BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION COMPUTER SCIENCE IN ENGLISH

DIPLOMA THESIS

Ground-Based Cloud Image Classification Using Deep Learning And Data Augmentation

Supervisor
Asist. Dr. Horea-Bogdan Muresan

Author
Denisa László

2024

ABSTRACT

Clouds play a critical role in Earth's atmospheric dynamics and serve as key indicators of weather patterns and climate change. Analyzing cloud formations from ground-based images provides valuable insights into environmental conditions. However, accurately classifying cloud types from images remains a challenging task that requires the development of intelligent systems.

This study tries to investigate the impact of training popular machine learning architectures from scratch for the classification of ground-based cloud images across eleven distinct categories. Data augmentation was used to enhance model robustness and adaptability to diverse cloud formations.

Experiments were conducted using three different base models: ResNet50, MobileNetV3Small, and EfficientNetV2B0. Various combinations of hyperparameters like learning rates, batch sizes and loss functions, were systematically explored to optimize model performance. The results were compared to the latest advancements in the field, ensuring a comprehensive evaluation of the employed methodologies. The EfficientNetV2B0 model achieved the highest performance with an accuracy of 97.26% and a loss of 0.08 on the test dataset.

Additionally, a mobile application that integrated the top-performing configurations from the three architectures, was developed. This application allows users, regardless of their expertise, to upload or capture an image and view the predicted cloud classification. This tool aims to assist meteorologists, weather enthusiasts, and the general public in identifying cloud types and understanding atmospheric conditions.

Contents

1 Introduction **1**

1.1 Context and Motivation 1

1.2 Objectives 3

1.3 Thesis Structure 3

1.4 Declaration of Generative AI and AI-assisted technologies in the writing process 4

2 Theoretical Foundations **5**

2.1 What Are Clouds And How Can We Classify Them 5

2.1.1 High Clouds 6

2.1.2 Mid-height Clouds 7

2.1.3 Low Clouds 8

2.2 CNN Fundamentals 9

2.2.1 Convolutional Neural Networks 9

2.2.2 Training a CNN 11

2.3 Related Work 13

2.3.1 CloudNet 13

2.3.2 Leveraging Deep Learning 14

3 Methodologies **15**

3.1 CCSN Database 15

3.1.1 Contrails 15

3.1.2 Database Details 15

3.2 Data Augmentation 16

3.3 Base Model 18

3.3.1 ResNet-50 19

3.3.2 MobileNetV3Small 20

3.3.3 EfficientNetV2B0 21

4 Training and Results **23**

4.1 Experiments 24

4.2 Results 25
4.2.1 ResNet50 25
4.2.2 MobileNetV3Small 29
4.2.3 EfficientNetV2B0 31
4.2.4 State of the art comparison 36

5 Mobile Application **38**
5.1 Technologies 38
5.2 System Design 39

6 Conclusions **41**

Bibliography **43**

Chapter 1

Introduction

1.1 Context and Motivation

Cloud classification plays a crucial role in meteorology, aiding in weather prediction and understanding atmospheric dynamics. In meteorological studies, clouds are categorized into distinct types based on their appearance, altitude, and formation processes. These classifications provide valuable insights into weather patterns and atmospheric dynamics, enabling forecasters to predict weather events with greater accuracy. Moreover, cloud classification contributes to climate modeling efforts by improving our understanding of cloud-climate interactions and their impact on global climate patterns. As climate change continues to alter atmospheric conditions, accurate cloud classification becomes imperative for assessing its effects on weather variability and long-term climate trends.

Efforts to monitor and classify clouds are essential for enhancing our understanding of atmospheric processes and improving weather prediction capabilities. By studying cloud formations and their associated characteristics, meteorologists can better anticipate weather patterns, mitigate the impacts of extreme weather events, and inform decision-making for various sectors, including agriculture, transportation, and public health. [Nor00]

Clouds also play a crucial role in regulating UV radiation levels reaching the Earth's surface. They act as natural filters, absorbing and scattering solar radiation, as a result, influencing UV exposure levels. Understanding cloud dynamics is essential for assessing UV radiation risks and implementing appropriate protective measures for human health and environmental conservation. [Ama93] Cloud types significantly impact the reflection and absorption of solar radiation, influencing Earth's energy balance and affecting physical, dynamical, chemical, and biological processes within the climate system. By reflecting solar radiation, clouds directly influence the surface energy budget while also reducing long wave cooling, which

primarily affects energy lost from the atmosphere. Thus, understanding cloud types and their radiative properties is crucial for comprehending UV radiation dynamics and their implications for Earth's climate. [CPG05]

The primary objective of this thesis is to enhance cloud classification accuracy through the application of advanced machine learning techniques. This research aims to analyze the impact of applying data augmentation in addition to training popular architectures from scratch, to determine which methods significantly improve the performance of a cloud classifier. By exploring the effectiveness of architectures such as ResNet50, MobileNetV3Small, and EfficientNetV2B0, we seek to demonstrate their functionality in cloud classification tasks and contribute to the growing body of knowledge in this domain.

This thesis also delves into the role of data augmentation in improving model performance. By simulating real-world variations and imperfections in cloud images, data augmentation helps the models generalize better, ultimately leading to more accurate classifications. Another key aspect of this research is investigating the impact of training models from scratch with uninitialized weights. This approach allows us to examine how the lack of pre-trained weights affects the classifier's ability to generalize and perform well on unseen data.

By benchmarking the results against existing literature, we aim to establish the competitiveness of this approach and highlight its potential benefits. Improved cloud classification accuracy supports the development of more precise weather models, leading to better short-term and long-term weather forecasts. Furthermore, accurate cloud classification enhances climate modeling by providing deeper insights into cloud-climate interactions, aiding scientists in understanding the impacts of climate change on atmospheric dynamics.

In addition to its contributions to meteorology and climate science, this research has practical applications that can benefit public and environmental health. Understanding cloud dynamics and their influence on UV radiation can help in assessing UV radiation risks and informing protective measures for human health and environmental conservation. The development of a cloud classification app based on the findings of this research can assist meteorologists, weather enthusiasts, and the general public in identifying and understanding cloud types, providing a valuable tool for educational and practical purposes.

By achieving these objectives, this research aims to demonstrate the effectiveness of modern machine learning techniques in cloud classification and highlight their potential applications in enhancing weather prediction, climate modeling, and public awareness of atmospheric phenomena.

1.2 Objectives

This study investigates the intricate design and implementation of a cloud classification system, aiming to create an efficient method capable of accurately identifying diverse cloud types from ground-based images. The ultimate goal is to seamlessly integrate this system into a mobile application, empowering individuals to easily identify clouds through their own photographs. But this task is not without challenges, the most significant of which being the natural resemblance of cloud forms, which makes accurate classification extremely difficult.

To achieve this, the research evaluates the effectiveness of machine learning architectures such as ResNet50, MobileNetV3Small, and EfficientNetV2B0 in cloud classification. It explores the impact of data augmentation techniques, aiming to enhance model generalization and accuracy on unseen data by simulating real-world variations in cloud images. The study also investigates the effects of training models from scratch versus using pre-trained models, providing insights into the most effective training strategies.

1.3 Thesis Structure

The theory, design, and testing of various object classification models are all discussed in this thesis. There are six chapters in the thesis, which are as follows:

- The first chapter (current one) serves as an introduction to the problem statement and outlines the objectives of the study.
- The second chapter explores theoretical aspects, explaining how clouds are categorized based on their characteristics, their influence on weather patterns, fundamentals of Convolutional Neural Networks (CNNs) and a review of recent advancements in cloud classification research.
- Chapter three provides a comprehensive overview of the CCSN database used in the study, including a brief introduction to condensation trails. Additionally, it explores the data augmentation techniques employed and provides detailed insights into the architecture of the utilized base models.
- In the fourth chapter, the focus shifts to the personal experiments conducted to address the cloud classification challenge, presenting the results obtained from these experiments and comparing them with the current state-of-the-art approaches.

- Chapter five offers an overview of the development process of the mobile application. It discusses the technologies used in the app's development, along with insights into the system design of the final application.
- The final chapter presents a concise conclusion, summarizing the key findings of the study and potentially offering avenues for future research or development in the field.

1.4 Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used QuillBot, ZeroGPT and ChatGPT for rephrasing in order to improve the language and readability of the thesis. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the content of the thesis.

Chapter 2

Theoretical Foundations

2.1 What Are Clouds And How Can We Classify Them

Clouds are visible collections of small water droplets or ice crystals in the atmosphere of the Earth. Since most of the sunlight is reflected by the closely packed microscopic water droplets inside, they often appear white. Our eyes interpret all of the combined sunlight's wavelengths as white. The reason clouds get darker before it rains is because water vapor condenses into raindrops, creating bigger gaps between the droplets. Cloud behaviour and appearance is determined by a wide variety of physics and dynamics, including certain principles of radiation. [Ran03]

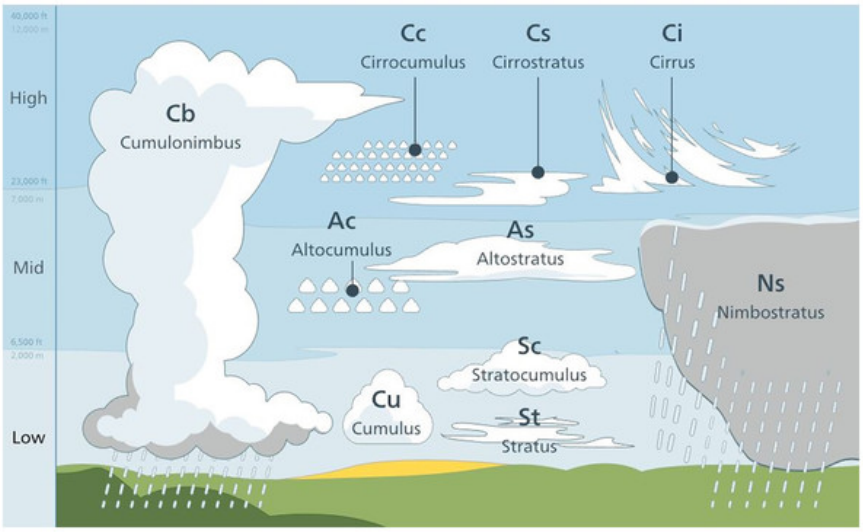

Figure 2.1: Cloud Types By Height
[ZLZS18]

They are grouped under ten categories (Altocumulus, Altostratus, Cumulonimbus, Cirrocumulus, Cirrus, Cirrostratus, Cumulus, Nimbostratus, Stratocumulus, Stratus), classified by two big factors: height and shape. Based on their height they

are grouped under 3 big categories: high clouds, mid-height clouds, low clouds, presented in Figure 2.1

2.1.1 High Clouds

High-level clouds have the prefix ”cirro-” and can be observed at over 7000 meters above ground level. The low temperatures at such heights sometimes give these clouds, which are primarily composed of ice crystals, a thin, streaky, white appearance. High clouds are classified into three primary types: Cirrus, Cirrostratus, and Cirrocumulus. [Ran03]

Figure 2.2: Cirrus [ZLZS18]

Figure 2.3: Cirrostratus [ZLZS18]

Figure 2.4: Cirrocumulus [ZLZS18]

High altitude clouds with a hair-like texture and short length are called Cirrus clouds. These fragile clouds resemble clumps of hair and are wispy with a silky polish. They are the whitest clouds in the sky throughout the day.

When dry air ascends, the small quantity of water vapor in the air condenses into ice, forming Cirrus clouds. A Cirrus cloud consists solely of ice crystals, which cause it to appear white and have a variety of sizes and shapes. They frequently occur prior to a warm front, which is the combination of the air masses at high elevations and an indication of approaching weather change.

Transparent high clouds known as Cirrostratus, envelop a significant portion of the sky. They occasionally create ”halo phenomena”, which are colored or white rings, patches, or arcs of light surrounding the Moon or the Sun. Sometimes, Cirrostratus clouds can be so thin to the point that the only sign of their presence in the sky is the halo..

If Cirrostratus appear following cirrus and spread from one location, across the sky, they can occasionally indicate the arrival of a warm front. This means that precipitation may follow in the next 12 to 24 hours, or within as little as 6–8 hours if the current is moving quickly.

