

THESIS

UTILIZATION OF CONVOLUTIONAL NEURAL NETWORKS IN THE CLASSIFICATION
OF SNOWFLAKES BASED ON IMAGES BY A MULTI-ANGLE SNOWFLAKE CAMERA

Submitted by

Adam Hicks

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2019

Master's Committee:

Advisor: Branislav Notaros

Christine Chiu
Ali Pezeshki

Copyright by Adam Hicks 2019

All Rights Reserved

ABSTRACT

UTILIZATION OF CONVOLUTIONAL NEURAL NETWORKS IN THE CLASSIFICATION OF SNOWFLAKES BASED ON IMAGES BY A MULTI-ANGLE SNOWFLAKE CAMERA

Recent developments in machine learning are applied to in-situ data collected by a Multi-Angle Snowflake Camera (MASC), incorporating convolutional and residual networks in big data environments. These networks provide the following benefits: require little initial preparation and automatic feature extraction, high accuracy and through transfer learning techniques, and relatively small training sets. The networks have large supporting communities and are popular for image processing and classification tasks specifically. In this paper, a convolutional neural network (CNN) is adapted and tasked with classifying images captured from two storm events in December 2014 and February 2015 in Greeley, Colorado. A training data set containing 1400 MASC images was developed by visual inspection of recognizable snowflake geometries and sorted into six distinct classes. The network trained on this data set achieved a mean accuracy of 93.4% and displayed excellent generality. A separate training data set was developed sorting flakes into three classes showcasing distinct degrees of riming. The network was then tasked with classifying images and estimating where flakes fell within this riming scale. The riming degree estimator yields promising initial results but would benefit from larger training sets. Future applications are discussed.

ACKNOWLEDGMENTS

I would first like to acknowledge my advisor Dr. Branislav Notaros for his guidance both as a mentor and educator during my time at Colorado State University. I would like to thank Sanja, Pranav and Cam for making the Electromagnetics lab a fun work environment and for the random discussions that helped relieve the stress of a given semester. I would like to thank my wife, Erica, for putting up with me through this process and always being so understanding (even though she has little interest in machine learning or electromagnetics). Lastly, I'd like to thank the CSU-ECE department faculty and staff for making my studies wonderful and always providing the best effort to help me when I looked lost.

TABLE OF CONTENTS

ABSTRACT.....	ii
ACKNOWLEDGEMENTS.....	iii
LIST OF TABLES.....	v
LIST OF FIGURES.....	vi
1. INTRODUCTION.....	1
2. DATA DESCRIPTION.....	3
2.1 DATA COLLECTION SITE.....	3
2.2 DATA COLLECTION DEVICE.....	4
2.3 IMAGE PRE-PROCESSING.....	6
2.4 CLASSIFICATION SCHEME.....	7
2.4.1 GEOMETRIC TRAINING SET.....	8
2.4.2 RIMING DEGREE ESTIMATION TRAINING SET.....	9
3. METHOD.....	11
3.1 NEURAL NETWORK ARCHITECTURE.....	11
3.2 CONVOLUTIONAL NEURAL NETWORKS.....	14
3.3 RESIDUAL NETWORKS.....	17
3.4 SOFTWARE IMPLIMENTATION.....	18
4.RESULTS.....	19
4.1 GEOMETRIC CLASSIFICATION.....	20
4.2 RIMING DEGREE ESTIMATION.....	22
5. FUTURE WORK.....	26
6. CONCLUSION.....	27
REFERENCES.....	29

LIST OF TABLES

TABLE 1 – RIMING DEGREE CHARACTERISTICS	10
---	----

LIST OF FIGURES

FIGURE 1 – FIELD SITE	4
FIGURE 2 – MASC SYSTEM WITH EXAMPLE IMAGES	5
FIGURE 3 – CLASSIFICATION EXAMPLES	8
FIGURE 4 – FEED FORWARD NETWORK ARCHITECTURE.....	12
FIGURE 5 – CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE (SIMPLIFIED)	15
FIGURE 6 – RESIDUAL LEARNING BLOCK DIAGRAM	18
FIGURE 7- LEARNING CURVE FOR CNN TRAINED ON GEOMETRIC TRAINING SET	20
FIGURE 8 – CONFUSION MATRIX OF GEOMETRICALLY TRAINED CNN	21
FIGURE 9 – LEARNING CURVE FOR CNN TRAINED ON RIMING DEGREE TRAINING SET	23
FIGURE 10 – RIMING DEGREE ESTIMATION EXAMPLES	24
FIGURE 11 – HUMAN ERROR EXAMPLES.....	25
FIGURE 12 – FEATURE BIAS EXAMPLES	26

1. Introduction

The advent of dual polarimetric radar for weather observation and research has increased researchers' capabilities to access and log vital data points within a given weather event. With excellent temporal and spatial resolution, researchers can accurately characterize shapes of the hydrometers that compose a storm [1]. Polarimetric radar provide the horizontal reflectivity, Z_h , differential reflectivity, Z_{dr} , and correlation coefficient, ρ_{hv} , of a field the radar is directed towards. This information gives insight to shape and type of hydrometeor within a storm and is calculated based on models developed by past observation. Scattering models fall short when left to spheroid approximations for frozen hydrometeors [2]. This is especially true at higher frequencies or for larger particles [3] and increases the need for accurate accounting of the varying macro-characteristics of snow within a storm. Atmospheric scientists have drawn strong correlation between the environmental conditions present and the shape snow takes as it forms within a storm [4]. Conditions within a storm are not homogenous, resulting in a wide variety of shapes that continue to change on their path to the ground. While polarimetric radar provides excellent coverage, utilizing ground based (in-situ) devices in tandem with radar has proven more effective in understanding storm composition [5]. One such in-situ device developed specifically to sample snowflakes in free fall is the Multi-Angle Snowflake Camera (MASC). The MASC can capture high resolution images of individual snowflakes, which provides researchers an avenue to study the macro-physical characteristics and make statistical predictions concerning a storm's composition. A deployed MASC is capable of capturing thousands of images an hour, and with typical storms lasting several hours, a fast, accurate and automatic method to organize and process

image data based on shared characteristics is crucial to increased understanding. There exist several classification algorithms known to atmospheric research communities [e.g. 6, 7] but they are limited to large swaths of a storm and not local sampled images.

Machine learning algorithms have made huge strides in the past decade [8] and have a massive community invested in improving and expanding existing algorithms. Previous attempts have been made to apply machine learning to snowflake classification with varying degrees of success. Early backpropagation neural networks were used in [9] and multinomial logistic regression (MLR) in [10]. Praz et al. set the standard for expert research into hydrometeor classification with a separate algorithm for feature extraction and optimization. Convolutional neural networks (CNN) provide several advantages that other algorithms may not include with comparable or better results. CNNs were developed with image processing in mind, which makes them computationally more efficient when compared to other multilayer back-propagation neural networks. They automatically extract features during training, which simplifies any system they are integrated into. CNNs can store these features, which increases their versatility as they are capable of transferring learning from one data set to another and are not limited to specific parameters inherent either to the data set (e.g. resolution, color, or size.) or capturing method (e.g. hardware imperfections transferred into data). Research into deep learning has extended their ability to process complex data without major changes to the algorithm. Finally, CNNs are well understood algorithms that are extremely popular for image processing with a wealth of resources available for implementation. In this paper, the steps needed to develop a hydrometeor classifier using CNNs are presented and advantages to in-situ research are highlighted.

This paper is organized as follows. The method and location for data acquisition is introduced and classification scheme presented in section 2. A brief introduction to the machine

learning algorithms used is discussed in section 3, with key concepts covered for those unfamiliar. The software implementation is also covered in this section for interested parties. Results of training the two networks are presented and implications discussed in section 4. A future perspective in section 5 allows for continued discussion of the results presenting potential avenues of research and is followed by a brief summary in the conclusion.

2. Data Description

This section describes how and under what conditions data was collected. Steps required in preprocessing of image data are covered. The classification criteria for geometric shapes and riming degree estimation are discussed.

2.1 Data Collection Site

The images that compose the training set were taken primarily from two winter weather events using a modified MASC system located at a surface instrumentation field site that was established as part of MASCRAD (MASC + RADar) project [11,12,13,14]. The MASCRAD field site, Figure 1, is located at the Easton Valley View Airport, in La Salle, outside of Greeley, Colorado. This site includes a modified MASC system, a two-dimensional video disdrometer (2DVD), a PLUVIO all-weather precipitation gauge, and a VAISALA weather station, amongst other advanced in-situ measurement instruments. These devices are situated within a double fence intercomparison reference (DFIR) and operate under the umbrella of the CSU-CHILL Radar, a state-of-the-art polarimetric weather radar located 12.92 km away.



