Convolutional Neural Networks for Hydrometeor Classification using Dual Polarization Doppler Radars

Yuping Lu
University of Tennessee
Knoxville, TN, USA
yupinglu89@gmail.com

Jitendra Kumar
Oak Ridge National Laboratory
Oak Ridge, TN, USA
jkumar@climatemodeling.org

***Abstract*—Traditional fuzzy logic hydrometeor classification algorithm is a common way to classify precipitation type from dual polarization doppler radar. We propose a deep learning-based method to estimate hydrometeors efficiently using observed radar variables such as horizontal reflectivity (Z_H), differential reflectivity (Z_{DR}), correlation coefficient (ρ_{HV}) and specific differential phase (K_{DP}) from National Weather Service NEXRAD collected at Vance AFB facility at the first elevation angle from January 1st, 2015 to July 31th, 2019. We stack matrices of values from these four polarimetric variables as one 3D array. Samples are preprocessed and divided into training, validation and test set with four target hydrometeor categories (Ice Crystals (IC), Dry Snow (DS), Light and/or Moderate Rain (RA) and Big Drops (rain) (BD)). We developed and optimized five Convolutional Neural Networks (CNNs) architectures and achieved an accuracy of 87.23% and 93.736% respectively using modified *ResNet* with two different input data selection strategies for hydrometeor classification. Training data selection strategies were important to ensure use of available samples in training for robust performance evaluated by applying the models to novel time period beyond what was use to train the model. Seasonal variation in atmospheric conditions lead to seasonal patterns of liquid vs solid forms of precipitation, that poses challenge for classifier and offer insights into domain specific approaches required for problem of hydrometeor identification. Computationally efficient and scalable approach for classification of hydrometeors offer opportunities to effectively use the large volumes of rich time series of radar observations that are becoming increasingly available.**

***Index Terms*—Convolutional neural network, Hydrometeor classification, Dual polarization doppler radar, Atmospheric science**

I. Introduction

Hydrometeor classification for dual polarization doppler radar is the process to identify the precipitation type based on the scattering properties of precipitation particles. Scattering properties of hydrometeors of different shape, size and orientation can be used to characterize and identify them. The classification of weather radar echoes for dual polarization doppler radar often consists of two steps [1]. The first step is to differentiate between meteorological and non-meteorological echoes using dual-polarization capability of doppler radars. The second step is to classify these identified meteorological echoes by combining the collected polarimetric variables (i.e., horizontal reflectivity (Z_H), differential reflectivity (Z_{DR}), correlation coefficient (ρ_{HV}), specific differential phase (K_{DP}) etc.), which have different properties due to the difference in shape, size, orientation, phase state, and fall orientation of each type of hydrometeor [2], [3].

Commonly used method for hydrometeor classification is the fuzzy logic algorithm, which defines a non linear relationship between polarimetric variables and a specific type of hydrometeor with the help of membership functions [1], [4]. The idea was first explored in [5] and [6], and have since become the dominating hydrometeor classification algorithm with different variations and refinements. [3] included measurement conditions and three-dimensional temperature information to classify hydrometeors. [7] proposed a semi-supervised method for dual-polarization radars. However, fuzzy logic method is susceptible to noise in input data, and being a bin based method is unaware of information from neighboring radar cells [7].

Deep learning methods like convolutional neural networks (CNNs) have proven to be effective in image recognition and classification. They have received wide adoption and attention especially after the work by [8] at the 2012 *ImageNet* Large-Scale Visual Recognition Challenge (ILSVRC-2012) with the help from improvements in computer hardware. *VGG* network increased the depth of CNNs, which was up to 11 layers more than *AlexNet* [9]. *ResNet* addressed the degradation problem and went even deeper with the number of layers up to 152 [10]. *DenseNet* improved feature propagation and feature reuse by connecting each layer with every other layer [11].

CNNs have been applied to automated detection of bird roosts using dual polarization doppler radar data [12]. They are also been used to classify hydrometeors by incorporating into the fuzzy logic algorithm [13]. In this paper, we use only CNNs to classify precipitation types without fuzzy logic. The general idea is to treat dual polarization radar data as multi-dimensional images to do the training and testing. However, unlike traditional images with three channels (Red, Green and Blue), radar data has multiple channels and each channel stores the data from a specific polarimetric variable. Our results show that CNNs can provide an robust and accurate classification

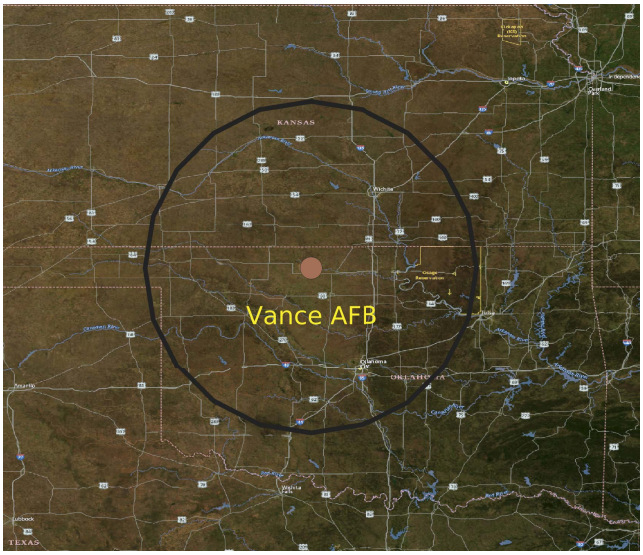


Fig. 1. Location and measurement extent of NEXRAD radar station at Vance Air Force Base, Oklahoma, USA

of hydrometeors based on dual polarization doppler radars.