Cirrocumulus clouds consist of several little white clouds called cloudlets, usually grouped together at high elevations. The small cloud formations, consisting mostly of ice crystals, are evenly arranged and often grouped together to look like waves in the sky.

Cirrocumulus clouds are usually associated with clear skies as their rain never reaches the ground. However, often times, they appear before bad weather.

2.1.2 Mid-height Clouds

Water droplets have the potential to crystallize into ice at extremely low temperatures and make up the majority of mid-level clouds. This type of clouds often lie between 2000 and 7000 meters above ground level. Mid-height clouds are classified into three primary types: Altostratus, Altocumulus, and Nimbostratus. [Fit17]

Figure 2.5: Altostratus [ZLZS18]

Figure 2.6: Altocumulus [ZLZS18]

Figure 2.7: Nimbostratus [ZLZS18]

Large, thin clouds at mid-level are called Altostratus. These clouds usually consist of both water droplets and ice crystals, and in certain areas they are thin enough to allow a slight view of the Sun through the clouds. Typically, they have few distinguishing characteristics and are spread out across a large region.

Altostratus frequently develop ahead of a warm or obstructed front. As the front passes through, bringing rain or snow, the Altostratus layer deepens and thickens until it becomes Nimbostratus. For this reason, witnessing it often indicates that a shift in weather is about to occur.

The term "cloudlets" refers to the tiny, spherical clusters that make up Altocumulus clouds, which are mid-level cloud patches or layers. On the other hand, Altocumulus come in a variety of forms because of their numerous types.

Altocumulus clouds, which are typically observed in settled weather, are primarily made up of droplets, though they can also contain ice crystals. Even though these clouds rarely result in rainfall, when they do, it remains in the atmosphere.

Nimbostratus clouds are dense enough to block the sun, they are grey, black, and uniform layers of clouds. These clouds, responsible for steady rainfall, are frequently associated with frontal systems caused by mid-latitude cyclones.

Most of the sky is often covered by these mid-level clouds, which are commonly associated with continuous light rain or snowfall. Nimbostratus often causes precipitation that can last for many hours until the associated front passes..

2.1.3 Low Clouds

Regardless of their characteristics, low-level clouds do not have a prefix in their names, which are based on ”strato-” or ”cumulo-”. Below 2000 meters (above ground level), low clouds are made mostly of ice crystals (and snow), except during cold winter storms when they are primarily liquid water droplets or super cooled droplets. Mid-height clouds are classified into four primary types: Stratus, Cumulus, Stratocumulus, and Cumulonimbus. [HJ14]

Figure 2.8: Stratus [ZLZS18]

Figure 2.9: Cumulus [ZLZS18]

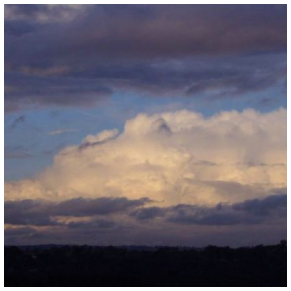

Figure 2.10: Stratocumulus [ZLZS18]

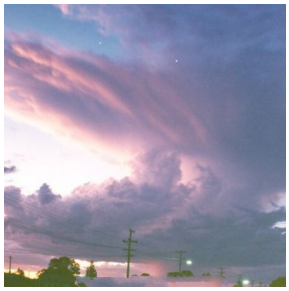

Figure 2.11: Cumulonimbus [ZLZS18]

Low-level layers that are predominantly gray or white in color are called Stratus clouds. They are the type of cloud that sits closest to the ground and are sometimes visible at ground level as mist or fog. In their ”nebulosus” state, they often experience long periods of gloomy and bleak days that last for extended periods of time.

Stratus typically results in little to no precipitation, yet if it is thick enough, it may cause a little drizzle. If it becomes cold enough, this drizzle may also turn into light snowfall.

Cumulus clouds are individual clouds that have a fluffy, cauliflower-like look and are commonly observed in nice weather. When the Sun shines on them, the tops of these clouds appear as beautiful white fluffy tufts, while their bottoms are usually somewhat dark.

Cumulus clouds usually indicate good weather and are seen on clear, sunny days. Yet, under specific conditions, Cumulus clouds have the potential to mature

into Cumulonimbus or towering Cumulus Congestus clouds, resulting in precipitation.

Cloud formations called Stratocumulus clouds are characterized by clusters or patches of cloud that sit low in the sky and range in color from bright white to dark grey. These are the most common clouds found on Earth, known for their easily recognizable bases, with some being darker than others. They have the ability to be connected, but usually there are gaps separating them.

Even though Stratocumulus clouds can form in various weather conditions, ranging from calm and dry to more rainy situations, they are often not the root of the issue. Although receiving anything heavier than a gentle drizzle from Stratocumulus clouds is uncommon, they are occasionally mistaken for rain clouds.

Cumulonimbus clouds are ominous clouds that have multiple layers and that reach far into the sky in the form of columns or towers. Cumulonimbus clouds, also referred to as thunderclouds, are the sole clouds that can produce lightning, thunder, and hail. The cloud may only be a few hundred feet above the Earth's surface, and its base is frequently flat with a very dark feature hanging underneath it that resembles a wall.

Severe weather phenomena such as lightning strikes, tornadoes, heavy rainfall and hailstorms are associated with Cumulonimbus clouds.

2.2 CNN Fundamentals

2.2.1 Convolutional Neural Networks

CNNs, short for Convolutional Neural Networks, are the most well-known and frequently used algorithms. CNN's main strength lies in its capacity to automatically and individually recognize important characteristics. CNNs are commonly used in various fields including face recognition, speech processing and computer vision. Just like in a regular neural network, CNNs were inspired by neurons in the brains of animals, specifically cats. [AZH+21] Three major advantages of the CNN were identified in this book [GBC16]: sparse interactions, sharing parameters and equivalent representations. Different from traditional fully connected networks, CNNs use local connections and shared weights, because there are incredibly few parameters used in this function, the network operates faster and the training process is made simpler.

The focal point of CNN architecture is the convolutional layer. It is made up of a range of convolutional filters, which are also called kernels. The feature map is created by applying these filters to the input image, which is represented as N-dimensional metrics.

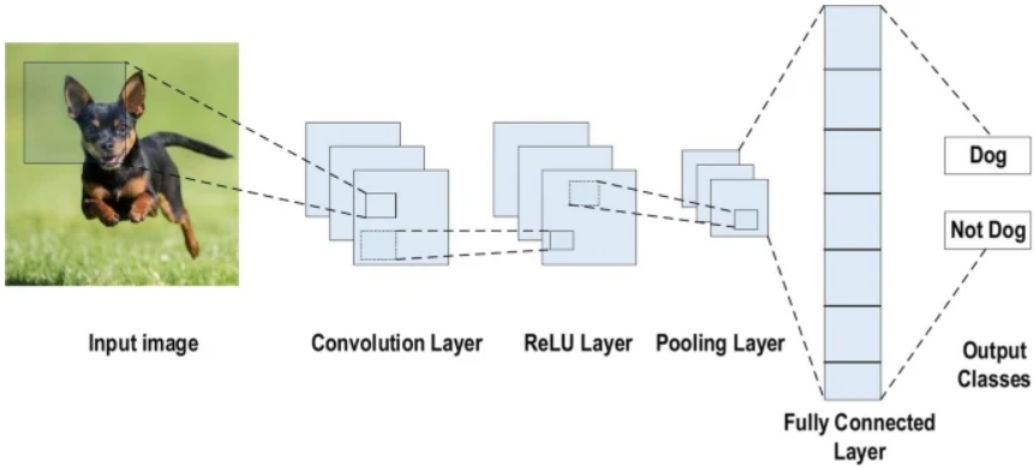

Figure 2.12: Example of a CNN Architecture
[AZH+21]

The pooling layers' main goal is to down-sample the feature maps produced by convolutional operations. Put differently, this technique shrinks large feature maps to create smaller ones. At the same time, it retains the majority of the key data during each step of the pooling process. There are many types of pooling methods that can be used in various pooling layers such as: gated, tree, min, max, average, global average and global max pooling. The min, max and global average pooling methods are widely recognized and commonly applied.

The fully connected layer is commonly located at the top of a CNN architecture. In this layer, the Fully Connected (FC) technique is utilized, connecting each neuron to every other neuron in the layer above. Functioning as a feed-forward artificial neural network, it performs based on the identical principles of a conventional multi-layer perceptron neural network. The last pooling or convolutional layer gives input to the FC layer. After being flattened, the feature maps are transformed into a vector that serves as the input for this layer. The output of the fully-connected layer represents the final CNN output.

A popular technique for generalizing is the dropout method, in which neurons are randomly removed during each training epoch. By doing this, the model is compelled to understand multiple distinct characteristics, resulting in an equal distribution of feature selection capabilities across all neurons in the group. The neuron that has been dropped will not be involved in either the forward or backward propagation during the training process. On the other hand, when testing, the full-scale network is used for making predictions.

2.2.2 Training a CNN

Training a Convolutional Neural Network (CNN) is a process in which the network learns to recognize patterns and features within input data. It involves iteratively adjusting the network's parameters (such as weights and biases) based on the differences between its predictions and the actual labels of the training data. Through techniques like back-propagation and optimization algorithms such as stochastic gradient descent [Ama93], the CNN gradually improves its ability to extract relevant features from the input images and make accurate predictions. Essentially, a trained CNN acts as a sophisticated image classifier, capable of identifying objects, shapes, and textures within images with remarkable accuracy.

Optimizers are algorithms used to update the parameters of a neural network during the training process, with the aim of minimizing the loss function and improving the model's performance. They determine the direction and magnitude of parameter updates based on gradients computed from the loss function and are critical for efficient training.

Among the various optimizers, Adam [KB14] stands out as a popular choice due to its adaptive learning rate properties. Adam dynamically adjusts the learning rates for each parameter based on the magnitudes of past gradients and the current state of the optimization, allowing for faster convergence and better performance, especially in tasks with high-dimensional parameter spaces or noisy gradients. By incorporating momentum and adaptive learning rates, Adam strikes a balance between exploration and exploitation, enabling neural networks to efficiently navigate complex optimization landscapes and achieve superior performance.

Loss functions are essential components in training machine learning models, as they quantify the disparity between the predicted outputs and the actual targets during the training process. These functions serve as optimization objectives, guiding the model towards making more accurate predictions.

Categorical cross-entropy [MMZ23] is a widely used loss function, particularly in classification tasks with multiple classes. It calculates the difference between the predicted class probabilities and the true class labels, penalizing deviations from the actual distribution of classes. By minimizing the categorical cross-entropy loss, the model learns to assign higher probabilities to the correct classes and lower probabilities to incorrect ones, ultimately improving its classification performance. In essence, the choice of a suitable loss function, such as categorical cross-entropy, plays a crucial role in shaping the learning behavior of the model and optimizing its predictive capabilities.

Cosine similarity is a valuable loss function, particularly in tasks involving similarity measurement and clustering. It calculates the cosine of the angle between

two non-zero vectors, which in this context represent the predicted and actual class embedding. The cosine similarity score ranges from -1 to 1, where 1 indicates identical orientation and higher similarity, and -1 indicates completely opposite orientation. By maximizing cosine similarity, the model learns to produce embedding that closely align with the actual class embedding, as a result, enhancing its ability to correctly identify similar classes and distinguish dissimilar ones. In essence, the choice of a suitable loss function, such as cosine similarity, plays a crucial role in shaping the learning behavior of the model and optimizing its capability to recognize and cluster similar data points accurately.

Learning rates are a crucial hyperparameter in the training of machine learning models, including neural networks, as they determine the step size at which the model parameters are updated during optimization. The selection of the learning rate has a substantial impact on both the convergence patterns and the eventual effectiveness of the model. An excessively high learning rate can result in overshooting, which can cause the optimization process to diverge or oscillate near the optimal point. On the other hand, a too low learning rate may result in slow convergence or getting stuck in local minima. Finding the right balance is essential, as an appropriate learning rate allows the model to efficiently navigate the optimization landscape, making steady progress towards minimizing the loss function. [AZH+21]

In deep learning tasks, evaluation measures are essential for optimizing classifiers. They play crucial roles in both the training and testing stages of the traditional data categorization pipeline. These metrics serve as discriminators to choose the best-optimized solution by influencing the classification algorithm's optimization during training. This guarantees that the classifier provides highly accurate estimates for further evaluations. Once the model is trained, evaluation metrics shift to assessing its efficiency during testing, serving as evaluators by measuring performance on unseen data. [AZH+21]

To assess the classifier performance, key metrics are specified, including true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These metrics provide the basis for assessing the efficacy of the classifier by distinguishing between correctly and incorrectly categorized cases. Some of the most used and well-known metrics are Accuracy 2.1, Precision 2.2, Recall 2.3 and F1-Score 2.4.

