camera provides a view of 180° with a high resolution and has a wireless connection with the server station [47] [48]. A total of 1000 images were taken and were resized to 227×227 pixels before splitting them into training and testing data with 70:30 ratio. MATLAB was used for classifying the cloud images. Using Deep CNN technique, the systems classified the images in to four classes: “Clear sky”, “Partly cloudy”, “Mostly cloudy”, and “Overcast”. The accuracy of this system was 97.8%. Although the suggested system is a good approach for weather forecasting, it does not perform well in rain, snow and dust. High dynamic range images with more classes used in combination of meteorological data and neural networks can improve the accuracy of this model

Tran-Trung et al. [49] studied the use of Local Binary Patterns (LBP) and its limitations for classifying clouds in naturally striated formations. He used Local Ternary Patterns to address the issue and combined features such as color characteristics, Local Binary Patterns and Local Ternary Patterns to reduce the number of histograms needed for picture characterization. Cloud classification plays a vital role in weather forecasting and disaster management as different cloud patterns indicate different weather conditions [50]. Using ground-based digital cameras is an effective way to gather information such as cloud parameters, temporal resolution, and spectral resolution etc. The traditional weather forecast systems are costly therefore innovative techniques are suggested that will also give improved results such as Support Vector Machine (SVM), Fast Fourier Transform Projection and Local Binary Patterns etc.

Alessandro et al. [51] presented a solution in 2023 for classification of sky images. The algorithm calculates Clear Sky Index (CSI) to forecast radiance condition for the scheduling of a photo-voltaic power grid, which is very important for energy stability and mitigation of greenhouse effects [52] [53]. A custom data set has been developed in this study. The data set was collected through cameras over a period of 72 days. The models used in this study are Neural networks and Random Forest for solving the classification problem over four classes

i.e. “Clear Sky”, “Overcast”, “Partially-Cloudy”, “Over-irradiance”. The models developed in the study achieved almost 90% accuracy.

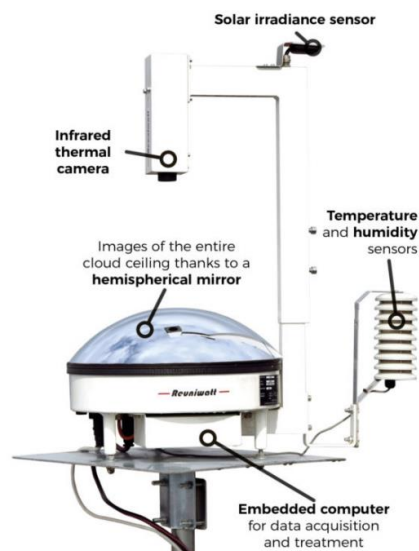
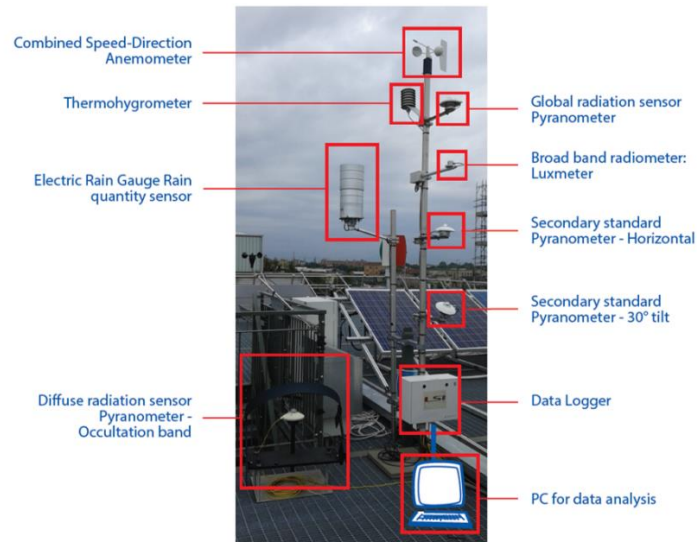


Figure 2.2: Weather station sensors (above) and 3D scheme of a whole-sky camera (below) [32]

Cao, Yingyue, and Hanpeng Yang. [54] created an easy-to-use mobile application that predicts weather based on the images take from the mobile camera. The conventional models for weather prediction are expensive and sometimes rely on spatial textures [55]. A database containing the form of clouds was used along with deep learning techniques. The model yields pretty good accuracy for clouds including fog, fair-weather and rain but poor results for Tornado clouds. The images captured form the mobile cameras are classified using Convolutional Neural Network and the same technique is used for training the application.

The model's accuracy is satisfactory for showers, good weather, and fogs, but tornado accuracy still needs to be improved. The paper explains the creation of a mobile app that creates a rough weather prediction based on smartphone images of clouds. Although the application gives good results but they can be improved by providing more training set and making the application flexible to intake poor-quality images

Sahaya et al. [56] focused on developing a weather prediction model that can help farmers to identify the velocity flow essential for farming activities. Accurately predicting weather before time is critical for farmers in India [57]. Rainfall data was collected, and feature extraction was done. Using normalization, these features were ranked. MATLAB was used for simulation and error rate was predicted using NN, SVM, BT, AND RF. The model can be used for making important decision in agriculture.

Can et al. [58] Performed various method to improve the strength of deep CNN cloud classification model. Cloud cover is essential in climate modelling, renewable energy resources and, other meteorological applications. The researcher used a digital camera with visible spectrum to collect the cloud images for the database. Labels and colored images of 8 cloud types were collected from Baidu website. The cloud types used in this research are altostratus, altocumulus, cumulus, cumulonimbus, stratocirrus, cirrocumulus, cirrus and nimbostratus. Models with different layers were built. The performance of these models was tested by comparing the output of this method with the observations of experienced researchers. Although the results produced from this research are pretty good, they aren't absolutely correct as clouds appear different to different observers.

Mariza Pereira, et al. [59] introduced a simple yet useful approach for calculating the clouds in the sky. This information is essential in building accurate weather models [60]. This study was conducted at the Brazilian Antarctic Station and a digital camera with visible color spectrum was used for collecting the image dataset [61]. Cloud information was collected using the "Hue", "Intensity", and "Saturation space". Pictures were captured for a period of 50 days and the results of 29 images were compared with the observations of experienced meteorologists. The results of the prescribed model are mostly similar to the predictions made by specialists.

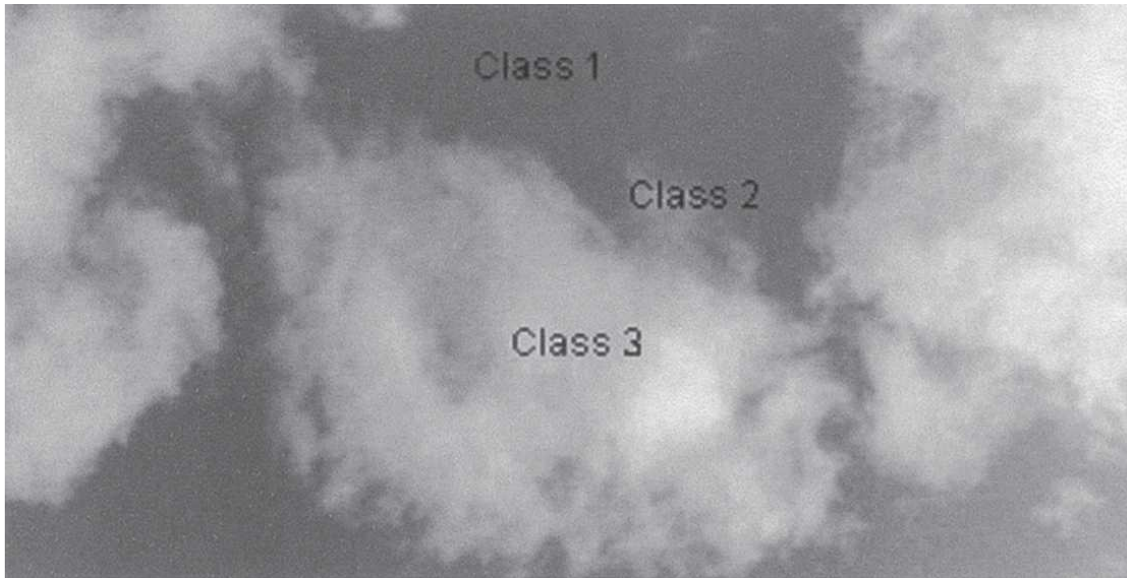


Figure 2.3: The three predefined classes of sky state shown in an image. [38]

2.3.4 Advantages of low-cost solutions

In view of the past work done in the domain of weather prediction using cloud classification, digital cameras are preferable in terms of cost and accessibility. Given are some of the benefits of low-cost solutions:

- The solution can be easily accessed by local people
- More systems can be installed in the same budget as required for a complicated system
- Sufficient amount of dataset can be gathered which improves the accuracy results of weather prediction models

2.4 Research gap

After the detailed study of the previous research done in the respective domain, few gaps were observed that can be addressed in this research. Most of the work done in the domain of weather prediction is based on satellite data which is an expensive way to predict weather and they do not provide localized results. Researchers started using the ground-based cameras to reduce the cost of weather prediction but, in Pakistan this area is still unexplored. Moreover, the publically available datasets are insufficient and most of them have imbalanced images for some classes.

Chapter 3

Research Methodology

In this chapter, we provide an overview of methodology used for classification of cloud images.

3.1 Workflow of Methodology

The overall workflow of various steps used in methodology is shown in Figure 3.1. The important steps include developing a custom data set, preprocessing and scaling, features extraction, training of machine learning models, hyper-parameters tuning and performance evaluation.

Custom Data Set Development: This step involves gathering images of clouds from various sources and combining them into a single dataset. The sources include data set obtained using image scraping from various websites, publicly available datasets like Kaggle, and specialized datasets like Singapore whole-sky imaging categories (SWIMCAT) dataset.

Image Preprocessing and Scaling: Image preprocessing involves preparing the images for analysis by removing noise, resizing, and standardizing them. Scaling involves normalizing the pixel intensities of the images to a fixed range to ensure that the data is consistent across all images.

K-Means Clustering for identification of various classes of Cloud Images: K-Means clustering is an unsupervised learning algorithm that partitions data into k clusters based on their similarity. In this step, K-means clustering is used to identify different classes of cloud images based on their visual features.

Features Extraction using Scale Invariant Features Transformation (SIFT) and Histogram of Oriented Gradients (HoG): SIFT and HoG are feature extraction techniques used to extract relevant information from images. SIFT is used to identify distinctive features in images that can be used to match different images. HoG is used to extract the gradients of the images, which can be used to identify patterns and structures within the images.

Development of Machine Learning Algorithms using Decision Tree, Random Forest and Deep Neural Networks: In this step, various supervised learning algorithms such as Decision Tree, Random Forest, and Deep Neural Networks are used to classify the cloud

images based on the extracted features. These algorithms learn from the training data and create a model to predict the class of new data.