II. Datasets and Processing

Our study focused on National Weather Service (NWS) at Vance Air Force Base, Oklahoma, USA (Figure 1). We used NEXRAD Level-III products from weather surveillance radar (1988 doppler WSR-88D S-band dual-polarization radar) [14]. NEXRAD radar performs scans at multiple elevation angles, however, we chose the observations at the first elevation angle (0.5 degree) in plan positional indicator (PPI) mode for the period January 1st, 2015 to July 31th, 2019.

NEXRAD dataset were downloaded from National Oceanic and Atmospheric Administration's (NOAA) National Centers for Environmental Information (NCEI). Raw data volume for our period of study was about 500GB. We extracted data from four polarimetric variables measured at the first elevation angle, which are horizontal reflectivity (Z_H), differential reflectivity (Z_{DR}), correlation coefficient (ρ_{HV}) and specific differential phase (K_{DP}), from the raw datasets for further processing. We also extracted variable $radar_echo_classification$ as data labels or training targets. All extracted polarimetric variables were stored in the form of a 360×1200 matrix. Coarse resolution Z_H available as 360×230 matrix was processed to match the dimension of other variables. A time series of all radar scans, with four polarimetric variables for each scan, were assembled as 3D array for analysis.

The 3D radar data array was processed in small chunks of 4×30×30 for model training. Similarly, we processed $radar_echo_classification$ and selected the dominant hydrometeor category as the label for each chunk and attach that label to the corresponding sample. Due to sparse nature of radar that depend on presence or absence of a meteorological echoes, a chunk can have missing or invalid values. To ensure we have sufficient signal for classification, we applied a filter and selected chunks with at least 45 (out of 30×=900) pixels with valid values, and the dominant hydrometeor category to be occupying at least 22 pixels to avoid excessive noise in the input data. If these conditions are not met, the sample was not included in the analysis. Four categories (Ice Crystals (IC), Dry Snow (DS), Light and/or Moderate Rain (RA) and Big Drops (rain) (BD)) are used in this study as they are the most common forms of precipitation in the region. The samples with labels other than these four categories, which were few, were excluded. This was done to reduce the issue of class imbalance while training CNNs. Python ARM Radar Toolkit (*Py-ART*) [15] was used for processing raw radar datasets.

Data from January 1st, 2015 to October 1st, 2018 are used for training, validation and testing of CNN models. We employ two strategies to select these samples (Figure 2). For the first strategy (*S1*), we carefully select the samples so that each category has the same amount of samples. The total sample size is 460,000 with 400,000 samples in training set, 30,000 samples in validation set and 30,000 samples in test set. The mean and standard deviation of each channel [Z_H, Z_{DR}, ρ_{HV}, K_{DP}] of the training set are [0.7324, 0.0816, 4.29, 0.7663] and [0.1975, 0.4383, 13.1661, 2.118] respectively. We then normalize the three sets with the above means and standard deviations. For the second strategy (*S2*), we use all the samples and divide them into 2,063,867 training set, 257,984 validation set and 257,998 test set. The mean and standard deviation of each channel of this training set are [0.7518, 0.0341, 11.1675, 1.2187] and [0.1988, 0.3581, 11.8194, 2.1971] respectively. Again, we normalize the three sets with the calculated means and standard deviations. Notably, using all available samples makes a key difference (in terms of data space mean and standard deviation) for RA and BD categories. We use the rest of sample from November 1st, 2018 to July 31th, 2019 to do further tests on the trained models for their applicability in operational mode on the "novel time period" representing future never seen before data.

III. Methodology

CNNs were first proposed 30 years ago by [16] but has gained increased popularity across the academia and industry with availability of GPU-based computing and software frameworks for neural network. At the beginning, CNNs had only several layers, for example, the *LeNet-5* only has 5 layers [17]. CNNs have become deeper and deeper since the introduction of *AlexNet* in 2012. In this paper, we try to find the nonlinear relationship between the four chosen polarimetric variables and hydrometeor types. Selected CNN models, which are *AlexNet*, *VGG*, *ResNet* and *DenseNet*, have layers ranging from 8 to 121. We modify these models to fit dual polarization doppler radar data.