Accuracy determines the percentage of correctly predicted classes to the total number of instances evaluated.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.1}$$

Precision is used in order to calculate the positive patterns that are correctly pre-

dicted by all predicted patterns in a positive class.

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

Recall is used to calculate the percentage of positive patterns that are correctly classified.

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

F1-Score computes the harmonic average between recall and precision rates

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{2.4}$$

2.3 Related Work

2.3.1 CloudNet

In 2018, CloudNet and the CCSN database, presented in more detail in Section 3.1, are introduced in this article [ZLZS18]. CloudNet is an optimized convolutional neural network created with the purpose of ground-based cloud classification.

CloudNet originated from an improvement of AlexNet [KSH12]. It consists of two fully connected layers and five convolutional layers, resulting in a clear-cut and uncomplicated architecture. The network takes RGB images with a resolution of 227 X 227 as input and gives a collection of label predictions as output. In addition, an optimization of this network is the subtraction of the mean RGB value from each pixel during training, this optimization helped the accuracy and speed of the training part.

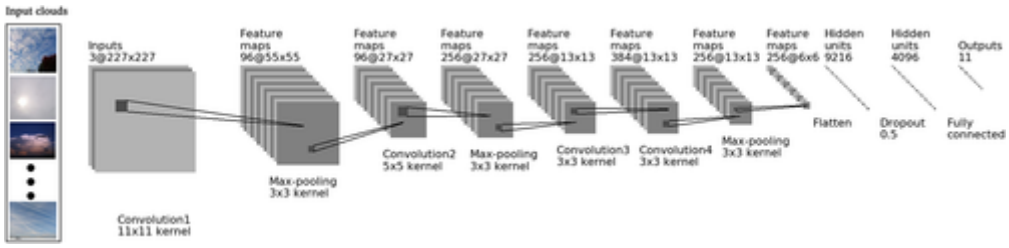

Figure 2.13: CloudNet Architecture
[ZLZS18]

In addition to the introduction of CloudNet, the Cirrus Cumulus Stratus Nimbus database was created in order to help improve CloudNet's accuracy. The previous used database for this problem was SWIMCAT [DLW15] which consisted of only 784 images separated into only 5 categories (clear sky, patterned clouds, thick dark clouds, thick white clouds, veil clouds) which was insufficient for a thorough classi-

fication. CCSN introduced more categories of clouds, with more overall images, for a more complex classification.

In the article, the model was evaluated using precision, accuracy and recall. An overall score was calculated using these measures for each of the eleven categories of clouds in CCSN, these scores vary from 87% to 91%. These findings are better that other state-of-the art approaches from that time like [XCZ+16] and [YCX17], whose results vary from 81% to 87%. So CloudNet gets an average accuracy score of 88% by implementing a rather simple convolutional neural network with a more complex database.

2.3.2 Leveraging Deep Learning

The most recent advancement for this problem was published in January of 2024, in this article [GKB+24]. Deep learning and image processing where leveraged in order to obtain better results than previously implemented solutions.

The main difference in this approach is the utilization of fine-tuning popular deep learning pre-trained models. The base models used in this paper are MobileNetV2, InceptionV3, EfficentNetV2L, VGG-16, Xception, ConvNeXtSmall and ResNet-152V2. Freeze-out fine-tuning was implemented here by freezing out a third of the model while training.

During training, the optimizer chosen for this implementation was Adam and the loss was of type categorical cross entropy with a learning rate of 0.0001 for 100 epochs on the CCSN dataset. The model is structured this way: data augmentation layer, preprocessing layer, pre-trained base model, pooling layer, dropout layer, prediction layer.

One of the models from this paper, the Xception one, showed the best accuracy when the findings were compared to the most recent technological approaches in the literature. They also evaluated the implementations of each base model with one another, results presented in Table 2.1

Models	Accuracy (%)
MobileNetV2	96.36
VGG-16	86.33
ResNet-152V2	95.70
Inception V3	95.31
EfficientNetV2L	96.48
Xception	97.66
ConvNeXtSmall	97.27

Table 2.1: 2024 Fine-tuning Results [GKB+24]

Chapter 3

Methodologies

3.1 CCSN Database

The database chosen for this project is Cirrus Cumulus Stratus Nimbus data set (CCSN), introduced in [ZLZS18]. The data set consists of 2,543 distinct ground-based cloud photos that have been classified through multiple subjective classification rounds based on visual attributes and the expertise of meteorological professionals. The cloud photos in this database are all in JPEG format, with a resolution of 256 × 256 and significant intraclass variance. The dataset is divided into the ten categories presented in Section 2.1, plus an additional category: Contrails.

3.1.1 Contrails

A common sight in the sky is the formation of Contrails, which are made of water vapor released by an aircraft's engines. There isn't enough of a Contrail to produce weather on the ground. When there is high pressure and little to no other cloud in the area, we typically see them in clear skies. They have an overall warming effect on global climate, because they can spread to form heat-trapping Cirrus clouds.

3.1.2 Database Details

In Figure 3.1, we can see a detailed breakdown of the number of images distributed across the 11 cloud categories. Additionally, each category includes a concise description of the corresponding cloud type, as previously discussed. This table provides an overview of the dataset composition and highlights the key characteristics of each cloud type, which are essential for understanding the classification challenges and the diversity of the data used in training and testing the models. 3.1

All images used as examples of clouds in Section 2.1 are part of the CCSN Database. Figure 3.2 shows a sample from the dataset.

Category	Number of images	Type	Description
Ci	139	Cirrus	Fibrous, white feathery clouds of ice crystals
Cs	287	Cirrostratus	Milky, translucent cloud veil of ice crystals
Cc	268	Cirrocumulus	Fleecy cloud, cloud banks of small, white flakes
Ac	221	Altocumulus	Grey cloud bundles, compound like rough fleecy cloud
As	188	Altostratus	Dense, gray layer cloud, often even and opaque
Cu	182	Cumulus	Heap clouds with flat bases in the middle or lower level
Cb	242	Cumulonimbus	Middle or lower cloud level thundercloud
Ns	274	Nimbostratus	Rain cloud; grey, dark layer cloud, indistinct outlines
Sc	340	Stratocumulus	Rollers or banks of compound dark gray layer cloud
St	202	Stratus	Low layer cloud, causes fog or fine precipitation
Ct	200	Contrails	Line-shaped clouds produced by aircraft engine exhausts

Figure 3.1: Database partition [ZLZS18]

3.2 Data Augmentation

In this thesis, the incorporation of a robust data augmentation layer was a crucial step to enhance the model's ability to generalize and perform well on diverse and unseen cloud images. The data augmentation layer employed a series of transformations aimed at simulating real-world variations and imperfections that might be present in the dataset. Specifically, the augmentation process included random horizontal flipping, random rotations within the range of -0.2 to 0.3 radians, and random contrast adjustments between 0.2 and 0.5. Additionally, a custom random brightness adjustment was applied, followed by the introduction of Gaussian noise with a standard deviation of 0.1.

Image preprocessing, particularly through data augmentation, played a pivotal role in improving the model's robustness and accuracy. By artificially expanding the dataset with different versions of the original images, the models were exposed to a broader spectrum of cloud appearances and conditions. This exposure helps the models to better learn the underlying features that distinguish different cloud types, rather than overfitting to specific patterns seen in the training data. Consequently, the models became more adept at handling variations in cloud formations, lighting

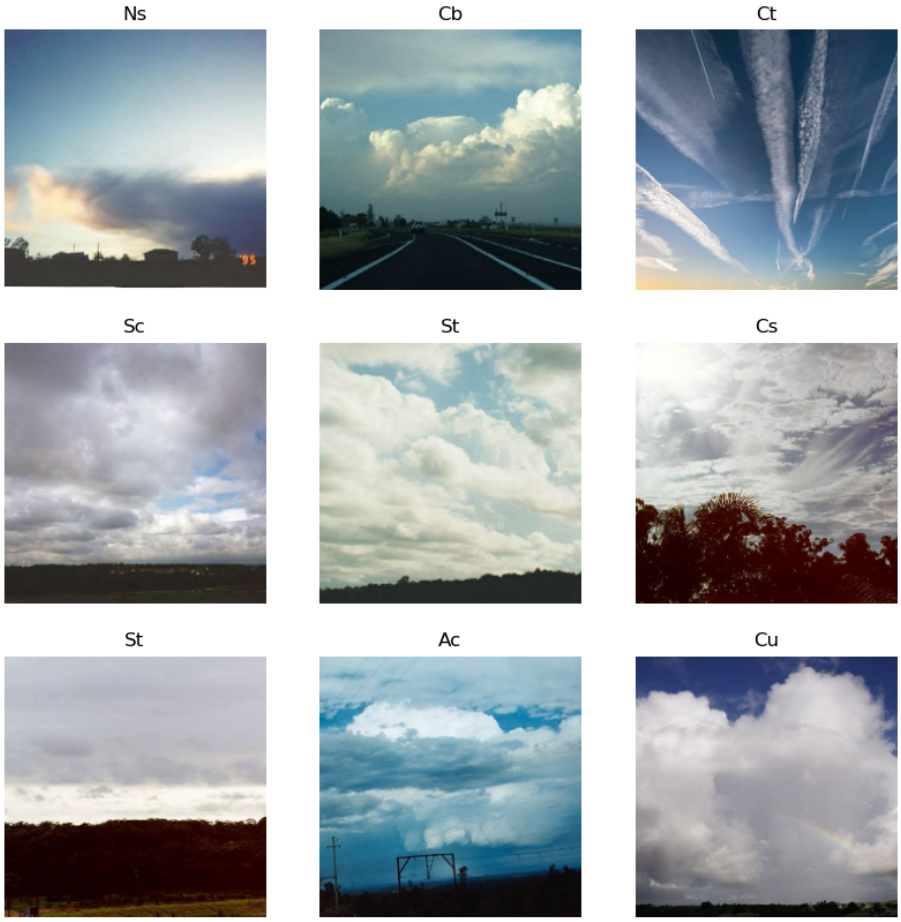

Figure 3.2: Samples from the dataset

conditions, and atmospheric interference that they might encounter in real-world scenarios.

Figure 3.3 shows a sample of data that used augmentation, exemplifying the transformation applied to the original cloud image. This sample visually demonstrates the effectiveness of the augmentation techniques in introducing realistic variations and enhancing the dataset's diversity.

