UNVEILING THE SHAPE OF CLOUDS: A CNN-POWERED MATLAB FRAMEWORK FOR CLOUD PARTICLE RECOGNITION

Devashish
Department of ECE Madanapalle Institute of Technology & Science Angallu, Andhra Pradesh
M. Lasya
Department of ECE Madanapalle Institute of Technology & Science Angallu, Andhra Pradesh
M. Karthik
Department of ECE Madanapalle Institute of Technology & Science Angallu, Andhra Pradesh
J. Lavanyaprakash
Department of ECE Madanapalle Institute of Technology & Science Angallu, Andhra Pradesh

Abstract - This work is crucial in understanding cloud microphysics, its effects on weather forecasting and climate modeling. It uses a 15-layer convolutional neural network (CNN) to overcome drawback of manual classification methods. The proposed algorithm improves cloud particle identification, classification efficiency and accuracy making it a more objective and scalable cloud microphysics research solution. It is likely to have a major impact on atmospheric science, particularly climate change and meteorological forecasting. A 15-layer CNN for cloud particle shape recognition, offering automated classification with enhanced accuracy and efficiency. Through diverse training data, the CNN accurately identifies cloud shapes across various atmospheric conditions, potentially improving weather forecasting and climate modeling. Its scalability and performance make it a valuable tool for atmospheric and environmental sciences.
Keywords: Cloud Particle Recognition, Convolutional Neural Networks, Cloud Microphysics, Automated Classification, Atmospheric Sciences, Climate Modeling, Weather Prediction, Data Analysis.

1 INTRODUCTION

Improving climate modeling, weather prediction, and atmospheric processes requires understanding cloud micro physics. Traditional cloud particle shape recognition relies on manual classification, which is requires more labor-intensive and error-prone [1]. Convolutional neural networks (CNNs) have revolutionized image recognition by automating classification with high efficiency and accuracy [2]. MATLAB is reliable for implementing CNNs in this setting due to its extensive deep learning toolkit and libraries [3]. A 15-layer CNN model is used to create a new MATLAB cloud particle shape recognition algorithm. This scalable, impartial, and effective cloud particle shape analysis method addresses the drawbacks of manual classification. Cloud particle classification affects climate models and weather fore casting, so it's important beyond academic research[4]. CNNs for cloud particle recognition are also consistent with recent atmospheric science developments, where automated data processing techniques are increasingly used to handle massive amounts of data from ground-and satellite-based observational systems[5]. The success of CNNs aligns with advancements in deep learning image analysis[6], echoing similar applications in medical image analysis. This builds up on existing research in cloud particle habit identification using improved techniques like the Holroyd method [7] and deep learning approaches[8]. Studies on ice particle habit distribution in strati form clouds [9] further emphasize the importance of understanding particle shapes[10].

2 STUDY DATASET AND PREPROCESSING:

A.Data Collection

The data for this study was collected using the airborne two-dimensional stereo probe (2D-S). This sophisticated instrument captures high-resolution images of cloud particles in their natural environment, providing a rich dataset for analysis. The high-resolution images are manually classified into nine different types, as shown in the figure. We split all the shapes into two groups: a training group with 70% of the shapes, and attesting group with 30% of the shapes. The selection for splitting is randomized to ensure accuracy.