A. Architectures

CNNs usually consist of several layers of convolution(Conv), one or more fully connected layers, and other operations such as rectified linear units (ReLU) [18], batch

TABLE I

CNN ARCHITECTURE FOR NEXRAD HYDROMETEOR CLASSIFICATION, NUMBER OF PARAMETERS (IN MILLIONS) AND COMPUTATIONAL PERFORMANCE (GFLOPS) OF EACH MODEL. ONLY ONE FULLY CONNECTED LAYER USED IN VGG 19-LAYER MODEL WITH BATCH NORMALIZATION, SO VGG IN THIS PAPER ONLY HAS 17 LAYERS.

layer name	output size	8-layer
conv1	16×16	4×4, 64, stride 2
pooling	8×8	2×2 max pool, stride 2
conv2	8×8	5×5, 192
pooling	4×4	2×2 max pool, stride 2
conv3	4×4	3×3, 384
conv4	4×4	3×3, 256
conv5	4×4	3×3, 256
pooling	2×2	2×2 max pool, stride 2
	1×1	256-d fc, 256-d fc, 4-d fc, softmax
Params(M)		2.78
GFLOPS		0.06

(a) *AlexNet*

layer name	output size	17-layer
conv1_x	30×30	$\begin{bmatrix} 3 \times 3, 16 \end{bmatrix} \times 2$
pooling	16×16	2×2 max pool, stride 2
conv2_x	16×16	$\begin{bmatrix} 3 \times 3, 32 \end{bmatrix} \times 2$
pooling	8×8	2×2 max pool, stride 2
conv3_x	8×8	$\begin{bmatrix} 3 \times 3, 64 \end{bmatrix} \times 4$
pooling	4×4	2×2 max pool, stride 2
conv4_x	4×4	$\begin{bmatrix} 3 \times 3, 128 \end{bmatrix} \times 4$
pooling	2×2	2×2 max pool, stride 2
conv5_x	2×2	$\begin{bmatrix} 3 \times 3, 128 \end{bmatrix} \times 4$
pooling	1×1	2×2 max pool, stride 2
	1×1	4-d fc, softmax
Params(M)		1.26
GFLOPS		0.03

(b) *VGG*

layer name	output size	18-layer	101-layer
conv1	30×30	3×3, 16, stride 1	
	30×30	3×3 max pool, stride 1	
conv2_x	30×30	$\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 16 \\ 3 \times 3, 16 \\ 1 \times 1, 64 \end{bmatrix} \times 3$
conv3_x	15×15	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 128 \end{bmatrix} \times 4$
conv4_x	8×8	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 23$
conv5_x	4×4	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 3$
	1×1	average pool, 4-d fc, softmax	
Params(M)		0.7	2.68
GFLOPS		0.03	0.16

(c) *ResNet*

layer name	output size	121-layer
convolution	30×30	3×3 conv, stride 1
pooling	16×16	2×2 max pool, stride 2
dense block (1)	16×16	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
transition layer (1)	16×16	1×1 conv
	8×8	2×2 average pool, stride 2
dense block (2)	8×8	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
transition layer (2)	8×8	1×1 conv
	4×4	2×2 average pool, stride 2
dense block (3)	4×4	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
transition layer (3)	4×4	1×1 conv
	2×2	2×2 average pool, stride 2
dense block (4)	2×2	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$
	1×1	average pool, 4-d fc, softmax
Params(M)		6.95
GFLOPS		0.23

(d) *DenseNet*

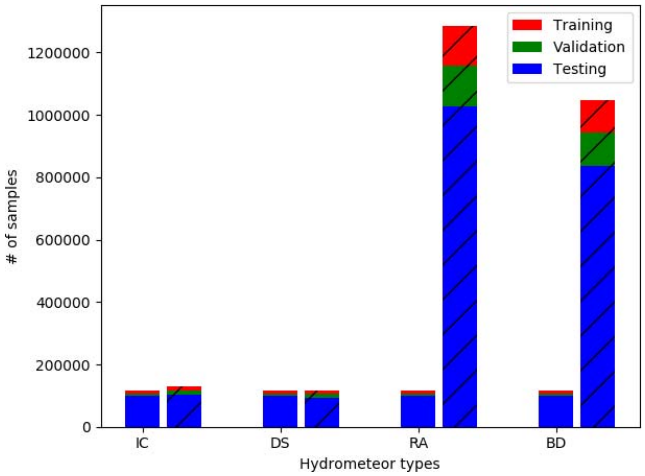


Fig. 2. Sample sizes available for training, validation and testing of CNN models under two input data strategy (*S1*: left bar; *S2*: right hatched bar) and their distribution across four target hydrometeor types.

normalization (BN) [19] and pooling [17]. The input data, or radar samples in this study, go through the whole network and get a score for each category using the *softmax* function. The category with the biggest score is the final classification result. We maintain the main structure of the selected models with the same number of layers, except *VGG-19_BN* with only 17 layers. Also, our sample size is much smaller $4 \times 30 \times 30$, compared to commonly used $3 \times 224 \times 224$ for image classification CNN models, so we change the kernel sizes and parameters of BN, ReLU, Conv and Pooling in each model.

Take *AlexNet* as an example, the $4 \times 30 \times 30$ radar sample first passes through a Conv layer with kernel size 4×4, stride 2, and padding 2, thus the feature map size is $64 \times 16 \times 16$. ReLU and 2×2 Pooling with stride 2 filters are then applied to reduce the size of feature map to $64 \times 8 \times 8$. The feature map then goes through a Conv layer, ReLU and Pooling again, but this time the Conv layer kernel size is 5×5. So the generated feature map becomes $192 \times 4 \times 4$. It then passes through another Conv layer with kernel size 3×3, padding 1 and a ReLU filter, and

repeat the process for two more times with different number of output channels. At this stage, the size of feature map is 256×4×4. A final Pooling filter with kernel size 2×2, stride 2 is applied, followed by a Dropout function. Finally, three fully connected layers are applied. Here the first fully connected layer is combined with ReLU and Dropout, the second fully connected layer is only followed by a ReLU filter (Table I(a)).