Performance Evaluation using Accuracy, Precision, Recall and F1 Score: The performance of the model is evaluated using various metrics such as accuracy, precision, recall, and F1 score. These metrics are used to assess the accuracy and effectiveness of the machine learning model.

Hyper-Parameters Tuning: Hyperparameters are parameters that are not learned by the machine learning algorithm but are set by the user. In this step, hyperparameters of the model are tuned to improve its performance.

Cross Validation: Cross-validation is a technique used to assess the performance of a machine learning model by splitting the data into training and validation sets. In this step, the model is evaluated using cross-validation to ensure that it can generalize to new data.

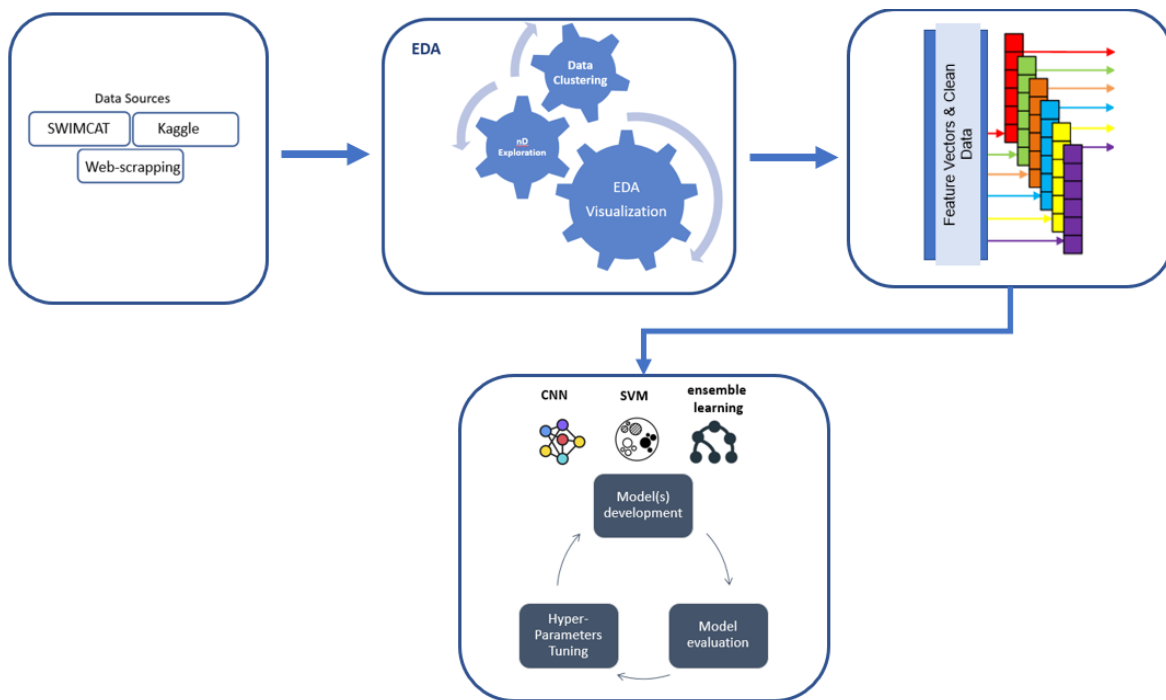


Figure 3.1: A generic workflow of machine learning methodology for classification of cloud data

3.2 Data Sources

Several studies have been conducted towards classification and segmentation of cloud images and local weather prediction. However, very little data is available in the public domain. Among the publicly available data sets, HYTA [62] and SWIMCAT (*Singapore Whole-sky IMaging CATegories database*) have been used in many studies.

3.2.1 SWIMCAT Database

Lee and Winker at *Nanyang Technological University* (NTU) Singapore used a ground based sky imaging camera i.e. WAHRISIS. From January 2013 to May 2014, a total of 784 patches were captured. The data was divided into 5 classes. These are “Clear Sky”, “Patterned Clouds”, “Thick Dark Clouds”, “Thick White Clouds” and “Veil Clouds”. The dimension of these image patches are 125 x 125 pixels.

3.2.2 Kaggle Data

Several small data sets are available for the problem of cloud classification. These data sets are mostly obtained from optical satellite sensor, pointed towards earth. Not much work has been done on data obtained from ground-based cameras. The two data sets that are available on Kaggle.

Category	Number of Images	Type
A	224	Clear Sky
B	89	Patterned clouds
C	251	Thick dark clouds
D	135	Thick white clouds
E	85	Veil clouds
Total	784	

Table 3.1: Number of instances of each class in SWIMCAT dataset

3.2.3 Custom Data Set

In this work, we merge the SWIMCAT data set with data sets obtained from Kaggle. Moreover, we collect open-source images from Google using keywords: ‘Clear Sky’, ‘Overcast’, ‘Cloudy’ and ‘Partially Cloudy’.

3.3 Exploratory Data Analysis (EDA)

In this step, we performed data visualization for our custom data set for get a sense of what the images look like, their dimensionality, color space and other characteristics.

3.4 Features Extraction

Variations in texture, thickness, statistical features etc. impose a significant challenge on the classification of ground-camera based images. Moreover, the changes in illumination or light variations make features extraction even more challenging. In this work, we use Bag-of-Features (BoF) as a pre-processing technique for features extraction and use its results for conventional machine learning algorithms i.e. Decision Tree, Support Vector Machine (SVM) and Random Forest (RF). Moreover, we use a deep Convolutional Neural Network (CNN) model for image classification.

3.5 Features Extraction using Bag of Features (BoF)

The use of Scale Invariant Feature Transform (SIFT) in classical computer vision has been widely popular for quite some time. Introduced by David Lowe, the algorithm first transforms the given input image into a feature-vector representation. The feature representation generated by SIFT remains invariant to basic transformational changes (scaling, translation, rotation). Moreover, this representation also remains partially invariant to changes in illumination and minor projects in the 3D space. The algorithm generates key points and then uses each point to for generation of image feature vectors. The generated features are called SIFT keys. (<https://machinelearningknowledge.ai/image-classification-using-bag-of-visual-words-model/>)

3.6 Feature Extraction using Histogram of Oriented Gradients

The Histogram of Oriented Gradients (HOG) is a well-known popular technique for features extraction related to object detection tasks. The main idea underlying the working of HOG is to compute gradient information stored in the image. This mainly involves the use of filters to identify important edges, corners, and shapes within the image. The main steps involved in the working of HOG are as follows:

- (i) **Pre-Processing:** In this step, gamma correction and Gaussian smoothing is performed.

- (ii) **Computation of Gradients:** The gradients on the preprocessed image are calculated using filters in horizontal and vertical directions. This results in two gradient images for each input image.
- (iii) **Computation of Gradient Magnitude and Orientation:** For each pixel in the image, additional features i.e., magnitude and direction of gradients are calculated. The magnitude shows the strength of the gradient whereas orientation represents the direction of gradient.
- (iv) **Histogram Generation:** In this step, the image area is divided into grid of small cells and a histogram of gradient orientation is computed for each cell. There are generally 9 orientation bins used in the histogram.

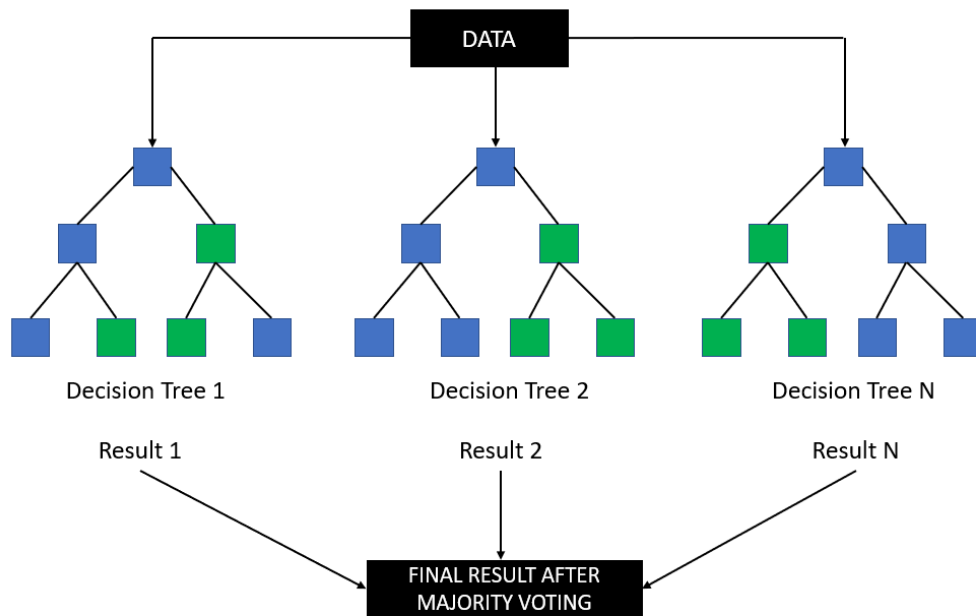


Figure 3.2: A schematic diagram to understand the working of Random Forest Models. The output of individual decision trees is aggregated, and final decision is computed based on majority votes. The output of many weak classifiers with random subsampling of training data can outperform individual decision trees.

3.7 Development of Machine Learning models

In this section, we briefly explain the machine learning models used in this study.

3.7.1 Decision Tree Classifier

Decision Trees, introduced by Ross Quinlan in 1986, are a subclass of supervised machine learning algorithms used to solve classification and regression problems. The main idea behind the working of decision trees is to split data in a recursive manner, based on most

important features. The main steps involved in the decision tree based algorithms are as follows:

- (i) The algorithm computes the disorder in the system (training data), and tries maximizing information gain or minimizing the entropy or Gini-index.
- (ii) The data divided into subsets become the branches or paths in the decision tree.
- (iii) Step (i) and (ii) are applied in a recursive manner until a stopping criteria becomes true. This is usually reaching to a specific depth of tree or achieving minimum samples per leaf.
- (iv) For each iteration, (i) to (ii), the algorithm assigns a decision label to each leaf node.

3.7.2 Support Vector Machine (SVM) Classifier

Support Vector Machines are a subclass of supervised algorithms. SVM based approaches are used solving regression and classification problems. Vapnik et al. proposed SVM algorithm in 1992. The main working of SVM is as follows:

- (i) Given two classes, comprising of red and blue points in 2D space, the SVM algorithm tries to find a hyperplane that best separates the two classes. The hyperplane uses the concept of “margins” which is actually the distance between hyperplane and closest points.
- (ii) The SVM algorithm iteratively expands the margins in order to find best-fit hyperplane.
- (iii) In many cases, data is not linear and it is not possible the classification problem using a straight line. Therefore, as a trick, the SVM procedure transforms data into a higher dimensional space.
- (iv) The hyperplane is computed in a higher dimensional space and then transformed to a 2D space for visualization.

3.7.3 Random Forest Classifier

Random Forest is a machine learning algorithm that employs multiple machine-learning algorithms to improve model’s performance. It is also known as ensemble learning model. The working of Random Forest based classifier can be explained as follows:

- i. The RF algorithm first selects a random subset from the data with replacements and generates the first decision tree.