The strategic use of data augmentation significantly contributed to the overall performance of the models. It enhanced the models' ability to accurately classify the eleven cloud categories despite the challenges of cloud image variability. This preprocessing step ensured that the models maintained high accuracy and generalization capabilities, as evidenced by the testing phase results. Therefore, image preprocessing, through effective data augmentation, was a fundamental element in

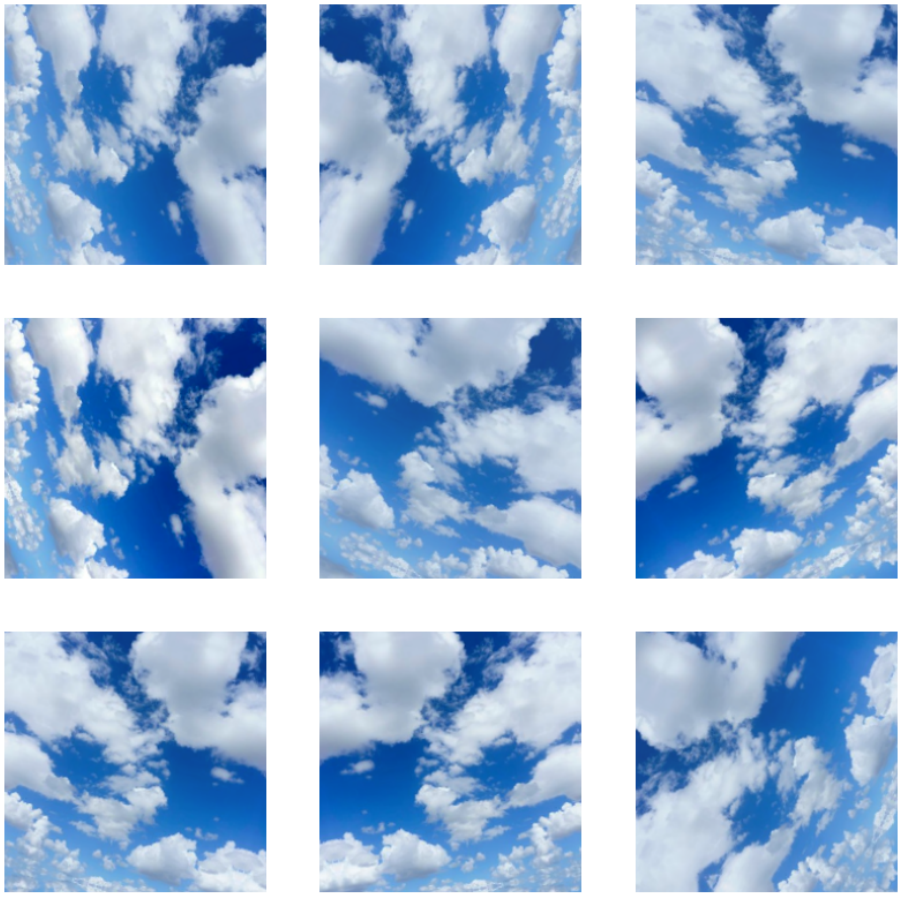

Figure 3.3: Samples of data augmentation

achieving the objectives of this research and developing a reliable and efficient cloud classification system.

3.3 Base Model

The architecture of the cloud classification model is designed to leverage advanced data augmentation techniques and a robust convolutional neural network to achieve high accuracy. The model begins with a data augmentation layer, explained in detail in the previous section. These augmentations simulate real-world variations and imperfections, helping the model generalize better to unseen data.

The base of the model employs one of following state-of-the-art architectures: MobileNetV3Small, ResNet50 or EfficientNetV2B0, each chosen for their unique

strengths in handling complex image classification tasks. The base model serves as the feature extractor, with its weights initialized from scratch to allow for customized learning specific to the cloud images. The base model is followed by a global average pooling layer, which reduces the spatial dimensions of the feature maps and prepares them for the final classification stage. A dropout layer with a rate of 0.2 is included to prevent overfitting by randomly dropping units during training.

The final classification layer consists of a dense layer with a softmax activation function, which outputs the probabilities for each of the cloud categories. The model is built using the Keras Functional API, starting with an input layer that processes the images through the data augmentation pipeline. The augmented images are then passed through the base model, followed by global average pooling, dropout, and the dense classification layer to produce the final predictions.

In the following sections, I will further explore the specifics of the three base models used, highlighting their individual architectures and contributions to the overall performance of the cloud classification system.

3.3.1 ResNet-50

ResNet, short for Residual Network, introduced a groundbreaking architectural innovation to address the challenges of training very deep neural networks. In this paper [HZRS16], a novel approach is proposed for building deeper networks by introducing residual connections.

ResNet50 is one of the models based on the ResNet design, made up of precisely 50 layers. It consists of a sequence of convolutional layers grouped into various blocks, where each block includes multiple convolutional layers with shortcut connections. The shortcut connections, which are also called skip connections or identity mappings, enable the network to skip over certain layers and directly pass information from earlier layers to later ones.

The core idea behind ResNet is the concept of residual learning, where instead of directly trying to learn the underlying mapping between inputs and outputs, the network learns residual functions. These residual functions capture the difference between the input and output features of a given layer, making it easier for the network to learn the desired mapping. Mathematically, this is expressed as F(x)+x, where F(x) represents the output of the layer and x is the input to the layer.

ResNet50 specifically consists of several building blocks, each containing multiple convolutional layers and shortcut connections. The basic building block is the residual block, which can be further divided into two types: the identity block and the convolutional block. The identity block preserves the input dimensions, while

the convolutional block adjusts the dimensions to match the output shape.

At its core, ResNet50 achieves impressive performance by enabling the training of very deep neural networks with hundreds of layers while mitigating the vanishing gradient problem. The introduction of residual connections allows for the direct flow of information, facilitating the training of deeper networks and ultimately leading to improved accuracy on various image recognition tasks.

ResNet50 was selected as one of the base models for the cloud classification model based on various factors, including insights derived from this paper [MA21]. That paper provided valuable comparative analyses of different convolutional neural network architectures, highlighting the superior performance of ResNet50 in image classification tasks compared to VGG16 and VGG19. The research findings demonstrated that ResNet50 exhibited improved accuracy and efficiency, particularly in handling complex datasets with a large number of classes, making it a compelling choice for the cloud classification task.

Furthermore, ResNet50's architectural innovations, such as residual connections, addressed the challenges of training very deep neural networks by mitigating issues like vanishing gradients and enabling smoother gradient flow during backpropagation. This architectural design allowed for the construction of deeper networks while maintaining manageable computational complexity, thus enhancing the model's capacity to capture intricate features and patterns inherent in cloud images.

3.3.2 MobileNetV3Small

MobileNetV3Small is a variant of the MobileNet architecture designed to provide efficient convolutional neural networks for mobile vision applications. Introduced in [HZC+17], MobileNetV3Small aims to strike a balance between model efficiency and accuracy, making it well-suited for resource-constrained environments such as mobile devices.

MobileNetV3Small achieves its efficiency through a combination of depthwise separable convolutions, linear bottlenecks, and efficient inverted residuals with linear shortcuts. These architectural elements greatly decrease the computational expenses and model size, while still achieving similar levels of accuracy as bigger and more computationally demanding models.

Depthwise separable convolutions, a key feature of the MobileNet architecture, split the standard convolutional operation into separate depthwise and pointwise convolutional layers. This separation allows the model to independently process spatial and channel-wise information, reducing the number of parameters and computations required for each convolutional operation.

Linear bottlenecks further enhance efficiency by using lightweight bottleneck structures with fewer channels, followed by expansion and projection layers. This design minimizes the computational burden without sacrificing the model's representational capacity, enabling MobileNetV3Small to achieve a favorable trade-off between efficiency and accuracy.

In addition, efficient inverted residuals with linear shortcuts optimize feature reuse and information flow within the network. These inverted residual blocks consist of a lightweight bottleneck layer followed by a linear projection layer, allowing for efficient utilization of computational resources while maintaining feature expressiveness.

MobileNetV3Small was chosen as a base model for the cloud classification model due to its exceptional efficiency and competitive performance on mobile vision tasks. Its lightweight architecture makes it well-suited for deployment on devices constrained by issues regarding resources, ensuring fast inference times and minimal computational overhead. Moreover, MobileNetV3Small's efficient design aligns with the objectives of the cloud classification model, enabling accurate and real-time classification of cloud images on mobile devices while conserving computational resources.

3.3.3 EfficientNetV2B0

EfficientNetV2B0 is a variant of the EfficientNet architecture introduced in [TL19]. Its purpose is to create a convolutional neural network structure that is both effective and scalable, delivering high performance in various tasks without using excessive computational resources.

EfficientNetV2B0 builds upon the principles of model scaling, compound scaling, and neural architecture search to achieve superior efficiency and accuracy. The architecture consists of multiple layers of efficient building blocks, including depthwise separable convolutions, inverted residuals, and squeeze-and-excitation blocks.

Depthwise separable convolutions, a key component of EfficientNetV2B0, decompose the standard convolutional operation into separate depthwise and pointwise convolutions. This separation reduces the number of parameters and computations required for each convolutional layer, resulting in significant efficiency gains without compromising model performance.

Inverted residuals further enhance efficiency by employing lightweight bottleneck structures with fewer parameters and computations. These inverted residual blocks allow for efficient feature extraction and information flow within the network while minimizing computational overhead.

Squeeze-and-excitation blocks, another important feature of EfficientNetV2B0,

adaptively recalibrate channel-wise feature responses to enhance feature representational power. By selectively emphasizing informative channels and suppressing irrelevant ones, these blocks improve the discriminative capability of the network and enhance overall performance.

EfficientNetV2B0 was chosen as a base model for the cloud classification model due to its remarkable efficiency and scalability. Its streamlined architecture and impressive performance render it perfectly suited for deployment on devices with limited computational resources, guaranteeing swift inference times and minimal computational burden. Furthermore, EfficientNetV2B0's capability to attain cutting-edge accuracy across diverse tasks perfectly aligns with the goals of the cloud classification model, facilitating precise and efficient classification of cloud images while conserving computational resources.

Chapter 4

Training and Results

The training process for the cloud classification model involved experimenting with various configurations of base models, learning rates, loss functions, and batch sizes to optimize performance. The dataset, containing images of various cloud types, was prepared for training by splitting it into training, validation, and test sets. This was accomplished using a custom partition function. This function ensured that the dataset was shuffled and then split according to specified proportions: 70% for training, 20% for validation, and 10% for testing. The shuffling process, with a fixed seed, guaranteed that the split distribution remained consistent across different runs, aiding in reliable performance comparisons.

Image preprocessing played a crucial role in enhancing the robustness of the model by employing a data augmentation pipeline, which included random flips, rotations, contrast adjustments, brightness variations, and the addition of Gaussian noise, explained in more detail in Section 3.2. This augmentation strategy aimed to simulate various real-world conditions and improve the model's ability to generalize to unseen data.

The cloud classification model was developed using three different base architectures to compare their performance: ResNet50, MobileNetV3Small, and EfficientNetV2B0. These models were chosen for their proven track records in image classification tasks and their different strengths in terms of efficiency, accuracy, and computational requirements. Each base model was initialized without pre-trained weights to test the thesis' hypothesis regarding "learning-from-scratch".

The novelty of this research lies in the combination of extensive data augmentation with the training of popular architectures from scratch, a method not commonly explored in previous works. Most existing cloud classification models rely heavily on transfer learning with pre-trained weights due to the relatively small size of available cloud image datasets. However, inspired by this article [BD20], which demonstrated that the cosine similarity loss function could significantly improve performance on small datasets without pre-training, this study adopted a similar

approach. By applying cosine similarity as a loss function, the aim was to explore its potential benefits in training deep learning models from scratch on a cloud image dataset, providing a fresh perspective and potentially new insights into effective cloud classification techniques.

This approach diverges from traditional methods that primarily focus on transfer learning with pre-trained models. By emphasizing the potential of learning from scratch combined with robust data augmentation, this study contributes to the field by offering an alternative methodology that could lead to improved classification performance on small, specialized datasets.

4.1 Experiments

Each model was compiled with the Adam optimizer and trained over multiple epochs, with performance metrics such as accuracy, precision, recall, F1 score, and AUC (Area Under ROC Curve) being monitored. The custom metrics function included precision and recall for each class, a multi-label confusion matrix, and a ROC (Receiver Operating Characteristic Curve) AUC score, allowing for comprehensive evaluation across all cloud types. The training history for each model variant was recorded, and the results were analyzed to identify the best-performing configuration.

The training procedure involved a systematic exploration of hyperparameters, more specifically learning rates, loss functions, and batch sizes. The learning rates tested included 0.00001, 0.0001, and 0.001, while batch sizes of 32 and 64 were evaluated. Two loss functions were considered: cosine similarity and categorical cross-entropy. The combination of these parameters was crucial in determining the optimal setup for each base model.

While categorical cross-entropy is typically the preferred loss function for classification tasks, the decision to also use cosine similarity was inspired by findings from this article [BD20], as previously mentioned. This study, presented in 2020, demonstrated that cosine loss significantly outperforms cross-entropy loss on small datasets. The authors found that, without pre-training, the cosine loss function achieved a 30% higher accuracy on the CUB-200-2011 dataset compared to cross-entropy. These findings were consistent across other popular datasets as well.

