



HAL
open science

Satellite Imagery-Based Cloud Classification Using Deep Learning

Rukhsar Yousaf, Hafiz Zia Ur Rehman, Khurram Khan, Zeashan Hameed Khan, Adnan Fazil, Zahid Mahmood, Saeed Mian Qaisar, Abdul Jabbar Siddiqui

► **To cite this version:**

Rukhsar Yousaf, Hafiz Zia Ur Rehman, Khurram Khan, Zeashan Hameed Khan, Adnan Fazil, et al.. Satellite Imagery-Based Cloud Classification Using Deep Learning. *Remote Sensing*, 2023, 15, 10.3390/rs15235597 . hal-04425577

HAL Id: hal-04425577

<https://hal.science/hal-04425577>

Submitted on 30 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Article

Satellite Imagery-Based Cloud Classification Using Deep Learning

Rukhsar Yousaf¹, Hafiz Zia Ur Rehman², Khurram Khan³, Zeashan Hameed Khan^{4,*}, Adnan Fazil¹, Zahid Mahmood⁵, Saeed Mian Qaisar^{6,7} and Abdul Jabbar Siddiqui^{8,9}

¹ Institute of Avionics & Aeronautics (IAA), Air University (AU), Islamabad 44000, Pakistan; rukhsaryousaf15@gmail.com (R.Y.); adnan.fazil@mail.au.edu.pk (A.F.)

² Department of Mechatronics and Biomedical Engineering, Air University (AU), Islamabad 44000, Pakistan; hafizzia@mail.au.edu.pk

³ Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute, Topi 23460, Pakistan; khurram.jadoon@giki.edu.pk

⁴ Center for Intelligent Manufacturing & Robotics (IRC-IMR), King Fahd University of Petroleum & Minerals (KFUPM), Dhahran 31261, Saudi Arabia

⁵ Department of Electrical and Computer Engineering, COMSATS University Islamabad, Abbottabad Campus, Abbottabad 22060, Pakistan; zahid0987@cuiatd.edu.pk

⁶ Electrical and Computer Engineering Department, Effat University, Jeddah 21478, Saudi Arabia; sqaisar@effatuniversity.edu.sa

⁷ CESI LINEACT, 69100 Lyon, France

⁸ Department of Computer Engineering, King Fahd University of Petroleum & Minerals (KFUPM), Dhahran 31261, Saudi Arabia; abduljabbar.siddiqui@kfupm.edu.sa

⁹ SDAIA-KFUPM Joint Research Center for Artificial Intelligence, King Fahd University of Petroleum & Minerals (KFUPM), Dhahran 31261, Saudi Arabia

* Correspondence: zeashan.khan@kfupm.edu.sa

Abstract: A significant amount of satellite imaging data is now easily available due to the continued development of remote sensing (RS) technology. Enabling the successful application of RS in real-world settings requires efficient and scalable solutions to extend their use in multidisciplinary areas. The goal of quick analysis and precise classification in Remote Sensing Imaging (RSI) is often accomplished by utilizing approaches based on deep Convolution Neural Networks (CNNs). This research offers a unique snapshot-based residual network (SnapResNet) that consists of fully connected layers (FC-1024), batch normalization (BN), L2 regularization, dropout layers, dense layer, and data augmentation. Architectural changes overcome the inter-class similarity problem while data augmentation resolves the problem of imbalanced classes. Moreover, the snapshot ensemble technique is utilized to prevent over-fitting, thereby further improving the network's performance. The proposed SnapResNet152 model employs the most challenging Large-Scale Cloud Images Dataset for Meteorology Research (LSCIDMR), having 10 classes with thousands of high-resolution images and classifying them into respective classes. The developed model outperforms the existing deep learning-based algorithms (e.g., AlexNet, VGG-19, ResNet101, and EfficientNet) and achieves an overall accuracy of 97.25%.

Keywords: deep learning; SnapResNet model; weather forecasting; satellite image; predictions



Citation: Yousaf, R.; Rehman, H.Z.U.; Khan, K.; Khan, Z.H.; Fazil, A.; Mahmood, Z.; Qaisar, S.M.; Siddiqui, A.J. Satellite Imagery-Based Cloud Classification Using Deep Learning. *Remote Sens.* **2023**, *15*, 5597. <https://doi.org/10.3390/rs15235597>

Academic Editor: Yuriy Kuleshov

Received: 24 September 2023

Revised: 21 November 2023

Accepted: 29 November 2023

Published: 1 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Better weather predictions are essential for people's daily lives [1]. For everyone concerned with assuring aircraft and human safety, accurate weather forecasting has been a top priority [2]. The aviation industry strives for effective systems for precise forecasting since the issue has not yet been fully resolved [3]. In addition to other reasons, the weather is the biggest hazard to both aircraft and people on board since it may harm both the aircraft and the passengers [4]. Since, weather is a nonlinear, complex and time-varying

phenomenon, accurate prediction is still challenging despite the development of theories, observation systems, and prediction tools which have advanced significantly over the last decade [5]. Air traffic, severe weather alerts, maritime, agricultural, and utility businesses, the commercial sector, the military, and architectural design are scant sectors that rely on quick analysis and precise classification of satellite images. There are already hundreds of forecast models available, each designed for a particular objective. At the same time, several research communities have become increasingly interested in deep learning (DL)-based weather forecasting [6,7]. There are positive applications of DL in computer vision such as the tasks of image recovery [8], face detection [9], semantic segmentation [10], and saliency detection [11]. In addition, it has applications in remote sensing (RS) image processing such as image classification [12–14], scene classification [15], and image prediction [16].