- ii. Iteratively, the algorithm generates multiple decision trees and uses random sub-sampling of data for training.

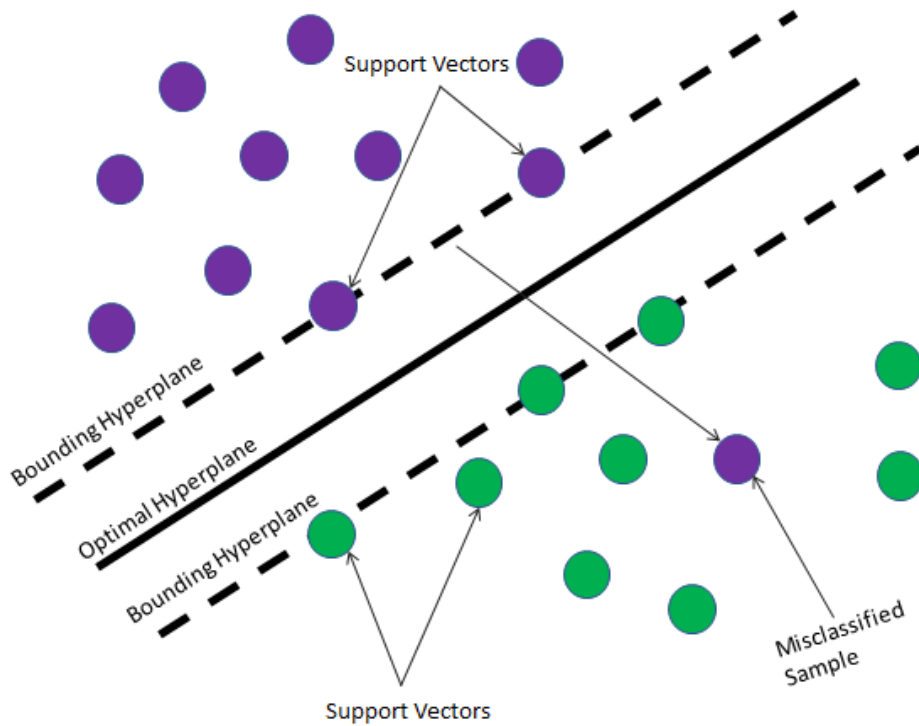


Figure 3.3: A schematic diagram to understand the working of Support Vector Machine (SVM). Instead of creating a decision boundary, SVM creates a hyper-plane using support vectors for better performance.

- iii. Step (i) and (ii) are repeated until a convergence criterion is met. This is usually when a specific depth of the tree is reached.

3.7.4 Classification using a Deep CNN model

One of the reasons the Convolution Neural Network (CNN) based models have been widely used in variety of application domains is their ability to extract features from raw images and videos. The process of convolution is central to automated features detection and extraction. The convolution operation uses a filter which is like a 3x3 matrix or window. The filter slides across the image data to find specific features of interest such as lines, edges and corners to larger objects and scenes. The lower layers of CNN focus on detection of lines and edges, while the higher layers focus on detection of objects and scenes.

After preprocessing and scaling on custom data, the training images are provided to the CNN model. Once the training process is completed, the performance of the model is evaluated on test data.

3.8 Hyper-parameters Tuning

There are several ways to improve the performance of machine learning algorithms. Some of these techniques include augmenting the training data and better features selection and engineering methods. However, parameters of a model of machine learning algorithms also influence the performance. One way to improve the performance of machine learning models is through hyper-parameter tuning. This involves selection of optimized values of model parameters by performing exhaustive or iterative experimentation. It is done by scanning the range of values of different model variables and testing each combination. To accomplish this, the data is divided into three subsets i.e., training data, validation data and test data. During the training process, validation data is used to check if the model is converging or improving accuracy for unseen data. This step can also be combined with hyper-parameter tuning. Several techniques are used for hyper-parameter tuning. The widely used techniques are GridScan, RandomScan and Bayesian optimization.

In this study, hyper-parameter tuning has been done for the following models:

- (i) For decision tree model, GridScan has been used on maximum depth, maximum and minimum samples required for splitting and maximum number of features.
- (ii) For RandomForest algorithm, GridScan is performed on the number of estimators, depth of individual decision trees, minimum and maximum number of samples required for splitting and maximum number of features using for training the model.
- (iii) For training and optimization of SVM, GridScan is performed to see the performance of various Kernel functions such as Linear, Polynomial and Radial Basis Function (RBF). Moreover, parameters tuning is also performed over “Gamma” variable to achieve a more accurate decision boundary.
- (iv) For the CNN model, we compare different settings for number of convolution layers, number of filters, filter size, stride and pooling.

3.9 Performance Evaluation

After applying the machine learning models to the data, different evaluation measures can be used to check how accurate or correct the model is. The problem under consideration for this research belongs to the supervised learning category, classification. The measures used in this methodology are accuracy, precision, recall, F1-score and AUC.

Accuracy: Accuracy as a performance measure for machine learning models is widely used. It involves computing the ratio of correctly predicted observations to the total number of predictions produced by a model and serves as a summary of a classification problem's performance. Nonetheless, accuracy is appropriate only for datasets that exhibit genuine balance, with equal representation of class labels. Accuracy is calculated as follows:

$$Accuracy = \frac{\text{correct predictions}}{\text{total prediction}}$$

Confusion Matrix: To calculate the performance of machine learning models, data of correct and false predictions is sorted in the form of a matrix plotting ground truth against predicted labels (Figure 4). It contains the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).

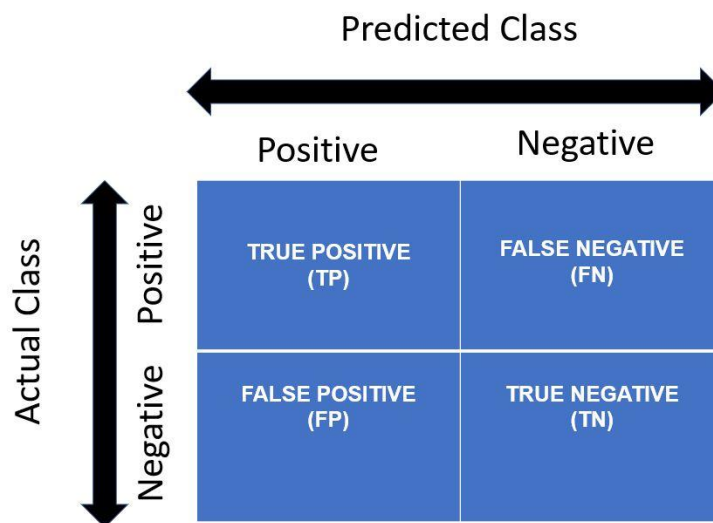


Figure 3.4: The confusion matrix is used to depict algorithm performance; negative and positive class labels are written on both the top and left sides of the matrix, the label on the top is referred to as the predicted label, and the label on the left is referred to as the actual class label

An example of cloud classification, there are four possible class labels i.e., “Clear Sky”, “Overcast”, “Pattern Clouds” and “Veil Clouds”. When an image of the “Clear Sky” class is

fed to the machine learning classifier and the output is also “Clear Sky”, it means that the prediction matches exactly with the ground truth and the output of the classifier is correct.

Precision

It is defined as the ratio of True positives over all the positives.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall

The proportion of observations is predicted as positive, and they are actually positive. It is more valuable when false negatives are crucial. It measures the quality based on the mistakes our model made. In the case of disease prediction, the cost of a false negative predicts a patient with no disease, which is deadly for the patient as the disease worsens. So, in that case, more focus would be on recall.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

In real-world problems, most of the time, one is either interested in precision or recall, depending on the cost or damage caused by the false positive or false negative.

F1-score

F1-score is the harmonic mean of both the precision and recall metric. After understanding the importance of the above terms, it has been realized that a trade-off exists between precision and recall. When both are equally important, the f1-score measure is used. If one has a low value, then the resultant f1 score has a low value.

$$F1_{score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

AUROC Curve

AUROC stands for the area under the receiver operator characteristic. It is a graph that evaluates the performance of the classification model. The graph is plotted between two values, true positive rate and false positive rate. The ratio of positive class examples that are predicted correctly is referred to as a true positive rate or recall.

On the other hand, the ratios of negative class examples are incorrectly predicted. It is an easy way to summarize the model's overall performance. A model with a higher AUC score is the best. The figure that shows the AUROC curve with two thresholds is presented below:

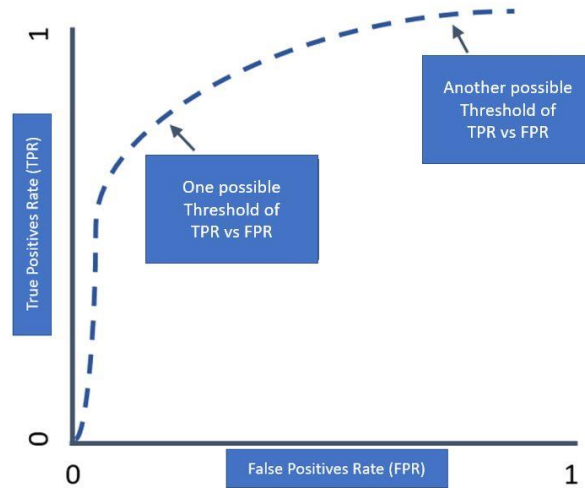


Figure 3.5: The Area under the Receiver Operating Characteristic curve with various thresholds is displayed in the figure; a graph is plotted between true positive and true negative rates; the greater the area is under the ROC curve illustrates an increase in the performance

Chapter 4

Results and Discussion

In this chapter, we present the results of the proposed machine learning based classification discussed in chapter 3.

4.1 Data Visualization

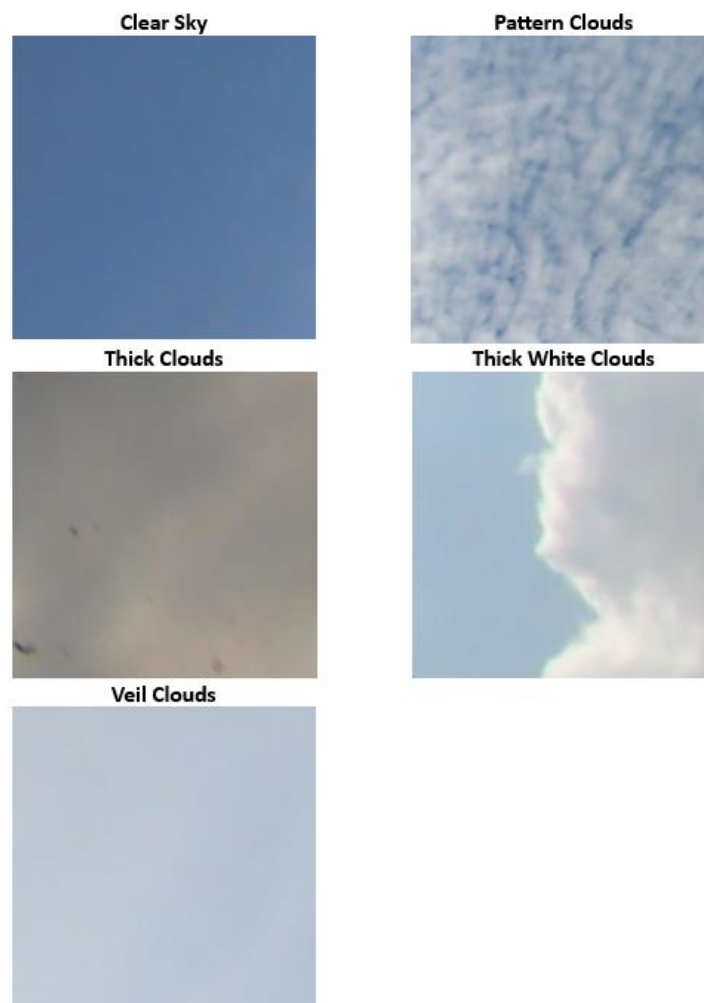


Figure 4.1: Five Random images from each class visualized using python matplotlib library.