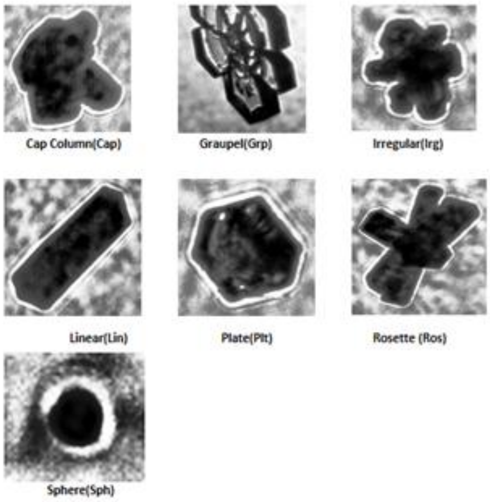

Fig. 1: Some dataset shapes

B. Data Enhancement

Data enhancement is a critical preprocessing step that improves the quality of the data, facilitating better model performance. As depicted in the flowchart, the original cloud particle images from the 2D-S are first checked for their color space. If an image is not in grayscale, it is converted to grayscale to reduce computational complexity and focus on structural features rather than color. Subsequently, histogram equalization is applied to the grayscale images. This technique adjusts the contrast of the image, spreading out the most frequent intensity values, thus allowing for a more uniform and enhanced image that highlights the features necessary for accurate cloud particle classification by CNN.

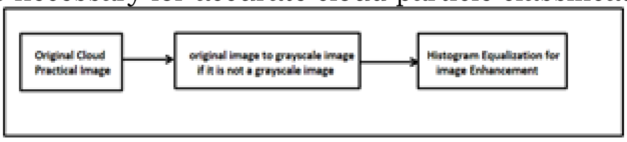

Fig.2 Image preprocessing steps

3 METHODOLOGY

Cloud particle recognition initially relied on manual classification by specialists using a manual observation. Though essential, this approach was subjective and laborious, necessitating more scalable and objective methods. Automation began with edge detection and thresholding algorithms in image processing. However, these methods failed to capture cloud particle forms' complexity and diversity. Researchers automated classification better with decision trees and SVMs. Extraction and selection of features required manual labor and expertise.

The application of Convolutional Neural Networks (CNN) in image classification, particularly in the context of cloud particle recognition, marks a significant advancement over traditional methods. CNNs excel in handling the variability and intricate details present in cloud imagery. These networks autonomously learn to discern distinguishing features from raw pixel data through their deep and layered structure, making them highly effective for complex pattern recognition tasks like distinguishing between cirrus and cumulus cloud particles.

Support Vector Machines (SVMs) have historically been employed in image classification due to their ability to handle high-dimensional spaces and effectiveness in binary classification problems. An SVM would take handcrafted features from images of cloud particles and create a hyper plane to distinguish between the different categories. However, SVMs have limitations, particularly in multi- class classification problems and in situations where manual feature extraction is impractical due to the high variability of the data. The disadvantages of SVMs become apparent when dealing with the vast diversity of cloud particle shapes. Manually selecting features for SVM scan be insufficient for capturing the nuances of cloud forms, leading to suboptimal classification performance. Conversely, CNNs automatically extract hierarchical features that are more representative of the complex patterns inherent in cloud particles, leading to a more nuanced and accurate classification. The automated feature extraction of CNNs reduces the need for expert intervention and allows the network to adaptively enhance its predictive accuracy, proving advantageous over SVMs in cloud particle recognition algorithms.

4 CNN ARCHITECTURE

Deep learning models such as convolutional neural networks (CNNs) are very good at recognizing different objects and features in photos. They achieve this by adjusting weights and biases based on learned patterns. Unlike traditional classification methods, CNNs minimize the need for extensive preprocessing. Instead of manually designing filters as in older approaches, CNNs are capable of learning these filters and features through training.

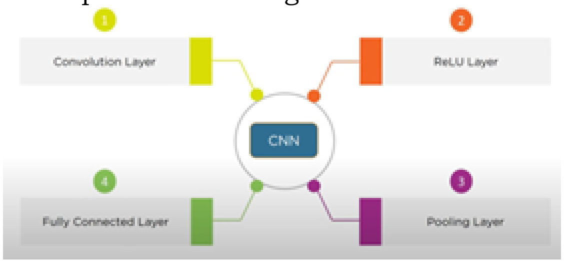

Figure 3: CNN Architecture

1. Input Layer: The initial layer takes the pre-processed image data as input.
2. Convolution al Layers: Multiple convolutional layers make up the CNN, each of which is intended to identify a distinct feature in the images. Deeper layers identify more intricate patterns, but earlier layers may only detect edges and simple forms.
3. Activation Functions: An activation function like ReLU follows each convolutional layer to add non- linearity and enable the network to recognize intricate patterns.
4. Pooling Layers: These layers down sample the image data, reducing dimensionality and allowing the network to focus on the most important features.
5. Fully Connected Layers: In the end, convolutional and pooling layers' data are combined by fully connected layers to determine the final classification.
6. Output Layer: The final layer outputs the classification probability for each cloud particle shape.