The research community in the field of satellite image analysis has thus turned its attention primarily to satellite image classification algorithms. A large-scale cloud image database for meteorological research (LSCIDMR) is proposed in [1] and these data are captured by the Himawari-8 satellite. As per the author's knowledge, it is the first publicly accessible benchmark database for meteorological research on cloud images [1]. The information in satellite cloud images regarding clouds, the atmosphere, and the seas is gathered by weather satellites. Weather satellites are crucial for weather analysis and meteorological catastrophe alerts since they are mainly used to observe Earth's weather conditions [17]. Research on weather forecasting has been conducted by meteorologists utilizing numerical weather prediction (NWP) [18]. The goal of this study is to apply a new method to forecast the weather with the maximum accuracy possible while resolving interclass similarities and class imbalance problems. For weather forecasting, numerous DL algorithms are used [19] such as MobileNet, VGGNet, ResNet, and EfficientNet.

The LSCIDMR dataset [1] is considered the most challenging satellite imagery dataset to date in terms of inter-class similarity and class imbalance. Unfortunately, the application of deep CNN architectures in remote sensing image classification is still somewhat restricted since deeper networks frequently encounter class imbalance and inter-class similarity in datasets. As a result, the trained model starts to favor certain classes over others and makes incorrect predictions. The inter-class similarity means have similar features but belongs to different classes of dataset as shown in Figure 1. Class imbalance is the unequal distribution of images in classes as shown in Table 1. The inter-class similarity is a challenge that can be resolved by a model with deep architecture. The researchers have been unable to fully realize deep networks' promise for remote sensing image classification because of this conundrum. Thus, in order to gather a complete perspective of deeper networks for the LSCIDMR dataset, effective techniques/learning procedures to counter inter-class phenomena are deemed necessary. To classify the LSCIDMR dataset using deep CNN architectures such as ResNets, a thorough optimization/training mechanism is to be investigated.

The following is a list of our key contributions in this work:

- The major objective of this research is to create a DL architecture that can obtain the highest satellite image classification accuracy. In order to achieve this goal, various training/optimization strategies are investigated to counter the inter-class phenomenon while dealing with imbalanced training data. The research community currently views the accurate classification of satellite images with significant inter-class similarity as a difficult task. By revealing the peak potential of individual deep CNN models in conjunction with ensembling methodologies, the network's discriminative capacity is improved.
- For satellite imagery-based weather forecasting, a modified version of the SnapResNet152 method is suggested. Models in the earlier studies were trained on different datasets; nevertheless, they still need to be fine-tuned to be utilized in satellite image classification problems. This research offers a unique snapshot-based residual network (SnapResNet) that consists of fully connected layers (FC-1024), batch normalization (BN), L2 Regularization, dropout layers, dense layer, and data augmentation in the

training regime. However, the data augmentation has been employed using limited data augmentation techniques such as flipping, rotating, and so on.

- We classify satellite images that were captured by the Himawari satellite. A detailed study has been performed on this dataset. The architectural changes have been presented to overcome the inter-class similarity problem while data augmentation resolved the problem of imbalanced classes. Furthermore, the snapshot ensembling technique is applied to avoid over-fitting that further improved network performance. The input to a system is a high-resolution satellite image, and the class of the input images were the output. The developed algorithm outperforms the existing deep learning-based algorithms.

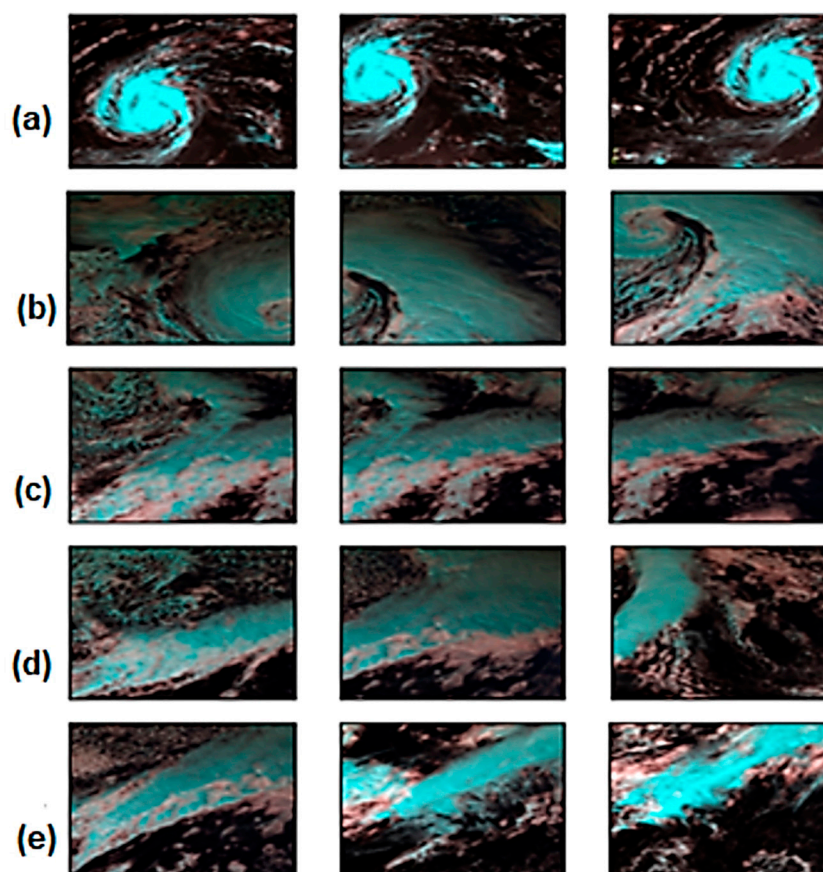


Figure 1. Inter-class similarity: (a) tropical cyclone (b) extra-tropical cyclone (c) high ice cloud. images from (a–c) all have blue hues and the (d) frontal surface (e) westerly jet has a bluish cyclonic curve with elongated cloud belts.