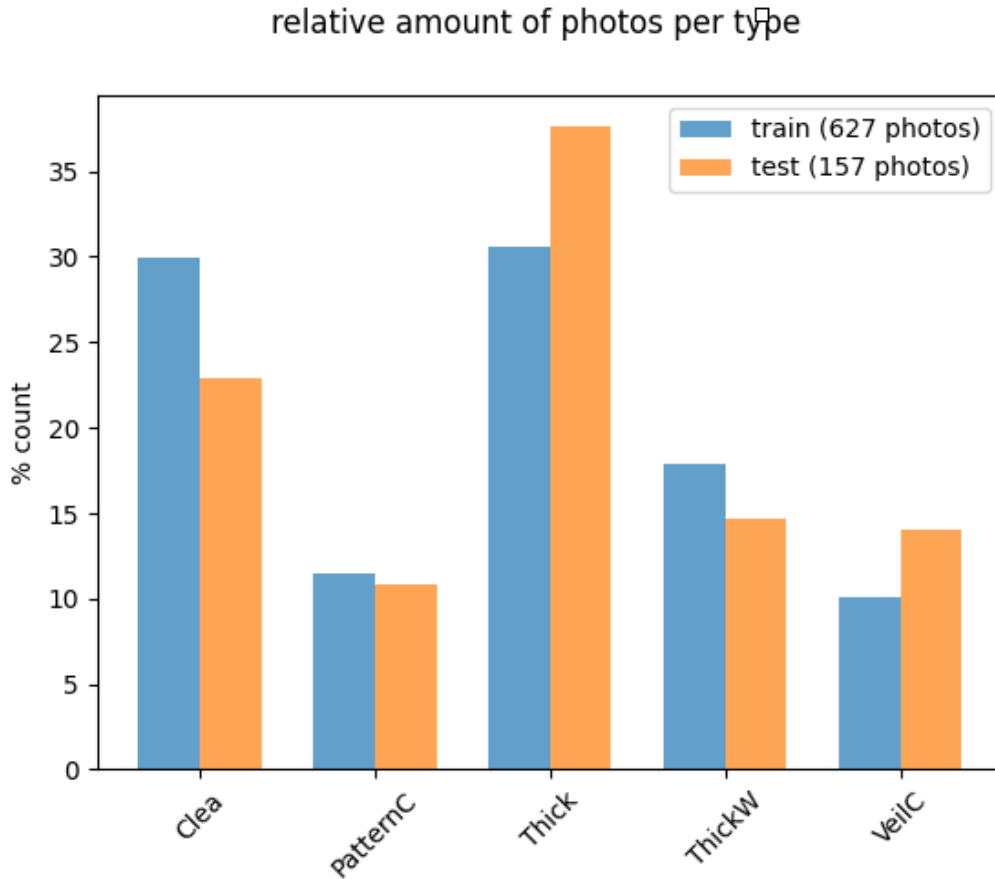


Figure 4.2: Relative number of Images for each class in SWIMCAT data set

4.2 Classification using Stochastic Gradient Descent Gradient (SGD) Algorithm

We started our analysis by using the Stochastic Gradient Descent (SGD) classifier to classify the cloud image we applied a preprocessing pipeline to prepare the image data for classification. The pipeline was implemented in Python, using the scikit-image and scikit-learn libraries. The preprocessing pipeline involved converting the RGB images to grayscale, extracting features using Histogram of Oriented Gradients (HOG), and scaling the feature vector using StandardScaler. It was implemented using custom transformers, which were designed to be applied in sequence. The pipeline was applied to the input data (X_{train}), and the final prepared data was stored in the $X_{train_prepared}$ variable.

In order ensure that structure of the data is in line with Python Keras library requirements, `shape()` function is used. The preprocessing allows the data to be used with Keras Tensorflow framework. The classification model is trained by the transformed data obtained after the preprocessing stage. The model shows high accuracy confirming the effectiveness of the preprocessing pipeline.

We varied the learning rate, batch size, and regularization strength hyper-parameters and used 5-fold cross-validation to evaluate the performance of each combination of hyper-parameters. The best set of hyper-parameters was found to be: learning rate = 0.01, batch size = 32, and regularization strength = 0.001.

After tuning the hyper-parameters, the overall accuracy of the SGD classifier improved from 84% to 86%, an improvement of 2%. The precision and recall of the classifier also improved, with precision increasing to 0.90 and recall increasing to 0.86. These results show that hyper-parameter tuning can significantly improve the performance of the SGD classifier.

4.3 Hyper-Parameters Tuning for Best Estimator

The code is setting up a parameter grid for hyper-parameter tuning using GridSearchCV from the scikit-learn library in Python.

The parameter grid contains two dictionaries, each specifying different sets of hyper-parameters to tune for the machine learning pipeline. The pipeline consists of a feature extraction step using the HOG (Histogram of Oriented Gradients) algorithm, followed by a classification step using either a Stochastic Gradient Descent (SGD) classifier or a Support Vector Machine (SVM) classifier with a radial basis function (RBF) kernel.

The hyper-parameters being tuned for the HOG feature extraction step are:

- orientations: the number of orientation bins in the HOG descriptor
- cells_per_block: the number of cells per block in the HOG descriptor
- pixels_per_cell: the size of each cell in the HOG descriptor in pixels

```

param_grid = [
  {
    'hogify__orientations': [8, 9],
    'hogify__cells_per_block': [(2, 2), (3, 3)],
    'hogify__pixels_per_cell': [(8, 8), (10, 10), (12, 12)]
  },
  {
    'hogify__orientations': [8],
    'hogify__cells_per_block': [(3, 3)],
    'hogify__pixels_per_cell': [(8, 8)],
    'classify': [
      SGDClassifier(random_state=42, max_iter=1000, tol=1e-3),
      svm.SVC(kernel='rbf')
    ]
  }
]

```

Figure 4.3: Parameter grid for hyper-parameters tuning using Grid-Search CV

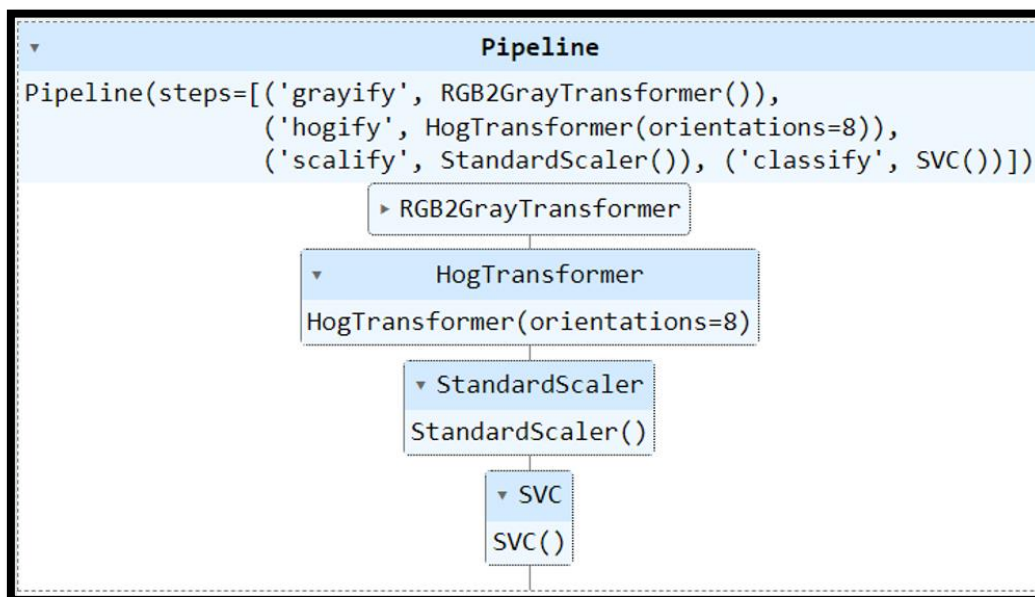


Figure 4.4: Configuration of the best estimator resulted after Grid Search

The first dictionary in the parameter grid specifies that the hyper-parameters should be varied over multiple values, with 8 or 9 orientations, 2x2 or 3x3 cells per block, and 8x8, 10x10, or 12x12 pixels per cell.

The second dictionary in the parameter grid specifies that the HOG hyper-parameters should be set to fixed values (8 orientations, 3x3 cells per block, and 8x8 pixels per cell), and the classify step should be tuned instead. Two classifiers are specified in the classify list: an SGD

classifier and an SVM classifier with an RBF kernel. The hyper-parameters being tuned for the classifiers are not specified in this code snippet, but would typically include regularization strength, learning rate, or kernel parameters, depending on the classifier being used.

