Università di Catania

UNIVERSITY OF CATANIA

Department of Mathematics and Computer Science

Ph.D. in Computer Science - XXXVII Cycle

Alessio Barbaro Chisari

Enhancing Atmospheric Particles Detection with Deep Learning and Computer Vision Techniques

---

Ph.D. Thesis

Supervisor: Prof. Sebastiano Battiato

Co-supervisor: Dr. Mario Valerio Giuffrida

Academic Year 2023 - 2024

Declaration of Authorship

I, Alessio Barbaro Chisari, declare that I have written this thesis, titled "Enhancing Atmospheric Particles Detection with Deep Learning and Computer Vision Techniques", independently and have not used any sources, resources or technical tools other than those specified. All statements that were taken from other publications, either literally or analogously, are marked. I have not submitted the work in the same or a similar form to any other examination authority. I agree that this work may be checked with anti-plagiarism software. Parts of this thesis have been submitted or published in conferences (*VISAPP 2022*, *ICIP 2024*, *ESANN 2024*, *MetroXRAINE 2024*) and journals (*IEEE Access*, *The Visual Computer* by Springer).

Abstract

This thesis presents a novel framework aimed at enhancing the detection and classification of atmospheric particles through the integration of deep learning and computer vision techniques. Motivated by the need for accurate environmental monitoring and robust decision support systems, particularly in regions affected by volcanic activity and extreme weather conditions, the research focuses on developing a Multimodal AI Engine that leverages heterogeneous data sources to provide reliable real-time insights.

The study introduces innovative methodologies for preprocessing raw lidar-based ceilometer data, transforming it into high-resolution, time-indexed images that serve as a robust foundation for training advanced deep learning models. A comprehensive benchmarking of various architectures, including Convolutional Neural Networks (CNNs) and Vision Transformers (ViT), is performed to evaluate their performance in detecting and classifying atmospheric phenomena. Moreover, the thesis explores the integration of Transfer Learning and Federated Learning paradigms, which not only improve model generalizability but also address critical issues related to data privacy and computational scalability.

Key contributions of this work include the development of novel datasets tailored to atmospheric particle detection, the evaluation of deep learning models under diverse operational scenarios, and the demonstration of practical applications. The research further highlights the potential of advanced AI methodologies to facilitate informed decision-making in high-stakes environmental contexts.

The findings indicate that the proposed framework significantly enhances the

accuracy and efficiency of atmospheric particle detection, while also identifying several limitations related to data quality, model interpretability, and computational resource demands. These insights provide a solid foundation for future research aimed at refining data fusion techniques, improving model transparency, and optimizing the deployment of AI systems in real-world settings.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Sebastiano Battiato, for his constant guidance, valuable advice, and continuous support throughout the course of this Ph.D. His mentorship, expertise and encouragement have been essential in shaping both my research, academic and personal growth.

My sincere thanks go to my co-supervisor, Dr. Valerio Giuffrida, for his invaluable scientific input and mentoring. I am especially grateful to him for hosting me during my research stay abroad at the University of Nottingham, an experience that has been both professionally enriching and personally meaningful.

I am also sincerely thankful to Vladimiro for his consistent support during my research activities at EHT and beyond. His availability, advice, and encouragement have played an important role in the development of this work and, of course, in my personal and professional growth. Thanks also to Roberto, Antonio, Luca and Andrea.

Special thanks to Bruno, true and sincere friend. Your help in the darkest period of my Ph.D. journey will never be forgotten, thank you from the bottom of my heart.

My deepest thanks go to Dr. Luca Guarnera for his generous guidance, collaborative spirit, and for always being willing to share his expertise and time.

To my family, the strongest pillar in my life. To my father Roberto and my mother Daniela, always ready to advise and support me with love and wisdom. To my sister Jessica, a constant and precious presence in every aspect of my life. And to my brother Edoardo, always by my side. Thank you for everything you do for me, every single day.

To Erminia, with whom I've shared every moment of this journey (as always). Thank you for your love, your support, and your patience. Your presence has been

fundamental through every challenge and every success. Thank you for always being there, even in the hardest moments, and for being my first supporter.

To my lifelong friends, those who have always been there, through every season: no list could ever be complete, but you know who you are, and you know how deeply I care for you.

A special thank you to Rino, Matteo, Jordan, and also Stefano and Valerio. Thank you for always being there.

I am grateful to all the colleagues and friends from the Department of Mathematics and Computer Science at the University of Catania for the stimulating discussions, collaboration, and the positive environment throughout these years.

To all those who, in one way or another, have been part of this path.

Thank you.

Contents

List of Figures

List of Tables

Part I

Introduction and Background

Chapter 1

Introduction

1.1 Motivation

The climate plays an undeniable role in shaping the lives of individuals and societies. The influence of weather events can be deep and affect everything from transport and energy production to the safety and well-being of people. For example, extreme weather conditions can lead to the closure of airports, disrupt public transport and damage infrastructure. In more extreme cases, weather phenomena can cause catastrophic events that put life and property at extreme risk. In these scenarios, it is clear that there is a need for tools aimed at preventing and forecasting phenomena of this type, using the best detection instruments in this field.

In particular, in Catania (Sicily, Italy), the eruption of the Mount Etna (one of the highest active volcanoes in Europe) represents a significant challenge. The volcanic ash it produces can impact daily life. Airports in the region are often forced to close down, causing flights to be cancelled and causing widespread travel disruption. It is not only Catania airport that suffers severe disruptions, but also the nearby airports as far as Reggio Calabria (Calabria, Italy) or Comiso (Sicily, Italy), as reported in Figure 1.1. For example, studies have shown that volcanic ash

Figure 1.1: Map of Sicily showing the location of major airports affected by volcanic activity from Mount Etna.

clouds can have severe impacts on aviation, as was evidenced by the 2010 eruption of Eyjafjallajökull in Iceland [51], which led to the largest air-traffic shutdown since World War II. But it is not only airports that suffer from such events. The build-up of volcanic ash on photovoltaic panels significantly reduces their efficiency, hindering the production of solar energy. In addition, volcanic ash can damage buildings, vehicles and crops, causing significant economic losses.

Given the critical impact of these events, the necessity of effective decision-making tools is crucial. Decision makers need accurate and well-timed information to respond effectively to these challenges. In recent years, AI has been increasingly integrated into various fields, demonstrating its potential in improving decision-making processes in a wide range of applications. Exploiting AI in the context of weather phenomena, particularly those related to volcanic ash and other atmospheric

particles, can provide valuable support to decision-makers.

While Mount Etna's frequent eruptions provide a compelling case study for the detection and classification of volcanic ash, the ultimate aim of this research is to advance our ability to forecast the movement of suspended particulates in the atmosphere more generally. The methodologies developed herein—namely, the transformation of ceilometer backscatter profiles into time-indexed visual representations and the application of deep learning models (supported by transfer and federated learning paradigms)—are designed to distinguish between diverse particulate types (e.g., volcanic ash, industrial emissions, Saharan dust and meteorological hydrometeors such as cloud droplets, rain, snow and hail). By demonstrating robust performance on tasks like cloud detection, this thesis lays the groundwork for extending its predictive engine to other scenarios of environmental and public-safety importance, including the dispersion of pollutants following industrial accidents and the transboundary transport of mineral dust. In so doing, it contributes to a more comprehensive decision-support framework capable of anticipating particulate distributions across multiple temporal and spatial scales.

In this way, the motivation of this research is strongly based on the desire to mitigate the effects of adverse weather events and improve the resilience of systems and infrastructure by recognising the different types of particles within the atmosphere. Using AI, and in particular applying advanced deep learning and computer vision techniques, this research aims to develop tools that can analyse and classify particles more accurately and efficiently. The long-term aim is to provide decision-makers with reliable information to help them make informed decisions, via a Multimodal AI Engine, thus reducing the negative impact of these events.

The importance of such research cannot be underestimated, especially in a region

like Catania, where the effects of Mount Etna's eruptions are a regular occurrence. The development of AI-driven tools that can predict and analyze these events in real-time offers a promising solution to the challenges faced by the region. Moreover, the broader implications of this research extend beyond volcanic activity, with potential applications in various atmospheric conditions that affect transportation, energy production, and public safety.

In summary, this research is motivated by the need to develop robust tools that leverage the power of AI to mitigate the impacts of adverse atmospheric events. By focusing on the unique challenges posed by Mount Etna and other atmospheric phenomena, this research aims to contribute to the broader effort to enhance resilience and improve decision-making processes in the face of atmospheric multi-hazard challenges.

1.2 Objectives

The primary objective of this research is the development of a Multimodal AI Engine capable of providing robust decision support in the context of atmospheric phenomena, particularly those involving volcanic ash and other airborne particles. The proposed AI Engine will be designed to integrate data from various meteorological instruments and models, sensors, allowing for a comprehensive analysis of atmospheric conditions, using a Trusted Execution Environment (TEE).

The application of neural networks, particularly deep learning models, has shown big potential in various fields, and this research seeks to harness that potential in the domain of atmospheric analysis. By employing advanced computer vision techniques, the AI Engine will be capable of detecting and classifying atmospheric

particles, providing real-time insights that can assist decision-makers in responding to environmental challenges.

Rather than treating each input separately, this thesis adopts a practical framework to draw together different types of data, including lidar, as well as data from high-resolution meteorological models. This approach not only enhances the accuracy of atmospheric particle detection but also provides a more holistic understanding of atmospheric conditions, thereby improving the reliability of the AI Engine's outputs.

Moreover, the research aims to address the challenges associated with the scarcity and quality of data in this domain. By developing novel methods for data preprocessing and augmentation, the research seeks to overcome these challenges, ensuring that the AI models are trained on high-quality data that accurately represents the complexity of atmospheric phenomena.

Ultimately, the goal is to create an AI-driven tool that can be deployed in real-world scenarios, providing decision-makers with the information they need to make timely and informed decisions. Whether it is managing air traffic during a volcanic eruption or optimizing the performance of solar panels in the face of adverse weather conditions, the AI Engine developed through this research has the potential to make a significant impact. To ensure data privacy and model integrity, the system leverages TEE, which enable secure processing of sensitive information during both training and inference. Furthermore, adopting a federated learning paradigm allows the model to be trained across multiple data sources without exposing proprietary data, thereby enhancing both security and scalability.

1.3 Contributions

The scientific contributions of this research are several and can be categorized in different key areas:

- **Development of Novel Datasets**: One of the primary contributions is the creation of innovative datasets derived from raw lidar data, which were processed into hourly images representing various atmospheric conditions. These datasets are not only unique but also crucial for training and benchmarking AI models in the context of atmospheric analysis.

- **Benchmarking of Deep Learning Architectures**: The research includes a comprehensive benchmarking of existing deep learning architectures, such as Vision Transformers (ViT) and Convolutional Neural Networks (CNNs), for the task of atmospheric particle detection. The results highlight the strengths and limitations of these models, providing valuable insights for future research.

- **Integration of Multimodal Data**: Another significant contribution is the integration of multimodal data from various meteorological instruments and models. This approach enhances the accuracy and robustness of the AI Engine, allowing it to provide more reliable outputs.

- **Application of Transfer and Federated Learning**: The research also explores the application of transfer and federated learning in the context of atmospheric analysis. This approach enables the training of AI models across multiple distributed datasets, enhancing the generalisability of the models while preserving data privacy.

- **Practical Applications**: The research demonstrates the practical applications of the developed AI Engine in real-world scenarios, such as improving the management of air traffic during volcanic eruptions and optimizing the performance of solar panels in adverse weather conditions.

- **Scientific Publications**: The research has led to several scientific publications, contributing to the body of knowledge in the fields of deep learning, computer vision, and atmospheric science. Scientific publications during the Ph.D. are listed below (* denotes equal contribution):

  - **Conference** Casella, B.*, Chisari, A. B.*, Battiato, S., & Giuffrida, M. V. (2022). *Transfer learning via test-time neural networks aggregation.* In proceedings VISAPP 2022. [20]

  - **Conference** Chisari, A. B., Ortis, A., Guarnera, L., Patatu, W. C., Giandolfo, R. A., Spampinato, E., Battiato, S., & Giuffrida, M. V. (2024). *On the cloud detection from backscattered images generated from a lidar-based ceilometer: Current state and opportunities.* In 2024 IEEE International Conference on Image Processing (ICIP) (pp. 144-150). IEEE. [30]

  - **Conference** Casella, B.*, Chisari, A. B.*, Aldinucci, M., Battiato, S., & Giuffrida, M. V. (2024). *Federated Learning in a Semi-Supervised Environment for Earth Observation Data.* In ESANN 2024 Proceedings-32th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (pp. 93-98). Michel Verlesian. [19]

  - **Journal** Chisari, A. B., Guarnera, L., Ortis, A., Patatu, W. C., Casella, B., Naso, L., Puglisi, G., Del Zoppo, V., Giuffrida, M. V., & Battiato, S.

(2025). *Cloud Detection Challenge - Methods and Results*. IEEE Access. [29]

– **Journal** Chisari, A. B., Guarnera, L., Ortis, A., Patatu, W. C., Battiato, S., & Giuffrida, M. V. (2025). *Benchmarking computer vision architectures for cloud detection from lidar ceilometer backscatter data*. The Visual Computer, 1-18. [28]

1.4 Thesis Outline

The structure of this thesis is as follows:

- **Chapter 1: Introduction** - This chapter provides an overview of the motivation, objectives, and contributions of the research, setting the stage for the detailed discussions that follow.
- **Chapter 2: Background on AI** - This chapter seeks to clarify the fundamental theoretical concepts necessary for a thorough comprehension of the thesis, like Machine Learning, Deep Learning, Transfer Learning, Federated Learning and Computer Vision.
- **Chapter 3: Introduction to Ceilometers** - This chapter introduces the instrumentation used in the research, particularly the types of ceilometers employed, the variables they measure, and how they were utilized in the study.
- **Chapter 4: Cloud Detection** - This chapter presents a cloud detection framework, detailing the methodology used in the research, including data preprocessing, image construction, and the labelling process using the Weather Research & Forecasting (WRF) model. It also presents the benchmarking of

deep learning architectures for cloud detection. Results highlight the influence of hyperparameters, data characteristics, and model choice on detection accuracy.

- **Chapter 5: Cloud Detection Challenge** - In this chapter, a cloud detection challenge is introduced to evaluate competing methods on a shared benchmark, showcasing the practical relevance and generalisability of the proposed approach across academic and industrial teams.

- **Chapter 6: Proposed Transfer Learning Approach** - The chapter explores the application of Transfer Learning to this context, using a novel test-time neural network aggregation strategy within a transfer learning paradigm, enhancing model performance across domains while addressing challenges such as selective forgetting and commutativity.

- **Chapter 7: Proposed Federated Learning Approach** - In this chapter, a federated learning framework is proposed to enable decentralised model training across multiple ceilometers, ensuring data privacy and scalability while maintaining competitive accuracy in heterogeneous environments.

- **Chapter 8: Final Discussion and Future Works** – The final chapter discusses the results of the research, drawing conclusions and outlining potential avenues for future work.

Chapter 2

Background on AI

This chapter seeks to clarify the fundamental theoretical concepts necessary for a thorough comprehension of the thesis. It begins by presenting key terminology related to Machine Learning, followed by an in-depth discussion of concepts and definitions relevant to the specific challenges addressed in this study, with particular emphasis on deep convolutional architectures. The fundamentals of Transfer Learning and Federated Learning, methodologies used in the work presented in this thesis, will also be introduced.

2.1 Machine Learning

The expanding field of Machine Learning (ML) within Artificial Intelligence (AI) enables computational systems to extract insights from historical data or past experiences, supporting the development of models for classification and prediction through statistical analysis and pattern recognition [62]. Unlike explicitly programmed approaches, ML algorithms refine themselves autonomously over time, improving their effectiveness in performing specific tasks [90]. ML tasks are generally defined by how the system processes input examples, typically represented as

structured features such as pixel values in images or frames in videos, which form the basis of datasets.

Machine learning tasks are typically categorized into different paradigms, each designed to address specific objectives within various data-driven contexts. Among these, classification and regression represent two fundamental approaches with extensive real-world applications [36]. Classification involves assigning input data to predefined categories based on recognizable patterns, and it is widely used in areas such as image recognition, natural language processing, and medical diagnostics. Key ML classification algorithms include Support Vector Machines (SVM) [31], decision trees [14], and k-Nearest Neighbors (k-NN) [2]. In contrast, regression focuses on predicting continuous numerical values from input features, helping to identify trends and relationships within datasets. This approach is particularly relevant in finance, economics, and engineering, where accurate forecasting and quantitative analysis are essential. Standard regression techniques include linear regression and polynomial regression [84].

Evaluating the performance of ML models requires the use of quantitative metrics. In classification tasks, commonly used metrics include accuracy, which measures the proportion of correct predictions, and error rate, which accounts for incorrect predictions. For regression models, evaluation is often based on Mean Squared Error (MSE) and Mean Absolute Error (MAE). MSE determines the average squared difference between predicted and actual values, capturing variance in predictions, while MAE measures the average absolute difference, providing a robust error assessment that is less influenced by outliers [115]. These metrics are formally defined as follows:

$$\text{Accuracy} = \frac{\#\text{ Correct predictions}}{\#\text{ Data samples}}, \tag{2.1}$$

$$\text{Error Rate} = \frac{\#\text{ Incorrect predictions}}{\#\text{ Data samples}}, \tag{2.2}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{2.3}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{2.4}$$

where y_i represents the actual value for the i-th instance, $\hat{y}_i$ denotes the predicted value, and n is the total number of instances.

ML algorithms can be categorized into three primary learning paradigms based on their training process: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning relies on labeled data to make predictions or decisions [61], while unsupervised learning focuses on identifying patterns and structures in unlabeled datasets [21]. Reinforcement learning, in contrast, involves iterative decision-making within dynamic environments, where an agent learns by interacting with its surroundings and receiving feedback in the form of rewards or penalties [6].

The effectiveness of ML models depends on several factors, including data quality, feature selection, model complexity, and the choice of algorithmic parameters [13]. Techniques such as cross-validation and regularization are frequently employed to prevent overfitting and enhance the model's ability to generalize to new data. Furthermore, model interpretability remains a critical issue, especially in domains where

transparency and accountability are essential, such as healthcare and criminal justice [78]. Addressing these concerns requires the development of explainable AI methodologies that clarify how ML models make decisions, fostering greater trust and reliability among users.

2.2 Deep Learning

Deep learning (DL) has emerged as a significant advancement in machine learning (ML), distinguishing itself from conventional ML techniques that typically depend on manually designed feature engineering and shallow models. In contrast, deep learning models harness hierarchical representations that are automatically learned from the data itself, enabling the extraction of intricate features across various levels of abstraction [70]. This fundamental shift in paradigm has been made possible by the evolution of neural network structures, which have progressively increased in complexity and depth over time, giving rise to deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs).

The history of neural networks can be traced back to the perceptron, introduced by Frank Rosenblatt in 1958. Drawing inspiration from biological neurons, the perceptron sought to simulate the learning and decision-making processes observed in the human brain. It consisted of a single layer of interconnected nodes (or neurons), each with its own weight and threshold, and performed binary classification by computing a weighted sum of input features, followed by a threshold-based activation function [118]. Despite its conceptual elegance, the perceptron's ability was limited to solving only linearly separable problems, where a single hyperplane could separate data points belonging to different classes. Complex problems, such as XOR

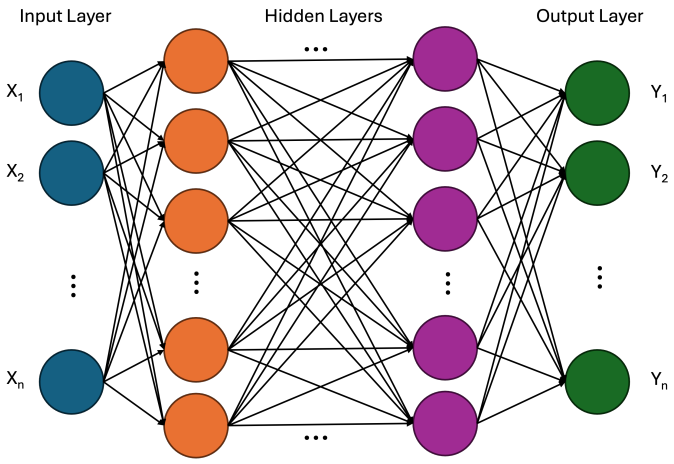


Figure 2.1: A visual representation of the Multi-Layer Perceptron (MLP).

classification, which are not linearly separable, rendered the perceptron inadequate for such tasks [88].

A breakthrough in neural network development came in the 1970s with the introduction of the multi-layer perceptron (MLP), displayed in Figure 2.1, which stacked multiple layers of neurons to learn more complex, non-linear decision boundaries. This allowed MLPs to represent complex patterns by composing simpler functions. Each neuron in an MLP receives inputs from neurons in the previous layer, computes a weighted sum, and applies a non-linear activation function to produce an output [147]. The architecture of MLPs typically includes an input layer, one or more hidden layers, and an output layer:

- **Input layer**: The first layer of the network, representing the features or attributes of the input data. It serves as the entry point for data into the network but does not perform computations.
- **Hidden layers**: Intermediate layers between the input and output layers. Neurons in these layers compute weighted sums of inputs from the previous

layer and apply non-linear activation functions to capture complex relationships.

- **Output layer**: The final layer of the network, producing the network's predictions or outputs.

The MLP can be mathematically defined as:

$$a^{(0)} = x \in \mathbb{R}^{n_0} \quad \text{(input vector)}, \tag{2.5}$$

For each layer $\ell = 1, 2, \ldots, L$:

$$z^{(\ell)} = W^{(\ell)}\, a^{(\ell-1)} + b^{(\ell)}, \tag{2.6}$$

$$W^{(\ell)} \in \mathbb{R}^{n_\ell \times n_{\ell-1}}, \tag{2.7}$$

$$b^{(\ell)} \in \mathbb{R}^{n_\ell}, \tag{2.8}$$

$$a^{(\ell)} = \sigma^{(\ell)}\big(z^{(\ell)}\big), \tag{2.9}$$

$$\sigma^{(\ell)} : \mathbb{R}^{n_\ell} \to \mathbb{R}^{n_\ell} \text{ (activation function)}, \tag{2.10}$$

The final output is:

$$y = a^{(L)} = \sigma^{(L)}\big(W^{(L)} a^{(L-1)} + b^{(L)}\big). \tag{2.11}$$

It can be generalised using:

$$f(x) \;=\; \sigma^{(L)}\Big(W^{(L)}\Big(\sigma^{(L-1)}\big(W^{(L-1)}(\cdots \sigma^{(1)}(W^{(1)}x + b^{(1)})\cdots) + b^{(L-1)}\Big) + b^{(L)}\Big). \tag{2.12}$$

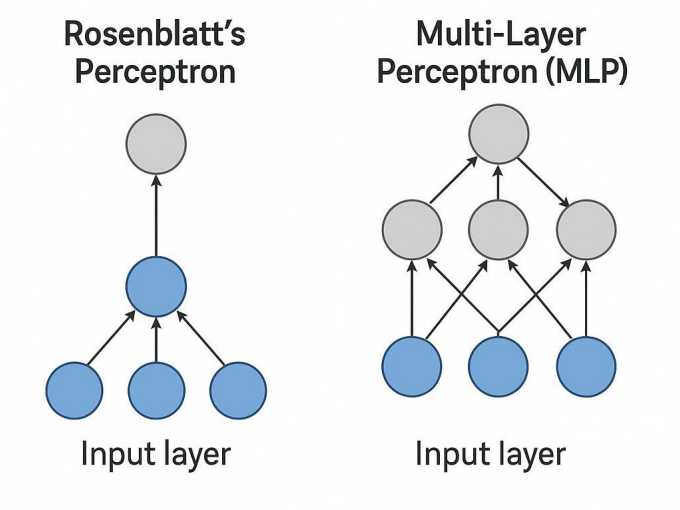


Figure 2.2: Comparison between Rosenblatt's Perceptron and the Multi-Layer Perceptron (MLP).

In which

$x \in \mathbb{R}^{n_0}$: input vector, of dimension n_0,

$a^{(0)} := x$: activation vector of the input layer,

$n_\ell \in \mathbb{N}$: number of neurons in layer ℓ,

$W^{(\ell)} \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$: weight matrix mapping layer $\ell - 1 \to \ell$,

$b^{(\ell)} \in \mathbb{R}^{n_\ell}$: bias vector for layer ℓ,

$z^{(\ell)} := W^{(\ell)}\, a^{(\ell-1)} + b^{(\ell)}$: pre-activation vector at layer ℓ,

$\sigma^{(\ell)} : \mathbb{R}^{n_\ell} \to \mathbb{R}^{n_\ell}$: activation function of layer ℓ,

$a^{(\ell)} := \sigma^{(\ell)}\big(z^{(\ell)}\big)$: activation (output) of layer ℓ,

$L \in \mathbb{N}$: total number of layers,

$y := a^{(L)} \in \mathbb{R}^{n_L}$: final output vector,

$f(x) := a^{(L)}$: the overall mapping implemented by the MLP.

A comparison between Rosenblatt's Perceptron and the Multi-Layer Perceptron (MLP) can be seen in Figure 2.2. The introduction of backpropagation in 1986 marked a revolution in neural network training by enabling the efficient optimization of network parameters through gradient descent. By iteratively adjusting the weights to minimize error, backpropagation enabled MLPs to learn complex mappings from inputs to outputs. This iterative process gradually refined the network's weights, allowing it to approximate underlying data distributions and significantly improve its predictive accuracy [120]. This also laid the foundation for deep neural networks (DNNs), which feature more intricate architectures with multiple interconnected layers of neurons that allow for the automatic extraction of complex features and hierarchical representations from raw data.

Further developments in training algorithms, computational resources, and data availability, particularly through the advent of graphical processing units (GPUs) and specialized hardware accelerators [27], have contributed to the widespread adoption and scalability of DNNs. This accessibility has enabled researchers to explore more complex neural network topologies, such as convolutional neural networks (CNNs) for image processing and recurrent neural networks (RNNs) for sequential data modeling.

2.2.1 Convolutional Neural Networks (CNNs)

Deep Neural Networks (DNNs) have become a prominent approach in machine learning due to their capability to extract layered representations directly from raw inputs. Among the various DNN architectures, Convolutional Neural Networks (CNNs) have been specifically designed to handle structured data formats, particularly visual data such as images. CNNs are fundamentally built on the convolution

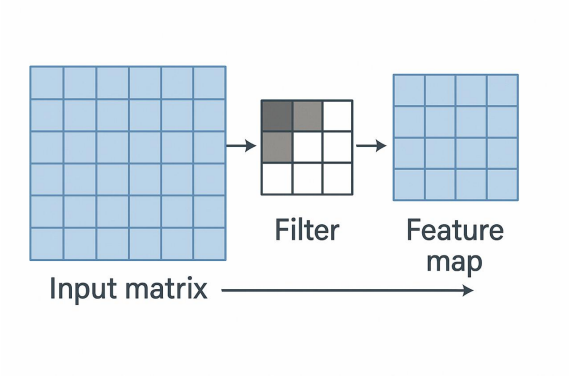


Figure 2.3: Illustration of the convolution process

operation, which is a mathematical mechanism for capturing local spatial dependencies within data.

The essential building blocks of CNNs are the convolutional layers. These layers produce feature maps by applying learnable filters, also referred to as kernels, to the input data. The convolution process entails sliding each filter across the input and computing the dot product between the filter's parameters and the local input values at each location. Through backpropagation, these filters are trained to identify salient features from the input, including edges, textures, and components of objects. The two-dimensional convolution process in CNNs can be mathematically defined as:

$$y[i,j] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[i+m, j+n] \cdot w[m,n] + b, \tag{2.13}$$

Where

- $y[i,j]$ denotes the resulting value in the output feature map at coordinates (i,j).
- $x[i+m, j+n]$ corresponds to the input value at position $(i+m, j+n)$.

- $w[m, n]$ indicates the weights of the convolutional filter.
- b is the bias term added to the result.
- M and N specify the height and width of the filter, respectively.

Figure 2.3 provides a visual representation of a convolutional operation. The resulting feature map y captures localized patterns in the input, where each element reflects the response to specific features learned by the filter.

In addition to convolutional layers, CNNs typically include other types of layers to enrich feature extraction and control data dimensionality. Pooling layers, for example, perform downsampling by aggregating data from local regions of the input, thereby reducing spatial resolution while retaining essential information [124].

Fully connected layers, usually positioned towards the end of the network, consolidate the information extracted by previous layers. These layers connect every neuron from the preceding layer, enabling the abstraction of high-level features that are essential for final classification or regression decisions. Between these layers, activation functions—most notably the Rectified Linear Unit (ReLU)—are introduced to inject non-linearity into the network, which enhances its representational capacity [95].

CNNs exploit the convolution operation to autonomously learn multi-level representations of structured inputs, proving especially effective in domains such as image classification, object localization, and semantic segmentation [23]. By combining convolutional, pooling, and fully connected layers, CNNs can systematically model spatial relationships in data, resulting in cutting-edge performance across numerous computer vision applications.

2.2.2 Other Neural Network Architectures

The deep learning field has seen the emergence of several novel neural network architectures, each designed to tackle specific challenges in a variety of domains. Notably, attention mechanisms have significantly enhanced neural networks in handling sequential or structured data. Generative Adversarial Networks (GANs) [49] revolutionized generative modeling by framing the problem as a game between a generator and a discriminator. GANs have been applied to diverse tasks such as image generation, style transfer, and data augmentation, facilitating the creation of highly realistic synthetic data for model training.

The Transformer architecture [141], which relies exclusively on attention mechanisms, has outperformed traditional recurrent models in tasks such as machine translation and natural language understanding. Additionally, multimodal foundation models [85] integrate information across multiple modalities, such as text, images, and audio, into a unified framework. These models are typically pre-trained on large multimodal datasets before being fine-tuned for specific tasks, making them highly versatile across domains such as multimedia content generation, autonomous vehicles, and healthcare.

Recurrent Neural Networks (RNNs) and their specialized variants, such as Long Short-Term Memory (LSTM) networks [55], have been pivotal in modeling sequential data. LSTMs, in particular, address the vanishing gradient problem inherent in standard RNNs by incorporating memory cells, gates for input, output, and forget, thus enabling RNNs to capture long-range dependencies in sequential data. These models have found extensive applications in fields such as speech recognition, language modeling, machine translation, and time-series prediction.