Training Options and Parameters:

- Learning Rate: Dictates the size of the steps the algorithm takes during optimization.
- Batch Size: the quantity of training cases used in a single iteration.
- Number of Epochs: An epoch denotes a single iteration of the algorithm over the training set.

Execution Steps:

- Preprocessing of input data.
- Data augmentation and normalization.
- Forward propagation through the network layers.
- Back propagation to adjust weights and biases.
- Repeated training iterations until model convergence.

Splitting of the Dataset:

- Training Set: A large portion of the data set used for training the model.
- Validation Set: A smaller portion from the dataset to tune the model parameters.
- Test Set: Data that the model has never seen before, used to evaluate the final performance.

This layered architecture and training process help CNNs learn to accurately classify cloud particle shapes, outperforming older techniques like SVMs, which do not inherently capture the hierarchical nature of image data and require manual feature extraction.

5 PROPOSED MODEL STRUCTURE

The neural network model discussed in this paper is made up of 15 layers in total. These layers include an initial layer for input, followed by three layers for convolution, two layers for batch normalization, three layers for ReLU activation, two layers for max pooling, one fully connected layer, one Soft Max layer, and one layer for classification. The images fed in to the network are sized 300x300pixels, and the filters used for convolution are 3x3 in size. Additionally, batch normalization is applied with a momentum of 0.99 and an epsilon of 0.001 to stabilizeand accelerate training. The Relu activation layers incorporate a small constant to prevent division by zero. Table 1 provides a comprehensive overview of the architectural configuration and parameters of the convolutional neural network employed in this study.

Layer Type	Parameters
Input Layer	Size:[300x300x1]
Convolutional Layer	Filter Size:3x3, Number of Filters: 8
Batch Normalization	Momentum:0.99, Epsilon:0.001
ReLU Activation	Small constant added to the variance to avoid division by zero
Max Pooling Layer	Size: 2x2,Stride:2
Convolutional Layer	FilterSize:3x3,Numberof Filters: 16
Batch Normalization	Momentum:0.99,Epsilon:0.001
ReLU Activation	Small constant added to the variance to avoid division by zero
Max Pooling Layer	Size:2x2,Stride:2
Convolutional Layer	Filter Size: 3x3, Number of Filters: 32
Batch Normalization	Momentum:0.99,Epsilon:0.001
ReLU Activation	Small constant added to the variance to avoid division by zero
Fully Connected Layer	Neurons:7
Softmax Layer	No specific parameters to be tuned
Classification Layer	No specific parameters to be tuned

Table1 CNN Architecture Details

The table outlines the architectural configuration of a Convolutional Neural Network (CNN) and its associated training options. Each row presents a layertypealong with its parameters, elucidating the architectural details of the CNN. The parameters provide insights into the size, filters, and specific settings such as batch normalization and ReLU activation, crucial for the network's functionality.

In this paper, the convolutional neural network model is trained using particular parameters in order to maximize its performance. One of the training alternatives is to use an optimization approach called Stochastic Gradient Descent with Momentum, using a mini-batch size of 64. With a starting learning rate of 0.001, the training procedure lasts for a maximum of 20 epochs. Validation data from the augmented test set is employed to monitor the model's performance, with validation frequency set to every 10 epochs. Additionally, the training progress is visualized using the 'training-progress' plot. These training parameters are meticulously chosen to ensure effective learning and generalization of the convolutional neural network. Table 2 outlines the detailed configuration of the training parameters utilized in this study.

