UiO : Department of Geosciences
University of Oslo

Development of a classification algorithm for ice crystal habit by using deep learning

Huiying Zhang

Master's Thesis, Autumn 2021

Abstract

Ice crystals, as an important component of clouds, have a strong influence on cloud radiative properties and precipitation formation. Moreover, ice crystal habits are controlled by the environment (temperature and humidity) in which they grow in and as such, are excellent tracers of in-cloud conditions. Therefore, ice crystal habit classification is an excellent tool to better understand the microphysical processes in clouds and thus, cloud radiative properties and precipitation formation. Over the past 50 years, researchers have improved the ability of algorithms to automatically and efficiently classify ice crystal habits. The most recent attempts have utilized machine learning and more specifically, a Convolutional Neural Network (CNN), due to its ability to catch the main features that describe ice crystal habits and recognize patterns between images.

However, the CNNs trained on standard ice crystal habit images are difficult to apply in reality, due to the complexity of ice crystals in nature, which are generally a combination of different habits, rimed, or aggregates, and the difference between training dataset and real-world dataset.

Therefore, in this thesis, a CNN is trained using images of ideal and complex ice crystals recorded by the HoloBalloon instrument during the NASCENT campaign in Fall 2019, in Ny-Ålesund, Norway. The dataset includes 16,259 images that were hand-labeled into 9 ice crystal habit classes. The best performing classification model ensemble (including 10 members), BestIce, achieved an overall accuracy of 87.55% and a class-wise accuracy of 91.72%. The models performed best when classifying plates and lollipops and frozen droplets and small ice with per-class accuracies of around 99.5% and 98%, respectively. To validate BestIce in a real-world application, the model is used to predict the ice crystals observed on a different day. When the prediction probability of BestIce was 99% or higher, which made up approximately 40% of the entire dataset, the global accuracy of the prediction was approximately 80%. However, when all of the ice crystals were classified with BestIce, the global accuracy fell to 63.31%. Nevertheless, the ability of BestIce to predict approximately 40% of the new dataset with such a high accuracy shows that the method developed in this thesis can be used to effectively classify ice crystals in a real-world setting.

Acknowledgements

Throughout the writing of this thesis, I have received a great deal of support and assistance.

I would first like to thank my supervisors, Prof. Trude Storelvmo, who always provide insightful feedback and steer me in the right direction; Dr. Robert Oscar David, who gives me the most patience and thorough help, offering advices and encouragement in both science and English with a perfect blend of insight and humor; Prof. Alexander Binder, who offers professional help in machine learning and coding, leading me to a real machine learning world rather than standing in the doorway and Prof. Morten Hjorth-Jensen, who took me opening the door of machine learning. Without your patient help and dedicated involvement in every step throughout the process, this thesis would have never been accomplished.

I would also like to deeply acknowledge the Hologroup members from ETH Zürich, Dr. Jan Henneberger, Julie Pasquier, Annika Lauber, Dr. Fabiola Ramelli. Thanks for providing the opportunity to join your meeting, and offering the insightful and exciting discussion. In particular, I would like to extend my sincere thanks to Julie for providing the hand-labeled dataset of ice crystal holographic images.

I wish to thank all people in UiO Cloud Group for interesting and invaluable discussion and feedback and all kinds of help.

I'd also like to extend my gratitude to my teacher in Cloud Physics and my boss at the Norwegian Meteorological Institute (Metno), Prof. Erik Berge. Your responsible attitude and enthusiasm for the topic made a strong impression on me and I have always carried positive memories of your classes with me. Additionally, thank you for offering me a job at Metno, which is financially important and a recognition of my ability.

Thanks to my best friend in Norway, Ove Westermoen Haugvaldstad, who offers unconditional help, strongest support, and kind company in both daily life and research (especially in coding) during my whole Masters' life. Additionally, I would also like to express my sincere thanks to my friends, Anna Lina Sjur, and Jan-Andrian Kallmyr. Thank you for all good memories of ours, hiking, cooking, Norwegian Christmas dinner, and of course drinking beers.

I would also like to thank the students for many great lunch breaks and speaking English only for me and the staff at MetOs for making the happy days of studying.

Thanks for University of Oslo for all goods I experienced here as an international student; Thanks to Norway for all kindness I received here as a foreigner.

Thanks to my friends, Yiyu Zheng, Guo Lin and Zijiang Yang. Thanks for our weekly online party and all kind encouragement from you.

Thanks to my friends, Lian Zhang, Zhihong Zhuo, Hui Tang, Jing Wei, Qingqi Shi, Maoxin Zhang, Yiyuan Guo, Shuyuan Huang, Shuangze Zou and Yaxi Li. Thank you for all kind help and great company for both online and in-person.

Lastly, my family and my boy friend deserves endless gratitude: Thank you for your unconditional supports, trust and love. Thank you for your phone that never loses connection so that I can contact you whenever I needed.

《谢师表》

记于辛丑年夏，七月初二雷雨夜。岁月匆匆，如白驹过隙；二载求学，今已将尽，亦某拙劣之论将呈际，感念万分，故作此以谢师恩。

余渡重洋万里而来，初时不善洋文言辞，故常困于此。饮食起居，学堂授课，皆举步维艰。然余出身布衣，忝得举家之力方至此。虽学堂已免束修，仅衣食住行，已是重负。有言：“天将降大任于是人也，必先苦其心志，劳其筋骨，饿其体肤，空乏其身，行拂乱其所为。”某虽愚笨，亦终日勤勉，上下求索。又得吾师孜孜教诲，吾友循循善诱于堂，于厅，于素日三餐闲聊，终有所成，畅所言，不逾矩。

承蒙吾导Trude Storelvmo，Robert Oscar David，Alexander Binder及Morten Hjorth-Jensen 不弃，忝列门墙，故得以在云物理及人工智能上得以发展进益。余忆过往相处种种，凡有所惑，无论请教于谁人，必亲躬，吾亦必得所答或得善诱。是日，彼时吾已归故里，然某有一疑惑不解之处。众所周知，此地于故里，有万里之远，时差亦有三。为解惑，吾导Rob, 于此处三更天与余远程会面，足足解惑一时辰有余，每每忆此，莫不动容，潸然泪下。余不善洋文撰文，每每新章毕，吾导必字句斟酌，然后以谆谆教导，徐徐善诱，以求吾略有所悟，并循环往复，以求上文。

呜呼，师，恩高于山，深于汪洋。某尽纸而不足书矣，浅薄言辞以尽心。

辛丑年丙申月己丑日

张卉颖

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation and Objective

Clouds, composed of liquid (liquid droplets) and/or ice (ice crystals), cover almost 70% of the Earth (Stubenrauch et al., 2013). Thus, they play an important role in the climate system via the radiation budget (i.e. Ehrlich et al., 2008;Sun and Shine, 1994;Matus and L'Ecuyer, 2017) and hydrological cycle (i.e. Field and Heymsfield, n.d.; Mülmenstädt et al., 2015).

Cloud radiative properties (i.e. Sun and Shine, 1994; Y. Zhang et al., 1999; Schlimme et al., 2005) and precipitation formation (Field and Heymsfield, n.d., Mülmenstädt et al., 2015) strongly depend on ice crystals due to their ability to grow at the expense of cloud droplets or short in-cloud lifetime. The ice crystal habits, which are determined by the environment (temperature and humidity) within which ice crystals grow (Bailey and Hallett, 2009), provide information about the conditions that ice crystals formed and spent the majority of their lifetime in (1.1). Therefore, to better understand the microphysical processes in clouds and further obtain a better understanding on radiation characteristics and precipitation formation of clouds, ice crystal habit classification is essential.

Convolutional Neural Network (CNN) is a class of neural networks within deep learning, which is often used to analyze images due to their ability to catch the main features from an image directly and recognize patterns. Previously, a CNN based on pre-trained models (eg:TL-ResNet18, TL-ResNet34) has been used to classify 10 standard ice crystal habits with 96%accuracy (Xiao et al., 2019). However, the CNNs trained on standard ice crystal habit images are hard to apply in reality due to the effect of the distribution shift (the difference) between training (standard ice for training) and test data(ice data from real world). In the real world, ice crystal habits are often much more complex than the well-selected ones used in their training dataset. Moreover, subjectivity from person is unavoidable as one 'answer' has to be given for each single ice crystal, even for some compound ice crystal (i.e. rimed-aggregate).

Therefore, the objective of this thesis is to develop an automatic classification model for ice crystal habits (holographic images directly from the real world) by using CNNs.

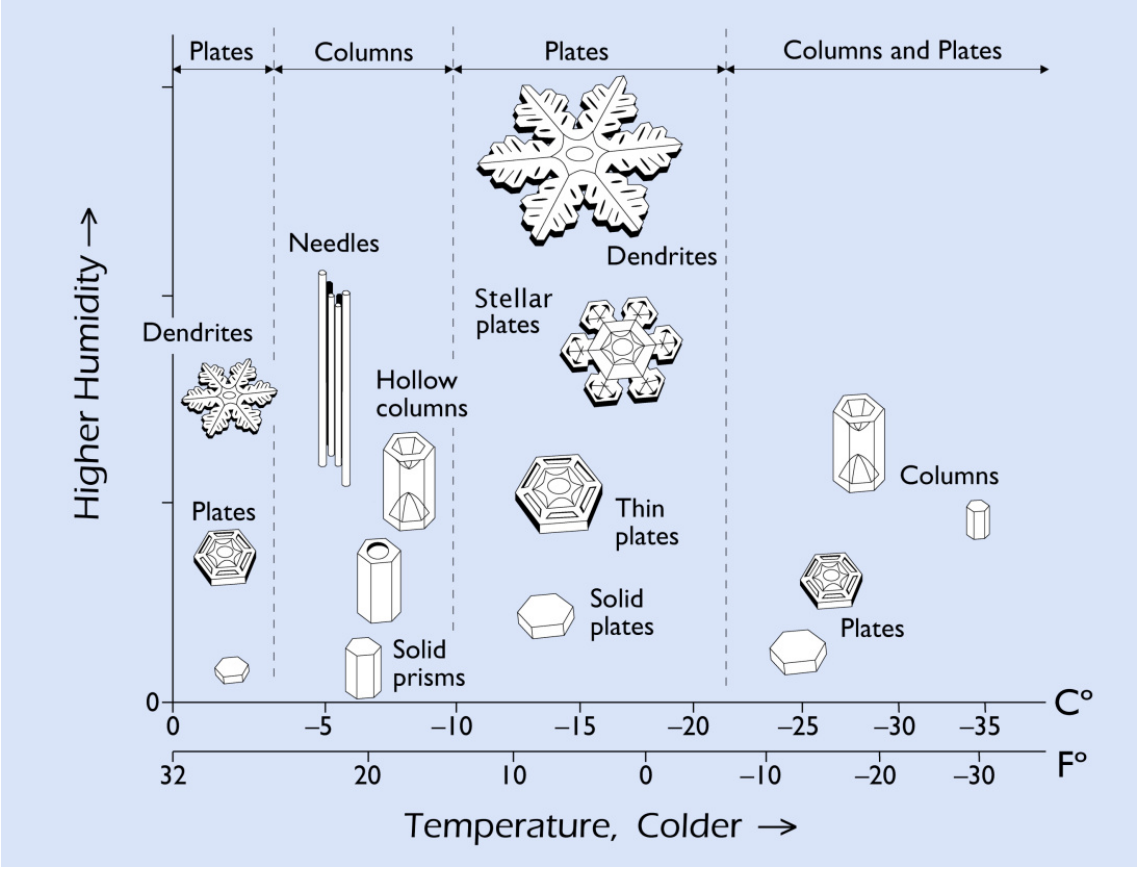

Figure 1.1: Ice crystal habit diagram with the change of humidity and temperatures (Libbrecht, 2016)

1.2 Thesis Outline

In Chapter 2, the role of clouds in the climate system via the radiation budget and precipitation are discussed. In particular, the importance of ice crystals and ice crystal habits is emphasised. Instruments for cloud particle detection are then introduced and compared. To better understand the development of ice crystal habit classification rules and methods, the history of ice crystal habit classification is presented.

In Chapter 3, CNNs are introduced. Then an overview of transfer learning and fine-tuning is presented. Finally, the optimization methods for upgrading CNNs, preprocessing methods (image augmentation), and the evaluation metrics used in this thesis are described.

In Chapter 4, the details regarding the dataset, data pre-processing, and deep learning model implementation are presented.

In Chapter 5, the results from evaluation of different models trained by the original (imbalanced) dataset are reported and then the impact of freezing the lower half of the model layers and including a physical attribute are investigated. Afterwards, the impacts of training on a rebalanced (remove 2/3 of the dominant class) and balanced (sample images from each of the classes with

equal probability) version of the original dataset with and without Test-Time Data Augmentation (a technique that can boost a model's performance) are evaluated. Finally, a new dataset is used for the validation.

In Chapter 6, a summary and some ideas for future work are provided.

The facilities used in this thesis are presented in Appendix C.

CHAPTER 2

Background

2.1 The role of clouds

Mixed phase clouds (MPCs), containing both ice crystals and liquid droplets, play an important role for precipitation (Mülmenstädt et al., 2015). The fraction of ice is an important component of precipitation as ice crystals can grow at the expense of the supercooled liquid droplets according to the Wegener–Bergeron–Findeisen (WBF) process. Due to the coexistence of ice and water, MPCs are very efficient at producing precipitation and are responsible for over 30 % and 50 % of the precipitation that falls over the Ocean and land in the mid-latitudes, respectively (Field and Heymsfield, n.d.; Mülmenstädt et al., 2015). Moreover, ice crystals grow much faster than water droplets because the ice saturation line is lower than the water saturation line, which means in the same environment, ice experiences higher supersaturation (e.g. Lamb and Verlinde, 2011). Thus, ice crystals grow and fall as precipitation faster than cloud droplets, which have to undergo collision-coalescence to reach sizes large enough to precipitate(e.g. Lamb and Verlinde, 2011). Thus, MPCs plays a critical role in the global hydrological cycle.

Clouds reflect incoming shortwave solar radiation, which acts to cool the Earth. Simultaneously, clouds can warm the Earth by absorbing and re-emitting longwave radiation from the surface. The amount of reflected versus trapped radiation strongly depends on the cloud height and optical thickness (Figure 2.1). Generally, high clouds have a net warming effect due to the dominance of the longwave cloud radiative effect, while low clouds have a net cooling effect due to the dominance of the shortwave cloud radiative effect. This is in part due to the fact that the albedo or reflectivity, depends on cloud thickness. As high clouds are often optically thin, they act as weak reflectors, while low clouds are generally optically thick so they act as strong reflectors. Secondly, the longwave cloud radiative effect depends on the cloud top height. Since high clouds have high cloud tops where the temperature is much lower than at the surface, they emit much less radiation (Blevin and Brown, 1971), acting to warm the climate. Meanwhile, since low clouds have cloud top temperatures closer to that of the surface, the radiated emission is not very different relative to the surface. Thus, low clouds have a weak longwave cloud radiative effect (J. Slingo and A. Slingo, 1991). The cloud optical thickness is determined by the concentration, size and phase of cloud

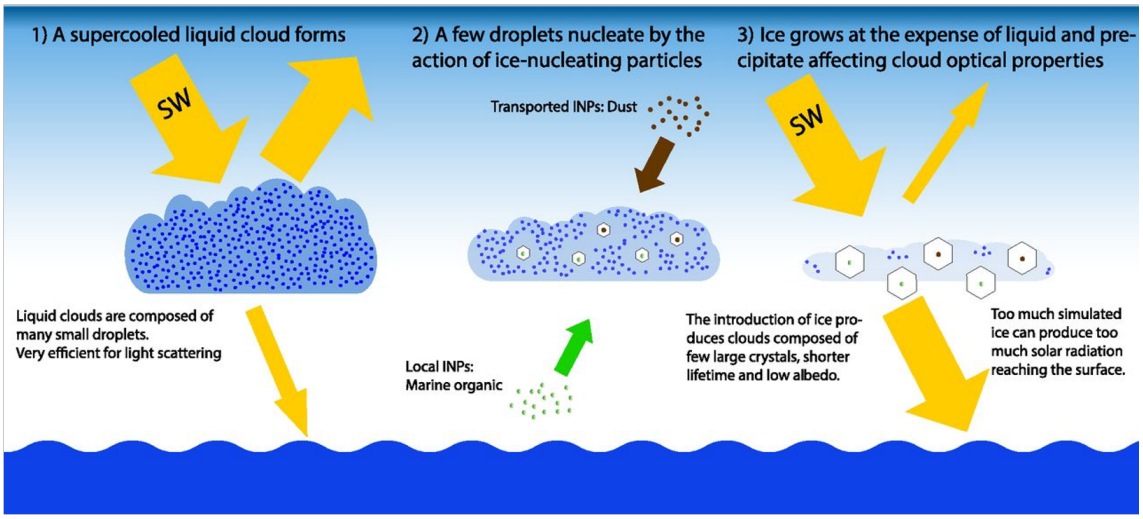

Figure 2.1: Radiation difference among different cloud phase (Vergara-Temprado et al., 2018)

particles. Correspondingly, there are two cloud types, warm clouds and cold clouds (which include ice clouds and MPCs). Warm clouds are composed of cloud droplets, while ice clouds consist of ice crystals and MPCs are composed of both cloud droplets and ice crystals. Moreover, for a MPC, the lifetime and amount of ice and liquid depends on the development of precipitation, as mentioned above, through the WBF process. Hence, knowing the microphysical properties including ice crystal number concentration (ICNC), size, spatial distribution (e.g., solely ice crystals or uniform mixture), and ice crystal habits (i.e. Sun and Shine, 1994; Y. Zhang et al., 1999; Schlimme et al., 2005) is essential for predicting how clouds influence radiation.

2.2 The role of ice crystal habits

This section will first give an introduction to ice crystal habits and then discuss their importance.

Ice crystal habits, put simply, can be understood as the shapes of ice crystals (i.e., what they look like). The primary habit denotes the distinct category for the ice crystal shape and is mainly determined by temperature. Generally, the two primary habits are thought to be ”Plates” and ”Columns”, but ice crystals can also be further characterised by multiple secondary habits that are mainly determined by supersaturation.

Thus, ice crystal habits provide insight into what happened during the ice crystals growth process. Let us start with the primary ice crystal habits. The criteria for how we distinguish columns and plates is according to the aspect ratio c/a where c and a are the basal and prism faces, respectively. As we can see in Figure 2.2, we assume R_B is the linear growth rate of basal faces and R_P is the linear growth rate of prism faces. When the basal face grows faster than the prism face, we get a columnar ice crystal; when it is the opposite, we get a plate. To be more intuitive, we could imagine the basal face as either short or tall and the prism face as either thick or thin. Thus, the

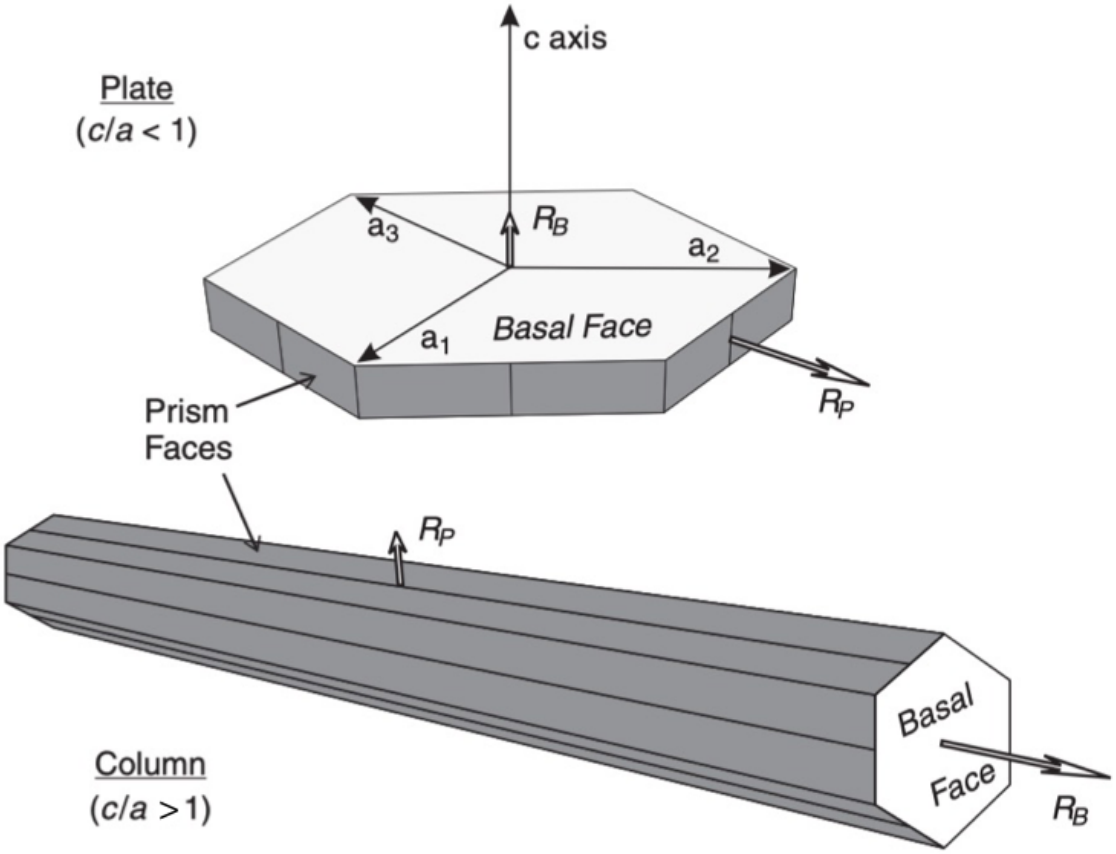

Figure 2.2: The primary habits of ice crystal. Top: plate; bottom: column (Lamb and Verlinde, 2011)

growth ratio is described as below:

$$\frac{R_B}{R_P} = \frac{dc/dt}{da/dt} = \frac{dc}{da} \tag{2.1}$$

The growth ratio in turn depends on the temperature. As can be seen from Figure 2.3, at temperatures warmer than -5 °, the prism face grows faster than the basal face; from -10 to -5 °, the opposite is true; at temperatures below -10 °, the prism face grow faster than the basal face again. However, these are just the most basic habits of ice crystals. In reality, ice crystal habits can vary even more. For example, from -10 to -5 °, ice crystal habits can be hollow columns, solid long needles, sheaths, and scrolls. Between 20 and 10 °, ice crystals can become thick plates of a skeleton form, fern-like with sector-like branches, stellar, ordinary dendrites, and hexagonal plates (K.-N. Liou and P. Yang, 2016). Moreover, some processes, for example riming and aggregation, that happen as ice crystals fall through a cloud, can also influence the ice crystal habits. Thus, classifying ice crystal habits is very important in order to better understand the microphysical conditions and processes in clouds.

Radiative properties differ between ice crystals and liquid droplets (Ehrlich et al., 2008; Sun and Shine, 1994). Ice crystal are generally larger and fewer than liquid droplets so that ice clouds typically have a lower albedo (as shown in the Figure 2.1). Thus, warm clouds composed purely

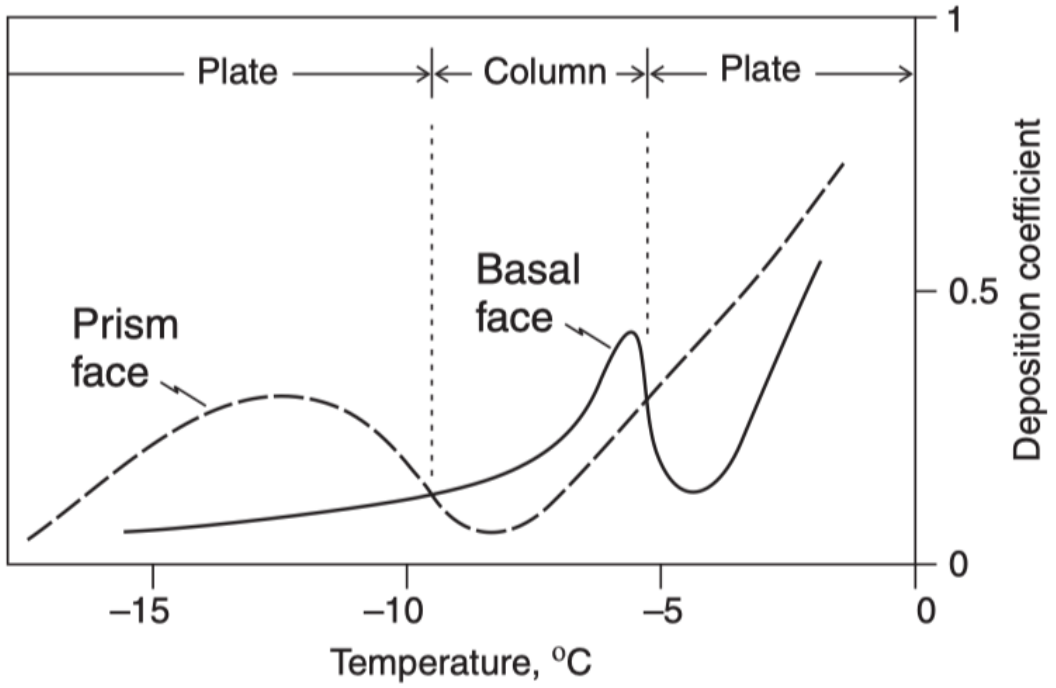

Figure 2.3: Deposition coefficient (proxy for growth rate) of the basal face (solid curve) and prism face (dashed curve) as a function of temperature (Lamb and Verlinde, 2011)

of liquid droplets have higher optical thickness and thus have a cooling effect compared to cold clouds (Lohmann, 2002). The radiative effects of ice crystals are a function of the crystal size (Stephens et al., 1990; Fu and K. N. Liou, 1993) and habit. These relationships have important implications for remote sensing estimates of precipitation rates and cloud radiative properties. In terms of precipitation, the habit impacts the riming efficiency and the ice crystal fall speed, which is important for estimating the mass of snow that reaches the ground. Thus, accurately identifying these properties is essential for accurate estimates of precipitation and cloud-climate interactions.