Moreover, a hybrid architecture known as the Recurrent Convolutional Neural

Network (RCNN) [107] combines the benefits of CNNs and RNNs. By incorporating recurrent connections within convolutional layers, RCNNs effectively capture both spatial and temporal dependencies in tasks like video frame analysis.

The proliferation of these advanced neural network architectures illustrates the ever-expanding potential of deep learning models across diverse applications, each pushing the boundaries of AI's capabilities and fostering innovation across various fields.

2.3 Transfer Learning

Transfer learning represents a paradigm shift in machine learning, moving beyond the traditional focus on isolated tasks to embrace the notion of leveraging prior knowledge to enhance learning in novel, yet related, scenarios [5]. In essence, it concerns the improvement of learning in a new target task through the transfer of knowledge acquired from one or more related source tasks that have already been learned [8]. This approach is inspired by the remarkable ability of human learners to recognise and apply relevant knowledge from past experiences when confronted with new challenges, demonstrating a natural propensity for knowledge transfer. Indeed, the more related a new task is to an individual's previous experience, the more readily they can achieve mastery. A complete survey on this topic is discussed in [138].

In stark contrast, conventional machine learning algorithms typically address each task in isolation, necessitating substantial amounts of task-specific data for effective learning. Transfer learning endeavours to overcome this limitation by developing methodologies that facilitate the transfer of knowledge learned in source

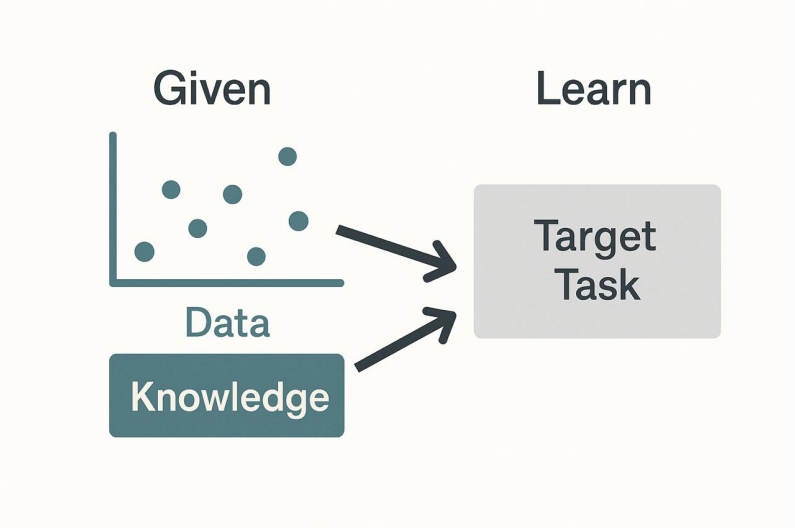


Figure 2.4: Representation of how Transfer Learning works

tasks to improve the learning efficacy and efficiency in a related target task (see Figure 2.4) [5, 10]. The advancement of such techniques holds the promise of making machine learning systems more aligned with the efficiency inherent in human learning processes.

2.3.1 Definition

The central goal of transfer learning is to enhance learning in a designated target task by exploiting knowledge gleaned from one or more source tasks [10]. As depicted in Figure 2.4, this involves supplementing the standard training data for the target task with knowledge derived from previously learned source tasks.

A critical distinction is made between transfer learning and multi-task learning. In multi-task learning, several tasks are learned concurrently, allowing for a bidirectional flow of information among them [15, 16]. Conversely, transfer learning, as defined herein, involves a unidirectional transfer of knowledge from source to target tasks. During the learning phase of the source task(s), the agent possesses no prior

knowledge of the impending target task. While it might be conceivable to address a multi-task learning problem using a transfer learning approach, the reverse is generally not feasible. This distinction is pertinent as real-world learning agents are more likely to encounter sequential transfer scenarios than truly simultaneous multi-task scenarios.

The primary objective of transfer learning is to achieve a discernible improvement in the learning process of the target task through the incorporation of source task knowledge. This improvement can manifest in several key aspects:

- **Initial Performance:** The level of performance achievable in the target task at the onset, utilising only the transferred knowledge, compared to the initial performance of an agent without any prior knowledge.
- **Learning Speed:** The rate at which the target task is learned given the transferred knowledge, in comparison to the time required to learn it from scratch.
- **Final Performance:** The ultimate level of performance that can be attained in the target task with the benefit of transfer, compared to the final performance without transfer.

However, it is crucial to acknowledge the phenomenon of negative transfer, wherein the application of knowledge from a source task inadvertently degrades performance in the target task [34]. A significant challenge in the development of effective transfer learning methods lies in fostering positive transfer between appropriately related tasks while concurrently mitigating the risk of negative transfer between tasks with less pertinent relationships.

Furthermore, the application of knowledge across tasks often necessitates a process of task mapping, whereby the characteristics of one task are mapped onto those of another to establish correspondences. While in many existing approaches this mapping is provided by a human expert, there is ongoing research dedicated to developing methods for automatic task mapping [139].

2.3.2 Advantages of Transfer Learning

Transfer learning offers several key advantages over traditional isolated machine learning approaches:

- **Improved Initial Performance:** By leveraging knowledge from related source tasks, a learning agent can often achieve a significantly higher starting performance on a new target task compared to learning from scratch [10].
- **Accelerated Learning:** The transferred knowledge can guide the learning process in the target task, leading to a faster rate of improvement and a reduction in the time required to achieve satisfactory performance.
- **Enhanced Final Performance:** In some cases, transfer learning can enable an agent to reach a higher level of ultimate performance on the target task than would be possible without the transferred knowledge.
- **Addressing Data Scarcity:** Transfer learning can be particularly beneficial when the target task has limited training data or labeled examples, as knowledge from data-rich source tasks can compensate for this deficiency [101, 106].

2.3.3 Disadvantages and Challenges of Transfer Learning

Despite its numerous benefits, transfer learning also presents several challenges and potential drawbacks:

- **Negative Transfer:** A significant risk is negative transfer, where knowledge from a source task detrimentally affects learning or performance in the target task, often due to a lack of sufficient relatedness or an inappropriate transfer method [34, 127].
- **Automatic Task Mapping:** Determining the correspondences between source and target tasks, particularly when their representations differ, remains a significant challenge. Many existing methods rely on manual specification of these mappings [34, 139].
- **Transfer Between Diverse Tasks:** Enabling effective transfer between tasks that are substantially different in their nature or domain is an ongoing area of research. Identifying and transferring abstract, high-level knowledge that is applicable across diverse domains is particularly challenging.
- **Transfer in Complex Domains:** Applying transfer learning successfully in complex, real-world domains, particularly in reinforcement learning, can be significantly more difficult than in simpler testbeds.

Transfer learning has emerged as a crucial and rapidly expanding subfield within machine learning, driven by both its alignment with principles of human learning and its practical potential to enhance the efficiency and effectiveness of machine learning systems. As computational resources continue to grow and machine learning is applied to increasingly intricate problems, the ability to effectively transfer

knowledge across tasks will only become more critical. Future research will likely focus on addressing the existing challenges, particularly the avoidance of negative transfer and the automation of task mapping, as well as exploring new frontiers in enabling transfer between more diverse and complex tasks.

2.4 Federated Learning

Federated Learning (FL) is an emerging distributed machine learning paradigm [87] designed to reconcile the growing need to train increasingly complex models on vast datasets with the imperative for individuals and organizations to protect their confidential data. Alternatively, FL also represents a further evolution in the distribution of the training process across multiple nodes.

2.4.1 Definition

An early comprehensive definition of FL is provided in a key paper by Kairouz and McMahan [63], which also addresses the challenges and unresolved issues in this domain:

"Federated learning is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each client's raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective."

In practical implementations, FL systems are typically more specialized than this broad definition implies. Generally, the models trained on the client side are deep neural networks (DNNs), and the exchanged updates consist of either the

weights or gradients of these DNNs. Given that these updates are represented as tensors, they are readily serialized, transmitted across the network, and aggregated using mathematical operations. Often, it is assumed that every client trains an identical DNN architecture and that each client's data follows an equivalent schema. Although these assumptions are quite strong in real-world scenarios, many studies have sought to relax these constraints [74, 110].

FL operates through an iterative cycle: clients compute local updates and transmit them to the central server, which aggregates these contributions and redistributes the updated model to the clients; each such iteration is referred to as a federated round or simply a round. The standard method for combining updates is federated averaging (FedAvg) [87], in which the contributions of different clients are averaged. This method, while straightforward, can yield satisfactory learning outcomes provided that the data distribution across clients is ideally independent and identically distributed (IID) [76]. An immediate enhancement of FedAvg involves weighting the updates based on the quantity of data used in their computation, thereby giving more influence to updates derived from larger datasets. In scenarios where data non-IID characteristics arise not only from differing quantities but also from distinct internal distributions, alternative techniques become necessary.

Since FedAvg is inherently synchronous, its computational performance can be limited. Consequently, asynchronous methods have been proposed, such as ASO-Fed [26], which integrates continuous local learning with asynchronous server-side aggregation, and FedBuff [98], which uses buffering to aggregate update batches asynchronously. Communication costs also significantly affect FL's overall efficiency. To address this, methods such as quantisation [128], compression [123], and distillation [150] have been developed to reduce the size of updates, thereby accelerating

data transfer and reducing the overall FL overhead.

Another challenge introduced by FL pertains to managing stateful objects. Standard optimizers in traditional deep learning maintain internal states that are coupled with the current weight configuration of the model, potentially causing performance degradation when local weights are replaced by those of the global model. Federated optimization techniques, such as FedSplit [105], FedProx [75], and federated adaptations of adaptive algorithms like ADAM [113], can be employed to mitigate these issues. Simultaneously, modern DNNs often include batch normalization layers, which are also stateful. When models are aggregated, neglecting these layers can lead to a decline in performance [18]. Two straightforward solutions involve either averaging the parameters of the batch normalization layers alongside the model weights or substituting them with stateless normalization layers, such as layer normalization [41]. An alternative strategy is to use a modified batch normalization approach, such as FixBN [160], that can accommodate shifts in weights.

Beyond these technical challenges, FL must also address privacy concerns, as it is frequently employed in contexts where data confidentiality is paramount. A variety of techniques can be integrated to enforce privacy guarantees. One widely adopted method is homomorphic encryption (HE) [156], which enables data to be transformed into an encrypted format that still supports specific operations, ensuring that computations performed in the encrypted domain yield results equivalent to those performed on the plaintext data [157]. Another approach is secure multi-party computation (SMC) [32], which allows multiple parties to jointly compute a function based on their local data without revealing the data to each other; this method ensures both result accuracy and resilience against adversarial attacks and can be

effectively combined with FL [77]. Differential privacy (DP) [42] is yet another measure, involving the introduction of controlled noise to the system so as to obfuscate the presence of any individual data sample, all while minimally impacting overall performance. This technique is particularly advantageous in FL settings [145], as noise can be incorporated at various stages (e.g., during local batch updates, prior to transmission to the server, or on the global model parameters) to mitigate membership inference attacks. Additional techniques can further enhance the robustness and privacy of FL systems: trusted execution environments (TEEs) [121] can facilitate secure local training or aggregation even on untrusted infrastructures, and blockchain technology [99] can provide secure client authentication and maintain an immutable log of update histories, sometimes incorporating smart contracts for the aggregation process [109].

2.4.2 Classification of Federated Learning

FL systems can be broadly categorized along two orthogonal dimensions: deployment scale and data partitioning. Regarding deployment scale, two scenarios are typically identified: cross-silo and cross-device. Cross-silo FL [56] generally involves a small federation (fewer than 100 clients) of powerful, server-grade machines or data centers. This setting is common among institutions such as hospitals, insurance companies, universities, and banks, where the hardware is capable of managing intensive deep learning workloads, the clients are reliably online, the network connectivity is fast and stable, and the process is managed by experts. This scenario is particularly suited for training large deep learning models on sensitive data, with a strong emphasis on security and the development of a unified model that accommodates diverse client data distributions without compromising privacy.

In contrast, cross-device FL [64] targets a large number of clients (more than 100), typically involving smartphones or other edge devices. These devices are generally less powerful, operate under energy constraints, and often rely on mobile networks. In this context, only smaller deep learning models are trained locally, and the aggregation process typically involves sampling only a subset of available local models due to the inherent bottlenecks of a centralized master-server framework. This dissertation focuses solely on the cross-silo FL scenario.

With respect to data partitioning, FL can be classified into three categories based on how the feature and sample spaces are distributed among clients: horizontal, vertical, and hybrid FL [154]. In horizontal FL—the most prevalent type—local datasets share an identical feature space (i.e., the data adhere to the same schema), though the individual data samples do not overlap across clients. In vertical FL [152], the clients have overlapping sample spaces but different subsets of features; the data would be partitioned along vertical columns in a tabular representation. Hybrid FL [158] describes situations in which both the feature and sample spaces do not completely align between clients but exhibit some degree of overlap. Each of these FL scenarios necessitates distinct deep learning models, training algorithms, and aggregation strategies to effectively address their unique feature and sample space characteristics.

2.5 Computer Vision

Computer Vision (CV), positioned at the confluence of artificial intelligence and computer science, constitutes a multidisciplinary field focused on enabling machines to analyze and derive understanding from visual inputs in a manner analogous to

human perception. Its scope is broad, encompassing tasks such as image classification, object detection, segmentation, and the interpretation of visual scenes. The overarching aim of computer vision is to formulate algorithms and systems capable of autonomously extracting significant insights from visual media—be it images or videos—thus emulating the perceptual mechanisms of human observers.

In alignment with the general ambitions of artificial intelligence, computer vision aspires not only to replicate the optical perception capabilities of humans but also to simulate the cognitive processes that underlie human interpretation of visual stimuli. This involves providing machines with the capacity to identify, interpret, and contextualize visual representations of objects and people.

Among the primary operations in CV is image classification, where the objective is to assign predefined labels to images based on their visual features. Beyond classifying whole images, object detection plays a vital role by identifying the presence and spatial location of multiple objects within an image. A significant advancement in this area came with the introduction of R-CNN [46], which employed a region-based strategy to improve detection precision. This approach was further refined with Faster R-CNN [114], which greatly increased both speed and accuracy, thereby facilitating its use in time-sensitive applications.

Another key task is image segmentation, where each pixel is assigned to a class category, allowing for a fine-grained analysis of image content. Fully Convolutional Networks (FCNs) [82] made notable progress in this regard by enabling pixel-wise semantic labeling. These techniques have proven especially useful in domains such as autonomous navigation and medical diagnostics.

The applicability of computer vision continues to expand across numerous domains, such as healthcare. CV technologies assist in the analysis of medical images to support diagnosis and treatment planning. For instance, the U-Net architecture [117] has shown remarkable effectiveness in segmenting biomedical images, thereby enhancing the accuracy of lesion and tumor identification. Similarly, autonomous vehicles depend extensively on CV systems for detecting obstacles, recognizing lanes, and navigating through complex environments. The integration of visual input with LiDAR data [57] has improved object recognition capabilities in self-driving systems.

In the security and surveillance sector, techniques such as facial recognition and object tracking are powered by computer vision algorithms. DeepFace [134] represented a major breakthrough in facial recognition accuracy, enabling deployment in real-world applications. Object tracking, which involves monitoring entities as they move across video frames, typically combines detection with temporal association strategies. These range from basic learning methods to sophisticated deep learning architectures.

Furthermore, CV technologies have been harnessed for the synthesis of realistic visual content, known as deepfakes. These synthetic media outputs can serve various purposes—ranging from benign (e.g., entertainment and parody) to malicious (e.g., misinformation and reputational harm). While some applications enhance user experience, the ethical ramifications—particularly in relation to deception and electoral manipulation—remain a subject of ongoing debate [89].

The evolution of computer vision has been extraordinary, driven largely by breakthroughs in deep learning and increased computational capacity. Nevertheless, the field still faces significant hurdles. Issues such as sensitivity to lighting variations,

occlusions, and perspective changes remain challenging. Additionally, biased training datasets may yield discriminatory outcomes [22], underscoring the importance of ethical scrutiny in the design and deployment of vision systems.

2.6 Frameworks and Tools for Machine Learning

During my Ph.D. journey, I employed a variety of tools and frameworks to develop, evaluate, and refine artificial intelligence models. Among these, several key technologies proved especially effective in supporting the implementation of deep learning architectures and computer vision solutions. A broad spectrum of tools and libraries is available to facilitate the development of AI algorithms, each offering unique capabilities tailored to different aspects of machine learning and deep learning workflows. These technologies form the backbone of modern machine and deep learning workflows, equipping researchers, engineers, and developers with the functionality required to tackle complex challenges and foster innovative solutions in AI.

- **Python** has emerged as the dominant programming language for machine learning and deep learning applications, owing to its simplicity, adaptability, and rich ecosystem of specialized libraries. Its clear syntax facilitates rapid prototyping and experimentation, making it an ideal choice for academic and industrial AI projects. [1]

- **NumPy** is a foundational scientific computing package for Python. It supports efficient manipulation of multi-dimensional arrays and matrices and provides a broad set of mathematical functions tailored to numerical computation.

[1] https://www.python.org/ - last accessed: September 2024

It serves as the backbone for numerous other Python-based ML libraries. [2]

- **Pandas** offers powerful data structures such as DataFrames and Series, streamlining the handling of structured data. It is widely employed in data preprocessing, cleaning, and transformation tasks—key steps in any machine learning pipeline. [3]

- **Matplotlib** is a highly versatile plotting library in Python that enables the generation of diverse visualizations, including line graphs, scatter plots, histograms, and bar charts. Its extensive customization capabilities make it indispensable for result analysis and data visualization. [4]

- **OpenCV** (Open Source Computer Vision Library) is a comprehensive toolkit for image and video processing, offering features for object detection, image transformation, feature extraction, and camera calibration. It is widely used in both academic research and industrial CV applications. [5]

- **scikit-learn** (sklearn) is a Python library offering an extensive suite of machine learning algorithms for tasks such as classification, regression, clustering, and dimensionality reduction. Renowned for its user-friendly API and ease of integration with other Python tools, it is a standard choice for many ML practitioners. [6]

- **PyTorch**, developed by Facebook AI Research, is an open-source deep learning framework known for its dynamic computation graph and flexibility. It

[2] https://numpy.org/ - last accessed: September 2024
[3] https://pandas.pydata.org/ - last accessed: September 2024
[4] https://matplotlib.org/ - last accessed: September 2024
[5] https://opencv.org/ - last accessed: September 2024
[6] https://scikit-learn.org/ - last accessed: September 2024

supports both imperative and symbolic programming styles, and provides a comprehensive environment for building, training, and deploying neural networks. [7]

- **Google Colaboratory** is a cloud-based platform that allows users to write and execute Python code in a Jupyter notebook environment. It provides free access to computational resources, including GPUs, making it ideal for machine learning and data analysis tasks. Colab facilitates collaborative work, enabling multiple users to share and edit notebooks in real time. [8]

[7] https://pytorch.org/ - last accessed: September 2024
[8] https://colab.research.google.com/ - last accessed: September 2024

Chapter 3

Introduction to Ceilometers

Ceilometers are critical instruments in atmospheric science, primarily used for measuring the height of cloud bases and assessing cloud cover. A ceilometer is a measuring device mostly used in meteorology that can detect the height of a cloud base, by emitting a modulated light beam directed to the sky. Clouds height are then computed by measuring the time-of-flight of the beam return back from the sky to the send. Concentrations of aerosols, such as water vapour or pollutants in the atmosphere, can also be determined from the backscatter effect of the emitted laser beam. Different kinds of ceilometers exist, depending on the technology used to take measurements. In contrast to numerous other remote sensing instruments like satellites or radar, the ceilometer presents unique features that render it particularly valuable in specific environmental contexts. Specifically, ceilometers provide direct vertical measurements of the height of clouds above the ground, which is useful for monitoring changes in the atmosphere at a local level. Furthermore, these tools provide measurements at very short time intervals, allowing changes in clouds over time to be monitored in greater temporal resolution with respect to many satellite platforms or other remote sensing approaches. In addition, ceilometers are particularly useful for measuring the height of low clouds, which can be difficult to detect

through other types of measurement devices. This chapter will explore the various types of ceilometers, the specific atmospheric variables they measure, and how these devices have been utilized in this study. Furthermore, the chapter will discuss examples from the state-of-the-art research where different types of ceilometers have been employed, providing a comprehensive understanding of their practical applications in the field.

3.1 Types of Ceilometers

Ceilometers come in various forms, each leveraging different technologies to fulfill their role in atmospheric measurements. Understanding the differences between these types is crucial for selecting the appropriate instrument for a given meteorological study. This section will cover LIDAR-based ceilometers, optical-based ceilometers, acoustic ceilometers, radar ceilometers, and hybrid systems. Each subsection will explain how each tool works, including a few examples from recent scientific literature to illustrate their practical applications.

3.1.1 LIDAR-Based Ceilometers

LIDAR (Light Detection and Ranging) ceilometers are among the most advanced in atmospheric science, providing highly accurate cloud base measurements. They function by emitting laser pulses into the atmosphere and measuring the time it takes for the light to reflect back after hitting the cloud base. This allows for precise measurements of cloud height, even in challenging environmental conditions.

In [93], lidar-based ceilometers are primarily used to determine cloud base height by measuring the attenuated backscatter of laser light from atmospheric particles.

In addition to this, they can be used to monitor air quality by correlating near-range backscatter with particulate matter (PM10) concentrations in dry weather conditions. Ceilometers are also employed to estimate the convective mixing height in the atmosphere, and their advanced optics allow them to detect fine structures in the boundary layer, aiding in detailed atmospheric profiling.

In [94], lidar-based ceilometers are used to measure cloud base height (CBH) variations over Pune, India, from 2017 to 2023. The ceilometer, with a range of 0–7500 meters, reveals seasonal and diurnal trends in CBH. During the monsoon season, the average CBH is the lowest with minimal daily variation, while the pre-monsoon, post-monsoon, and winter seasons show a gradual rise in CBH in the morning, peaking in the afternoon, and decreasing in the evening. The study also highlights the dominance of low cloud bases (CBH $< 2000m$) throughout the year, with high cloud bases (6000–7500 m) being the least frequent and showing a semi-diurnal pattern during winter. This information is valuable for weather prediction, climate modeling, and environmental monitoring.

Lidar-based ceilometers in the Alicenet network [9], established in Italy in 2015, are used for vertically-resolved monitoring of aerosol particles, which is crucial for studying aerosol-climate interactions and their impact on air quality and human health. Alicenet consists of single-channel and dual-channel, polarization-sensitive lidar systems operating in diverse environments (urban, coastal, mountainous, and volcanic) across Italy. These systems continuously monitor the vertical distribution of aerosols and contribute to the EUMETSAT E-PROFILE program. The data processing chain developed for Alicenet converts raw lidar data into valuable information on aerosol properties, such as attenuated backscatter, aerosol mass, and

vertical stratification. The network provides near real-time and long-term monitoring, supporting sectors such as air quality, solar energy, and aviation safety. Some ceilometers from Alicenet network were used in this research.

In [91], lidar-based ceilometers are used to study the transition zone (TZ) between clouds and cloud-free air by analyzing backscatter profiles. This study compares two cloud detection algorithms—one from the ceilometer manufacturer (Vaisala) and another from Cloudnetpy (ACTRIS Cloudnet)—to explore how aerosols and clouds transition in the atmosphere. The results show that near cloud boundaries, particles produce higher backscatter values, indicating a gradual shift between cloudy and clear conditions. The analysis reveals that TZ conditions occur as frequently as clouds and are most prevalent below 800 meters at night, varying across seasons. This highlights the importance of treating the transition between clouds and aerosols as a continuum in atmospheric studies.

3.1.2 Optical-Based Ceilometers

Optical-based ceilometers typically employ triangulation or time-of-flight methods, where a light source (often a laser or infrared beam) is projected upward, and the angle or time it takes for the reflected light to return is used to estimate cloud base heights. Although they have been largely supplanted by more advanced LIDAR systems, optical ceilometers remain relevant due to their simplicity, cost-effectiveness, and ability to function in a range of weather conditions. Despite being less accurate than LIDAR in providing detailed atmospheric profiling, optical ceilometers are still widely used in aviation, weather stations, and for basic cloud detection in areas where more advanced technologies are not readily available. Their reliability, particularly in scenarios where continuous monitoring of cloud base heights is required

without the need for vertical aerosol profiling, makes them a practical choice for many meteorological applications. Moreover, improvements in optical technologies have allowed for better accuracy and performance, making them an efficient tool for specific use cases where full lidar-based systems may not be necessary, such as in small airports or remote locations. The longevity and ongoing use of these systems highlight their enduring importance in cloud detection and basic atmospheric observation.

3.1.3 Acoustic Ceilometers

Acoustic ceilometers, also known as SODAR (Sonic Detection and Ranging) systems, represent another method for measuring cloud base heights using sound waves. These systems work by emitting acoustic pulses into the atmosphere and measuring the time it takes for the sound waves to reflect off cloud bases or temperature inversions and return to the instrument. This approach allows SODAR systems to function effectively in conditions where optical systems may struggle, such as during heavy fog, precipitation, or in low visibility scenarios. One of the key advantages of acoustic ceilometers is their ability to operate in adverse weather conditions where light-based systems may fail, offering reliable measurements even when clouds or atmospheric particulates obscure the sky. Additionally, SODAR systems are useful for assessing atmospheric turbulence, wind profiles, and temperature layers, making them valuable for applications beyond simple cloud base detection. However, acoustic ceilometers have their limitations. They are particularly sensitive to ambient noise, such as traffic or industrial sounds, which can interfere with the accuracy of the measurements. Furthermore, while effective for determining cloud base height in challenging conditions, SODAR systems generally lack the vertical resolution and

detailed profiling capabilities of lidar systems, making them less suitable for more complex atmospheric studies. Despite these challenges, SODAR remains a valuable tool for meteorological and environmental monitoring, particularly in specialized scenarios where traditional optical or lidar-based systems may not perform optimally.

3.1.4 Radar Ceilometers

Radar ceilometers, which use radio waves to detect cloud base heights, are especially effective in severe weather conditions, such as heavy rain, snow, or dust storms, where optical or acoustic systems may encounter significant limitations. By emitting radio waves and measuring the time it takes for the signal to reflect off clouds or other atmospheric features, these instruments can provide reliable cloud height measurements even in low-visibility or extreme weather scenarios. One of the primary advantages of radar ceilometers is their robustness and ability to operate in challenging environments. Unlike optical systems, which can struggle in dense cloud cover, fog, or precipitation, radar systems are not affected by visibility issues, making them particularly valuable for continuous monitoring during storms or in areas prone to extreme weather. Their resistance to interference from atmospheric particles such as rain, dust, or snow allows them to maintain accurate measurements where other systems might fail. Radar ceilometers are commonly used in critical applications such as aviation, where real-time and reliable cloud base data is essential for flight safety, and in remote or industrial environments where weather conditions can rapidly change. While these systems are typically more expensive and complex than optical or acoustic ceilometers, their high reliability, especially in adverse conditions, makes them a preferred choice for operations requiring uninterrupted cloud

detection and weather monitoring.

3.2 Variables Measured by Ceilometers

Ceilometers, depending on their technology and design, can measure a range of atmospheric variables. These measurements are crucial for meteorological studies, aviation safety, and climate research. This section details the primary variables measured by ceilometers and their importance in the broader context of atmospheric science. Table 3.1 lists all the variables that the ceilometer collects in its measurements.

Variable	**Type**	**Unit**	**Description**
time	double	seconds	End point of the measurement (UTC)
range	float	m	Measurement distance of the device (independent of direction and height of installation location)
range_hr	float	m	Measurement distance of the device for high resolution
layer	int	-	Layer index
latitude	float	degree	Latitude of the installation location
longitude	float	degree	Longitude of the installation location
azimuth	float	degree	Azimuth angle of the device (direction of the laser indicator)
zenith	float	degree	Zenith angle of the device (direction of the laser indicator)
altitude	float	m	Height of installation of the device above sea level
wavelength	float	nm	Wavelength of the laser in nm
average_time	int	ms	Average time per recording
range_gate	float	m	Spatial resolution of the measurement
range_gate_hr	float	m	Spatial resolution of the high-resolution measurement
life_time	int	h	Propagation time of the laser

error_ext	int	-	32-bit status code
state_laser	byte	percent	Laser quality index
state_detector	byte	percent	Signal detector quality
state_optics	byte	percent	Optical quality index
temp_int	short	K	Internal temperature of the housing
temp_ext	short	K	External temperature of the housing
temp_det	short	K	Temperature of the detector
temp_lom	short	K	Temperature of the measurement unit
laser_pulses	int	-	Number of laser pulses emitted during a measurement (lp)
p_calc	short	counts	Calibration pulse (normalization of the measurement unit over time)
scaling	float	-	Scaling factor (normalization of measurement units relative to each other) (cs)
base	float	counts	Height of the baseline of the raw signal (primarily influenced by daylight) (b)
stddev	float	counts	Standard deviation of the raw signal
beta_raw	float	-	Normalized backscatter signal, corrected for range ((P_raw / lp) - b) / (cs * o(r) * p_calc) * r * r, with P_raw = sum(P_raw_hr) * range_gate_hr / range_gate)
beta_raw_hr	float	-	High-resolution backscatter signal, normalized, corrected for range ((P_raw_hr / lp) - b) / (cs * o(r) * p_calc) * r * r)
pbl	short	m	Aerosol layers
pbs	byte	-	Quality index for aerosol layers (1: good, 9: bad)
tcc	byte	-	Degree of coverage (overall)
bcc	byte	-	Degree of coverage of the lower cloud layer
sci	byte	-	Sky Condition Index (0: no precipitation, 1: rain, 2: fog, 3: snow, 4: precipitation or particles on the window pane)

vor	short	m	Vertical visibility
voe	short	m	Opacity of the detected vertical visibility
mxd	short	m	Maximum detection distance
cbh	short	m	Cloud base height
cbe	short	m	Calculated cloud base blur
cdp	short	m	Cloud penetration depth
cde	short	m	Calculated cloud penetration depth blur
cho	short	m	Height offset (calculated in cbh, mxd, vor, and pbl; corresponds to altitude when usealtitude=1, otherwise 0)

Table 3.1: Complete list with all variables detected by the ceilometer.

3.2.1 Cloud Base Height

The primary measurement provided by all types of ceilometers is the height of the cloud base, which plays a crucial role in several key fields, including aviation safety, weather forecasting, and climate research. In aviation, accurate cloud base height data is essential for determining safe flying conditions, especially for takeoffs and landings, as low cloud ceilings can reduce visibility and create hazards for pilots. For weather forecasting, cloud base height is a critical variable in predicting upcoming weather patterns, such as precipitation or storm development, helping meteorologists assess atmospheric stability and cloud dynamics.