Figure 1. Field Site: MASCRAD Snow Field site at Easton Valley Airport, near Greeley, Colorado. In this image: 2/3-scaled double fence intercomparison reference (DFIR) containing surface instrumentation: MASC (top right), 2DVD (top left), Pluvio all-weather precipitation gauge (bottom right) and a meteorological particle spectrometer (MPS) (bottom left).

The image data was collected during two events in the 2014-2015 deployment season of the MASCRAD project. The first took place from December 23rd to 31st in 2014, with the highest density of particles falling during the early morning hours of December 26th. To include enough graupel images, a second weather event was required [15]. This event took place on February 21st – 22nd in 2015.

2.2 Data Collection Device

The Multi-Angle Snowflake Camera, or MASC is the centerpiece of the MASCRAD project. The system was developed by Tim Garrett et al. and the finer details of the system are covered in [16]. For the purposes of this paper, a summary and brief description follows. The

original MASC system is an instrument used to capture high-resolution images and fall speeds of hydrometeors in freefall from three different coplanar perspectives. These cameras are 5 Megapixel (MP) Unibrain Fire-I 980b digital cameras with identical 12.5mm Fujinon lenses. The cameras are spaced on a horizontal ring with 36° separation between cameras with camera-to-common focal center distances of 10 cm. The system used at Colorado State University, Figure 2, has been modified to include two additional cameras at an elevated angle of 55° above the horizon.

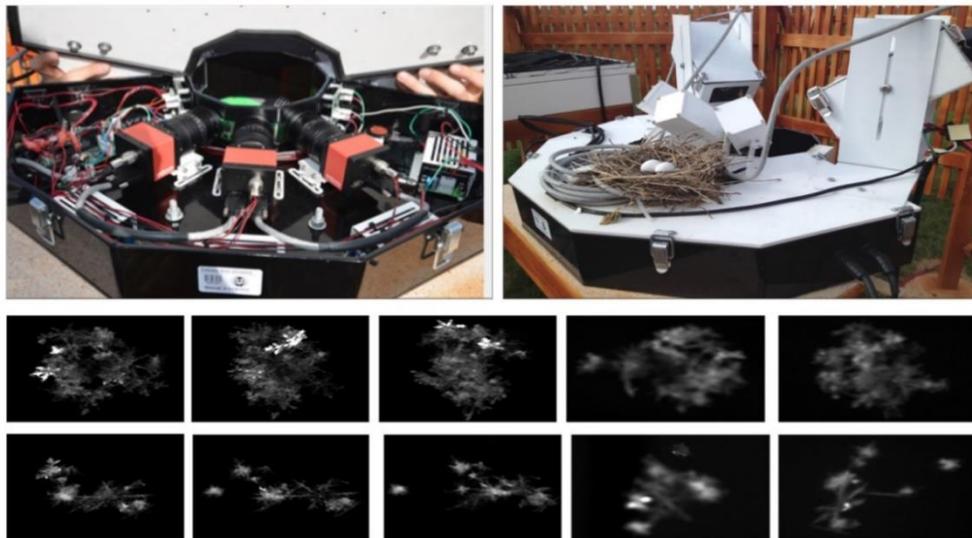


Figure 2. MASC system with example images: Multi-angle snowflake camera (MASC), with three cameras in the horizontal plane for capturing high-resolution photographs of winter precipitation (top left); The CSU MASC system has been modified to included two cameras situated on an elevated plane, poised at 55° above the horizon to aid in the Visual Hull Project (top right); Example images from each camera, the first three columns are from the horizontal cameras and the final two are from the lower resolution raised cameras (bottom two rows, from left to right).

These cameras are 1.2 MP Unibrain Fire-I 785b with 12.5 mm lenses and were included to improve the 3D virtual reconstruction using the visual hull method [17,18]. The system has a horizontal

resolution of 35 μm for the three horizontal cameras and a vertical resolution of 40 μm at 1-m/s fall speed. As hydrometeors fall through the horizontal ring, a near-IR emitter-detector pair sensor array (located on the top rim of the capture volume within the ring) simultaneously triggers the cameras and a flash (LEDs). The cameras have a maximum triggering rate of 2 Hz, a hardware limitation within the cameras, not set by the emitter-sensor pair array. To calculate fall speed, the time between upper near-IR emitter-sensor trigger and lower near-IR emitter-sensor is calculated then divided into the distance between the two sensor planes.

2.3 Image Pre-Processing

As a CNN automatically extracts features (numerical descriptors common between classes) during training that are to be used in classification, the image data base requires minimal preprocessing. A simple brightness thresholding is utilized to remove the majority of blank, dim or blurry images. Generally, a MASC is deployed during storming conditions, those with heavy wind or flurried snow, which can cause the MASC to trigger without a flake in the focal area, making the need for thresholding crucial to processing. After thresholding to remove poor quality images, the creation of a training set requires a one-to-one correlation between an image of a flake and the class the image is being assigned to, i.e. there can only be one flake per image. The task of separating images was automatically performed using a cropping script developed to find the brightest point of an image, locate the surrounding edges through their calculated standard deviations, and remove the flake to be saved in another location. The script then performs this action again until all independent bright spots have been cropped in a given image. It isn't perfect (~10% of flakes need repairs) but is exceptionally more efficient than cropping images by hand.

2.4 Classification Scheme

With machine learning algorithms, there are two categories of how learning is conducted: supervised or unsupervised. CNNs are supervised learning algorithms, therefore they require the development of a labeled data set, referred to as the training set. This training set allows a human operator to dictate how the network makes decisions, by providing desired data for the network to make comparisons against. This contrasts unsupervised learning, where a network runs until it converges on a pattern or an end condition is reached. It is no surprise that supervised learning is more efficient.

The training set is developed by human inspection following a predetermined classification scheme. The idiom is that every snowflake is unique, therefore it is no surprise that there are a variety of attempts to classify them [19,20], with little commonality between schemes. The scheme used in this paper was adopted from Praz et al., who developed a snowflake classifier using Multinomial Logistic Regression (MLR) and will be summarized in this section.

The scheme utilizes the nine categories of snowflakes detailed in [21], with some changes due to data availability and simplification. They introduce the category of Aggregate particles, which are single snowflakes that are the result of the in-air collision of two or more particles and Small Particles, snowflakes whose feature characteristics are too small to categorize. They also combine the category of needle and column type snowflakes, as they share similar characteristics. Their result was 10 individual categories that include Aggregate (AG), Small Particle (SP), Columnar Crystal (CC; the resulting combination of needle and column particulates), Planar Crystal (PC), a combination of columnar crystals, a combination of planar crystals, combination of columnar and planar crystals, graupel (GR), irregular snow crystal and germ of snow. Due to limited sample representation, the training set used in this paper includes only the most populated

five classes (AG, CC, PC, SP, and GR) seen in Figure 3, although more classes may be added to the classifier as sample data is accrued. A different training set is used for each classifier, and therefore their development considerations are unique to each set.

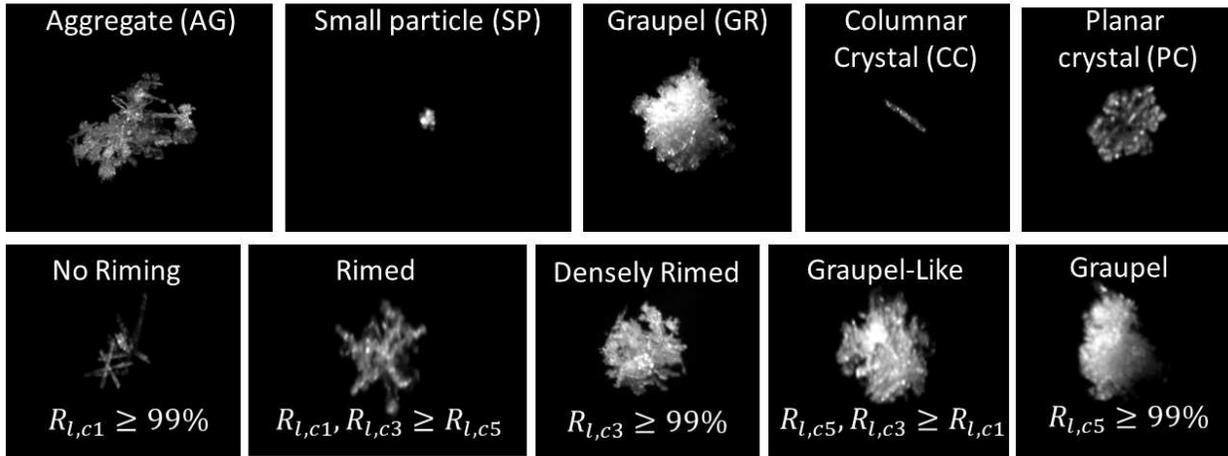


Figure 3. Classification Examples: Examples of MASC images characterizing the geometric classes (top row) and riming degree estimations (bottom row).