In conclusion, ice crystal habit classification is needed for a better assessment of cloud microphysical pathways, and for assessing the impact of habit on cloud radiative properties and precipitation estimates from remote sensing.

2.3 Instruments for detection of cloud microphysical properties

A large amount of research has been conducted on the instruments used to measure ice crystals in clouds. These techniques have been developed for different applications, including single-particle detection, bulk liquid and ice water detection, shape measurements and cloud particle optical properties. However, here the focus is on single-particle detection techniques. The instruments used for single particle methods can be divided into two main measurement techniques, namely light scattering and imaging sensors ('Cloud Ice Properties: In Situ Measurement Challenges' 2017).

Some examples of light scattering probes include the Forward Scattering Spectrometer Probe (R. Knollenberg, 1976), the Cloud Droplet Probe (Lance, 2012), the Cloud and Aerosol Spectrometer (CAS Baumgardner et al., 2001), the Cloud and Aerosol Spectrometer (Baumgardner et al., 2001), the Cloud and Aerosol Spectrometer with polarization (Glen and Brooks, 2013), the Backscatter Cloud Probe (Beswick et al., 2014). Meanwhile, common imaging sensors include the Two-Dimensional Stereo spectrometer (2D-S, R. P. Lawson, O'Connor et al., 2006) and Precipitation spectrometers (R. G. Knollenberg, 1970), the High Volume Precipitation Spectrometer (HVPS, R. P. Lawson, Stewart, Strapp et al., 1993), the Cloud and Precipitation Imaging Probes (CIP Baumgardner et al., 2001), the Two-Dimensional Stereo spectrometer (R. P. Lawson, O'Connor et al., 2006), the High Volume Precipitation Spectrometer (R. P. Lawson, Stewart and Angus, 1998), the Cloud Particle Imager (Baumgardner et al., 2001), and holographic imagers such as the Holographic Detector for Clouds (J. P. Fugal and Shaw, 2009), HOLographic Imager for Microscopic Objects II (HOLIMO II, Henneberger et al., 2013). For this thesis we only consider holography.

Holography techniques have been widely used since 1975(i.e.Trolinger, 1975, Borrmann et al., 1993, R. Lawson and Cormack, 1995) and have been conducted on several platforms including aircraft (i.e. Beals et al., 2015, J. P. Fugal and Shaw, 2009, Spuler and J. Fugal, 2011), tethered-balloon systems (Ramelli et al., 2020), cable cars (Beck et al., 2017), mountaintop research stations (i.e. Borrmann et al., 1993; Henneberger et al., 2013) and in the laboratory (Amsler et al., 2009).The basic principles of holography are as follows:

In-line holography utilizes a collimated/straight light source with a known wavelength to illuminate a sample volume of a given size. During the illumination, as shown in Figure (2.4), the laser irradiates a coherent reference wave through the well-defined sample volume, which contains an ensemble of cloud particles. When cloud particles meet the reference wave U_R (see Fig 2.4), they scatter U_R and produce scattered wavefronts U_S (see Fig 2.4) by interfering with U_R. The resulting interference pattern known as a hologram is recorded by the camera (see Fig 2.4) as a 2D picture. The ring pattern on the 2D image is the superposition of the reference wave U_R and the scattered wave U_S (see Fig 2.4) and the intensity of the ring pattern can be described by the modulus squared of the superimposed waves as follows:

$$\begin{aligned} I_H &= |U_S + U_R|^2 \\ &= U_R * U_R + U_S * U_S + U_R U_S^* + U_R^* U_S \end{aligned} \tag{2.2}$$

where U_R*U_R is the mean intensity of the reference wave, which can be seen as the constant background, U_S*$U_S$is the mean intensity of the scattered wave, which is usually ignored due to its much smaller order of magnitude, $U_R * U_S^*$ is the virtual image and $U_R^* * U_S$ is the real image. These four terms describe the particles' position and two-dimensional shape.

The hologram is then reconstructed in different planes by the software HOLOSuite (modified version of J. P. Fugal and Shaw, 2009) where the original position of the particles responsible for

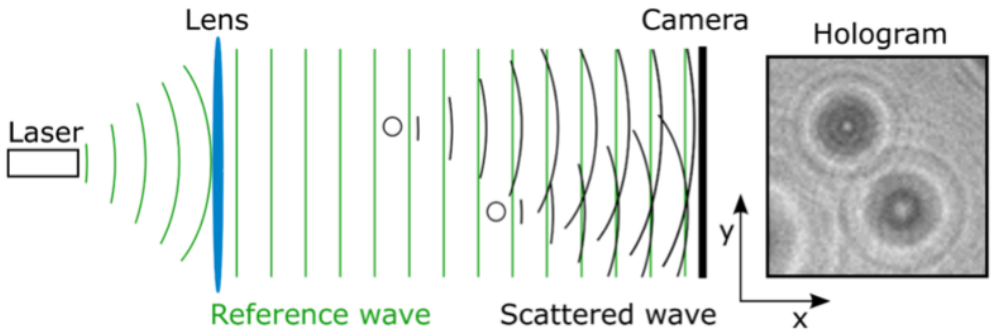

Figure 2.4: Working process of digital in-line holography (Touloupas et al., 2020)

U_S can be identified. The working principles of HOLOsuite are mainly divided into 2 steps:

1. Reconstruction
2. Classification of particles into 3 classes – liquid droplets (circular particles), ice crystals (non-circular particles) and artifacts (parts of the interference pattern, scratches on the windows, noise, etc) by machine learning.

Among all the single-particle detection measurements as mentioned above, holographic techniques have the following advantages and disadvantages compared to other techniques

Advantages:

1. Holography is able to measure over a wide range of particle sizes, typically ranging from 6 μm to 1 cm ('Cloud Ice Properties: In Situ Measurement Challenges' 2017). The minimum particle detection size is decided by the diffraction limit of the optics (i.e., the resolution) and by the fact that the particle must be larger than two pixels wide to be resolvable (J. P. Fugal and Shaw, 2009). For the maximum detection size, it is partly dependent on the detector size and the ability of the post-processing code to determine the correct focal plane of large particles so that they are in focus (J. P. Fugal and Shaw, 2009).
2. It catches the information directly from the real image, so there is no need to convert the intensity of scattered light measured at specific angles to particle sizes, unlike some other light scattering probe (i.e. CAS). This also means that assumptions of particle shape, orientation, refractive index and scattering direction can be avoided.
3. In contrast to other single particle techniques, such as, triggered particle imagers, CPI (SPECinc, Colorado USA), holography offers a well-defined sample volume independent of particle size and air speed (J. P. Fugal and Shaw, 2009). Thus, it effectively avoids the

uncertainties in the calculation of the effective sample volume and subsequent cloud particle concentrations.

4. Holography provides the particle position information in the sample volume and thus offers a high-resolution spatial distribution of the particles on a millimeter scale (Beals et al., 2015). Thereby, it allows us to investigate particle clustering, ice crystal shattering (J. P. Fugal and Shaw, 2009) and the spatial scales of mixing between liquid and ice (e.g. inhomogeneous cloud mixing) at the centimeter scale (Beals et al., 2015).

5. Holography largely reduces the error from shattering by selecting a sample volume about 1 or 2 cm away from the probe arms during the reconstruction.

Disadvantages:

1. Holography can produce a lot of artifact (noise) which largely increase the post-processing cloud particle classification work. However, an automated classification model (between droplets, artifacts and ice crystals) trained by machine leaning, is now included in the post-processing software, HOLOsuite. This largely reduces the effort needed to determine cloud microphysical properties (Touloupas et al., 2020).

2. Holography is extremely computationally expensive due to the extensive post-processing required. Generally, graphics processing units (GPUs) can do 10–15 times the operations of CPUs and thus can save large amount of computation time.Therefore, future work should be done to transfer the HOLOsuite software to GPU processing (Henneberger et al., 2013). Generally, GPUs can do 10–15 times the operations of CPUs and thus can save a large amount of computation time (Schlegel, 2015). However, the problem of high computational cost is alleviated by the development of both the software package (HOLOSuite J. P. Fugal and Shaw, 2009) and computer technologies.

In conclusion, holography is an important technique for cloud microphysical studies and has a large potential to be further developed.

2.4 Ice crystal classification history

In the past few decades, several studies have been carried out on the classification of ice crystals and solid hydrometeor images measured by optical array probes (OAP). I personally separate the history of ice crystal classificationinto into. three time periods:

1. Simple descriptors searching period

2. Advanced particle descriptors searching period

3. Automatic classification period.

Hydrometeor classification techniques began with easily found (dimension-related) particle features, such as size parameters. Cunningham (1978) utilized only the edge complexity of particles and their equivalent circle ratio, to classify hydrometeor images measured by Particle Measuring System probes. Since then, people started to use different methods to seek the most representative features of particles for classification. Rahman et al. (1981) generated a set of synthetic images for selecting geometrical parameters to classify binary two-dimensional images of hydrometeors. They found ten time domain features/geometrical parameters (i.e. circular deficiency, which is defined as the absolute value of the difference between the area of the target image and the circle) for hydrometeors classification. Duroure et al. (1994) analysed particle habit and size distribution of the population by using the geometrical measures S and P of individual particles, where S is the square area of the image, P is particle image perimeter. Another more simple approach is to classify ice crystals by comparing the particle maximum dimension and area ratio (e.g.,McFarquhar and Heymsfield, 1996; Intrieri et al., 2002). In the first time period, these techniques are relatively computationally-cheap and fast but they struggle to distinguish between some composite ice crystal habits, such as, irregular, aggregates, or bullet rosettes.

As computational power increased, advanced/high-computation methods were introduced to classify ice crystals for complex ice crystals classification. The first used a self-organized neural network algorithm, also based on particle dimension and area ratio, which achieved a habit identification accuracy of 69% for bullet rosettes and 87% for polycrystals (McFarquhar, Heymsfield et al., 1999). Korolev and Sussman (2000) proposed a method that could distinguish four families of snow particles by checking dimensionless ratios of simple geometrical parameters (Korolev and Sussman, 2000). In 2006, Feind (2006) confirmed the high accuracy of neural network methods through a comparison of different classification techniques. The main finding emphasized the importance of the key particle features be utilized. This means that if less dominant/relevant features are included, it may not help improve the classification accuracy and even make it worse. The Ice-crystal Classification with Principal Component Analysis tool applies principal component analysis into the classification (Lindqvist et al., 2012) and achieves an accuracy of over 80% . Praz et al. (2017) applied logistic regression to classify the images from the Multi-Angle Snowflake Camera (MASC). MASC is a ground-based snowflake imager that captures high-resolution (down to 35 microns per pixels) photographs of falling snowflakes from three different angles (Garrett et al., 2012). Praz et al. (2017) also used a logistic regression algorithm for ice crystal habit classification from an airborne 2D-S, a HVPS, and a CPI with over 90% accuracy. However, their technique still required manual feature extraction (i.e. aspect ratio).

Deep CNN-based feature extraction was not proposed until 2019 whenXiao et al. (2019) trained CNNs based on some pre-trained model (i.e.TL-ResNet18, TL-ResNet34) to automatically classify

10 standard ice crystal habits with 96%accuracy. This method largely increased ice crystal habit classification accuracy and efficiency. However, if the CNNs is trained on ideal ice crystal habits images, it is hard for the model to be applied on more complex ice crystal habits found in nature. As ice crystals are influenced by the environment they grow in, it is very likely that large differences in ice crystals will exist between different sampling days or even within the same sample period. Therefore, it is difficult for a model to perform well when it has to predict ice crystals that are different than the ones that it was trained with. Regardless, with the exception of some very complex and confusing ice crystals, the majority of ice crystals still fall within one of the classes determined during the training. Additionally, a field-collected dataset can be very imbalanced, for example, during one specific campaign, column shaped ice crystals could occupy over 50 % of the total ice crystal number. Moreover, for any supervised learning technique, a 'label' (i.e. column, plate) for each ice particle must be provided and thus, it is hard to avoid subjectivity during the hand-labeling process. Therefore, to avoid this subjective bias, Leinonen and Berne (2020) applied an unsupervised learning method, generative adversarial network (GAN) and K-medoids method for the classification of snowflakes obtained from MASCs. However, it is a completely unlabeled clustering approach and therefore, has no guidance at all as to which features are scientifically essential and which are not, so that it unavoidably retains some irrelevant and even meaningless information. Thus, unsupervised learning still struggles to classify ice crystal habits as defined by the atmospheric science community. Also, due to the relatively coarse resolution of the MASC, some important shape features were missed. Therefore, in this thesis, I develop an automatic ice crystal habit classification algorithm using a CNN as done by Xiao et al. (2019) but designed to work on ice crystals images without any selection (directly collected from the real world). The background on CNNs and the techniques surrounding them are described in the following chapter.

CHAPTER 3

Method

3.1 Neural networks

Neural networks (NNs) are a popular approach in machine learning and have become a hot topic in recent years. NNs are computational systems that can learn to perform tasks by considering examples, without being programmed with any task-specific rules. This approach was inspired by the human brain and how it passes information among neurons. Generally, NNs consist of three basic components, an input layer, hidden layers and an output layer (as shown in Figure 3.1). They are meant to mimic a biological system, wherein neurons interact by sending signals in the form of mathematical functions between layers. All layers can contain an arbitrary number of neurons, and each connection is represented by a weight variable.

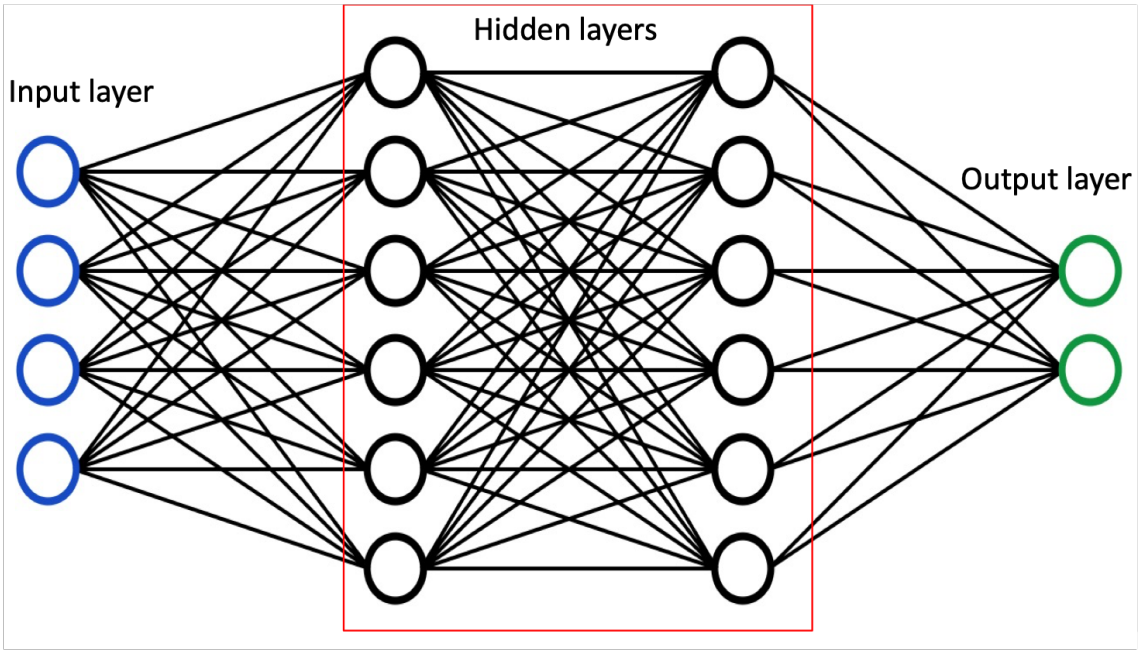

Figure 3.1: Fully connected neural network architecture with two hidden layers

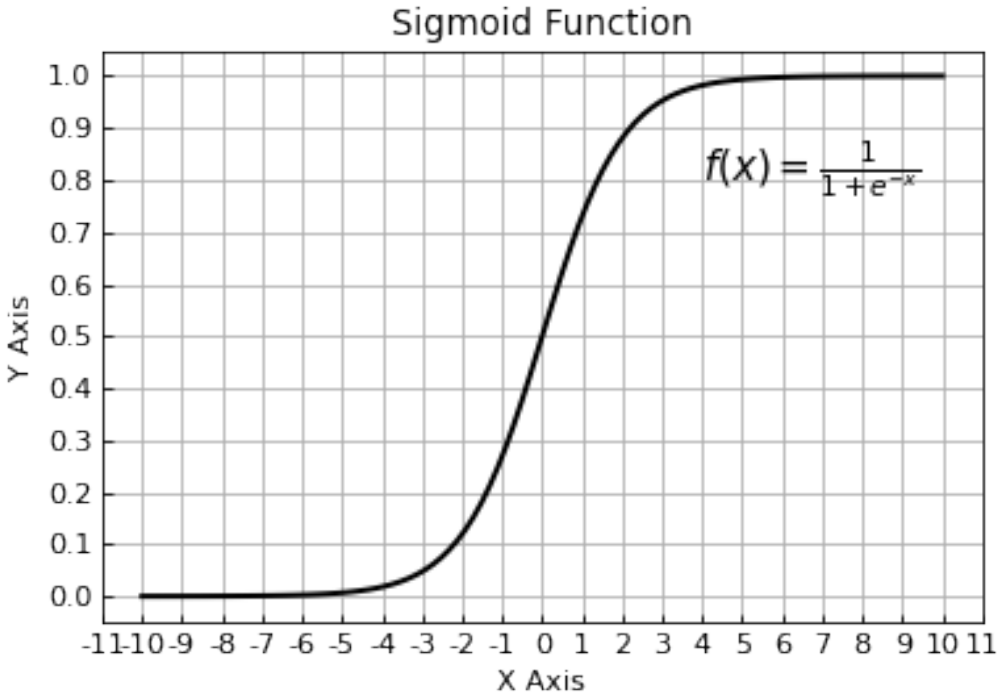

Figure 3.2: Sigmoid activation function

3.1.1 Feed-forward neural networks

The feed-forward neural network (FFNN) is a type of sing-direction NN, which transports information from an input layer to an output layer while never going back.

Generally, FFNNs pass information starting from the input layer. Each output $f(Z_i)$ in the next layer is connected to the neurons (input X_i) in the previous layer with their corresponding weight W_i. The input to the next layer is the obtained value of the activation function and the input of the activation function is the sum of the set of weighted outputs from the previous layer with an added bias term. The above process is described as the following equation:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b_i\right) = f(x) \tag{3.1}$$

where f is the activation function, x_i are the input values, w_i are the weights

There are many options for an activation function, which depends on the tasks' needs. For example, to do a binary classification with predicted classes 0 and 1, the output should be a probability value between 0 and 1. A Sigmoid function could be used to tackle this problem as shown in Figure (3.2) and where the sigmoid function is expressed as the following equation:

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.2}$$

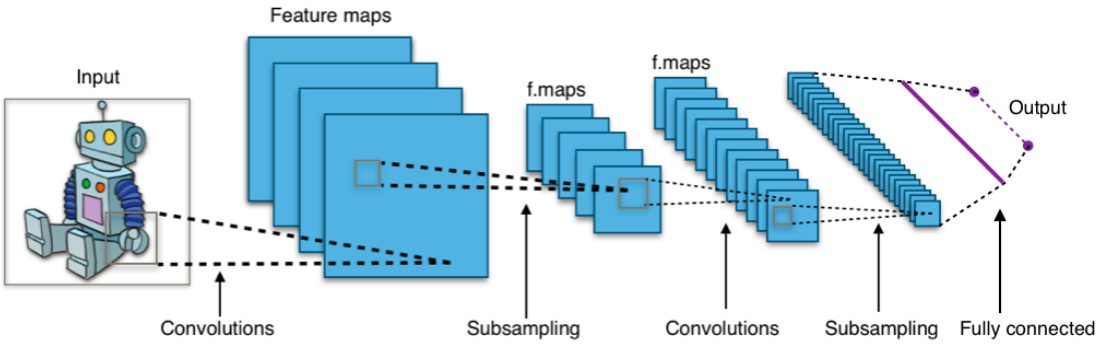

Figure 3.3: Typical CNN architecture for image recognition (Wikimedia, 2015)

Thus, the following layers pass the information in the same way. The output of neuron i in layer l becomes :

$$y^l = f^l(y^{l-1}) = f^l \circ f^{l-1}(y^{l-2}) = ... = f^l \circ f^{l-1} \circ ... \circ f^l \circ f^1(x) \tag{3.3}$$

After the information is passed, a final result is produced by the model. In order to make the prediction match the desired ground truth label, one has to select suitable parameters. One initializes the weights usually randomly with some constraints as discussed for example in He et al. (2015b) before one optimizes them. The optimization will be briefly discussed in the Section 3.4.1.

Based on the a basic NN framework as introduced above, several different types of NNs have been developed for different tasks. In this thesis, only Convolutional Neural Network is introduced in the following Section 3.2.

3.2 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a class of NN that, as previously mentioned, is often used to analyse images due to its ability to accurately differentiation between image features. CNNs emerged from the study of the brain's visual cortex, and have been used in image recognition since the 1980s (Géron, 2019). CNNs are able to capture the low-level features (eg: texture, edges) from input images with convolutional layers (as shown in Figure 3.3). Pooling layers reduce the size of the data output from a previous layer and computations so that it is easier to process further. Also, pooling layers extract the dominant features from the images. The dense or fully-connected layers perform the final step in the CNN. If the images are very simple and formatted, a FFNN may achieve almost the same performance/accuracy as a CNN. However, if the image is complex and not that well-formatted, CNNs will perform better because they are translationally invariant. Moreover, a FFNN would 'pass away' (break down) for large ice crystals due to the large number of neurons it requires to make good predictions (Ullah and Bhuiyan, 2018). Thus, a CNN is a better tool for analyzing complex images than a FFNN.

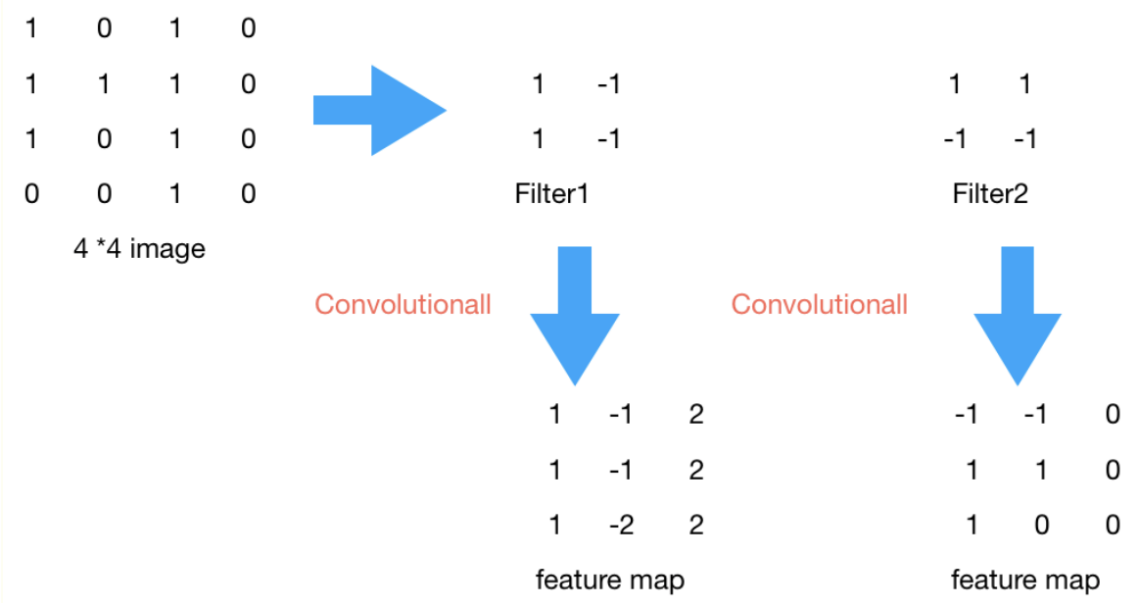

Figure 3.4: Calculation process of Convolutional layer to get feature map (*convolution network basics - Charlotte77* 2019)

As described previously, a CNN is generally composed of three main parts, convolutional layers, pooling layers and fully-connected layers. In the following sections, how these three components work in detail is introduced.