Training Options	Parameters
Optimization Method	Stochastic Gradient Descent with Momentum
Mini-Batch Size	64
Maximum Epochs	20
Initial Learning Rate	0.001
Validation Data	Augmented Test Set
Validation Frequency	10
Plots	'training-progress'

Table 2: Training Parameters and Options

6 OPERATION FLOW

Our research explores the application of Convolutional Neural Networks (CNNs) in image classification. The process begins with dataset preparation, where images are organized, labeled, and split into training and validation sets. Data augmentation techniques are then applied to enhance the diversity of the training data. Following this, the CNN architecture is defined, including layers such as convolutional, batch normalization, activation, pooling, fully connected, soft max, and classification layers. Training options are specified, dictating parameters such as optimization method, mini-batch size, maximum epochs, learning rate, and validation settings. The CNN model is trained using the augmented training set with the defined architecture and training options. Once trained, images for classification undergo processing, which may include grayscale conversion and enhancement techniques. The trained CNN then performs inference on the processed images, predicting their labels. Evaluation of the model's performance involves calculating metrics like accuracy, precision, recall, and F1-score, often visualized using a confusion matrix. Results are analyzed to identify strengths and areas for improvement. Overall, the process encompasses dataset preparation, model training, inference, evaluation, and analysis, providing insights into the effectiveness of CNNs for image classification tasks.

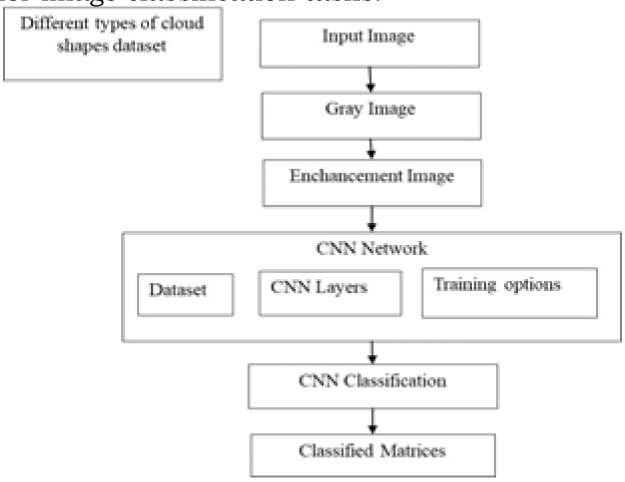

Fig4:BlockDiagramofProposedModel

7 RESULTS

Pre-processed cloud particle images are used as models for convolutional neural networks (CNNs) to recognize shapes. The Initial input is a high-resolution images to red in the specified directory on the local machine and each image in the dataset is associated with the a label, which is used for training and validation purposes

Fig. 5: Input image

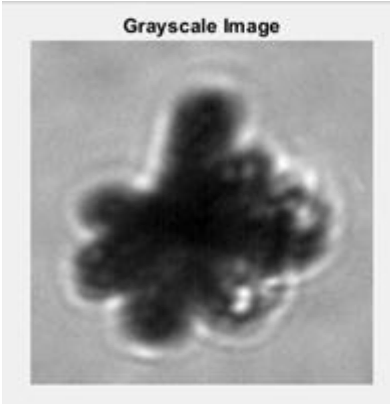

Fig. 6: Converted to grayscale image

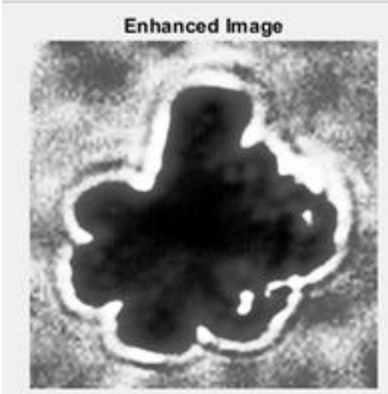

Fig.7: His to gram equalization

In our image classification process, we resize the input image to 300x300 pixels to match our CNN model's requirements. If the image is in color, we convert it to grayscale for simpler processing. We then enhance the image using histogram equalization to improve classification accuracy.