2.4.1 Geometric Training Set

At the heart of supervised learning is the comparison between the ideal value (established in the training set) and a value calculated by the network, that is later refined through the learning process. This is an oversimplification, with more in-depth analysis provided in section 3, but it is important to understand that the more unique the classes within a training set are from each other, the better the network will perform, given a limited training set (which seems obvious but cannot be over-stated). The training set for geometric classification was developed by selecting individual images of the flakes that best represented their respective classes, with decision emphasis placed on discernable flake silhouettes. The result is a training set of ~1450 samples.

2.4.2 Riming Degree Estimation Training Set

Riming degree estimations attempt to calculate the amount of cloud frozen droplets that accrue on a snowflake's surface as the flake falls through the atmosphere. The MASC system captures images with enough detailed resolution that the degree of riming can be considered a feature of the image. There are two approaches to riming degree estimation considered in this paper. The first approach divides riming degree into 5 classes utilizing the classification scheme adapted by Praz et al. The second approach is a proposal that capitalizes on the unique nature of CNNs and warrants further exploration.

Praz et al. classify riming degree based on the image criteria summarized in Table 1. These five degrees of riming, R_d are discrete classes between [1,5] as developed by [22] and are then mapped by Praz et al. to a continuous index, R_c between [0,1], using the sinusoidal function

$$R_c = \frac{1}{2} \left(\sin \left(\frac{\pi}{4} (R_d - 3) \right) + 1 \right).$$

Table 1. Riming Degree Characteristics: A physical description characterizing the degree of the riming on a given flake, with numerical representations as described by Mosimann et al, in column 2. Column 1 is degree estimations utilized by Praz et al. Column 3 is the probability estimates used in this paper.

R_c $\in [0,1]$	R_d $\in [1,5]$	$R_l \in [1,5]$	Coverage of the surface	Description
0	1 (none)	1.0 ($R_{l,c1} \geq 99\%$)	0%	No cloud droplets on the surface. Flakes are detailed and delicate in appearance.
0.15	2 (rimed)	1.01 to 2.99 ($R_{l,c1}, R_{l,c3} > R_{l,5}$)	$\simeq 50\%$	Up to half of the surface is covered with cloud droplets. There may be delicate features, but some clumping has occurred.
0.5	3 (densely rimed)	3.0 ($R_{l,c3} \geq 99\%$)	$\simeq 100\%$	The entire flake is covered with cloud droplets, but the general shape is conserved.
0.85	4 (graupel- like)	3.01 to 4.99 ($R_{l,c5}, R_{l,c3} > R_{l,c1}$)	$> 100\%$	The entire flake is heavily covered with cloud droplets, to the point where the original shape is barely recognizable
1.0	5 (graupel)	5.0 ($R_{l,c5} \geq 99\%$)	$\gg 100\%$	The entire flake is heavily covered with cloud droplets. Original shape is no longer distinguishable and has entered the class of graupel.

The degree label decisions are based on educated opinion through the observation of a captured image. Questions arise when pondering the level of accuracy that a human observer can achieve in their estimation. For example, the difference between a riming degree estimation of 3.1 and 3.2 is often arbitrary and a matter of opinion. To address this concern, it is proposed to utilize the posterior distribution that a CNN uses to make classification decisions as the deciding factor in riming degree estimation.

Developing the training set for this approach, the classification scheme focuses on the three easily identified classes for riming estimation, then allows the network to assign probability estimations (clarified at the end of section 3.2) for how closely a flake resembles those classes, labeled $R_{l,c}$ and characterized in Table 1. The estimate training classes are class 1, where flakes are the least rimed (no riming present), class 3, where a flake is rimed but geometric shape is preserved, and class 5, where the flake is fully rimed (graupel). This simplifies the classification

process and increases the consistency in decision making when applying labels while developing the training set. The result after classification is flakes that don't fall directly within these three classes are weighted somewhere between. For example, if a flake's riming degree estimation is $R_{l,c1} = 32\%$ $R_{l,c3} = 58\%$ $R_{l,c5} = 10\%$, the estimation on Praz et al.'s scale is akin to ~ 2.78 . The numerical values are used as an example, with a simple linear mapping. As more images are processed, a more accurate mapping can be developed if desired. This provides a continuous riming degree estimation and the opportunity for greater accuracy in estimation. Attention must be paid to the number of geometric classes represented in each riming category, as it is important to have uniform representation. This restriction limits the size of training set available, but the approach looks promising and will warrant further testing as processed data becomes available.

3. Method

A brief discussion of the network architecture is presented in this section. An overview of key concepts pertaining to CNNs and deep learning is provided. The software implementation and input parameters outlined before results of the network training is discussed.

3.1 Neural Network Architecture

The organic brain has an amazing ability to process information very quickly. Consider, for example, the act of driving a car or the simple act of walking down the street. The fact that these actions require little conscious thought highlights the ability of the brain to rapidly take incoming information, assign meaning, and make decisions. This is accomplished through a

complex interconnected system of neurons that process information in parallel. It is no surprise then that considerable effort has been made to mimic the complex way the brain processes information with machine learning algorithms.

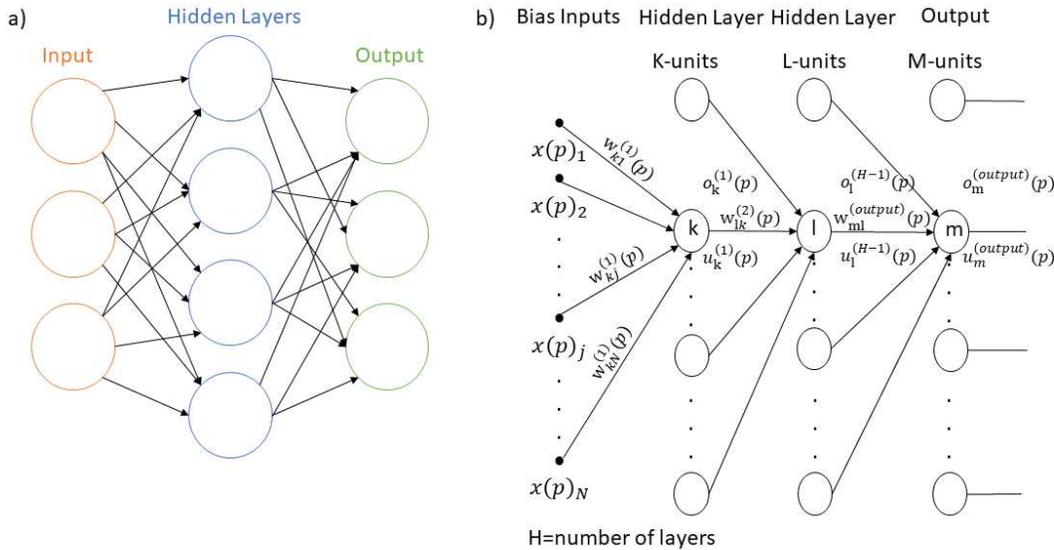


Figure 4. Feed Forward Network Architecture: (a) Feed Forward Network architecture demonstrating how each neuron is connected to every neuron in the previous layer. (b) Algorithm for feed forward learning process within the network.

With artificial neural networks, more commonly known as “neural networks,” the basic architecture of the brain is recreated in a logical algorithm. The basis for the network architecture used in this paper is called the “Multi-Layer Perceptron (MLP),” which is a feed forward neural network with back-propagation and general structure is represented in Figure 4a, and as an algorithm in Figure 4b. In general, there is an input signal which is passed to the hidden layers for processing, with final decision making the result of the output layer. The algorithm is “feed forward” in reference to the direction of information and the “back-propagation” describes the learning process, which is detailed below.