As previously mentioned, different configurations of learning rates, loss functions and batch sizes where tested, in the following way:

- Cosine Similarity + Learning Rate 0.001 + Batch Size 64
- Cosine Similarity + Learning Rate 0.0001 + Batch Size 64

- Cosine Similarity + Learning Rate 0.0001 + Batch Size 32
- Categorical Cross-Entropy + Learning Rate 0.00001 + Batch Size 32
- Categorical Cross-Entropy + Learning Rate 0.0001 + Batch Size 32
- Categorical Cross-Entropy + Learning Rate 0.0001 + Batch Size 64

In conclusion, the experimental phase was meticulously designed to explore various combinations of hyperparameters and model architectures. By systematically varying the learning rates, loss functions, and batch sizes, the study aimed to uncover the optimal training configuration for each base model. The inclusion of cosine similarity as a loss function, based on recent research, provided an innovative approach to handling small datasets without pre-training. This comprehensive evaluation set the stage for the subsequent analysis of the results, where the performance of each model configuration will be thoroughly assessed.

4.2 Results

This section presents the outcomes of the experiments, highlighting the performance of each model configuration in terms of accuracy, loss, F1 score, precision, and recall. The results will be discussed for each base model—ResNet50, EfficientNetV2B0, and MobileNetV3Small—emphasizing the configurations that yielded the best performance. Visual aids, such as training and validation accuracy/loss plots, will be included to illustrate the training dynamics.

4.2.1 ResNet50

The best-performing configuration for the ResNet50 base model was achieved using the categorical cross-entropy loss function with a learning rate of 0.0001 and a batch size of 32. This setup resulted in an accuracy of 95.70%, a precision of 95.71%, a recall of 95.42%, and an F1 score of 95.56%. This configuration outperformed others in terms of both accuracy and overall metric scores, making it the optimal choice for cloud classification tasks with the ResNet50 model.

Another effective configuration used the categorical cross-entropy loss function with a learning rate of 0.0001 and a batch size of 64, achieving an accuracy of 92.96%, a precision of 94.31%, a recall of 91.83%, and an F1 score of 93.06%.

The configuration with the categorical cross-entropy loss function, a learning rate of 0.00001, and a batch size of 32 also performed reasonably well, achieving an accuracy of 91.01%, a precision of 91.35%, a recall of 89.41%, and an F1 score of

90.37%. These results indicate the effectiveness of the categorical cross-entropy loss function for this model.

In contrast, using the cosine similarity loss function with a learning rate of 0.001 and a batch size of 64 resulted in an accuracy of 84.37%, a precision of 88.24%, a recall of 83.62%, and an F1 score of 85.86%. Although lower than the categorical cross-entropy configurations, the cosine similarity results still demonstrate competitive performance, showcasing its potential utility in certain scenarios.

The configuration with the cosine similarity loss function, a learning rate of 0.0001, and a batch size of 32 achieved an accuracy of 87.50%, a precision of 87.06%, a recall of 86.58%, and an F1 score of 86.82%.

Another configuration using the cosine similarity loss function with a learning rate of 0.0001 and a batch size of 64 achieved an accuracy of 87.10%, a precision of 86.88%, a recall of 86.50%, and an F1 score of 86.69%. This result was slightly better than the one with a learning rate of 0.001, indicating the importance of fine-tuning learning rates for optimizing performance.

Based on these results, the final model implemented in the application was the one trained with the categorical cross entropy loss function, a learning rate of 0.0001, and a batch size of 32.

The experimental results for the ResNet50 base model are summarized in this table 4.2.1

Loss Function	Learning Rate	Batch Size	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Categorical Cross Entropy	0.0001	32	95.70	95.71	95.42	95.56
Categorical Cross Entropy	0.0001	64	92.96	94.31	91.83	93.06
Categorical Cross Entropy	0.00001	32	91.01	91.35	89.41	90.37
Cosine Similarity	0.001	64	84.37	88.24	83.62	85.86
Cosine Similarity	0.0001	32	87.50	87.06	86.58	86.82
Cosine Similarity	0.0001	64	87.10	86.88	86.50	86.69

Table 4.1: Results of base model ResNet50

Figure 4.7 illustrates the learning curves encompassing both the training and validation phases. The accuracy and loss trajectories exhibit significant instability, with values fluctuating widely across epochs. However, precision, recall and F1 score metrics demonstrate more stable and consistent trends.

In a similar way, Figure 4.8 presents the multi-class confusion matrix, offering nuanced insights into the model's classification performance across various cloud types. The high occurrence of high values along the diagonal underscores the proficiency of the model in accurately classifying the majority of samples across all cloud types, demonstrating impressive precision and recall rates.

Ultimately, a curated sample of examples from the test dataset is featured in Figure 4.9, demonstrating both accurate and inaccurate classifications. This qualitative examination offers a concrete look at how the model actually performs in

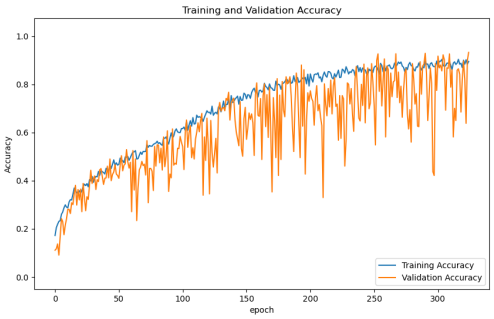

Figure 4.1: Accuracy

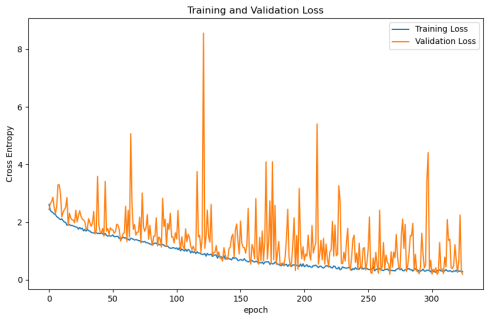

Figure 4.2: Loss

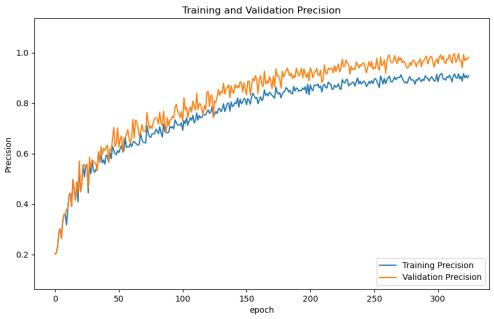

Figure 4.3: Precision

Figure 4.4: F1 Score

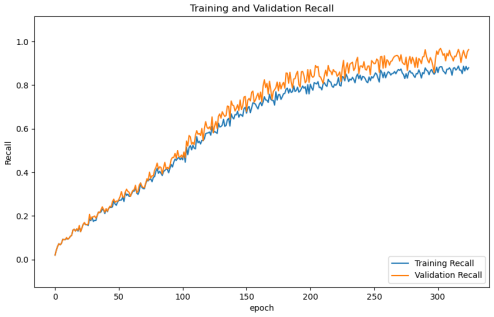

Figure 4.5: Recall

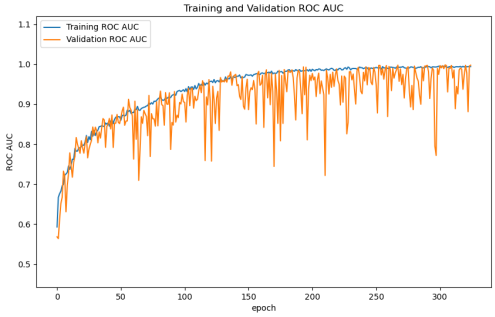

Figure 4.6: ROC AUC

Figure 4.7: Training And Validation Results For ResNet50

real-world scenarios. The precisely categorized photos highlight the model's ability to accurately distinguish between different types of clouds.

These visuals, combined with the numerical data, provide a comprehensive view of the ResNet50 model's performance in cloud classification tasks. Despite the instability observed in the accuracy and loss graphs, which showed significant fluctuations throughout training, the model's overall performance remains strong. This is evidenced by high precision, recall, and F1 scores, as well as convincing visual results. The selected configuration's ability to accurately classify a variety of cloud types underscores its effectiveness, even in the face of these training irregularities

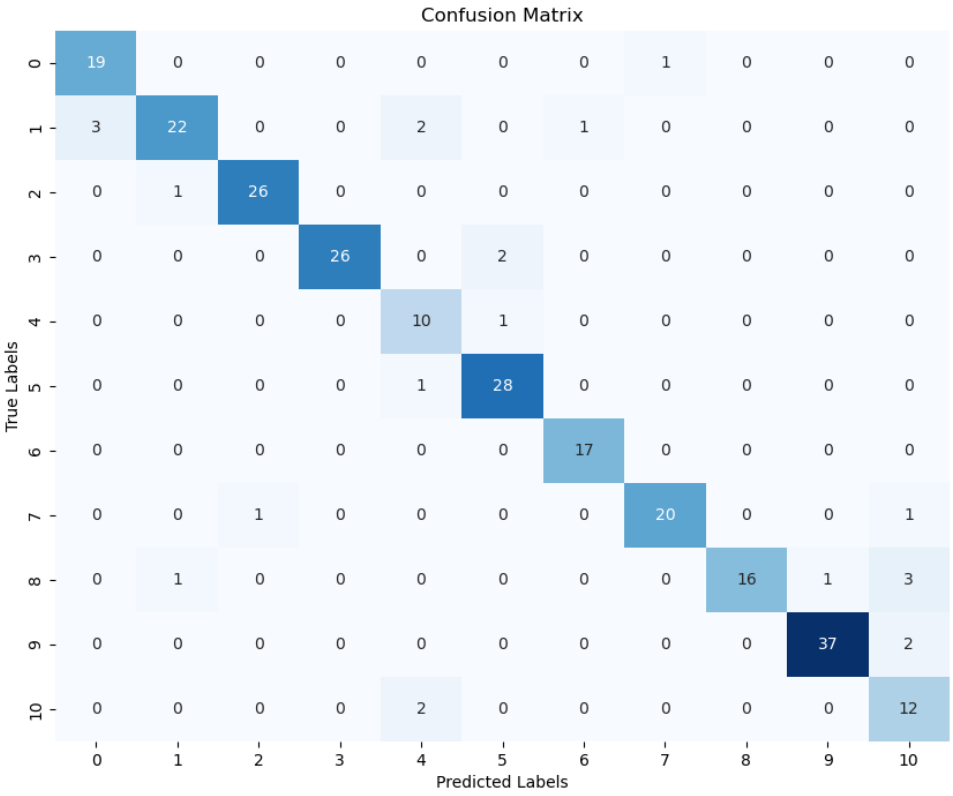

Figure 4.8: Multi Class Confusion Matrix Results For ResNet50

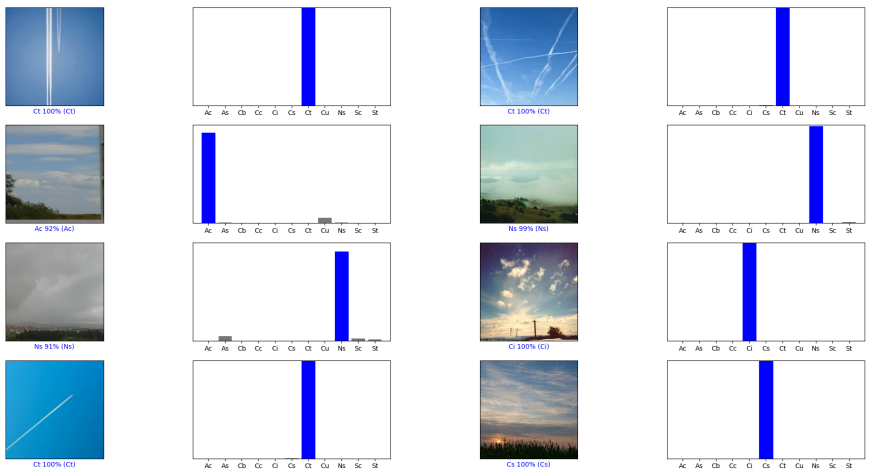

Figure 4.9: Sample From The Predictions Of The Test Dataset For ResNet50

4.2.2 MobileNetV3Small

The best-performing configuration for MobileNetV3Small was achieved using the categorical cross-entropy loss function with a learning rate of 0.0001 and a batch size of 32. This setup resulted in an accuracy of 94.53%, a precision of 95.29%, a recall of 93.62%, and an F1 score of 94.45%. This configuration outperformed others in terms of both accuracy and overall metric scores, making it the optimal choice for cloud classification tasks.