We believe that the proposed algorithm will improve the classification of satellite images having high resolution, class imbalance and high inter-class similarity. It is a challenging task to classify images that have very high inter-class similarity. Therefore, we are optimistic that the developed method provides an excellent insight into classifying satellite images. This paper is organized as follows: Related work is described in Section 2, with ResNet architecture, and SnapResNet details in Section 3. Experiments and results using different ResNet variants followed by comparison with state-of-the-art methods are presented in Section 4. Finally, Section 5 summarizes the work of this study and provides future research directions.

Table 1. Class imbalance: In LSCIDMR-S and LSCIDMR-M all ten classes have an unequal number of images. The label-less class is only included in LSCIDMR-S and has a large share of images. This imbalance leads to model biasing.

Type	Images in LSCIDMR-S	Images in LSCIDMR-M
Tropical Cyclone	3305	3.17
Extra-tropical Cyclone	4984	4.77
Frontal Surface	634	0.61
Westerly Jet	628	0.60
Snow	7631	8.33
Low Water Cloud	1774	95.14
High Ice Cloud	5278	91.99
Vegetation	7831	42.43
Dessert	4518	56.95
Ocean	4042	89.81
Label-less	63,765	-

2. Related Work

Authors in [20] proposed a novel lightweight data-driven weather forecasting model by investigating temporal modelling techniques of Long Short-Term Memory (LSTM) and temporal convolutional networks. The deep model involves two regressions, i.e., multi-input multi-output and multi-input single-output. This model demonstrates more accurate forecasting results up to 12 h as compared to the weather research forecasting (WRF) model. Booz et al. [21] proposed a deep learning-based weather forecast system where the real-world dataset is utilized for data volume and recency analysis. It is concluded that having more data is helpful in increasing the accuracy of the model as the recency of the data does not have a huge effect on the accuracy of the model.

Self-Organizing Map (SOM)-based Latent Dirichlet Allocation (LDA) was coined by Pushpa Mohan et al. [22] in which they reduced dimensionality using SOM and then reduced the amount of data utilized to forecast the climate. This method shows 7–23% improved accuracy for weather and crop prediction as compared to existing methods. Moreover, R. Meenal et al. [23] observed better weather forecasting could be obtained by using artificial intelligence-based methods such as machine learning and deep learning.

A DL-based LSTM was developed by Anil Utku et al. [24], and compared with traditional machine learning algorithms such as RF (Random Forest) and SVR (Support Vector Regression). It was concluded that the developed LSTM significantly outperforms the considered benchmarks. Pengwei Du et al. [25] presents an ensemble machine learning-based method (artificial neural network, support vector regression, Gaussian process) to forecast wind power production, which uses wind generation forecast made by numerical weather prediction (NWP) and, from the weather station, they collected data of meteorological observation. Singh et al. [26] used the deep learning-based UNET with residual learning as evidence of the idea for studying global data-driven precipitation models. This paper shows that the residual learning-based UNET can unravel the physical relationships to precipitation, thus allowing the physical constraints to be used in the dynamical operational models for improved forecasts of precipitation. Their results are crucial as they opened the way forward to the development of online, hybrid models for prediction.

Kim et al. [27] proposed an LSTM to predict precipitation. They used the tropospheric delay of the GNSS (Global Navigation Satellite System) signals to compute the PWV (Precipitable Water Vapor) for the numerical test, and then they used the deep learning approach to forecast precipitation. Scientific software used the Saastamoinen troposphere model and the Niell mapping function to process the GNSS data. The study shows that precipitation prediction based on LSTM performs better than that of ANN (Artificial Neural Network) using RMSE (Root Mean Squared Error). The addition of GNSS-based PWV as a feature prevented the over-fitting problem considerably as discussed in this study. Abdalla et al. [28] suggested that for the accommodation of non-linearity of time series datasets in

weather forecasting applications, deep learning architectures could be efficiently devised. In terms of the design of neural network topologies, geographical and temporal dimensions, datasets, and benchmarks, this study summarizes the most recent deep learning-based weather forecasting research. The learned findings are then highlighted, with an emphasis on the claimed accuracy and the prediction scale for model generalization. This is an essential requirement to judge if the model is appropriate for a local or regional weather prediction and also to establish trust in the short-term and long-term forecasts. Moreover, the paper formulates the independent and dependent variables for weather forecasting and evaluates the algorithms used for training the dataset with a focus on time efficiency in each study. Finally, the authors describe the dependent and independent variables for weather forecasting and conduct assessments for optimal time strategies in training the dataset. In another study, Kang et al. present a weather image recognition framework based on deep learning while considering haze, rain, and snow, in outdoor scenes [29]. Their system automatically classifies an input image into respective categories. To evaluate the proposed method, GoogLeNet and AlexNet are operated on an open weather image dataset and the feasibility has been verified.