Referring to Figure 4b, the forward pass of the learning algorithm applies the input value $x_{1\dots N}$ of training sample p with a weighted bias of $w_{kj}^{(1)}(p)$. This bias serves to both connect the input j to neuron k and to refine the error estimation of the classifier, E_p and are generally random values between a range loosely defined by the number of inputs [23]. The potential of neuron k is represented by

$$u_k^{(1)}(p) = \sum_{j=1}^{N+1} w_{kj}^{(1)}(p)x_j(p), \forall k \in [1, K]$$

$f(\cdot)$ is a non-linear activation function, commonly the logistic function [24] or more recently (and in the case of this paper), the Rectified Linear Unit (ReLU).

$$f(u) = \frac{1}{1 + \exp(-u)} \text{ or } ReLU(u) = \max(0, u)$$

This activation function determines the output of the current neuron, which in turn becomes the input of the next layer within the network.

$$o_k^{(1)}(p) = f\left(u_k^{(1)}(p)\right), \forall k \in [1, K]$$

This process is repeated until the final layer of the network is reached, labeled “Output” in Figure 4b. It is at this point the sum squared error of the classifier can be calculated.

$$E_p = \frac{1}{2} \sum_{m=1}^M \left(d_m^{(output)}(p) - o_m^{output}(p) \right)^2$$

Our goal is to minimize E_p , such that the desired output, $d_m^{(output)}(p)$, is as close to the calculated output, $o_m^{output}(p)$, as possible. It is unlikely that E_p is within acceptable parameters on the first pass of learning, therefore there needs to be some method to refine the weights to approach convergence. This requires the calculating the partial derivative of E_p with respect to the weight in the final layer, in Figure 4b that is $w_{ml}^{(output)}$. The full derivation falls outside the scope of this

paper but can be found in various forms in [24,25,26]. The eventual result for this three-layer network is

$$\Delta w_{ml}^{(output)} = -\mu \nabla E_p(p) = \mu e_m^{output}(p) f' \left(u_m^{(output)}(p) \right) o_l^{H-1}(p),$$

$$\forall m \in [1, M] \text{ and } l \in [1, L]$$

where

$$e_m^{output} = d_m^{(output)}(p) - o_m^{output}(p)$$

And

$$\mu = \text{learning factor}$$

Repeating this derivation for weight updates within the hidden layers shows that the calculated layer from the output is back propagated through the network.

$$\Delta w_{ml}^{(output)} = \mu e_m^{output}(p) f' \left(u_m^{(output)}(p) \right) o_l^{H-1}(p),$$

$$\Delta w_{ml}^{(output)} = w_{ml}^{(output)}(p+1) - w_{ml}^{(output)}(p), \text{ and } m \in [1, M]$$

$$\Delta w_{lk}^{(H-1)} = \mu f' \left(u_l^{(H-1)}(p) \right) o_k^1(p) \sum_{m=1}^M w_{ml}^{(output)}(p) e_m^{output}(p) f' \left(u_m^{(output)}(p) \right),$$

$$\Delta w_{lk}^{(H-1)} = w_{lk}^{(H-1)}(p+1) - w_{lk}^{(H-1)}(p), \text{ and } \forall l \in [1, L]$$

$$\Delta w_{kj}^{(1)} = \mu f' \left(u_k^{(1)}(p) \right) x_j(p) \sum_{l=1}^L w_{lk}^{(H-1)}(p) f' \left(u_l^{(H-1)}(p) \right) \left[\sum_{m=1}^M w_{ml}^{(output)}(p) e_m^{output}(p) f' \left(u_m^{(output)}(p) \right) \right]$$

$$\Delta w_{kj}^{(1)} = w_{kj}^{(1)}(p+1) - w_{kj}^{(1)}(p), \text{ and } \forall k \in [1, K]$$

This back and forth is continued until the error stops dropping or an acceptable value is reached.

3.2 Convolutional Neural Networks

With image classification, the input value of $x_{1...N}$ is a pixel of the image in the training set and this presents a major limitation to typical MLP architectures. Each input has an individual weight value per neuron in the network, so for even a moderately deep network (the number of

hidden layers in the network refers to how deep a network is), the result is upwards of hundreds of thousands of weights that require refinement, or even millions for very high resolutions, making this computationally inefficient for modern practical purposes. CNNs solve many of the problems MLPs experience for image processing and have proven useful with other data types [e.g. 27,28]. A visual representation of the CNN architecture is seen in Figure 5.

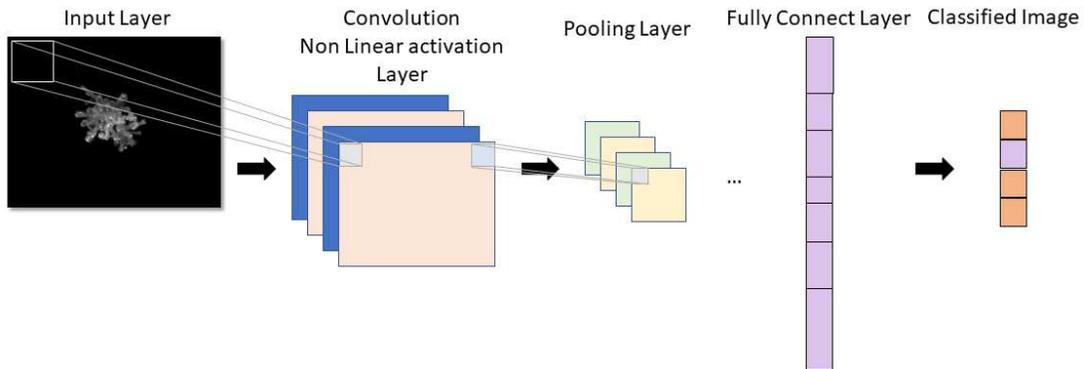


Figure 5. Convolutional Neural Network Architecture (simplified): Highlights the key steps within a typical convolutional neural network; The input layer is scanned with a $[i,j]$ weight matrix to create the feature map; The feature map is averaged or maxed to increase network efficiency in the pooling layer; Decision making occurs at the final fully connected layer.

The defining features of the CNN architecture are the inclusion of the convolution layer, a pooling layer and the fully connected layer. The convolution operator introduces several advantages to the architecture, namely sparse interactions, parameter sharing and equivariance to translation [29]. As is depicted in Figure 5, the convolution layer creates a series of feature maps by scanning a weight matrix of size $[i,j]$ over the surface of the input data. This reduces the number of parameters the network must consider and introduces weight sharing. Comparing to Figure 4b, the inputs are

blocked or shared between units with shared weights. The result is a reduction in memory requirements and an improvement of statistical efficiency [29]. After the convolution is performed, the linear activations that result are then passed through a non-linear activation function, like the ReLU which has largely replaced other activation functions, as it improves efficiency without a reduction in accuracy [29]. The convolution layer can be repeated with different sized weight matrices to extract more and more abstract features. Yann Le Cun and colleagues introduced the pooling, or subsampling layer to their model **LeNet5** to achieve shift invariance [24]. This is accomplished by either averaging or producing a max over a small window of the convolution layer. This step is especially vital for image classification, as it allows the network to extract features without caring exactly where the features are located [29]. An additional benefit of this property is that it makes transfer learning with pretrained networks possible [30], which significantly reduces the size of training set necessary for new classification schemes. **LeNet5** followed every convolution with a pooling layer, but this is unnecessary. It has been shown that the best results for complex data sets apply a few pooling layers after the first series of convolution layers and a final pooling layer after the next to last convolution layer [31]. The final few layers of the network will consist of fully connected layers which are akin to regular neural networks. The output of these layers after activation are passed to a *softmax* operator $\sigma(\cdot)$ where high-level decisions are made, and after several passes, a class label is applied. The learning process is summarized below.

$$y_j = \sigma(\underline{o})_j = \frac{\exp(z_j)}{\sum_{k=1}^K \exp(z_k)} \quad \forall j \in [1, K]$$

Where $y_j = p(C_j|\underline{x})$, a posterior distribution over the available classes, where C_j is the class and \underline{x} is the input vector. In section 2.4.2, a term “likelihood” is used to provide a label to the degree of riming present for a given flake, for added clarity $y_1 = R_{l,c1}$, $y_3 = R_{l,c3}$ and $y_5 = R_{l,c5}$. If \mathcal{W}

is the set of all parameters for the network, and the set of training samples is $\{\underline{x}_p, \underline{d}_p\}_{p=1}^P$, then using y_j as an input to a cost function $\varepsilon(\cdot)$, whose minimization is achieved through a modified error backpropagation [23].

$$\varepsilon(\mathcal{W}) = \frac{1}{N} \sum_{p=1}^P \left\| \underline{d}_p - \underline{y}(\underline{x}_p; \mathcal{W}) \right\|^2$$

3.3 Residual Networks

For complicated data sets, network comparisons have shown that increased depth improves network accuracy [32,33]. Deep networks have more capacity for different level (low/mid/high) features [34] and the top performing networks on the ImageNet dataset employed deep models [35]. The tradeoff is that the deeper the network becomes, the more the accuracy saturates and begins to quickly decline [35]. To combat this, a group of researchers from Microsoft Research, He et al., developed the residual network architecture. A residual network architecture is very similar to a convolutional neural network, with one addition. Every few convolution layers, a short cut identity is included Figure 6, which fits the layers to a residual mapping, instead of hoping they come to a desired mapping naturally [35].