3.2.1 Convolutional layer

The convolutional layer is the most important building block of a CNN. The main objective of the convolutional layer is to extract low-level features(i.e., edges, color) from the input image, by scanning through the entire image with a small filter. This means that the convolutional layer computes the output by applying the kernel (filter) to an input array. Neurons in the first convolutional layer are not connected to every single pixel in the input image, but only to pixels in their receptive fields(Géron, 2019). Conv2D layers are used in our model, which means that the input of the convolution operation is three dimensional. The "2D" in "Conv2D" actually means that the filter moves through the image in two dimensions. For example, for each 4×4 pixel region of the image, the convolution operation computes the dot products between the values and the weights that are defined in the filter. As can be seen in Figure 3.4, the original image is a black and white image represented by the pixel values of 1 or 0, respectively. For this 4×4 image, two convolution kernels of 2×2 are used. The convolution process starts by taking the dot product of the filter with the 2×2 submatrix in the top left corner of the image. At this position, the step size is set to 1, which means that the next submatrix of the image sent to the filter is shifted 1 column to the right. When the right side of the image is reached, the process is repeated on the subsequent row. Taking the first convolution kernel Filter1 as an example, the calculation of the feature map is as follows:

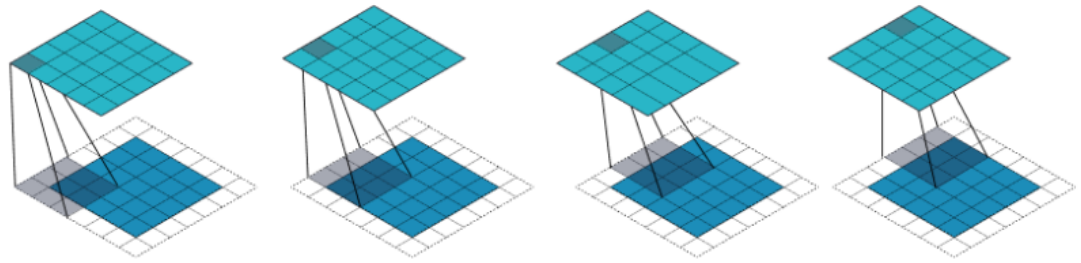

Figure 3.5: Convolution of a 5x5 input (blue) with 3x3 kernel (grey) with a stride of 2 and padding of 1. The feature map output is in green (Ingargiola, 2019)

$$fm1(1,1) = 1*1+0*(-1)+1*1+1*(-1) = 1 \tag{3.4}$$

$$fm1(1,2) = 0*1+1*(-1)+1*1+1*(-1) = -1 \tag{3.5}$$

$$fm1(1,3) = 1*1+0*(-1)+1*1+0*(-1) = 2 \tag{3.6}$$

The resulting feature map for the entire image for two filters can be seen in Figure 3.4. The process of applying the convolutions can be seen as a sliding a filter over the image. To better capture the features near the edges of the image, padding can be added around the image in the form of rows and columns of zeros, so that the filter can be applied further out 'over' the edges of the image. This is especially useful for larger filters. This process is exemplified in Figure 3.5

Then the input equation of every convolutional layer can be written as:

$$V = conv2(w, x, 'valid') + b \tag{3.7}$$

where w is the filter matrix (weights), x is the input matrix, 'valid' is the type of the convolutional computation which is described below and b is the bias. The output is as follows:

$$Y = f(V) \tag{3.8}$$

where f is the activation function. ReLu was used in this project and Relu is expressed as follows:

$$f(x) = max\{x, 0\} \tag{3.9}$$

It gives an output of x if x is positive and gives 0 otherwise.

3.2.2 Pooling layer

The pooling layer is mainly used to reduce the required computational power of the CNN. It works by reducing the spatial size of the convoluted feature by reducing the output from the convolutional

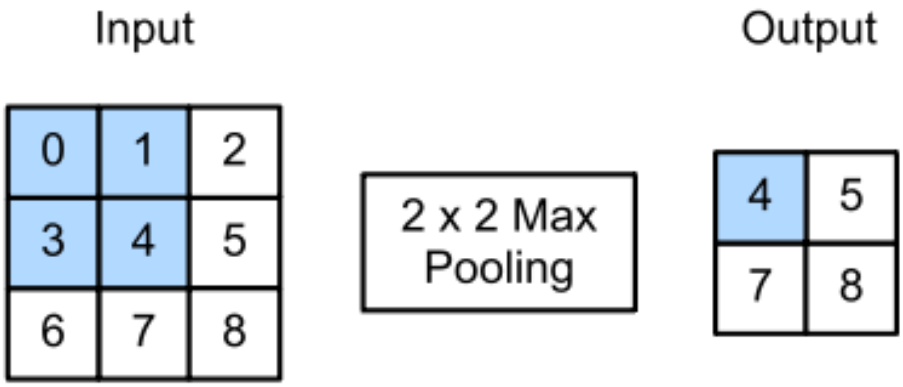

Figure 3.6: Maximum Pooling of a 3x3 input with 2x2 pooling window. The shaded area are the first output element and the input image elements used for the output computation: $max(0, 1, 3, 4) = 4$ (Ingargiola, 2019)

layers. Each neuron in a pooling layer is connected to the outputs of a limited number of neurons in the previous layer. Note that a pooling neuron has no weights, it just aggregates the inputs using max or average. Since the optimization complexity grows exponentially with the growth of dimensions, it extracts the dominant features in an area of the output from the convolutional layers (Géron, 2019). Max pooling is a form of down-sampling and also a noise reduction technique. Max pooling works similarly to convolutional layers but it uses a kernel instead of a filter with a step size larger than 1. Max pooling simply looks at all the values in a given submatrix and selects the largest value as the output. Thus, only the maximum input value in each kernel makes it to the next layer and the other inputs are dropped. Average pooling is another method of pooling that instead, returns the average of all the values in the kernel as the output. Average pooling can be seen simply as a noise reduction technique whereas max pooling also extracts dominant features (Géron, 2019). Thus, max pooling can keep the position and rotation of the feature constant, which is great for image processing. Additionally, it can reduce the number of model parameters and reduce problems with over-fitting. In both cases, the pooling process is shown in the Figure 3.6. Similarly to a convolutional layer, the pooling window starts from the upper-left of the input image (input from previous convolutional layer) and walks through the entire image from left to right and top to bottom, step by step. The output of each location is the the maximum (for max pooling) or average (for average pooling) value of the input subsection (shaded area in the Figure 3.6).

3.2.3 Fully connected layer

Fully connected layers are the last layers in the neural network. The core idea of a fully connected layer is that each neuron in the layer is connected to every neuron in the previous layer as shown in Figure 3.7. The fully connected layers use the flattened output from the last pooling and convolutional output as its input. The purpose of the fully connected dense layers is to perform the classification based on the features extracted by the convolutional layers. A feature vector is a

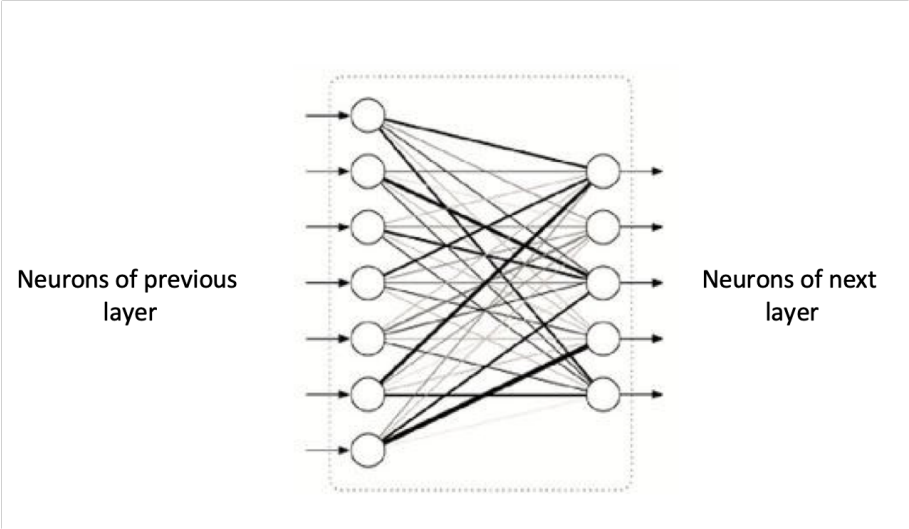

Figure 3.7: An example of Fully connected layer (Raju and Thirunavukkarasu, 2020)

vector of numerical features that describe some object/figures in pattern recognition in machine learning. Here, one could understand the feature vector as the final value obtained from previous convolutional and pooling layers. For a given feature vector x, to determine the probability for each of these categories i (in this case it should be different kinds of ice crystals (eg: column, plates)), $P(y = i|x;\theta)$, then the results of our hypothesis function would be a C dimensional vector whose sum of the vector elements is 1, and represents estimated probability values of these C types. So far, the basic structure of CNN has been introduced but there are various architectures of CNNs available. In this thesis, only Residual Networks and Densely Connected Networks are used, which are introduced in the following Subsection 3.2.4 and 3.2.5.

3.2.4 Residual Networks (ResNet)

For solving complex problem, neural networks are becoming deeper and deeper from a few layers (e.g., AlexNet) to over hundreds of layers(e.g., ResNet-152, DenseNet-264). However, in reality, it is generally difficult to achieve good performance from deep neural networks due to vanishing or exploding gradients. Especially, small gradients can quickly go to zero due to the large number of multiplications in the many layers of these deep neural networks. Thus, He et al., 2015a proposed a deep residual learning framework which properly solved this problem. As shown in Figure (3.8), the main difference between the right (residual block) and left (regular block) schematics is the so-called skip connection. The inputs x can forward propagate directly from the beginning to the next few subsequent layers and this skip connection prevents gradients from vanishing (going to zero).

These residual blocks can be implemented into models as is done in the ResNet (ResNet-18) model structure (see Figure (3.9))

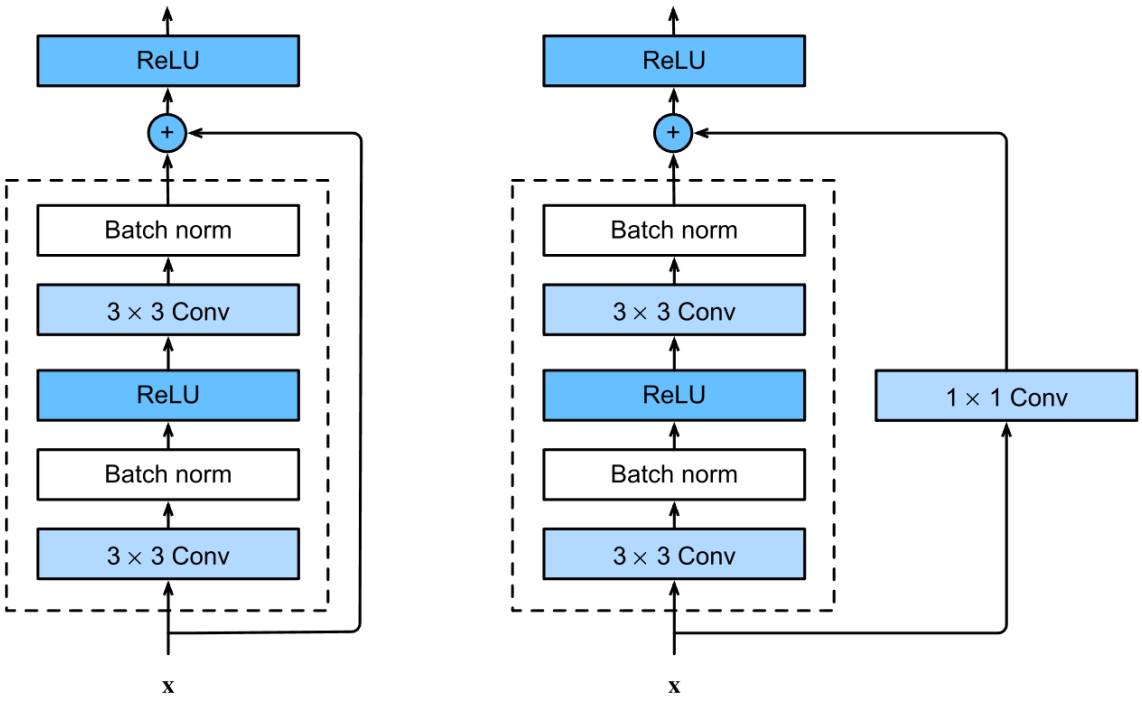

Figure 3.8: Comparison of structure of a regular block (left) and a residual block (right) (A. Zhang et al., 2020)

Therefore, in this thesis, three ResNet models, ResNet-18, ResNet-101 and ResNet-152, are used. The detailed information of each ResBlock is listed in the table A.1 in the appendix.

3.2.5 Densely Connected Networks (DenseNets)

Another method for solving the gradient vanishing problem, is DenseNets (Huang et al., 2018), which is a more advanced way than ResNets that includes more information of higher orders. Thus, the connection becomes a chain and the latter layer has connections with all the preceding layers. An example of a DenseNet connection is shown in Figure:(3.10) As shown in Figure (3.10), each layer of the Dense block concatenate incoming features from all previous layers and contribute to the output feature-maps of its own, to all subsequent layers. This can be expressed as follows:

$$\mathbf{x} \rightarrow [\mathbf{x}, f_1(\mathbf{x}), f_2([\mathbf{x}, f_1(\mathbf{x})]), f_3([\mathbf{x}, f_1(\mathbf{x}), f_2([\mathbf{x}, f_1(\mathbf{x})])]), \ldots] \tag{3.10}$$

By creating such short paths from early layers to later layers, Dense connection strengthens the feature transmission and reuse and thus, alleviates the vanishing-gradient problem.

In this thesis, three DenseNet models, DenseNet-121, DenseNet-169 and DenseNet-201 are used. The detailed information of each DenseBlock is listed in table A.2 in the Appendix A.

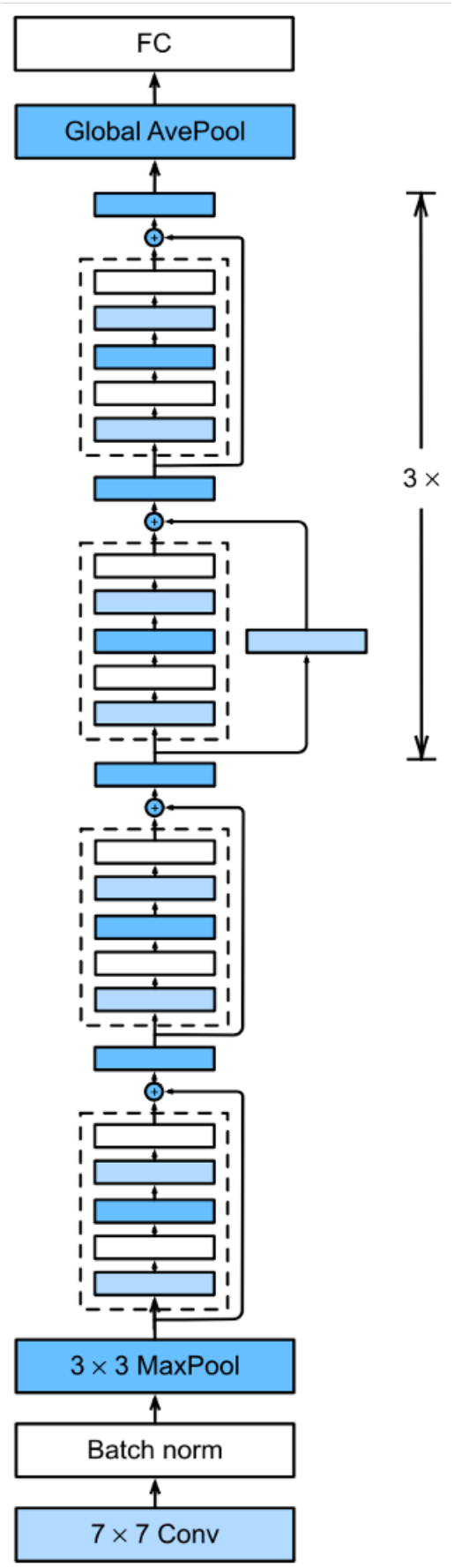

Figure 3.9: The ResNet-18 structure (A. Zhang et al., 2020)

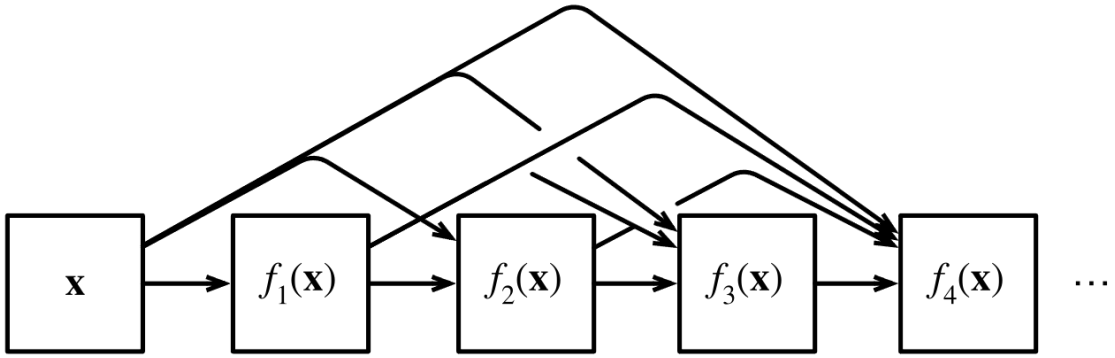

Figure 3.10: Dense connections in DenseNet models (**textcite**)zhang2020dive

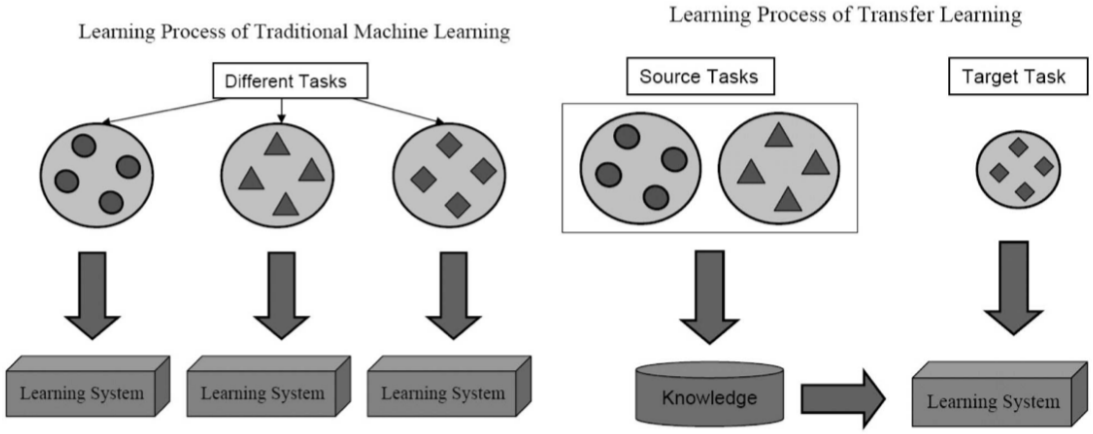

Figure 3.11: Difference between traditional machine learning (left) and transfer learning (right) (Pan and Q. Yang, 2010)

3.3 Transfer learning

When we look back on our learning process as a child, we don't learn everything from scratch. We can always utilise what we already know to help us gain new knowledge. For example, if we know that an apple is a fruit, it will help us identify a pear as a fruit. As another example, it is much easier to learn how to drive a motorbike if you already know how to ride a bicycle. Similarly, if we already understand basic statistics and math, then it is easier for us to learn machine learning. These real world scenarios are the core idea of transfer learning. Basically, transfer learning leverages the knowledge gained from another related task or domain and applies it to a similar problem of interest (Pan and Q. Yang, 2010). The main difference between the learning processes of traditional and transfer learning techniques is that in traditional machine learning the method is to learn each task separately from scratch, while transfer learning utilises the knowledge obtained from previous training for other tasks. An example of this difference is shown in Figure (3.11). Thus, the weights from pre-existing trained models can be used. For example, the DenseNet-121 or ResNet-152 can be used to help the training of a model for our own task, as they are both trained on the ImageNet

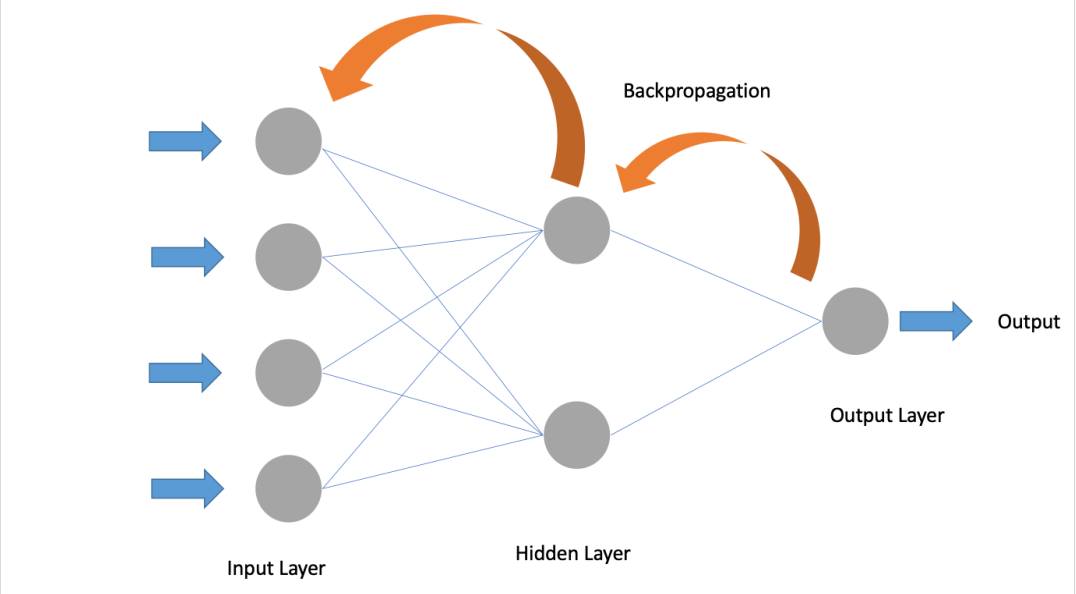

Figure 3.12: An example of a NN with Backpropagation

dataset, which contains around 1.3 million images and 1000 objects spanning over 1000 classes. By using a model already trained on so much data, the generalization error is avoided compared to when training a new model from scratch. Additionally, in situations with a limited dataset where training a model from scratch can be difficult, the use of transfer learning can assist in achieving good model performance.

3.4 Fine-tuning

In the previous section, it was concluded that the weights and structures from existing deep learning neural networks (eg. DenseNet-121, ResNet-18) can be used to develop new models for different tasks. That process is an example of Fine-tuning as Fine-tuning is one of the techniques in transfer learning. Wherein the knowledge gained from models trained on a source dataset (ImageNet dataset) is transferred to the target dataset (Ice crystal images) so that only after minor adjustments, the model obtains good performance. The two main advantages of fine-tuning are that only a small amount of a target dataset is required for training and that it is computationally cheaper than training from scratch.

3.4.1 Backpropagation

Backpropagation is an algorithm to efficiently compute the gradients (by exploiting chain rule on a directed acyclic graph), which then are used in an optimizer to adjust the weights. This is achieved by going backwards from the prediction to the first layer in the network as shown in the Figure 3.12.

Backpropagation computes the gradient of the loss function with respect to the inputs to each layer

and with respect to the parameters of each layer. The gradient of the loss function with respect to the inputs to each layer is used to backpropagate further. The gradient with respect to parameters can be used to reduce the loss on training data, because it is a sufficiently small step in parameters space in direction of the negative gradient, which means if you go towards this direction, the loss can be smaller in this specific parameter space and thus decrease the loss on training data. The algorithms used to minimize the value of the loss function are described in following Section 3.5.

3.5 Optimization

The goal of optimization is to minimize the loss function for deep learning and the following equation presents the optimization of a loss function $J(\theta)$:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \tag{3.11}$$

The value of θ is assigned, such that $J(\theta)$ proceeds in the direction of the negative gradient, which is for a sufficiently small step, the direction of the steepest decrease, and keeps iterating, to ultimately obtain the local minimum value. Where A is the learning rate, which determines how large a single 'step' we can take in the downward direction of the maximum decline of the cost function. This is a special case of optimization and it may be the simplest way to execute Stochastic Gradient Descent (SGD, will be introduced in the following Subsection 3.5.1). In this thesis, two optimization methods for training, SGD and Adam, are tested.

3.5.1 Stochastic Gradient Descent (SGD)

In reality, for training a deep learning network large amounts of data are needed. Therefore, when using traditional gradient descent, the training becomes computational expensive and it becomes hard for the network to converge. Thus, SGD is often used to avoid this scenario. The core idea of SGD is 'randomness'. During the training process, it randomly chooses a small number of points for each iteration rather than looping through all of the points in the entire dataset. Therefore, it greatly reduces the number of computations during deep learning training.