The main objective of this study is to employ an improved loss function that better explains the interclass interaction. In contrast to the typically used cross entropy or softmax loss, which is limited to the classification probability and score of individual scene classes [30]. In other words, classifying images that have inter-class similarity is not as accurate as classifying images which do not have inter-class similarity. To improve the accuracy of classifying images having inter-class similarity, authors in [13] suggest a Discriminative CNN architecture (D-CNN), in which, in addition to minimizing the cross-entropy loss, a metric learning regularization term (L2 regularization) is applied to improve the proposed architecture's discriminative capabilities. This enables images from the same image class to be clustered together, whilst images from other classes end up being separated.

A large-scale cloud image database for meteorological research (LSCIDMR) is proposed by Cong Bai et al. [1]. The authors claim that it is the first publicly accessible satellite cloud image database for meteorological research. LSCIDMR contains two annotations (Single and Multiple) with a total of 104,390 images covering 11 classes. Each class of image define a different types of features. In total, 414,221 multiple labels (LSCIDMR-M) and 40,625 single labels (LSCIDMR-S) are obtained. In single labels (LSCIDMR-S), each image only belongs to one class and has specific features of that class but, in multiple labels (LSCIDMR-M), images may belong to one or more classes and have features of one or more classes. LSCIDMR-M and LSCIDMR-S together provide a full version of LSCIDMR, and LSCIDMR-M is in some ways an extension of LSCIDMR-S. Numerous deep learning methods such as VGGNet-19, ResNet-101, AlexNet and EfficientNet are applied to these images and the obtained results are used as a baseline for future work.

The aforementioned references make a compelling argument for satellite image classification as a significant stage in developing systematic processes such as an intelligent classification system. The approaches shown above are just a few of the effective and well-done efforts aimed at addressing the image classification problem in diverse contexts. We believe that this research work will be a valuable addition to the field of satellite image classification.

3. Methodology

Remote sensing satellite-based cloud image classification is a challenging problem due to inter-class similarities and class imbalance issues. In order to address this issue, an extremely deep CNN network like ResNets is used. In addition, effective optimization strategies along with logical architecture alterations are adopted.

In ResNet, the added residual block draws the output from one of the preceding layers to the extracted feature map (Rectified Linear Unit (ReLU) activation) of the following layer. While allowing for significantly deeper network penetration, these identity mappings do not adversely affect the performance of the network. The deeper layers can now

learn more complicated features, which leads to increased classification performance in images. This significantly improves the network's capacity for learning features. On the massive ImageNet dataset, ResNets displayed astounding image classification ability. The innovative approach that made it possible to go deeper into the network beat the shallow rival networks. This encourages the usage of ResNets for satellite RS image classification.

3.1. Baseline Model Architecture

In this research, ResNet50, ResNet101, and ResNet152 are deployed to classify remote-sensing images. Firstly, the ResNet50 has been selected to be the baseline model for developing the methodological framework. Figure 1 displays the block diagram of the ResNet50 layered structure. The image was adopted from [31]. Here, it is crucial to highlight a limited key aspect of ResNet architecture. To reduce the number of trainable parameters and therefore training time, FCL is used as little as possible (single SoftMax-based FCL). Since the VGGNet framework served as inspiration, the largest kernel size is limited to 3×3 . Unlike in classic CNN designs like AlexNet [32] and VGGNet [33], none of the convolution blocks (except the one after the first convolution layer) include a pooling operation. Immediately following the final convolution block, a GAP (Global Average Pooling) layer is added. Each convolution block in the network is trained with batch normalization in addition to the global normalizing of the data.

The changing aspects of the target dataset have a significant role in the decision to deploy fully connected layers in CNNs. The entire spatial receptive field of the image is covered by FCL in contrast to feature maps which are extracted using convolution layers. The convolution layers have a specific receptive field which is determined by the kernel size. As a result, FCL improves the performance of the network by training complex non-linear functions within the feature space. The disadvantage of employing FCL is the increase in computing costs that comes along with an increase in the number of training parameters. For instance, AlexNet has 60 million total parameters, 58 million of which are linked to FCL. The 138 million parameters in VGGNet are contributed by FCL, who furthermore supplies 123 million parameters [32]. As a result, the computational cost of including FCL has compelled the researchers to create other strategies. Considering the foregone, the ResNet structure only includes the network's FCL, which is an essential last layer necessary for obtaining classification scores as probabilities using SoftMax activation.

In CNNs, the convolution kernel size is entirely determined by empirical findings or by accepted research community conventions. Most often, researchers choose a kernel size that has been successful in the literature for a dataset that is semantically comparable to the target data. In this regard, the inventors of ResNet were motivated by the success of VGGNet in the image classification challenge and used the same 3×3 convolution kernel in various stacked convolution blocks.

Pooling layers in CNNs are primarily used to reduce dimensionality and keep calculations within reasonable bounds. This goal has been accomplished by ResNets using 1×1 convolution kernels. It can be noticed that each convolution block uses a 1×1 bottleneck layer to limit the dimensions of the supplied activation input prior to successfully applying a 3×3 convolution kernel to increase the dimensions once more as an ultimate step. The 1×1 bottleneck layers provide the additional benefit of learning noteworthy features as they are a convolution layer, in addition to flexibility in handling output dimensions.