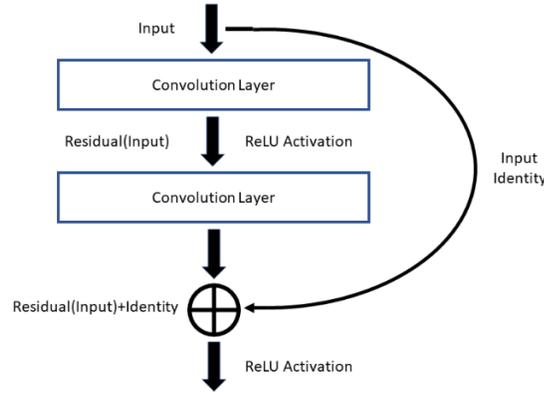


Figure 6. Residual Learning Block diagram: Represents the additional step introduced to a typical neural network architecture.

3.4 Software Implementation

A major benefit to utilizing convolutional neural networks is their widespread popularity for image classification. There are extensive software toolboxes available, both commercial and open source, with detailed walkthroughs for a variety of tasks and applications. Due to the automatic feature extraction inherent to their algorithm, CNNs can be applied and operated by non-expert technicians, increasing their functionality as a post-processing front end for big data tasks. The toolboxes and set up utilized in this paper are described in the section below.

The same network architecture is used for geometric classification and riming degree estimation and can be run in parallel. The network was implemented using Matlab™ 2018b, with the Deep Learning and Machine Learning Toolboxes at Colorado State University. The network architecture is the ResNet-50, which was chosen over AlexNet [36] or GoogleNet[33] due to its balance between accuracy and speed [37] and has been pretrained on the ImageNet database [38] to reduce the size of the training set data necessary to be an effective classifier. The more

complicated the classification task, the larger the data set needed to extract relevant features [35]. Geometric classification, however, is a common problem in image classification, and therefore it is an ideal candidate for pretraining. A network pretrained for image classification can reuse many of the features extracted and made task specific on a reduced training data set [39]. Each class in the training set is limited to the smallest populated class, for instance, the Planar Crystal class is the least populated and has 290 images, this sets an upper limit of 290 on all classes. This is done so overrepresentation does not occur in the training phase. The entire training data set is divided into two categories, “training” and “validation”. As is customary, 70% of the entire training data set is randomly selected and stored in the “training” category, with the remaining 30% saved for validation of the network performance. This separation reduces the likelihood of the network overfitting to data and allows for an accurate test for generalization. The training images are randomly reflected, translated and scaled within a defined range to improve the networks invariance to small changes [24]. Dropout [40] is employed on the pretrained network weights to further reduce any overfitting that may occur. The network performance is determined by mean square and the loss function is calculated as stochastic gradient descent. Validation occurs every three iterations over 10 epochs, although experimentation has shown that 6 epochs are enough to improve training time without loss of performance as is adopted for later tests. The learning rate for both networks is 0.0003.

4. Results

This section is reserved to discuss the results of the geometric classification and riming degree estimation. The learning curves for each network are presented with confusion matrices to show each networks generalization.

4.1 Geometric Classification

The results of network training for geometric classification are shown in Figure 7. The training data set included ~1450 images and training occurred over 900 iterations. The network achieves a mean accuracy of 93.4% with a loss function of ~0.2 and little variance between runs. Training time was 13 minutes and 23 seconds. This is very good accuracy given the size of the data set and the training time is reduced when limited to 6 epochs (from ~13 min. to ~8 min.).

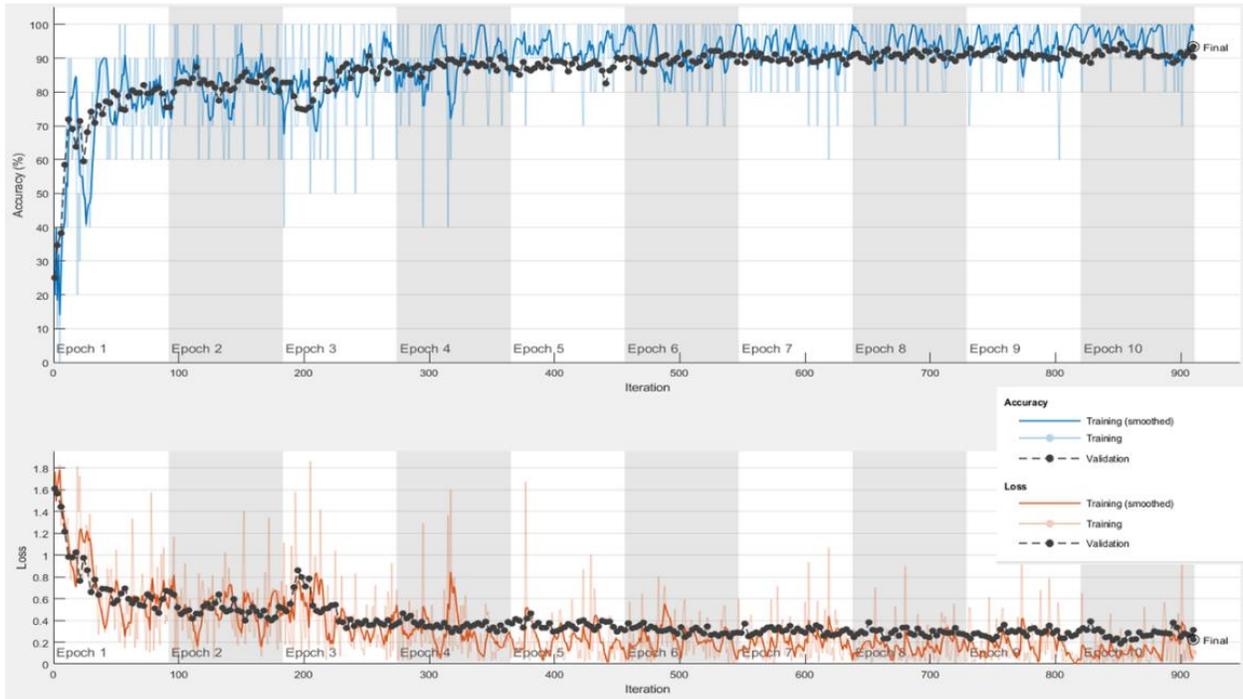


Figure 7. Learning Curve for CNN trained on Geometric Training Set: A mean average is validated every three iterations (black dots) over 10 epochs with a mean average of 93.4% (top), the same parameters are shared by the loss function evaluation with a final value of ~0.2 (bottom).

To test the generalization of the network, it is then used to classify ~400 flakes not included in training, the results of which are shown in Figure 8.

Confusion Matrix

Output Class	AG	65 16.5%	0 0.0%	2 0.5%	1 0.3%	0 0.0%	95.6% 4.4%
	CC	0 0.0%	76 19.2%	0 0.0%	1 0.3%	0 0.0%	98.7% 1.3%
	GR	1 0.3%	0 0.0%	75 19.0%	0 0.0%	0 0.0%	98.7% 1.3%
	PC	13 3.3%	1 0.3%	0 0.0%	75 19.0%	1 0.3%	83.3% 16.7%
	SP	0 0.0%	2 0.5%	2 0.5%	2 0.5%	78 19.7%	92.9% 7.1%
		82.3% 17.7%	96.2% 3.8%	94.9% 5.1%	94.9% 5.1%	98.7% 1.3%	93.4% 6.6%
	AG	CC	GR	PC	SP		Target Class

Figure 8. Confusion Matrix of Geometrically Trained CNN: The network was tasked with classifying 395 flakes to test generalization. The left axis is what the network labeled the input data to as, and the bottom axis is what the input data was. Green boxes represent correctly classified images, with total number in bold and percentage of total immediately below. Green text represents a correct classification, while the red percentage is the misclassification of the network. The overall network accuracy is shown in the bottom right corner.