3.2.2 Cloud Cover

Ceilometers also provide measurements of cloud cover extent, typically expressed as a fraction of the sky obscured by clouds (in oktas). This information is crucial for understanding solar radiation levels, as cloud cover significantly influences the amount

of sunlight that reaches the Earth's surface. By quantifying cloud cover, ceilometers help meteorologists assess how much solar radiation is available for various applications, including agriculture, climate studies, and renewable energy production. For solar energy generation, accurate cloud cover data is particularly important, as it directly affects the efficiency of solar panels. High cloud cover can reduce the amount of solar energy captured, impacting energy production forecasts and grid management. Additionally, understanding cloud cover patterns aids in predicting temperature fluctuations and weather phenomena, allowing for more accurate weather forecasts and improved climate modeling. Moreover, cloud cover measurements are vital for aviation and transportation sectors, where they help determine visibility conditions and potential weather hazards.

3.2.3 Cloud Thickness

Ceilometers have also the capability to measure cloud thickness by analyzing the time it takes for the emitted waves—be they light, sound, or radio—to traverse the cloud layer and return to the instrument. This measurement is vital for understanding the insulating effects of clouds, as thicker clouds can trap more heat in the atmosphere and influence temperature variations at the Earth's surface. Cloud thickness plays a significant role in various atmospheric processes, including precipitation formation and energy exchange between the surface and the atmosphere. By providing data on cloud thickness, ceilometers contribute to improved climate models that assess how clouds interact with solar radiation and contribute to the greenhouse effect. Additionally, this information is crucial for meteorological studies focused on understanding cloud dynamics, such as the processes that lead to

the development of different cloud types and their subsequent impact on weather patterns.

3.2.4 Backscatter

Backscatter refers to the portion of a signal that is reflected back towards its source after interacting with particles or surfaces in the atmosphere. In the context of ceilometers and other remote sensing technologies, backscatter is a crucial measurement used to characterize various atmospheric phenomena, including clouds, aerosols, and other particulate matter. When a ceilometer emits light, sound, or radio waves into the atmosphere, these waves encounter particles such as water droplets, ice crystals, or aerosols. Some of the emitted energy is scattered back towards the ceilometer, providing valuable information about the presence, concentration, and distribution of these particles. The intensity of the backscattered signal can indicate the density of the particles, helping to determine cloud base height, cloud thickness, and the vertical distribution of aerosols. Analyzing backscatter profiles is essential for understanding cloud dynamics, identifying different cloud types, and assessing the impact of aerosols on weather and climate.

3.3 Ceilometer Used in This Research

This section details the specific types of ceilometers and associated instrumentation used in this research.

Thanks to the collaboration with the company EHT S.C.p.A., this research employed a Lufft CHM 15k ceilometer that leverages *Light Detection and Ranging* (LiDAR) technology, as depicted in Figure 3.1. Short light pulses generated by a

Figure 3.1: EHT's Lufft CHM 15k lidar-based ceilometer.

solid-state laser microchip are emitted into the atmosphere, where are scattered by aerosols, droplets, and other air molecules. The portion of the light is reflected back is referred to as the backscatter (see Section 3.2.4), which is the information the device processes. The time-of-flight of the laser pulses is measured and used to calculate the distance of the scattering event. The height profile of the reflected signal is analyzed to calculate the backscatter intensity β-raw, from which the attenuated backscatter coefficient β-att is calculated with a valid calibration constant. Data acquired from the ceilometer are properly converted into time-height plots of backscatter coefficients (typically referred to as backscatter profiles [35]), where the x-axis represents the time, and the y-axis represents the altitude [68]. Such generated data are analysed with state-of-the-art CNNs, as well as a transformer-based

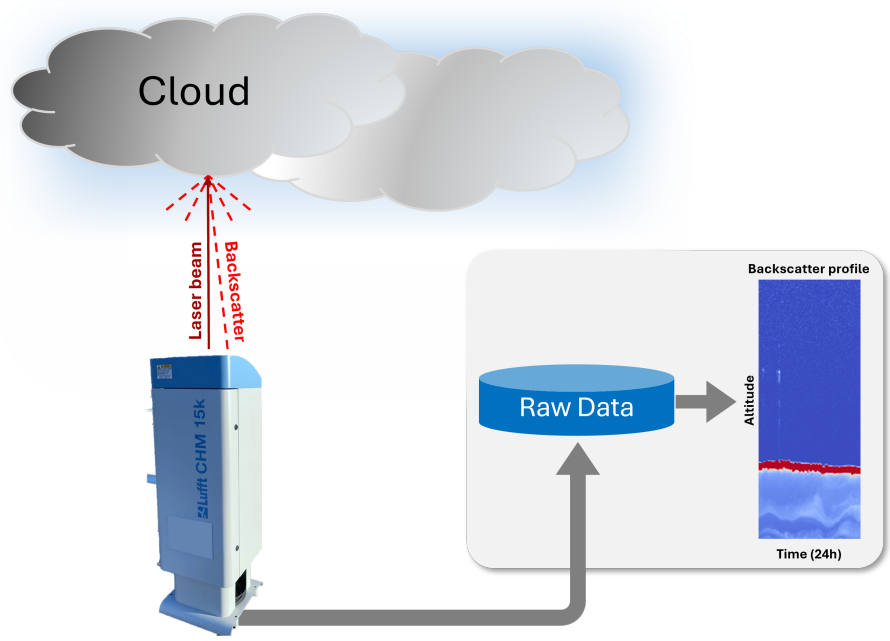


Figure 3.2: Visual representation of the data collection process. Backscatter raw data are utilized to generate a time-height plotting of backscatter coefficients (profile).

network, to detect the presence or absence of clouds. Other approaches in the literature analysing lidar imagery using deep networks have been proposed [33]. All the process is depicted in Figure 3.2.

The casing of this ceilometer consists of two layers of stainless aluminium. The outer casing has the task of mitigating solar radiation effects, wind, rain, and snow on the inner casing, containing the measuring unit. The lid protects the enclosure from dirt and other precipitation. A window is present on the lid to allow the laser beam to leave and enter back to the device. The partition wall in the lid separates the emission area from the sensible reception area. An air baffle inside the lid directs the airflow from both fans directly onto the glass panel of the inner case.

From this data, various useful parameters, *e.g.* the height of clouds and aerosol layers, are calculated. The detection system is based on a photon counting process. The narrow laser bandwidth allows an optical filter of approximately 1nm to be

placed in front of the detector to suppress background noise. The signal averaging allows to obtain a specific signal-to-noise ratio, which is crucial for lidar-based measurements that generate aerosol profiles. Compared to analogue measurement procedures, these processes are characterized by a very high sensitivity and detection accuracy.

This chapter provided a comprehensive overview of ceilometers, discussing their various types, the atmospheric variables they measure, and their specific application in this research. By understanding the capabilities and limitations of different types of ceilometers, this research gain valuable insights into their role in advancing atmospheric science and improving the accuracy of AI models designed for atmospheric analysis. The examples from the state-of-the-art research highlighted the practical applications of these instruments, demonstrating their importance in a wide range of meteorological studies. The knowledge gained from this chapter sets the foundation for the subsequent chapters, where the development and application of AI models for atmospheric analysis will be explored in greater detail.

Part II

Particles Detection

Chapter 4

Cloud Detection

This chapter delves into the core research focus of this thesis: Cloud Detection using ceilometer data. This study aims to enhance cloud monitoring techniques by leveraging the unique capabilities of ceilometers, which are widely used in atmospheric research for measuring cloud base height and aerosol profiles. In this context, the chapter will begin by discussing the methodology for collecting the dataset, detailing how ceilometer data was gathered over a specified period and location. The next step in the process is transforming this raw ceilometer data into images suitable for analysis. Since ceilometer outputs are typically in the form of time-series backscatter profiles, this section will describe the process of converting these profiles into image-like representations that can be used for cloud detection algorithms. The following section will explain the method chosen for labeling the data, which is a crucial step in a supervised learning task. Once the dataset is prepared, the chapter will move on to present the results obtained using state-of-the-art architectures in this field, including Convolutional Neural Networks (CNNs) and other deep learning architectures, in accurately identifying cloud patterns. A thorough comparison of these architectures will be provided, showcasing their strengths and limitations based on metrics such as accuracy, precision, recall, and computational efficiency. Insights

into the tuning of hyperparameters, model training, and validation processes will also be discussed.

Finally, the chapter will conclude by outlining the main insights for future work. This section will explore the potential avenues for improving cloud detection using ceilometer data. Additionally, the possibilities for extending this research to include real-time cloud monitoring or integration with broader atmospheric monitoring systems will be discussed, laying the groundwork for future advancements in the field of atmospheric science.

4.1 Introduction

Monitoring and understanding cloud formations and their dynamics are critical tasks in meteorology, as clouds play a pivotal role in weather prediction, climate modeling, and atmospheric studies. Clouds influence the Earth's energy balance, modulate temperatures, and impact precipitation patterns. Traditional methods for cloud observation often rely on satellite imagery and ground-based radar systems. This work utilizes data from a lidar-based ceilometer, an advanced remote sensing instrument capable of measuring cloud base heights and detecting atmospheric aerosols through backscatter analysis of a modulated light beam emitted into the sky.

Ceilometers offer distinct advantages over other remote sensing devices, providing continuous, high-resolution vertical measurements of cloud and aerosol distribution. This capability makes them particularly valuable for studying rapid changes in atmospheric conditions and for detecting phenomena that are challenging to observe through satellites, such as low-altitude clouds or localized aerosol concentrations.

Our study focuses on leveraging these unique capabilities to enhance cloud detection using state-of-the-art deep neural networks.

To better contextualize the role of ceilometers in cloud detection, this work briefly discusses alternative LiDAR-based methods and their characteristics. Our approach employs lidar-based ceilometers for continuous measurement, real-time monitoring of the lower atmosphere, specifically focusing on earth-to-satellite observations. Unlike high-sensitivity lidar systems [122], which integrate SNSPD technology to detect faint backscatter signals from high-altitude clouds with minimal noise, ceilometers are optimized for tracking cloud base heights and aerosol layers within the Planetary Boundary Layer (PBL). While hybrid radar-lidar techniques [163] combine millimeter-wave radar and multi-wavelength lidar for improved penetration and resolution, they require co-located instrumentation, limiting their deployment flexibility. Similarly, Airborne Laser Scanning (ALS) [153] provides high-resolution 3D cloud morphology but is constrained by high operational costs and limited temporal coverage.

In contrast, ceilometers operate at a single wavelength, offering a cost-effective, high-frequency sampling of atmospheric processes, making them particularly suitable for studying cloud dynamics in urban and industrial environments, where local emissions and surface heating play a crucial role. These comparisons highlight the unique advantages of ceilometers in providing continuous, cost-effective, and high-frequency atmospheric observations, making them particularly relevant for cloud monitoring in urban and industrial environments.

While existing approaches have utilized lidar-based imagery in combination with machine learning models [33], the availability of publicly accessible datasets remains

limited, creating a barrier for broader research and development in this field. To address this gap, our study introduces a newly curated dataset comprising backscatter profiles. Our new dataset differs significantly from existing cloud detection datasets in the literature. Traditional lidar remote sensing systems predominantly follow a satellite-to-earth (top-down) perspective, whereas our approach adopts a bottom-up (Earth-to-satellite) acquisition method. This inversion in the data collection paradigm captures atmospheric dynamics from a rarely explored viewpoint, introducing novel challenges and opportunities for cloud detection. This dataset represents a challenging benchmark for cloud detection due to its inclusion of diverse atmospheric conditions and varying cloud types observed around Mount Etna, an area known for its complex meteorological phenomena.

Ground-truth labeling of the dataset was performed using a high-resolution Weather Research and Forecasting (WRF) model, providing reliable reference data for model training and evaluation. Our study aims to provide a comprehensive performance benchmark for cloud detection on this dataset using several state-of-the-art deep learning architectures, including both convolutional neural networks (CNNs) and transformer-based models. Specifically, this research evaluated VGG-16 [130], ResNet50 [53], InceptionV3 [133], EfficientNet [135], and the Vision Transformer (ViT) [40]. Results indicated that ResNet50 achieved the highest accuracy among CNNs at 89.57%, while the transformer-based ViT reached a comparable performance of 89.36%.

The main contributions of this work are as follows:

- **Introduction of a Novel Dataset:** A new dataset of LiDAR ceilometer backscatter profiles is presented, collected over a three-month period near Mount Etna, Italy. This dataset, characterized by high temporal resolution

and diverse atmospheric conditions, serves as a valuable benchmark for cloud detection and atmospheric studies.

- **Comprehensive Benchmarking of State-of-the-Art Models:** The performance of cutting-edge deep learning architectures is evaluated, including CNN-based models (ResNet50, VGG16, InceptionV3, EfficientNet) and the Vision Transformer (ViT). This benchmarking provides a robust baseline for cloud detection task using lidar backscatter data.

- **High Accuracy Results:** Among the tested models, ResNet50 achieved the highest accuracy (89.57%), closely followed by ViT (89.36%). These results highlight the efficacy of residual learning and transformer-based approaches in analyzing complex atmospheric patterns.

- **Support for Future Research:** By making the dataset publicly accessible and offering detailed performance benchmarks, this work lays the foundation for future advancements in cloud detection and lidar-based atmospheric research.

- **Broader Application Potential:** The dataset and methodology introduced in this study open new opportunities for leveraging lidar ceilometer data to detect other atmospheric phenomena, such as aerosols, pollutants, and volcanic emissions.

These contributions represent a significant step forward in utilizing lidar-based systems and advanced deep learning techniques for accurate and scalable atmospheric monitoring. This work is an extension of the work presented in [30] by the

authors. It includes a more in-depth analysis of the state-of-the-art, a larger number of experiments and an in-depth and detailed comparison of results, not present in [30].

4.2 Related Works

In this section, the main state-of-the-art papers on data from ceilometers using AI and non-AI approaches will be explored. The aim is to provide a comprehensive and in-depth view of what is being done by researchers worldwide with this data using the most appropriate deep learning and computer vision architectures. The chapter will also focus on interpreting how certain atmospheric parameters such as the Atmospheric Boundary Layer or the Convective Boundary Layer have relevant impacts. The Atmospheric Boundary Layer (ABL) plays a critical role in determining air quality, weather patterns, and the overall dynamics of the Earth's atmosphere. As the lowest part of the atmosphere, its interactions with the Earth's surface influence the transport and dispersion of aerosols and other pollutants, which is crucial in urban environments like Wuhan, China. The growing use of advanced remote sensing techniques such as lidar and ceilometers has led to significant advancements in measuring and understanding the ABL. In addition, some studies in the field of Federated Learning will be analysed to better understand the use of this type of approach and how it can be useful in the context that has just been presented.

In [69], the authors study the convective boundary layer (CBL) height in Wuhan, China, using lidar data. They analyze the diurnal and seasonal cycle of the CBL, highlighting its impact on aerosol particle transport and dispersion. During the

day, the CBL grows, promoting the mixing and distribution of particles in the atmosphere, while at night, the CBL height decreases, limiting this mixing. A significant aspect of the study is the identification of an "entrainment" zone between the CBL and the troposphere, showing how seasonal variations in CBL height influence surface-level fine particle concentrations. In [143], the authors investigates the nocturnal boundary layer (NBL) in Wuhan, China, using lidar data to analyze how the height of the NBL is influenced by various meteorological parameters such as temperature, humidity, and wind speed. The study also examines how these variations in NBL height affect the concentration of fine particulate matter (PM2.5) near the surface, showing that a higher NBL tends to reduce particle accumulation, while a lower NBL promotes it. This works are closely related as both explore the dynamics of the atmospheric boundary layer (ABL) in Wuhan, but during different times of the day. [69] focuses on the diurnal behavior of the convective boundary layer (CBL) and its role in particle transport, while [143] examines the nocturnal conditions of the NBL and their impact on particle accumulation. [80] focuses on the analysis of the convective boundary layer (CBL) and the entrainment zone (EZ) in Wuhan, China, using a tilted polarized lidar. The study investigates the instantaneous depth of the atmospheric boundary layer (ABL) and its evolution through four typical phases (formation, growth, steady-state, and decay) during clear days across different seasons. The authors propose a novel approach to determine the thickness of the entrainment zone (EZT) by using the half-width of the variance profile of aerosol backscatter ratio fluctuations. The paper observes that the thickness of the EZT varies significantly between seasons, with winter and autumn showing a generally lower average thickness and standard deviation compared to spring and

summer. The study also notes that during the growth phase of the CBL, fluctuations and the thickness of the EZT are more pronounced, while these characteristics become less variable during the steady-state phase. [44] focuses on the study of aerosols in the Earth's atmosphere, particularly within the Mixing Layer (ML), the lowest part of the atmosphere where contaminant dispersion occurs. The thickness of the ML, influenced by surface conditions, is crucial for understanding atmospheric dynamics, thermodynamics, and air quality. Recently, networks of single-wavelength backscatter lidars, the ceilometers, have been implemented, primarily used by meteorological services to monitor the spatio-temporal distribution of aerosols. The paper introduces a fully automated approach for calibrating ceilometers, enabling the precise determination of the particle backscatter coefficient (β_p) regardless of weather conditions and the limited signal sensitivity of ceilometers. Additionally, the development of an automatic ML height detection algorithm, named COBOLT (Continuous Boundary Layer Tracing), allows for uninterrupted tracking of complete diurnal cycles of the ML, offering improved accuracy compared to existing algorithms. The paper demonstrates COBOLT's reliability through comparisons with radiosonde data and other algorithms, showcasing practical applications such as statistics of β_p profiles and comparisons of ML height between rural and urban areas. [126] examines aerosol layers present in the free troposphere above Wuhan, China, and their seasonal variation using 532 nm lidar measurements conducted throughout 2013. The authors identified 402 aerosol layer events in the free troposphere, with most layers located between 1 and 4 km in altitude. The majority of these layers are optically thin, with an aerosol optical depth (AOD) below 0.1. Seasonal variations show that the maximum thickness of aerosol layers occurs in spring, while the minimum is observed in autumn. Furthermore, the study reveals

that aerosol particles in these layers exhibit seasonal differences in shape and composition, with a higher presence of non-spherical and mixed particles during spring, autumn, and winter. [149] describes observations conducted in Wuhan, China, between 2010 and 2013, focusing on nine cases of humid aerosol layers or liquid water layers that slowly rose to altitudes of around 2-4 km during the winter. Initially, these layers were nearly transparent, with a low backscatter ratio and low depolarization ratio, indicating a low density of non-spherical particles or ice crystals. As these layers gradually lifted, they evolved into nearly opaque liquid cloud layers, and at that point, ice crystals suddenly formed at the upper edge of the cloud layer, suggesting that the water droplets were freezing. The freezing temperatures, estimated from radiosonde data, ranged between -3 and -8°C. In two cases observed over extended periods (more than 16 hours), the layers were located just below an inversion layer. The ice development in the layers was followed by the formation of precipitation.

Another state-of-the-art approach utilizing ceilometers involves the analysis and segmentation of lidar images for cloud monitoring. In this context, the primary challenge lies in accurately identifying the geometry and temporal position of clouds within the lidar imagery, which is crucial for improving our understanding of atmospheric processes. The application of deep learning techniques, such as convolutional neural networks, marks a significant advancement in cloud segmentation, enhancing the accuracy of cloud localization and classification detected by ceilometers. [33] proposes an innovative approach for segmenting laser radar (lidar) images by focusing on the geometry and temporal position of clouds. To achieve this goal, the authors employ a Fully Convolutional Network (FCN), a type of neural network specifically designed to analyze and segment images in a detailed manner. The method

described in the work relies on an innovative combination of semi-supervised and supervised learning techniques. Initially, the network is pre-trained using image-level annotations, which provide a general classification but lack pixel-level detail. This phase helps establish the foundation for cloud recognition by teaching the network to identify cloud features at a higher level. Next, the network is further refined by pre-training it with cloud positions provided by the MPLCMASK algorithm, an existing method for cloud masking. This step enables the FCN to learn cloud recognition based on reference data, enhancing its understanding of cloud geometries and positions in lidar imagery. Finally, to achieve optimal precision, the network undergoes fully supervised training using manually labeled data, where cloud positions are accurately marked. The authors of [162] explores the use of deep learning to monitor temporal variations in the Planetary Boundary Layer Height (PBLH). The authors combine two key techniques: an edge detection method based on lidar signals and a Convolutional Long Short Term Memory (LSTM) neural network. This approach allows for the estimation of PBLH height even under challenging conditions, such as rain or cloud cover. The convolutional LSTM network analyzes lidar images over time, predicting changes in the PBLH height. This model is inspired by previous applications, such as predicting the movement of numbers in image sequences. The results demonstrate that the model can accurately forecast temporal changes in PBLH height, offering a robust solution for continuous monitoring in various weather conditions.

Several approaches have been proposed for detecting clouds using lidar-based techniques, with significant differences in the type of instruments employed and the atmospheric layers they probe. While our instrument leverages lidar-based ceilometer systems to capture backscatter data from the atmospheric boundary layer, other

studies have employed alternative lidar configurations with distinct operational principles and observational capabilities. The approach in [122] incorporates a superconducting nanowire single-photon detector (SNSPD) in high-sensitivity atmospheric lidar, enhancing detection of faint backscatter signals from high-altitude clouds with minimal noise. However, these systems primarily target the mid-to-upper atmosphere and demand sophisticated calibration to mitigate signal attenuation. In contrast, hybrid radar-lidar methods, such as [163], integrate millimeter-wave cloud radar with ground-based multi-wavelength lidar. While radar provides superior penetration depth, lidar ensures finer resolution at lower altitudes. However, these composite systems face challenges in distinguishing drizzle from cloud droplets and require co-located instrumentation, limiting their spatial flexibility. Airborne laser scanning (ALS) lidar, as demonstrated in [153], employs aircraft-mounted pulsed lasers in the near-infrared spectrum to generate high-resolution 3D cloud reconstructions. Despite its ability to capture detailed cloud morphology, ALS is constrained by operational costs, limited temporal resolution, and dependency on flight schedules. Our tool differs from these instruments by focusing on earth-to-satellite observations using lidar-based ceilometer. Ceilometers operate at a single wavelength to continuously monitor the lower atmosphere. Their primary advantage lies in their ability to provide near real-time, high-frequency sampling of the planetary boundary layer (PBL), the atmospheric region most directly influenced by human activity and surface-level meteorological processes. Unlike high-power research lidar systems, ceilometers are optimised for detecting cloud base heights and aerosol layers at altitudes ranging from a few tens of metres to several kilometres. This capability makes them particularly relevant for studying cloud formation and dynamics in urban and industrial environments, where localised emissions and thermal

effects strongly modulate atmospheric composition.

For a comprehensive review of cloud detection, including the use of ceilometer data, please refer to [83]. Backscatter profiles acquired by ceilometers have been shown to be highly correlated in the presence of atmospheric particulate matter, as demonstrated in previous studies [92, 93]. The efficacy of ceilometer data has also been instrumental in detecting volcanic emissions during the 2010 eruption of the Icelandic volcano Eyjafjallajökull [43]. Given its potential, the exploration of sophisticated data mining techniques for the analysis of ceilometer-acquired data has been a subject of discussion since the inception of the Data Mining Project [7]. In pursuit of this objective, a noteworthy contribution to the research community emerged from the work of Wiegner et al. [148], wherein an approach to calibrate measurements from a Jenoptik CHM 15kx ceilometer was presented. Later, Arun et al. [4] delved into the synergy between ground-based ceilometer observations and satellite data from remote sensing sources in their study. By combining these distinct datasets, they aimed to enhance the precision of cloud detection, highlighting the evolving landscape of data fusion for atmospheric analysis.

In [66], the authors proposed a technique for detecting specific meteorological phenomena, such as fog and clouds, using a lidar-based ceilometer. The methodology involved the application of classical machine learning methods, including Support Vector Machines (SVM), as well as shallow neural networks. These techniques leveraged raw data obtained from the ceilometer as predictive features, enabling the accurate identification of atmospheric events. Similarly, in [58], the authors undertook cloud classification by taking advantage of both ceilometer data with sky images captured by a camera. Within their study, a random forest approach was employed to perform multi-class classification, effectively discerning various cloud

types. This integration of data sources facilitated comprehensive cloud identification. In [19], ceilometer data have been utilized to evaluate a federated learning approach incorporating both labeled and unlabeled samples in a semi-supervised setting. This methodology aims to enhance model performance by leveraging feature extraction from unannotated data, contributing to the broader research on privacy-preserving machine learning for Earth observation applications. Sleeman et al. [132] used lidar-based ceilometer data to detect the Planetary Boundary Layer Height (PBLH) with the use of machine learning techniques. In [35], they introduced an unsupervised methodology for classifying meteorological occurrences, leveraging k-means clustering. An autoencoder was trained to learn a suitable representation of backscatter profiles, subsequently organized into clusters. While demonstrating promise, this technique was presented as a prototype proof-of-concept. Notably, the absence of labeled data and a comprehensive evaluation hampered its full validation. Conversely, the study in [33] addressed cloud detection through Fully Convolutional Networks. In their approach, backscatter profiles were provided into their model via a *mask algorithm*, and the model was trained in a supervised fashion, as they labeled a dataset of backscatter profiles. This dataset enabled an in-depth quantitative performance analysis of their proposed methodology, setting it apart from prior works in the literature. An et al. [3] developed a cloud detection algorithm based on FY-3E satellite infrared channels for early morning observations. Their method utilizes dynamic thresholds and auxiliary data (such as SST, LST, and snow cover masks) to adjust for varying land surface conditions and improve detection accuracy. In contrast, Li et al. [73] proposed a Residual Dual U-Shape Network (RD-UNet) with improved skip connections, which effectively integrates multi-scale features to better detect thin clouds and refine cloud boundaries. Although both approaches rely on

satellite imagery, our research diverges by employing ceilometer lidar backscatter data, offering a bottom-up perspective that captures high temporal resolution and detailed vertical structure information. The system used in this research uses lidar-based ceilometers to gather backscatter data primarily from the atmospheric boundary layer. Conversely, other research efforts have adopted different lidar setups, each with unique functional mechanisms and observational strengths. The method described in [122] features a high-sensitivity atmospheric lidar system equipped with a Superconducting Nanowire Single-Photon Detector (SNSPD), which improves the identification of weak backscatter signals from upper-atmosphere clouds while minimizing noise. Nonetheless, such systems are mainly designed for mid-to-high atmospheric layers and require intricate calibration techniques to address signal degradation. Alternatively, hybrid radar-lidar approaches, such as the one in [163], merge millimeter-wave cloud radar with ground-based lidar operating at multiple wavelengths. In these setups, radar excels at deep penetration, whereas lidar provides finer detail at lower elevations. However, these combined systems struggle to differentiate between drizzle and cloud particles and rely on co-located instruments, restricting their deployment flexibility. Airborne Laser Scanning (ALS) lidar, exemplified in [153], uses aircraft-mounted, pulsed near-infrared lasers to construct high-resolution, three-dimensional representations of cloud structures. Although effective in capturing cloud morphology, ALS faces limitations due to its high operational expenses, restricted temporal coverage, and reliance on flight schedules. The system used in this research stands apart by employing ceilometers for ground-to-satellite measurements. These instruments operate at a single wavelength and offer continuous monitoring of the lower atmosphere. Their key strength lies in their ability to deliver real-time, high-frequency measurements of the Planetary Boundary Layer

(PBL)—the atmospheric zone most affected by surface-level weather and human-induced changes. Unlike powerful research lidar systems, ceilometers are tailored for monitoring cloud base altitudes and aerosol concentrations within a vertical range spanning from a few dozen meters to several kilometers. This makes them especially suitable for examining cloud behavior and evolution in urban and industrial settings, where local emissions and heat exchanges significantly influence atmospheric properties. The current research landscape demonstrates diverse approaches leveraging ceilometer data, often relying on distinct datasets and traditional machine learning methods (*e.g.*, SVM). However, the complexity of backscatter profiles, as collected in this study, makes them more suitable for deep neural networks. To address gaps in the existing literature, this work introduces a high-resolution dataset tailored for deep learning applications in cloud detection. Collected near an active volcano, this dataset captures unique and challenging atmospheric conditions, enabling rigorous benchmarking of advanced models and fostering future research. Traditional remote sensing systems using lidar technology are predominantly satellite-based and acquire data from a top-down (satellite-to-earth) perspective. Although these approaches have been extensively studied and provide valuable atmospheric and surface observations, they differ fundamentally from our methodology, which adopts a bottom-up (Earth-to-satellite) perspective. This inversion in the data acquisition paradigm introduces a new and exciting aspect to cloud detection, as the proposed dataset captures atmospheric dynamics from a perspective rarely explored in the literature. Because of this unique perspective, this dataset differs significantly from existing datasets, making it new and ahead of its time. The distinct nature of these data presents new challenges and opportunities for the research community, which motivated the proposal of a dedicated scientific challenge. The promotion of advances

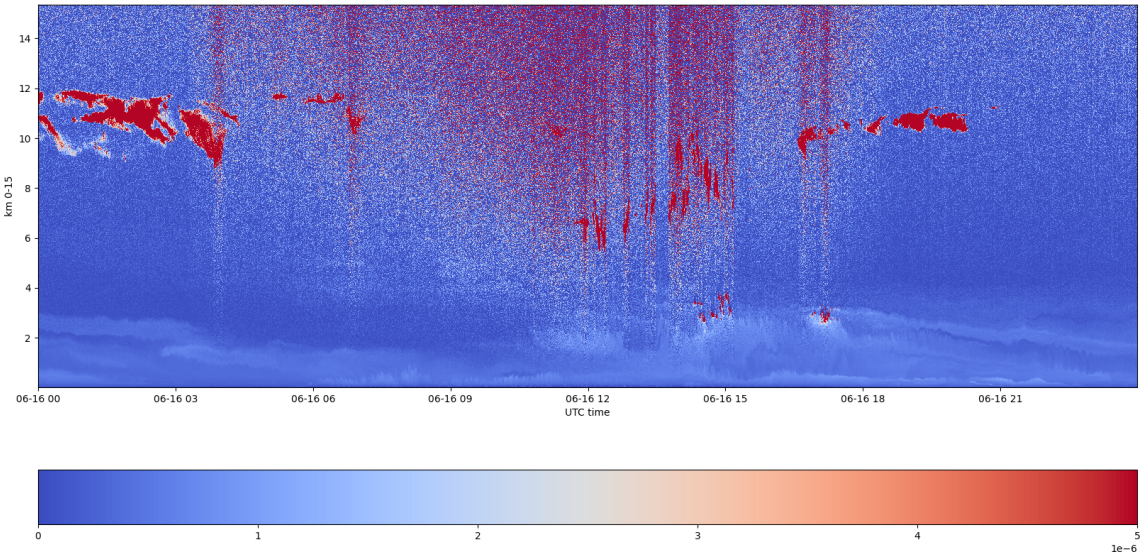


Figure 4.1: Backscatter profile of 24-hour measurements taken on the 16^{th} of June 2022. As explained in Chapter 3, the color of the plot depends on the intensity of the measured particle: intense blue means absence of particulate; red means intense presence of particulate.

in cloud detection methodologies and the improvement of the understanding of atmospheric phenomena observed by ground-based LiDAR systems are among the objectives of this study.