3.5.2 Adam

Gradient descent algorithms with momentum are inspired from physics. Let us imagine rolling a ball in a frictionless bowl. Instead of stopping at the bottom of the bowl, the accumulated momentum keeps the ball rolling back and forth. If the decay rate (the rate that learning rate changes/decays over time with) is set to 0, then it is exactly the same as the original gradient descent. In contrast, if the decay rate is set to 1, then as in the analogy of the frictionless bowl, the ball will continue rocking back and forth, which is not desirable. Therefore, one usually chooses a decay rate around 0.8-0.9, this is like a surface with a bit of friction, so that the ball eventually slows

down and stops. An adaptive gradient algorithm (Adagrad), does not track the sum of gradients like momentum, but tracks the sum of gradients squared, and uses this method to adjust gradients in different directions. However, the main issue of Adagrad is that it is very slow. This is because the sum of squares of gradients only increases but never decreases. Kingma and Ba (2017) solves this problem by adding a decay rate using the Root Mean Square Propagation (RMSProp) method. Therefore, Adam is an optimization algorithm that combines RMSprop (Tieleman and Hinton, 2012) and the SGD method with momentum, as was first proposed by Kingma and Ba (2017). In machine learning optimization, some features are very sparse and thus the average gradient of sparse features is usually small, so these features are trained at a much slower rate. One way to solve this problem is to set a different learning rate for each feature, but this can quickly become messy. The core idea of Adam is that for one feature, the more you've updated, the less you'll update in the future. Thus, it gives other features (such as sparse features) a chance to catch up. The extent to which this feature is updated depends on how far it has been moved in a given dimension, and this extent is measured by the sum of gradient squares. Adam is an adaptive learning rate method rather than SGD, which uses a single learning rate throughout the entire training process. Thus, the weight decay is adapted for different parameters.

3.6 Image augmentation

As previously discussed, by using transfer learning and fine-tuning, the amount of data required to train a model is greatly reduced. However, in some cases, there still is not enough data to just fine tune deep learning models. Additionally, even if there is enough data, the dataset can be too imbalanced for good model performance to be achieved. For example, one dominant class can account for over 50% of the entire dataset. In this situation, it is hard to get a high-performance model.

Thus, image augmentation is an efficient way for addressing these problems. Image augmentation creates a similar but disjointed dataset relative to the original training dataset by randomly changing the original dataset. The changing datatset is under the constraint that after the augmentation, the image should maintain the same ground truth label. By doing so, image augmentation increases the size of the training data set while maintaining the features of the original training dataset. This is typically conducted by performing the following augmentations to images in the original dataset:

1. Image Flipped: horizontally(Figure:3.13)
2. Image Flipped: vertically (Figure:3.14)
3. Rotation (Figure:3.15)

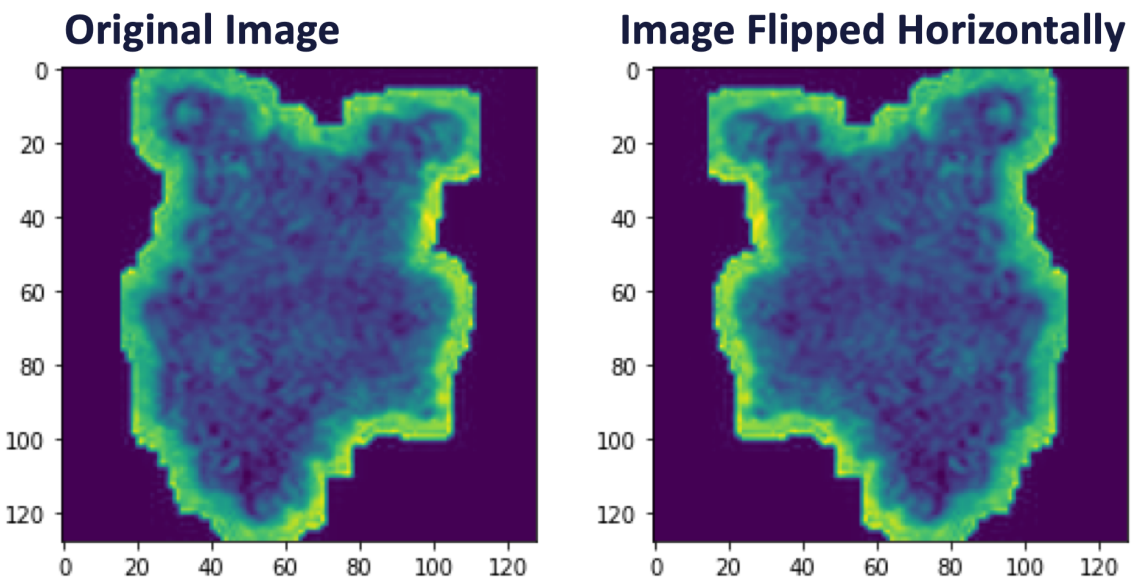

Figure 3.13: Example of ice crystal image flipped horizontally. Original image (left) and horizontally flipped image (right)

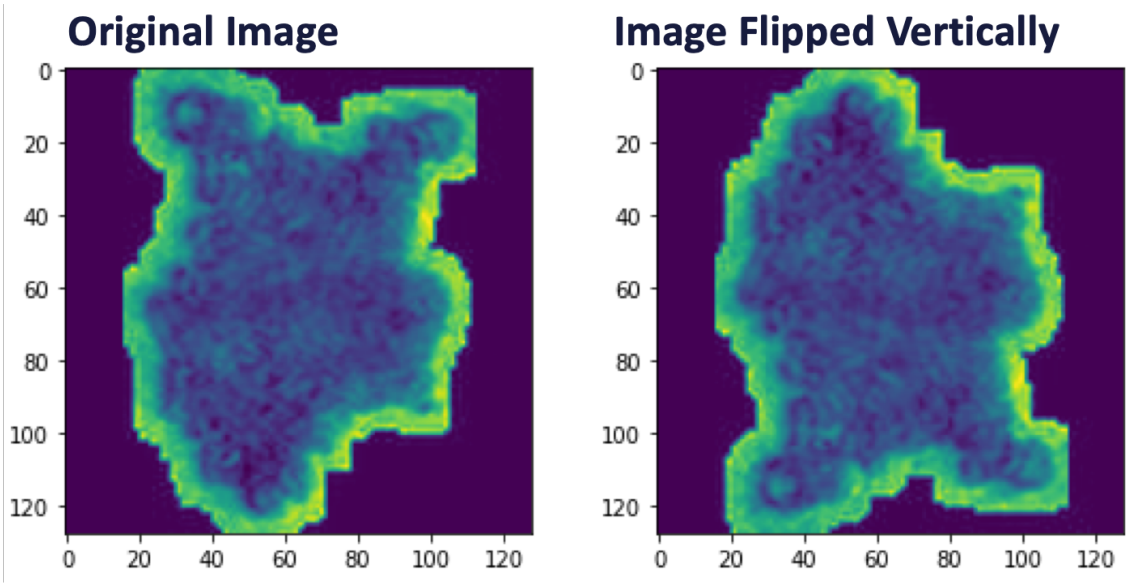

Figure 3.14: Example of ice crystal image flipped vertically. Original image (left) and vertically flipped image (right)

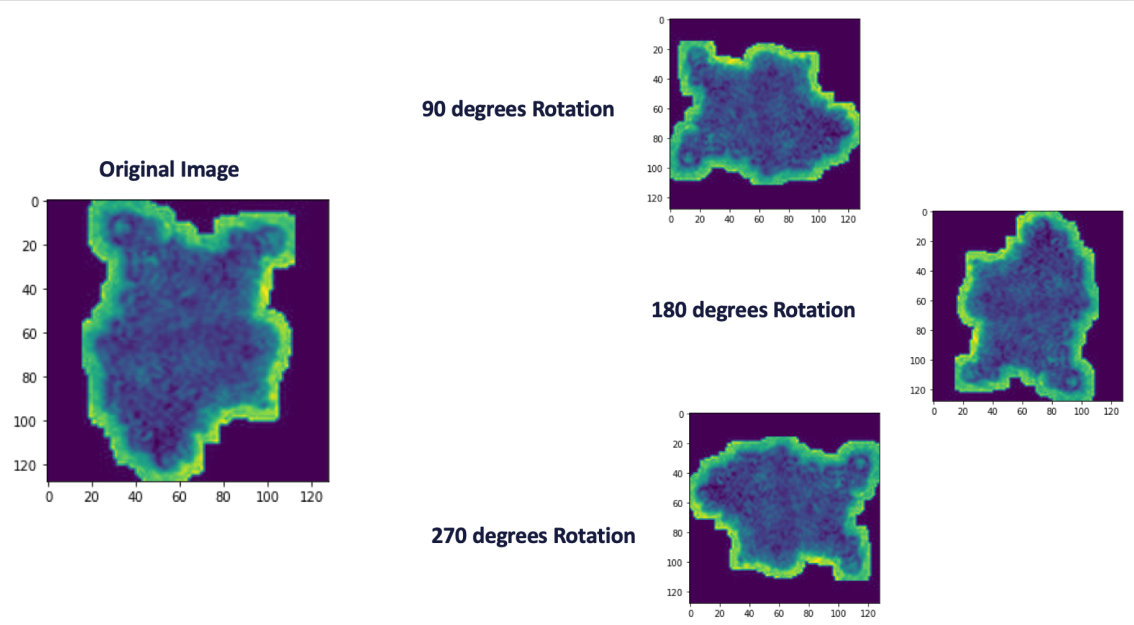

Figure 3.15: Example of ice crystal image rotation for 90, 180 and 270 degrees rotation

4. Cropping: an image can be cropped such that the interesting components of the object are emphasised or such that the target (eg: in our case, ice crystal) is shown in different positions.

5. Changing Colors: the image color can be adjusted from four aspects: brightness, contrast, saturation, and hue.

The augmentation method used depends on the task. For example, in our case, we don't care about the color of ice crystals and our original ice crystal images are black-white. Thus, changing color is not a useful method in our case. Thus, rotating or flipping is more beneficial.

3.7 Evaluation metrics

3.7.1 Confusion matrix

A confusion matrix is an error table that shows the performance of a classification model (or "classifier") for at least two classes from a given dataset, which compares the predicted classes by the model to the true classes. Below, is an example of a confusion matrix for a given classification model that is trained to predict whether a value (class) is positive (P) or negative (N): (3.16):

True positive rate: $(\text{TPR}) = \frac{TP}{TP+FN}$ (worst value $= 0$; best value $= 1$)

True negative rate: $(\text{TNR}) = \frac{TN}{TN+FP}$ (worst value $= 0$; best value $= 1$)

Positive predictive value: $(\text{PPV}) = \frac{TP}{TP+FP}$ (worst value $= 0$; best value $= 1$)

Negative predictive value: $(\text{NPV}) = \frac{TN}{TN+FN}$ (worst value $= 0$; best value $= 1$)

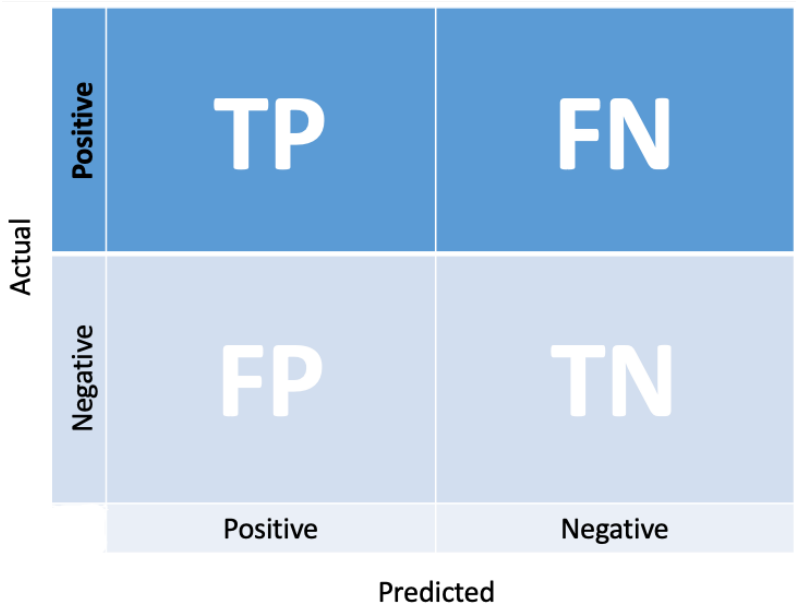

Figure 3.16: Confusion matrix for binary classification.

where TP or true positive, represents the instances when the model predicts that the class is positive and indeed, the actual class is positive. Similarly, TN or true negative, represents the instances when the model correctly predicts that the class is negative. In contrast the FP or false positive, represents the instances when the model predicts that the class is positive but the actual class is negative. Similarly, the FN or false negative, represents the instances when the model incorrectly classifies a class as negative when the actual class is positive.

3.7.2 Overall accuracy

The overall accuracy gives an overview of how a classification model performs. It is usually expressed as a percent, from 0% accuracy, where the model was completely wrong, to 100% accuracy, where the model predicted all of the targets correctly. The equations can be expressed as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.12}$$

The overall accuracy gives quick and intuitive feedback on how a model performs over the entire dataset. However, if a dataset is imbalanced, meaning that the classes are not equally represented in the dataset, the overall accuracy can be misleading as it may just indicate that the model is predicting the dominant classes correctly but not the rarer classes. Especially if the rarer classes are significantly underrepresented in the dataset and their correct classification is statistically insignificant.

3.7.3 Overall false discovery rate

The overall false discovery rate (FDR) is the opposite of the overall accuracy. It is expressed as a ratio of the number of the objects that the model predicted wrong to the total number of objects that the model predicted(include both model predicted wrong and right). Thus, it can be expressed as follows:

$$\text{FDR} = 1 - \frac{TP + TN}{TP + TN + FP + FN} \tag{3.13}$$

3.7.4 Per-class accuracy

For better understanding how deep learning classification model performs in each single class, a per-class accuracy metric to evaluate the model is required. Per-class accuracy accounts for the accuracy of the model for a given class and the number of instances the class occurs. For a binary classification, the equations are as follows:

$$\text{AccuracyPositive} = \frac{TP}{TP + FP} \tag{3.14}$$

$$\text{AccuracyNegative} = \frac{TN}{TN + FN} \tag{3.15}$$

3.7.5 Per-class false discovery rate

The per-class FDR is the opposite of the per-class accuracy, which follows the same fashion as overall accuracy and overall FDR. Thus, the per-class FDR represents the false rate (ratio of false prediction to whole class dataset) of each class. For a binary classification, the equations are as follows:

$$\text{FDRPositive} = 1 - \frac{TP}{TP + FP} \tag{3.16}$$

$$\text{FDRNegative} = 1 - \frac{TN}{TN + FN} \tag{3.17}$$

3.7.6 Balanced accuracy

Balanced accuracy is a metric to evaluate the performance of a classification model, especially for imbalanced dataset. This often happens in ice crystal habit datasets as ice crystals retain the information of the environment in which they grew in and all subsequent processes that they undergo until their point of measurement. Due to these processes, ice crystal habit datasets are frequently dominated by a single habit or are irregular shaped and thus the dataset is often unbalanced.

Balanced accuracy is based on two more commonly used metrics: sensitivity (true positive rate) and specificity (true negative rate). The formulas for true positive rate and true negative rate have already been mentioned above. Balanced accuracy is simply the arithmetic mean of these two as follows:

$$\text{BalancedAccuracy} = \frac{TPR + TNR}{2} \tag{3.18}$$

Balanced accuracy represents the performance of a classification model when a dataset is imbalanced. However, in reality, the importance of each class needs to be considered, such that the evaluation is unbiased.

3.7.7 Class-wise accuracy

Base on the balanced accuracy, class-wise accuracy takes the frequency of each class into consideration as follows (Grandini et al., 2020):

$$\text{Balanced Accuracy weighted } = \frac{\sum_{k=1}^{K} \frac{TP_k}{\text{Total}_{\text{row}_k} \cdot w_k}}{K \cdot W} \tag{3.19}$$

where w_k is the weight of class k. This metric takes care of both tracking the performance of each class and considers the importance of each class. One should notice that balanced accuracy is a special case of class-wise accuracy for equal weights and 2 classes. Thus, for imbalanced datasets, the class-wise accuracy is more representative for understanding the model performance for all classes in the dataset.

CHAPTER 4

Implementation

In this chapter, the data, its pre-processing and the implementation details of the CNN are introduced.

4.1 Data

The two data used in this thesis for training the ice crystal habit classification model and validating the trained model, are both from the Fall 2019 portion of the NASCENT campaign, which took place in Ny-Ålesund, Norway. The ice crystal images were recorded by the balloon-borne holographic imaging platform HoloBalloon Ramelli et al., 2020. HoloBallon measures cloud particles between 6 μm and 2 mm. The cloud particle images were automatically classified into artifacts, cloud droplets and ice crystals as mentioned before in Section 2.3 and only ice crystals were used here.

As described in section 2.3, the ice crystal images are reconstructed from the holograms using the HOLOsuite software (J. P. Fugal and Shaw, 2009). After reconstruction, the ice crystals are stored as 2D complex images where each pixel is represented by a complex number. Although during reconstruction, the amplitude and phase information of a particle is obtained, here we only use the amplitude images to train the CNN (Examples of amplitude and phase images are as shown in the Figure 4.2). The values representing each pixel of the amplitude image ranges from 0 to 255.

The original dataset (the one for training) included 16,259 ice crystal images, which were hand-labeled into 9 classes, according to the needs for this campaign. The classes are: 'Column', 'Plate', 'Lollipop', 'Aggregate', 'Irregular', 'Frozen droplets', 'Small ice', 'Rimed' and 'Column plate'. The details of the dataset are shown in Table 4.1. Also, some example images for each class are shown in Figure 4.3

The dataset for validation, NEWTEST, which was also collected during the fall portion of the NASCENT campaign. The new ice crystal images were also recorded by HoloBalloon and subsequently hand-labeled into the same 9 classes as with the training dataset. The number of ice crystals in each class is listed in the Table 4.2.

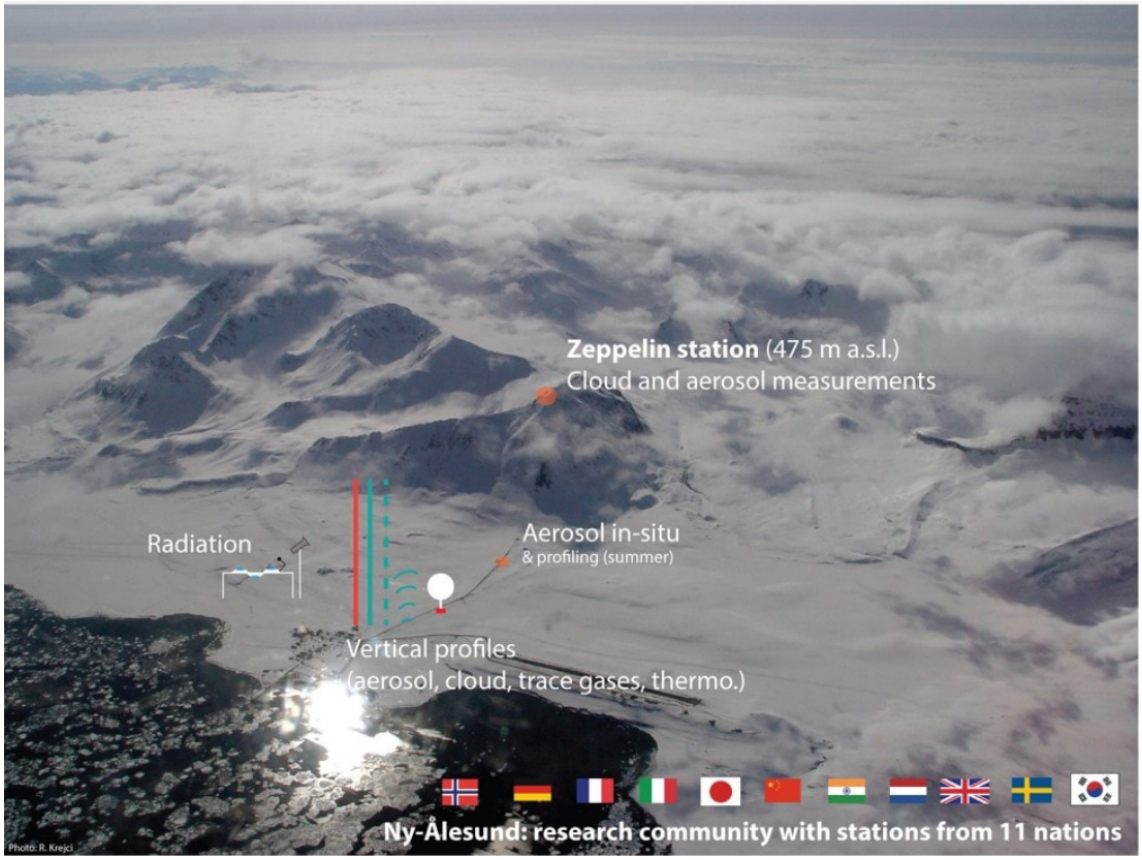

Figure 4.1: Position of HoloBalloon during the NASCENT campaign. Figure taken from *The Ny-Ålesund Aerosol Cloud Experiment (NASCENT) 2019-2020* n.d.

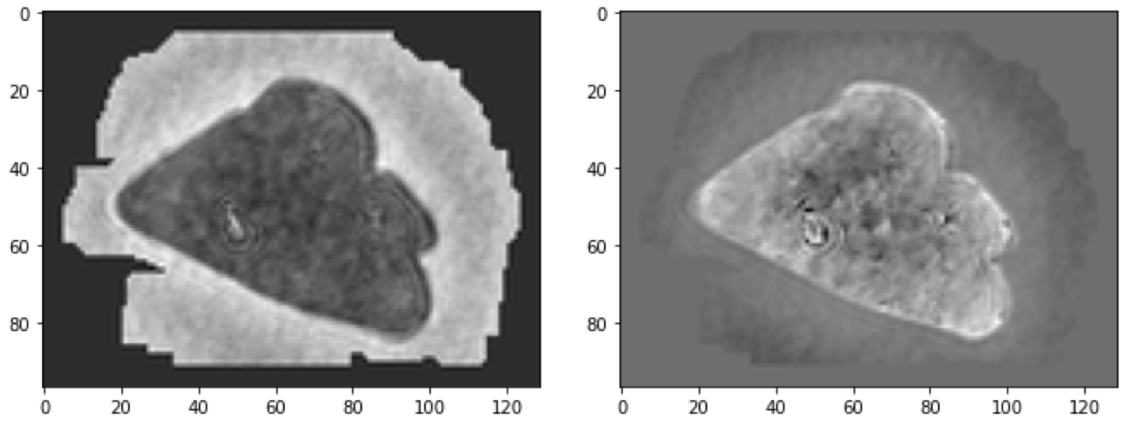

Figure 4.2: Example of the amplitude (left) and phase (right) information from a reconstructed ice crystal using HOLOsuite

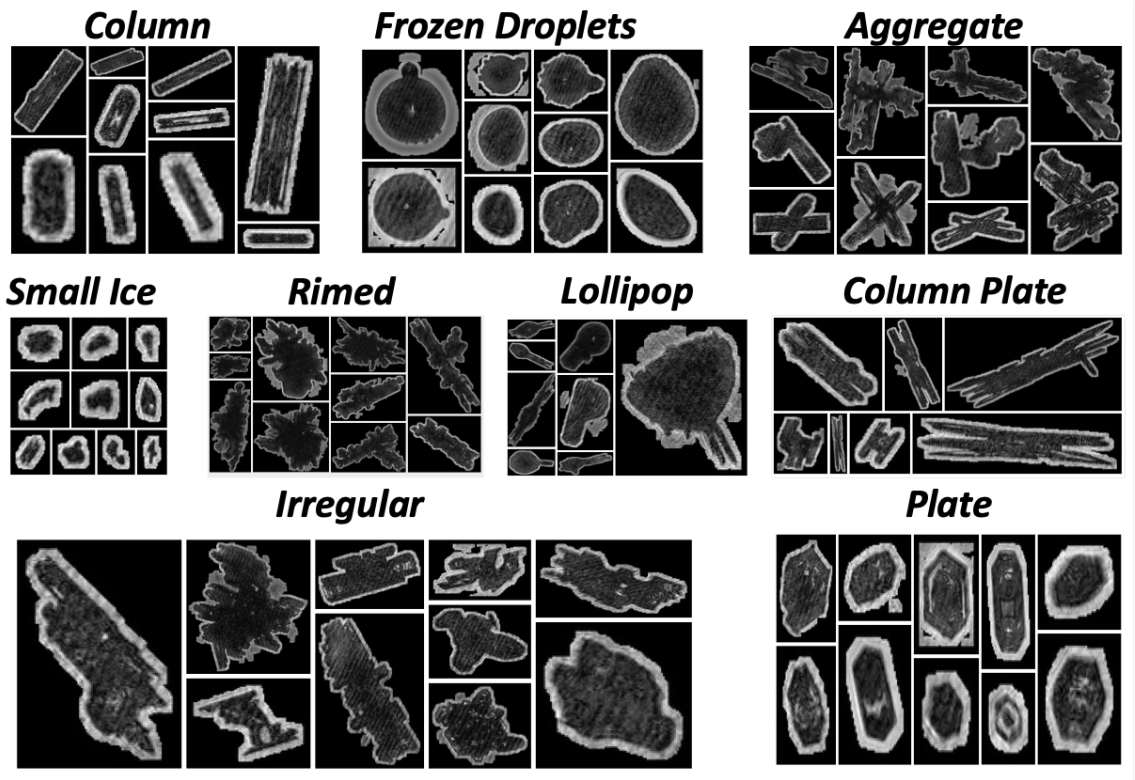

Figure 4.3: Example images of ice crystals separated into the nine habit classes.