The relevant information from the confusion matrix can be seen in the bottom row and right-most column. The bottom row is the percentage value of when the network was presented with an image and correctly classified it on a per class basis. The right-most column is out of class accuracy, i.e. how often the network confused an image within a class for something else.

The confusion matrix shows that the network has some issues making decisions between planar crystal and aggregate flakes. This is consistent with established reasoning, as the planar crystal class is most visibly similar to aggregate flakes and was also the smallest represented class for training which limited the diversity of available samples. Increasing the number of samples should increase the network's ability to differentiate between the two classes.

4.2 Riming Degree Estimation

With riming degree estimation, two networks were trained for comparison. One of the networks kept the discrete five classes determined using Praz et al's [1,5] classification scheme (Table 1), while the other network removes the two classes labeled 2 and 4, relying instead on the posterior distribution, or "likelihood" estimation, to assign a continuous label to flakes that fall within the edge cases. Comparing the learning curves of the two networks (Figure 9), it is clear that classes 2 and 4 are difficult for the network to analyze.

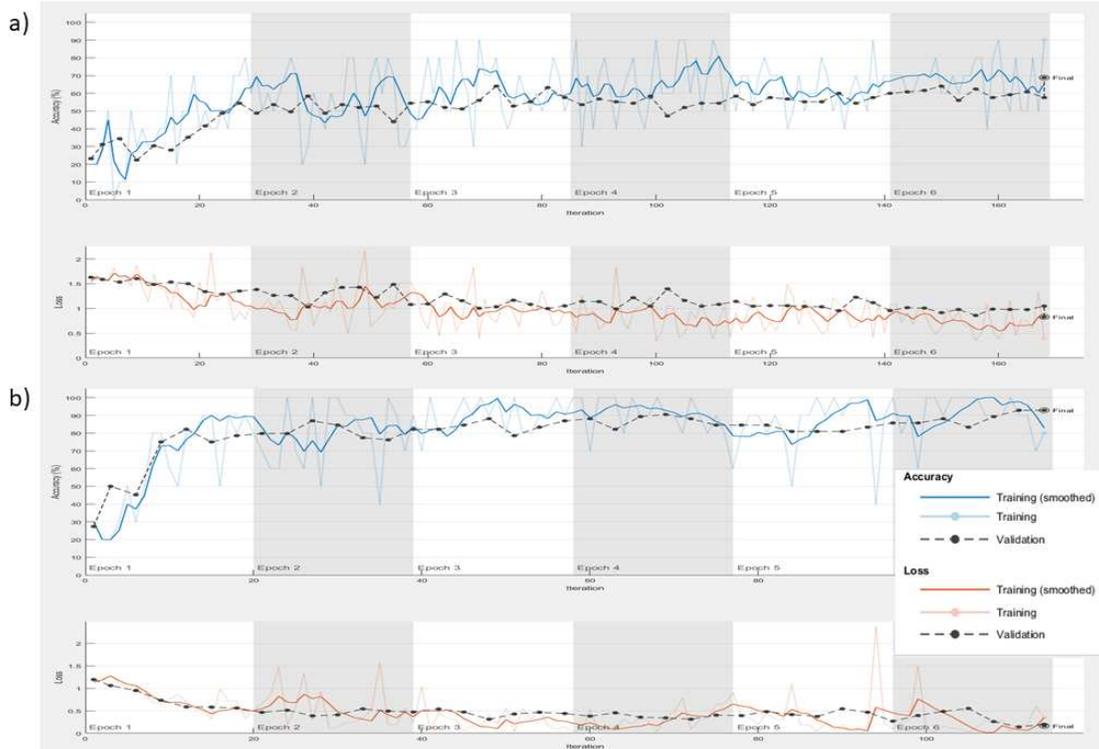


Figure 9. Learning Curve for CNN trained on Riming Degree Training Set: (a) A network was trained using classes composed of images representing the five discrete riming degree values [1,5] described in Table 1;**(b)** A separate network was trained using the same image data, only removing the images compromising class 2 and 4.

The result of training is a mean accuracy of 68.8% and a loss of ~1 for the network trained on five classes (Figure 7a) and a mean accuracy of 92.4% with a loss of <0.3 for the network with only three classes (Figure 7b). The second network is then tasked with classifying a data set that includes samples from all five classes. Results of the applied “likelihood” percentage, $R_{l,c\#}$, can be seen in the following Figure 10. Here is shown the network’s capability to apply a riming degree estimation determined by features found in the edge case classes used in training. In Figure 10, examples of individual degree estimations are shown in the top-most row, while the network’s capability to distinguish between flakes with similar characteristics is showcased on the bottom.

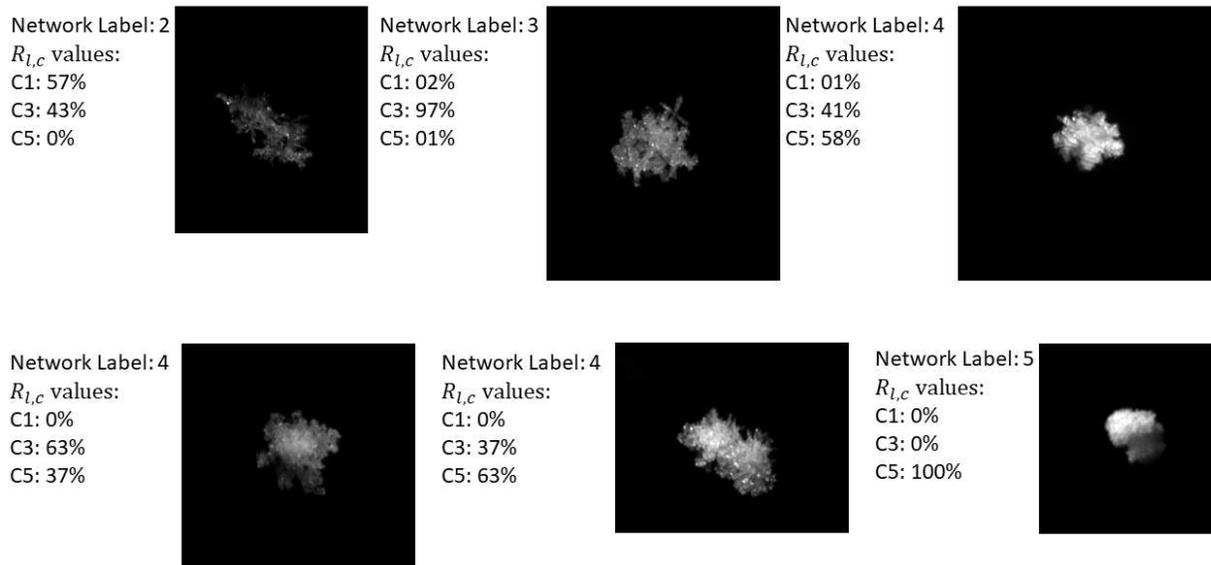


Figure 10. Riming Degree Estimation Examples: The values represent the output of the riming degree estimation network, the top row highlights the networks range, while the bottom most demonstrates its ability to differentiate between similar flakes.

An inherent benefit of this estimator is that it removes some of the guess-work involved when a human user classifies flakes individually. While the human processor may introduce error through fatigue or the result of an immediacy bias, a network will not suffer from these potential pitfalls and may catch misclassified images (Figure 11).