4.3 Proposed Method

As detailed in Chapter 3, the ceilometer carried out measurements every 15 seconds, enabling precise quantification of atmospheric particle concentration. By analyzing the reflected signals, cloud layer coverage could be determined. The selected ceilometer generates and processes a substantial volume of raw data, as shown in Table 3.1. Figure 3.2 illustrates the data collection procedure.

Once the raw data were acquired, the parameters of interest were normalized

using a specific calibration factor for the lidar-based ceilometer. Several parameters contributed to constructing backscatter profiles. These profiles were plotted with time represented on the x-axis and particle height (reflected in the backscatter coefficient) on the y-axis. The plot's colors indicate the intensity of the measured particles: deep blue signifies minimal particulate presence, while red indicates high concentrations. The scale ranges numerically from 0 to $5 \cdot 10^{-6}$. A backscatter profile for each day of data collection was generated, further dividing it into hourly intervals, resulting in 24 profiles per day. Figure 4.1 provides an example of the processed data. In total, 1,568 images of dimensions 150×1000 were created, each representing an hour-long measurement period.

An extract of the Python code used to generate the images from the raw data is presented below.

```
def raw_data_img(days):
    day = days
    all_files = os.listdir(path)
    elem_list = []
    for elem in all_files:
        elem_list.append(path+"/"+elem)
    data = nc.MFDataset(elem_list)

    bsc=data.variables['beta_att'][:, :]
    bsc=np.transpose(bsc)
    kaltitude=data.variables['range'][:]/1000
    hr_altitude=kaltitude[:512]
    hr_bsc=bsc[:512]
    cdf_time=data.variables['time'][:]

    time = nc.num2date(cdf_time, data.variables['time'].units,
    ↪  only_use_python_datetimes=True)
    str_time = [i.strftime("%Y-%m-%d %H:%M") for i in time]
    date_time=[datetime.strptime(i, "%Y-%m-%d %H:%M") for i in str_time]

```

```
20      fig, ax=plt.subplots(figsize=(1.5, 10))
21      pl=ax.pcolormesh(date_time, kaltitude, bsc, cmap='coolwarm', vmin=0.0,
        ↪  vmax=0.000005)
22      plt.axis('off')
23      plt.tight_layout()
24      plt.savefig(path_save+"/"+day+"-"+hour+".png")
```

The generated backscatter profiles were labeled using the *Weather Research and Forecasting* (WRF) Model, a mesoscale numerical prediction system designed for atmospheric research and operational forecasting. Figure 4.2 presents the workflow of the WRF model, which features two dynamic cores, a data assimilation system, and a software architecture optimized for parallel computing. The model serves a broad spectrum of meteorological applications, covering scales from tens of meters to thousands of kilometers. Its spatial resolution of 1×1 km offers greater detail compared to typical global forecast models, which often operate at 27×27 km. The WRF model leverages global weather data from the Global Forecast System (GFS), provided by the *National Center for Atmospheric Research* (NCAR) [96].

The WRF model produces netCDF files representing a 3D geographic grid, as depicted in Figure 4.3. Latitude and longitude are aligned with the x-axis and y-axis, while 40 pressure levels are represented on the z-axis. The displayed images differ in spatial resolution and the quality of the WRF model outputs. The first image has a resolution of 9×9 km, meaning that each point on the spatial grid is spaced 9 km apart. This results in a relatively coarse depiction of atmospheric conditions, as finer details of meteorological phenomena are not captured at this scale. In contrast, the second image utilizes a higher-resolution grid of 3×3 km, achieved through the nesting technique within the WRF model. Nesting involves embedding one or more high-resolution grids (referred to as nested domains) within a coarser

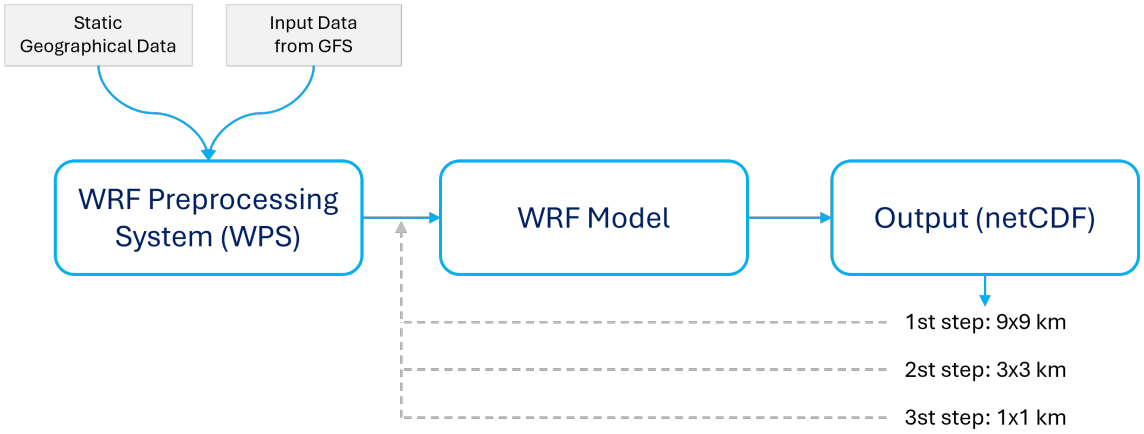


Figure 4.2: Adopted workflow for the employed WRF model. Note that the outputs of the first and second steps serve as inputs to the last step. GFS is the Global Forecast System. Global weather data from the GFS are used as the first input.

grid (the parent domain). In this case, the 3×3 km grid is nested inside the parent domain of 9×9 km. During this process, the WRF model incorporates meteorological data from the parent domain to enhance the local representation of atmospheric phenomena, resulting in more precise forecasts for specific areas of interest. Finally, the third image presents data with a resolution of 1×1 km, which offers nine times the precision of the initial grid. At this level, nesting is further refined by adding a third nested domain, enabling the model to capture highly detailed meteorological features, such as localized variations in temperature, wind, and precipitation. This setup enabled us to isolate the central point of the reference domain, corresponding to the ceilometer's geographical location, and determine cloud presence or absence at each pressure level. This process provided hourly cloud cover data for the ceilometer's location, serving as ground-truth labels for each backscatter profile with high reliability.

An extract of the Python code used to assign the correct labels to each image is presented below.

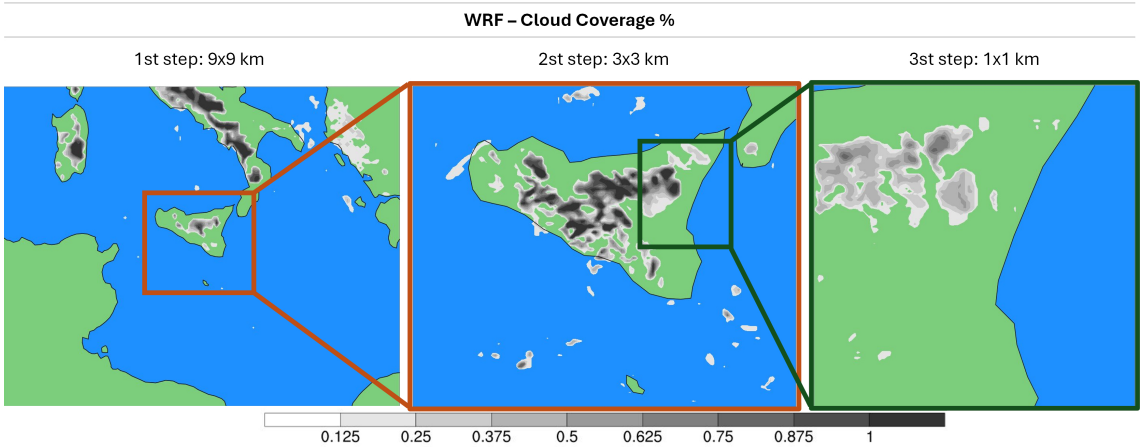


Figure 4.3: Visual representation of the output of WRF model showing the percentage cloud cover.

```
    def raw_data_wrf(years, months, days):
        for hour in range(0, cloud_var.shape[0]):
        clouds = False
        # We have multiple layers in high
        for high in range(0, cloud_var.shape[1]):
            # sn means south-north
            for sn in range(0, cloud_var.shape[2]):
                if (cloud_var.shape[2] / 2 - 1) <= sn <= (cloud_var.shape[2] / 2
                ↪  + 1):
                    # we means west-east
                    for we in range(0, cloud_var.shape[3]):
                        if (cloud_var.shape[3] / 2 - 1) <= we <=
                        ↪  (cloud_var.shape[3] / 2 + 1):
                            if cloud_var[hour][high][sn][we] != 0:
                                # This is the centre of the grid, with an
                                ↪  appropriate margin
                                clouds = True
        print("in hour " + str(hour) + " are there clouds? " + str(clouds))
        f.writelines([str(hour), "\n", str(clouds), "\n"])
```

The labeled backscatter profiles were subsequently used to train several state-of-the-art deep learning models, including VGG-16 [130], ResNet50 [53], InceptionV3 [133], EfficientNet [135], and ViT [40], utilizing the PyTorch framework.

VGG-16 is a classic Convolutional Neural Network (CNN) known for its simple and uniform architecture, which serves as a strong baseline in image classification tasks. Inception v3 introduces the concept of inception modules, allowing the network to capture multi-scale features efficiently while reducing computational cost. ResNet-50 employs residual connections to enable the training of deeper networks and mitigate the vanishing gradient problem, proving highly effective in various vision tasks. EfficientNet scales network width, depth, and resolution in a principled manner, achieving high accuracy with fewer parameters. Lastly, Vision Transformer (ViT) adopts a transformer-based architecture that operates directly on image patches, providing an alternative to convolutional approaches and achieving state-of-the-art results in image recognition. The dataset is publicly available at https://zenodo.org/records/10616434.

4.4 Experimental Results

All models were trained using a standard supervised learning approach with Cross Entropy loss. Training and inference were conducted on 1-hour-long ceilometer measurements. All models were initialized with ImageNet-pretrained weights to enhance training stability and robustness. To mitigate overfitting during training, horizontal-flip data augmentation was used. For each experimental setup, the dataset was divided into training, validation, and test sets, with proportions of 49% (769 samples), 21% (329 samples), and 30% (470 samples), respectively, totaling 1050 samples belonging to the True class.

The exploration of different hyper-parameter configurations was carried out, and two optimizers were employed: Stochastic Gradient Descent (SGD) and Adam. The

following subsection outlines the combinations tested to identify the optimal model configuration.

4.4.1 Performance Analysis and Comparison of Model Configurations

Figure 4.4 and Figure 4.5 show the performance results of various state-of-the-art deep learning models trained on a dataset of cloud detection using ceilometer backscatter profiles. The models evaluated include VGG16, EfficientNet, InceptionV3, ResNet50, and Vision Transformer (ViT), and were tested using multiple configurations and optimization strategies. The Figures primarily differ based on the optimizer employed (SGD for Figure 4.4 and ADAM for Figure 4.5). Specifically, the diagrams in Figure 4.4 are organized according to training parameters like learning rate, momentum, and weight decay, whereas the diagrams in Figure 4.5 are grouped based solely on learning rate and weight decay. The models were evaluated using standard metrics, including Accuracy, F1-score, Precision, and Recall, to gauge their efficacy in detecting clouds accurately.

The VGG16 model exhibited notable variability across different configurations. The best performance was observed with configuration 3 using SGD, yielding an accuracy of 86.38%, an F1-score of 0.9077, a Precision of 0.8495, and a Recall of 0.9744. The high recall value indicates that VGG16 was proficient at capturing relevant positive instances (i.e., cloud presence). However, its slightly lower precision compared to recall highlights the model's tendency to generate false positives. Interestingly, configurations with a higher learning rate (0.01) performed poorly, achieving a mere accuracy of 32.98%. This suggests that higher learning rates caused instability, potentially leading to divergence or overfitting during the training process.

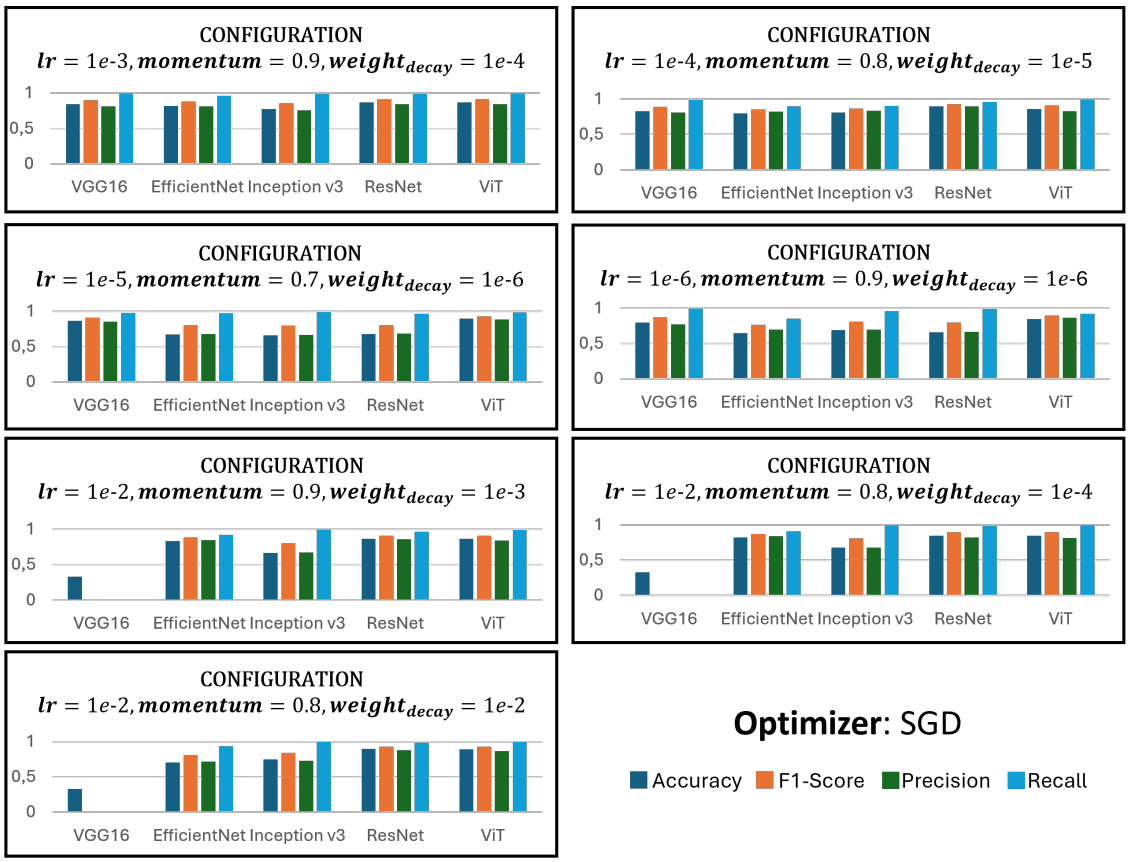


Figure 4.4: Performance results of various state-of-the-art deep learning models trained for cloud detection using ceilometer backscatter profiles, evaluated with SGD optimizer and organized by training parameters such as learning rate, momentum, and weight decay. Models include VGG16, EfficientNet, InceptionV3, ResNet50, and Vision Transformer (ViT), with metrics like Accuracy, F1-score, Precision, and Recall used for assessment. Missing values in some graphs indicate that the value of the metric in question is close to or equal to 0.0.

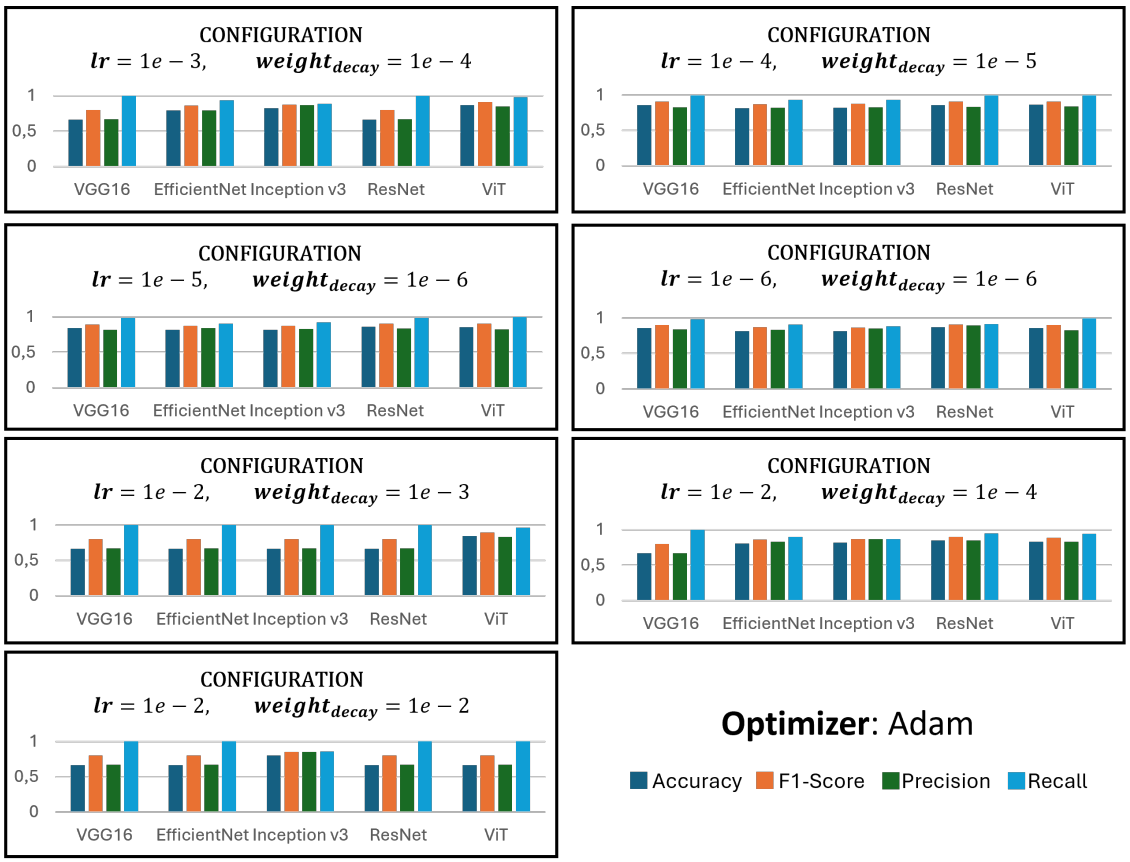


Figure 4.5: Performance results of various state-of-the-art deep learning models trained for cloud detection using ceilometer backscatter profiles, evaluated with the ADAM optimizer and organized by training parameters such as learning rate and weight decay. Models include VGG16, EfficientNet, InceptionV3, ResNet50, and Vision Transformer (ViT), with metrics like Accuracy, F1-score, Precision, and Recall used for assessment.

EfficientNet demonstrated solid performance, with its best configuration using SGD achieving an accuracy of 83.19% alongside an F1-score of 0.8824 and a Precision of 0.8450. This result reflects a balance between precision and recall, suggesting that EfficientNet effectively identified cloud instances without a significant number of false positives or negatives. However, when optimized with Adam, EfficientNet's performance slightly decreased, with the highest accuracy obtained being 81.91%. This difference underscores that certain architectures benefit more from one optimizer over another, with SGD appearing more suitable for EfficientNet in this context.

InceptionV3 showed a moderate performance range, achieving its highest accuracy of 82.77% with Adam optimization. This configuration exhibited a balanced F1-score (0.8751) and Precision (0.8656), indicating robust performance across different metrics. When optimized with SGD, InceptionV3 achieved a comparable accuracy of 81.06% but required the maximum number of epochs (59), reflecting a slower convergence rate compared to Adam. This suggests that InceptionV3 might be more efficient with Adam, especially for datasets with complex patterns like backscatter profiles.

ResNet50 emerged as the top-performing model, achieving the highest accuracy of 89.57% using SGD with configuration 2 (learning rate = 10^{-4}, momentum = 0.8, weight decay = 10^{-5}). The corresponding F1-score of 0.9273 highlights the model's superior robustness and generalization capabilities. ResNet50 maintained consistent performance even when optimized with Adam, achieving an accuracy of 87.23% under configuration 4. The consistently high recall values indicate the model's strong ability to detect relevant instances, making it well-suited for real-world applications in cloud detection.

The Vision Transformer (ViT) model demonstrated competitive results, closely following ResNet50. The highest accuracy achieved by ViT was 89.36% using SGD with configuration 3 (learning rate = 10^{-5}), accompanied by strong precision (0.8795) and recall (0.9808). ViT's performance underscores the potential of transformer-based architectures for complex tasks such as cloud detection. Even when optimized with Adam, ViT maintained robust performance, achieving an accuracy of 85.96% with configuration 4. This suggests that transformer-based models, when properly tuned, can rival traditional convolutional networks in such specialized tasks.

Overall, a comparison of optimizers across models revealed that SGD tended to produce higher accuracy scores compared to Adam, though Adam often provided faster convergence, requiring fewer epochs. Lower learning rates generally resulted in more stable and higher accuracies, while higher learning rates (e.g., 0.01) frequently led to poor performance, indicating potential issues with stability and overfitting during training. In terms of model performance, ResNet50 consistently outperformed other architectures, demonstrating the efficacy of residual connections for feature extraction from backscatter data. ViT, while slightly behind ResNet50 in accuracy, showed promise, especially given its strong recall and balanced performance metrics. The high recall scores across many configurations suggest a strong capability to capture positive instances (cloud presence), but they also highlight the need to balance precision, as seen with models like VGG16 and EfficientNet.

In conclusion, the results highlight that ResNet50 and Vision Transformer are highly effective models for cloud detection using ceilometer backscatter profiles. Their robust performance, particularly in terms of recall, demonstrates their strong

	SGD			ADAM	
#	**Learning Rate**	**Momentum**	**Weight Decay**	**Learning Rate**	**Weight Decay**
1	10^{-3}	0.9	10^{-4}	10^{-3}	10^{-4}
2	10^{-4}	0.8	10^{-5}	10^{-4}	10^{-5}
3	10^{-5}	0.7	10^{-6}	10^{-5}	10^{-6}
4	10^{-6}	0.9	10^{-6}	10^{-6}	10^{-6}
5	10^{-2}	0.9	10^{-3}	10^{-2}	10^{-3}
6	10^{-2}	0.8	10^{-2}	10^{-2}	10^{-2}
7	10^{-2}	0.8	10^{-4}	10^{-2}	10^{-4}

Table 4.1: The first half of the table shows the parameters used for experiments with SGD. The second half of the table shows the parameters used for experiments with Adam.

Model	**Configuration**	**Optimiser**	**Accuracy**	**F1-score**	**Precision**	**Recall**	**Training Time (min.) / Last epoch**
VGG16	#3	SGD	86.38	0.91	0.85	0.97	22 m / 39
	#4	Adam	86.17	0.91	0.84	0.98	24 m / 38
EfficientNet	#5	SGD	83.19	0.88	0.85	0.92	16 m / 14
	#4	Adam	81.91	0.87	0.84	0.91	37 m / 59
Inception v3	#2	SGD	81.06	0.87	0.83	0.90	30 m / 59
	#1	Adam	82.77	0.88	0.87	0.88	10 m / 15
ResNet 50	#2	SGD	**89.57**	**0.93**	**0.90**	0.96	27 m / 59
	#4	Adam	**87.23**	**0.91**	**0.90**	0.92	23 m / 59
ViT	#3	SGD	89.36	**0.93**	0.88	**0.98**	129 m / 59
	#1	Adam	86.81	**0.91**	0.85	**0.98**	25 m / 9

Table 4.2: Test performance of the considered state-of-the-art models. Bold values highlight the best-performing model for each evaluation metric when using either SGD or Adam as optimizers. Please refer to Table 4.1 for the hyper-parameter configurations.

suitability for real-world atmospheric monitoring applications. This analysis underscores the potential of these models to accurately identify and classify cloud presence, providing a reliable foundation for further advancements in environmental monitoring and data-driven atmospheric analysis.

4.4.2 Final Model

The final hyper-parameters used for training with SGD and Adam optimizers are reported in Table 4.1. An early stopping criterion was applied, whereby training would terminate if the variation in validation loss remained within a margin of $\delta = 0.05$ for at least three consecutive epochs. In total, 70 experiments were carried

out, all of which are available at the following GitHub repository: https://github.com/alessiochisari/CeilometerDatasetBenchmark.

These experiments were performed on Google Colab Pro equipped with a *Tesla T4* GPU with 16GB GDDR6 memory. Table 4.2 shows the results obtained by training the models with the best set of hyper-parameters (*c.f.* Table 4.1) for each of the two chosen optimizers.

Figure 4.6 provides a detailed view of the performance trends for several deep learning architectures evaluated across four key metrics: *Accuracy* (%), *Precision* (%), *Recall* (%), and *F1-score* (%). Each metric is plotted as a function of the number of training epochs, shown on the x-axis, up to a maximum of 60 epochs. However, the training process incorporates an early stopping mechanism, which terminates the training when the difference in loss between consecutive epochs falls below a predefined threshold for a fixed number of epochs. As a result, the number of epochs varies across models and optimizer configurations, reflecting differences in learning dynamics and convergence. For each architecture, the graphs show the best-performing solutions corresponding to the two optimisers, Adam and SGD, with all the previously mentioned metrics (accuracy, precision, recall and F1-score) obtained according to the best parameter configuration shown in Table 4.2.
The metrics themselves provide valuable insights into the strengths and weaknesses of the models. For instance, in the case of VGG16, the recall metric consistently increases across epochs, demonstrating the model's ability to effectively identify positive instances (e.g., cloud presence). However, the precision curve shows some fluctuations, suggesting a propensity for occasional false positives. EfficientNet, on the other hand, achieves a balance between precision and recall, with its performance metrics stabilizing effectively as training progresses. This indicates strong

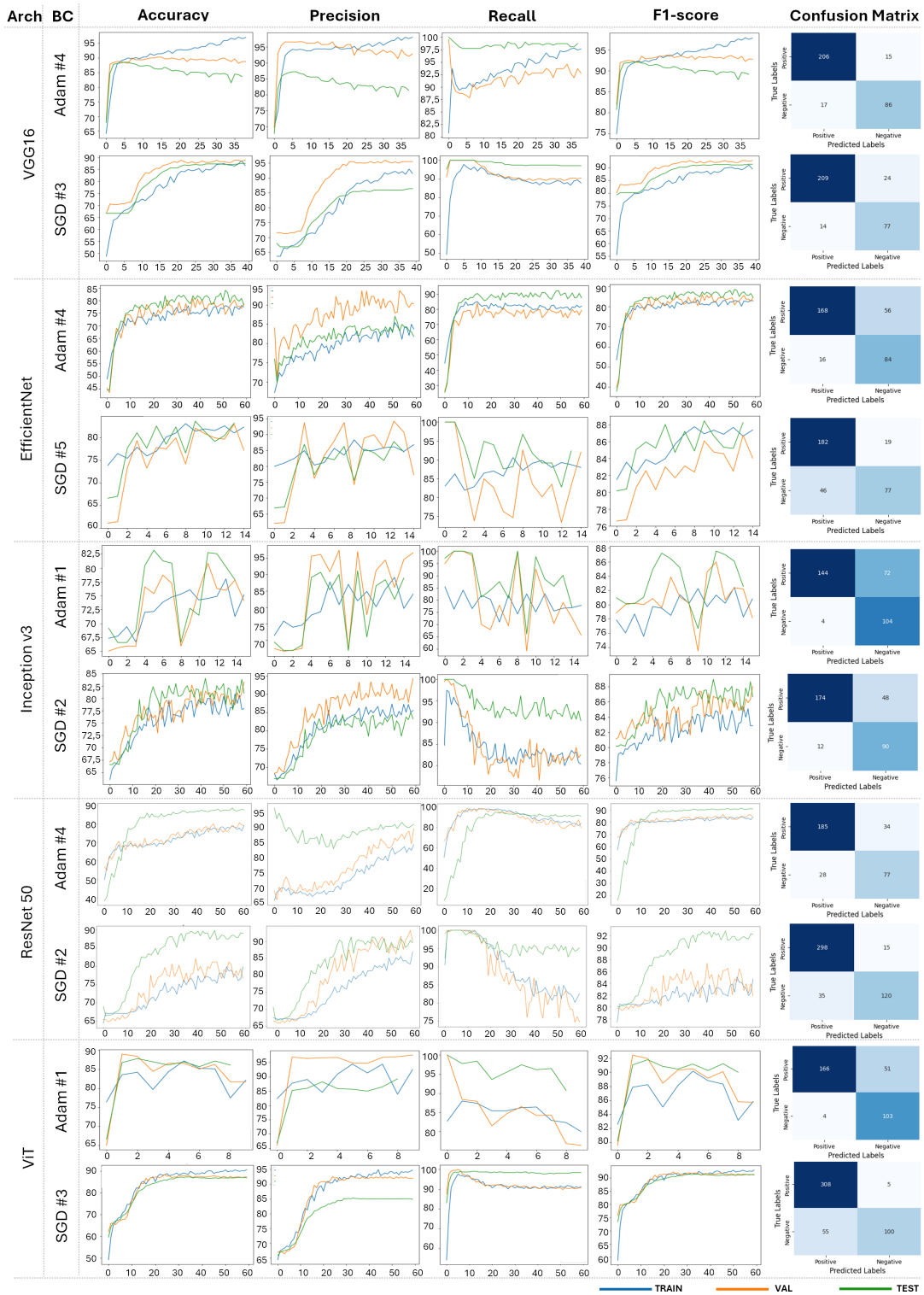


Figure 4.6: Results of the best models for each architecture compared to the Adam and SGD optimizers. The names of the architectures are shown in the Arch (Architecture) column. The configuration number (Table 4.2) of the best results are given in the column BC (Best Configuration).

generalization capabilities, though the model may take slightly longer to converge compared to others.