Table 4.1: Information about the original dataset

Category	Number of images	Description
Column	9087.0	Columnar ice crystal
Plate	239.0	Plate-like ice crystal
Lollipop	201.0	Look like a frozen droplets with a column in the middle
Aggregate	1934.0	Composed of two or more ice crystals stuck together
Irregular	684.0	A single ice crystal with a complex shape that is irregular
Frozen droplets	688.0	Cloud or drizzle droplets that have frozen
Small ice	416.0	Ice crystals that are usually smaller than 75 μm and indistinguishable
Rimed	1548.0	Ice crystals which contain a rimed boundary
Column plate	1462.0	Ice crystal that contains both columnar and plate-like features, often resemble an 'H'

Table 4.2: NEWTEST dataset: ice crystal numbers in each class

Category	Number of images
Column	83
Plate	100
Lollipop	2
Aggregate	154
Irregular	418
Frozen droplets	66
Small ice	113
Rimed	391
Column plate	25

4.2 Framework, Structure and Implementation

In this section, the data pre-processing, structure of the deep neural networks and the implementation of the neural network for ice crystal habit classification are introduced.

4.2.1 Data Preprocessing

As we introduced in section 2.4, the dataset for training the deep-neural network consists of in-focus 2-D holographic images from HoloBalloon. The size of the ice crystal images ranged from 17 to 807 pixels, corresponding to ice crystals with maximum dimensions between 51 μm and 2.4 mm. The original images extracted from the HOLOsuite software are all in-focus and tightly cropped, there are no boundary areas (black boundary) on the four sides of the images (as shown in the image on the left side of Figure 3.5). Therefore, to ensure that the entire ice crystal image is surrounded by some black pixels, the ice crystal images are padded by 10 black pixels on all sides of the image as shown in Figure 3.5. This ensures that the edge features are captured equally around the ice crystal.

People learn to classify ice crystal habits from general shapes, textures, symmetry etc. As for Neural networks, it is not so obvious as they catch features from the numbers representing each pixel. As such, the neural network sees an array of pixel values ranging from 0 to 255 that correspond to a given color. However, if the variation in pixel values spans too many numbers, it can cause issues for the CNN during training. Thus, the pixel values are standardised to between -1 (black) and 1 (white). Additionally, to reduce the size of the input image and ultimately save computational time during training of the CNN, all of the ice crystal images are resized to 128 $\times$ 128 pixels, such that key features of the ice crystal shape are retained without losing too much information.

As discussed in section 3.6, image augmentation is a useful tool for augmenting limited and/or imbalanced datasets. Thus, we randomly flip (horizontally and vertically) and rotate (180 degree)

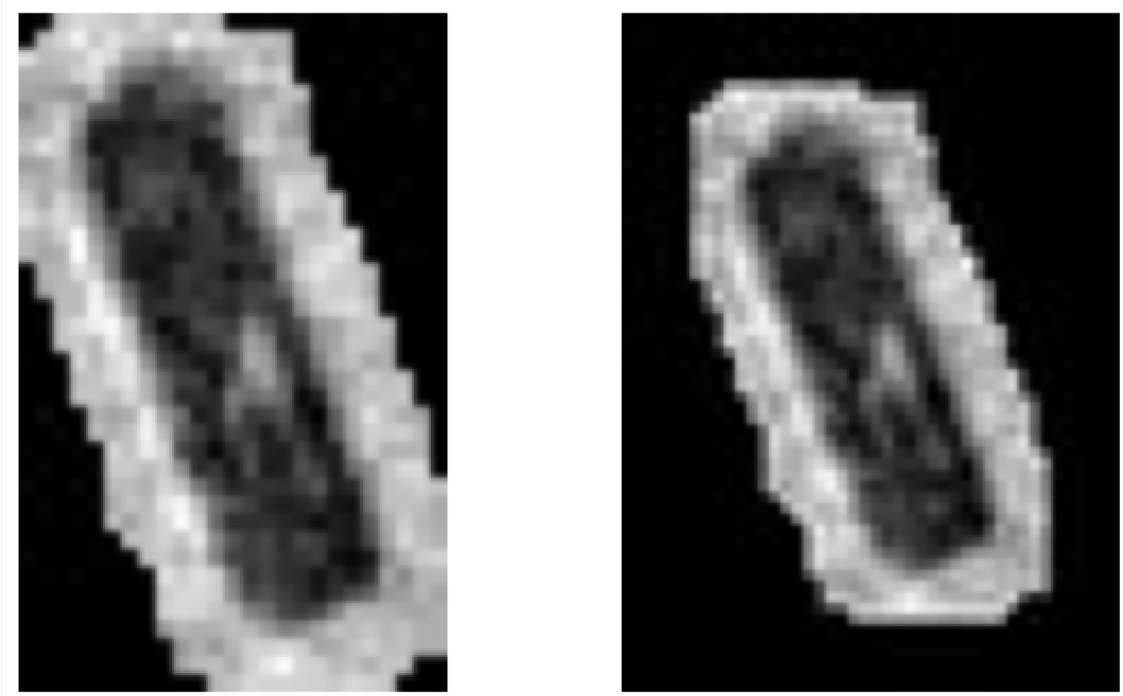

Figure 4.4: Adding 10 pixels boundary on each side of the images. Original image (left), image added boundary (right).

the images for the training dataset for all the tests in this thesis.

4.2.2 Train/Test/Validation Set Splitting

In this thesis, we divided our original ice crystal dataset, which included 16,259 ice crystal images, into a training set, a validation set, and a test set with a ratio of 7:2:1, respectively, as shown in the Figure 4.5. This division is repeated 10 times, such that the test sets are disjointed for each of the 10 splits. For each single model, the test set is 1/10 of the overall dataset and is disjointed from the validation and training sets. Therefore, for each training, we can get a model ensemble with 10 members (10 slightly different models). The performance of this model ensemble is evaluated by going through all 10 members on their corresponding test sets. In the following Chapter 5, all the changes to the dataset such as rebalancing or balancing (described in the Section 5.2 and 5.3), are only done on the training and validation sets of the dataset. The test set always remains the same such that any changes on the dataset does not influence the model performance comparison between training with the original dataset and changed (rebalanced or balanced) dataset.

4.2.3 Experiments

As shown in Figure(4.6), we use the pre-trained model (DenseNet121) as an example to describe the training process. At the bottom is the input data, a 128 $\times$ 128 ice crystal image. Then the pre-trained DenseNet121 model conducts feature extraction over 4 dense blocks on the input image.

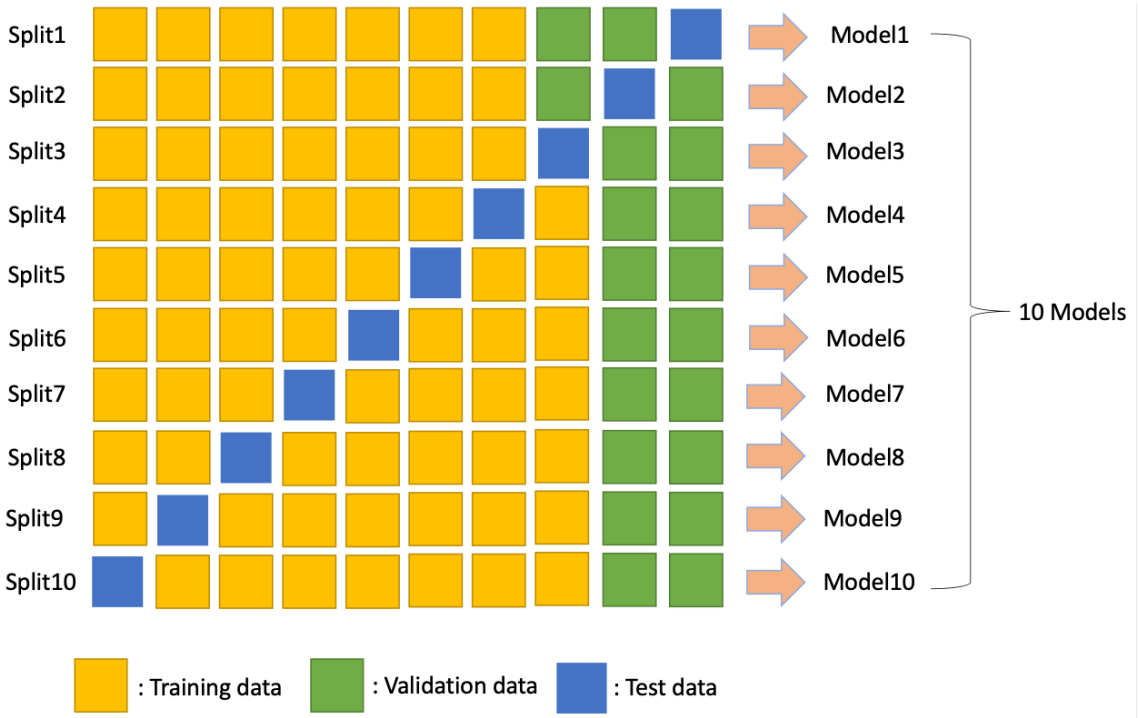

Figure 4.5: Train, validation and test dataset split. During rebalanced and balanced training, the training and validation sets are resampled to achieve class balancing. However, the test set is always used unmodified in order to report comparable scores.

This DenseNet121 model includes 121 layers in all and the last layer is the classifier. The classifier then predicts the input image into one of the following nine classes: 'Column', 'Plate', 'Lollipop', 'Aggregate', Irregular', Frozen Droplets' , 'Small Ice', 'Rimed', and 'Column plate'. In this example (see Figure(4.6), the input is a 'plate' and the CNN correctly classifies the image as 'plate', as can be seen by the plate category encompassed in the red rectangle. The model is trained with batch sizes of 32 and 64 images for the validation set and training set respectively. Batch size is the number of images in each small group.

The results of the experiments are presented in Chapter 5.

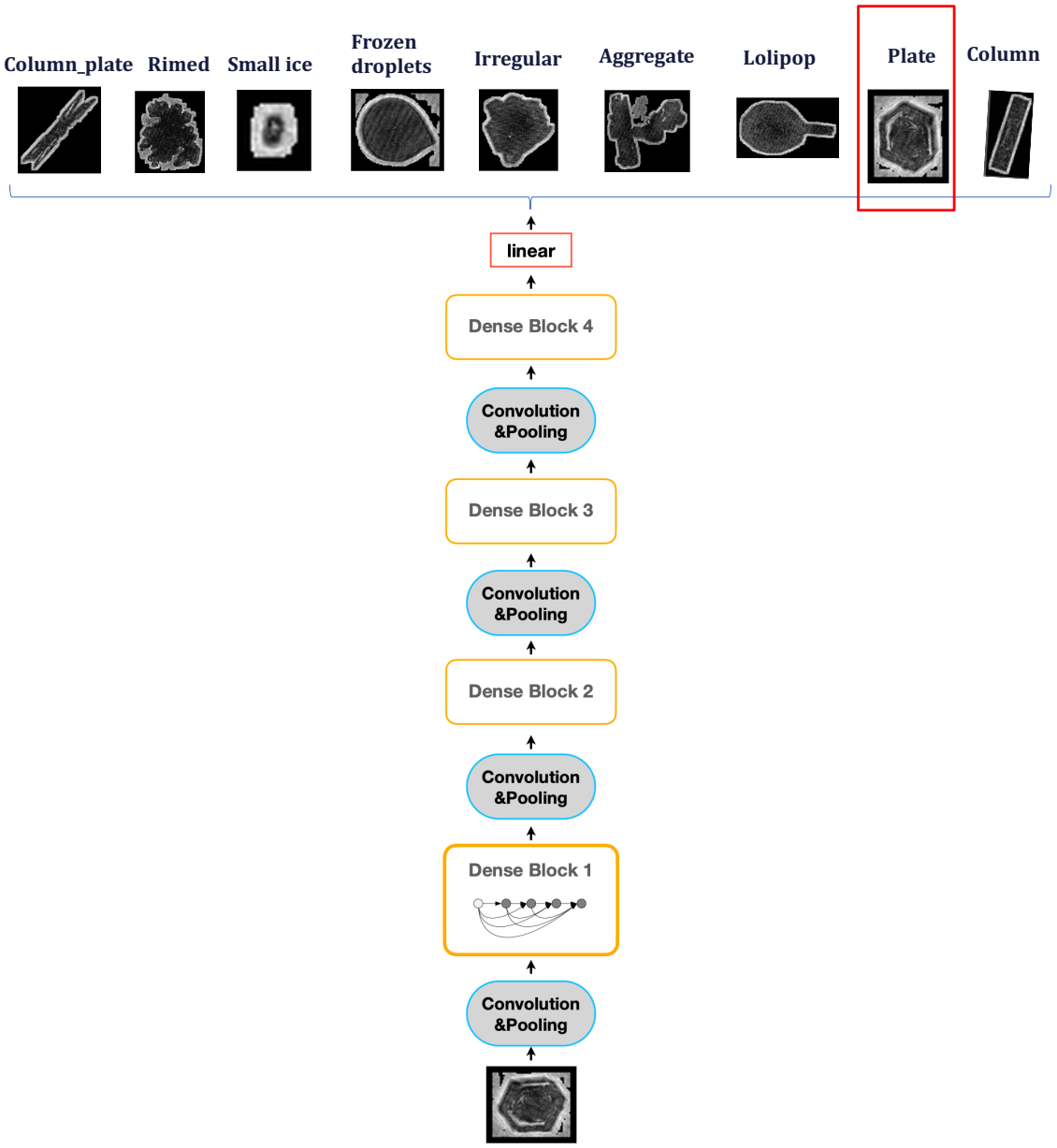

Figure 4.6: Densenet121 Architecture

CHAPTER 5

Results and Discussion

In this Chapter, the results from the training and evaluation of six different pre-trained models (DenseNet-121, DenseNet-169, DenseNet-201, Resnet-18, Resnet-101, and Resnet-152) with fixed learning rates and optimisers using the original dataset are discussed in Subsection 5.1.1 and 5.1.2. Additionally, the impact of freezing the lower half of the model layers during the training of the two best performing models and training with a very small learning rate (0.00001) are also investigated in Subsection 5.1.3. The influence of including a physical attribute in the training, such as the aspect ratio, is discussed in Subsection 5.1.4. Afterwards, the impact of training the two best models on a rebalanced and balanced version of the original dataset with and without Test-Time Data Augmentation are described in Section 5.2 and 5.3 respectively. Finally, the best performing model is applied on another new dataset, NEWTEST, which is disjoint with training, test and validation sets, to investigate the application value of this model.

5.1 Original (Unbalanced) Dataset

The original dataset is the data with training, validation and test set unchanged, directly from Subsection 4.2.2. In this section, the performance of 6 pre-trained models(eg:DenseNet121) with different optimizers (eg: SGD) and learning rates for ice crystal classification are tested. Additionally, the impact of fine-tuning when the lower half of the layers of the neural network are frozen, is evaluated. Moreover, the importance of adding physical information (e.g., aspect ratio) to assist the neural network training is investigated.

5.1.1 Pre-trained model tests

Different pre-trained model structures and their weights trained on the ImageNet dataset are used as base models, as described in Section 3.3. The structure of the pre-trained model is used as the base structure of the model to be trained on the ice crystal dataset; The parameters of the pre-trained model are transferred to the model to be trained as the initial parameters. Then the model is trained on my tasks (ice crystal classification) and dataset (ice crystal images) and evaluated. The model is retrained to adapt the weights for the ice crystal images using a fixed learning rate of 0.005

Table 5.1: Pre-trained model tests

Optimizer	Learning_rate	Model	Overall	Class-wise
SGD	0.005	ResNet-101	0.8317	0.7049
		ResNet-152	0.8281	0.6937
		ResNet-18	0.8316	0.6942
		DenseNet-169	0.8308	0.7009
		DenseNet-201	0.8293	0.7050
		DenseNet-121	**0.8310**	**0.7073**

Table 5.2: Optimiser method tests

Model	Learning_rate	Optimiser	Overall	Class-wise
Densenet121	0.005	**SGD**	**0.8320**	**0.7074**
		Adam	0.7852	0.5806
	0.0001	SGD	0.8157	0.6589
		Adam	**0.8329**	**0.7014**

and the SGD optimizer. The pre-trained models evaluated were DenseNet-121, DenseNet-169, DenseNet-201, Resnet-18, Resnet-101, and Resnet-152. Their structures are as described in the table (A.2) and table (A.1) respectively. As introduced in Section 4.2.2, for each pre-trained model choice, a set of 10 models is obtained. Thus, the final results of overall accuracy and class-wise accuracy for this set of models is the average over 10 models on its corresponding test sets.

As shown in table (5.1), the overall accuracy for all of the models is around 83%, while their class-wise accuracy is around 70%. The difference between the overall accuracy and the class-wise accuracy means that the performance of the models on different ice crystal classes varies significantly. Using the results from this, I selected the relatively best pre-trained model, DenseNet-121, which has an overall accuracy of 0.831 and a class-wise accuracy 0.7073 to be used in the subsequent tests.

5.1.2 Optimizer method tests

In an attempt to improve the performance of the DenseNet-121 model, I tried training the model with two fixed learning rates and two different optimizers, SGD and Adam. The learning rates were chosen to cover a relatively large span ranging from 0.005 to 0.0001. For the SGD optimizer tests, we set the momentum parameters with a decay of 0.9, to obtain a reasonable weight for the last convolutional layer (Sutskever et al., 2013). Meanwhile for the Adam optimzer tests, no decay was included.

The learning rate and SGD tests yield better performance than the initial pre-trained model test (see Section 5.1.1). In particular, for the large learning rate (fast adjustment speed) of 0.005, SGD

has a better performance than Adam, while for the small learning rate (slow adjustment speed), Adam performs better than SGD (see table (5.2),).

1. DenseNet-121, SGD, learning rate 0.005
2. DenseNet-121, Adam, learning rate 0.0001

Hereafter I use DenseNetFastSGD to describe the combination of the model DensNet-121 trained with a learning rate of 0.005 and the SGD optimizer and DenseNetSlowAdam to describe the model DenseNet-121 trained with a learning rate of 0.0001 and the Adam optimizer.

DenseNetSlowAdam has a higher overall accuracy than DenseNetSlowAdam while DenseNetFastSGD has a higher class-wise accuracy than DenseNetSlowAdam. They both have their strengths. Thus, for the following tests, DenseNetSlowAdam and DenseNetFastSGD are used.

5.1.3 Freezing layers tests

As mentioned in section 3.4, fine-tuning can help save computation time and is 'friendly' to small datasets. Thus, I use DenseNetSlowAdam and DenseNetFastSGD from the optimizer tests and then freeze the lower half of the layers to train the model. From the Figure (5.1) we can see that the lower half of the layers are close to the input data. Thus, these layers are responsible for learning the very basic and rough features of the input images. Therefore, even though the pre-trained model is trained on the ImageNet dataset, it will not be very different from the model trained on the ice crystal dataset as both datasets are composed of images. However, as the upper half of the layers are close to the output, they are responsible for learning the more detailed and unique information of the ice crystal images. Therefore, if one focus on the training of these layers, it may reduce the computation time required for training and obtain better model performance. As these layers include the more detailed information about the ice crystals, I include two additional test groups with a very small learning rate of 0.00001, for both SGD and Adam. As shown in table (5.3), the two groups with a very small learning rate (0.00001) didn't perform well, with both lower than 80% overall accuracy and 56% and 37% class-wise accuracy when using SGD and Adam, respectively. For the two optimal groups, they obtain relatively similar overall and class-wise accuracy as when training all of the layers. These results confirmed our hypothesis that the upper half of the layers contain more detailed and distinct information about the ice crystals and the lower half of the layers only include rough information about the images, which does not significantly influence the model performance.

5.1.4 Additional information tests

As shown in previous research, deep learning can achieve better performance when physical knowledge is included. For example, Zhuo and Tan (2021) used tropical cyclones intensity and

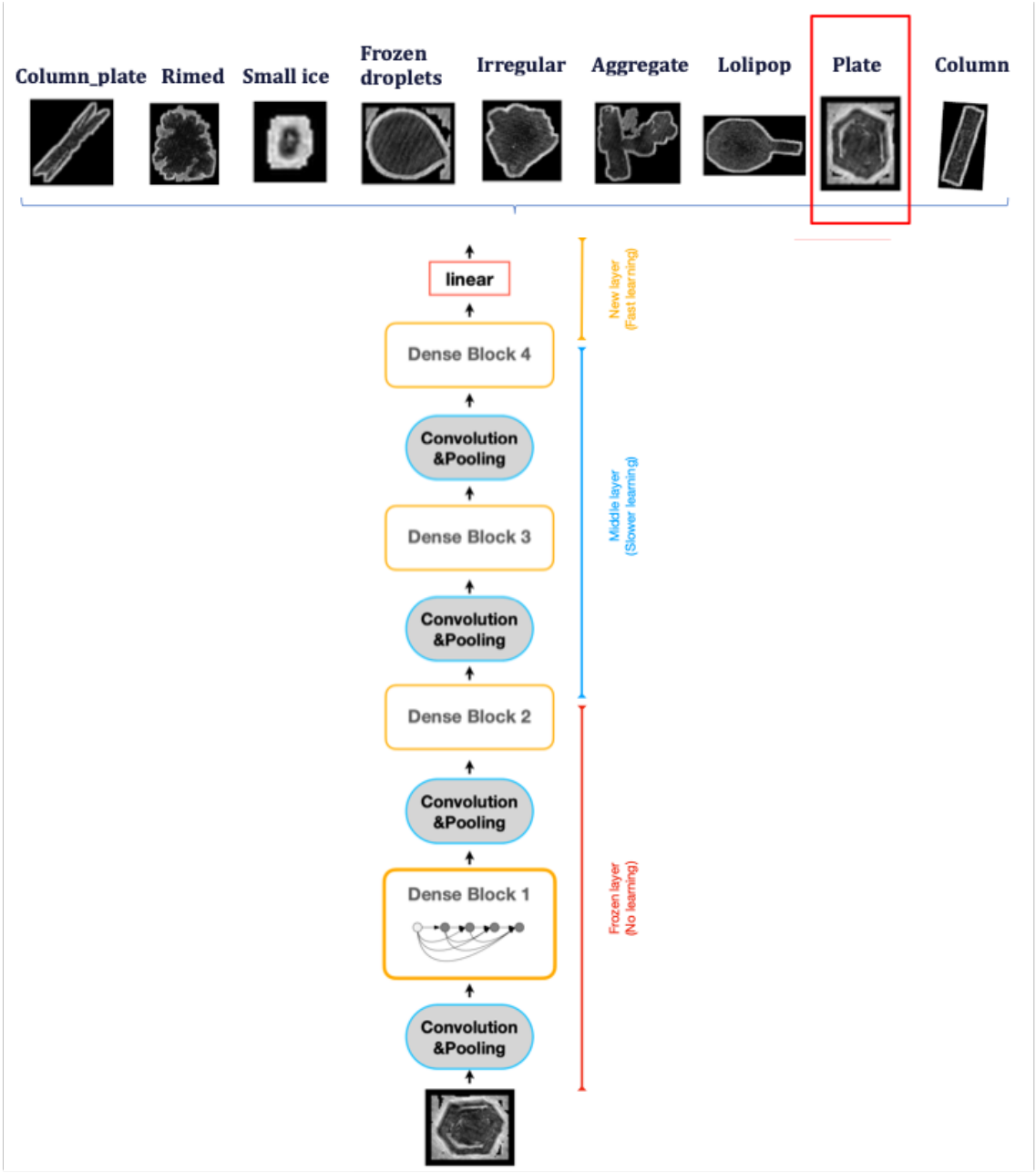

Figure 5.1: Densenet121 Fine-tuning by freezing half previous layers

Table 5.3: Freezing layers tests

Model	Learning_rate	Optimiser	Overall	Class-wise
Densenet121	0.005	SGD	0.8274	0.6843
	0.0001	Adam	0.8279	0.6911
	0.00001	SGD	0.7313	0.3677
		Adam	0.7956	0.5599

Table 5.4: Additional information tests

Model	Learning_rate	Optimiser	Additional Info	Overall	Class-wise
Densenet121	0.005	SGD	Aspect Ratio	0.8307	0.7046
Densenet121	0.0001	Adam	Aspect Ratio	0.8327	0.7039

wind radii to help improve the performance of a deep learning neural network. Thus, in this section, I introduce an additional parameter containing the physical attribute, aspect ratio into the model training. The aspect ratio of an ice crystal is the ratio of the basal and prism faces of ice crystal. As columns, lollipops and column plate ice crystals usually have an aspect ratio larger than 1 while plates and frozen droplets are usually close to 1, this additional information could potentially help with the classification. For rimed, irregular and aggregate ice crystals, their shapes and thus aspect ratios are more random so the aspect ratio information is not expected to improve the model performance for these classes.