After the last convolution block in ResNets, a GAP layer was added that not only decreases spatial dimensions but also takes the role of the flattening operation that is typically used to determine the input dimensions for FCL. Our study is particularly interested in a different viewpoint on the utilization of the GAP layer. The output from the last convolution block of the ResNet50 architecture, which is shown in Figure 2, includes 2048 activation maps, which is equal to the number of 1×1 kernels that are a fixed entity in this architecture. Regardless of the size of the original input to the network, applying the GAP layer on these 2048 activation maps produces a 2048-dimensional column vector. The obligation of adjusting the input image to match FCL dimensions is not obligatory, as

found in the case of VGGNet and AlexNet since the addition of the GAP layer has reduced the ResNet architecture without putting any constraint on the size of the input image. Thus, any input image size (such as 256×256) can be used for LSCIDMR to retrain the same architecture. This aids in preventing the loss of crucial feature data that are essential to the process of image classification.

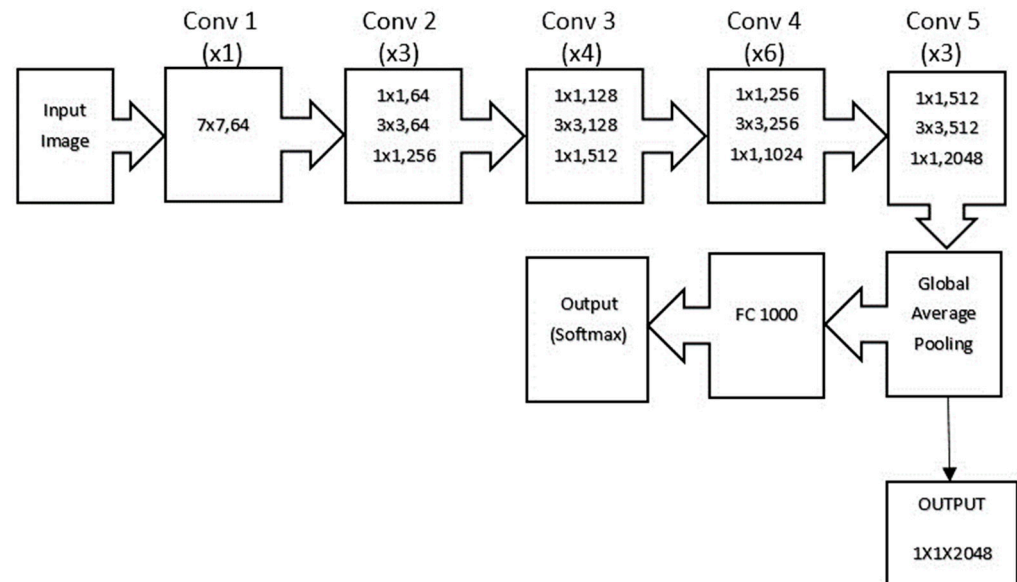


Figure 2. ResNet50 architecture, used as a baseline model for the proposed scheme [31].

CNN training is accelerated by the idea of normalizing the input data to the network. A comparable concept applied to the input of the network's hidden layers is called batch normalization. When working with very deep networks, the input data distribution per mini-batch may alter as the weights are updated farther into the network. Due to this, the learning algorithm is forced to follow a continually shifting target, which causes convergence problems. Internal covariate shift is the name given to this phenomenon [34,35]. This issue is resolved by BN layers by standardizing the input to the network's deep layers, which stabilizes the training process and significantly decreases training time [36]. This is due to the fact that the batch norm makes training independent of the learning method and the network's starting setup. Similar to dropout, batch norm has a regularization impact that aids in preventing the overfitting phenomena [37]. When standardizing, the mean and variance are calculated for each mini-batch of the input data. This introduces some noise into the training data and has the effect of regularizing it, keeping the optimization function from becoming too complicated. This BN layer's regularization impact will vary depending on the mini-batch size. The regularization impact decreases as the mini-batch size increases.

Although the stabilization of the training process and subsequent reduction in training time are the main advantages of BN layers, the regularization effect that they provide is still of great interest to the proposed research. It will be utilized with the LSCIDMR dataset, which has an imbalance in volume and causes overfitting problems when training deep networks like ResNets.

Initially, ResNet50 was utilized [38] by changing the number of neurons in the FC layer to account for the 10 classes in the LSCIDMR dataset as compared to the 1000 item categories in ImageNet. Figure 3 displays the modified ResNet50 architecture exploited to train the LSCIDMR dataset. Moreover, we initially tested the training-from-scratch method using the basic ResNet architecture. With this strategy in mind, it was observed a clear overfitting occurrence where the network overfitted the training data. The train-test performance bias, which is calculated as the difference between train and test accuracy, illustrates the network's lack of generalizability because of the overfitting phenomenon.

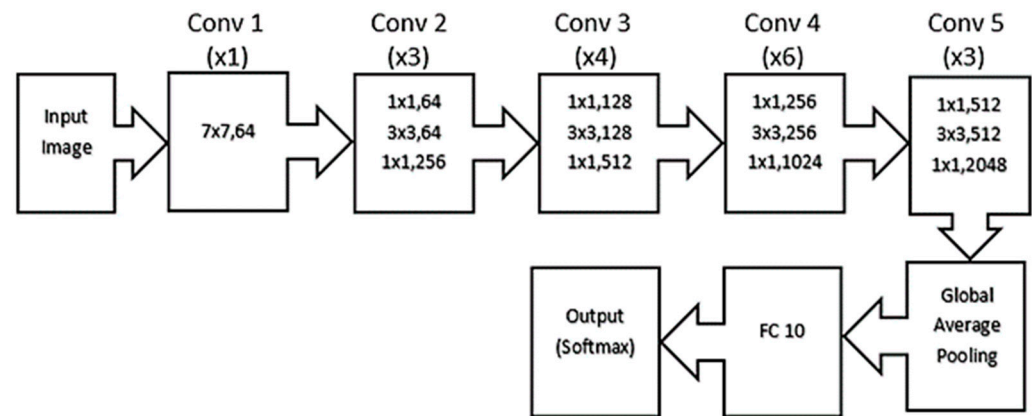


Figure 3. The original ResNet50 architecture for 10 classes of the LSCIDMR dataset.