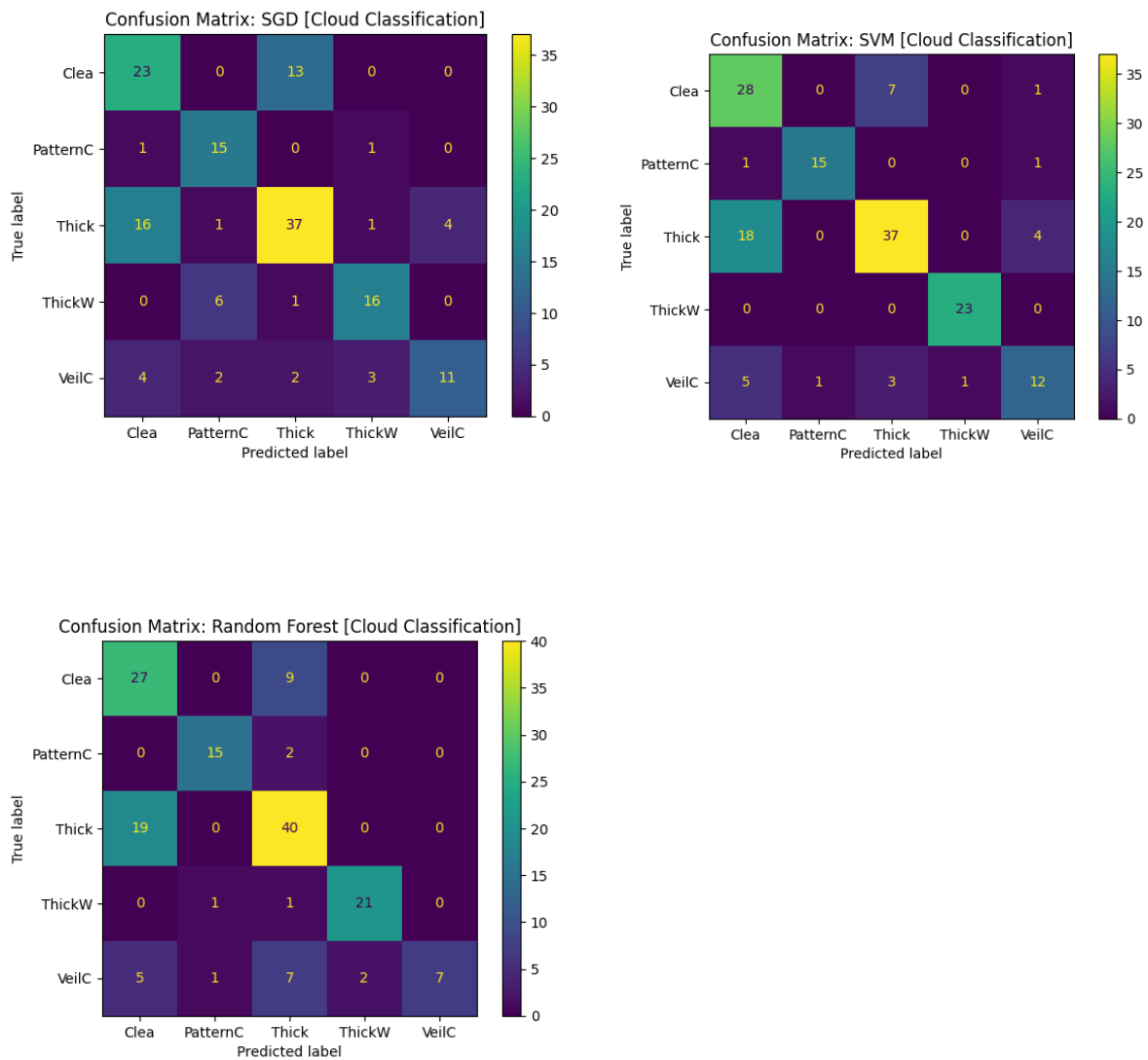


Figure 4.5: Confusion Matrices of three machine learning models used for classification, i.e. the SGD classifier, Support Vector Classifier and Random Forest model

The confusion matrix above summarizes the performance of a classification model that was used to classify 157 instances into five different categories: Clea, PatternC, Thick, ThickW, and VeilC. The matrix shows that the model achieved an overall accuracy of 65%. It means that 65% of the instances were classified correctly. However, the performance varied for each class, with precision ranging from 0.52 to 0.76, recall ranging from 0.50 to 0.88, and F1-score ranging from 0.57 to 0.73.

The results show that the model performed well for some classes, such as PatternC and ThickW, achieving precision and recall above 0.70 and F1-score above 0.73. However, the model had lower precision and recall for other classes, such as Clea and VeilC, with values below 0.60. These results suggest that the model may be better suited for certain classes than others, and further improvements could be made to enhance its performance for all classes. The detailed summary of the classification model is presented in the form of confusion matrix. By using the number of True Positives, True Negatives, False Positives and False Negatives, important performance evaluation matrices such as Accuracy, Precision, Recall and F1 scores can be calculated. The confusion matrix in Figure 4.5 shows that three different models i.e. SGD classifier, Support Vector Classifier (SVC) and Random Forest algorithm were used to classify 157 test cases. The results show that more than 70% of the instances were correctly classified by the SVC and Random Forest model. These models performed reasonably well as the precision ranges from 0.53 to 1.0 for various classes, recall ranges from 0.3 to 0.9 and F1-score is between 0.48 to 0.91.

The model showed the highest precision and recall for the ThickW class with a score of 0.91 for both precision and recall, indicating that the model was highly accurate at classifying instances belonging to this class. However, the model had the lowest recall score of 0.32 for VeilC, indicating that the model struggled to accurately classify instances belonging to this class. The macro-average of the model is 0.72, indicating that the model is quite reliable overall. Finally, the weighted average of the model is 0.69, which suggests that the model is not biased towards any particular class but has slightly lower performance for some of the smaller classes. Overall, the confusion matrix provides a comprehensive and informative summary of the classification model's performance, highlighting areas where improvements could be made.

	Precision	Recall	F1-Score
ClearSky	0.52	0.64	0.57
PatternClouds	0.62	0.88	0.73
ThickClouds	0.70	0.63	0.66
ThickWhite	0.76	0.70	0.73
VeilClouds	0.73	0.50	0.59
Accuracy = 0.65			
Macro Avg	0.67	0.67	0.66
Weighted Avg	0.66	0.65	0.65

Table 4.1: Performance of Stochastic Gradient Descent (SGD) classifier

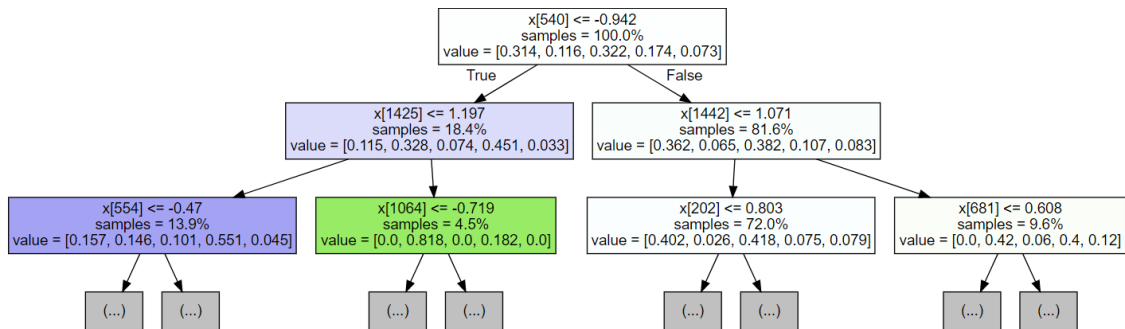


Figure 4.6: Visualization of a Decision Tree in the Random Forest Model

	Precision	Recall	F1-Score
ClearSky	0.54	0.78	0.64
PatternClouds	0.94	0.88	0.91
ThickClouds	0.79	0.63	0.70
ThickWhite	0.96	1.00	0.98
VeilClouds	0.67	0.55	0.60
Accuracy = 0.7324			
Macro Avg	0.78	0.77	0.76
Weighted Avg	0.75	0.73	0.73

Table 4.2: Performance of Support Vector Machine (SVM)

	Precision	Recall	F1-Score
ClearSky	0.53	0.75	0.62
PatternClouds	0.88	0.88	0.88
ThickClouds	0.68	0.68	0.68
ThickWhite	0.91	0.91	0.91
VeilClouds	1.00	0.32	0.48
Accuracy = 0.7006			
Macro Avg	0.80	0.71	0.72
Weighted Avg	0.75	0.70	0.69

Table 4.3: Performance of Random Forest Classifier (RFC)

4.4 Classification using Convolutional Neural Networks

We designed a custom convolutional neural network (CNN) model for classification of cloud models having five classes.

The input image has a shape of (125, 125, 3), where the first two dimensions represent the image height and width, and the third dimension represents the color channels (RGB). The batch size is set to 32, which means that 32 images will be processed in each iteration of the model training process.

The model architecture consists of two convolutional layers with 32 and 64 filters, respectively, and each filter has a kernel size of 3x3. The activation function used for both convolutional layers is the Rectified Linear Unit (ReLU) function, which introduces non-linearity to the model. Between the two convolutional layers, there is a max pooling layer with a pool size of 2x2, which reduces the spatial size of the feature maps.

After the max pooling layer, there is a dropout layer with a rate of 0.2, which randomly drops out 20% of the neurons in the layer to reduce overfitting. The flattened output from the dropout layer is then passed to two fully connected (dense) layers with 128 and 5 neurons, respectively. The activation function for the first dense layer is ReLU, while the second dense layer uses a softmax activation function, which produces a probability distribution over the five classes.

During training, the model is optimized using categorical cross-entropy loss and the Adam optimizer. The model is trained for 10 epochs, which means that the entire training dataset is passed through the model 10 times.

The CNN model was trained for 100 epochs and the accuracy and loss plots are shown in Figure 4.7 and Figure 4.8 respectively. These results show that after 20 epochs, the performance gap between the training and test data increases. The model shows high accuracy on the training data whereas the validation accuracy starts to drop with increase in validation loss.

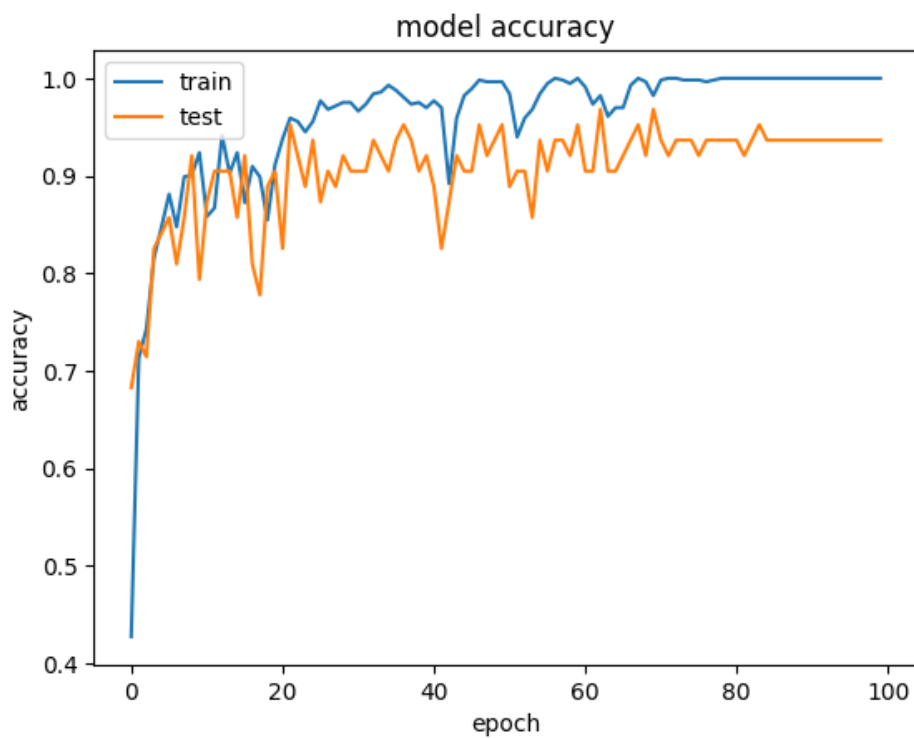


Figure 4.7: Accuracy plot for training and test data using CNN classification without a Dropout layer