Similarly, more details about other modified CNN models we developed and applied can be found in Table I. Our implementation of the *VGG-19_BN* only has 17 layers after removing two fully connected layers as our radar samples are much smaller than images. *ResNet-18* has the smallest number of parameters while *DenseNet-121* has the largest number of parameters. *VGG-19_BN* and *ResNet-18* have the similar computational performance (in GFLOPS), which is lower than other models while *DenseNet-121* has has the best computational performance in terms of GFLOPS.

B. Implementation

All CNN models in this paper are implemented in Python using *PyTorch* [20] framework. We also implement our own dataset class for radar data for data loading and processing using *Py-ART*. The preprocessing scripts and source code are available on github. https://github.com/YupingLu/Radar

IV. Results and discussions

NEXRAD raw datasets were preprocessed on Atmospheric Radiation Measurement (ARM) *Stratus* cluster at Oak Ridge National Laboratory (ORNL) [21], a Cray CS400 with Intel Xeon E5-2698 v3 and 256 GB of RAM per node. Training and testing of the CNN models were performed on NVIDIA DGX station which is a workstation designed for deep learning with four Tesla V100 GPUs and capable of 500 TFLOPS peak performance [22].

A. Training and Validation

All CNN models were trained using stochastic gradient descent (SGD) method. We set the batch size = 256, epochs = 600, learning rate = 0.1, Nesterov momentum = 0.9 [23], and weight decay = 1e-3 for the training of all neural networks. Only for *DenseNet-121*, we set weight-decay to 2e-3 to avoid overfitting. Weight initialization was also performed before the training [24]. We also employ dynamic learning rate once learning stagnate reducing it by a factor of 0.5. Our experiments show that using dynamic learning rate can help reach a high accuracy rate faster with fewer epochs compared to a fixed learning rate or by manually reducing learning rate at specific epochs. Among all selected models, *ResNet* reached the highest accuracy using the samples from the first selection strategy (*S1*). Figure 3 shows the training and validation accuracy and loss of *ResNet-18* during the 600 epochs. Validation accuracy and loss fluctuate heavily at the beginning of iterations, however, the convergence curves smoothen out after 100 epochs. Figure 4 shows the convergence behavior (training accuracy and loss) of all five CNN models based on inputs from strategy *S1*. The summary of accuracy and loss on NEXRAD validation set based on input strategy *S1* using best trained models are shown in Table II. All CNN models performed well to classify hydrometeor for validation set based on input strategy *S1*, while *ResNet-18* performed the best.

TABLE II
Accuracy and loss on NEXRAD validation and testing set based on input strategy *S1* using best trained CNN models

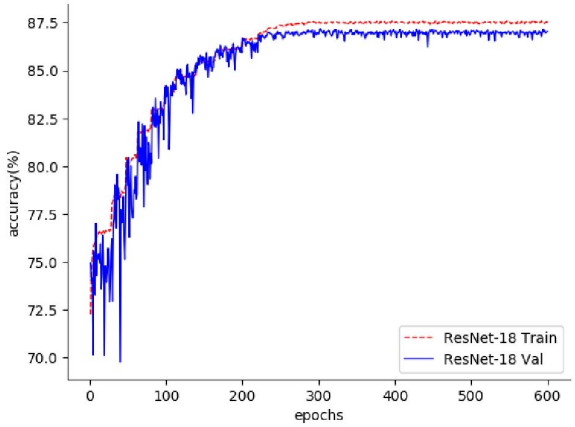
Model	Validation		Testing	
	Accuracy	Loss	Accuracy	Loss
AlexNet	85.617%	0.342	85.880%	0.337
VGG_19-BN	86.520%	0.322	86.323%	0.325
ResNet-18	87.177%	0.305	87.230%	0.306
ResNet-101	87.043%	0.311	87.190%	0.310
DenseNet-121	85.457%	0.353	85.287%	0.353

Best trained model for each of the five CNNs using input data based on strategy *S1* were applied to NEXRAD test data set that were not used during training or validation of the models. In a pattern similar to training and validation set, *ResNet-18* achieved the highest accuracy of 87.23%, while *DenseNet-121* had the lowest accuracy of 85.29% (Table II).

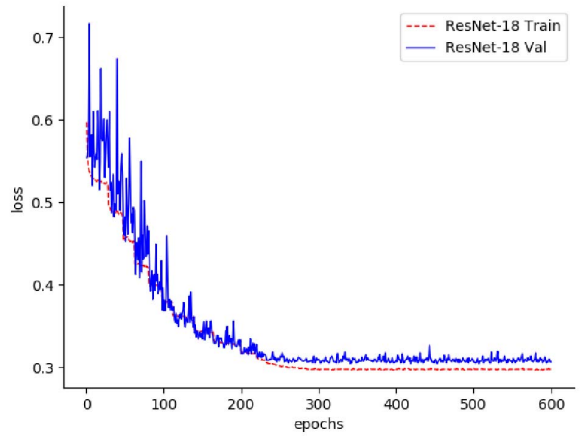
ResNet-18 was also applied to the data from novel time period, however, a significant performance degradation was observed (Figure 6(a)). This degradation in performance of the model can be attributed to input data strategy (*S1*) where in effort to have balanced sample across all hydrometeor types. At our study site, liquid precipitation forms (RA and BD) are experienced more frequently all through the year and by using only a subset, models were not able to learn all storm conditions experienced at the site. To address this issue we implemented input data strategy *S2* (where we used all available samples) to retrain best performing *ResNet-18* and *ResNet-101* models. Hyper-parameters used during previous training (using *S1* strategy) were kept the same during this retraining. Newly trained *ResNet-18* and *ResNet-101* both exhibited improved accuracy at 93.479% and 93.736%. Best performing *ResNet-101* trained using *S2* input data strategy was applied to the novel time period and demonstrated high accuracy (Figure 6(b)). Figure 5 shows the convergence pattern for *ResNet-101* training. With a larger training dataset (Figure 2), a faster convergence was achieved in fewer number of iterations.