In contrast, using the cosine similarity loss function with the same learning rate and batch size achieved a slightly lower performance, with an accuracy of 92.96%, a precision of 93.73%, a recall of 92.11%, and an F1 score of 92.92%. Despite being slightly lower than the categorical cross entropy configuration, the cosine similarity results were still competitive and demonstrated the potential of this loss function in handling small datasets without pre-training.

The configurations with the lowest performance were those with the categorical cross entropy loss function and a very low learning rate of 0.00001, which achieved an accuracy of 60.93%, a precision of 72.44%, a recall of 40.01%, and an F1 score of 51.55%. This indicates that too low a learning rate significantly hampers the model's ability to learn effectively.

Based on these results, the final model implemented in the application was the one trained with the categorical cross entropy loss function, a learning rate of 0.0001, and a batch size of 32.

The experimental results for the MobileNetV3Small base model are summarized in this table 4.2.2

Loss Function	Learning Rate	Batch Size	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Categorical Cross Entropy	0.0001	32	94.53	95.29	93.62	94.45
Categorical Cross Entropy	0.0001	64	91.01	91.67	90.84	91.26
Categorical Cross Entropy	0.00001	32	60.93	72.44	40.01	51.55
Cosine Similarity	0.001	64	90.62	94.15	85.65	89.68
Cosine Similarity	0.0001	32	92.96	93.73	92.11	92.92
Cosine Similarity	0.0001	64	85.15	85.82	85.23	84.06

Table 4.2: Results of base model MobileNetV3Small

To further illustrate the performance of the best configuration (categorical cross-entropy with a learning rate of 0.0001 and batch size of 32), visual aids will be several presented below. These include the learning curves for training and validation, the multi-class confusion matrix, and a sample from the predicted test dataset.

Figure 4.16 displays the learning curves for both training and validation phases, as well as precision, recall, F1 Score and ROC. These metrics are plotted over each epoch. The training accuracy and validation accuracy curves show a consistent improvement, stabilizing around similar values towards the end of the training. This

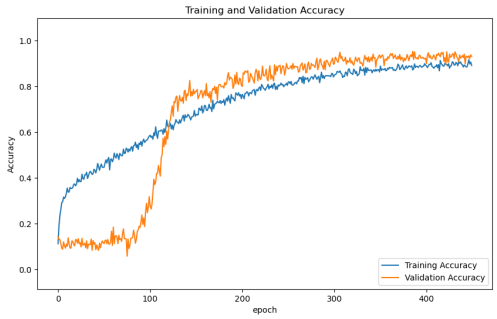

Figure 4.10: Accuracy

Figure 4.11: Loss

Figure 4.12: Precision

Figure 4.13: F1 Score

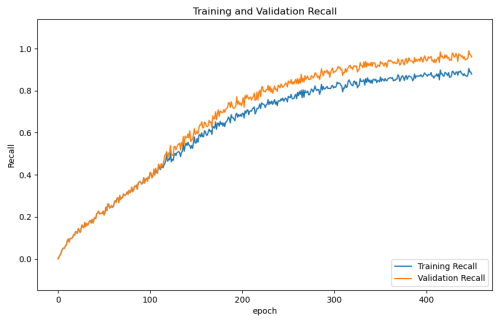

Figure 4.14: Recall

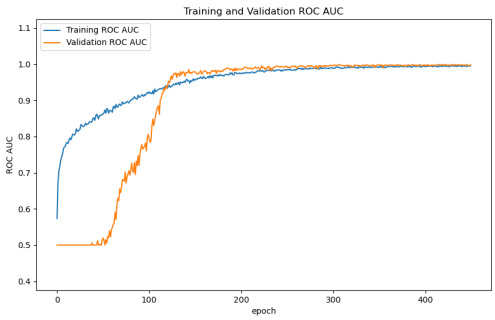

Figure 4.15: ROC AUC

Figure 4.16: Training And Validation Results For MobileNetV3Small

indicates that the model generalizes well without significant overfitting. The training and validation loss curves similarly show a decrease, which further confirms effective training.

Figure 4.17 shows the multi-class confusion matrix, which provides detailed insights into the classification performance for each cloud type. Every row in the matrix corresponds to the real class, and each column corresponds to the predicted class. The values along the diagonal indicate the amount of accurate predictions made for each class. The diagonal's high values show that the model accurately identified most samples for every type of cloud, demonstrating strong precision and

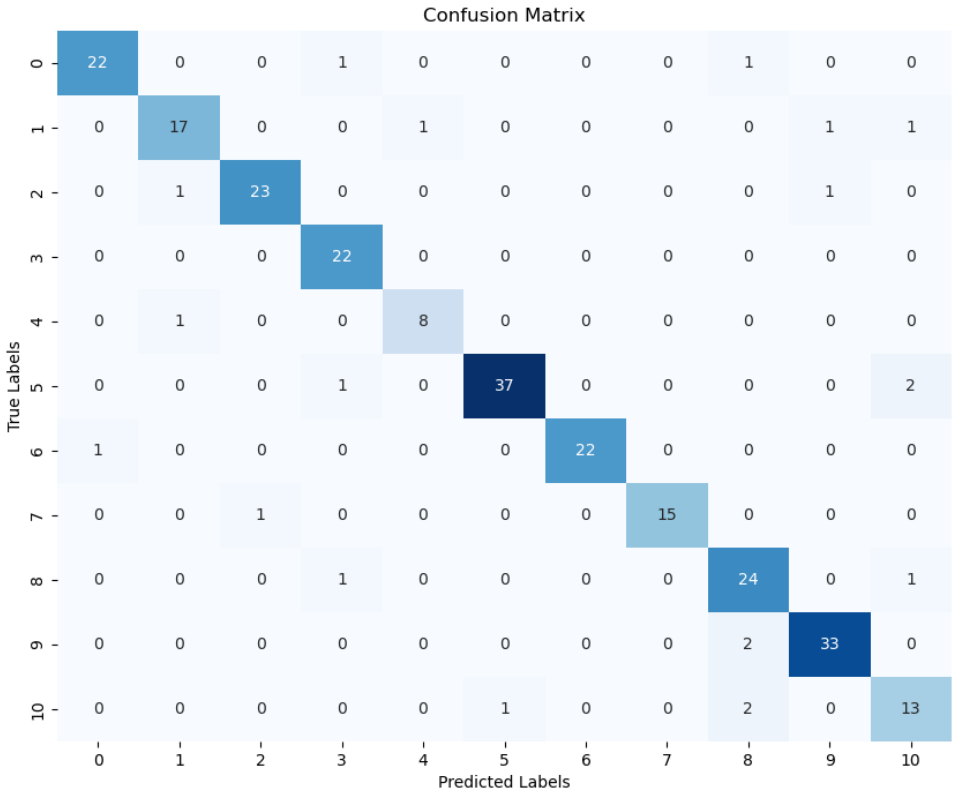

Figure 4.17: Multi Class Confusion Matrix Results For MobileNetV3Small

recall. However, there are some off-diagonal values indicating misclassifications. For example, the model occasionally confused certain classes, such as predicting class 2 instead of class 1, and class 8 instead of class 10. Despite these misclassifications, the overall performance reflects high precision and recall for the majority of the cloud types.

Figure 4.18 presents a selection of correctly and incorrectly classified samples from the test dataset. This qualitative analysis provides a visual understanding of how the model performs on real-world data. The correctly classified images show that the model accurately identifies various cloud types, while the misclassified samples help identify potential areas for further improvement.

4.2.3 EfficientNetV2B0

The best-performing configuration that used the EfficientNetV2B0 model as a base was achieved using the categorical cross-entropy loss function with a learning rate of 0.0001 and a batch size of 64. This setup resulted in an accuracy of 97.26%, a precision of 98.03%, a recall of 97.22%, and an F1 score of 97.62%. This configuration

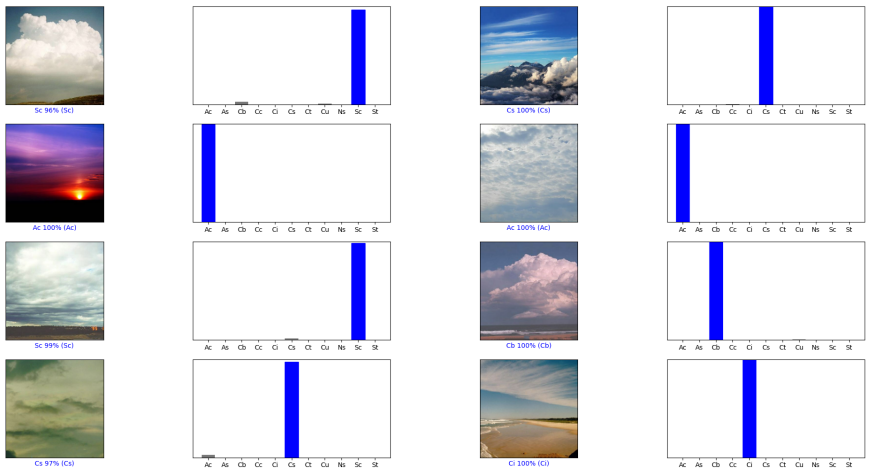

Figure 4.18: Sample From The Predictions Of The Test Dataset For MobileNetV3Small

outperformed others in terms of both accuracy and overall metric scores, making it the optimal choice for cloud classification tasks with the EfficientNetV2B0 model.

The configuration using the categorical cross entropy loss function with a learning rate of 0.0001 and a batch size of 32 also performed well, achieving an accuracy of 96.09%, a precision of 96.45%, a recall of 96.05%, and an F1 score of 96.25%. While slightly lower than the best-performing configuration, these results indicate strong performance and validate the effectiveness of the categorical cross entropy loss function for this model.

In contrast, using the cosine similarity loss function with a learning rate of 0.001 and a batch size of 64 resulted in an accuracy of 94.53%, a precision of 95.13%, a recall of 94.49%, and an F1 score of 94.81%. Although lower than the categorical cross entropy configurations, the cosine similarity results were still competitive, demonstrating its potential utility.

The configurations with the lowest performance were those with the categorical cross entropy loss function and a very low learning rate of 0.00001, which achieved an accuracy of 57.03%, a precision of 75.09%, a recall of 37.94%, and an F1 score of 50.41%. This indicates that too low a learning rate significantly hampers the model's ability to learn effectively.

Based on these results, the final model implemented in the application was the one trained with the categorical cross entropy loss function, a learning rate of 0.0001, and a batch size of 64. his configuration provided the highest overall performance, making it the best choice for the cloud classification task at hand.

The experimental results for the EfficienNetV2B0 base model are summarized in

this table 4.2.3

Loss Function	Learning Rate	Batch Size	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Categorical Cross Entropy	0.0001	32	96.09	96.45	96.05	96.25
Categorical Cross Entropy	0.0001	64	97.26	98.03	97.22	97.62
Categorical Cross Entropy	0.00001	32	57.03	75.09	37.94	50.41
Cosine Similarity	0.001	64	94.53	95.13	94.49	94.81
Cosine Similarity	0.0001	32	94.14	93.64	94.00	93.82
Cosine Similarity	0.0001	64	91.79	91.98	90.58	91.27

Table 4.3: Results of base model EfficientNetV2B0

To further illustrate the performance of the best configuration (categorical cross-entropy with a learning rate of 0.0001 and batch size of 64) for EfficientNetV2B0, we present several visual aids. These include the learning curves for training and validation, the multi-class confusion matrix, and a sample from the predicted test dataset.

Figure 4.25 presents a comprehensive view of the training and validation phases for EfficientNetV2B0. The plot illustrates the evolution of accuracy, loss, precision, recall, F1 Score, and ROC over each epoch. Notably, both the training and validation accuracy curves demonstrate consistent improvement throughout training. Additionally, the decreasing trends observed in the training and validation loss curves indicate that the model optimizes its performance over successive epochs, further validating its effectiveness in capturing relevant features and minimizing errors.

Figure 4.26 offers a detailed view of the multi-class confusion matrix, offering valuable insights into the classification performance across different cloud types. Each row corresponds to the actual class labels, while each column represents the predicted labels. Notably, the diagonal elements of the matrix depict the number of correct predictions for each class. The prominent values along the diagonal signify the model's proficiency in accurately classifying the majority of samples across various cloud types, underscoring its high precision and recall rates. This visualization provides a nuanced understanding of the model's classification capabilities and highlights its robustness in distinguishing between different cloud formations. Nevertheless, certain values not on the diagonal suggest misclassifications occurred, like class 7 being mistakenly classified as class 4 and class 10 being incorrectly labeled as class 3 and class 8. Even though there were some errors in classification, the model's strong ability to accurately differentiate between various types of cloud formations is shown by the high precision and recall rates. This particular visualization demonstrates a strong overall performance when contrasted with the confusion matrices presented before.