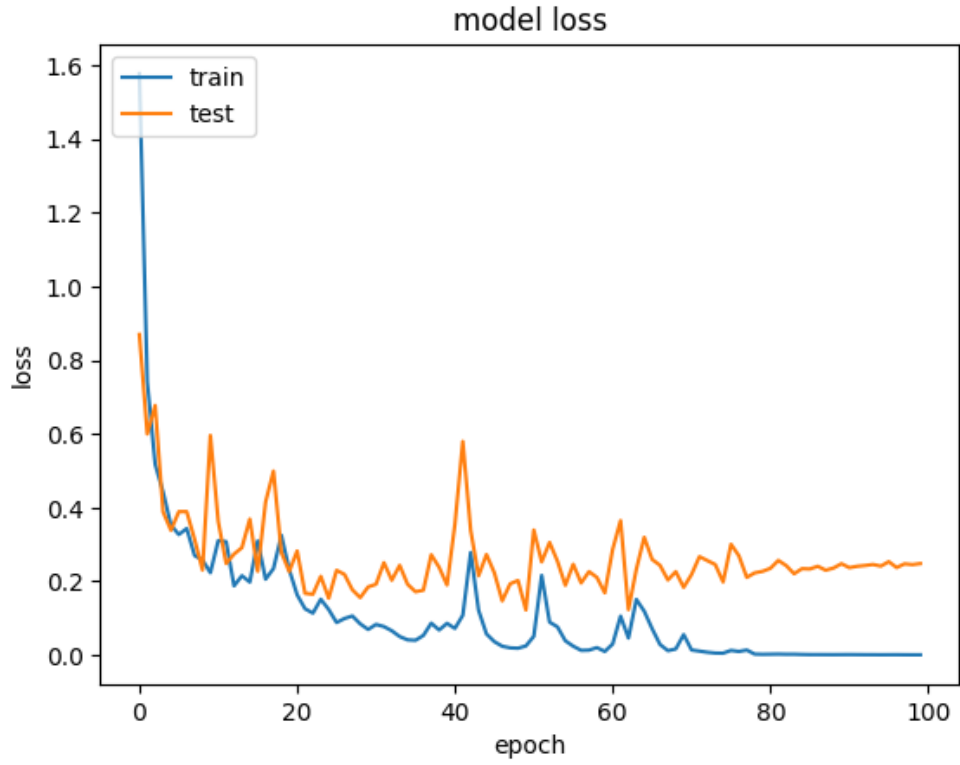


Figure 4.8: Loss plot for training and test data using CNN classification without a Dropout layer

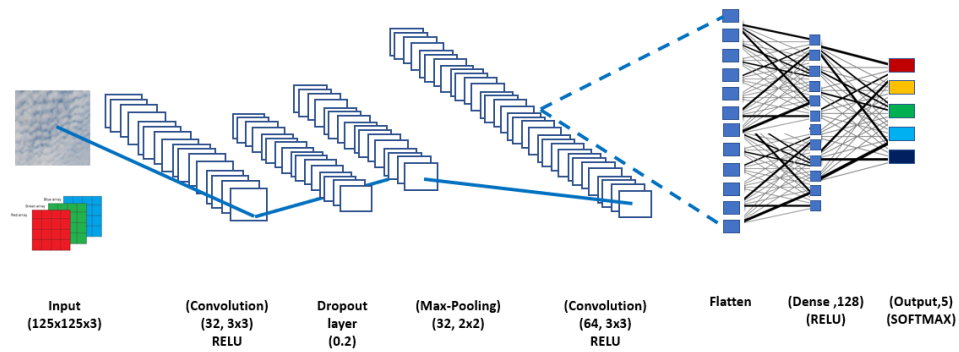


Figure 4.9: Revised CNN model with 20% drop out layer to solve the problem of overfitting.

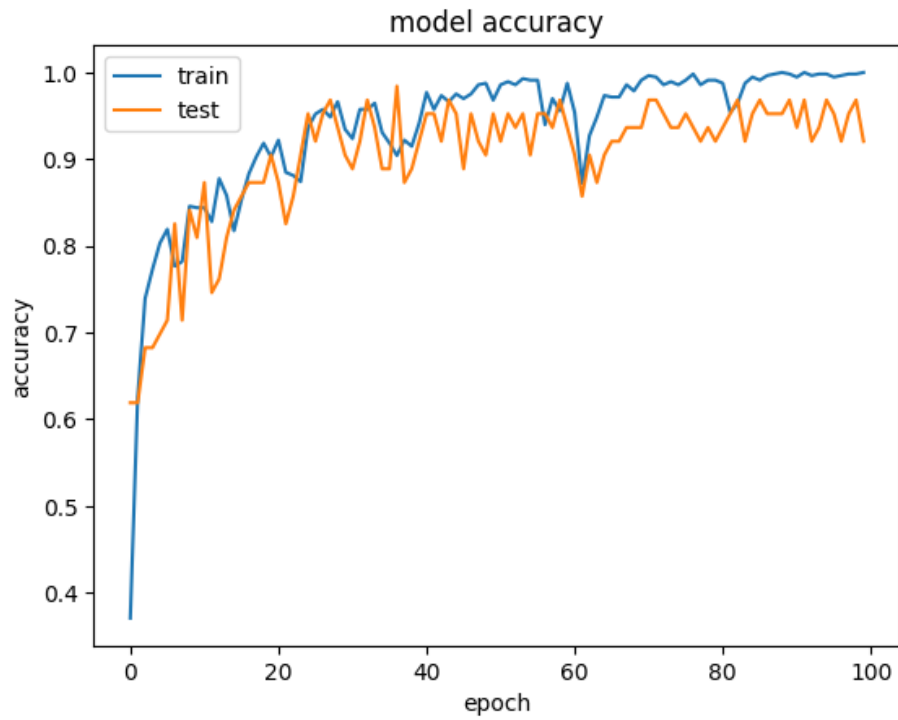


Figure 4.10: Accuracy plot for training and test data using CNN classification by adding 20 percent dropout.

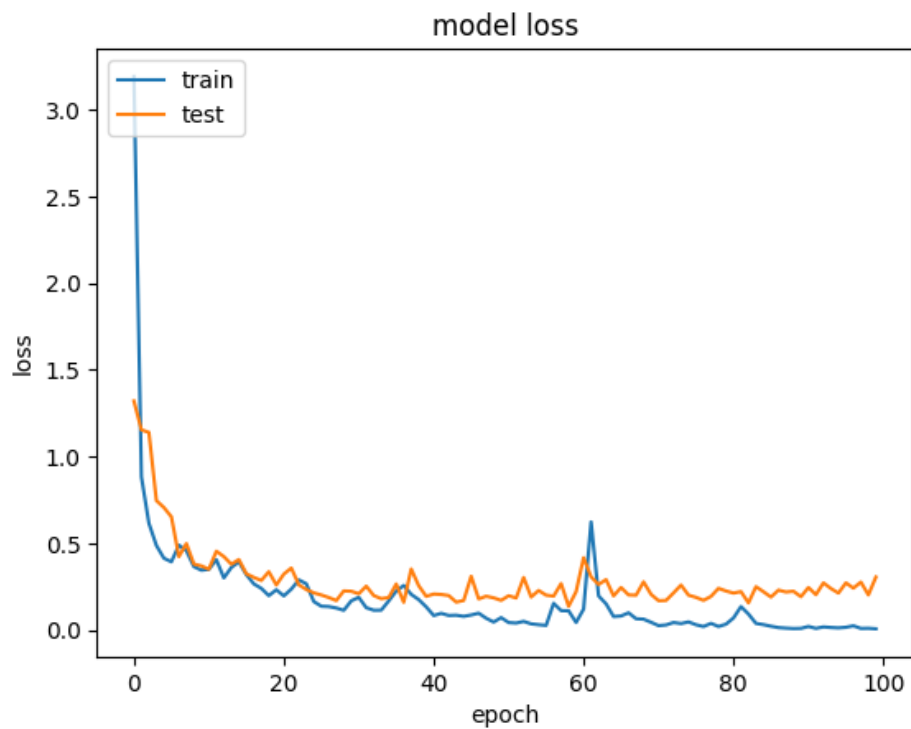


Figure 4.11: Loss plot for training and test data using CNN classification by adding 20 percent dropout

To fix the issue of model over-fitting, we introduce a dropout layer as shown in the code listing (Table) below. The revised CNN architecture with dropout layer is shown in Figure 4.9.

Figures 4.10 and 4.11 show that the model is able to generalize and overcome the overfitting problem and the performance gap is reduced.

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Dropout
# Define hyperparameters
batch_size = 32
epochs = 10
input_shape = (125, 125, 3)
num_classes = 5