B. Overfitting

While neural networks developed in current study have millions of parameters and only four hydrometeor categories to classify, it is easy for the neural networks to overfit to the training data and lack generalization to be applicable to future or novel time periods. Thus we implemented a number of methods during the training process to avoid overfitting. We employed a large time series of data and thus a fairly large and distributed samples to the training step. It helped reduce the gap between training loss and test loss, however, some overfitting issues still persisted. While our initial attempt for training was to use a chunk size of 4×60×60, we iteratively reduced the size to 4×30×30 used for all reported results. This allowed improved resolution to detect small scale features

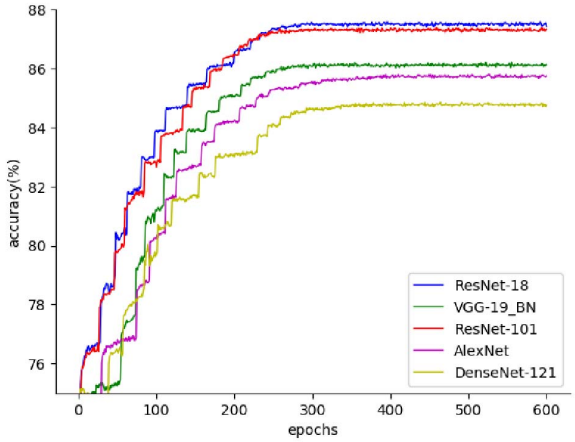


(a) Improvement in accuracy of *ResNet-18* over training epochs.

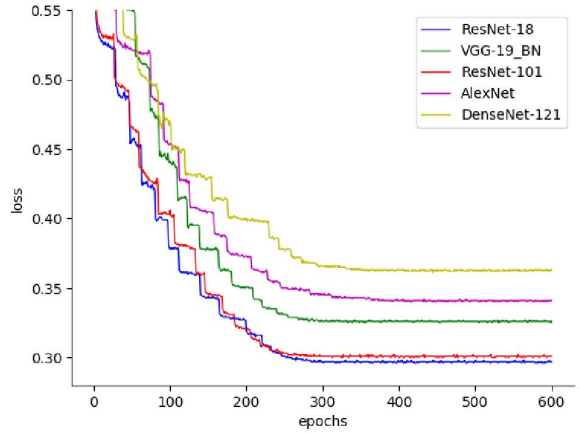


(b) Reduction in Cross-Entropy loss term over training epochs.

Fig. 3. Training and validation accuracy and Cross-Entropy loss of *ResNet-18* based on input strategy *S1*.



(a) Improvement in accuracy of all five CNNs over training epochs.



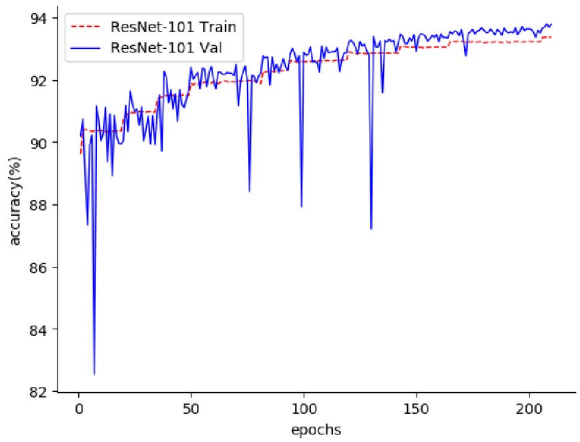
(b) Reduction in Cross-Entropy loss term for all five CNNs over training epochs.

Fig. 4. Training accuracy and Cross-Entropy loss of all five models based on input strategy *S1*.

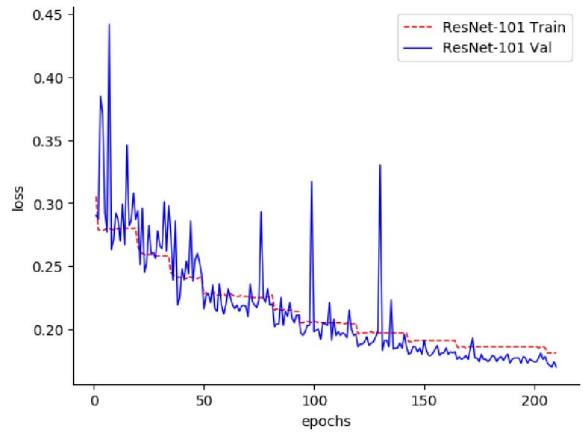
(weather fronts) and also increased the sample size. We also applied data augmentation strategies to solve the overfitting problem. For example, we increase each training sample to $4\times44\times44$ by padding mean value of the original sample. Then we randomly crop $4\times30\times30$ samples from the padded data for training. Using only part of the information of the radar sample helps the models to generalize and make right predictions even under noisy conditions and improves generalization. Similarly, we also horizontally and vertically flip the cropped sample randomly with a 0.5 probability. We also performed hyper-parameter tuning during training. Through a number of schemes we were able to overcome the issue of overfitting in the reported results.