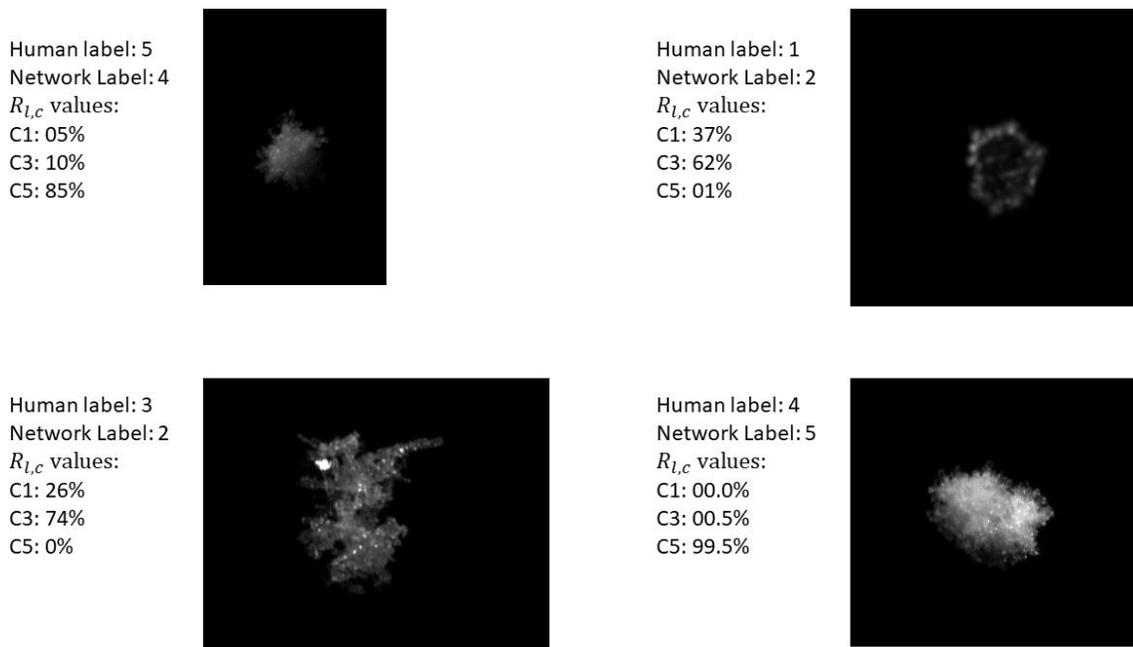


Figure 11. Human Error Examples: Utilizing a neural network for riming degree estimation has the advantage of quantified data driven decision making.

Developing a riming degree estimator relies almost entirely upon features that are unique to images of snowflakes. Best results will be derived from training sets where features unrelated to riming degree estimation are equally represented, so that they are effectively removed from the decision-making process. Without equal representation, the network can develop a bias based on which feature is more prevalent. An example of this bias can be seen in Figure 12 and is a result of columnar crystal images being overrepresented in the “no riming” category. In Figure 12a, the estimator gives an accurate prediction indicated by the $R_{l,c\#}$ values, but the bias is evident in flakes Figure 12b and 12c, resulting in a severe under estimation and overestimation respectively. In the storms sampled, flakes demonstrating no degree of riming were relatively rare, which is why columnar crystal flakes compose the bulk of the data set. To achieve equal representation of

unrelated flake features, a wider variety of storms from different season will need to be processed, which is discussed further in section 5.

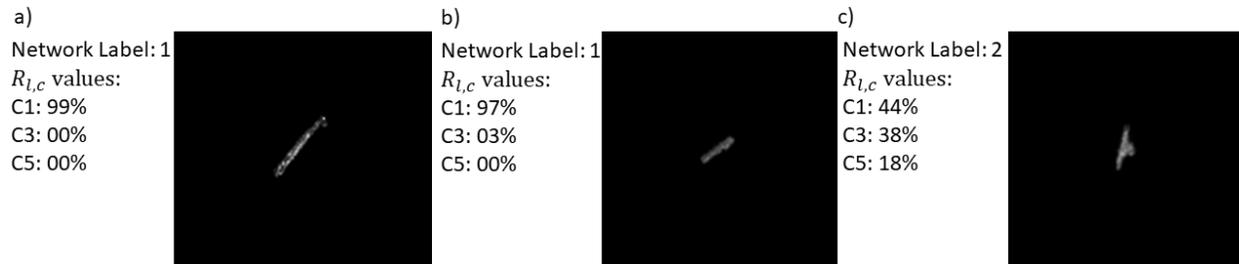


Figure 12. Feature Bias Examples: (a) An image of a correctly labeled image by the Riming degree estimator. (b) The network underestimates the degree of riming in the image due to feature bias present in the “no riming” class. Expected results for this image should be a higher value for $R_{l,c3}$. (c) The network overestimates the degree of riming due to feature bias from the “graupel” class.

5. Future Work

In this section, the future tasks and research projects related to snowflake classification are briefly discussed. The first task for the network will be to process the backlog of storms that have been recorded in the MASCRAD project. This will aid in related research projects, such as linking the macro-characteristics of snowflakes to the scattering properties through 3D modeling and numerical analysis [17,18]. Currently, the network is best equipped for small-batch processing (few hundred flakes per batch) due to the size of its training data set. By processing more data with diverse environmental conditions, the network will eventually be able to process bulk data that number in the thousands. With additional flake variety, sub-classes and rarer classes may be introduced to the network, expanding from the five classes currently utilized. It may prove

beneficial to explore methods to reduce the dependency are large data sets. Whether methods to introduce scarcity sampling into the input or more advanced dimensionality reducing techniques like wavelets.

The previous discussion in section 4.2 has shown the promising results of the riming degree estimation by using a CNN, but there remain developments to be made and tested before the network is relied upon in academic research. Additional data from the backlog of storms will serve riming degree estimation as well. Feature extraction requires a large number of images and features unique to snowflake images cannot be compensated for by pretraining. More diverse flakes will help remove any bias the network develops. Looking for means to incorporate other data types into the network's decision making might prove beneficial. Introducing features such as fall speed, volume of flake or mass would make riming degree estimations more legitimate to the research community. Processing data from different seasons may lead to additional classification tasks, like wet or dry snow, and will be considered as they arise.

6. Conclusion

In this paper, recent developments in machine learning are applied to the problem of hydrometeor classification. Utilizing convolutional neural networks, the task of classifying flakes based on geometric characteristics and riming degree is undertaken. Convolutional neural networks are ideal for image classification due to their efficient data handling, automatic feature extraction, versatility and relative ease of application. A training set is developed primarily from two winter precipitation events and consists of 1450 flakes. Six geometric classes are defined based on observable physical characteristics of the snowflakes. These classes are aggregate, columnar

crystal, planar crystal, small particle, graupel and a combination of columnar and planar (although the latter is discarded due to rarity until further data is processed). Geometric training focused on individual flakes that are easily identifiable as members of a single class. The result of training is a network with 93.4% classification accuracy. This performance is sufficient to begin processing the hours of recorded backlog data from the MASCRAD project and begin growing the training data set for continued network development.

The riming degree estimator shows promise. The training set is composed of images from all geometric classes separated (where applicable) into three categories: no riming, rimed and graupel. The network achieves 92% accuracy when estimating flakes that fall into these categories of riming. The probabilistic estimation that results at the output layer of the CNN is then used to gauge where a flake falls within these three degrees of riming (classes 2 and 4 from Praz et al). The high degree of accuracy is maintained by the network in determining whether a flake is more rimed or less rimed (belonging in class 2 or 4) but more sample data must be developed to increase precision of estimation and remove the influence of feature bias. A method for mapping the estimation to fit Mosimann et al. must be developed as more output data becomes available.

The network developed in this paper will be used in the processing of MASCRAD data, but the architecture is ideal for any solid hydrometeor classification task and is suitable as a post-processing front end to any image-based particle recording device. The network is ideal for users with limited experience in image processing, machine learning or atmospheric research. Organized data by physical characteristics and accurate riming degree estimations will help further research into hydrometeor scattering. This will potentially improve propagation models as higher frequencies continue to be explored, whether for use in remote sensing of hydrometeors, communication or other related fields.