InceptionV3 presents an interesting case, with moderate performance across all metrics. It demonstrates reliability in capturing relevant patterns in the data, as evidenced by its recall trends, though its overall accuracy is slightly lower compared to top-performing models. Adam optimization appears to benefit this architecture, as the model converges more quickly and achieves its best results with fewer epochs compared to SGD.

ResNet50 stands out as the best-performing architecture across all metrics. Its accuracy and F1-score remain consistently high, and the recall metric underscores its exceptional ability to detect positive instances with minimal false negatives. This performance is likely due to the advantages of residual connections, which help the model capture hierarchical features more effectively. The Vision Transformer (ViT) also delivers impressive results, rivaling ResNet50 in accuracy and recall. Its performance demonstrates the potential of transformer-based architectures for tasks involving complex patterns in atmospheric data. While Adam leads to faster convergence for ViT, the final performance metrics are marginally better when the model is optimized with SGD.

Regarding training times and last epoch (last column of Table 4.2), it can be noted that them vary across models, influenced by the early stopping criterion used to prevent overfitting. The VGG16 model requires 22–24 minutes with 38–39 epochs, demonstrating quick convergence. Similarly, Inception v3 shows a range from 10 minutes for configuration #1 to 30 minutes for configuration #2, both stopping after 59 epochs, as early stopping was likely triggered to prevent overfitting. The EfficientNet model, with more complex architecture, requires longer training times

(16–37 minutes) and up to 59 epochs, reflecting the balance between model complexity and training duration. ResNet 50 achieves efficient performance with training times of 23–27 minutes, stopping after 59 epochs for both configurations. The Vision Transformer (ViT) takes the longest training time (129 minutes for configuration #3), with 59 epochs, due to its computational intensity, though configuration #1 converges in 25 minutes and 9 epochs. All models are pre-trained, and the use of early stopping ensures that training is halted before overfitting occurs. Thus, while VGG16 and Inception v3 train faster, EfficientNet and ViT offer higher accuracy at the cost of longer training durations. After completing the training of all models, the inference time is generally negligible and takes between 10 to 20 seconds for a batch size of 12 samples, or approximately 1 to 1.7 seconds per sample. The final column of plots (Figure 4.6), showing the confusion matrices provided by the best model for each involved architecture, offers additional insights into the classification performance of each model. The diagonal elements of these matrices represent correctly classified instances, while off-diagonal elements indicate misclassifications. For models like ResNet50 and ViT, the confusion matrices reveal a strong ability to correctly classify both positive and negative instances, reinforcing their suitability for the task.

Overall, the analysis highlights the interplay between model architecture, optimizer choice, and the early stopping mechanism. ResNet50 emerges as the most robust and reliable model, followed closely by ViT, while architectures like EfficientNet and InceptionV3 offer competitive alternatives with specific strengths. The early stopping mechanism ensures efficient training, preventing overfitting and reducing computational costs, while still enabling a thorough evaluation of model performance. These results demonstrate the promise of advanced neural network

architectures for challenging tasks such as cloud detection from lidar backscatter data.

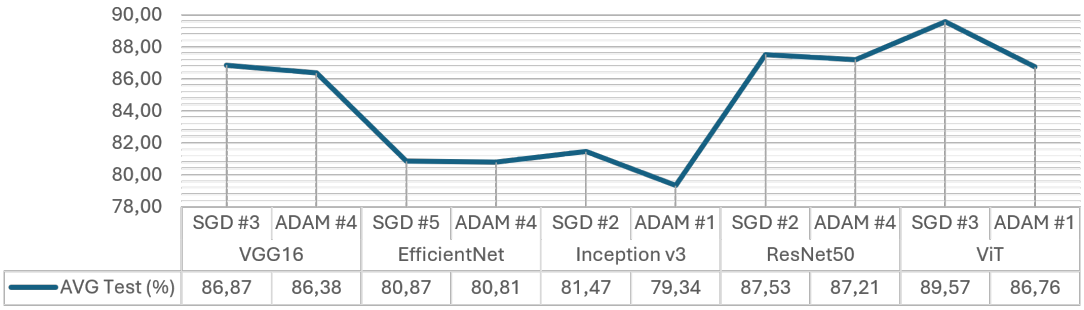
4.5 Discussion

The results presented in this study highlight several critical observations regarding the performance of state-of-the-art deep learning architectures for cloud detection using lidar-based ceilometer backscatter data. The implications of these findings, as well as the strengths and limitations of the proposed methodology, are discussed below.

4.5.1 Performance Analysis of Models

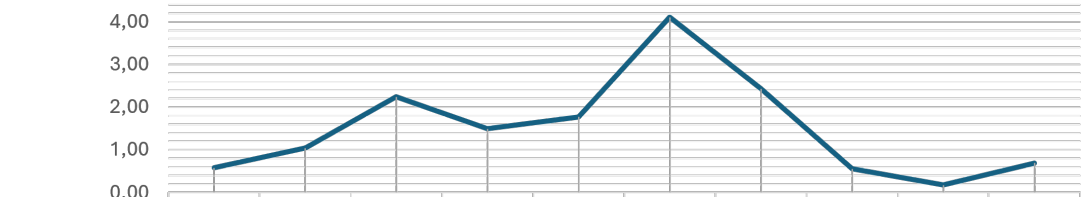
Our experimental results revealed that ResNet50 achieved the highest accuracy (89.57%) among CNN-based architectures, closely followed by the Vision Transformer (ViT) with 89.36%. This performance gap suggests that residual connections in ResNet50 provide significant advantages in extracting hierarchical features from complex backscatter profiles. Meanwhile, ViT's ability to model global dependencies demonstrates the potential of transformer-based approaches for atmospheric data analysis. These findings are consistent with the growing success of hybrid and transformer models in computer vision.

Other architectures, such as VGG16, EfficientNet, and InceptionV3, showed lower, albeit competitive, performance. Notably, VGG16 exhibited high recall values, indicating its reliability in identifying cloud presence, but at the cost of increased false positives. EfficientNet, while slightly behind in accuracy, offered a balanced



	SGD #3	ADAM #4	SGD #5	ADAM #4	SGD #2	ADAM #1	SGD #2	ADAM #4	SGD #3	ADAM #1
	VGG16		EfficientNet		Inception v3		ResNet50		ViT	
AVG Test (%)	86,87	86,38	80,87	80,81	81,47	79,34	87,53	87,21	89,57	86,76

(a)



	SGD #3	ADAM #4	SGD #5	ADAM #4	SGD #2	ADAM #1	SGD #2	ADAM #4	SGD #3	ADAM #1
	VGG16		EfficientNet		Inception v3		ResNet50		ViT	
SD Test (%)	0,58	1,04	2,24	1,49	1,77	4,11	2,43	0,55	0,17	0,69

(b)

Figure 4.7: Average (AVG) (a) and Standard Deviation (SD) (b) of the 10 best experiments of the best model configurations.

trade-off between precision and recall, which could be beneficial for specific real-time applications where false negatives are particularly detrimental.

The results shown in Figure 4.7 present the average test accuracy and corresponding standard deviation for the top 10 experiments using the best configurations of Adam and SGD optimizers across various architectures. These metrics offer complementary insights: average accuracy reflects general performance, while standard deviation reveals how stable the model is across multiple runs.

Starting with VGG16, both optimizers achieve comparable accuracy: SGD reaches 86.87%, slightly outperforming Adam at 86.38%. However, SGD shows more consistent results, with a lower standard deviation (0.58%) compared to Adam (1.04%). This suggests that while performance is similar, SGD yields more stable outcomes. For EfficientNet, results are nearly identical in terms of accuracy—SGD at 80.87%

and Adam at 80.81%. Yet, Adam exhibits slightly better stability with a lower standard deviation of 1.49% versus 2.24% for SGD. This makes Adam marginally more reliable in repeated runs, despite similar performance. The Inception v3 architecture reveals a clearer distinction. SGD significantly outperforms Adam in both metrics: it achieves a higher average accuracy (81.47% vs. 79.34%) and a notably lower standard deviation (1.77% vs. 4.11%). These results indicate that SGD is both more accurate and considerably more stable, making it the preferred choice for this architecture. In contrast, ResNet50 shows minimal differences in accuracy—SGD slightly leads with 87.53% over Adam's 87.21%. However, stability tells a different story: Adam has an impressively low standard deviation of 0.55%, compared to SGD's 2.43%. This highlights Adam's robustness for ResNet50, despite the small accuracy gap. Lastly, in the case of the Vision Transformer (ViT), SGD achieves the highest overall accuracy at 89.57%, surpassing Adam's 86.76%. Additionally, SGD offers superior consistency with a remarkably low standard deviation of 0.17%, while Adam records 0.69%. Here, SGD stands out as both the most accurate and the most stable optimizer.

The comparison across architectures underscores the importance of evaluating both accuracy and stability when selecting an optimizer:

- Adam is particularly effective with ResNet50 and EfficientNet, where it provides higher consistency.
- SGD clearly outperforms Adam in Inception v3 and ViT, excelling in both accuracy and robustness.
- In VGG16, the two optimizers are comparable, but SGD provides slightly better and more stable results.

These findings suggest that optimizer selection should balance both performance and repeatability, especially in real-world applications where consistency across multiple training runs is crucial.

4.5.2 Optimizer Sensitivity and Hyperparameter Impact

A notable observation was the sensitivity of model performance to optimizer choice and hyperparameter configurations. Models optimized with SGD generally outperformed those trained with Adam, particularly in achieving higher accuracy and stability. However, Adam demonstrated faster convergence, which could be advantageous for computationally constrained scenarios. This optimizer-dependent performance underscores the importance of hyperparameter tuning for specialized tasks like cloud detection.

4.5.3 Dataset Characteristics and Challenges

The dataset curated for this study, derived from lidar ceilometer backscatter profiles, presented unique challenges due to its high temporal resolution and the variability of atmospheric conditions. The presence of diverse cloud types, combined with the influence of Mount Etna's complex meteorological phenomena, created a demanding environment for model training and evaluation. Despite these challenges, the models achieved promising results, demonstrating the potential of lidar data for real-world cloud detection tasks.

One limitation of the dataset is its geographical specificity, as data were collected exclusively in the vicinity of San Giovanni La Punta, Catania, Italy. Future studies

could benefit from expanding the dataset to include backscatter profiles from multiple regions with varying climatic conditions. This would enhance the generalizability of the models and facilitate cross-regional comparisons.

4.5.4 Practical Implications and Applications

The high accuracy and recall achieved by ResNet50 and ViT make these models viable candidates for deployment in operational meteorological systems. Their robust performance in detecting cloud presence from lidar data could complement existing satellite and radar systems, particularly for identifying low-altitude clouds and localized aerosol concentrations.

Moreover, the potential for applying this approach to detect other atmospheric phenomena, such as pollutants or volcanic emissions, is noteworthy. The ability of lidar-based systems to capture fine-grained vertical profiles of the atmosphere could pave the way for monitoring air quality, early warning systems for natural disasters, and climate research.

4.5.5 Strengths, Limitations and Future Works

The proposed approach for cloud detection using lidar-based ceilometer backscatter data demonstrates several notable strengths that distinguish it from traditional remote sensing methodologies. One of the main strengths lies in its novel bottom-up data acquisition. Unlike conventional satellite-based techniques, our method captures atmospheric dynamics from an Earth-to-satellite perspective, allowing for high-frequency, real-time monitoring of the lower atmosphere. This unique vantage point enables the detailed observation of phenomena such as low-altitude cloud

formations and localized aerosol concentrations that are often underrepresented in traditional approaches.

Another strength is the robust performance of the deep learning models. The benchmarking experiments highlighted that architectures such as ResNet50 and Vision Transformer (ViT) are particularly effective in extracting hierarchical and global features from complex backscatter profiles. Their high accuracy and recall not only validate the efficacy of the method but also underscore the potential for integrating these models into operational meteorological systems for tasks like early-warning detection and real-time environmental monitoring.

Despite these significant advantages, several limitations must be acknowledged. The geographical scope and temporal window of the dataset are relatively limited, as the data were collected exclusively near San Giovanni La Punta, Catania, Italy over a three-month period. This geographical and temporal confinement may affect the generalizability of the models when applied to regions with different climatic conditions or extended seasonal variations. Moreover, the performance of the models is notably sensitive to the choice of hyperparameters and optimizer configurations. As observed in our experiments, variations in learning rate, momentum, and weight decay can lead to significant fluctuations in accuracy and stability, indicating a potential challenge in achieving consistent performance across diverse training scenarios.

Looking ahead, several future directions can be pursued to build upon the current work. First, expanding the dataset, both in terms of geographic diversity and duration, would enhance model robustness and facilitate cross-regional comparisons, making the approach more broadly applicable. Additionally, investigating alternative and hybrid architectures, such as integrating CNNs with transformer-based

models, could leverage the complementary strengths of local feature extraction and global context modeling, potentially leading to further performance improvements. Finally, significant efforts should be directed towards real-time deployment. This involves optimizing the computational efficiency of the models and reducing latency to facilitate their incorporation into practical, operational meteorological systems, especially in edge computing scenarios where resources are limited.

4.6 Conclusion and Future Works

In this study, a novel approach for cloud detection using lidar-based ceilometer backscatter data was proposed and evaluated, benchmarked against state-of-the-art deep learning models. By leveraging a newly curated dataset characterized by high temporal resolution and diverse atmospheric conditions, the efficacy of advanced neural network architectures in accurately detecting cloud presence was demonstrated.

The experimental results highlight ResNet50 as the top-performing model, achieving an accuracy of 89.57%, with the Vision Transformer (ViT) closely following at 89.36%. These findings underscore the advantages of residual learning and transformer-based global attention mechanisms in capturing complex patterns in backscatter profiles. Additionally, models such as VGG16, EfficientNet, and InceptionV3 provided competitive performance, offering alternative solutions based on specific application requirements, such as reduced false negatives or computational efficiency.

This work contributes to the field in two significant ways:

- **Dataset Availability:** a publicly accessible dataset of labeled backscatter

profiles acquired over three months near Mount Etna, Italy, was provided. This dataset represents a valuable resource for advancing research in cloud detection and atmospheric studies.

- **Comprehensive Benchmarking:** Several deep learning architectures were evaluated under varying hyperparameter configurations and optimization strategies, establishing a robust baseline for future developments in LiDAR-based cloud detection.

Despite these achievements, our study identifies areas for improvement, including expanding the dataset to cover diverse geographic regions, exploring alternative architectures, and optimizing models for real-time deployment. Future research could also explore the application of this approach to detect other atmospheric phenomena, such as aerosols, pollutants, or volcanic emissions, further broadening the utility of lidar-based systems in environmental monitoring.

In conclusion, our results validate the potential of combining lidar ceilometer data with advanced deep learning techniques to enhance cloud detection capabilities. This integration represents a significant step forward in developing accurate, scalable, and automated solutions for atmospheric monitoring, with implications for meteorology, climate research, and environmental protection.

Chapter 5

Cloud Detection Challenge

The dataset acquired was also used to perform a challenge: The *Cloud Detection Challenge*, hosted by IEEE MetroXRAINE 2024 conference, which aimed to advance cloud detection leveraging lidar-based ceilometer data. In this chapter, the challenge will be discussed in depth: from the motivation to the analysis of the best submitted solutions.

The challenge invites participants to develop state-of-the-art binary classification models for cloud detection. Unlike conventional imagery, these profiles provided unique insights into atmospheric conditions, capturing variations in cloud presence over time. Participants were tasked with surpassing our benchmark using a deep network to outperform our results in terms of accuracy, precision, recall, and F1 score [12].

Several teams from around the world participated in the challenge, presenting solutions that leveraged advanced techniques to address the dataset's unique challenges. The competition served as a platform for testing innovative methods, fostering collaboration, and advancing the field of atmospheric data analysis. This work details the competition's key aspects, highlights the diverse approaches adopted by participating teams, and evaluates the performance of our proposed solution in

comparison to others.

5.1 Description

The goal of the proposed challenge[1] was to encourage teams around the world to develop innovative solutions outperforming our baseline [30] performance of this new dataset. The performance of the models was evaluated using the following metrics: (i) accuracy; (ii) F1 score; (iii) precision; (iv) recall. The winner has been determined by the team with the best value across all metrics. In the event of a tie, the values of precision and recall had higher priority.

5.2 Significance of the challenge

The Cloud Detection Challenge holds paramount importance in computer vision applications and image analysis. The ability to accurately identify the presence of clouds in satellite imagery or landscape photographs carries profound implications across various sectors, including meteorology, environmental monitoring, agriculture, and satellite imaging. By using a new dataset based on new types of data and, therefore, new types of images, there is an opportunity to increase research on this topic by integrating new data sources with those already known. Precise cloud classification is essential for understanding climate changes, predicting weather phenomena, and optimizing agricultural operations. The challenge not only calls for innovation in developing advanced models but also provides an opportunity to make substantial contributions to scientific and technological progress in strategic sectors dependent

[1] https://iplab.dmi.unict.it/cloud-detection-challenge/ - last accessed: September 2024

on image analysis accuracy. By participating in this challenge, researchers and developers can showcase their skills and abilities in computer vision, contributing to creating solutions that push beyond current technological frontiers. The results have a tangible impact on real-world applications, enhancing our understanding of the environment and supporting informed decision-making across various industries.

5.3 Criteria of judging a submission

The evaluation of submissions in the Cloud Detection Challenge was designed to ensure a fair and comprehensive assessment of each proposed solution. Given the complexity of lidar-based backscatter data, the criteria focused on both the accuracy of predictions and the balance between the other performance metrics. This approach aimed to highlight not only the ability of the models to correctly classify cloudy vs. clear skies but also their robustness in handling edge cases and maintaining consistency across varying conditions. The following criteria outline the specific metrics and their role in determining the effectiveness of the submitted models.

1. **Classification Accuracy**: Precision in correctly distinguishing between the two classes is crucial. Accuracy will be used as a starting point to assess the overall model performance.

2. **Precision and Recall**: Precision indicates the proportion of true positive predictions among all positive predictions. Recall measures the proportion of true positive predictions among all actual positive instances. A good balance between precision and recall is desirable, but their importance may vary depending on the context of the problem.

3. **F1 Score**: The F1 score is the harmonic mean of precision and recall. This metric provides a balance between the two and can be particularly useful in cases where minimizing both false positives and false negatives is important.

5.4 Baseline

ResNet-50 was chosen as the best architecture in the binary classification task, presented in [30] in which this baseline solution is presented. The dataset is divided into training and validation sets in a 70-30 ratio, and preprocessing steps include resizing, normalization, and data augmentation to enhance model generalization. The ResNet-50 model, pretrained on ImageNet, is adapted with a custom classification head to suit the binary task.

Methodology. The proposed approach uses a convolutional neural network (CNN) approach for classification, leveraging the ResNet-50 architecture pretrained on ImageNet.

Data Preprocessing. Dataset Structure: The dataset is organized into train and validation directories, each containing two subfolders (true and false). The split ratio is 70% - 30%. Transformations:

- Resized to 224×224 pixels.
- Resizing crop to 224 pixels.
- Random horizontal flips (probability: 50%) to increase diversity.
- Normalized using the ImageNet mean ([0.485, 0.456, 0.406]) and standard deviation ([0.229, 0.224, 0.225]).

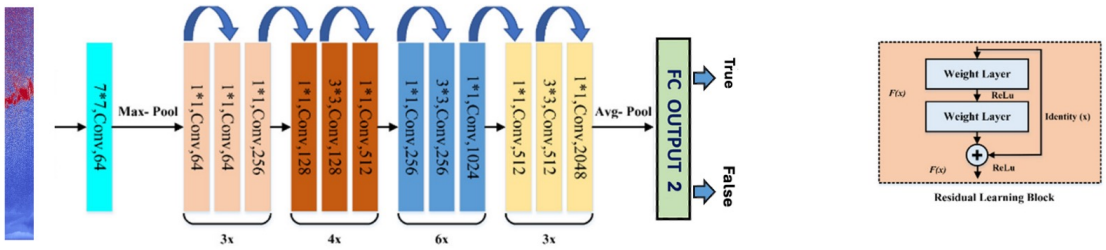


Figure 5.1: Baseline architecture for binary cloud classification: the diagram illustrates a ResNet-50 pre-trained network, employing residual blocks for deep learning of spatial and temporal features from backscatter profiles acquired via ceilometer. The final output is a binary classification, indicating 'True' or 'False' for cloud presence (figure adapted from [1]).

Model Architecture. Base Model: ResNet-50, a 50-layer deep CNN, is utilized as the backbone, as represented in Figure 5.1. Its pretrained weights on ImageNet facilitate robust feature extraction. Custom Head: The fully connected layers are replaced or extended to suit the binary classification task, ensuring compatibility with the dataset while preserving essential learned features.

Training Strategy. Device: Training is performed on a GPU-enabled Google Colab Pro instance. Loss Function: CrossEntropyLoss is used to optimize model predictions for binary classification. Optimizer: Stochastic Gradient Descent (SGD) is employed, enabling controlled weight updates via adjustable learning rates. Metrics: Performance is evaluated using accuracy, F1-score, precision, and recall.

Experimental Setup. Hardware: Google Colab Pro provides GPU acceleration, significantly reducing training time and computational overhead. Data Access: The dataset resides in Google Drive, mounted as a root directory within the Colab environment for seamless integration. Hyperparameters: Batch size and learning rates are defined but may vary across iterations. The number of epochs and early stopping criteria ensure optimal convergence.

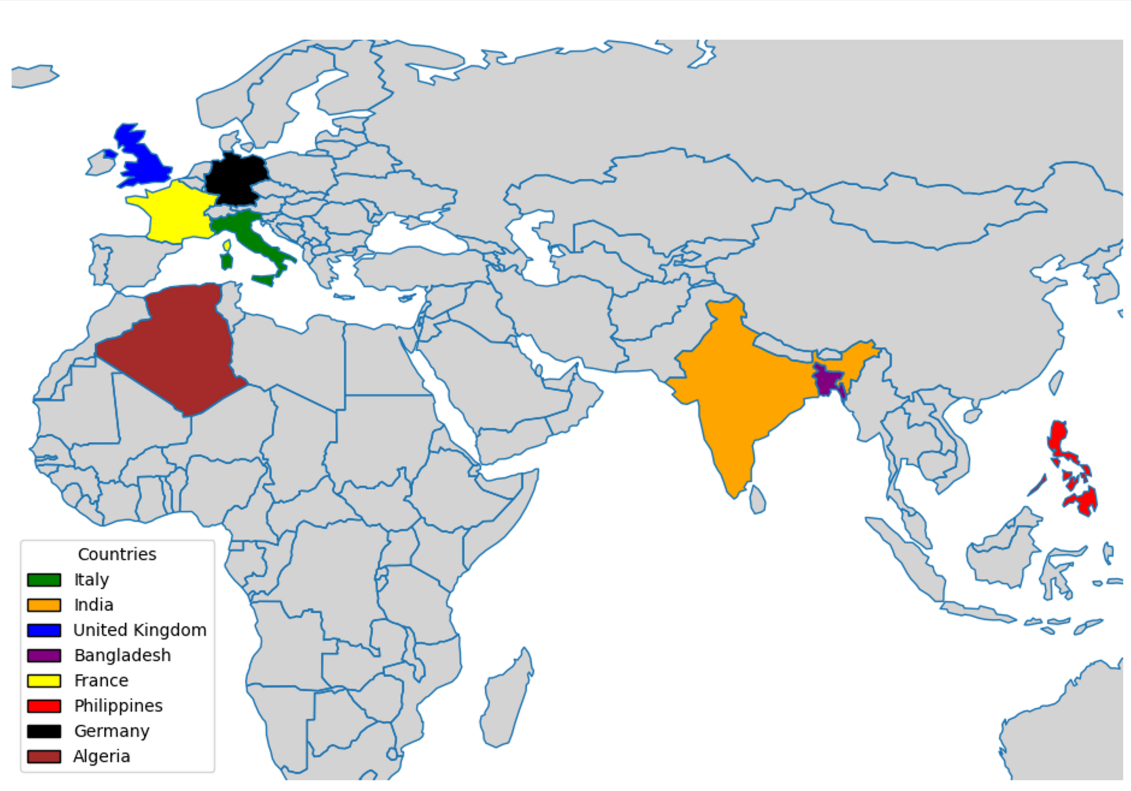


Figure 5.2: Registered participants by affiliation country.

5.5 Participant Submissions to the Challenge

A total of 11 teams from all over the world registered for the challenge. In particular, 63.6% of the participants are single researchers, while 36.4% are research groups. Figure 5.2 shows the countries of origin of the teams registered to the challenge, while Figure 5.3 shows the number of teams from the respective country. The top two solutions were selected based on (i) the absolute values of the evaluation metrics and (ii) the best-proposed architecture. In the following subsection, details of the solutions proposed by each participant are provided, including a brief presentation, the methodology, and the adopted experimental approach.

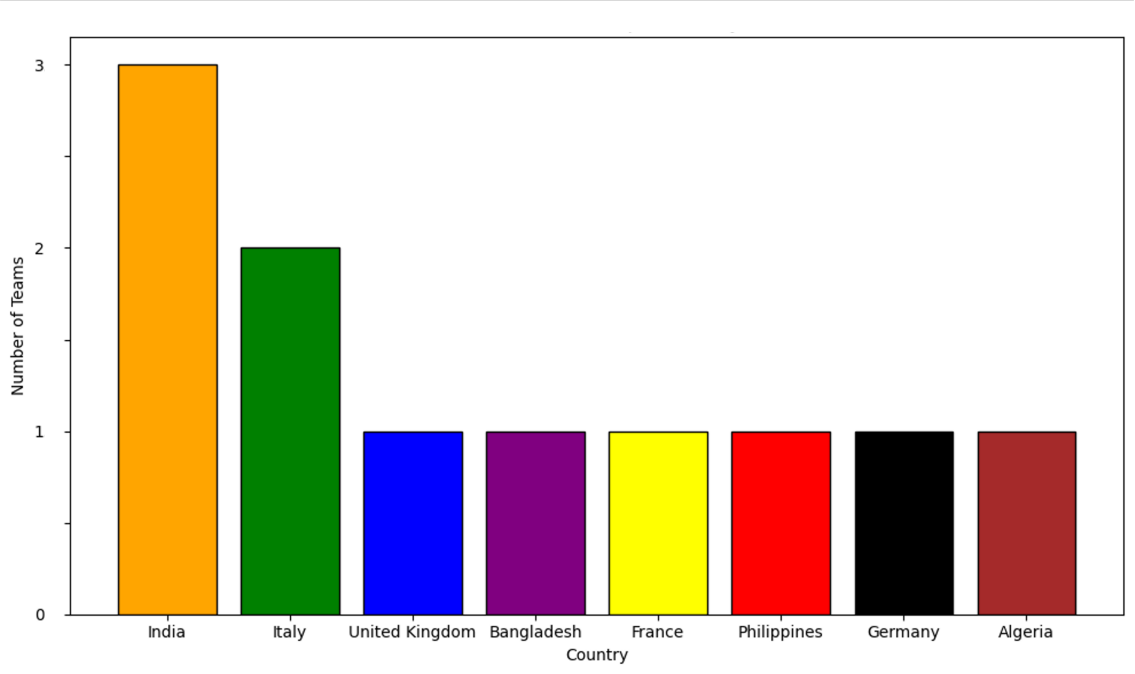


Figure 5.3: Number of participating teams per country.

5.5.1 Alpha Research Group - Unversity of Turin (UniTO)

Alpha Research Group (represented by Bruno Casella - University of Turin) used a pretrained version of the Vision Transformer (ViT) [40] as shown in Figure 5.4. The idea behind using a transformer architecture comes from the intrinsic nature of the dataset, as it contains both spatial and temporal features. Taking inspiration from the BEVT paper [142], which proposes a BERT [38] pretraining of Video Transformers and states that for difficult actions, the spatial priors learning should be decoupled from the temporal priors learning, the UniTO researcher hypothesized that temporal features could benefit from spatial features and vice versa.

Methodology. Each image is resized to 384×384. Training and validation data are normalized (mean and standard deviation of 50%). The pretrained ViT was trained by minimizing the binary cross-entropy loss with mini-batch gradient descent using

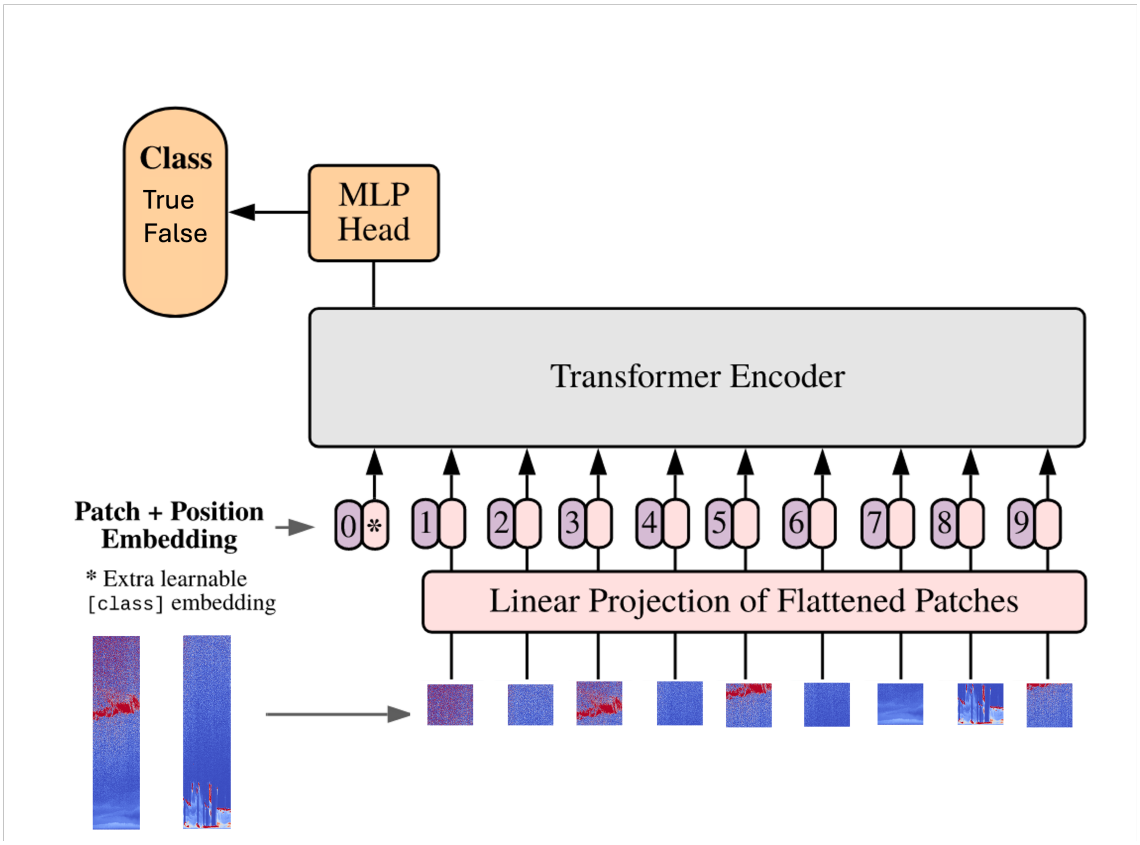


Figure 5.4: Alpha Research Group's proposed architecture.

the SGD optimizer, with a learning rate of 0.0001, momentum of 0.7, weight decay of 0.000001, and batch size was 12. An early stopping criterion with patience 1 and delta 0 was set for a maximum of 60 training epochs. As augmentation techniques, the participant adopted random horizontal flips, applied with a probability of 50%.