When I include the aspect ratio information in the DenseNetSlowAdam and DenseNetFastSGD both overall accuracy and class-wise accuracy have a slight decrease (not more than 1%) (see table 5.4). This indicates that the addition of the aspect ratio did not improve the model performance. Although this is in contrast to our expectations, this could be due to the fact that the aspect ratio information is an inherent part of the images the model is trained on. Therefore, it could be that the model is already learning the aspect ratio as an important feature during classification. Another possibility is that the aspect ratios between the classes where the aspect ratio information is expected to help are not significantly different.

5.1.5 Summary and Discussion

According to the above tests on pre-trained models, optimizers with different learning rates, freezing of the lower half layers of the model and adding physical information, I found that:

1. The differences between the different pre-trained models (e.g., DenseNet-121) are very small and the best model was DenseNet-121.

2. For the optimizer tests with different learning rates, SGD performed best with a learning rate of 0.005 (larger learning rate) while Adam performed best with a learning rate of 0.0001 (smaller learning rate).

3. Freezing the lower half layers of the model slightly degraded the model performance as expected since the majority of the unique crystal features are captured in the upper layers of the model. Thus, even though the parameters and weights of the lower half layers of the model are kept, one can still get a similar model performance as when one adjusts all of the parameters and weights in the model.

Figure 5.2: Confusion matrix for DenseNet121 with SGD optimzer and 0.005 learning rate. Bottom Black Row: 1> White: The number of actual ice crystals in this class (The final box shows the overall number of ice crystals); 2> Green: Per-class accuracy (The final box shows the overall accuracy); 3> Red: Per-class FDR (The final box shows the FDR); Leftmost Black Column: 1> White: The number of ice crystals predicted in this class (The final box shows the overall number of ice crystals); 2> Green: Prediction Per-class accuracy (The final box shows the overall accuracy); 3> Red: Prediction Per-class FDR (The final box shows the FDR). The Boxes in the Middle: The y-axis represents predicted results while the x-axis represents actual results. For example, the second box in the first row means that 49 ice crystals are predicted as column but the actual labels of these 49 ice crystals are plate. The percentage in this box represents the ratio of these ice crystals in the overall 16259 ice crystals.

4. Incorporating physical information such as the aspect ratio didn't improve the model performance.

To determine the reasons why I obtained the above results, I use a confusion matrix to evaluate one of DenseNetSlowAdam and DenseNetFastSGD. For this evaluation I selected DenseNetFastSGD,

which is the DenseNet-121 model trained with the SGD optimizer and a learning rate of 0.005. The overall accuracy of DenseNetFastSGD was 83.20 % and class-wise accuracy 70.74%. As shown in Figure (5.2), the column class has 9087 ice crystals in all, which accounts for 56% of the entire dataset. Only the "column" class is predicted with a near perfect per-class accuracy of 94.73 %. This result suggests that the column class is well predicted due to its frequent occurrence in the dataset and the dataset is very imbalanced. Moreover, for the classes with limited frequency such as "plate" and "lollipop", which contain 239 and 201 images, respectively, the per-class accuracy dropped to 59.83% and 66.67%. Thus, the unbalanced nature of the dataset strongly influenced the learning of the neural network. On the other hand, if we analyze the model performance from the false rate side, it can be noticed that the "aggregrate" and "irregular" classes are miss-predicted into every class and every other class can be mis-predicted as "aggregrate" and "irregular" classes. In particular, the model struggles to discriminate between the "aggregate", "irregular" and "rimed" classes. To further investigate why this occurs, some example images of these classes are presented in Figure (5.3). Even by eye, classifying the example crystals in to the correct classes can be subjective and extremely challenging. This is due to the fact that the "irregular" class is made up of single ice crystals with complex shapes stemming from different cloud processes such as fragmentation and sublimation. Thus, the "irregular" class is by definition composed of randomly shaped ice crystals. The same can be said for the "aggregate" and "rimed" class. As the "rimed" class consists of any underlying habit, including both the "aggregate" and "irregular" class, that is coated by cloud droplets (small bumps at the edges of the particle, see Fig. 5.3), the 'Rimed' class can also be very difficult to classify. Similarly, the 'Aggregate' class, which is composed of images where at least two ice crystals are stuck together, can have very random and irregular shapes.

For 'Frozen droplets', the model performs relatively well with a per-class accuracy of 84.74%.

In conclusion, rare classes, like 'Plate', 'Lollipop' and 'small ice', show much worse performance than the class 'Column'. And this is an objective problem (lack of samples) which can be solved. However, for more subjective issues, like confusing classes 'irregular' and 'aggregate', people even cannot give a consistent answer for all cases, let alone the NN model. Therefore, for further improving the performance of the deep learning neural network, I need to tackle the most serious and objective problem in the datatset, which is the large imbalance of the data.

5.2 Rebalanced Dataset

To solve the imbalance problem in the dataset, I remove 2/3 of the dominant "column" class from the training and validation set, while the number of images for the rest of the classes remain unchanged. The test set is kept unmodified and is the same as in the previous Section 5.1, which means the results from the rebalanced dataset and the original (unbalanced) dataset can be compared.

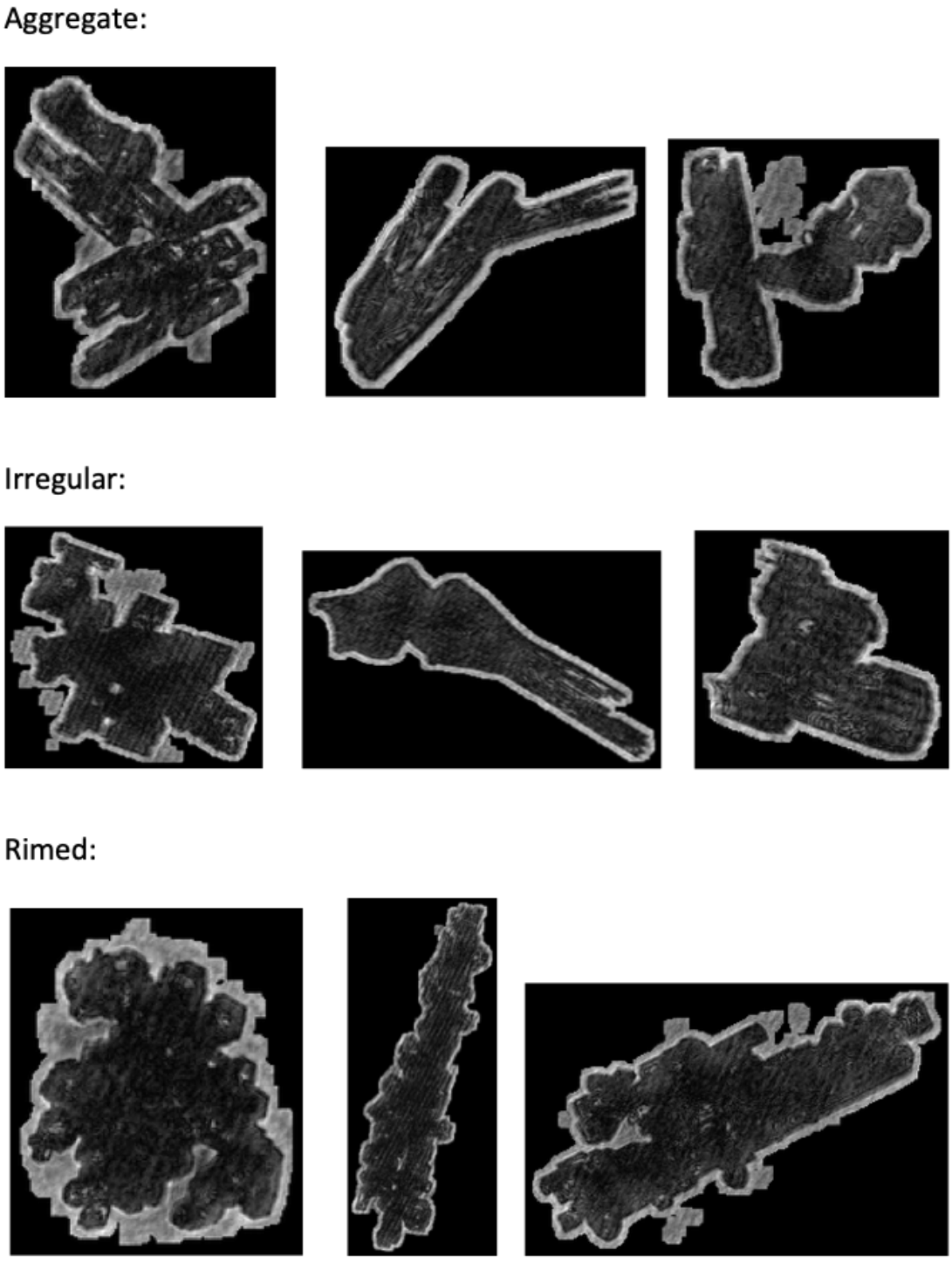

Figure 5.3: Examples of aggregate (top), irregular (middle) and rimed (bottom) ice crystals

Table 5.5: Rebanlanced Dataset

Model	Learning_rate	Optimiser	Augmentation	Overall	Class-wise
Densenet121	0.005	SGD	No	0.8785	**0.8714**
			Yes	**0.8922**	0.8617
	0.0001	Adam	No	0.8905	**0.8780**
			Yes	**0.9014**	0.8683

5.2.1 Non-Augmentation test

To test the impact of rebalancing the dataset, the two model groups (DenseNetSlowAdam and DenseNetFastSGD) from the previous tests are trained. As shown in the table (5.5), for DenseNetFastSGD the class-wise accuracy is significantly improved from 70.74% to 87.14%. Similarly, the class-wise accuracy of DenseNetSlowAdam is improved from 70.14% to 87.80%. By artificially balancing the dataset, I achieve an average increase of 17% improvement in the class-wise accuracy. This is consistent with the discussion in section 5.1.5, which proposes that training with a balanced dataset improves class-wise accuracy. Somewhat unexpected is the improvement in the overall accuracy of the models. It might be that the improvement to the few-sample classes (i.e. lollipop) is much larger than the loss in the large-sample classes (i.e. column) and thus, the overall accuracy is improved. Also, the reduction in the proportion of the large-sample classes may force the model to only learn the dominant features of these classes, which may avoid the overfitting of these large-sample classes. Thus, the composite effect ends up showing an improvement in overall accuracy.

5.2.2 Augmentation test

As introduced in section 3.6, image augmentation is a useful tool for improving model performance, increasing the dataset size and reducing generalization errors during the training process. However, it can also be used after the final training of a model to test the robustness of the model and give it more chances to correctly classify an image. Thus, to test the model I have the model predict the class of an image on multiple versions of that image (eg: flipped, rotated or cropped)(eg: Ayhan and Berens, 2018, Szegedy et al., 2014). The image is then predicted into a particular class by averaging all of the predictions on the augmented images. This is called Test-Time Data Augmentation (TTA). Generally, by doing this, I get a higher overall accuracy. In contrast to train-time data augmentation, the model is not changed, rather, the TTA is a technique giving the trained model a better chance to classify a given image correctly.

Thus, in this section, I use the TTA technique for presenting the performance of the models trained on the rebalanced dataset. I created 16 augmented copies of each image in the test set by rotating every 45 degree and then vertically flipping and horizontally flipping the 8 images from the

rotation so that I obtain 16 augmented copies of each image. After that, the model predicts the class for each of the 16 images, and then the ensemble of the predictions for the 16 augmented images is averaged and the averaged results are then used to classify the initial image.

As shown in the table(5.5), compared to the results without TTA, for DenseNetFastSGD, the overall accuracy improved from 87.85% to 89.22 % while class-wise accuracy decreased from 87.14% to 86.17%. Similarly for DenseNetSlowAdam, the overall accuracy improved from 89.05% to 90.14 %and the class-wise accuracy decreased from 87.80% to 86.83 %.

Thus, by performing the TTA the overall model performance slightly improves at the cost of the class-wise accuracy.

5.2.3 Summary and Discussion

By synthesis balancing (rebalancing) the dataset, the model performance was greatly improved. Also, through TTA, the overall accuracy of the model improved by around 1% on both models trained (DenseNetFastSGD and DenseNetSlowAdam). Thus, for understanding how well the above models perform on each single class, I choose the test with the best overall accuracy, DenseNetSlowAdam with TTA, and use the confusion matrix to evaluate the performance of this model (see Figure (5.4). Compared to the best performing model trained on the original dataset (DenseNetSlowAdam), the class-wise accuracy improved by 17% from 70.74% to 87.80%. The "column" class has the same excellent performance and in fact even has a slight increase in per-class accuracy. Moreover, for the "plate" and "lollipop" classes, which have very limited data, their performance largely improved from 59.83% and 66,67% to 88.28% and 89.05%, respectively. This is clear proof that training on an unbalanced dataset is one of main obstacles in achieving good model performance (Guo et al., 2008). Additionally, it can be seen that there are still two classes, "column plate" and "irregular", with per-class accuracy below 80%. For the "irregular" class, the model often mispredicted it into almost every other class. As previously mentioned, this makes sense as by definition, the "irregular" class can be any shape and therefore a single irregular ice crystal can be similar to any of the other ice crystal classes. Meanwhile the "column plate" class is often mispredicted into the 'Column' and 'Irregular' classes. The reason for this misprediction into the 'Irregular' class is as explained above, while for the misprediction into the "column" class, this can be explained by looking at some examples of the 'Column' and 'Column plate' classes (see Figure(5.5). As can be seen in Figure 5.5, in several instances columns and column plates are very alike, with their only discerning feature being the extension of two arms on the end of a column, such that a column plate looks a bit like an 'H'. Although it seems that through synthetic rebalancing, a good model can be obtained, a rebalanced dataset is after all a synthesis dataset. In reality, the proportion of each class cannot always be the same as was artificially created here. Therefore, the results from training with a truly balanced dataset are discussed in the following section.

Figure 5.4: Confusion matrix for DenseNet121 with Adam optimzer and 0.0001 learning rate, and using rebalanced dataset. The same way as previous Figure 5.2

5.3 Completely balanced dataset

To avoid arbitrarily changing the original dataset while producing a balanced dataset, one can use other methods to balance the dataset. One approach is to sample images from each of the classes in the training and validation set with equal probability during the training (Lemaıtre et al., 2017). Note that the test set of the completely balanced dataset is still the same as the previous rebalanced and original (unbalanced) datasets.

5.3.1 Non-Augmentation test

As shown in table (5.6), the class-wise accuracy of DenseNetFastSGD trained with the balanced dataset is 90.82%. This is a significant improvement when comparing the results trained from the

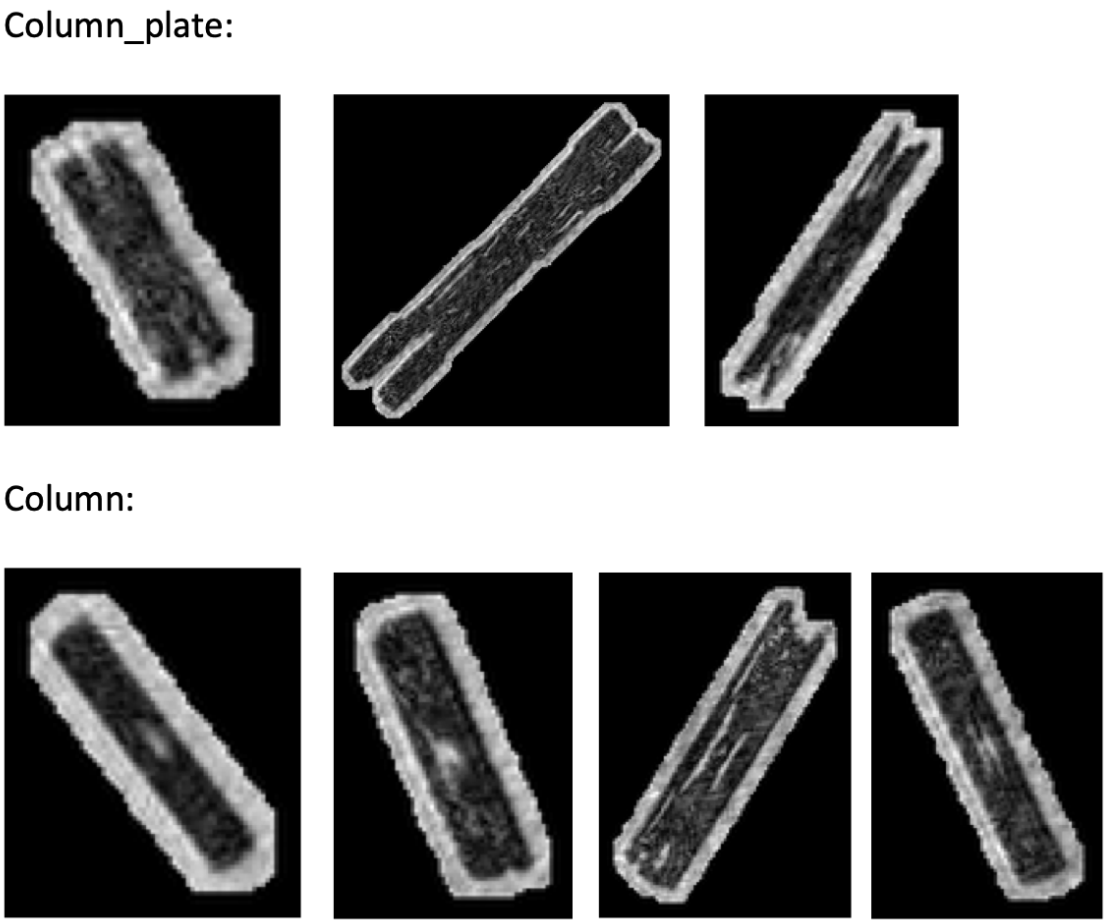

Figure 5.5: Examples of column plate (top) and column (bottom) ice crystals

Table 5.6: Completely balanced dataset

Model	Learning_rate	Optimiser	Augmentation	Overall	Class-wise
Densenet121	0.005	SGD	No	**0.8605**	**0.9082**
			Yes	0.8583	0.8954
	0.0001	Adam	No	0.8755	**0.9172**
			Yes	**0.8780**	0.9080

rebalanced dataset (87.14%). Likewise, DenseNetSlowAdam trained with the balanced dataset (BestIce) also has an improved class-wise accuracy from 87.80% to 91.72%. By selecting images from each class with the same probability during training, the model class-wise accuracy improved on average by around 3%.

5.3.2 Augmentation test

As with the rebalanced dataset tests, I also use the TTA technique for the balanced dataset testing. I created 16 augmented copies of each image in the test set in the same way as before.

As shown in table (5.6), compared to the results without TTA, for DenseNetFastSGD, both the overall accuracy and class-wise accuracy decrease by less than 1%, while for DenseNetSlowAdam, the overall accuracy increases by 0.15% and the class-wise accuracy has a 0.92% decrease. Thus using the TTA during the prediction of ice crystal habits does not improve the model performance and can be omitted to save time.

5.3.3 Summary and Discussion

By selecting samples with equal probability from each class, I largely improved the performance of the models without changing the dataset. Also, the TTA technique did not significantly improve the model performance when the model was trained on a balanced dataset. To evaluate the models performance on a class-wise basis I select the model with best overall performance (BestIce) and use the confusion matrix to evaluate its performance (see Figure (5.6). Using BestIce improves the overall accuracy from 83.29% to 87.55% and the class-wise accuracy by 21% from 70.74% to 91.72% relative to DenseNetFastSGD trained on the original dataset. With BestIce the "plate" and "lollipop" (group 1), "frozen droplets" and "small ice"(group 2) classes all have near perfect performance with around 99.5% and 98% per-class accuracy, respectively. The "irregular" and "rimed" classes also have a per-class accuracy of over 90%. However, the "aggregate" class has the worst performance with a per-class accuracy of 73.73%. This is mainly due to the model misclassifying aggregates into the "column plate" and "column" classes. More generally, the model also often misclassifies columns into the "column plate" class and vice versa leading to a per-class accuracy of around 87% for these classes. According to the previous discussion in section 5.1.4, I know that these three classes can be quite similar and are often difficult to classify by eye.

5.4 Validation

To prove the application value of the newly-developed models in the previous sections, in this section, the best performing model set (including 10 members), BestIce, is applied on a new dataset, NEWTEST as introdected in Section 4.1.

As shown in Table 4.2, this dataset is very small and imbalanced. The number of ice crystals in some of the rare classes (i.e. Lollipop) is even as low as 2, which can cause a high bias during the evaluation with respect to the class-wise accuracy. For example, if one 'Lollipop' is mispredicted, it would result in the lollipop per-class accuracy being 50 % and thus, largely influence the class-wise accuracy of the entire dataset. Therefore, in this section, class-wise accuracy is not used as a model performance metric.

Note that the way of getting prediction results is first of all, to use these 10 members in BestIce to classify the ice crystal images in NEWTEST dataset one by one and then average the results obtained from these 10 members. The averaged results is the final results to be evaluated.

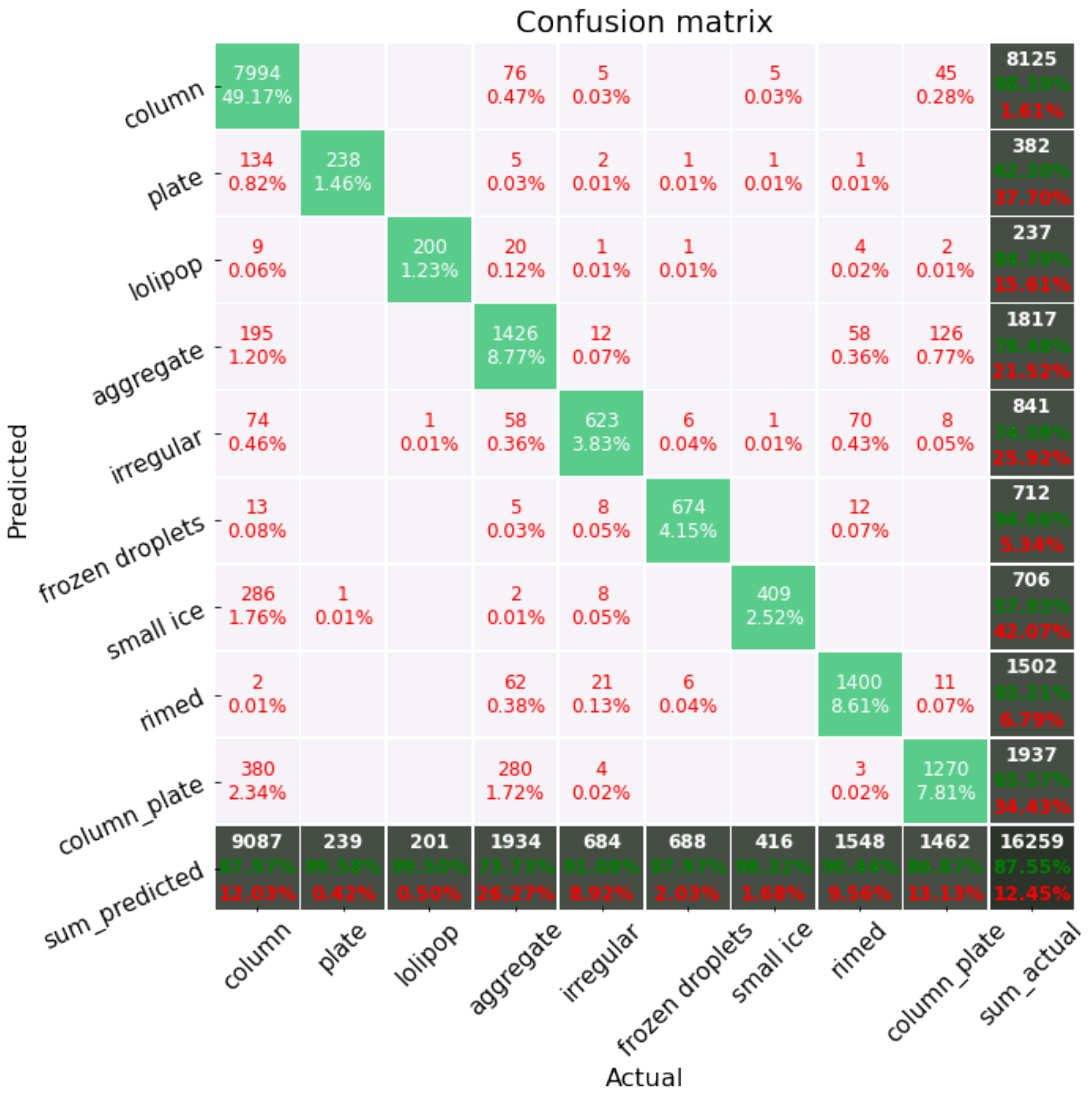

Figure 5.6: Confusion matrix for DenseNet121 with AdamW optimizer and 10-4 learning rate, sampling each class with equal probability. The same way as previous Figure 5.2

The global accuracy of BestIce, which is the average performance of 10 members of BestIce predicted on the NEWTEST dataset, is 63.31%, which is much lower than the results obtained from the previous Section 5.3. To determine the reasons why I obtained the above results, I first look at some examples of ice crystal examples in each class and compare those samples to the original dataset. As shown in Figure 5.7, 5.8 and 5.9, the NEWTEST dataset looks very different from the original. For the 'column' class, the new columns are much shorter, more rectangular and in some cases even square-like. In contrast, the columns in the original dataset were long and thin. Similarly the 'Frozen droplets' and 'lollipop' classes are not as standard as in the original dataset. For example, the frozen droplets are not as round as in the original dataset and some are even rimed. Additionally, the ice crystals in the 'Column Plate' class do not look like an 'H' as they did in the original dataset, which means that the column plates are frequently mispredicted. As