References

1. Straka, J., Zrníć, D. S., and Ryzhkov, A. V., (2000). Bulk hydrometeor classification and quantification using polarimetric radar data: Synthesis of Relations. In *J. Appl. Meteor.*, vol. **39**, 1341–1372.
2. Tyynelä, J., Leinonen, J., Moisseev, D., and Nousiainen, T., (2011). Radar Backscattering from Snowflakes: Comparison of Fractal, Aggregate, and Soft Spheroid Models. In *J. Atmos. Oceanic Technol.*, vol. **28**, 1365–1372.
3. Kim, M. J., (2006). Single scattering parameters of randomly oriented snow particles at microwave frequencies, *Journal of Geophysical Research*, col. 111, D14201, doi:10.1029/2005JD006892.
4. Libbrecht, K. G., (2017). Physical Dynamics of Ice Crystal Growth, *Annu. Rev. Mater. 2017*. 47:271-295, doi: 10.1146/annurev-matsci-070616
5. Zhang, G., S. Luchs, A. Ryzhkov, M. Xue, L. Ryzhkova, and Q. Cao, (2011). Winter Precipitation Microphysics Characterized by Polarimetric Radar and Video Disdrometer Observations in Central Oklahoma. *J. Appl. Meteor. Climatol.*, vol. **50**, 1558–1570.
6. Chandrasekar, V., Keranen, R., Lim, S., and Moisseev, D., (2013). Recent advances in classification of observations from dual polarization weather radars, *Atmos. Res.* 119, 97-111, doi: 10.1016/j.atmosres.2011.08.014.
7. Besic, N., Figueras i Ventura, J., Grazioli, J., Gabella, M., Germann, U., and Berne, A., (2016). Hydrometeor classification through statistical clustering of polarimetric radar measurements: a semi-supervised approach, *Atmos. Meas. Tech.* 9, 4425-4445, doi: 10.5194/amt-9-4425-2016.
8. Minar, M.N., Naher, J., (2018). Recent Advances in Deep Learning: An Overview. 10.13140/RG.2.2.24831.10403.
9. Feind, R. E., (2006). Comparison of three classification methodologies for 2D probe hydrometeor images obtained from the armored T-28 aircraft, PhD thesis, South Dakota School of Mines and Technology.
10. Praz, C., Yves-Alain, R., and Berne, A., (2017). Solid hydrometeor classification and riming degree estimation from pictures collected with a Multi-Angle Snowflake Camera, *Atmos. Meas. Tech.*, 10, 1335-1357.
11. Notaros B., V. N. Bringi, C. Kleinkort, G.-J. Huang, E. Chobanyan, M. Thurai, O. Notaros, A. Manic, A. Newman, P. Kennedy, J. Hubbert, T. Lim, W. Brown, and M. Ilic,

- (2015a). Measurement and Characterization of Winter Precipitation at MASCRAD Snow Field Site. *Proceedings of the 2015 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting*, July 19-25, 2015, Vancouver, BC, Canada.
12. Notaros, B. M., V. N. Bringi, A. J. Newman, C. Kleinkort, G.-J. Huang, P. Kennedy, and M. Thurai, (2015b). Accurate Characterization of Winter Precipitation Using In-Situ Instrumentation, CSU-CHILL Radar, and Advanced Scattering Methods. *2015 AGU Fall Meeting*, 14-18 December 2015, San Francisco, CA.
13. Kennedy, P. C., C. Kleinkort, G.-J. Huang, M. Thurai, A. Newman, J. Hubbert, S. Rutledge, V. N. Bringi, and B. M. Notaros, (2015). Preliminary Results from the Multi-Angle Snowflake Camera and Radar (MASCRAD) Project. *American Meteorological Society's 37th Conference on Radar Meteorology*, 14-18 September, 2015, Norman, OK.
14. Bringi, V. N., B. Notaros, C. Kleinkort, G.-J. Huang, M. Thurai, and P. Kennedy, (2015). Comprehensive Analysis of an Unusual Winter Graupel Shower Event Recorded by an S-Band Polarimetric Radar and Two Optical Imaging Surface Instruments. *American Meteorological Society's 37th Conference on Radar Meteorology*, 14-18 September, 2015, Norman, OK.
15. Bang, W., Kim, K., GyuWon, L., Thurai, M., Kennedy, P., Bringi, V.N., Notaros, B. (2016). Microphysical Characteristics Analysis of Three Heavy Snowfall Events from the MASCRAD Campaign in Greeley, Colorado, USA, *URSI Asia-Pacific Radio Science Conference*, August 21-25, 2016, Seoul, Korea.
16. Garrett, T. J., Fallgatter, C., Shkurko, K., and Howlett, D., (2012). Fall speed measurement and high-speed resolution multi-angle photography of hydrometeors in free fall, *Atmos. Meas. Tech.*, 5, 2625-2633, doi:10.5194/amt-5-2625-2012.
17. Kleinkort, C., G.-J. Huang, E. Chobanyan, A. Manic, M. Ilic, A. Pezeshki, V. N. Bringi, and B. Notaros, (2015a). Visual Hull Method Based Shape Reconstruction of Snowflakes from MASC Photographs. *Proceedings of the 2015 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting*, July 19-25, 2015, Vancouver, BC, Canada.
18. Kleinkort, C., G.-J. Huang, S. Manic, A. Manic, P. Kennedy, J. Hubbert, A. Newman, V. N. Bringi, and B. Notaros, (2015b). 3D Shape Reconstruction of Snowflakes from Multiple Images, Meshing, Dielectric Constant Estimation, Scattering Analysis, and Validation by Radar Measurements. *American Meteorological Society's 37th Conference on Radar Meteorology*, 14-18 September, 2015, Norman, OK.
19. Korolev, A. and Sussman, B, (2000). A technique for habit classification of cloud particles, *J. Atmos. Ocean. Tech.*, 17, 1048-1057.

20. Grazioli, J., Tuia, D., Monhart, S., Schneebeli, M., Raupach, T., and Berne, A., (2014). Hydrometeor classification from two-dimensional video disdrometeor data, *Atmos. Meas. Tech.*, 7, 2869-2882, doi:10.5194/amt-7-2869-2014.
21. Magono, C. and Lee, C.W.(1966). Meteorological classification of natural snow crystals, *J. Fac. Sci., Hokkaido Univ., Series VII*, 2, 321-335.
22. Mosimann, L., Weingartner, E., and Waldvogel, A. (1994). An analysis of accreted drop sizes and mass on rimed snow crystals, *J. Atmos. Sci*, 51, 1548-1558.
23. Azimi, M. Lecture. ECE 656- Machine Learning and Adaptive Systems. Colorado State University, Fort Collins. Feb. 2nd – April 30th 2018.
24. Murphy, K. P., (2012). Machine Learning, A Probabilistic Perspective, *The MIT Press*
25. Haykin, S., (2009). Neural Networks and Learning Machines Third Edition, *Pearson Education Inc.*
26. Svozil, D., Kvasnička, V., Pospíchal, J., (1997). Introduction to multi-layer feed-forward networks, *Chemometrics and Intelligent Laboratory Systems* 39 (1997) 43-62
27. Collobert, R., Weston, J., (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning, *Proceedings of the 25th international conference on Machine learning*, 160-167 doi:10.1145/1390156.1390177
28. O’Shea, T. J., Corgan, J., and Clancy, T. C., (2016). Convolutional Radio Modulation Recognition Networks. In: Jayne C., Iliadis L. (eds) *Engineering Applications of Neural Networks. EANN 2016. Communications in Computer and Information Science*, vol 629. Springer, Cham. doi: 10.1007/978-3-319-44188-7_16.
29. Goodfellow, I., Bengio, Y., and Courville, A., (2016). Deep Learning, *The MIT Press*
30. Torrey, L., Shavlik, J., (2009). Transfer Learning, *Handbook of Research on Machine Learning Applications*, by IGI Global.
31. Romanuke, V., (2017). Appropriate Number of Standard 2x2 Max Pooling Layers and Their Allocation in Convolutional Neural Networks for Diverse and Heterogeneous Datasets, *Info. Tech. and Mgmt. Sci.*, Sciendo, vol. 20(1), pages 12-19, December.
32. Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
33. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., (2015). Going deeper with convolutions. In *CVPR*.
34. Zeiler, M.D. and Fergus, R., (2014). Visualizing and understanding convolutional neural networks. In *ECCV*.

35. He, K., Zhang, X., Ren, S., Sun, J., (2016). Deep Residual Learning for Image Recognition, In *CVPR*.
36. Krizhevsky, A., Sutskever, I., and Hinton, G. E., (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp 1097-1105.
37. Mathworks, (2018), Pretrained Convolutional Neural Networks. August 2018.
https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html#mw_45a8c0b2-26fa-48e9-905a-a7ed7b87bfc8
38. Russakovsky, O.,*, Deng, J.,*, Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., and Fei-Fei, L. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.
39. del-Rio, F., Messina, P., Dominguez, V., & Parra, D. (2018). Do Better ImageNet Models Transfer Better... for Image Recommendation? *CoRR*, *abs/1807.09870*.
40. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *JMLR*. 15 1929-1958
41. MATLAB and Machine Learning and Deep Learning Toolbox Release 2018b, The MathWorks, Inc., Natick, Massachusetts, United States
https://www.mathworks.com/solutions/deep-learning.html?s_tid=hp_brand_deeplearning