Hence, using the same settings and a transfer learning (fine-tuning) strategy, we trained the network on the LSCIDMR dataset. Several transfer learning strategies range from training the whole network to training just a small number of network layers. In the version considered here, the learned weights from the extensive ImageNet dataset served as the initial point for proposed model building as we trained the whole network. This step was taken due to the inclusion of BN layers, which normalize input to hidden layers according to the considered batch size during the training process. As freezing any part of the proposed network requires the target image dataset to be in perfect synchronization with the previously established normalized data distributions for the ImageNet dataset, this is a very hard procedure. As a result, we utilized the no-layer freezing strategy. When fine-tuned using ImageNet weights, the classification accuracy, which had previously peaked at 80% using the training-from-scratch technique, has now surpassed 90%. The reasoning for this improvement is that the CNN weights were previously started at a location that was a great distance from the local minimum of the loss function being sought by Stochastic Gradient Descent (SGD). As a result, there has been a considerable increase in classification accuracy in recent times. It is crucial to remember that overfitting, which previously reduced model performance when trained from scratch, continues to be an issue in the network and is waiting for efficient solutions.

3.2. Learning Rate

The learning rate is one of the crucial hyperparameters and can affect the model's accuracy to the greatest degree. Initially, a fixed value (0.001) of the learning rate is considered to train ResNet50. From the literature on deep learning, it was observed that a scheduled learning rate performs better than a fixed value. The basic idea is to start the training process with higher learning rates so that the optimizer can move more quickly toward the global minimum. As training progresses, the learning rate values gradually decrease to allow the optimizer to move smoothly toward the target without exceeding the minimum. The step decay plan and the outperforming cosine annealing schedule were also tested as shown in Figure 4. Following a predetermined timetable, the learning rate is reduced in the step decay after a number of epochs. The cosine annealing schedule, on the other hand, follows a cosine curve for each epoch.

It is suggested to apply the cyclic cosine annealing function in [32], which is given in Equation (1):

$$\alpha(t) = \frac{\alpha_0}{2} \left(\cos \left(\frac{\pi \text{mod}(t-1, \lceil T/M \rceil)}{\lceil T/M \rceil} \right) + 1 \right) \quad (1)$$

where T is the total number of training epochs, M is the number of cycles, (t) is the needed learning rate at epoch t , and α_0 is the initial learning rate. We only want to train the model once, i.e., $M = 1$ in this scenario.

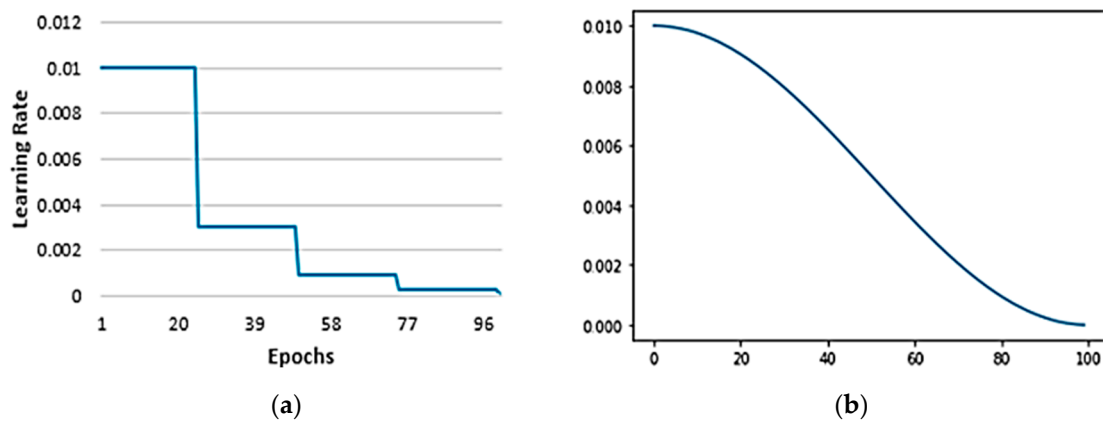


Figure 4. Learning rate schedules. (a) Step decay. (b) Cosine decay.

3.3. Dropout Layer

As was previously mentioned, the training data are being overfitted by the ResNet50 architecture that is presently being used with the LSCIDMR dataset. Effective methods to combat this problem are urgently needed since they prevent ResNets from reaching their full potential in image classification. To generate a model that fits the test data well or to reduce overfitting, there are certain things we may do on both the model and data sides.

Dropout is a frequently used common strategy in deep learning models that produces a regularization effect and improves the network's generalization capacity [39]. Probabilistically, dropout reduces the input supply to the network layers by a certain percentage. This is accomplished by multiplying some of the input neurons by 0 and some by 1. It makes sense that by the addition of noise to the input data, which strengthens the network's resiliency to input, the danger of network overfitting is reduced.