C. Discussion

CNN models developed using dual-polarization doppler radar were able to classify the hydrometeors with very high accuracy. Using four polarimetric variables from NEXRAD radar (Figure 7(a.b.d.e), these models were trained to classify hydrometeors (Figure 7(f)) while using NEXRAD's hydrometeor classification as training label (Figure 7(c)). For purpose of illustration, data collected at 10:40AM on January 7, 2018 was used for Figure 7). Central plains of Oklahoma where out study site is located experiences a wide range of atmospheric conditions and storm systems across seasons making the hydrometeor detection a challenging problem. Model training using two input data selection strategies suggest that large sample size covering wide range of atmospheric conditions to train the model for robust performance when they are applied to future/novel time period beyond the period they were trained on. *ResNet-101* model demonstrated robustness and high accuracy on novel time period of 9 months (Figure 6(b)). Accuracy of the model however varies through the seasons,

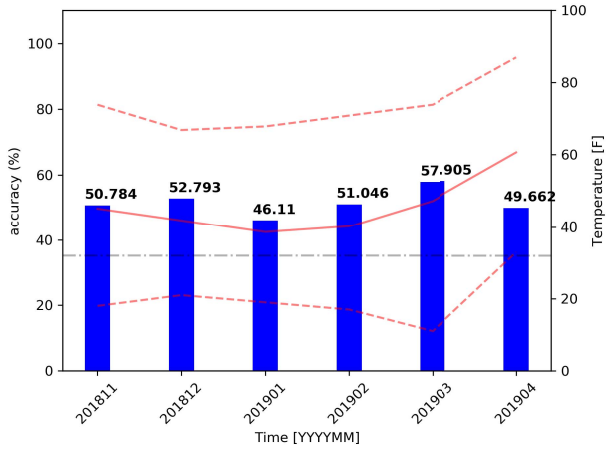


(a) Improvement in accuracy of *ResNet-101* over training epochs.

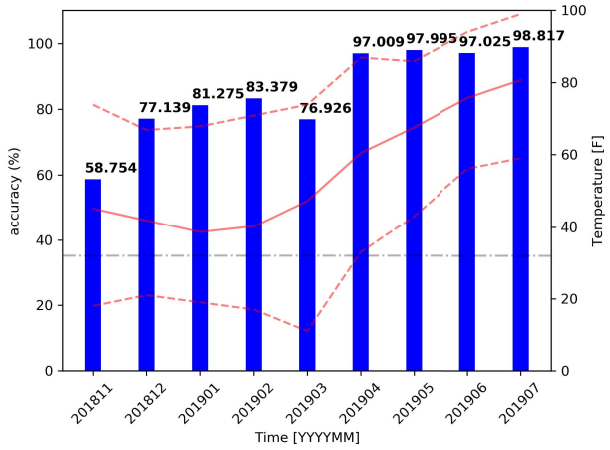


(b) Reduction in Cross-Entropy loss term for *ResNet-101* over training epochs.

Fig. 5. Training accuracy and Cross-Entropy loss of *ResNet-101* using training data selection strategy *S2*.



(a) *ResNet-18* (training data selection strategy *S1*) accuracy during novel time period



(b) *ResNet-101* (training data selection strategy *S2*) accuracy during novel time period

Fig. 6. The accuracy of best CNN models trained using two input data strategy when applied to novel time period. Average monthly (solid red line) and minimum/maximum range of air temperature at the site show strong correspondence to the variability in model accuracy. Dotted horizontal line denotes the freezing temperature.

with high accuracy during the warm season when precipitation is predominantly in liquid form (RA and BD), while the performance is slightly lower during cold months when precipitation occurs in form of ice and snow (IC and DS). During the novel time period of November 2018 – July 2019, IC and DS are dominant form of precipitation during winter months (November – March), while RA ad BD dominates spring and summer months (April – July) (Table III). Accuracy of *ResNet-101* during these months and across different hydrometeor types (Table IV) highlights the need for further improvement of models for IC and DS categories under-represented in the training set.