Figure 4.27 showcases a curated assortment of samples from the test dataset, featuring both accurately and inaccurately classified instances. This qualitative examination offers a tangible insight into the model's performance when applied to

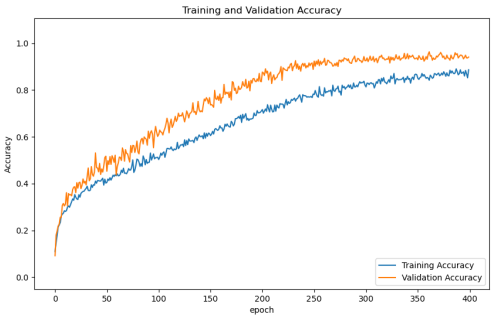

Figure 4.19: Accuracy

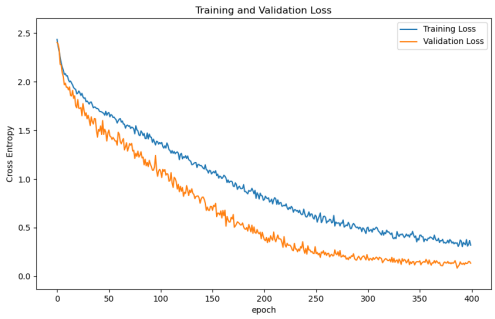

Figure 4.20: Loss

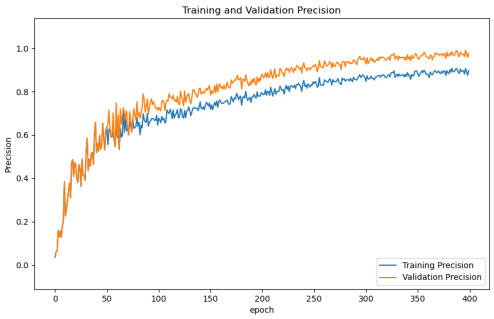

Figure 4.21: Precision

Figure 4.22: F1 Score

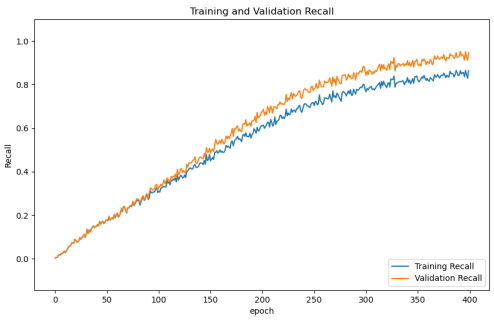

Figure 4.23: Recall

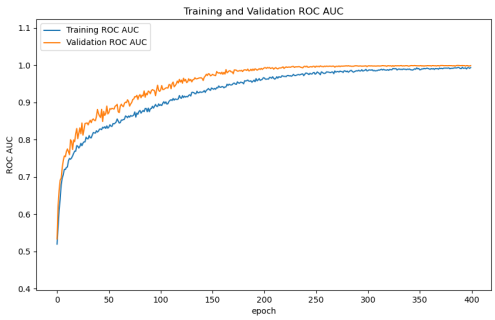

Figure 4.24: ROC AUC

Figure 4.25: Training And Validation Results For EfficientNetV2B0

real-world data. The accurately classified images highlight the model's adeptness in correctly identifying various cloud types, underscoring its proficiency in real-world scenarios.

These visual representations serve as invaluable additions to the numerical metrics, offering a complete view of how well the model works in cloud classification projects.

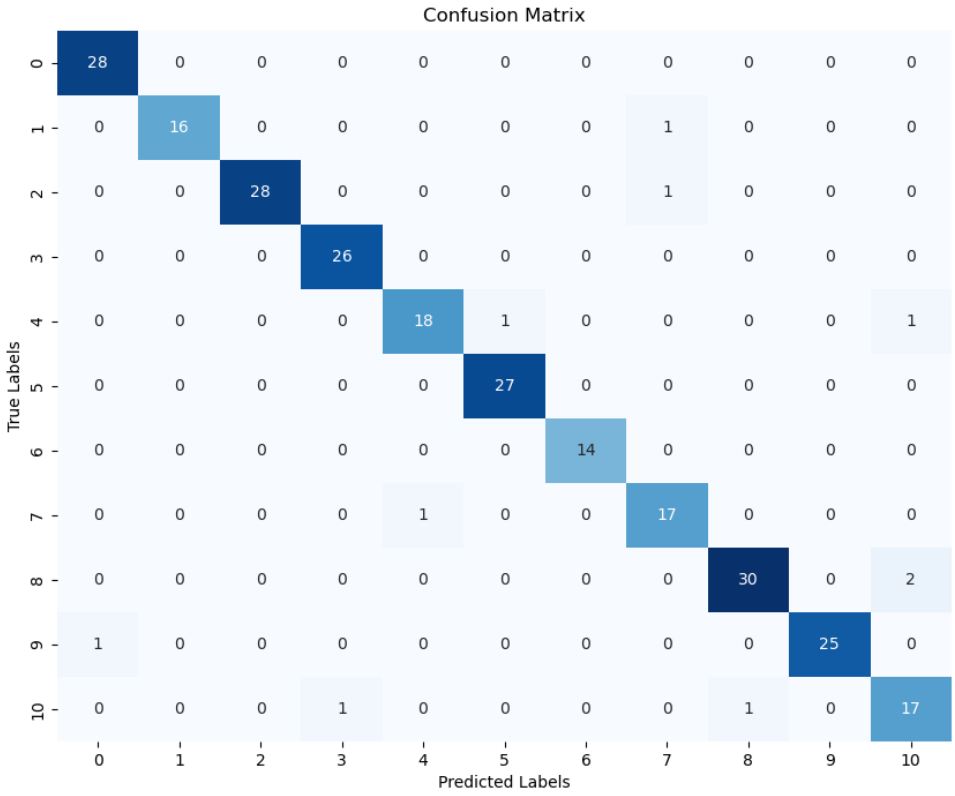

Figure 4.26: Multi Class Confusion Matrix Results For EfficientNetV2B0

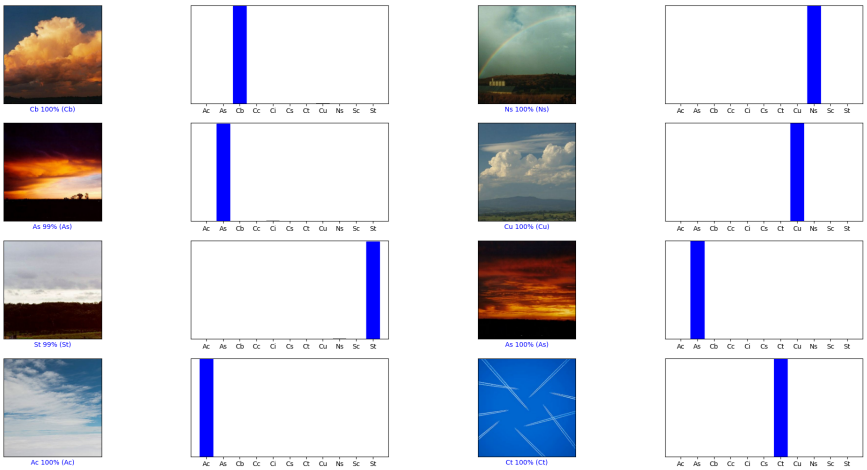

Figure 4.27: Sample From The Predictions Of The Test Dataset For EfficientNetV2B0

Study	Accuracy (%)
[ZLZS18]	88.8
[LQC+22]	92.70
[ZCH+22]	95.60
[GS23]	97.45
[GKB+24]	97.66
ResNet50	95.70
MobileNetV3Small	94.53
EfficientNetV2B0	97.26

Table 4.4: Comparison between the highest accuracy from recent literature and this study

4.2.4 State of the art comparison

The cloud classification models evaluated in this study, including ResNet50, MobileNetV3Small, and EfficientNetV2B0, exhibit commendable performance when juxtaposed with state-of-the-art solutions documented in existing literature. Notably, the study [ZLZS18] achieved an accuracy of 88.8%, followed by [GKB+24], which attained an accuracy of 97.66%. These studies reflect the evolving landscape of cloud classification methodologies, with the latter demonstrating substantial progress in achieving higher accuracy rates.

Comparing the achieved results to the state-of-the-art solutions, it's evident that the models developed in this study offer competitive performance. ResNet50, one of the base architectures employed, attained an accuracy of 95.70%, showcasing robust classification capabilities. Despite not surpassing the highest accuracy achieved in the literature, ResNet50 demonstrates reliability and effectiveness in cloud classification tasks.

Similarly, MobileNetV3Small, another base model utilized in this study, yielded an accuracy of 94.53%. While this accuracy falls slightly below the top-performing solutions in the literature, MobileNetV3Small maintains competitive performance levels, positioning it as a viable option for cloud classification applications.

The most noteworthy performance in this study was exhibited by the EfficientNetV2B0 model, which achieved an accuracy of 97.26%. This surpasses the accuracy reported in several prior studies and aligns closely with some of the most recent and advanced cloud classification models. The notable accuracy achieved by EfficientNetV2B0 underscores its efficacy and its potential as a leading choice for cloud classification tasks.

Incorporating models with uninitialized weights and applying data augmentation techniques played a pivotal role in enhancing the performance of the cloud classification models evaluated in this study. By initializing the models without pre-trained weights, we embraced the concept of "learning from scratch," allowing

the models to adapt specifically to the cloud classification task at hand. This approach ensures that the models are not biased towards features learned from unrelated datasets, thereby potentially improving their ability to generalize and capture cloud-related patterns effectively.

Moreover, the application of data augmentation techniques further strengthened the robustness and generalization capabilities of the models. Techniques such as random flips, rotations, contrast adjustments, brightness variations, and the addition of Gaussian noise introduced variability and diversity into the training data, mimicking real-world scenarios and augmenting the dataset's size. This augmentation process helps the models learn more robust and invariant representations of cloud images, thus improving their ability to generalize to unseen data and enhancing overall classification performance.

The effective strategy of using models with uninitialized weights and implementing data augmentation methods led to models better suited for dealing with the complexities and variations found in cloud images. The improved flexibility and ability to apply to various situations provided by these tactics probably played a role in the strong results seen in different base architectures, confirming the usefulness of these approaches in cloud classification tasks.

In summary, the achieved results, while not consistently surpassing the very best solutions reported in the literature, exhibit competitive performance across multiple base architectures. The attained accuracy validate the effectiveness of ResNet50, MobileNetV3Small, and notably EfficientNetV2B0 in cloud classification tasks, as a result contributing significantly to the expanding body of knowledge within this domain.

Chapter 5

Mobile Application

In order to actually use the models that I have presented in the previous chapter, I implemented a mobile application using Kotlin. The application houses 3 models for the user to choose from, the best performing instance for each base model presented: ResNet50, MobileNetV3Small and EfficientNetV2B0.

5.1 Technologies

The app, coded in Kotlin, serves as a platform for cloud classification using various deep learning models. It comprises two main activities: ChooseModel and PerformClassification. Each activity contributes to a seamless user experience, facilitating model selection and image classification.

The ChooseModel activity serves as the entry point, presenting users with a selection of deep learning models tailored for cloud classification. After entering the app, users encounter options such as ResNet50, MobileNetV3Small, and EfficientNetV2B0. These models were originally created using the Keras library, but converted using TensorFlow Lite for optimized deployment on mobile devices, ensuring minimal computational overhead and maximum performance efficiency. Additionally, the chosen ResNet50 model was further optimized by quantization to reduce its size, making it more suitable for deployment on mobile devices. The quantization process involved converting the model to a quantized format using TensorFlow Lite, which significantly reduced its size without compromising performance. These models differ in architecture and performance, catering to diverse user preferences and device capabilities. Upon selection, the activity takes users to the PerformClassification activity, passing along the chosen model's name. This streamlined transition ensures a straightforward user journey, enhancing engagement and usability.