The existence of BN layers in the ResNet design has previously been covered in great depth. There is no need to include dropout layers in the convolutional portion of the network since BN offers a regularization effect that is comparable to that of dropout in addition to a number of additional benefits [40]. Dropouts are often employed in the FCL section of networks; however, the ResNet50 design under consideration does not include an FC layer that is suitable for dropout coupling. The only place for dropout (in red) is immediately following the GAP layer, as shown in Figure 5. A detailed examination of the outcomes showed that, while employing dropout to probabilistically drop the GAP layer output, we permanently lost several inputs that were essential to the network's capacity for feature discrimination. Each node in the GAP layer contains distinct feature data that were taken from the last convolution block's activation maps. Therefore, the only feasible way to deploy a dropout approach was to change the existing design by adding an additional completely linked layer that would act as a foundation.

Determining the number of neurons in this new FC layer was the natural next step. In the deep learning community, it is recommended to gradually reduce the number of neurons as one advances through FC layers. Just enough neurons should exist to adequately capture the target dataset's variety. Considering this, we chose to add a second FCL, consisting of 1024 neurons, following the GAP layer in the initial ResNet50 architecture.

3.4. Weight Decay Regularization

Weight decay [41] is another well-liked and straightforward method of keeping the model from becoming unnecessarily complicated. The network fits such a complicated curve on the training data while working with several parameters on a small amount of data that it performs badly on the test set. When the loss function tries to become unduly complicated, weight decay is supposed to punish it. One strategy is to punish the function by including all the weight factors in it. To obtain only positive values, we can apply L2 or L1 regularization approaches. However, such a substantial addition can cause a

significant model loss, necessitating unrealistic (mainly 0) weight settings for the best fit. The answer is to multiply the weight parameter squares (L2 norm) by a tiny integer, add the product to the loss, and then solve the problem. Numerous CNN training approaches for the job of image classification have used this modest amount, known as the weight decay factor [32,33,41]. Weight decay regularization has been employed with the newly added FC-1024 in the ResNet50 design, as shown in Figure 6, as weight decay is often used with FCL as these layers contain most of the network parameters. This design is known as FC-1024 ResNet50.

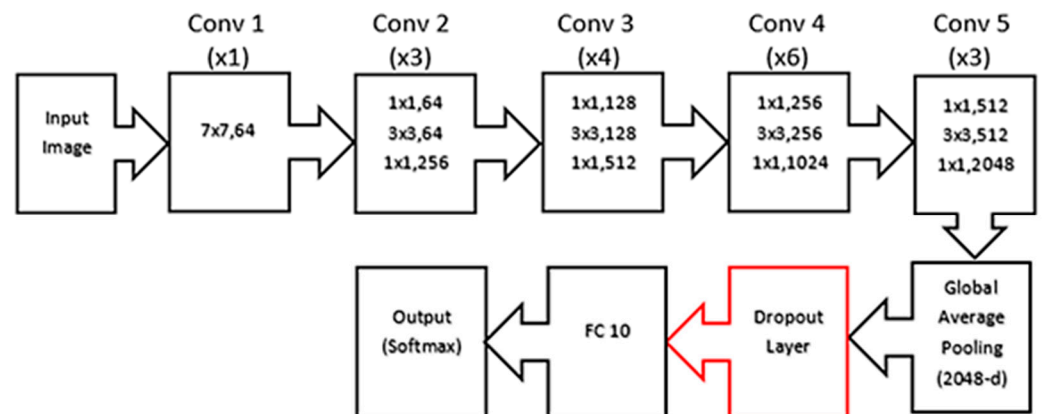


Figure 5. Modified design after the addition of a dropout layer (in red) in the original ResNet design to reduce the overfitting that has been observed in the model.

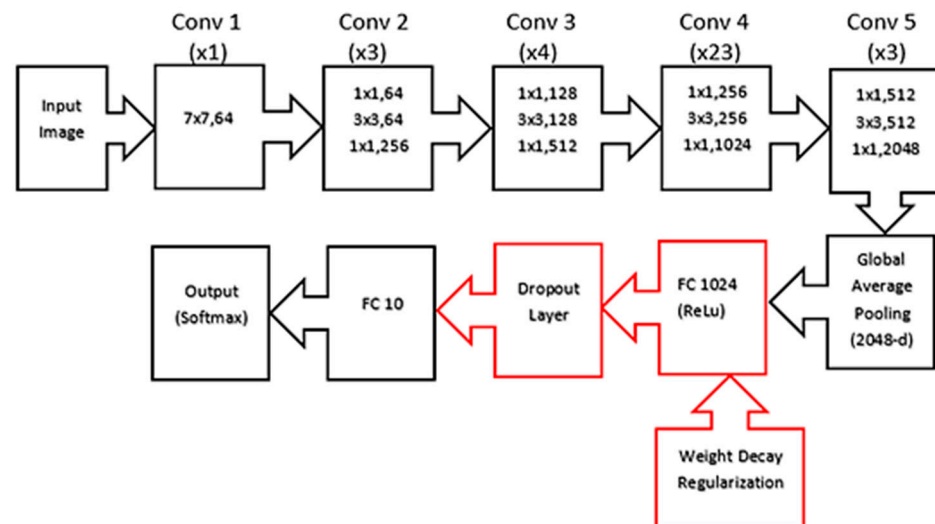


Figure 6. Proposed FC-1024 ResNet50 architecture; addition of FC-1024 to classifying images and weight decay regularization (additional red block) to help the model to generalize better hence improving the model performance on unseen or test datasets.