Among various CNN models explored in our study, *ResNet* has the best performance because it adds shortcut connections to skip one or more layers to help transfer information to deeper layers. *AlexNet* and *VGG_19-BN* also performed well with slightly lower performance compared to *ResNet*. Among *AlexNet* and *VGG_19-BN*, VGG_19-BN performs better with fewer parameters and deeper layers. *DenseNet-121* while has more direct connections between layers and also has the deepest layers, performs the worst due to its complexity and large numbers of parameters. These observation are specific to variants of these network modified in our study for classification of NEXRAD radar observations. Our experiments with training data selection strategy highlights the need for further improvement of the model to address data imbalance issue across hydrometeor categories, and further understanding of

TABLE III

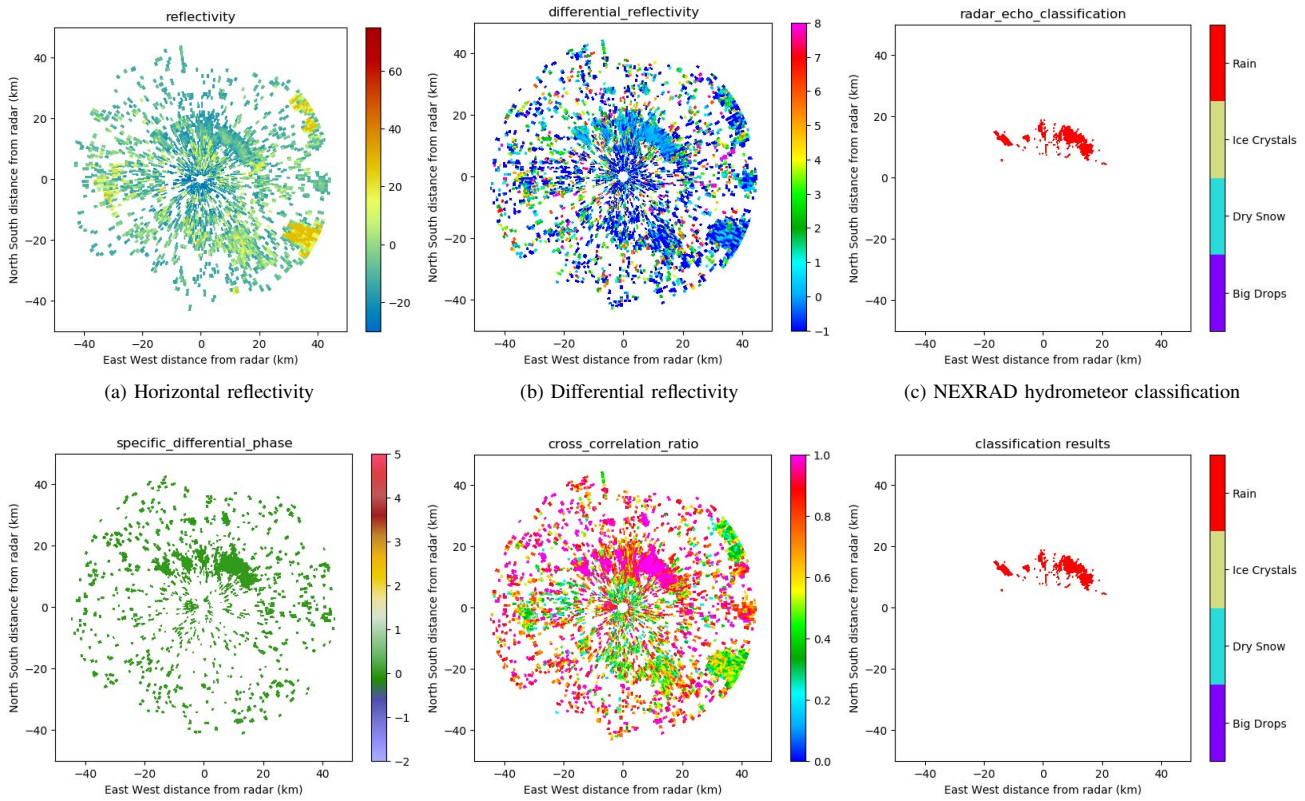
THE NUMBER OF SAMPLES ACROSS DIFFERENT HYDROMETEOR CATEGORIES DURING NOVEL TIME PERIOD OF NOVEMBER 2018 – JULY 2019.

Category	201811	201812	201901	201902	201903	201904	201905	201906	201907
All Categories	72204	50370	66718	72663	60302	77843	111265	106101	47373
Ice Crystals (IC)	25626	12777	7391	12227	13085	74	0	0	0
Dry Snow (DS)	10942	4144	7596	6302	4595	642	4	0	0
Light and/or Moderate Rain (RA)	16965	28043	48363	51169	28736	31467	29584	29573	4890
Big Drops (rain) (BD)	18671	5406	3368	2965	13886	45660	81677	76528	44483

TABLE IV

THE ACCURACY OF EACH CATEGORY OF FUTURE MONTH NEXRAD DATA USING *ResNet-101*

Category	201811	201812	201901	201902	201903	201904	201905	201906	201907
All Categories	58.754%	77.139%	81.275%	83.379%	76.926%	97.009%	97.995%	97.025%	98.817%
Ice Crystals (IC)	25.708%	44.079%	36.450%	40.255%	44.043%	54.054%			
Dry Snow (DS)	36.191%	56.347%	18.984%	46.906%	49.489%	52.181%			
Light and/or Moderate Rain (RA)	96.186%	99.497%	98.863%	99.572%	97.181%	98.567%	97.556%	96.132%	94.806%
Big Drops (rain) (BD)	83.322%	55.235%	67.577%	59.292%	75.076%	96.636%	98.159%	97.371%	99.258%



(a) Horizontal reflectivity (b) Differential reflectivity (c) NEXRAD hydrometeor classification

(d) Specific differential phase (e) Cross correlation ratio (f) *ResNet-101* based hydrometeor classification

Fig. 7. Four polarimetric variables from NEXRAD S-band radar observations collected at 10:40AM on January 7, 2018 were classified to identify hydrometeors using *ResNet-101* using training data selection strategy *S2*.

the known seasonal patterns when hydrometeors in liquid vs solid forms are dominant.