# Create model
model2 = Sequential()
model2.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
input_shape=input_shape))
model2.add(MaxPooling2D(pool_size=(2, 2)))
model2.add(Dropout(0.2))
model2.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model2.add(MaxPooling2D(pool_size=(2, 2)))
model2.add(Flatten())
model2.add(Dense(128, activation='relu'))
model2.add(Dense(num_classes, activation='softmax'))
```

Chapter 5

Conclusion and Future Work

The idea of a cost-effective approach for weather prediction can be beneficial in many cases such as extreme weather event prediction, local weather prediction for farmers etc. The previous approaches for accurate weather prediction have relied on the use of sensors (temperature, humidity, air-pressure etc.). However, these approaches require the use and fusion of big data. In the recent years, a lot of progress has been made in computer vision, especially towards the development of accurate models for classification, detection, and segmentation. This study explores the use of ground-based camera images towards weather prediction. The major computational process in this direction is the classification of cloud images. This study develops a custom dataset by using existing cloud images from Singapore Wholesky Imaging CATegories (SWIMCAT) dataset and augmenting it with Kaggle data. Moreover, additional images are obtained using web-scraping.

Four machine learning algorithms i-e. Stochastic Gradient Descent (SGD), Support Vector Classifier (SVC), Random Forest Algorithm (RF) and CNN based deep network architecture have been used for classification. The conventional machine learning algorithms i-e. SGD, SVC and RF were optimized using hyper-parameter tuning and Grid Scan search. The results show accuracy in the range of 65% to 76%. In order to further improve the performance, a custom CNN architecture was used for classification. The model demonstrated 95% accuracy. However, the training history showed large performance gap between training and testing data. To address this problem of over-fitting, modifications were made in the CNN architecture, resulting in a generalized accuracy of 98.5%.

The work done in this study, shows that ground-based images can be used for classification of clouds. This work can be extended in multiple directions. Firstly, the machine learning model can be interfaced with off-the-shelf ground camera and an IoT processor to develop a functional prototype. Secondly, we aim to introduce a new architecture i-e. Network of Receiver Station (**NORAN**) by using multiple ground-based image sensors for wider area coverage and accurate weather forecasting.

REFERENCES

- [1] Stephens, Graeme L. "The useful pursuit of shadows: the study of clouds has profoundly influenced science and human culture and stands poised to lead climate science forward again." *American Scientist* 91.5 (2003): 442-449.
- [2] Land, Brett Eadon. *Cloud: The Religious History of a Symbol*. Diss. University of California, Santa Barbara, 2018.
- [3] Brehm, Barbara L. "Weather: Operational Considerations on the Battlefield." (1991).
- [4] Ward, Robert De C. "Weather controls over the fighting in the Italian war zone." *The Scientific Monthly* (1918): 97-105.
- [5] Parolini, Giuditta. "Weather, climate, and agriculture: Historical contributions and perspectives from agricultural meteorology." *Wiley Interdisciplinary Reviews: Climate Change* 13.3 (2022): e766.
- [6] Elits, M. "The role of weather and weather forecasting in agriculture." *DTN* (2018).
- [7] Pidcock, R., R. Pearce, and R. McSweeney. "How climate change affects extreme weather around the world." (2017).
- [8] Fahad, Shah, and Jianling Wang. "Climate change, vulnerability, and its impacts in rural Pakistan: a review." *Environmental Science and Pollution Research* 27 (2020): 1334-1338.
- [9] Rangno, Arthur L. "The Classification of Clouds." *Handbook of Weather, Climate, and Water: Dynamics, Climate, Physical Meteorology, Weather Systems, and Measurements* (2003): 387-405.
- [10] Goodman, A. H., and A. Henderson-Sellers. "Cloud detection and analysis: A review of recent progress." *Atmospheric Research* 21.3-4 (1988): 203-228.
- [11] Kazantzidis, Andreas, et al. "Cloud detection and classification with the use of whole-sky ground-based images." *Atmospheric Research* 113 (2012): 80-88.

- [12] Mahajan, Seema, and Bhavin Fataniya. "Cloud classification: principles and applications." *International Journal of Hydrology Science and Technology* 12.2 (2021): 202-213.
- [13] Ohring, G., et al. "Applications of satellite remote sensing in numerical weather and climate prediction." *Advances in Space Research* 30.11 (2002): 2433-2439.
- [14] Pierro, Marco, et al. "Data-driven upscaling methods for regional photovoltaic power estimation and forecast using satellite and numerical weather prediction data." *Solar Energy* 158 (2017): 1026-1038.
- [15] Kalsi, S. R. "Satellite based weather forecasting." *Satellite remote sensing and GIS applications in agricultural meteorology* 331 (2002).
- [16] Saunders, Roger. "The use of satellite data in numerical weather prediction." *Weather* 76.3 (2021): 95-97.
- [17] Neiburger, Morris, and Harry Wexler. "Weather satellites." *Scientific American* 205.1 (1961): 80-97.
- [18] Li, Zhiwei, et al. "Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors." *ISPRS Journal of Photogrammetry and Remote Sensing* 150 (2019): 197-212.
- [19] Ahmed, Tashin, and Noor Hossain Nuri Sabab. "Classification and understanding of cloud structures via satellite images with EfficientUNet." *SN Computer Science* 3 (2022): 1-11.
- [20] Dev, Soumyabrata, Yee Hui Lee, and Stefan Winkler. "Color-based segmentation of sky/cloud images from ground-based cameras." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10.1 (2016): 231-242.
- [21] Papin, Christophe, Patrick Bouthemy, and Guy Rochard. "Unsupervised segmentation of low clouds from infrared METEOSAT images based on a contextual spatio-temporal labeling approach." *IEEE Transactions on Geoscience and Remote Sensing* 40.1 (2002): 104-114.

- [22] Mahrooghy, Majid, et al. "On the use of a cluster ensemble cloud classification technique in satellite precipitation estimation." *IEEE journal of selected topics in applied earth observations and remote sensing* 5.5 (2012): 1356-1363.
- [23] Weiss, John M., Sundar A. Christopher, and Ronald M. Welch. "Automatic contrail detection and segmentation." *IEEE transactions on geoscience and remote sensing* 36.5 (1998): 1609-1619.
- [24] Farmer, Michael E., and Anil K. Jain. "A wrapper-based approach to image segmentation and classification." *IEEE transactions on image processing* 14.12 (2005): 2060-2072.
- [25] Calbo, Josep, and Jeff Sabburg. "Feature extraction from whole-sky ground-based images for cloud-type recognition." *Journal of Atmospheric and Oceanic Technology* 25.1 (2008): 3-14.
- [26] Calbó Angrill, Josep, and Josep Abel González Gutiérrez. "Empirical studies of cloud effects on UV radiation: A review." © *Reviews of Geophysics*, 2005, vol. 43, núm. 2, p. RG2002 (2005).
- [27] Change, IPCC Climate. "The scientific basis." (2001).
- [28] Chen, Z., D. Zen, and Q. Zhang. "Sky model study using fuzzy mathematics." *Journal of the Illuminating Engineering Society* 23.1 (1994): 52-58.
- [29] World Meteorological Organization. *International cloud atlas*. World Meteorological Organization, 2017.
- [30] World Meteorological Organization. *International cloud atlas vol II*. WMO, Hydrology & Water Resources Department, 1987.
- [31] Zhuo, Wen, Zhiguo Cao, and Yang Xiao. "Cloud classification of ground-based images using texture–structure features." *Journal of Atmospheric and Oceanic Technology* 31.1 (2014): 79-92.
- [32] Shields, Janet E., et al. "Daylight visible/NIR whole-sky imagers for cloud and radiance monitoring in support of UV research programs." *Ultraviolet Ground-and Space-Based Measurements, Models, and Effects III*. Vol. 5156. SPIE, 2003.

- [33] Heinle, Anna, Andreas Macke, and Anand Srivastav. "Automatic cloud classification of whole sky images." *Atmospheric Measurement Techniques* 3.3 (2010): 557-567.
- [34] Long, C. N., D. W. Slater, and Tim P. Tooman. Total sky imager model 880 status and testing results. Richland, WA, USA: Pacific Northwest National Laboratory, 2001.
- [35] Pfister, G., et al. "Cloud coverage based on all-sky imaging and its impact on surface solar irradiance." *Journal of applied meteorology and climatology* 42.10 (2003): 1421-1434.
- [36] Lu, D., J. Huo, and W. Zhang. "All-sky visible and infrared images for cloud macro characteristics observation." *Proc. 14th Int. Conf. on Clouds and Precipitation*. Vol. 2. 2004.
- [37] Long, Charles N., et al. "Retrieving cloud characteristics from ground-based daytime color all-sky images." *Journal of Atmospheric and Oceanic Technology* 23.5 (2006): 633-652.
- [38] Murdock, Calvin, Nathan Jacobs, and Robert Pless. "Building dynamic cloud maps from the ground up." *Proceedings of the IEEE International Conference on Computer Vision*. 2015.
- [39] Murdock, Calvin, Nathan Jacobs, and Robert Pless. "Webcam2satellite: Estimating cloud maps from webcam imagery." *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2013.
- [40] Doretto, Gianfranco, et al. "Dynamic textures." *International Journal of Computer Vision* 51 (2003): 91-109.
- [41] Fitzgibbon, Andrew W. "Stochastic rigidity: Image registration for nowhere-static scenes." *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 1. IEEE, 2001.
- [42] Soatto, Stefano, Gianfranco Doretto, and Ying Nian Wu. "Dynamic textures." *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 2. IEEE, 2001.
- [43] Šinko, Martin, et al. "Development of a system for collecting and processing sky images and meteorological data used for weather prediction." *Transportation Research Procedia* 40 (2019): 1548-1554.

- [44] Longo, M., et al. "Towards the development of residential smart districts: The role of EVs." 2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe). IEEE, 2017.
- [45] Urquhart, B., et al. "Sky imaging systems for short-term solar forecasting, chapter 9 in: Solar Energy Forecasting and Resource Assessment, edited by: Kleissl, J." (2013).
- [46] Alonso-Montesinos, J., and F. J. Batlles. "The use of a sky camera for solar radiation estimation based on digital image processing." *Energy* 90 (2015): 377-386.
- [47] Zhen, Zhao, et al. "Research on a cloud image forecasting approach for solar power forecasting." *Energy Procedia* 142 (2017): 362-368.
- [48] Tapakis, R., and A. G. Charalambides. "Equipment and methodologies for cloud detection and classification: A review." *Solar Energy* 95 (2013): 392-430.
- [49] Tran-Trung, Kiet, Ha Duong Thi Hong, and Vinh Truong Hoang. "Weather Forecast Based on Color Cloud Image Recognition under the Combination of Local Image Descriptor and Histogram Selection." *Electronics* 11.21 (2022): 3460.
- [50] Feng, Cong, et al. "Convolutional neural networks for intra-hour solar forecasting based on sky image sequences." *Applied Energy* 310 (2022): 118438.
- [51] Niccolai, Alessandro, et al. "Very Short-Term Forecast: Different Classification Methods of the Whole Sky Camera Images for Sudden PV Power Variations Detection." *Energies* 15.24 (2022): 9433.
- [52] Brahma, Banalaxmi, and Rajesh Wadhvani. "Solar irradiance forecasting based on deep learning methodologies and multi-site data." *Symmetry* 12.11 (2020): 1830.
- [53] Jang, Han Seung, et al. "Solar power prediction based on satellite images and support vector machine." *IEEE Transactions on Sustainable Energy* 7.3 (2016): 1255-1263.
- [54] Cao, Yingyue, and Hanpeng Yang. "Weather Prediction using Cloud's Images." 2022 International Conference on Big Data, Information and Computer Network (BDICN). IEEE, 2022.

- [55] Marais, Willem J., et al. "Leveraging spatial textures, through machine learning, to identify aerosols and distinct cloud types from multispectral observations." *Atmospheric Measurement Techniques* 13.10 (2020): 5459-5480.
- [56] Sheela, M. Sahaya, et al. "Weather and Climate Forecasting System for Cultivation using Naive's Algorithm." *2022 2nd International Conference on Computing and Information Technology (ICCIT)*. IEEE, 2022.
- [57] Turner, B. J., I. Zawadzki, and U. Germann. "Predictability of precipitation from continental radar images. Part III: Operational nowcasting implementation (MAPLE)." *Journal of Applied Meteorology and Climatology* 43.2 (2004): 231-248.
- [58] Lai, Can, et al. "The cloud images classification based on convolutional neural network." *2019 International Conference on Meteorology Observations (ICMO)*. IEEE, 2019.
- [59] Souza-Echer, Mariza Pereira, et al. "A simple method for the assessment of the cloud cover state in high-latitude regions by a ground-based digital camera." *Journal of Atmospheric and Oceanic Technology* 23.3 (2006): 437-447.
- [60] Seluchi, Marcelo E., and Sin Chan Chou. "Evaluation of two Eta Model versions for weather forecast over South America." *Geofísica Internacional* 40.3 (2001): 219-237.
- [61] Sabburg, Jeff, and Joe Wong. "Evaluation of a ground-based sky camera system for use in surface irradiance measurement." *Journal of Atmospheric and Oceanic Technology* 16.6 (1999): 752-759.
- [62] Li, Q., Lu, W., & Yang, J. (2011). A hybrid thresholding algorithm for cloud detection on ground-based color images. *Journal of atmospheric and oceanic technology*, 28(10), 1286-1296.

#This code is only a template to load cloud data/images, organize files and develop a CNN model for classification.