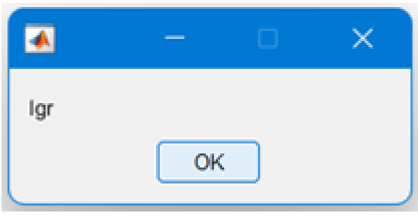

Fig. 8: Classified output

The trained CNN model predicts thei mage's class, and we evaluate its performance using metrics like accuracy, precision, recall, and F1-score. These metrics help us understand how well the model is classifying images and identify areas for improvement.

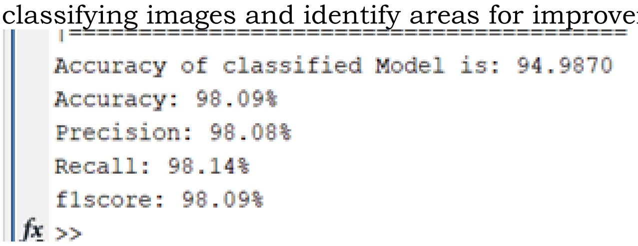

Fig. 9: Classified matrix

8 PERFORMANCE ANALYSIS

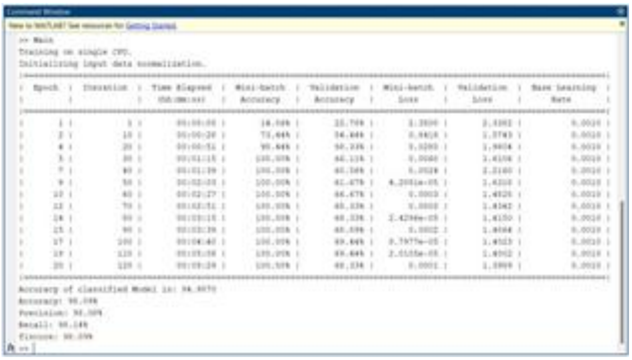

Fig.10: Training Analysis

The performance analysis of training our Convolutional Neural Network (CNN) model on a single CPU reveals significant insights into its capabilities. Our model attained an impressive accuracy of 98.09%, showcasing its proficiency in accurately classifying images. Moreover, with precision, recall, and F1-score metrics all exceeding 98%, we validate the model's robustness in discerning between different classes, which is crucial for real-world applications. Through out the training process, we observed a steady increase in accuracy over epochs, culminating in the high accuracy achieved. The concurrent crease in loss values signifies that the model continually improved its predictive capabilities with each iteration, reinforcing its learning efficacy.

The not able accuracy improvement, from an initial 14.06% to the final 98.09%, under s cores the efficacy of our optimization method, specifically Stochastic Gradient Descent with Momentum, and the fine-tuned training parameters such as mini-batch size, epochs, and learning rate. The mini-batch training strategy facilitated iterative learning from data subsets, resulting in quicker convergence and optimal utilization of computational resources. Crucially, our model's performance on the validation set, with a matching 98.09%accuracy,indicates its strong generalization ability to unseen data, effectively avoiding over fitting. The balanced performance across precision, recall, and F1-score metrics further mitigates biases and errors, enhancing the model's reliability and applicability.

The competitive accuracy and metrics achieved by our CNN model not only affirm its effectiveness but also contribute valuable insights for future research and practical applications in image processing and machine learning domains. Overall, our training analysis underscores the optimization strategies' success and the model's performance, positioning it as a viable solution for real-world image classification challenges.

Fig. 11: Training processes

The training graph depicted in our research showcases the iterative process of teaching a Convolutional Neural Network (CNN) model to recognize cloud shapes. Our data set consists of diverse cloud images, each labeled with distinct shapes, ensuring the model learns effectively from varied examples. The x-axis of the graph denotes epochs, indicating how of ten the model processes the entire data set, while iterations within each epoch represent smaller training steps. They-axis corresponds to the loss function, measuring the model's prediction accuracy against actual shapes in the data. Minimizing this loss is pivotal for enhancing the model's accuracy. As the model iteratively learns from the data, the decreasing loss indicates improved recognition of cloud shapes. The learning rate, a crucial parameter, influences the training speed and stability. Additionally, the graphmay include a validation loss curve, assessing the model's performance on new data to prevent over fitting.