Experimental Setup. The Alpha Research Group used a dedicated server with an Intel Xeon Processor (Skylake, IBRS, 8 sockets of one core) and one Tesla T4 GPU to run the experiments. PyTorch 2.0.1 was adopted as deep learning framework. Each epoch required around 2 minutes. The validation loss, in combination with the early stop criterion, was used as the evaluation metric.

5.5.2 Koexai (industrial sector)

This section highlights Koexai's contribution to the Cloud Detection Challenge, focusing on the innovative approach employed for the automatic classification of atmospheric conditions using ceilometer data. The challenge aimed to create a robust model for accurately differentiating between cloudy and clear skies based on images created from data captured by these devices. To achieve this, Koexai designed a dual-stream deep learning architecture that processes both RGB and grayscale images to enhance feature extraction and improve classification performance.

Data Preparation. To prepare the dataset for training and validation, the original dataset was split into 80% for training and 20% for validation. This division was performed while ensuring a balanced distribution of target classes, utilising the Bhattacharyya distance [11] to minimise biases. Although clouds typically cover only a portion of each image, labels indicating the presence or absence of clouds were provided at the image level. This highlights an opportunity for improving automatic classification by refining the dataset labelling, in addition to enhancing the classification model itself. No domain-specific transformations were applied to the dataset during this process.

Model Architecture. The proposed model utilises two ResNet-101 [53] backbones operating in parallel: one dedicated to processing RGB images and the other to handling their grayscale counterparts, as shown in Figure 5.5. This dual architecture allows for a more comprehensive feature extraction from the data, with the RGB stream capturing colour and texture details, while the grayscale stream emphasises structural and contrast-based attributes. The ceilometer images were re-scaled to 224×224 pixels to align with the input format of the ResNet-101 models. The last 1024 feature maps from each backbone were then average pooled and concatenated.

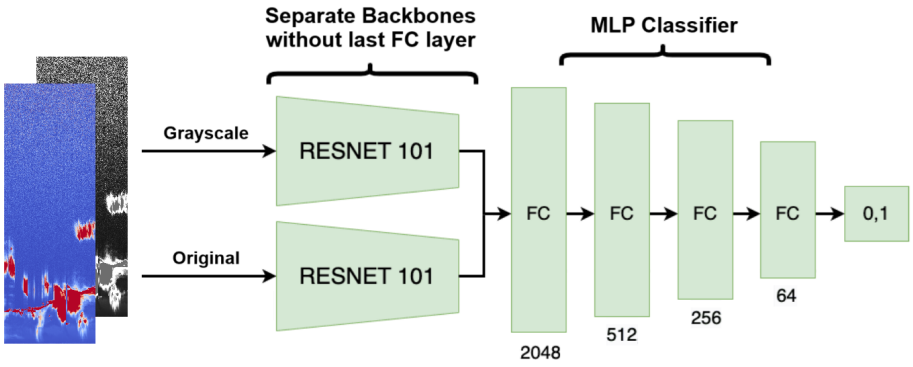
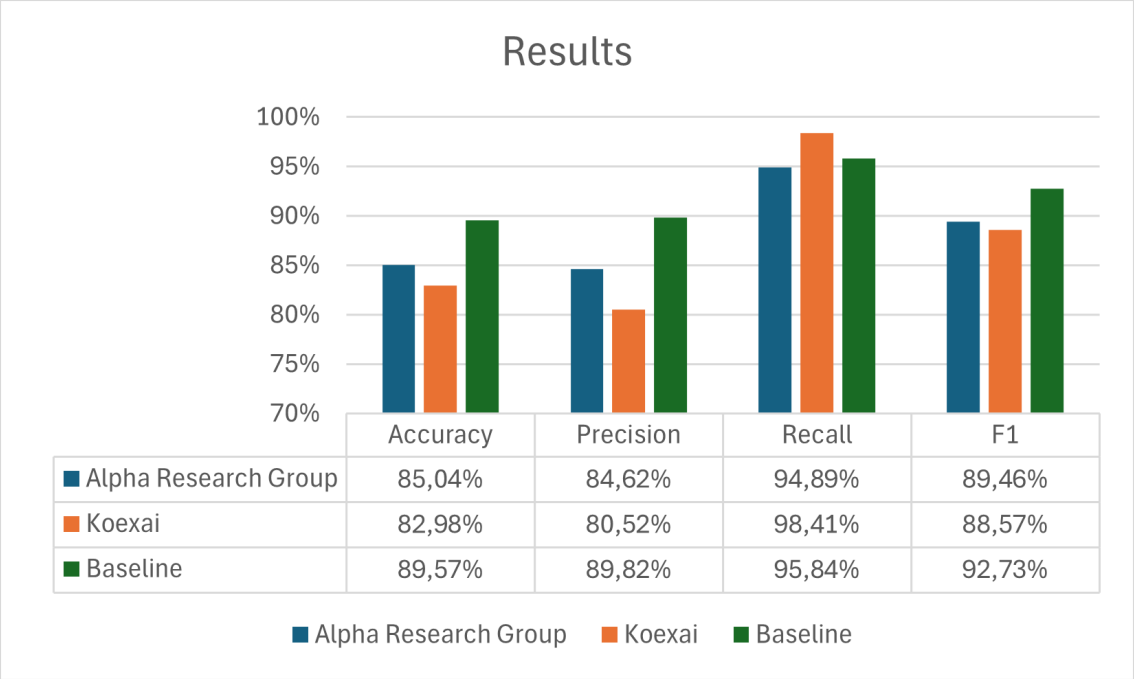


Figure 5.5: Koexai's proposed architecture.

These concatenated features were subsequently passed through a multi-layer perceptron (MLP) consisting of three fully connected hidden layers with 512, 256, and 64 neurons, employing LeakyReLU activation functions. The architecture culminates in a final sigmoid activation function for binary classification.

Training Strategy. The model was implemented using PyTorch and trained for 500 epochs with the Adam optimizer [67], utilising default parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$). To enhance training efficiency, several techniques were employed, including an adaptive learning rate, early stopping criteria, gradient clipping, and weight decay set to (1×10^{-5}). The weights of both ResNet-101 networks were initialised with pre-trained weights from ImageNet and fine-tuned separately to adapt them to the specific classification task.

Results and Discussion. Although quantitative metrics such as accuracy and F1-score could not be computed on the test set due to challenge constraints, these metrics were assessed on the validation set. The results demonstrated that the model effectively distinguished between cloudy and clear skies, with minimal signs of overfitting. The dual-stream architecture outperformed single backbone models



	Accuracy	Precision	Recall	F1
Alpha Research Group	85,04%	84,62%	94,89%	89,46%
Koexai	82,98%	80,52%	98,41%	88,57%
Baseline	89,57%	89,82%	95,84%	92,73%

Figure 5.6: Graphical results for each team.

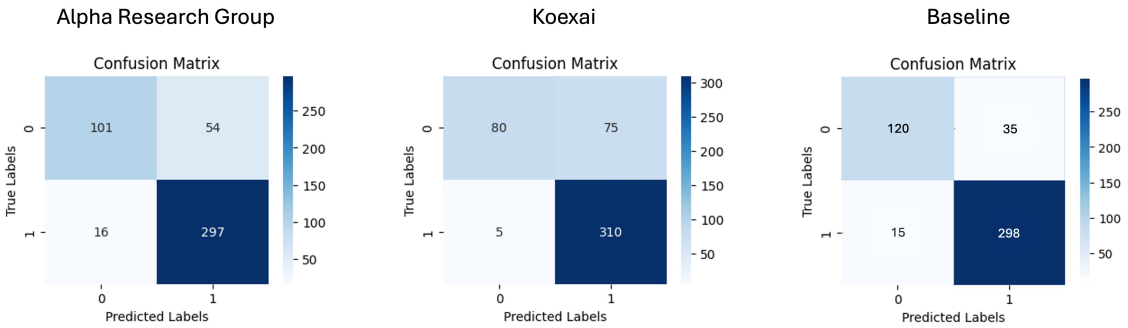


Figure 5.7: Confusion matrix for each team solution: [fist] Alpha Research Group's solution, [second] Koexai's solution, [third] Baseline solution. 0 stands for label "False", 1 for label "True".

by capturing a wider range of visual features and leveraging information from both channels to enhance overall model quality. Koexai's innovative approach highlights the potential of dual-stream architectures in improving cloud detection capabilities. The model's robustness suggests that it is well-suited for integration into automated weather monitoring systems and can be further adapted for other meteorological tasks, such as cloud type classification or atmospheric anomaly detection.

5.6 Ranking and Discussion

The Cloud Detection Challenge aimed to foster innovative approaches for analyzing lidar-based ceilometer data, pushing the boundaries of binary cloud detection. The results obtained from participating teams reveal key insights into the efficacy of diverse architectures and methodologies when applied to a novel and complex dataset. This section provides a comprehensive analysis of the results obtained from the challenge. A focus is placed on performance metrics, architectural choices, and the broader implications of the proposed solutions. Figure 5.6 shows the key values for each metric for each team participating in the competition, while Figure 5.7 shows all confusion matrices for each proposed solution. Each Figure includes the baseline results.

5.6.1 Computational Needs

The computational needs associated with each of the proposed approaches, as delineated in Table 5.1, can be characterized by several markedly distinct facets. Koexai's approach is associated with a substantially longer training time (3 hours, 59 minutes, and 12 seconds) when juxtaposed with the training durations of the Alpha Research

	Training Time	**Epochs**	**SO**	**Hardware**
Alpha	20m:28s	10	Ubuntu	Intel Xeon CPU Tesla T4 GPU
Koexai	3h:59m:12s	500	Debian 12	Intel Core i5 8th CPU GTX 1080 Ti GPU
Baseline	26m:39s	60	Ubuntu	Intel Xeon CPU Tesla V100 GPU

Table 5.1: Comparison of the computational needs of each proposed solution.

Group and Baseline methods, which are approximately 20 minutes, 28 seconds and 26 minutes, 39 seconds, respectively. This pronounced disparity in training times is primarily attributable to the divergent training strategies adopted by the respective approaches. Specifically, Koexai's proposed method required complete training of the architecture from its initial, uninitialized state. In contrast, both the Alpha Research Group and the Baseline approaches capitalized on the benefits of employing pre-trained architectures. In addition to the differences in overall training duration, a significant discrepancy is evident in the number of training epochs implemented across the approaches. Koexai's proposed approach, which engaged in full training from scratch, was run for 500 epochs, while the Alpha Research Group approach required only 10 epochs and the Baseline approach was limited to 60 epochs. These lower epoch counts are not simply indicative of a truncated training process but rather are the result of the employment of an early stopping criterion, used to avoid overfitting during the training process. The third aspect warranting detailed consideration is the influence of the computational hardware on the overall training process. Koexai used a system configured with an Intel Core i5 8th generation CPU and an NVIDIA GTX 1080 Ti GPU. Both Alpha Research Group and the Baseline approaches were executed on systems that utilized Intel Xeon CPUs in conjunction with NVIDIA GPUs that are particularly well-suited for deep learning applications, such as the Tesla T4 GPU for the Alpha Research Group and the Tesla V100 GPU

for the Baseline. Koexai, by virtue of its full-from-scratch training regimen and extended epoch count, naturally incurs a higher computational cost relative to the pre-trained models utilized by the Alpha Research Group and Baseline approaches. Moreover, the disparities in hardware further contribute to the observed variations in training durations.

5.6.2 Performance Summary

The challenge dataset provided a benchmark for state-of-the-art solutions, with the baseline model achieving 89.57% accuracy, 92.73% F1-score, 89.82% precision, and 95.84% recall. These metrics served as a reference point for evaluating the success of participant submissions.

Among the participants:

- *Koexai Team*: Surpassed the baseline in recall, indicating a strong ability to correctly identify true positive cases (cloudy skies). However, other metrics, including accuracy, F1-score, and precision, remained slightly below the baseline.

- *Alpha Research Group*: Delivered competitive results with a transformer-based approach, but the performance did not exceed the baseline in any key metric.

These outcomes highlight the difficulty of outperforming the robust baseline, which leveraged deep learning to effectively capture the temporal and spatial nuances of the ceilometer data. In terms of the various metrics, it can be stated that overall *accuracy* was high across all solutions, reflecting their ability to classify samples as cloudy or clear skies effectively. However, no model consistently outperformed the baseline, suggesting that while general predictions were reliable, subtle challenges

such as ambiguous atmospheric conditions may have limited further improvements. *Precision* remained strong for both the baseline and participant models, indicating that most predicted cloudy conditions were indeed correct, although some solutions that prioritized recall saw a slight trade-off in precision. *Recall*, on the other hand, was a standout metric for the Koexai team, whose dual-stream CNN architecture effectively leveraged both RGB and grayscale information to enhance sensitivity to cloudy conditions, even in challenging scenarios. Lastly, the *F1-score* highlighted the baseline's strength, as it remained unbeaten, showcasing its ability to balance the correct identification of clouds while minimizing false positives.

The varied results reflect both the strengths and limitations of each approach:

- **Baseline Superiority**: The baseline model's performance underscores the effectiveness of architectures carefully tailored to the dataset's unique characteristics.
- **Koexai's Precision-Recall Tradeoff**: By excelling in recall, the Koexai team demonstrated the importance of architectural innovation for addressing false negatives. However, this came at the cost of reduced precision, suggesting areas for future improvement.
- **Challenges in Architectural Optimization**: The transformer-based solution proposed by Alpha Research Group showed the best results but struggled to outperform the baseline. This indicates the need for further refinement in handling the dataset's temporal and spatial complexities.

While the overall performance was impressive, a few limitations became apparent. For instance, differences in data preprocessing approaches, such as normalization and augmentation, had a noticeable impact on the model outcomes. Fine-tuning

these steps could lead to meaningful improvements. Additionally, the complexity of certain architectures, like transformers, introduced risks of overfitting, especially given the relatively small dataset size.

5.6.3 Discussion

The analysis of the results achieved by participating teams highlights the complexity of the proposed challenge and the diverse methodologies employed to tackle the binary classification of backscatter images. While all solutions demonstrated promising results, none managed to outperform the baseline metrics, underscoring the effectiveness of the baseline framework as a robust starting point.

Among the participants, the Koexai team stood out by leveraging a dual-stream architecture, which significantly improved recall. This result demonstrated the model's ability to accurately identify cloudy conditions, even in challenging scenarios. However, this improvement came at a slight cost to overall precision, indicating room for further optimization to balance these metrics. Similarly, the transformer-based approach proposed by the Alpha Research Group showcased the potential of attention mechanisms for analyzing complex data, effectively capturing both temporal and spatial features present in the backscatter profiles. Despite these strengths, the difficulty of surpassing the baseline highlights the challenges posed by the dataset's unique characteristics and the inherent complexity of backscatter data. Certain limitations were evident in the proposed solutions, such as the lack of ensemble approaches that could combine the strengths of different models and the absence of advanced strategies to handle class imbalances in the dataset.

The high accuracy and F1-scores observed across most solutions demonstrate the value of backscatter profiles for atmospheric analysis, with potential applications

in areas like environmental monitoring and precision agriculture. The ability to distinguish between clear and cloudy conditions with high accuracy reinforces the role of lidar-based data in complementing traditional weather prediction methods.

Looking ahead, future iterations of the challenge could explore multi-class classification to identify specific cloud types or incorporate complementary meteorological data, such as temperature or humidity, to enhance predictive capabilities. Furthermore, leveraging semi-supervised or unsupervised learning techniques could maximize the dataset's utility and further address the challenges of class imbalance.

Overall, the results validate the competition framework as a valuable benchmark for advancing innovation in atmospheric data analysis. At the same time, they provide insightful directions for future improvements and refinements, paving the way for more robust and versatile solutions.

5.7 Conclusion and Future Works

The Cloud Detection Challenge has demonstrated the potential of lidar-based ceilometer data for advancing binary cloud detection, emphasizing both the opportunities and challenges inherent in this domain. The outcomes showcase the capability of state-of-the-art deep learning methods to extract meaningful insights from backscatter profiles, which provide unique temporal and spatial details of atmospheric conditions. While the baseline model set a high standard for accuracy, F1-score, precision, and recall, the varied performances of the participant teams highlight the complexity of the task and the room for innovation.

Despite the notable successes, the challenge underscored areas for improvement. Differences in preprocessing strategies, risks of overfitting in complex architectures,

and the dataset's inherent characteristics all posed obstacles that prevented any solution from consistently outperforming the baseline. These findings suggest that future work should explore advanced preprocessing techniques, ensemble methods, and strategies to handle class imbalances effectively. Additionally, integrating complementary data sources, such as meteorological or atmospheric parameters, could further enhance model performance and applicability.

Looking forward, the insights gained from this challenge open the door to numerous exciting directions. Expanding the task to multi-class classification, incorporating additional environmental variables, and exploring semi-supervised learning could significantly enhance the versatility and robustness of these models. The advancements achieved through this competition not only contribute to the field of atmospheric monitoring but also provide a foundation for broader applications in environmental analysis and beyond. By building on these results, future efforts can continue to push the boundaries of innovation in cloud detection and related domains.

Part III

Transfer and Federated Learning Approaches

Chapter 6

Proposed Transfer Learning Approach

This chapter explores the potential of transfer learning to address the challenge of data scarcity in the context of atmospheric particle detection from ceilometer backscatter profiles. Due to the limited availability of annotated data from our specific ceilometer setup, we investigated how knowledge acquired from related domains could be reused to improve performance in similar but under-resourced scenarios. To this end, a novel aggregation-based transfer learning strategy was tested on standard benchmark datasets (MNIST [71] and SVHN [97]), simulating the adaptation process across different data distributions.

Although this method has not yet been applied directly to ceilometer data, it offers promising insights into how domain adaptation could enhance model robustness. Crucially, the knowledge acquired through cloud detection—understood as a prototypical case of atmospheric classification—may serve as a foundation for distinguishing among diverse types of suspended particles, including volcanic ash, dust, and sand. By transferring learned features across related atmospheric phenomena,

we aim to move towards a more comprehensive and fine-grained understanding of airborne particulate matter. Future work will focus on applying the proposed method directly to ceilometer-generated imagery, enabling improved generalisation across measurement sites and particle types.

6.1 Introduction

Deep Learning (DL) has demonstrated superior performance than traditional ML methods in a variety of tasks. This is due to being able to extract discriminative features from the data for the task at hand via end-to-end training. Such discriminative features are suitable for the dataset the network was trained on. However, a deep network will not perform as good as in a different dataset due to the *domain shift* (or dataset bias) [159].

A way to address the domain shift is via Transfer Learning (TL), where the information learnt by a trained network is (re)used in another context. Several approaches to transfer learning have been proposed in literature, such as sample reweighting [125], feature distributions minimisation [140, 79], distillation [54], and so on (for a recent survey on TL, please see [161]).

However, transfer learning techniques may be affected by catastrophic forgetting [50], where a network forgets the information learnt from a previous task when transferred to a new one. Furthermore, generally, transfer learning requires further training steps to accommodate for new data, even though the learnt task remains unchanged.

The benefit of transfer learning has been demonstrated extensively in the last years [146], even in distributed training scenario [25]. In this context, a central

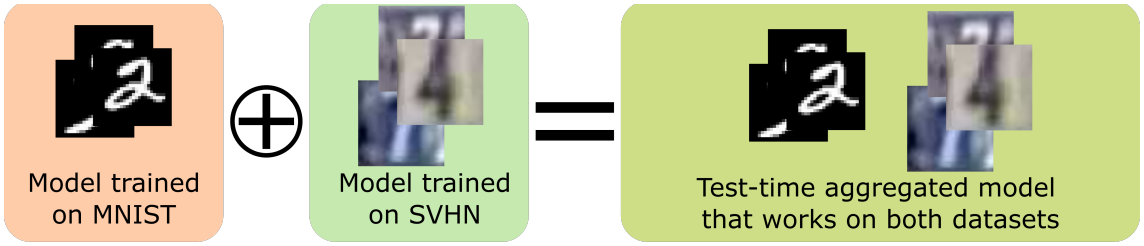


Figure 6.1: Pictorial representation of the proposed method that performs test-time neural network aggregation [20].

model is trained on several datasets that have never directly seen, as they are located in different machines (federated learning). However, this training paradigm raised another question: what if one (or more) datasets used to train the centrally trained model needs to be removed? Machine unlearning [48] is studied for several reasons, especially when sensible data are used (e.g., medical imaging). However, it is generally hard to selectively *scrub* the parameters of a model such that it cannot perform well on a portion of the dataset, whilst it retains comparable performance as before on the rest of the dataset.

In this work, a new proof-of-concept technique for transfer learning (TL) is proposed, which inherently allows for selective forgetting by aggregating network parameters *without any further training*. This approach can be applied to different datasets, provided they all share the same task. The proposed methodology is illustrated in Figure 6.1 and works as follows: a VGG-like [131] deep neural network is trained for each dataset, denoted as N_i for $i = 1, \ldots, n$, where n represents the number of datasets. Additionally, a VGG-like network, named N, is trained using all datasets as inputs. All networks are trained end-to-end with an *aggregation* regularizer, ensuring that the weights learned by N are aggregated from all the other networks N_i. This training paradigm ensures that the networks N_i also learn how

to be aggregated. Furthermore, by requiring that the aggregation function is invertible, the model inherently enables selective forgetting. In the experiments, $n = 2$ datasets were used, and the sum of weights was chosen as the aggregation function, which can easily be inverted via subtraction. This function is applied only to the parameters of the feature extractors. All networks trained within this end-to-end framework, including N^*, share the same classifier. Experimental results demonstrate that test-time network aggregation is feasible, outperforming the baseline.

The key contributions of our approach can be summarised as follows:

1. the *aggregation regulariser* during training is proposed;
2. network aggregation is achieved at test time (no further training is required);
3. our transfer learning technique does not suffer from catastrophic forgetting;
4. our approach can also be used for selective forgetting (assuming networks are aggregated via an invertible function).

6.2 Related Work

The aggregation of network parameters is a form of transfer learning. Typically, TL generally addresses a better initial and steeper growth performance [136] by reuse of the convolutional filter parameters of CNNs. For example, fine-tuning is the simplest way to achieve transfer learning: a model, pre-trained on a dataset, e.g. ImageNet [37], is used as starting point for other datasets and tasks [116]. Although intuitive and easy to do, fine-tuning typically underperforms wrt other transfer learning approaches [129, 52]. More sophisticated methods have been proposed [100], but several of them suffer from *negative transfer* [119, 102, 137, 144, 161]: the process

of transferring knowledge is harmful because the knowledge is not transferable across all the domains (in particular when the source and target datasets are not related).

Another issue affecting transfer learning approaches is *catastrophic forgetting*, where new knowledge permanently replaces information learnt from previous tasks [50]. In fact, several approaches to TL, such as *Batch Spectral Shrinkage* [24], attempts to solve such an issue. However, these approaches still rely on a training procedure to adapt to a new dataset (or task). However, the following question arises: is it possible to achieve transfer learning (TL) without catastrophic forgetting at test time? This is accomplished by aggregating the weights of the trained networks.

The idea of aggregating the parameters of deep neural networks is not new in the literature. A framework that aggregates knowledge from multiple models is the *Transfer-Incremental Mode Matching* (T-IMM) [45], which enables for adaptive merging of models. It is a re-interpretation of IMM [72], a work in the context of life-long learning aiming at the sequential aggregation of models retaining good performance on all the prior tasks, rather than on transfer learning. T-IMM belongs to the field of incremental learning, a subtly different area concerning lifelong learning, in which the parameters of the i-th model are used as initialisation for model $i+1$. More recently, Zoo-Tuning was proposed to adaptively aggregate multiple trained models [129]. To achieve network aggregation, the authors proposed the *AdaAgg* layer. However, this approach assumes that models are already pre-trained before being aggregated (involving a two-step learning). In our work, models are randomly initialised and then trained once end-to-end and simultaneously.

Lifelong (or continual) learning describes the scenario in which new tasks arrive sequentially and should be incorporated into the current model, retaining previous knowledge [103]. Approaches to lifelong learning are mainly aimed to mitigate

catastrophic forgetting [112, 111, 155]. According to Parisi *et al.* (2019), there are three main approaches to lifelong learning: (i) retraining with regularisation; (ii) network expansion; (iii) selective network retraining and expansion. In the first case, neural networks are retrained with constraints to prevent forgetting. Network expansion approaches perform architectural changes (e.g., adding neurons) to the network to add novel information. The last approaches update only a subset of neurons and allow expansion (if necessary). Our proposed method loosely follows the paradigm of regularisation approaches with an important difference: no retraining of the architecture is performed neither transfer learning nor selective forgetting.

Our approach to network aggregation inherently allows network decomposition for selective forgetting. Recently, several related works have focused on machine unlearning [47, 48]. Overall, these approaches assume that the portion of the dataset that the model should unlearn is given to a *scrub* function that aims to remove the information learnt from the dataset to be forgotten, impacting (although minimally) the performance of the scrubbed model on the rest of the dataset. Our approach is different: no data are required to be provided for selective forgetting. Instead, the aggregated model trained on two (or more) datasets can be changed by applying the inverse of the aggregation function (in our case, a simple subtraction).

6.3 Proposed Method

Figure 6.2 displays the proposed approach: the general idea is to aggregate the weights of two different neural networks trained on two different datasets (sharing the same underlying task). The ultimate goal is to obtain n individual networks N_i such that their composition $N_1 \oplus N_2 \oplus \ldots \oplus N_i \approx N^*$ (note that the operator $\oplus$

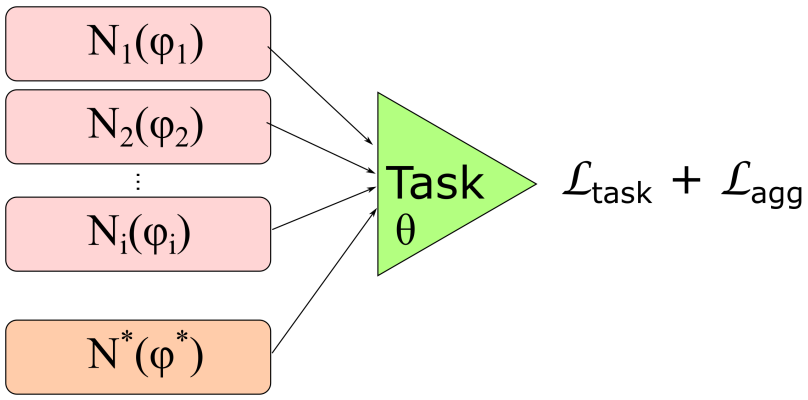


Figure 6.2: Graphical representation of the proposed method. Each network N_i is parametrised by a set of weights φ_i. The aggregated network N^* is parametrised by φ^*. There is no weight sharing between these networks. However, all the networks share the same task network (i.e., a classifier). The total objective function is given as a combination two loss functions: (i) task loss (i.e., cross-entropy); (ii) aggregation loss (see Section 6.3.3).

refers to a generic network aggregation operator, which details will be provided in Section 6.3.3). An individual network N_i will be referred to as the *dataset-specific network*, while N^* will be referred to as the *aggregated network*. As anticipated in Section 6.1, all these networks used as feature extractors share the same task network.

6.3.1 Task Network

As shown in Figure 6.2, the task network is shared across the aggregated and dataset-specific networks. This network is parametrised by the set of weights θ: the output provided by all of the N_i and N^* is used as input of the task network, and its output is the prediction (i.e., softmax activation in case of classification).

The task network is trained with a task-specific loss function as $\mathcal{L}_T(z, y; w)$, that takes the training data z (in form of representation) and target variables y as inputs,

and it is parametrised by a set of weights w (that includes θ and the parameters of the feature extractors). Cross-entropy loss for $\mathcal{L}_T$ was used in this work. For other tasks (i.e., regression), a different loss function may be used (e.g., mean squared error).

6.3.2 Dataset-Specific Network

Each dataset-specific network N_i functions as a feature extractor for the dataset it is trained on. A VGG-like network [131] was chosen for the experiments. Specifically, following the architectural adjustments made by others [81], a VGG-16 network with Group Normalisation [151] was used.[1]

Each network N_i is parametrised by the set of weights φ_i that is trained via standard supervised learning. In fact, each N_i is trained with a different label dataset; all of those datasets maintain the same underlying task. This means that each dataset $\mathcal{D}_i$ contains a set of input data $\mathcal{X}_i$, such that $x^{(i)} \in \mathcal{X}_i$, and a set of target values $\mathcal{Y}_i$, such that $y^{(i)} \in A$ (the set A is a generic set defined by the task, i.e. if the task is classification, A will contain all the possible classes).

For each of these networks, a specific loss function is used during training:

$$\mathcal{L}_{T_i}\left(x^{(i)}, y^{(i)}; \phi_i \cup \theta\right) = \mathcal{L}_T\left(N_i\left(x^{(i)}\right), y^{(i)}; \phi_i \cup \theta\right). \tag{6.1}$$

6.3.3 Aggregated Network

The aggregated network is similar to the dataset-specific networks: it shares the same architecture but not the weights. In fact, this network is parametrised by the

[1] Attempts were also made with Batch Normalisation [59] and no normalisation, but these approaches did not yield successful results.

set of weights ϕ^*.