```
"from google.colab import drive drive.mount("/content/drive")
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
import pathlib
```

```
ClearSky_path = pathlib.Path('/content/drive/MyDrive/Colab Notebooks/Data/ClearSky')
```

```
PatternCloud_path = pathlib.Path('/content/drive/MyDrive/Colab Notebooks/Data/PatternCloud')
```

```
ThickDark_path = pathlib.Path('/content/drive/MyDrive/Colab Notebooks/Data/ThickDark')
```

```
ThickWhite_path = pathlib.Path('/content/drive/MyDrive/Colab Notebooks/Data/ThickWhite')
```

```
VeilCloud_path = pathlib.Path('/content/drive/MyDrive/Colab Notebooks/Data/VeilCloud')
```

```
from PIL import Image
```

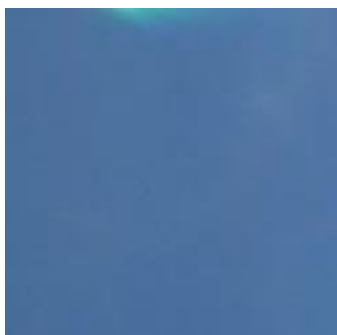
```
im1 = Image.open('/content/drive/MyDrive/Colab
```

```
Notebooks/Data/ClearSky/A_100img.png') im2 =
```

```
Image.open('/content/drive/MyDrive/Colab
```

```
Notebooks/Data/ClearSky/A_101img.png')
```

```
im1.show()
```



```

import cv2

import matplotlib.pyplot as plt %matplotlib inline

#reading image

img1 = cv2.imread('/content/drive/MyDrive/Colab
Notebooks/Data/ClearSky/A_100img.png') gray1 = cv2.cvtColor(img1,
cv2.COLOR_BGR2GRAY)

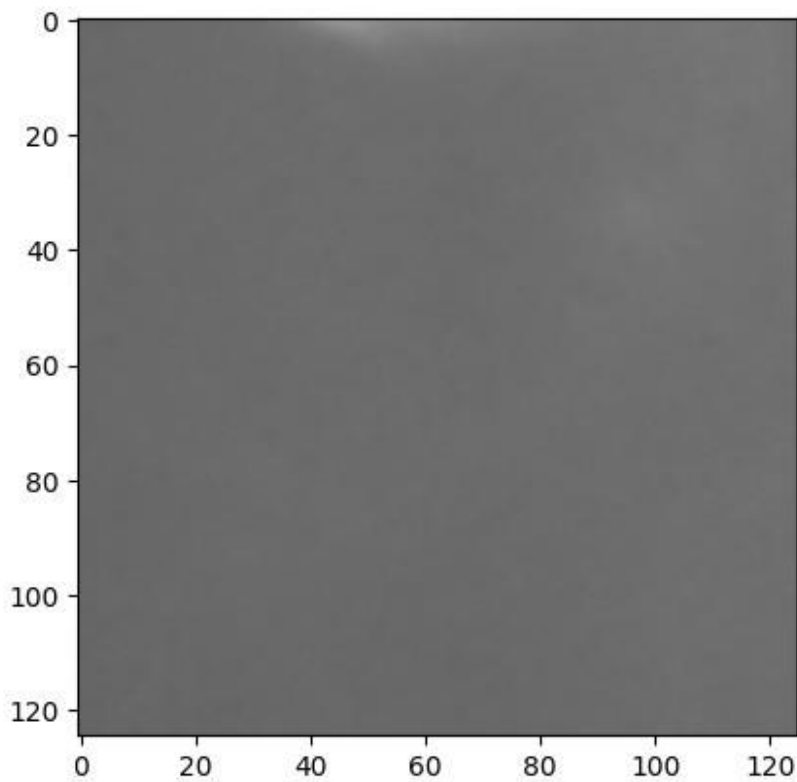
#keypoints

sift = cv2.xfeatures2d.SIFT_create() keypoints_1, descriptors_1 =
sift.detectAndCompute(img1,None)

img_1 = cv2.drawKeypoints(gray1,keypoints_1,img1) plt.imshow(img_1)

<matplotlib.image.AxesImage at 0x7efd3ebf7af0>

```



```

import glob as gb

data_dir = list(gb.glob('/content/drive/MyDrive/Colab
Notebooks/Data/*')) data_dir

['/content/drive/MyDrive/Colab Notebooks/Data/VeilCloud',

```

```

'/content/drive/MyDrive/Colab Notebooks/Data/ClearSky',
'/content/drive/MyDrive/Colab Notebooks/Data/ThickDark',
'/content/drive/MyDrive/Colab Notebooks/Data/PatternCloud',
'/content/drive/MyDrive/Colab Notebooks/Data/ThickWhite']

%matplotlib inline

import matplotlib.pyplot as plt import numpy
as np import os import pprint

pp = pprint.PrettyPrinter(indent=4)

X = np.array(data['data']) y =
np.array(data['label'])

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
shuffle=True, random_state=42,

)

from keras.models import Sequential

from keras.layers import Dense, Conv2D, Flatten

#create model model =
Sequential() #add model
layers

model.add(Conv2D(64, kernel_size=3, activation='relu', input_shape=(125,125,3)))

model.add(Conv2D(32, kernel_size=3, activation='relu')) model.add(Flatten())

model.add(Dense(5, activation='softmax'))

from keras import layers from keras import models
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3),
activation='relu',input_shape=(125, 125, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3),

```

```

activation='relu')) model.add(layers.MaxPooling2D((2,
2))) model.add(layers.Conv2D(128, (2, 2),
activation='relu')) model.add(layers.MaxPooling2D((2,
2))) model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))

model.add(layers.Dense(5, activation='softmax')) model.summary()

```

Model: "sequential_21"

Output Shape	Param #	Layer (type)
(None, 123, 123, 32)	896	conv2d_63 (Conv2D)
(None, 61, 61, 32)	0	max_pooling2d_49 (MaxPoolin g2D)
conv2d_64 (Conv2D)	(None, 59, 59, 64) 18496	
max_pooling2d_50 (MaxPoolin	(None, 29, 29, 64) 0	g2D)
conv2d_65 (Conv2D)	(None, 28, 28, 128) 32896	
max_pooling2d_51 (MaxPoolin	(None, 14, 14, 128) 0	g2D)
flatten_21 (Flatten)	(None, 25088) 0	dense_35
(Dense)	(None, 128) 3211392	dense_36 (Dense)
(None, 5)	645	

Total params: 3,264,325

Trainable params: 3,264,325

Non-trainable params: 0

```

X_train = X_train.astype("float32") / 255 #X_train = np.expand_dims(X_train, -1)

```

```

from keras.utils import to_categorical from sklearn.preprocessing
import LabelEncoder

```

```

# Create LabelEncoder object label_encoder = LabelEncoder()

```

```

# Encode target variable

```

```

y_train_encoded = label_encoder.fit_transform(y_train)

```

```

print(y_train_encoded.shape) y_train_OHC = to_categorical(y_train_encoded)

```

```

from keras.models import Sequential

```



```

from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense from keras.preprocessing.image
import ImageDataGenerator

# Define hyperparameters
batch_size = 32 epochs = 10

input_shape = (125, 125, 3) num_classes = 5

# Create model model =
Sequential()

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=input_shape))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Flatten())

model.add(Dense(128, activation='relu')) model.add(Dense(num_classes, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

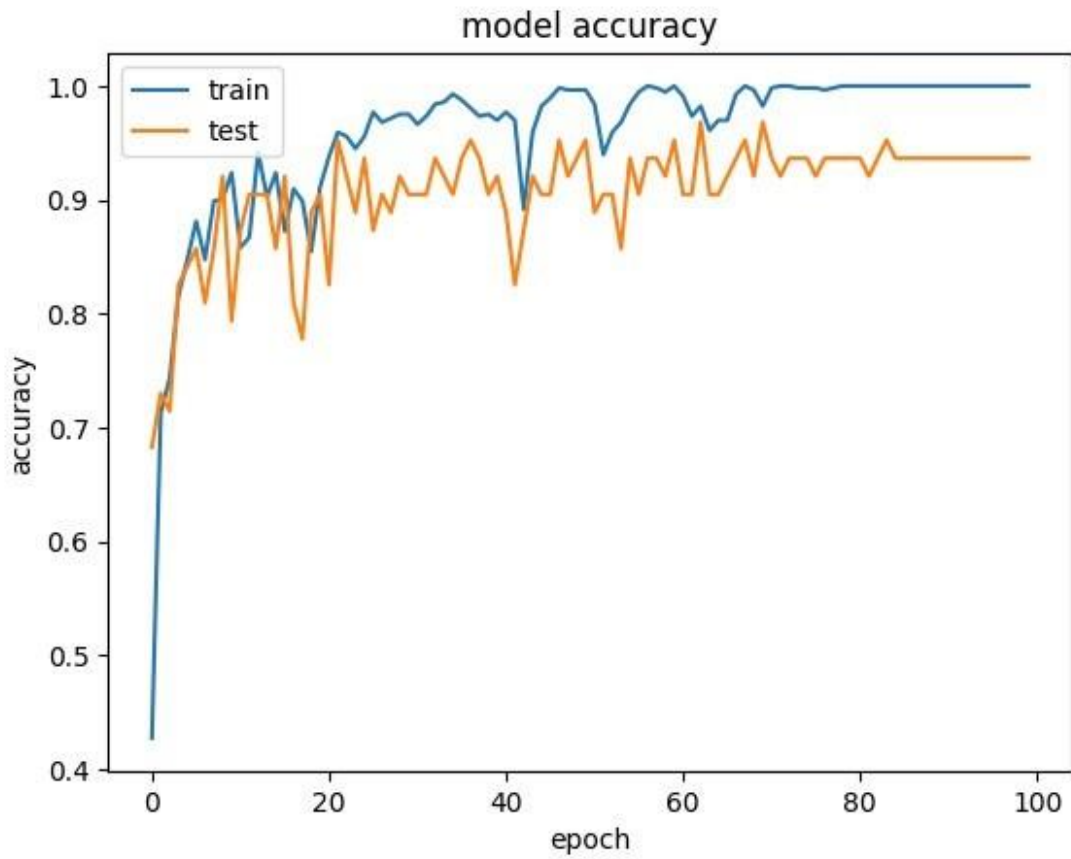
history = model.fit(X_train, y_train_OHC, batch_size=batch_size, epochs=100, validation_split=0.1)

# list all data in history print(history.history.keys()) # summarize
# history for accuracy plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy']) plt.title('model
accuracy') plt.ylabel('accuracy') plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left') plt.show()

dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])

```



```
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss']) plt.title('model loss')
plt.ylabel('loss') plt.xlabel('epoch') plt.legend(['train',
'test'], loc='upper left') plt.show()
```