In the PerformClassification activity, users engage in image selection, capture,

and subsequent classification based on the chosen deep learning model. This multifaceted process underscores the app's functionality and its integration with device hardware and software components. Users are presented with options to select an image from the device's gallery or capture one using the camera. Once an image is chosen or captured, it is displayed on the screen, offering users visual feedback and enhancing interaction. The core functionality of image classification is initiated upon user action. The model is invoked for inference, generating predictions regarding the cloud type depicted in the image. The classification result is then displayed to the user, providing valuable insights into the atmospheric phenomena captured in the image.

Beyond its primary functionalities, the app incorporates several features to enhance user experience and performance. Edge-to-edge display utilization ensures optimal utilization of screen real estate, fostering an immersive interaction environment. Furthermore, window insets are employed to adjust UI elements dynamically, ensuring consistent visual presentation across devices and orientations.

A key aspect of the app's functionality lies in its utilization of a text file containing the classification labels. This file serves as a mapping mechanism, associating model outputs with human-readable labels. By providing contextual information about the classification results, users can better interpret and comprehend the app's output, enriching their overall experience.

5.2 System Design

The essence of user interaction within the app is encapsulated in a use case diagram. Here, users are presented with distinct operations, each facilitating a unique aspect of their engagement with the app's functionalities. Central to this interaction are three primary use cases: Choosing Classification Models, Uploading Images, Viewing Predictions.

As previously stated the user selects a classification model and should either take or upload a picture from the galley. The predictions are then shown, once the necessary data has been submitted and the ”predict” button has been pressed, as seen in Figure 5.1.

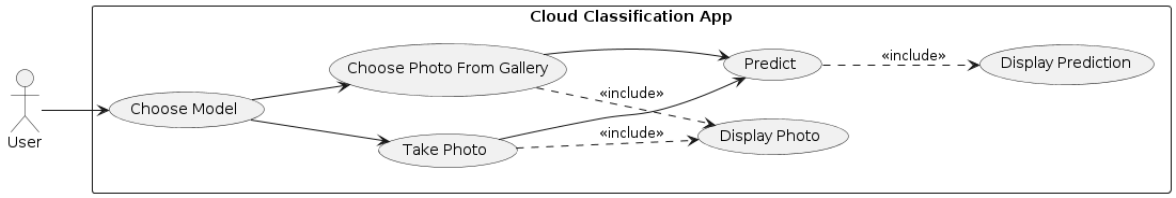

Figure 5.1: Use Case Diagram

The sequence diagram shows the sequential order in which the interactions between the system's components take place. The selected model's name is delivered to the next screen whenever the user selects the model they wish to utilize. This screen requires the user to take a picture or select one from the gallery after which it is shown on the screen. Following the display of the image and the pressing of the prediction button, inference occurs on the selected model based on information received from the previous screen, and the user is presented with the prediction result, as seen in Figure 5.2.

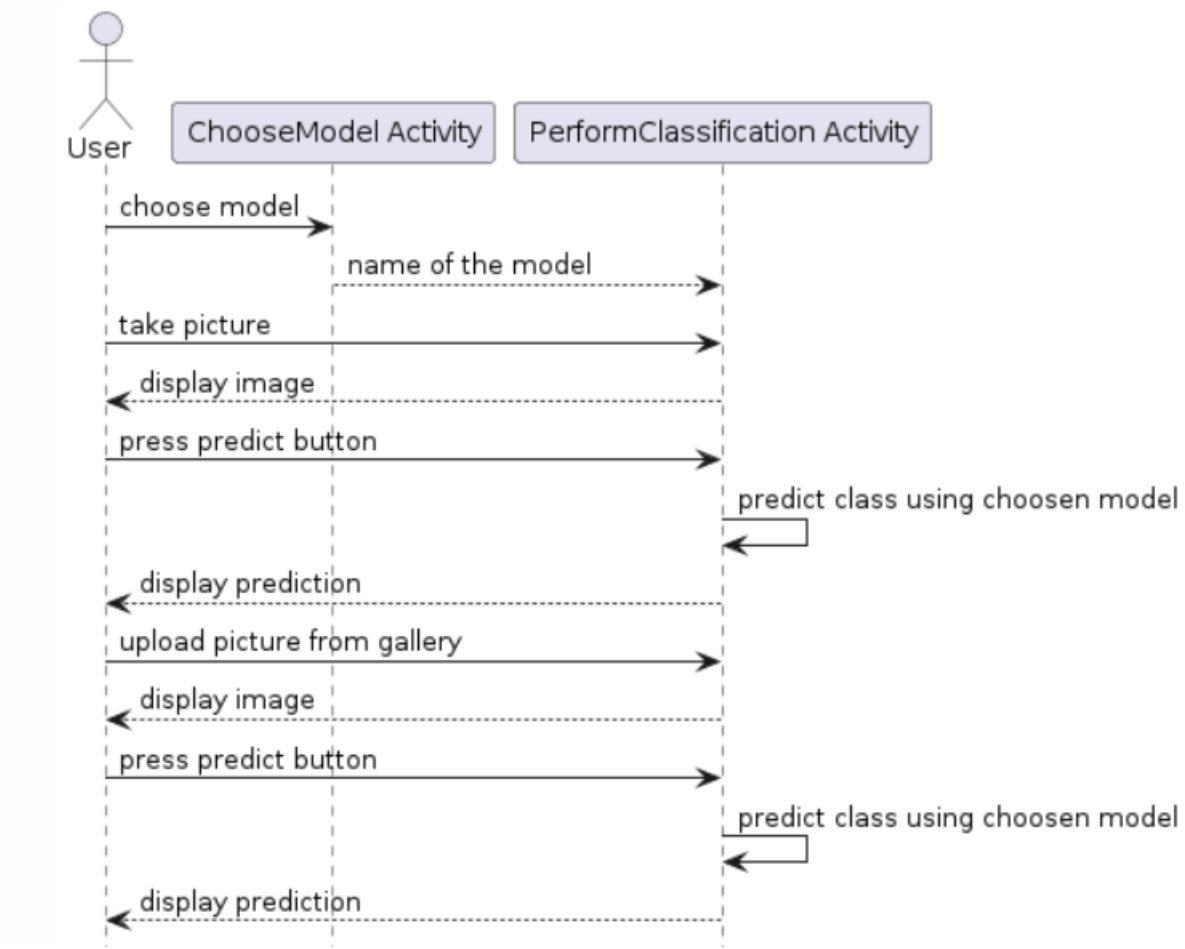

Figure 5.2: Sequence Diagram

Chapter 6

Conclusions

In conclusion, this thesis delved into the realm of cloud classification, an essential task for improving weather forecasting and understanding atmospheric phenomena. By leveraging machine learning models, more specifically ResNet50, MobileNetV3Small, and EfficientNetV2B0, we aimed to accurately classify various cloud types from ground-based images. Among the models explored, the best result was achieved using EfficientNetV2B0 as a base model, which attained an accuracy of 97.26%. While the results were competitive, they may not consistently surpass the highest accuracy reported in recent literature. However, they affirm the effectiveness of the chosen base architectures in cloud classification tasks and contribute to the growing body of knowledge in this domain.

Moreover, the utilization of models with uninitialized weights, combined with rigorous data augmentation techniques, proved crucial in enhancing classification performance. This underscores the importance of carefully selecting model architectures and preprocessing techniques to achieve optimal results in cloud classification tasks.

Moving forward, future work in this area could focus on addressing several limitations encountered during this research. These include the constraint of restricted hardware resources, which interfered with the exploration of a broader range of parameter combinations during model training. Additionally, the relatively small size of the available dataset posed challenges, suggesting the need for expanding the dataset with more diverse images to improve classification performance.

In addition to the research contributions made in this thesis, there are exciting opportunities to translate these findings into practical applications, particularly through the development of a cloud classification app. This app could serve as a valuable tool for cloud enthusiasts, meteorologists, and weather enthusiasts alike, providing them with a user-friendly platform to accurately identify and classify different cloud types in real-time. By leveraging the power of machine learning models like ResNet50, MobileNetV3Small, and EfficientNetV2B0, users can obtain instanta-

neous and accurate information about prevailing cloud formations, contributing to a deeper understanding of atmospheric conditions.

Moreover, integrating these developed models into weather prediction systems holds significant potential for enhancing the accuracy and reliability of weather forecasts. By incorporating cloud classification data obtained from the app into existing weather models, meteorologists can refine their predictions and gain insights into short-term and long-term weather patterns. This integration could lead to more precise weather forecasts, aiding in disaster preparedness, agricultural planning, and resource management.

Furthermore, beyond its utility in cloud classification, the app could also find applications in environmental monitoring and ecological research. For example, researchers could utilize the app to track changes in cloud cover over time, providing valuable insights into climate trends and environmental changes. Additionally, the app could be deployed in citizen science initiatives, allowing users to contribute cloud classification data to larger-scale research projects aimed at understanding the impacts of climate change on cloud dynamics and atmospheric processes.

As the field of artificial intelligence continues to advance, we anticipate the emergence of superior algorithms and methodologies that will further enhance cloud classification and related tasks. Continued exploration and innovation in this domain hold the potential to yield valuable insights into our natural environment and contribute to the ongoing efforts to address environmental challenges.

Bibliography

[Ama93] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.

[AZH+21] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.

[BD20] Bjorn Barz and Joachim Denzler. Deep learning on small datasets without pre-training using cosine loss. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1371–1380, 2020.

[CPG05] Josep Calbó, David Pages, and Josep-Abel González. Empirical studies of cloud effects on uv radiation: A review. *Reviews of Geophysics*, 43(2), 2005.

[DLW15] Soumyabrata Dev, Yee Hui Lee, and Stefan Winkler. Categorization of cloud image patches using an improved texton-based approach. In *2015 IEEE international conference on image processing (ICIP)*, pages 422–426. IEEE, 2015.

[Fit17] Richard J Fitzgerald. International cloud atlas. *Physics Today*, 70(5):76–76, 2017.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[GKB+24] Mehmet Guzel, Muruvvet Kalkan, Erkan Bostanci, Koray Acici, and Tunc Asuroglu. Cloud type classification using deep learning with cloud images. *PeerJ Computer Science*, 10:e1779, 2024.

[GS23] Emmanuel Kwabena Gyasi and Purushotham Swarnalatha. Cloudmobinet: An abridged mobile-net convolutional neural network model for ground-based cloud classification. *Atmosphere*, 14(2), 2023.

[HJ14] Robert A Houze Jr. *Cloud dynamics*. Academic press, 2014.

[HZC+17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[LQC+22] Xiaotong Li, Bo Qiu, Guanlong Cao, Chao Wu, and Liwen Zhang. A novel method for ground-based cloud image classification using transformer. *Remote Sensing*, 14(16), 2022.

[MA21] Sheldon Mascarenhas and Mukul Agarwal. A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification. In *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, volume 1, pages 96–99. IEEE, 2021.

[MMZ23] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *International Conference on Machine Learning*, pages 23803–23828. PMLR, 2023.

[Nor00] Joel R Norris. What can cloud observations tell us about climate variability? *Space Science Reviews*, 94(1):375–380, 2000.

[Ran03] Arthur L Rangno. The classification of clouds. *Handbook of Weather, Climate, and Water: Dynamics, Climate, Physical Meteorology, Weather Systems, and Measurements*, pages 387–405, 2003.

[TL19] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[XCZ+16] Yang Xiao, Zhiguo Cao, Wen Zhuo, Liang Ye, and Lei Zhu. mcloud: A multiview visual feature extraction mechanism for ground-based cloud image categorization. *Journal of Atmospheric and Oceanic Technology*, 33(4):789–801, 2016.

[YCX17] L Ye, Z Cao, and Y Xiao. Deepcloud: Ground-based cloud image categorization using deep convolutional features, ieee t. geosci. remote, 55, 5729–5740, 2017.

[ZCH+22] Wen Zhu, Tianliang Chen, Beiping Hou, Chen Bian, Aihua Yu, Lingchao Chen, Ming Tang, and Yuzhen Zhu. Classification of ground-based cloud images by improved combined convolutional network. *Applied Sciences*, 12(3), 2022.

[ZLZS18] Jinglin Zhang, Pu Liu, Feng Zhang, and Qianqian Song. Cloudnet: Ground-based cloud classification with deep convolutional neural network. *Geophysical Research Letters*, 45(16):8665–8672, 2018.