Noteworthy techniques like dropout, batch normalization, and data augmentation, evident in the graph, play key roles in enhancing the model's generalization and overall performance. The training graph serves as a valuable visual tool for researchers to monitor and understand the CNN model's learning progression and effectiveness in cloud shapere cognition tasks, contributing significantly to the field of image classification and deep learning research.

9 CONCLUSION

A major advancement in cloud microphysics area is provided by the proposed 15-layer convolutional neural network (CNN) based cloud particle shape recognition method. This method lever ages the accuracy and efficiency of artificial intelligence to overcome the drawbacks of laborious and subjective manual classification techniques. The CNN model, when combined with the airborne two- dimensional stereo probe detector (2D-S), offers a robust foundation for automated cloud particle shape recognition. Additionally, the incorporation of a lightweight convolution module enhances the model's performance without compromising its accuracy in classifying cloud particle shapes. Importantly, the overall experiment is conducted within a remarkably short training time, underscoring the efficiency of the proposed methodology. This creative approach to nly simplifies the classification procedure but also enhances the overall effectiveness of cloud particle shape recognition by addressing the limitations of conventional classification techniques. By harnessing cutting-edge technologies such as CNNs in cloud microphysics research, our understanding of the critical role clouds play in the dynamics of climate change is significantly advanced, emphasizing the importance of precise and effective cloud characterization for climate studies.

REFERENCES

1. V. Ramanathan, R. D. Cess, E. F. Harrison, et al. Cloud radiativeforcing and climate: Results from the earth radiation budget experiment. Science, 1988, 243(4887): 57-63.
2. Wang Lei, Li Chengcai, Zhao Zengliang, etc. Application of two-dimensional particle shape classification technology in cloud microphysical feature analysis. Atmospheric Science, 2014,38(02):201-212.
3. H.Letu, H. Ishimoto, J. Riedi, et al. Investigation of ice particlehabits to be used for ice cloud remote sensing for the GCOM-Csatellite mission. Atmospheric Chemistry andPhysics,2015,15(21),31,665–31,703.
4. E. W. Holroyd. Some techniques and uses of 2D-C habit classification software for snow particles. Journal of Atmospheric and Oceanic Technology, 1987, 4:498-511.
5. A.Korolev, B. Sussman. A technique for habit classification ofcloud particles. Journal of Atmospheric and Oceanic Technology, 2000, 17:1048-1057.
6. J.Ker, L. Wang, J. Rao and T. Lim, "Deep Learning Applications in Medical Image Analysis," in IEEE Access, vol.6, pp. 9375-9389, 2018, doi: 10.1109/ACCESS.2017.2788044.
7. Huang Minsong, Lei Hengchi. An improved Holroyd cloud particle habit identification method and its application. Acta Meteorologica Sinica, 2020, 78(02):289-300.
8. Xiao Haixia, Zhang Feng. Classification of ICE Crystal Habits Observed From Airborne Cloud Particle Imager by Deep Transfer Learning. Earth and Space Science, 2019, 6:36-46.
9. Huang Minsong, Lei Hengchi. Ice particle habit distribution in strati form clouds at different precipitation stages. Transactions of Atmospheric Sciences, 2020, 45(02): 247-256.
10. Huang Minsong, Lei Hengchi, Wang Xiujuan, Ice particle habit classification method with improved thresholds and its application. Climate and Environment Research, 2020, 25(04):419-428..
11. Huang Minsong, Lei Hengchi. An improved Holroyd cloud particle habit identification method and its application. Acta Meteorologica Sinica, 2020, 78(02): 289-300.