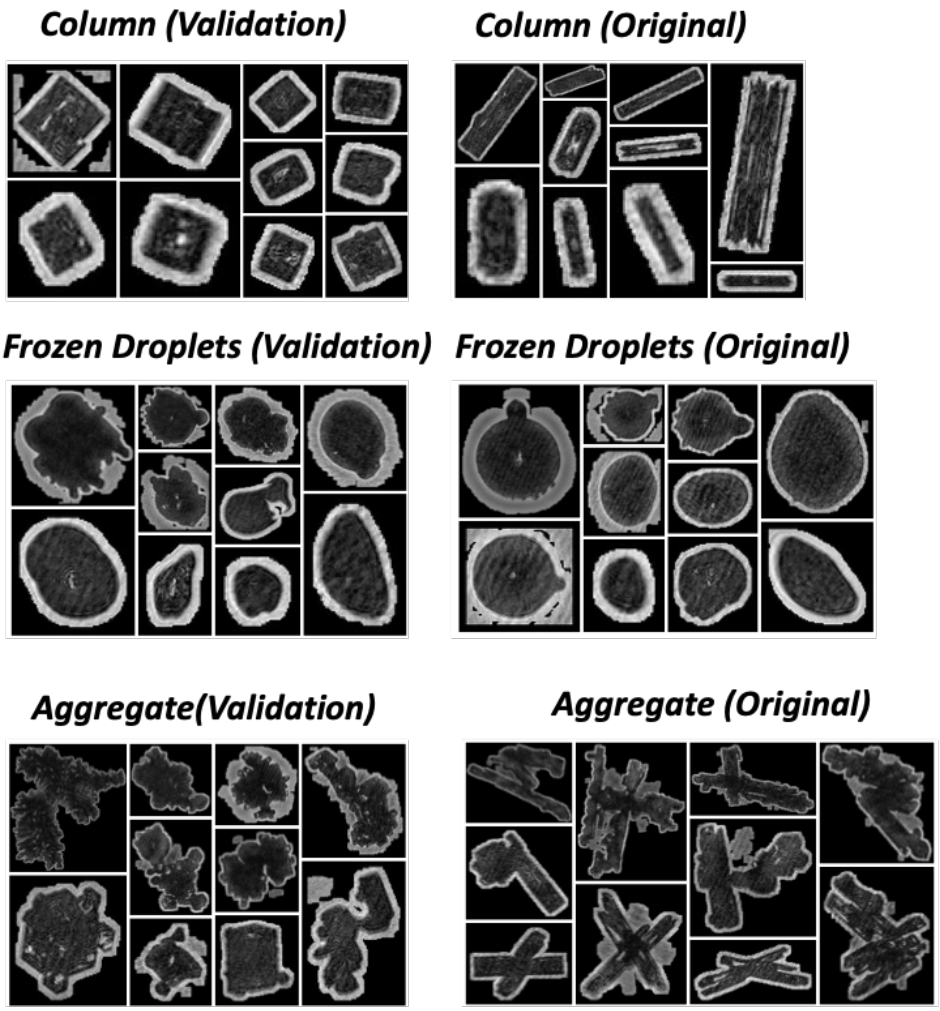

Figure 5.7: Comparison among class column, frozen droplets and aggregate of the ice crystals from the hand-labeled NEWTEST dataset and original dataset.

with the original dataset, the model struggles to correctly differentiate between the 'Aggregate', 'Irregular', and 'Rimed' classes due to their often similar features. For example, as can be seen in Figure 5.7, the first ice crystal in the 'Aggregate' class could be labeled as 'Rimed' or 'Aggregate'. In fact, this is a rimed-aggregate and therefore this kind of ice crystal is called a compound ice crystal, which can be any one of these two habit depending on the research purpose. For example, for investigation of the aggregation process, it can be labeled as 'Aggregate'; if the focus is rather on the riming process, the labeling could be the opposite ('Rimed').

However, for those confusing ice crystals, BestIce may give a result that for example 45 % is an 'Aggregate' and 35 % is a 'Rimed' rather than a very 'confident' answer that 90% is an 'Aggregate' or 95 % is a 'Rimed'. Thus, a proper predicted probability may be the direction of tackling this confusing ice crystal issue. Generally, a normal predicted probability (how confident the CNN model is that it should be some specific class) is ranging from 0-100%. To determine the threshold

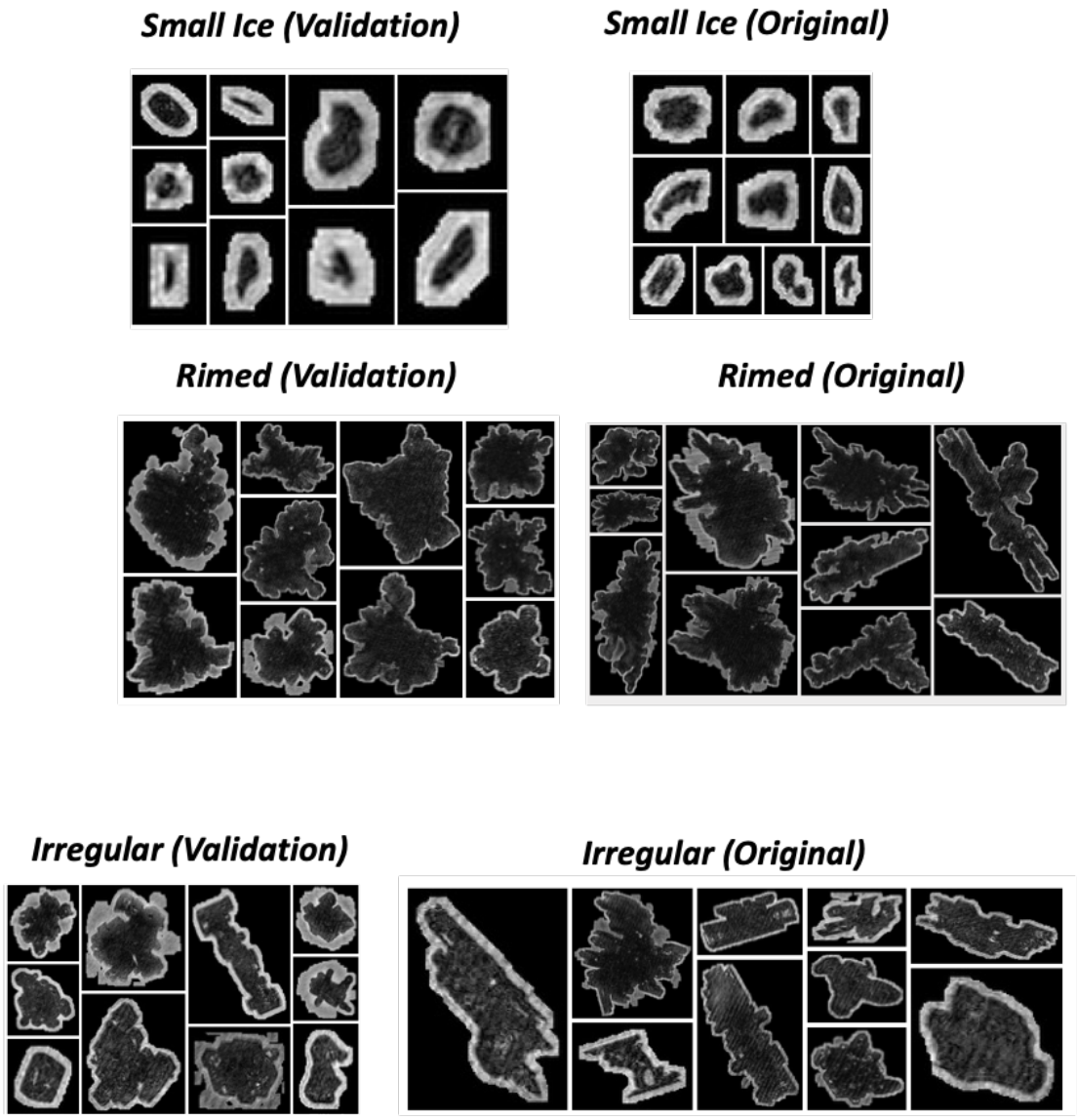

Figure 5.8: Comparison among class small ice, rimed and irregular of the ice crystals from the hand-labeled NEWTEST dataset and original dataset.

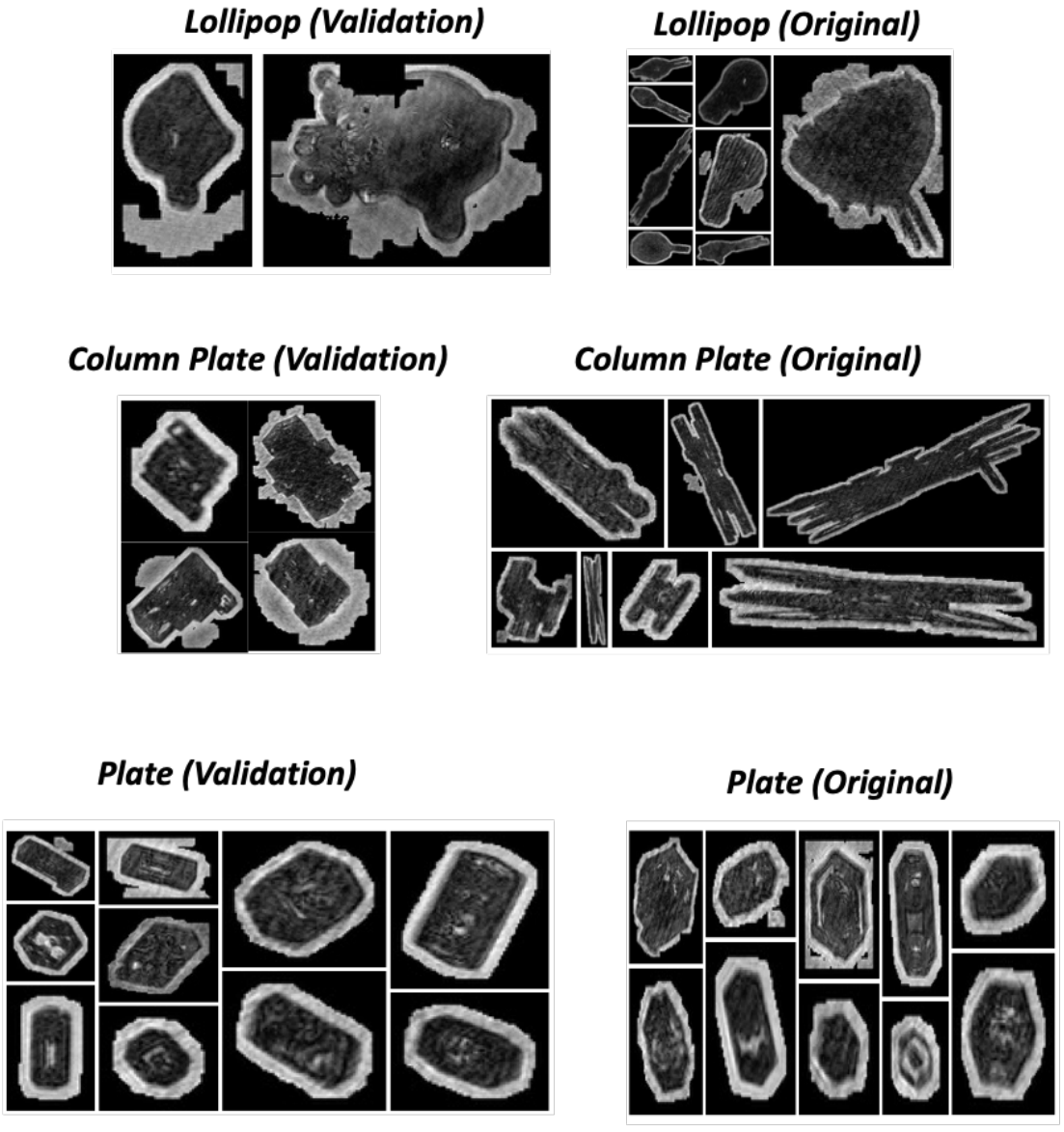

Figure 5.9: Comparison among class lollipop, column plate and plate of the ice crystals from the hand-labeled NEWTEST dataset and original dataset.

where the predicted probability is accurate enough to be trusted to correctly classify an ice crystal into the correct class, the global accuracy and cumulative number of ice crystals as a function of probability must be known. As shown in Figure 5.10 the global accuracy and the number of ice crystals both increase with increasing model confidence. This increase is particular pronounced at a prediction probability of around 99%. In fact approximately 40% of the ice crystals are predicted with a 99% probability that they belong to a particular class. The global accuracy also increases by more than 17.5% to over 80% at the 99% probability threshold. Therefore, without fine-tuning or further training, the model should only be used when it is 99% confident that an ice crystal belongs to a certain class in future work.

Therefore, to evaluate the model performance when the predicted probability of BestIce was over 99%, the confusion matrix is shown in Figure 5.11. The overall accuracy improved from 63.31% to 80.62% and the number of samples decreased from 1352 to 609. As discussed previously, the 'Column' and 'Column Plate' classes are very different from the original dataset and thus, the predicted results are unexpectedly bad as 68.18% and 0% respectively. Also, there are no lollipop crystals predicted with a probability of over 99%, which also improved our hypothesis that lollipop images are confusing and not standard as the lollipop images in NEWTEST dataset have rimed boundaries and are incomplete (as can be seen in the second lollipop top in Figure 5.9). Consistent with the training dataset and the lower prediction probabilities, the model struggles to differentiate between 'Aggregate', 'Irregular' and 'Rimed'. Meanwhile, the more unique classes like 'frozen droplets', 'small ice' and 'Plate' all show good performance with 88.57%, 98.84% and 90% per-class accuracy, respectively.

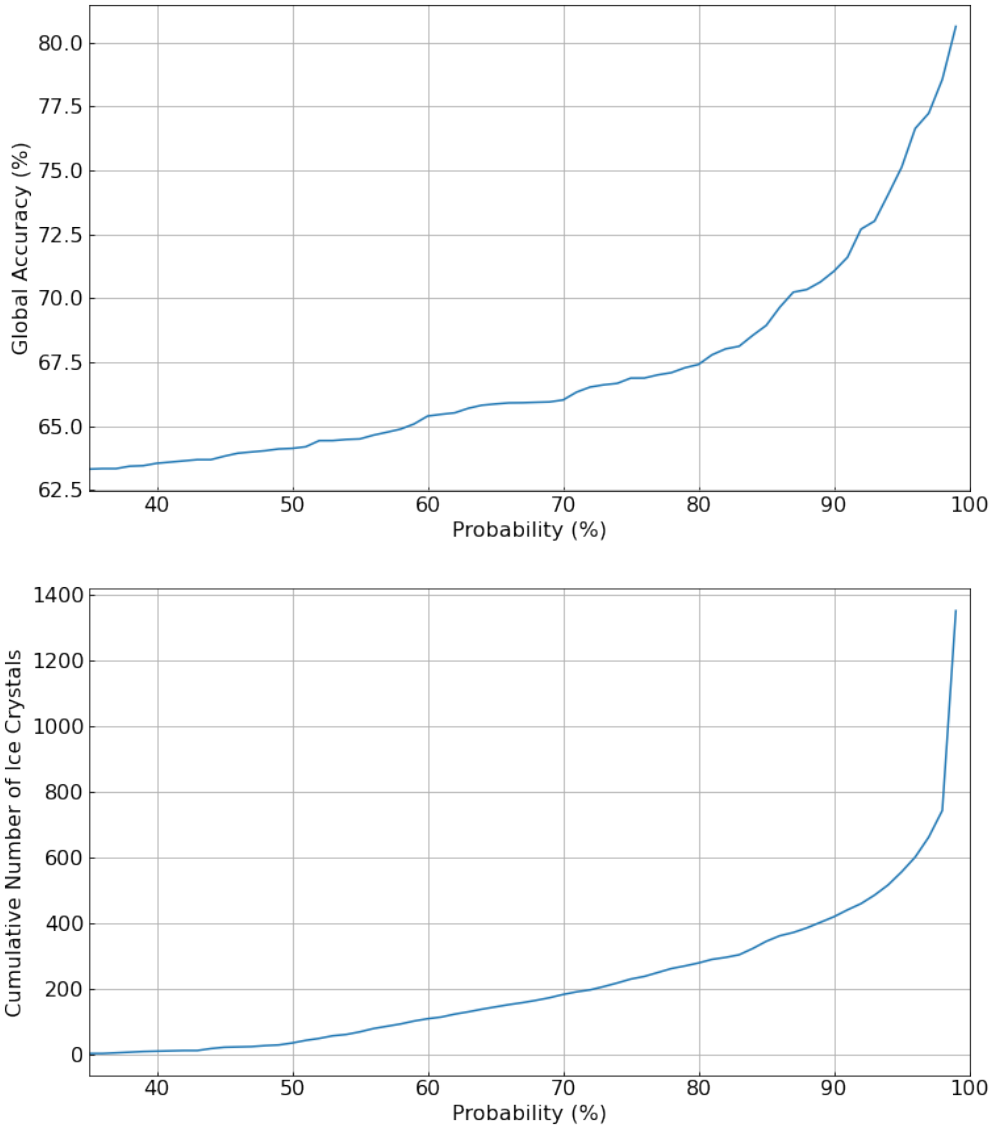

Figure 5.10: Global accuracy of the model (Top) and the cumulative number of ice crystals (Bottom) as a function of the predicted probability of an ice crystal belonging to a particular class

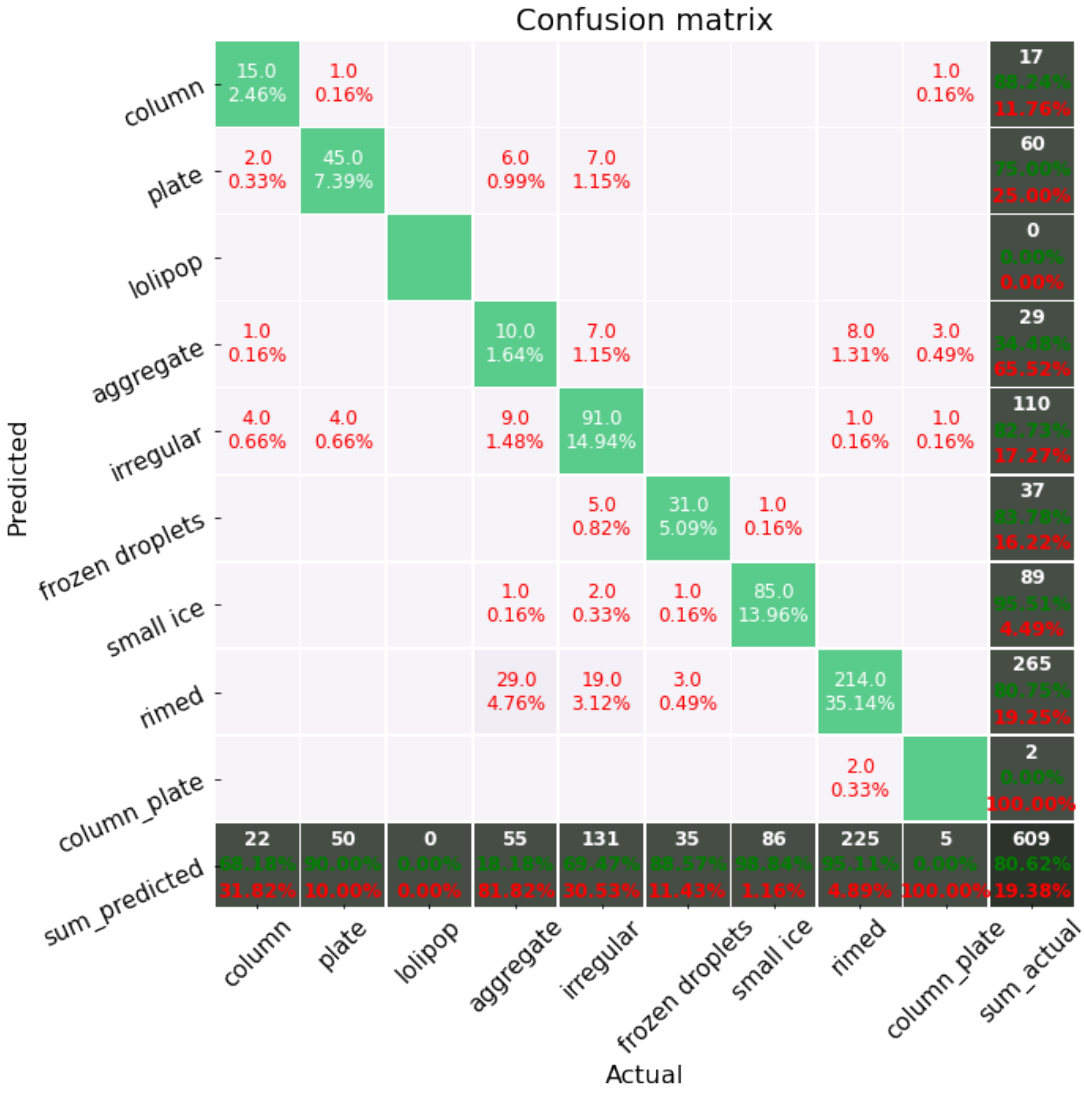

Figure 5.11: Confusion matrix for ice crystals predicted with BestIce when the prediction probability was over 99 % on NEWTEST dataset.The same way as previous Figure 5.2

CHAPTER 6

Conclusion and Outlook

6.1 Conclusion and Discussion

In this thesis, I developed automatic classification models for ice crystal habits by using convolutional neural networks and transfer learning. The best model, BestIce, achieved an overall accuracy of 87.55% and a class-wise accuracy of 91.72%. The model performed best on the 'Plate' and 'Lollipop', and, 'Frozen droplets' and 'Small ice' classes with per-class accuracies of around 99.5% and 98%, respectively. This automatic classification model is applied to classify and analyse another ice crystal data, NEWTEST, also collected during the Fall 2019 portion of the NASCENT campaign. By setting a 99% predicted probability, BestIce achieved over 80% overall accuracy and thus proved that the model can 'liberate' us from the tedious hand-labeling process.

Six different pre-trained models (DenseNet-121, DenseNet-169, DenseNet-201, Resnet-18, Resnet-101, and Resnet-152) with fixed learning rates and optimisers were trained on the original dataset. The results showed that DenseNet-121 had a slightly better performance. Thus, the DenseNet-121 model was used to perform two different optimizer (SGD and Adam) with different learning rate tests. The best combinations of optimzers and learning rates were the SGD optimzer with a learning rate of 0.005 (DenseNetFastSGD) and Adam with a learning rate of 0.0001 (DenseNetSlowAdam). Due to their superior performance, DenseNetFastSGD and DenseNetSlowAdam were trained with the lower half of the model layers frozen. For both models, freezing the lower half of the layers did not improve model performance. This is likely due to the unbalanced nature of the dataset. Furthermore the information acquired by the model in the lower layers is primarily bulk in nature and therefore, small changes to these layers during training has been shown to have minimal impacts on the overall model performance (Goutam et al., 2020).

The best performing models from the previous tests, DenseNetFastSGD and DenseNetSlowAdam were subsequently trained on a rebalanced and a balanced version of the original dataset.

The rebalanced dataset was constructed by removing 2/3 of the dominant "column" class, while the rest of the classes remained unchanged. When training the models on the rebalanced dataset, the class-wise accuracy improved by 17% from 70.74% to 87.80% relative to the best model trained

on the original dataset. Additionally, through TTA, the overall accuracy of DenseNetSlowAdam increased by around 1% while for DenseNetFastSGD, the change was negligible. Thus, in this case, TTA is not very helpful, but that might be different in different situations as TTA has been shown to improve model performance in other fields (i.e. images of tissue and cell cultures Moshkov et al., 2020).

The balanced dataset was created by sampling images from each class of the original dataset with equal probability during the training. By doing so, the overall accuracy and class-wise accuracy of DenseNetSlowAdam increased from 83.29% to 87.55% and 70.74% to 91.72%, respectively. Additionally, training on the balanced dataset achieved excellent performance in some single classes with a per-class accuracy of approximately 99.5% for 'Plates' and 'Lollipops' and 98% for 'Frozen droplets' and 'Small ice' crystals.

Finally, the best performing model, BestIce, was applied on a new dateset, NEWTEST, as the validation/real-world application. Even though the new dataset was very different from the original dataset used for model training, setting the prediction probability threshold at 99% really improved the global accuracy of the model to approximately 80%. In particular, for some unique classes and classes where the ice crystals were similar to the ones in the original dataset, such as 'Frozen droplets', 'Small ice' and 'Plate', the model per-class accuracy was 88.57%, 98.84% and 90%, respectively. Thus, the application value of BestIce has been proved and confirmed.

In this thesis, classification models for ice crystal habits were developed and tested. The best model achieved an overall accuracy of 87.55% and a class-wise accuracy of 91.72%. However, it should be noted that the hyperparameter choices other than the epoch, such as learning rate and different architectures and different balancing schemes were conducted on the test set. This is not fully correct, as it can lead to overfitting of the test set. The correct way to do this would be for every train/val/test split to evaluate all hyperparameter choices on the validation set, and evaluate only the one final model on the test set.