The aggregated network is trained so that its weights can be expressed as the sum of the dataset-specific networks. To achieve this, the *aggregation* regularizer was proposed. Each network N_i consists of L_i layers, and each layer ℓ is parametrized by weights $W_i^\ell \in \varphi_i$.[2] Similarly, the aggregated network N^* includes several layers L, each parametrized by W^ℓ. It is important to note that all feature extractors share the same architecture, meaning that $L = L_1 = L_2 = \ldots = L_n$.

During training, it is desired that:

$$W^\ell = W_1^\ell \oplus W_2^\ell \oplus \ldots \oplus W_n^\ell, \tag{6.2}$$

i.e., the weights at layer ℓ in the aggregated network should be equal to the aggregation of the corresponding layer weights in the dataset-specific networks. This constraint is reformulated as a regulariser during training.. Assuming that $\ominus$ is the inverse operator of $\oplus$, the aggregation regulariser is then expressed as:

$$\mathcal{L}_{agg}(\Phi) = \sum_{\ell=1}^{L} W^\ell \ominus \left[W_1^\ell \oplus W_2^\ell \oplus \ldots \oplus W_n^\ell\right], \tag{6.3}$$

where $\Phi = \phi^* \cup (\bigcup_{i=1}^n \phi_i)$ is the set of all the weights in all the feature extractors.[3] Although the networks learn a non-linear mapping w.r.t the task, the aggregation regulariser in Equation (6.3) learns the weights Φ such that the network aggregation can be performed with a linear operation (assuming that $\oplus$ is linear).

The aggregated network takes all the input data that are used for each dataset-specific network $\mathcal{D} = \bigcup_{i=1}^n \mathcal{D}_i$ and it is trained in a supervised manner w.r.t. the

[2]Certain types of layers, such as convolutional layers, may be parametrized by multiple weights, such as kernels and biases. For clarity, these weights are incorporated within W_i^ℓ.

[3]Similarly as in [45], Only the weights of convolutional layers are aggregated.

task $\mathcal{L}_T$ as follows:

$$\mathcal{L}_{T^*}(x, y; \phi^* \cup \theta) = \mathcal{L}_T(N^*(x), y, \phi^* \cup \theta), \tag{6.4}$$

where $(x, y) \in \mathcal{D}$, i.e. inputs and labels are taken from all the datasets used to train the dataset-specific networks.

6.3.4 Objective Function

As shown in Figure 6.2, the objective functions used to train our model is the following:

$$J(x, y; \Theta) = \mathcal{L}_{task} + \mathcal{L}_{agg}, \tag{6.5}$$

where $\Theta = \Phi \cup \theta$ is the set of all the parameters in the network. the loss function $\mathcal{L}_{task}$ is given as the sum of all the task-specific loss functions expressed in Equation (6.1) and Equation (6.4):

$$\begin{aligned}\mathcal{L}_{task}(x, y; \Theta) = \mathcal{L}_{T^*}(x, y; \phi^* \cup \theta) \\ + \sum_{i=1}^{n} \mathcal{L}_{T_i}\left(x^{(i)}, y^{(i)}; \phi_i \cup \theta\right).\end{aligned}$$

After training, there is no guarantee that the regularizer in Equation (6.3) will ensure that Equation (6.2) is satisfied. However, the optimization of Equation (6.5) ensures that $W^\ell \approx W_1^\ell \oplus W_2^\ell \oplus \ldots \oplus W_n^\ell$. Therefore, N can be retrieved by aggregating all the weights trained for each dataset-specific network N_i — this model will be referred to as $\hat{N}$, such that $\hat{N}^* \approx N^*$.

One of the datasets can be selectively forgotten from $\hat{N}$ by applying a simple arithmetic operation. To remove the k-th dataset, the operation $\hat{N}^{\ominus}N_k$ can be performed, without the need for any further training or adaptation steps.

6.3.5 Implementation Details

The number of datasets, and consequently the number of dataset-specific networks, was set to $n = 2$. This choice allowed for the demonstration of the effectiveness of the approach and the establishment of a baseline. The number of groups for Group Normalisation was set to 32. The sum was chosen as the aggregation operator for the following reasons: (i) it has an inverse, namely subtraction, and (ii) it is differentiable. The task-specific loss function was defined as cross-entropy loss, as the entire network is trained for a classification task. Stochastic Gradient Descent (SGD) was used as the optimizer, with a learning rate of $\eta = 0.01$. The baseline model was trained for 20 epochs, while training for the proposed method lasted 200 epochs. The approach was implemented in PyTorch [104] on Google Colaboratory.

6.4 Experimental results

Dataset MNIST [71] was used as $\mathcal{D}_1$ and SVHN format 2 [97] as $\mathcal{D}_2$. MNIST contains $60,000$ binary images of size 28×28 for training and $10,000$ images for testing. SVHN contains $73,257$ color images of size 32×32 for training and $26,032$ for testing. These two datasets were selected for the following reasons: (i) they are designed for the same classification task (10-class), and (ii) the data are drawn from different distributions.

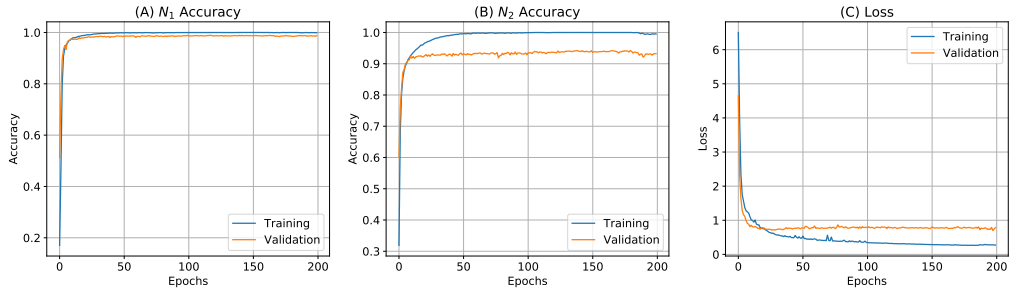


Figure 6.3: Training and validation accuracies and losses of proposed method. (A) N_1 training and validation accuracies; (B) N_2 training and validation accuracies; (C) Total training and validation loss.

Preprocessing In order to use the same architecture for both datasets, MNIST images were rescaled to 32×32. The SVHN images were converted to grayscale. For data augmentation, random horizontal flips were applied with a probability of 50%.

Baseline Our method was compared with a standard VGG-16 network with Batch Normalization [59]. The following baseline experiments were conducted:

1. trained it only on MNIST – following the notation adopted in this work, this trained network was called N_1;
2. trained it only on SVHN – named N_2;
3. trained it on both (N^*);
4. The weights trained on N_1 and N_2 were taken to perform $N_1 \oplus N_2$ at test time.

Experimental results are shown in Table 7.2.

	Trained on		Tested on		
	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_1 \cup \mathcal{D}_2$
Baseline					
N_1		–	99.04%	8.67%	33.75%
N_2	–		57.63%	90.29%	80.78%
N^*			98.73%	90.58%	92.85%
$N_1 \oplus N_2$			9.75%	7.59%	8.19%
Proposed Method					
N_1		–	98.90%	41.45%	57.35%
N_2	–		45.30%	92.41%	79.31%
N^*			98.40%	86.47%	89.77%
$N_1 \oplus N_2$			96.41%	68.03%	75.88%

Table 6.1: Testing performance of the proposed method compared to the baseline performance. $\mathcal{D}_1$ indicates MNIST; $\mathcal{D}_2$ indicates SVHN. The models obtained via aggregation (i.e., $N_1 \oplus N_2$) are obtained at test time by aggregating the weights of the networks.

6.5 Discussion

Our purpose is to demonstrate that the performance of our aggregated network $N_1 \oplus N_2$ is better than the baseline. Overall, our method achieves comparable performance with the baseline for individual tasks (i.e., N_1 and N_2). However, there is a slight loss in performance in N^*, that is, the network trained on both MNIST and SVHN, with our method. The baseline achieves approx. 92% accuracy, whereas N^* trained with our method achieves approx. 89%.

Although this minor performance reduction, our method achieves high performance with test-time weight aggregation. After N_1 and N_2 are trained with the baseline and our method, weights are aggregated by applying the $\oplus$ operator. Table 7.2 clearly shows that our training procedure outperforms the baseline (8% vs 75% testing accuracy). This demonstrates that a traditional training of two networks cannot be aggregated, leading to catastrophic forgetting on both datasets. Our training approach with Equation (6.3) enables the networks to explicitly learn

	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_1 \cup \mathcal{D}_2$
Commutativity			
$N_1 \oplus N_2$	96.41%	68.03%	75.88%
$N_2 \oplus N_1$	96.41%	68.03%	75.88%
Selective forgetting			
Gray N_1	98.90%	41.45%	57.35%
$(N_1 \oplus N_2) \ominus N_2$	98.55%	47.90%	61.92%
$(N_2 \oplus N_1) \ominus N_2$	98.55%	47.90%	61.92%
Gray N_2	45.30%	92.41%	79.31%
$(N_1 \oplus N_2) \ominus N_1$	34.67%	90.88%	75.28%
$(N_2 \oplus N_1) \ominus N_1$	34.67%	90.88%	75.28%

Table 6.2: Commutativity and Selective forgetting testing results. The training is performed in both $\mathcal{D}_1$ and $\mathcal{D}_2$. The two datasets are the same as in Table 6.1. Highlighted rows are copied from Table 6.1 to ease comparison.

an aggregation operation that can be reproduced at test time.

Ideally, the performance of $N_1 \oplus N_2$ should be as close as possible to N^*. As shown in the last two lines of Table 7.2, there is an approximate loss of 14% accuracy. Several reasons for this gap in accuracy are hypothesized: (i) our method may require more training time; (ii) Group Normalisation may be having an impact at test time (as specified in Section 6.3.3, only the weights of convolutional layers are aggregated); (iii) use of Weight Standardisation (WS) can improve performance [108, 81].

Regarding training time, the training and validation accuracies and losses of our method are plotted in Figure 6.3. It can be observed that 50 epochs are sufficient for both datasets. However, experimental results indicate that additional training time leads to better performance in the aggregated model. It is hypothesized that the optimization of Equation (6.3) may require more time to learn more aggregable models. Concerning Weight Standardization, it is suggested that improved performance could be achieved by changing the aggregation function from the sum to the mean. Otherwise, the aggregated weights may no longer be zero-centered.

6.5.1 Commutativity

The aim here is to demonstrate whether our method is commutative: does the performance of $N_1 \oplus N_2$ match the performance of $N_2 \oplus N_1$? Theoretically, commutativity should be strictly related to the $\oplus$ operator. However, this is not strictly guaranteed in our framework, as we only aggregate the convolution weights in the networks N_i (see Section 6.3.3). Group Normalization layers also include learnable parameters that are not involved in the network aggregation. To verify whether our method is commutative, the operation $N_2 \oplus N_1$ was also performed, and the results are reported in Table 6.2. It can be observed that the performance in both scenarios is identical. Therefore, it can be concluded that our approach is commutative.

6.5.2 Selective Forgetting

For the same reasons outlined in Section 6.5.1, an experimental evaluation is conducted to determine whether selective forgetting is possible with our method. Due to the presence of Group Normalization layers, $(N_1 \oplus N_2) \ominus N_1 \approx N_2$, meaning that by removing the contribution of N_1, the resulting network does not exactly match N_2 (and vice versa). This led to the following question: does the network obtained by $(N_1 \oplus N_2) \ominus N_1$ perform as well as N_2? The experimental results of selective forgetting are presented in Table 6.2.

Forgetting SVHN The weights of N_2 were removed from the aggregated networks (considering both $N_1 \oplus N_2$ and $N_2 \oplus N_1$ as aggregated networks), and the SVHN testing set was provided to this new network. The testing accuracy achieved was 47.90%, compared to 41.45% for N_1. Therefore, the resulting network does exhibit selective forgetting of SVHN, though not completely, with an approximate performance increase of +6%.

Forgetting MNIST A similar experiment was performed by removing the weights of N_1 from the aggregated networks. Differently than before, the testing accuracy of the resulting network is 34.67%, compared with 45.30%. This experimentally demonstrates that our method has completely forgotten the information learnt from the MNIST dataset.

Retained information In the two previous experiments, it was shown that the resulting network had forgotten information from either of the two datasets. However, it is necessary to verify whether the network can still perform well on the other dataset. Overall, the performance on MNIST dataset is very similar (from 98.90% to 98.55%), whereas in the case of SVHN there is approx 2% loss of performance (from 92.41% to 90.88%) – although the overall testing error is above 90%. This also demonstrates that the proposed method retains information from both tasks with a loss in performance up to 2%.

6.6 Conclusion and Future Works

Our training method enables for network aggregation at test time, i.e. the weights of two networks (trained on two different datasets) are aggregated together, such that the resulting network can work on both datasets without any further training/adaptation step.

This is achieved by introducing an *aggregation regularizer*, which enables the networks to learn the aggregation operation within an end-to-end training framework. The sum was used as the aggregation operator due to its invertibility and differentiability. VGG-like architectures were employed as feature extractors, with Group Normalization used in place of Batch Normalization.

Our experimental results demonstrated that the proposed approach allows for test-time transfer learning without any further training steps. Furthermore, it was showed that our training procedure is commutative: the aggregated network $N_1 \oplus N_2$ obtains the same performance of $N_2 \oplus N_1$. Moreover, it was demonstrated that our method allows for selective forgetting (at the cost of up to 2% testing performance).

The proposed method has some limitations: (i) it requires that all the networks involved in the training share the same architecture; (ii) the selective forgetting does not allow to forget a subset of the dataset; (iii) it was evaluated on just two benchmark datasets (although the proposed framework can easily accommodate for multiple datasets). As future work, the study will generalise this approach exploring the training with N_i deep neural networks, for $i = 1, ..., n$, with n being the number of datasets, in a federated learning scenario, using different type of data such as the ceilometers one.

Chapter 7

Proposed Federated Learning Approach

This chapter introduces a federated learning framework designed to enable collaborative model training among multiple ceilometer data owners while preserving data privacy and security. Given the sensitive nature and limited accessibility of raw atmospheric measurements, the proposed method ensures that no raw data is exchanged between sites. Instead, local models contribute to the training process via a shared encoder structure and aggregated parameter updates. The approach was tested on multiple real-world ceilometer datasets, demonstrating its capability to maintain high performance while adhering to privacy-preserving constraints—offering a promising solution for future collaborative monitoring networks.

7.1 Introduction

In recent years, the massive adoption of data-driven technologies has required effective artificial intelligence methods addressing the increasing privacy requirements, such as the European GDPR regulation. Federated Learning (FL) [87] has emerged

as a promising approach for dealing with private and sensitive data to train machine learning models. In a typical federated scenario, there are two entities: a server and many different clients. By aggregating locally trained deep learning (DL) models sent by the clients, the server produces a global model without sharing locally-stored data in each of the clients. The current state-of-the-art FL algorithm is FedAvg [86], which aggregates the locally trained models by averaging their parameters. Privacy preservation is achieved by keeping data locally. Built on the assumption that clients hold labelled data, most of the FL literature focuses on supervised learning problems. However, in real-world scenarios, as the parties involved may not have sufficient domain expertise or resources, the data may also be unlabelled. Therefore, the absence of annotated labels currently represents a challenge in FL.

This research proposes FedRec, an FL pipeline for image classification tasks, in which clients without ground-truth labels assist client training on annotated data. In our scenario, some clients are trained on labelled data, while others are trained on unlabelled data in an unsupervised fashion. It was used an encoder-decoder model performing image reconstruction in those clients where labels are missing or lacking, in which the encoder architecture matches the feature extractor of those clients trained in a supervised fashion. This results in a mechanism of data augmentation for the labelled data, as the aggregation involves more parties contributing to the extraction of image features. At this purpose, we conducted experiments on five datasets collected using appropriate instruments, the ceilometers, across Italy. All the datasets contain images reconstructed from measurements taken by ceilometers, which, by counting the number of photons reflected by particles in the atmosphere, are able to estimate the height and presence of clouds in the sky.

Our contribution can be summarised as follows: (1) proposing FedRec, a federated semi-supervised learning (FSSL) approach in which unlabelled data are used in conjunction with labelled data to capture features serving as data augmentation for the latent space encoded from fully labelled data. We aim to improve the performance of the supervised model by using additional data from different locations in Italy. (2) Comparing with the simplest approach based on FedAvg and labelled data and show its limitations. (3) Demonstrating the efficacy of FedRec through extensive experiments on novel datasets specially collected to support researchers in the field of deep learning applied to environmental phenomena, showing that we outperform the traditional method.

7.2 Related Work

The primary goal of FL is to train a global inference model while keeping data scattered across different silos, thus preserving privacy. While most of the FL literature focuses on supervised tasks, some recent works adopt SSL techniques for exploiting the increasing volume of unlabelled data. In SemiFL [39], clients have completely unlabelled data, while the server holds a small amount of annotated samples. SemiFL proposes alternate training to fine-tune the global model with labelled data and generate pseudo-labels with the global model.

FedMatch [60] introduces an inter-client consistency loss that aims to maximise the agreement between the models trained at different clients. In particular, in FedMatch, each client samples the top-k nearest clients and ensures consistency by regularising the local model output with the top-k client models. Additionally, FedMatch adopts the decomposition of model parameters for disjoint learning on

labelled and unlabeled data.

Another work addressing the increasing communication and computational cost due to the inter-client knowledge sharing based on model weights is ProtoFSSL [65], an approach based on prototype learning that exchanges lightweight prototypes between clients. Each federation client creates pseudo-labels based on shared prototypes to compute the loss. A prototype-based inter-client knowledge sharing significantly reduces both communication and computation costs.

7.3 Methodology

In contrast to previous studies based on disjoint learning or prototype learning, a method for FSSL was proposed, based on parameter exchange leveraging both labelled and unlabelled data. In our approach, some clients hold only unlabelled data, while others hold ground-truth labels. Depending on the availability of annotated data, each client will solve a different task. Some clients will undertake a classification task, while others will tackle an unsupervised learning problem. Specifically, in FedRec, unlabelled data are utilised to solve an image reconstruction task with an encoder-decoder architecture. Our method involves aggregating the weights of model architectural components that perfectly match between supervised and unsupervised clients. A recent work [17] on vertical FL shows that aggregating only identical architectural parts of different models is a promising approach. Since our goal is to augment the supervised task with unlabelled data from other clients, we enforced the encoder architecture to align with the feature extractor of the image classification network. Although the different tasks, the encoder should extract image features serving as a data augmentation technique for the supervised clients, thus

resulting in an increased FL generalization property. As a result, the weights of the encoder and the feature extractors of the classification models are averaged. Finally, we perform parameter aggregation of the fully connected layers of the classification models.

7.4 Experiments

Our experiments aim to investigate the classification performance of our proposed federated model trained by aggregating features extracted from both labelled and unlabelled data. We report the results of a typical isolated scenario in which a model is trained on data from a single institution, a centralised experiment in which the data are gathered in a single data lake, a naive federated approach based on FedAvg and only labelled data, and our proposed method.

Testbed setup: Our experiments were conducted in a simulated federation utilising an Intel® Xeon® processor (Skylake, IBRS, eight sockets of one core) and one Tesla T4 GPU. The federation comprised one server, two clients holding labelled data, and three clients holding unlabelled data. The baseline experiments, including the isolate, centralised, and FedAvg approach on the two clients with labelled data, were run on the same dedicated machine. For reproducibility purposes, the code and the data used for our experiments are available at the following link: https://github.com/CasellaJr/FedRec.

Datasets: All the datasets were obtained through measurement reconstructions of appropriate instruments, such as the ceilometers. A ceilometer is a measuring device mostly used in meteorology that can detect the height of a cloud base by emitting a modulated light beam directed to the sky. This makes it possible not only to

Location	Latitude	Longitude	Samples	Positive (%)	Period
S.G.L.P.	37d 34' 44" N	15d 06' 11" E	1568	1050 (66.96%)	01/01/23 - 14/03/23
C.G.	37d 34' 16" N	12d 39' 35" E	2193	890 (40.58%)	01/06/23 - 31/08/23
Roma	41d 50' 32" N	12d 38' 50" E	2208	N.A.	01/07/23 - 30/09/23
Taranto	40d 29' 37" N	7d 13' 01" E	2160	N.A.	01/01/21 - 31/03/21
Aosta	45d 44' 32" N	7d 21' 24" E	2180	N.A.	01/07/21 - 30/09/23

Table 7.1: Statistics of the datasets.

recognise the clouds in the sky but also to determine their height. For this purpose [1] ALICEnet[2] was used, a cooperative network of lidar-ceilometers coordinated by CNR-ISAC and operated in collaboration with other Italian research institutions, universities, and environmental agencies. We used five locations around Italy: San Giovanni La Punta (S.G.L.P.), Capo Granitola (C.G), Roma, Taranto, and Aosta. Table 7.1 summarizes the statistics of the datasets. S.G.L.P. and C.G. datasets contain annotated data, while the latter are unlabeled. The first dataset, S.G.L.P., has previously been publicly released [30] and it was labelled using the output of a *Weather Research and Forecasting* (WRF) model specially set up to produce weather simulations at the coordinates of the corresponding ceilometer. The authors labelled the C.G. dataset manually. Both S.G.L.P and C.G. are used to solve the cloud detection binary classification task. In cloud detection, we have a positive label if a cloud is detected and a negative otherwise. S.G.L.P. and C.G. clients split their data into training (70%) and testing (30%) data. As we were interested only in improving the classification performance, the clients holding unlabeled data used the entire set as training data to increase the generalisability of the extracted features.

Models: We employed a ResNet-18 as a feature extractor on S.G.L.P. and C.G., trained by minimising the cross-entropy loss with mini-batch gradient descent using the SGD optimizer with a learning rate of 10^{-4}, momentum 0.8 and weight decay

[1] thanks to the company EHT S.C.p.A.

[2] https://www.alice-net.eu/ - last accessed: September 2024

10^{-5}. The local batch size was 8. We employed an encoder-decoder architecture to solve the image reconstruction task in Roma, Taranto, and Aosta. A ResNet-18 serves as the backbone of the encoder part for capturing image features and encoding them into the latent space. The decoder, mapping the encoded representation back to the original feature space, is made of four convolutional and upsampling layers. This architecture was trained by minimising the MSE with mini-batch gradient descent using the SGD optimizer with a learning rate of 10^{-4}, momentum 0.8 and weight decay 10^{-5}. The local batch size was 8. As evaluation metrics, we focus on the accuracy and F1-score.

Discussion: Table 7.2 shows the results of our experiments in terms of accuracy and F1-score. Looking at the accuracies, FedRec performs slightly better than FedAvg with only labelled data on the S.G.L.P dataset, while it outperforms the naive approach on the C.G. data. Federated F1-scores are comparable between the two datasets and methods. FedRec achieves better results than the baseline on the C.G. dataset, while it is beaten on the S.G.L.P. data, even if scores are really closer to each other. Isolated accuracies show that the classifier correctly discriminates the majority of samples. However, accuracy alone may not adequately capture the classifier's performance, especially in the presence of class imbalance, as it will tend to favour the majority class. F1-score, being the harmonic mean of precision and recall, is a better metric for evaluating performance on unbalanced datasets. Isolated F1-scores show a moderate balance between precision and recall. This means that while the classifier identifies some true positives, it may also produce a notable number of false positives, or false negatives, or both. Moreover, the similarity between the class imbalance percentage and the isolated F1-scores on both datasets suggests that the classifier's performance is comparable to randomly guessing the

positive class. Both the naive federated and FedRec approaches outperform the isolated F1-scores. We hypothesise that this is because the federated model has a higher ability to generalise, due to a greater number of image features it was trained on. Results that were obtained in a centralised setting, in which both the annotated datasets were aggregated in a single data lake, show that both accuracy and F1-score are a weighted mean (with a bigger impact of C.G. due to the greater amount of samples) of the isolated performance. In particular, in the centralized scenario, we obtained an accuracy of $0.933 \pm .001$ and an F1-score of $0.517 \pm .00$. We hypothesize that, while counter-intuitive, FL benefits from its intrinsic alternate training nature, leading to a better feature extraction process.

We did not report the results of the image reconstruction task on the unlabelled clients, as we were only interested in the supervised performance. We used the unlabelled data as a data augmentation technique to extract a broader and more general set of image features and assist in the supervised training.

Finally, Table 7.3 reports the global model results after fine-tuning on the local dataset for one epoch. Although the model's accuracy benefits from a fine-tuning iteration on the local dataset, there is a drop in F1-scores. In particular, the performance goes down to the isolated case. We hypothesise that this occurs due to the small size of the datasets. Even though the aggregated model benefits from a more accurate feature extraction process, an epoch of fine-tuning on the local datasets leads to overfitting, probably due to the small dataset size.

Dataset	Accuracy			F1-score		
	Isolated	Federated		Isolated	Federated	
		Naive	FedRec		Naive	FedRec
S.G.L.P.	$.742 \pm .012$	$.689 \pm .005$	$.691 \pm .038$	$.668 \pm .0$	$.808 \pm .002$	$.805 \pm .001$
C.G.	$.988 \pm .001$	$.427 \pm .052$	$.587 \pm .093$	$.405 \pm .0$	$.576 \pm .0$	$.581 \pm .001$

Table 7.2: Comparison between our proposed method, centralised baselines and state-of-the-art methods. Results (mean) are obtained with five averaged runs.

Dataset	Accuracy		F1-score	
	FedRec	FedRec fine-tuning	FedRec	FedRec fine-tuning
S.G.L.P.	$.691 \pm .038$	$.848 \pm .009$	$.808 \pm .002$	$.668 \pm .0$
C. G.	$.587 \pm .093$	$.984 \pm .011$	$.576 \pm .0$	$.405 \pm .0$

Table 7.3: Global model results after fine-tuning on the local dataset for one epoch. Results (mean) are obtained with five averaged runs.

7.5 Conclusion and Future Works

This work proposes FedRec, a method for FSSL leveraging unlabeled data to help supervised training on annotated data. In particular, clients with solely unlabeled data use an encoder-decoder architecture for doing image reconstruction, in which the encoder part matches the feature extractor of a classification model. The trained feature extractors are aggregated via weight averaging, as well as the fully connected layers of the classification models. We show the effectiveness of our method by comparing its accuracy and F1-score performance against the isolated, centralised and federated baseline based on FedAvg of just the supervised models.

Future work will focus on investigating potential improvements in the computational costs of our method. Specifically, while reducing communication costs by sharing only a smaller subset of layers from the local models could be beneficial, it may lead to an increase in computation time due to a larger volume of data and more clients participating in the federation. A potential strategy to address this challenge could involve applying an early stopping technique on the unlabeled data. Finally,

future research will explore the possibility of enhancing learning performance by determining the optimal model for both supervised and unsupervised tasks.

Part IV

Conclusions and Future Works

Chapter 8

Final Discussion and Future Works

The concluding chapter of this thesis is dedicated to summarizing the major contributions of the research, reflecting on the implications of the findings, and proposing potential avenues for future work. This chapter serves as a synthesis of the work presented, linking the theoretical advancements with practical applications and setting the stage for future explorations in the field of atmospheric science and artificial intelligence (AI).

8.1 Summary of Contributions

The primary achievement of this research is the design and implementation of a Multimodal AI Engine that leverages state-of-the-art deep learning and computer vision methodologies for enhanced atmospheric particle detection. The work has made several significant contributions to the scientific community:

- **Novel Datasets and Preprocessing**: The research introduced innovative datasets derived from raw lidar-based ceilometer backscatter data. These

datasets were processed into high-resolution, time-indexed images that address challenges related to data scarcity and variability in atmospheric studies.

- **Benchmarking Deep Learning Architectures**: An extensive comparative analysis of various deep learning models—including Convolutional Neural Networks (CNNs) and Vision Transformers (ViT)—was conducted. This evaluation provided insights into performance trade-offs among different configurations, optimizers, and hyperparameters, thereby identifying the most effective architectures for detecting clouds and particulate matter.

- **Integration of Multimodal and Federated Learning Approaches**: By incorporating transfer learning and federated learning paradigms, the study developed a scalable approach that aggregates knowledge from heterogeneous data sources while preserving data privacy. This integration enhances model generalizability and resilience in real-world settings.

- **Practical Applications in Environmental Monitoring**: The AI Engine has demonstrated practical utility in critical areas such as air traffic management during volcanic eruptions and optimizing solar panel performance under adverse weather conditions. The successful application of these technologies underscores their potential for broad environmental and meteorological use.

- **Scientific Impact**: The findings have been disseminated through multiple peer-reviewed publications and conference presentations, thereby contributing to ongoing academic discourse in the fields of deep learning, computer vision, and atmospheric science.

8.2 Limitations of the Study

Although the research makes several noteworthy contributions, it is important to acknowledge certain limitations. First, the quality and heterogeneity of the data remain critical challenges. The reliance on ceilometer sensor outputs, while innovative, introduces an inherent degree of noise and variability that may affect the precision of model predictions. Additionally, the AI Engine's performance is sensitive to the representativeness of the training data, and its generalizability to regions with different atmospheric conditions requires further validation. Another limitation concerns the computational complexity of the deep learning architectures employed. Their resource-intensive nature could impede real-time deployment of the system, particularly in settings with constrained computational resources. Finally, like many deep learning models, the current system suffers from a lack of transparency in its decision-making process, limiting its interpretability and potentially hindering user trust in high-stakes operational contexts.

8.3 Implications of Findings

The implications of this research are significant and multifaceted. The demonstrated effectiveness of integrating deep learning with multimodal data not only improves atmospheric particle detection but also provides a robust framework for environmental monitoring in challenging conditions. The ability to accurately detect and classify atmospheric particles has practical benefits for air traffic management, renewable energy optimization, and disaster response. Moreover, the application of federated learning approaches within this study emphasizes the potential for preserving data privacy while leveraging diverse, distributed datasets—a feature increasingly critical

in today's data-driven research landscape. Collectively, these findings suggest that advanced AI methodologies can serve as powerful tools for addressing complex environmental challenges, paving the way for more adaptive, transparent, and efficient monitoring systems in the future.

8.4 Future Works

Building on the successes and limitations of the current study, several future research directions are proposed:

- **Enhanced Data Fusion and Multimodal Integration**: Future work should explore more advanced methods for fusing heterogeneous data sources, such as integrating satellite imagery, radar data, and ground-based sensor networks. This could lead to a more comprehensive understanding of atmospheric phenomena.

- **Optimization for Real-Time Deployment**: Efforts should be directed toward reducing computational complexity through the development of lightweight model architectures and more efficient inference algorithms. Such advancements would facilitate the deployment of the AI Engine in real-time operational environments.