V. CONCLUSIONS

We developed and applied five commonly used CNN models with our modifications to classify hydrometeors using NEXRAD dual-polarization doppler radar. Our results indicate that CNNs are an efficient tool for radar based classification of hydrometeors. Modified *ResNet-101* model achieved test accuracy of 93.736% and performed well for classification of hydrometeors (Table IV; Figure 6(b)) when applied to nine months of future/novel time period beyond its training period. With efficient implementation and scalability on GPU-based computational platforms, it offers opportunities to build effec-

tive hydrometeor classification models from growing number of weather radar system utilizing the voluminous time series datasets.

Our future efforts would focus on addressing seasonal variability in model performance by implementation improved schemes for training using class imbalanced data [25]. Availability of labeled data is a key bottleneck for application of supervised methods like CNNs to radar datasets. While NEXRAD data used in this study had labeled information available for training, that is often not the case at other radar stations operating similar S–band radar, or at Atmospheric Radiation Measurement program sites operating C–, K– and X–band radars. Transfer learning approaches in conjunction with CNNs can provide opportunities to train the models at locations where labeled datasets are available and then transfer, adapt, and retrain for their application at sites where little to none labeled datasets may be available.

ACKNOWLEDGMENT

This research was supported by the Atmospheric Radiation Measurement (ARM) user facility, a U.S. Department of Energy (DOE) Office of Science user facility managed by the Office of Biological and Environmental Research. This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid- up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

REFERENCES

[1] H. S. Park, A. Ryzhkov, D. Zrnić, and K.-E. Kim, "The hydrometeor classification algorithm for the polarimetric WSR-88D: Description and application to an MCS," *Weather and Forecasting*, vol. 24, no. 3, pp. 730–748, 2009.

[2] J. M. Straka, D. S. Zrnić, and A. V. Ryzhkov, "Bulk hydrometeor classification and quantification using polarimetric radar data: Synthesis of relations," *Journal of Applied Meteorology*, vol. 39, no. 8, pp. 1341–1372, 2000.

[3] H. Al-Sakka, A.-A. Boumahmoud, B. Fradon, S. J. Frasier, and P. Tabary, "A new fuzzy logic hydrometeor classification scheme applied to the French X-, C-, and S-band polarimetric radars," *Journal of Applied Meteorology and Climatology*, vol. 52, no. 10, pp. 2328–2344, 2013.

[4] B. Dolan and S. A. Rutledge, "A theory-based hydrometeor identification algorithm for x-band polarimetric radars," *Journal of Atmospheric and Oceanic Technology*, vol. 26, no. 10, pp. 2071–2088, 2009. [Online]. Available: https://doi.org/10.1175/2009JTECHA1208.1

[5] J. Straka and D. Zrnic, "An algorithm to deduce hydrometeor types and contents from multi-parameter radar data," in *Preprints, 26th Conf. on Radar Meteorology, Norman, OK, Amer. Meteor. Soc*, 1993, pp. 513–515.

[6] J. Straka, "Hydrometeor fields in a supercell storm as deduced from dual-polarization radar," in *Preprints, 18th Conf. on Severe Local Storms, San Francisco, CA, Amer. Meteor. Soc*, vol. 55, 1996, pp. 1–5.

[7] R. Bechini and V. Chandrasekar, "A semisupervised robust hydrometeor classification method for dual-polarization radar applications," *Journal of Atmospheric and Oceanic Technology*, vol. 32, no. 1, pp. 22–47, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[12] C. Chilson, K. Avery, A. McGovern, E. Bridge, D. Sheldon, and J. Kelly, "Automated detection of bird roosts using NEXRAD radar data and Convolutional Neural Networks," *Remote Sensing in Ecology and Conservation*, vol. 5, no. 1, 2018.

[13] H. Wang, Y. Ran, Y. Deng, and X. Wang, "Study on deep-learning-based identification of hydrometeors observed by dual polarization doppler weather radars," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, p. 173, 2017.

[14] W. H. Heiss, D. L. McGrew, and D. Sirmans, "NEXRAD: next generation weather radar (WSR-88D)," *Microwave Journal*, vol. 33, no. 1, pp. 79–89, 1990.

[15] J. J. Helmus and S. M. Collis, "The Python ARM Radar Toolkit (Py-ART), a library for working with weather radar data in the Python programming language," *Journal of Open Research Software*, vol. 4, 2016.

[16] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[17] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[20] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.

[21] "Atmospheric Radiation Measurement (ARM) Next Generation Computing Facility: Stratus," https://adc.arm.gov/tutorials/cluster/stratusclusterquickstart.html, accessed: 2019-08-06.

[22] "NVIDIA DGX Station," https://www.nvidia.com/en-us/data-center/dgx-station/, accessed: 2019-05-06.

[23] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[25] Z. L. Langford, J. Kumar, and F. M. Hoffman, "Wildfire mapping in Interior Alaska using deep neural networks on imbalanced datasets," in *Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW 2018)*, Institute of Electrical and Electronics Engineers (IEEE). Conference Publishing Services (CPS), Nov. 2018.