6.2 Outlook

There are several directions that can be explored in future work to improve the performance of BestIce when it comes to tackling issues with compound ice crystals (composed of two or more classes) and unseen ice crystal habits (habits that are not included in the nine classes used during training).

1. As presented in both Section 5.6 and 5.4, the confusing issue among classes 'Aggregate', 'Irregular', 'Rimed' has always existed. Therefore, one could consider to merge these three classes into a single class.

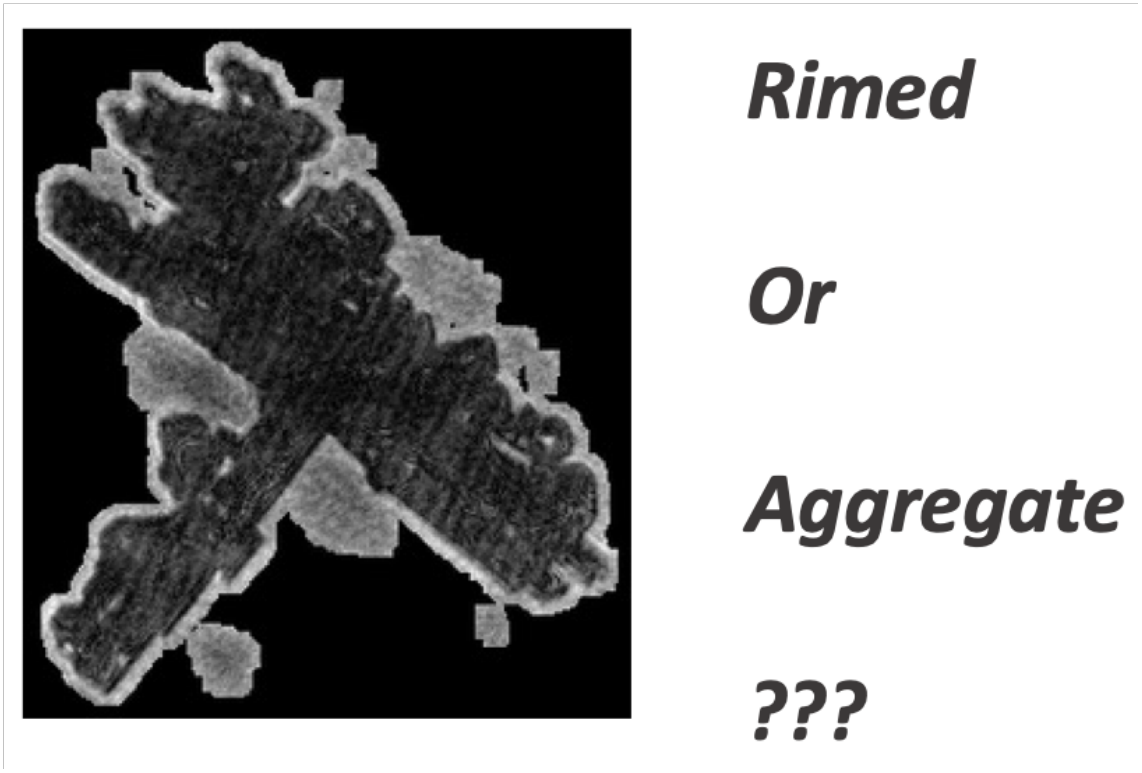

Figure 6.1: Example of a compound ice crystal holographic image.

2. As discussed in Section 5.4, one main obstacle of getting a perfect overall accuracy is the large difference between the original and NEWTEST dataset. Thus, in the future, one can improve BestIce on known classes (i.e. column, lollipop) by building a bigger training dataset.

3. Setting the prediction probability threshold at 99% might be the best threshold overall, while for each single class, the best threshold may be different. For example, some unique classes, such as 'Lollipop', may not need such a high threshold. Thus, the optimal threshold for each single class can be further explored.

4. As discussed in Section 5.4, it is very likely that some compound ice (as shown in the Figure 6.1) crystals exist in a dataset, especially in situations conducive to light riming. To get an objective answer/label, one can develop a multi-label classification model for confusing compound ice by using Classification Transformers (Lanchantin et al., 2020).

5. Future campaign data will almost certainly include some unseen ice crystal habits. Due to different meteorological conditions with different microphysical processes, ice crystal habits can vary from just a few to several hundreds(Kikuchi et al., 2013). Despite this, the model can be quickly adapted through a simple fine-tuning procedure and then be used to classify ice crystals in the new dataset. However, the efficiency and feasibility of this idea needs to be further investigated.

Appendices

APPENDIX A

Figures and Tables

Table A.1: ResNet model architecture, Taken from He et al., 2015a

Layer Name	**Output Size**	**18-layer**	**101-layer**	**152-layer**
cov1	112×112	7×7, 64, stride 2		
conv2 x	56×56	3×3 max pool, stride 2		
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3 x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4 x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5 x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax		
FLOPs		1.8×10^9	3.8×10^9	11.3×10^9

Table A.2: DensetNet models architectures, Taken from Huang et al., 2018

Layer Name	**Output Size**	**DenseNet-121**	**DenseNet-169**	**DenseNet-201**
Convolution	112×112	7×7, 64, stride 2		
Pooling	56×56	3×3 max pool, stride 2		
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv		
	28×28	2×2 average pool, stride 2		
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv		
	14×14	2×2 average pool, stride 2		
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 48$
Transition Layer (3)	14×14	1×1 conv		
	7×7	2×2 average pool, stride 2		
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1\text{conv} \\ 3 \times 3\text{conv} \end{bmatrix} \times 32$
Classification Layer	1×1	7×7 global average pool		
		1000D fully-connected, softmax		

APPENDIX B

Acronyms

Adagrad Adaptive Gradient Algorithm

CAS Cloud and Aerosol Spectrometer

CIP Precipitation Imaging Probes

CNN Convolutional Neural Network

CPI Cloud Particle Imager

DenseNets Densely Connected Networks

ECR Equivalent Circle Ratio

FDR Overall False Discovery Rate

FFNNs Feed-forward Neural Network

GAN Generative Adversarial Network

GPUs Graphics Processing Units

HVPS High Volume Precipitation Spectrometer

ICNC Ice Crystal Number Concentration

IC-PCA Ice-crystal Classification with Principal Component Analysis

MASC Multi-Angle Snowflake Camera

MPCs Mixed Phase Clouds

NN Neural Networks

OAP Optical Array Probes

PCA Principal Component Analysis

PMS Particle Measuring System

ResNet Residual Networks

RMSProp Root Mean Square Propagation

SGD Stochastic GradientDescent

2D-S Two-Dimensional Stereo spectrometer

TTA Test-Time Data Augmentation

WBF Wegener–Bergeron–Findeisen

APPENDIX C

Facilities preparation

The instructions to access the code used in this thesis and the hardware the code was run on are described in the following sections.

C.1 Code available

The code for all of the analysis and training in this thesis is available on GitHub at https://github.com/zhanghuiying2319/Master/Thesis/cnn.

The commands to download the code are summarized as follows:

Listing C.1: Linux command for how to install the environment

```
git clone https://github.com/zhanghuiying2319/Master.git
pip install -r requirements.txt
```

C.2 Hardware

The experiments in this thesis have been conducted on Ml-nodes at University of Oslo (UiO). The table (Table C.1) below describes the hardware of the resources used.

Table C.1: Hardware of ML-nodes

Name	Status	CPUs/ RAM(GiB)	GPU	OS and software
ml1.hpc.uio.no ml2.hpc.uio.no ml3.hpc.uio.no	Production	28 cores (Intel Xeon)/128	4 X RTX2080Ti	RHEL 8.3 with module system

Bibliography

Amsler, P., Stetzer, O., Schnaiter, M., Hesse, E., Benz, S., Moehler, O. and Lohmann, U. (Oct. 2009). 'Ice crystal habits from cloud chamber studies obtained by in-line holographic microscopy related to depolarization measurements'. In: *Appl. Opt.* vol. 48, no. 30, pp. 5811–5822.

Ayhan, M. and Berens, P. (2018). 'Test-time Data Augmentation for Estimation of Heteroscedastic Aleatoric Uncertainty in Deep Neural Networks'. In:

Bailey, M. P. and Hallett, J. (2009). 'A Comprehensive Habit Diagram for Atmospheric Ice Crystals: Confirmation from the Laboratory, AIRS II, and Other Field Studies'. In: *Journal of the Atmospheric Sciences* vol. 66, no. 9, pp. 2888–2899.

Baumgardner, D., Jonsson, H., Dawson, W., O'Connor, D. and Newton, R. (2001). 'The cloud, aerosol and precipitation spectrometer: a new instrument for cloud investigations'. In: *Atmospheric Research* vol. 59-60. 13th International Conference on Clouds and Precipitation, pp. 251–264.

Beals, M. J., Fugal, J. P., Shaw, R. A., Lu, J., Spuler, S. M. and Stith, J. L. (2015). 'Holographic measurements of inhomogeneous cloud mixing at the centimeter scale'. In: *Science* vol. 350, no. 6256, pp. 87–90. eprint: https://science.sciencemag.org/content/350/6256/87.full.pdf.

Beck, A., Henneberger, J., Schöpfer, S., Fugal, J. and Lohmann, U. (2017). 'HoloGondel: in situ cloud observations on a cable car in the Swiss Alps using a holographic imager'. In: *Atmospheric Measurement Techniques* vol. 10, no. 2, pp. 459–476.

Beswick, K., Baumgardner, D., Gallagher, M., Volz-Thomas, A., Nedelec, P., Wang, K.-Y. and Lance, S. (2014). 'The backscatter cloud probe ndash; a compact low-profile autonomous optical spectrometer'. In: *Atmospheric Measurement Techniques* vol. 7, no. 5, pp. 1443–1457.

Blevin, W. R. and Brown, W. J. (Jan. 1971). 'A Precise Measurement of the Stefan-Boltzmann Constant'. In: *Metrologia* vol. 7, no. 1, pp. 15–29.

Borrmann, S., Jaenicke, R. and Neumann, P. (1993). 'On spatial distributions and inter-droplet distances measured in stratus clouds with in-line holography'. In: *Atmospheric Research* vol. 29, no. 3, pp. 229–245.

'Cloud Ice Properties: In Situ Measurement Challenges' (Apr. 2017). English. In: *Meteorological Monographs* vol. 58, pp. 9.1–9.23.

convolution network basics - Charlotte77 (2019). zh-cn.

Cunningham, R. (1978). 'Analysis of Particle Spectral Data from Optical Array (PMS) 1D and 2D Sensors.' In:

Duroure, C., Larsen, H., Isaka, H. and Personne, P. (1994). '2D image population analysis'. In: *Atmospheric Research* vol. 34, no. 1. 11th conference on clouds and precipitation, pp. 195–205.

Ehrlich, A., Wendisch, M., Bierwirth, E., Herber, A. and Schwarzenböck, A. (Jan. 2008). 'Ice crystal shape effects on solar radiative properties of Arctic mixed-phase clouds—Dependence on microphysical properties'. In: *Atmospheric Research* vol. 88, no. 3, pp. 266–276.

Feind, R. E. (2006). 'Comparison of three classification methodologies for 2D probe hydrometeor images obtained from the armored T-28 aircraft'. In: *South Dakota School of Mines and Technology*.

Field, P. R. and Heymsfield, A. J. (n.d.). 'Importance of snow to global precipitation'. In: *Geophysical Research Letters* vol. 42, no. 21 (), pp. 9512–9520. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2015GL065497.

Fu, Q. and Liou, K. N. (1993). 'Parameterization of the Radiative Properties of Cirrus Clouds'. In: *Journal of Atmospheric Sciences* vol. 50, no. 13, pp. 2008–2025.

Fugal, J. P. and Shaw, R. A. (2009). 'Cloud particle size distributions measured with an airborne digital in-line holographic instrument'. In: *Atmospheric Measurement Techniques* vol. 2, no. 1, pp. 259–271.

Garrett, T. J., Fallgatter, C., Shkurko, K. and Howlett, D. (2012). 'Fall speed measurement and high-resolution multi-angle photography of hydrometeors in free fall'. In: *Atmospheric Measurement Techniques* vol. 5, no. 11, pp. 2625–2633.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition [Book]*.

Glen, A. and Brooks, S. D. (2013). 'A new method for measuring optical scattering properties of atmospherically relevant dusts using the Cloud and Aerosol Spectrometer with Polarization (CASPOL)'. In: *Atmospheric Chemistry and Physics* vol. 13, no. 3, pp. 1345–1356.

Goutam, K., Balasubramanian, S., Gera, D. and Sarma, R. R. (2020). 'LayerOut: Freezing Layers in Deep Neural Networks'. In: *SN Computer Science* vol. 1, no. 5, pp. 1–9.

Grandini, M., Bagli, E. and Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview*. arXiv: 2008.05756 [stat.ML].

Guo, X., Yin, Y., Dong, C., Yang, G. and Zhou, G. (2008). 'On the class imbalance problem'. In: *2008 Fourth international conference on natural computation*. Vol. 4. IEEE, pp. 192–201.

He, K., Zhang, X., Ren, S. and Sun, J. (2015a). *Deep Residual Learning for Image Recognition*. arXiv: 1512.03385 [cs.CV].

— (2015b). *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. arXiv: 1502.01852 [cs.CV].

Henneberger, J., Fugal, J. P., Stetzer, O. and Lohmann, U. (2013). 'HOLIMO II: a digital holographic instrument for ground-based in situ observations of microphysical properties of mixed-phase clouds'. In: *Atmospheric Measurement Techniques* vol. 6, no. 11, pp. 2975–2987.

Huang, G., Liu, Z., Maaten, L. van der and Weinberger, K. Q. (2018). *Densely Connected Convolutional Networks*. arXiv: 1608.06993 [cs.CV].

Ingargiola, A. (Apr. 2019). *Deep-dive into Convolutional Networks*. en.

Intrieri, J. M., Fairall, C. W., Shupe, M. D., Persson, P. O. G., Andreas, E. L., Guest, P. S. and Moritz, R. E. (2002). 'An annual cycle of Arctic surface cloud forcing at SHEBA'. In: *Journal of Geophysical Research: Oceans* vol. 107, no. C10, SHE 13-1-SHE 13–14. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2000JC000439.

Kikuchi, K., Kameda, T., Higuchi, K., Yamashita, A. et al. (2013). 'A global classification of snow crystals, ice crystals, and solid precipitation based on observations from middle latitudes to polar regions'. In: *Atmospheric research* vol. 132, pp. 460–472.

Kingma, D. P. and Ba, J. (2017). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG].

Knollenberg, R. (1976). *Three New Instruments for Cloud Physics Measurements: The 2-D Spectrometer, the Forward Scattering Spectrometer Probe, and the Active Scattering Aerosol Spectrometer*. American Meteorological Society.

Knollenberg, R. G. (1970). 'The Optical Array: An Alternative to Scattering or Extinction for Airborne Particle Size Determination'. In: *Journal of Applied Meteorology and Climatology* vol. 9, no. 1, pp. 86–103.

Korolev, A. and Sussman, B. (2000). 'A Technique for Habit Classification of Cloud Particles'. In: *Journal of Atmospheric and Oceanic Technology* vol. 17, no. 8, pp. 1048–1057.

Lamb, D. and Verlinde, J. (Jan. 2011). *Physics and chemistry of clouds*. English (US). United Kingdom: Cambridge University Press.

Lance, S. (2012). 'Coincidence Errors in a Cloud Droplet Probe (CDP) and a Cloud and Aerosol Spectrometer (CAS), and the Improved Performance of a Modified CDP'. In: *Journal of Atmospheric and Oceanic Technology* vol. 29, no. 10, pp. 1532–1541.

Lanchantin, J., Wang, T., Ordonez, V. and Qi, Y. (2020). *General Multi-label Image Classification with Transformers*. arXiv: 2011.14027 [cs.CV].

Lawson, R. P., O'Connor, D., Zmarzly, P., Weaver, K., Baker, B., Mo, Q. and Jonsson, H. (2006). 'The 2D-S (Stereo) Probe: Design and Preliminary Tests of a New Airborne, High-Speed, High-Resolution Particle Imaging Probe'. In: *Journal of Atmospheric and Oceanic Technology* vol. 23, no. 11, pp. 1462–1477.

Lawson, R. P., Stewart, R. E. and Angus, L. J. (1998). 'Observations and Numerical Simulations of the Origin and Development of Very Large Snowflakes'. In: *Journal of the Atmospheric Sciences* vol. 55, no. 21, pp. 3209–3229.

Lawson, R. P., Stewart, R. E., Strapp, J. W. and Isaac, G. A. (1993). 'Aircraft observations of the origin and growth of very large snowflakes'. In: *Geophysical Research Letters* vol. 20, no. 1, pp. 53–56. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/92GL02917.

Lawson, R. and Cormack, R. (1995). 'Theoretical design and preliminary tests of two new particle spectrometers for cloud microphysics research'. In: *Atmospheric Research* vol. 35, no. 2, pp. 315–348.

Leinonen, J. and Berne, A. (2020). 'Unsupervised classification of snowflake images using a generative adversarial network and *K*-medoids classification'. In: *Atmospheric Measurement Techniques* vol. 13, no. 6, pp. 2949–2964.

Lemaıtre, G., Nogueira, F. and Aridas, C. K. (2017). 'Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning'. In: *The Journal of Machine Learning Research* vol. 18, no. 1, pp. 559–563.

Libbrecht, K. G. (2016). *Ken Libbrecht's field guide to snowflakes*. Voyageur Press.

Lindqvist, H., Muinonen, K., Nousiainen, T., Um, J., McFarquhar, G. M., Haapanala, P., Makkonen, R. and Hakkarainen, H. (2012). 'Ice-cloud particle habit classification using principal components'. In: *Journal of Geophysical Research: Atmospheres* vol. 117, no. D16. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2012JD017573.

Liou, K.-N. and Yang, P. (2016). *Light scattering by ice crystals: fundamentals and applications*. Cambridge University Press.

Lohmann, U. (2002). 'A glaciation indirect aerosol effect caused by soot aerosols'. In: *Geophysical Research Letters* vol. 29, no. 4, pp. 11-1-11–4. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2001GL014357.

Matus, A. V. and L'Ecuyer, T. S. (2017). 'The role of cloud phase in Earth's radiation budget'. In: *Journal of Geophysical Research: Atmospheres* vol. 122, no. 5, pp. 2559–2578.

McFarquhar, G. M., Heymsfield, A. J., Macke, A., Iaquinta, J. and Aulenbach, S. M. (1999). 'Use of observed ice crystal sizes and shapes to calculate mean-scattering properties and multispectral radiances: CEPEX April 4, 1993, case study'. In: *Journal of Geophysical Research: Atmospheres* vol. 104, no. D24, pp. 31763–31779.

McFarquhar, G. M. and Heymsfield, A. J. (1996). 'Microphysical Characteristics of Three Anvils Sampled during the Central Equatorial Pacific Experiment'. In: *Journal of Atmospheric Sciences* vol. 53, no. 17, pp. 2401–2423.

Moshkov, N., Mathe, B., Kertesz-Farkas, A., Hollandi, R. and Horvath, P. (2020). 'Test-time augmentation for deep learning-based cell segmentation on microscopy images'. In: *Scientific reports* vol. 10, no. 1, pp. 1–7.

Mülmenstädt, J., Sourdeval, O., Delanoë, J. and Quaas, J. (2015). 'Frequency of occurrence of rain from liquid-, mixed-, and ice-phase clouds derived from A-Train satellite retrievals'. In:

Geophysical Research Letters vol. 42, no. 15, pp. 6502–6509. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2015GL064604.

Pan, S. J. and Yang, Q. (2010). 'A Survey on Transfer Learning'. In: *IEEE Transactions on Knowledge and Data Engineering* vol. 22, no. 10, pp. 1345–1359.

Praz, C., Roulet, Y.-A. and Berne, A. (2017). 'Solid hydrometeor classification and riming degree estimation from pictures collected with a Multi-Angle Snowflake Camera'. In: *Atmospheric Measurement Techniques* vol. 10, no. 4, pp. 1335–1357.

Rahman, M. M., Quincy, E. A., Jacquot, R. G. and Magee, M. J. (1981). 'Feature Extraction and Selection for Pattern Recognition of Two-Dimensional Hydrometeor Images'. In: *Journal of Applied Meteorology and Climatology* vol. 20, no. 5, pp. 521–535.

Raju, A. and Thirunavukkarasu, S. (2020). *Convolutional Neural Network Demystified for a Comprehensive Learning with Industrial Application, Dynamic Data Assimilation - Beating the Uncertainties*. BoD–Books on Demand.

Ramelli, F., Beck, A., Henneberger, J. and Lohmann, U. (2020). 'Using a holographic imager on a tethered balloon system for microphysical observations of boundary layer clouds'. In: *Atmospheric Measurement Techniques* vol. 13, no. 2, pp. 925–939.

Schlegel, D. (2015). 'Deep machine learning on Gpu'. In: *University of Heidelber-Ziti* vol. 12.

Schlimme, I., Macke, A. and Reichardt, J. (2005). 'The Impact of Ice Crystal Shapes, Size Distributions, and Spatial Structures of Cirrus Clouds on Solar Radiative Fluxes'. In: *Journal of the Atmospheric Sciences* vol. 62, no. 7, pp. 2274–2283.

Slingo, J. and Slingo, A. (1991). 'The response of a general circulation model to cloud longwave radiative forcing. II: Further studies'. In: *Quarterly Journal of the Royal Meteorological Society* vol. 117, no. 498, pp. 333–364.

Spuler, S. and Fugal, J. (2011). 'Design of an in-line, digital holographic imaging system for airborne measurement of clouds.' In: *Applied optics* vol. 50 10, pp. 1405–12.

Stephens, G. L., Tsay, S.-C., Stackhouse, P. W. and Flatau, P. J. (1990). 'The relevance of the microphysical and radiative properties of cirrus clouds to climate and climatic feedback'. In: *J. Atmos. Sci.* vol. 47, pp. 1742–1753.

Stubenrauch, C. J. et al. (2013). 'Assessment of Global Cloud Datasets from Satellites: Project and Database Initiated by the GEWEX Radiation Panel'. In: *Bulletin of the American Meteorological Society* vol. 94, no. 7, pp. 1031–1049.

Sun, Z. and Shine, K. P. (1994). *Studies of the radiative properties of ice and mixed-phase clouds.*

Sutskever, I., Martens, J., Dahl, G. and Hinton, G. (17–19 Jun 2013). 'On the importance of initialization and momentum in deep learning'. In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Dasgupta, S. and McAllester, D. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, pp. 1139–1147.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2014). *Going Deeper with Convolutions*. arXiv: 1409.4842 [cs.CV].

The Ny-Ålesund Aerosol Cloud Experiment (NASCENT) 2019-2020 (n.d.). https://www.aces.su.se/research/projects/the-ny-alesund-aerosol-cloud-experiment-nascent-2019-2020/.

Tieleman, T. and Hinton, G. (2012). 'Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude'. In: *COURSERA: Neural networks for machine learning* vol. 4, no. 2, pp. 26–31.

Touloupas, G., Lauber, A., Henneberger, J., Beck, A. and Lucchi, A. (2020). 'A convolutional neural network for classifying cloud particles recorded by imaging probes'. In: *Atmospheric Measurement Techniques* vol. 13, no. 5, pp. 2219–2239.

Trolinger, J. D. (1975). 'Flow Visualization Holography'. In: *Optical Engineering* vol. 14, no. 5, pp. 470–481.

Ullah, H. and Bhuiyan, M. (May 2018). 'Performance Evaluation of Feed Forward Neural Network for Image Classification'. In: *Journal of Science and Technology* vol. 10.

Vergara-Temprado, J., Miltenberger, A. K., Furtado, K., Grosvenor, D. P., Shipway, B. J., Hill, A. A., Wilkinson, J. M., Field, P. R., Murray, B. J. and Carslaw, K. S. (2018). 'Strong control of Southern Ocean cloud reflectivity by ice-nucleating particles'. In: *Proceedings of the National Academy of Sciences* vol. 115, no. 11, pp. 2687–2692. eprint: https://www.pnas.org/content/115/11/2687.full.pdf.

Wikimedia (2015). *English: typical CNN architecture*. [Online; accessed 16-December-2015]. URL: https://commons.wikimedia.org/wiki/File:Typical_cnn.png.

Xiao, H., Zhang, F., He, Q., Liu, P., Yan, F., Miao, L. and Yang, Z. (2019). 'Classification of Ice Crystal Habits Observed From Airborne Cloud Particle Imager by Deep Transfer Learning'. In: *Earth and Space Science* vol. 6, no. 10, pp. 1877–1886. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019EA000636.

Zhang, A., Lipton, Z. C., Li, M. and Smola, A. J. (2020). *Dive into Deep Learning*. https://d2l.ai.

Zhang, Y., Macke, A. and Albers, F. (1999). *Effect of crystal size spectrum and crystal shape on stratiform cirrus radiative forcing*.

Zhuo, J.-Y. and Tan, Z.-M. (2021). 'Physics-augmented Deep Learning to Improve Tropical Cyclone Intensity and Size Estimation from Satellite Imagery'. In: *Monthly Weather Review*.