- **Mitigating Domain Shift**: Investigating advanced transfer learning strategies that minimize negative transfer and improve domain adaptation is essential. Future studies might focus on developing dynamic adjustment techniques to better handle shifts between training and operational data distributions.

- **Scalability and Decentralized Learning**: Expanding the federated learning framework to incorporate a larger number of diverse data sources is another promising direction. Exploring asynchronous federated learning methods and adaptive communication protocols could improve scalability and reduce latency.

- **Integration with Policy and Decision-Making Frameworks**: Future work could also focus on developing interfaces and protocols that allow the AI Engine to interact seamlessly with decision support systems used by environmental agencies and governmental bodies, thereby enhancing its practical impact.

In conclusion, this thesis has advanced the state-of-the-art in atmospheric particle detection through the innovative integration of deep learning and computer vision techniques. The novel datasets, benchmarking, and incorporation of both multimodal and federated learning approaches have contributed significantly to the fields of environmental monitoring and AI-driven decision support. While certain limitations persist, they offer clear directions for future research that promise to further enhance model performance, interpretability, and scalability. The broader implications of these findings underscore the transformative potential of AI in mitigating the adverse effects of atmospheric phenomena and supporting informed decision-making in the face of environmental challenges.

Bibliography

[1] Luqman Ali et al. "Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures". In: *Sensors* 21.5 (2021), p. 1688.

[2] Naomi S Altman. "An introduction to kernel and nearest-neighbor nonparametric regression". In: *The American Statistician* 46.3 (1992), pp. 175–185.

[3] Ni An et al. "A cloud detection algorithm for early morning observations from the FY-3E satellite". In: *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023), pp. 1–15.

[4] S. H. Arun et al. "Fog/low clouds detection over the Delhi Earth Station using the Ceilometer and the INSAT-3D/3DR satellite data". en. In: *International Journal of Remote Sensing* 39.12 (June 2018), pp. 4130–4144. ISSN: 0143-1161, 1366-5901. DOI: 10.1080/01431161.2018.1454624. URL: https://www.tandfonline.com/doi/full/10.1080/01431161.2018.1454624 (visited on 08/27/2023).

[5] Mehran Asadi and Manfred Huber. "Effective Control Knowledge Transfer through Learning Skill and Representation Hierarchies." In: *IJCAI*. Vol. 7. Citeseer. 2007, pp. 2054–2059.

[6] Andrew G Barto. "Reinforcement learning". In: *Neural systems for control*. Elsevier, 1997, pp. 7–30.

[7] Juraj Bartok et al. "Data mining and integration for predicting significant meteorological phenomena". en. In: *Procedia Computer Science* 1.1 (May 2010), pp. 37–46. ISSN: 18770509. DOI: 10.1016/j.procs.2010.04.006. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877050910000074 (visited on 08/27/2023).

[8] Jonathan Baxter. "A model of inductive bias learning". In: *Journal of artificial intelligence research* 12 (2000), pp. 149–198.

[9] Annachiara Bellini et al. "Alicenet–An Italian network of Automated Lidar-Ceilometers for 4D aerosol monitoring: infrastructure, data processing, and applications". In: *EGUsphere* 2024 (2024), pp. 1–47.

[10] Shai Ben-David and Reba Schuller. "Exploiting task relatedness for multiple task learning". In: *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*. Springer. 2003, pp. 567–580.

[11] Anil Bhattacharyya. "On a measure of divergence between two statistical populations defined by their probability distribution". In: *Bulletin of the Calcutta Mathematical Society* 35 (1943), pp. 99–110.

[12] Christopher M Bishop and Hugh Bishop. *Deep learning: Foundations and concepts*. Springer Nature, 2023.

[13] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[14] Leo Breiman et al. *Classification and regression trees*. Routledge, 2017.

[15] James L Carroll and Kevin Seppi. "Task similarity measures for transfer in reinforcement learning task libraries". In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 2. IEEE. 2005, pp. 803–808.

[16] Rich Caruana. "Multitask learning". In: *Machine learning* 28 (1997), pp. 41–75.

[17] Bruno Casella and Samuele Fonio. "Architecture-Based FedAvg for Vertical Federated Learning". In: *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing.* , Taormina (Messina), Italy, 2024. ISBN: 9798400702341. DOI: 10.1145/3603166.3632559. URL: https://doi.org/10.1145/3603166.3632559.

[18] Bruno Casella et al. "Experimenting with normalization layers in federated learning on non-iid scenarios". In: *IEEE Access* (2024).

[19] Bruno Casella et al. "Federated Learning in a Semi-Supervised Environment for Earth Observation Data". In: *ESANN 2024 Proceedings-32th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning.* Michel Verlesian. 2024, pp. 93–98.

[20] Bruno Casella et al. "Transfer Learning via Test-Time Neural Networks Aggregation". en. In: *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications.* arXiv:2206.13399 [cs]. 2022, pp. 642–649. DOI: 10.5220/0010907900003124. URL: http://arxiv.org/abs/2206.13399 (visited on 05/05/2024).

[21] M Emre Celebi and Kemal Aydin. *Unsupervised learning algorithms.* Vol. 1. Springer, 2016.

[22] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. "Bias in machine learning software: Why? how? what to do?" In: *Proceedings of the 29th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*. 2021, pp. 429–440.

[23] Tsuhan Chen. "From low-level features to high-level semantics: are we bridging the gap?" In: *Seventh IEEE International Symposium on Multimedia (ISM'05)*. IEEE Computer Society. 2005, pp. 179–179.

[24] Xinyang Chen et al. "Catastrophic Forgetting Meets Negative Transfer: Batch Spectral Shrinkage for Safe Transfer Learning". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 1906–1916. URL: http://papers.nips.cc/paper/8466-catastrophic-forgetting-meets-negative-transfer-batch-spectral-shrinkage-for-safe-transfer-learning.pdf.

[25] Yiqiang Chen et al. "Fedhealth: A federated transfer learning framework for wearable healthcare". In: *IEEE Intelligent Systems* 35.4 (2020), pp. 83–93.

[26] Yujing Chen et al. "Asynchronous online federated learning for edge devices with non-iid data". In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE. 2020, pp. 15–24.

[27] Sharan Chetlur et al. "cudnn: Efficient primitives for deep learning". In: *arXiv preprint arXiv:1410.0759* (2014).

[28] Alessio Barbaro Chisari et al. "Benchmarking computer vision architectures for cloud detection from lidar ceilometer backscatter data". In: *The Visual Computer* (2025), pp. 1–18.

[29] Alessio Barbaro Chisari et al. "Cloud Detection Challenge-Methods and Results". In: *IEEE Access* (2025).

[30] Alessio Barbaro Chisari et al. "On the Cloud Detection from Backscattered Images Generated from a Lidar-Based Ceilometer: Current State and Opportunities". en. In: *2024 IEEE International Conference on Image Processing (ICIP)*. Abu Dhabi, United Arab Emirates: IEEE, Oct. 2024, pp. 144–150. ISBN: 9798350349399. DOI: 10.1109/ICIP51287.2024.10647352. URL: https://ieeexplore.ieee.org/document/10647352/ (visited on 10/05/2024).

[31] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20 (1995), pp. 273–297.

[32] Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure multiparty computation and secret sharing*. Cambridge University Press, 2015.

[33] Erol Cromwell and Donna Flynn. "Lidar Cloud Detection With Fully Convolutional Networks". en. In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa Village, HI, USA: IEEE, Jan. 2019, pp. 619–627. ISBN: 978-1-72811-975-5. DOI: 10.1109/WACV.2019.00071. URL: https://ieeexplore.ieee.org/document/8659255/ (visited on 08/27/2023).

[34] Tom Croonenborghs, Kurt Driessens, and Maurice Bruynooghe. "Learning relational options for inductive transfer in relational reinforcement learning". In: *Inductive Logic Programming: 17th International Conference, ILP 2007, Corvallis, OR, USA, June 19-21, 2007, Revised Selected Papers 17*. Springer. 2008, pp. 88–97.

[35] Michael Dammann et al. "Towards the Automatic Analysis of Ceilometer Backscattering Profiles using Unsupervised Learning". en. In: ().

[36] Kajaree Das and Rabi Narayan Behera. "A survey on machine learning: concept, algorithms and applications". In: *International Journal of Innovative Research in Computer and Communication Engineering* 5.2 (2017), pp. 1301–1309.

[37] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[38] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/V1/N19-1423. URL: https://doi.org/10.18653/v1/n19-1423.

[39] Enmao Diao, Jie Ding, and Vahid Tarokh. "SemiFL: Semi-Supervised Federated Learning for Unlabeled Clients with Alternate Training". In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans*. 2022. URL: http://papers.nips.cc/paper%5C_files/paper/2022/hash/71c3451f6cd6a4f82bb822db25cea4fd-Abstract-Conference.html.

[40] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. en. arXiv:2010.11929 [cs]. June 2021. URL: http://arxiv.org/abs/2010.11929 (visited on 08/27/2023).

[41] Zhixu Du et al. "Rethinking normalization methods in federated learning". In: *Proceedings of the 3rd International Workshop on Distributed Machine Learning*. 2022, pp. 16–22.

[42] Cynthia Dwork. "Differential privacy". In: *International colloquium on automata, languages, and programming*. Springer. 2006, pp. 1–12.

[43] H. Flentje et al. "The Eyjafjallajökull eruption in April 2010 – detection of volcanic plume using in-situ measurements, ozone sondes and lidar-ceilometer profiles". en. In: *Atmospheric Chemistry and Physics* 10.20 (Oct. 2010), pp. 10085–10092. ISSN: 1680-7324. DOI: 10.5194/acp-10-10085-2010. URL: https://acp.copernicus.org/articles/10/10085/2010/ (visited on 08/27/2023).

[44] Alexander Geiss. "Automated calibration of ceilometer data and its applicability for quantitative aerosol monitoring". en. In: ().

[45] Robin Geyer, Luca Corinzia, and Viktor Wegmayr. "Transfer Learning by Adaptive Merging of Multiple Models". In: *Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning*. Ed. by M. Jorge Cardoso et al. Vol. 102. Proceedings of Machine Learning Research. PMLR, Aug. 2019, pp. 185–196. URL: https://proceedings.mlr.press/v102/geyer19a.html.

[46] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[47] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. "Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[48] Aditya Golatkar et al. "Mixed-Privacy Forgetting in Deep Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 792–801.

[49] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[50] Ian J Goodfellow et al. "An empirical investigation of catastrophic forgetting in gradient-based neural networks." In: *arXiv preprint arXiv:1312.6211* (2013).

[51] Magnús T. Gudmundsson et al. "Eruptions of Eyjafjallajökull Volcano, Iceland". en. In: *Eos, Transactions American Geophysical Union* 91.21 (May 2010), pp. 190–191. ISSN: 0096-3941, 2324-9250. DOI: 10.1029/2010EO210002. URL: https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2010EO210002 (visited on 10/13/2024).

[52] Xueting Han et al. "Adaptive Transfer Learning on Graph Neural Networks". In: (July 2021).

[53] Kaiming He et al. *Deep Residual Learning for Image Recognition*. en. arXiv:1512.03385 [cs]. Dec. 2015. URL: http://arxiv.org/abs/1512.03385 (visited on 08/27/2023).

[54] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (2015).

[55] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[56] Chao Huang et al. "Promoting collaboration in cross-silo federated learning: Challenges and opportunities". In: *IEEE Communications Magazine* 62.4 (2023), pp. 82–88.

[57] Kemiao Huang and Qi Hao. "Joint multi-object detection and tracking with camera-LiDAR fusion for autonomous driving". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 6983–6989.

[58] J. Huertas-Tato et al. "Automatic Cloud-Type Classification Based On the Combined Use of a Sky Camera and a Ceilometer: Cloud class using ceilometer and camera". en. In: *Journal of Geophysical Research: Atmospheres* 122.20 (Oct. 2017), pp. 11, 045–11, 061. ISSN: 2169897X. DOI: 10.1002/2017JD027131. URL: http://doi.wiley.com/10.1002/2017JD027131 (visited on 08/27/2023).

[59] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[60] Wonyong Jeong et al. "Federated Semi-Supervised Learning with Inter-Client Consistency & Disjoint Learning". In: *9th International Conference on Learning Representations, ICLR 2021, Austria, May 3-7, 2021*. 2021. URL: https://openreview.net/forum?id=ce6CFXBh30h.

[61] Tammy Jiang, Jaimie L Gradus, and Anthony J Rosellini. "Supervised machine learning: a brief primer". In: *Behavior therapy* 51.5 (2020), pp. 675–687.

[62] Michael I Jordan and Tom M Mitchell. "Machine learning: Trends, perspectives, and prospects". In: *Science* 349.6245 (2015), pp. 255–260.

[63] Peter Kairouz et al. "Advances and open problems in federated learning". In: *Foundations and trends® in machine learning* 14.1–2 (2021), pp. 1–210.

[64] Sai Praneeth Karimireddy et al. "Breaking the centralized barrier for cross-device federated learning". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 28663–28676.

[65] Woojung Kim et al. "Federated Semi-Supervised Learning with Prototypical Networks". In: *CoRR* abs/2205.13921 (2022). DOI: 10.48550/ARXIV.2205.13921. arXiv: 2205.13921. URL: https://doi.org/10.48550/arXiv.2205.13921.

[66] Yong-Hyuk Kim, Seung-Hyun Moon, and Yourim Yoon. "Detection of Precipitation and Fog Using Machine Learning on Backscatter Data from Lidar Ceilometer". en. In: *Applied Sciences* 10.18 (Sept. 2020), p. 6452. ISSN: 2076-3417. DOI: 10.3390/app10186452. URL: https://www.mdpi.com/2076-3417/10/18/6452 (visited on 08/27/2023).

[67] Diederik P Kingma. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[68] Wei Kong and Fan Yi. "Convective boundary layer evolution from lidar backscatter and its relationship with surface aerosol concentration at a location of a central China megacity". In: *Journal of Geophysical Research: Atmospheres* 120.15 (2015), pp. 7928–7940.

[69] Wei Kong and Fan Yi. "Convective boundary layer evolution from lidar backscatter and its relationship with surface aerosol concentration at a location of a central China megacity". en. In: *Journal of Geophysical Research: Atmospheres* 120.15 (Aug. 2015), pp. 7928–7940. ISSN: 2169-897X, 2169-8996. DOI: 10.1002/2015JD023248. URL: https://agupubs.onlinelibrary.wiley.com/doi/10.1002/2015JD023248 (visited on 10/05/2024).

[70] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[71] Yann LeCun, Corinna Cortes, and CJ Burges. "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2 (2010).

[72] Sang-Woo Lee et al. "Overcoming Catastrophic Forgetting by Incremental Moment Matching". In: *31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA*. 2017.

[73] Ao Li, Xinghua Li, and Xiaoshuang Ma. "Residual dual U-shape networks with improved skip connections for cloud detection". In: *IEEE Geoscience and Remote Sensing Letters* 21 (2023), pp. 1–5.

[74] Daliang Li and Junpu Wang. "Fedmd: Heterogenous federated learning via model distillation". In: *arXiv preprint arXiv:1910.03581* (2019).

[75] Tian Li et al. "Federated optimization in heterogeneous networks". In: *Proceedings of Machine learning and systems* 2 (2020), pp. 429–450.

[76] Xiang Li et al. "On the convergence of fedavg on non-iid data". In: *arXiv preprint arXiv:1907.02189* (2019).

[77] Yong Li et al. "Privacy-preserving federated learning framework based on chained secure multiparty computing". In: *IEEE Internet of Things Journal* 8.8 (2020), pp. 6178–6186.

[78] Zachary C Lipton. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3 (2018), pp. 31–57.

[79] Mattia Litrico et al. "Semi-Supervised Domain Adaptation for Holistic Counting under Label Gap". In: *Journal of Imaging* 7.10 (2021). ISSN: 2313-433X. DOI: 10.3390/jimaging7100198. URL: https://www.mdpi.com/2313-433X/7/10/198.

[80] Fuchao Liu et al. "Measurement report: characteristics of clear-day convective boundary layer and associated entrainment zone as observed by a ground-based polarization lidar over Wuhan (30.5° N, 114.4° E)". en. In: *Atmospheric Chemistry and Physics* 21.4 (Mar. 2021), pp. 2981–2998. ISSN: 1680-7324. DOI: 10.5194/acp-21-2981-2021. URL: https://acp.copernicus.org/articles/21/2981/2021/ (visited on 10/05/2024).

[81] Aaron Loh et al. "Supervised Transfer Learning at Scale for Medical Imaging". In: 2021. URL: https://arxiv.org/pdf/2101.05913.pdf.

[82] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[83] Seema Mahajan and Bhavin Fataniya. "Cloud detection methodologies: variants and development—a review". en. In: *Complex & Intelligent Systems* 6.2 (July 2020), pp. 251–261. ISSN: 2199-4536, 2198-6053. DOI: 10.1007/s40747-019-00128-0. URL: http://link.springer.com/10.1007/s40747-019-00128-0 (visited on 08/27/2023).

[84] Dastan Maulud and Adnan M Abdulazeez. "A review on linear regression comprehensive in machine learning". In: *Journal of applied science and technology trends* 1.2 (2020), pp. 140–147.

[85] Brandon McKinzie et al. "MM1: methods, analysis and insights from multimodal LLM pre-training". In: *European Conference on Computer Vision*. Springer. 2024, pp. 304–323.

[86] Brendan McMahan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA*. 2017.

[87] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[88] Marvin Minsky and Seymour Papert. "An introduction to computational geometry". In: *Cambridge tiass., HIT* 479.480 (1969), p. 104.

[89] Yisroel Mirsky and Wenke Lee. "The creation and detection of deepfakes: A survey". In: *ACM computing surveys (CSUR)* 54.1 (2021), pp. 1–41.

[90] Tom M Mitchell. "Does machine learning really work?" In: *AI magazine* 18.3 (1997), pp. 11–11.

[91] Jaume Ruiz de Morales et al. "A method to assess the cloud-aerosol transition zone from ceilometer measurements". In: *Atmospheric Research* 310 (2024), p. 107623.

[92] Christoph Muenkel et al. "Aerosol concentration measurements with a lidar ceilometer: results of a one year measuring campaign". en. In: ed. by Klaus Schaefer et al. Barcelona, Spain, Feb. 2004, p. 486. DOI: 10.1117/12.511104. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.511104 (visited on 08/27/2023).

[93] Christoph Münkel et al. "Retrieval of mixing height and dust concentration with lidar ceilometer". en. In: *Boundary-Layer Meteorology* 124.1 (July 2007), pp. 117–128. ISSN: 0006-8314, 1573-1472. DOI: 10.1007/s10546-006-9103-3. URL: https://link.springer.com/10.1007/s10546-006-9103-3 (visited on 08/27/2023).

[94] Mrunal Naik et al. "Cloud base height variability observed using a Laser-Based Ceilometer over a tropical station Pune, India". In: *International Journal of Remote Sensing* (2024), pp. 1–14.

[95] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.

[96] National Centers for Environmental Prediction, National Weather Service, NOAA, U.S. Department of Commerce. *NCEP GFS 0.25 Degree Global Forecast Grids Historical Archive*. Place: Boulder CO. 2015. URL: https://doi.org/10.5065/D65D8PWK.

[97] Yuval Netzer et al. "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. 2011. URL: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[98] John Nguyen et al. "Federated learning with buffered asynchronous aggregation". In: *International conference on artificial intelligence and statistics*. PMLR. 2022, pp. 3581–3607.

[99] Michael Nofer et al. "Blockchain". In: *Business & information systems engineering* 59 (2017), pp. 183–187.

[100] Maxime Oquab et al. "Learning and transferring mid-level image representations using convolutional neural networks." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pages 1717–1724.

[101] Sinno Jialin Pan, James T Kwok, Qiang Yang, et al. "Transfer learning via dimensionality reduction." In: *AAAI*. Vol. 8. 2008, pp. 677–682.

[102] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22 (10 Oct. 2010). ISSN: 1041-4347. DOI: 10.1109/TKDE.2009.191.

[103] German I. Parisi et al. "Continual lifelong learning with neural networks: A review". In: *Neural Networks* 113 (2019), pp. 54–71. ISSN: 0893-6080. DOI:

https://doi.org/10.1016/j.neunet.2019.01.012. URL: https://www.sciencedirect.com/science/article/pii/S0893608019300231.

[104] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[105] Reese Pathak and Martin J Wainwright. "FedSplit: An algorithmic framework for fast federated optimization". In: *Advances in neural information processing systems* 33 (2020), pp. 7057–7066.

[106] Theodore J Perkins and Doina Precup. "Using options for knowledge transfer in reinforcement learning". In: (1999).

[107] Pedro Pinheiro and Ronan Collobert. "Recurrent convolutional neural networks for scene labeling". In: *International conference on machine learning*. PMLR. 2014, pp. 82–90.

[108] Siyuan Qiao et al. "Micro-Batch Training with Batch-Channel Normalization and Weight Standardization". In: (Mar. 2019). URL: http://arxiv.org/abs/1903.10520.

[109] Youyang Qu et al. "Blockchain-enabled federated learning: A survey". In: *ACM Computing Surveys* 55.4 (2022), pp. 1–35.

[110] Alain Rakotomamonjy et al. "Personalised federated learning on heterogeneous feature spaces". In: *arXiv preprint arXiv:2301.11447* (2023).

[111] Jason Ramapuram, Magda Gregorova, and Alexandros Kalousis. "Lifelong generative modeling". In: *Neurocomputing* 404 (2020), pp. 381–400. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2020.02.115. URL: https://www.sciencedirect.com/science/article/pii/S0925231220303623.

[112] Dushyant Rao et al. "Continual unsupervised representation learning". In: *arXiv preprint arXiv:1910.14481* (2019).

[113] Sashank Reddi et al. "Adaptive federated optimization". In: *arXiv preprint arXiv:2003.00295* (2020).

[114] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[115] Ye Ren, Le Zhang, and Ponnuthurai N Suganthan. "Ensemble classification and regression-recent developments, applications and future directions". In: *IEEE Computational intelligence magazine* 11.1 (2016), pp. 41–53.

[116] Angie K Reyes, Juan C Caicedo, and Jorge E Camargo. "Fine-tuning Deep Convolutional Networks for Plant Recognition." In: *CLEF (Working Notes)* 1391 (2015), pp. 467–475.

[117] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241.

[118] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[119] Michael T. Rosenstein et al. "To transfer or not to transfer". In: *In NIPS'05 Workshop, Inductive Transfer: 10 Years Later*. 2005.

[120] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[121] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. "Trusted execution environment: What it is, and what it is not". In: *2015 IEEE Trustcom/BigDataSE/Ispa*. Vol. 1. IEEE. 2015, pp. 57–64.

[122] Daniela Salvoni et al. "Lidar measurement of clouds profile with a Superconducting Nanowire Single Photon Detector". In: *2021 IEEE 14th Workshop on Low Temperature Electronics (WOLTE)*. IEEE. 2021, pp. 1–4.

[123] Felix Sattler et al. "Robust and communication-efficient federated learning from non-iid data". In: *IEEE transactions on neural networks and learning systems* 31.9 (2019), pp. 3400–3413.

[124] Jurgen Schmidhuber. "Multi-column deep neural networks for image classification". In: *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 3642–3649.

[125] Bernhard Schölkopf, John Platt, and Thomas Hofmann. "Correcting Sample Selection Bias by Unlabeled Data". In: *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*. 2007, pp. 601–608.

[126] Junyi Shao, Fan Yi, and Zhenping Yin. "Aerosol layers in the free troposphere and their seasonal variations as observed in Wuhan, China". en. In: *Atmospheric Environment* 224 (Mar. 2020), p. 117323. ISSN: 13522310. DOI: 10.1016/j.atmosenv.2020.117323. URL: https://linkinghub.elsevier.com/retrieve/pii/S1352231020300649 (visited on 10/05/2024).

[127] Manu Sharma et al. "Transfer Learning in Real-Time Strategy Games Using Hybrid CBR/RL." In: *IJCAI*. Vol. 7. 2007, pp. 1041–1046.

[128] Nir Shlezinger et al. "UVeQFed: Universal vector quantization for federated learning". In: *IEEE Transactions on Signal Processing* 69 (2020), pp. 500–514.

[129] Yang Shu et al. "Zoo-Tuning: Adaptive Transfer from A Zoo of Models". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 9626–9637. URL: https://proceedings.mlr.press/v139/shu21b.html.

[130] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. en. arXiv:1409.1556 [cs]. Apr. 2015. URL: http://arxiv.org/abs/1409.1556 (visited on 08/27/2023).

[131] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *ICLR 2015* (2015).

[132] Jennifer Sleeman et al. "A Deep Machine Learning Approach for Lidar Based Boundary Layer Height Detection". en. In: *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*. Waikoloa, HI, USA: IEEE, Sept. 2020, pp. 3676–3679. ISBN: 978-1-72816-374-1. DOI: 10.1109/

IGARSS39084.2020.9324191. URL: https://ieeexplore.ieee.org/document/9324191/ (visited on 08/30/2023).

[133] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. en. arXiv:1512.00567 [cs]. Dec. 2015. URL: http://arxiv.org/abs/1512.00567 (visited on 08/27/2023).

[134] Yaniv Taigman et al. "Deepface: Closing the gap to human-level performance in face verification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1701–1708.

[135] Mingxing Tan and Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. en. arXiv:1905.11946 [cs, stat]. Sept. 2020. URL: http://arxiv.org/abs/1905.11946 (visited on 08/27/2023).

[136] Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. "Safety in numbers: Learning categories from few examples with multi model knowledge transfer." In: *roceedings of IEEE Computer Vision and Pattern Recognition Conference*. 2010.

[137] Lisa Torrey and Jude Shavlik. "Transfer Learning". In: *Handbook of Research on Machine Learning Applications and Trends* (2010). DOI: 10.4018/978-1-60566-766-9.ch011.

[138] Lisa Torrey and Jude Shavlik. "Transfer learning". In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[139] Lisa Torrey et al. "Relational macros for transfer in reinforcement learning". In: *International Conference on Inductive Logic Programming*. Springer. 2007, pp. 254–268.

[140] Eric Tzeng et al. "Adversarial Discriminative Domain Adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[141] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[142] Rui Wang et al. "BEVT: BERT Pretraining of Video Transformers". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 14713–14723. DOI: 10.1109/CVPR52688.2022.01432. URL: https://doi.org/10.1109/CVPR52688.2022.01432.

[143] Wei Wang et al. "Evaluating the Governing Factors of Variability in Nocturnal Boundary Layer Height Based on Elastic Lidar in Wuhan". en. In: *International Journal of Environmental Research and Public Health* 13.11 (Nov. 2016), p. 1071. ISSN: 1660-4601. DOI: 10.3390/ijerph13111071. URL: https://www.mdpi.com/1660-4601/13/11/1071 (visited on 10/05/2024).

[144] Zirui Wang et al. "Characterizing and Avoiding Negative Transfer". In: (Nov. 2018).

[145] Kang Wei et al. "Federated learning with differential privacy: Algorithms and performance analysis". In: *IEEE transactions on information forensics and security* 15 (2020), pp. 3454–3469.

[146] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big data* 3.1 (2016), pp. 1–40.

[147] Paul Werbos. "Beyond regression: New tools for prediction and analysis in the behavioral sciences". In: *PhD thesis, Committee on Applied Mathematics, Harvard University, Cambridge, MA* (1974).

[148] M. Wiegner and A. Geiß. "Aerosol profiling with the Jenoptik ceilometer CHM15kx". en. In: *Atmospheric Measurement Techniques* 5.8 (Aug. 2012), pp. 1953–1964. ISSN: 1867-8548. DOI: 10.5194/amt-5-1953-2012. URL: https://amt.copernicus.org/articles/5/1953/2012/ (visited on 08/27/2023).

[149] Cheng Wu and Fan Yi. "Local ice formation via liquid water growth in slowly ascending humid aerosol/liquid water layers observed with ground-based lidars and radiosondes". en. In: *Journal of Geophysical Research: Atmospheres* 122.8 (Apr. 2017), pp. 4479–4493. ISSN: 2169-897X, 2169-8996. DOI: 10.1002/2016JD025765. URL: https://agupubs.onlinelibrary.wiley.com/doi/10.1002/2016JD025765 (visited on 10/05/2024).

[150] Chuhan Wu et al. "Communication-efficient federated learning via knowledge distillation". In: *Nature communications* 13.1 (2022), p. 2032.

[151] Yuxin Wu and Kaiming He. "Group normalization". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19.

[152] Zhaomin Wu, Qinbin Li, and Bingsheng He. "Practical vertical federated learning with unsupervised representation learning". In: *IEEE transactions on big data* (2022).

[153] Wai Yeung Yan. "Airborne LiDAR data artifacts: What we know thus far". In: *IEEE Geoscience and Remote Sensing Magazine* (2023).

[154] Qiang Yang et al. "Federated machine learning: Concept and applications". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–19.

[155] Fei Ye and Adrian G. Bors. "Learning Latent Representations Across Multiple Data Domains Using Lifelong VAEGAN". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 777–795. ISBN: 978-3-030-58565-5.

[156] Xun Yi et al. *Homomorphic encryption*. Springer, 2014.

[157] Chengliang Zhang et al. "{BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning". In: *2020 USENIX annual technical conference (USENIX ATC 20)*. 2020, pp. 493–506.

[158] Xinwei Zhang et al. "Hybrid federated learning: Algorithms and implementation". In: *arXiv preprint arXiv:2012.12420* (2020).

[159] Sicheng Zhao et al. "A review of single-source deep unsupervised visual domain adaptation". In: *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[160] Jike Zhong, Hong-You Chen, and Wei-Lun Chao. "Making batch normalization great in federated deep learning". In: *arXiv preprint arXiv:2303.06530* (2023).

[161] Fuzhen Zhuang et al. "A Comprehensive Survey on Transfer Learning". In: *Proceedings of the IEEE* 109.1 (2021), pp. 43–76. DOI: 10.1109/JPROC.2020.3004555.

[162] Dorsa Ziaei et al. "Convolutional LSTM for Planetary Boundary Layer Height (PBLH) Prediction". en. In: ().

[163] Weijie Zou et al. “Robust lidar-radar composite cloud boundary detection method with rainfall pixels removal”. In: *IEEE Transactions on Geoscience and Remote Sensing* (2024).


UNIONE EUROPEA
Fondo Sociale Europeo
Ministero dell'Università e della Ricerca
PON
RICERCA E INNOVAZIONE
REACT EU