When training the FC-1024 ResNet50 architecture with 20% probabilistic neuronal dropout and a weight decay factor of 0.0005, improved image classification accuracy is seen compared to the original design without FCL, dropout, and weight decay. The utilization of dropout with FCL has proved successful in contrast to GAP + dropout due to the fact that every neuron in FCL has the essence of the GAP layer; therefore, losing specific neurons in FCL during network training does not result in a permanent loss of critical feature information.

3.5. Data Augmentation

In order to address the overfitting phenomena, we have solely examined the strategies that can be integrated into the network design side. The data augmentation strategies are also essential in this respect and have shown themselves to be successful in addressing network overfitting problems since they attack the fundamental cause of the issue, namely, a dearth of training data. These methods' task is to enhance the data in such a manner that the model may discover new descriptive characteristics. Although the target dataset contains variations in perspective, size, lighting, occlusion, color, etc., there are several data augmentation strategies for computer vision applications that aim to improve model performance [42]. The LSCIDMR dataset, which is the benchmark for satellite-based cloud image classification performance, is regarded as the most challenging dataset as it contains 10 classes and possesses high interclass similarity along with class imbalance. As was already mentioned, the dataset is highly imbalanced. Some classes of datasets have images in thousands while some of the classes have a few hundred images. This, in addition, leads to model biasing towards classes having a large number of images and inhibits the model from showing its full potential. In order to solve class imbalance, separate augmentation is applied to those classes so that these classes have almost the same number of images as the other classes.

The newly suggested random erasing data augmentation technique was used initially instead of immediately going on to the common data augmentation techniques that are focused on geometric changes in images. Random erasing is intended to address image occlusions. Any unclear objects in an image are referred to as "occlusions". By arbitrarily removing discrete parts of the image and replacing them with random pixel values, this technique forces the network to concentrate on the overall/global structure of the image rather than overfitting to a particular collection of local features. The network will be able to learn extra descriptive properties unique to each class in the dataset by improving the data in this way, which should increase the accuracy of image classification. More importantly, it will prevent the network from depending solely on image details, which will partially solve the issue of inter-class similarity in the case of satellite RSIC. The FC-1024 ResNet50 model is trained next using random erasing to inherit the advantages of this method and to compare the results. In line with the aforementioned benefits, the random erasing data augmentation strategy alone significantly increased the image classification accuracy.

The most popular data augmentation techniques, based on geometric and color space transformations, are often effective for a wide range of jobs. The methods for data augmentation are Image flip (vertical and horizontal), Image rotation, Height and width shift, Brightness variation, Image zoom, Image shear, Scale variations, Contrast variations and multi-augmentation. Applying 2 or more of the data augmentations outlined above at once is known as the multiple-augmentation technique. The utilization of all data augmentation approaches led to a change in the composition of the input data, which negatively influenced the generalization of the model on the test set. We added one approach at a time to the data, isolating this problem. Through this process, we were able to identify the data augmentation techniques—such as picture zoom, shear, height, and breadth shift—that were behind the data domain change. Finally, we trained the model once again using the other strategies after excluding these three data augmentation methods. Now, the trained model's overall test accuracy has increased relative to all previously described training regimes.

3.6. Proposed Training Regime

At this time, the training regimen for the FC-1024 ResNet50 architecture that had been used was deemed complete. Figure 7 provides a synopsis of the training methodology's route map. We trained the network using SGD as the optimizer for 100 epochs during the given training regime, utilizing a 90 percent training ratio.

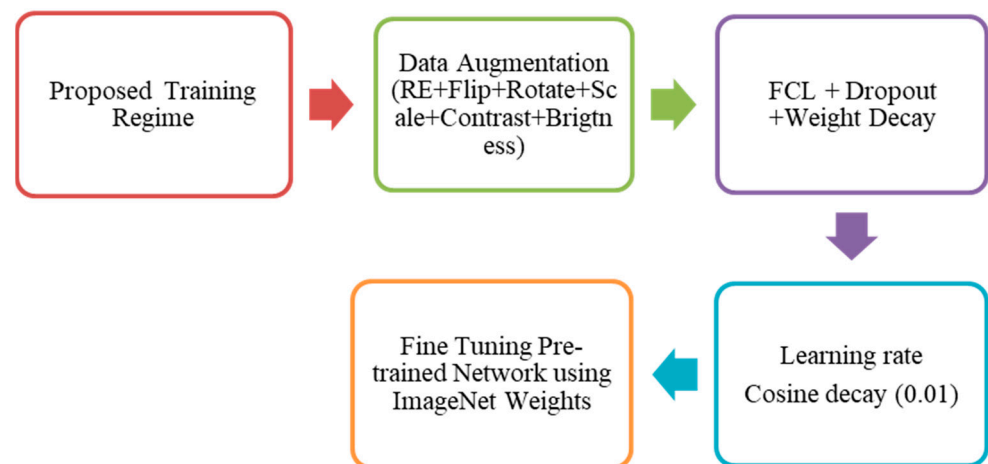


Figure 7. Proposed training regime, highlighting the steps to further improve the proposed model.

3.7. Proposed SnapResNet Architecture

Snapshot ensembles utilize the non-convex character of the basic cost function, which is reduced throughout the network training process. When compared to the test set, the network's performance varies as a result of SGD convergence on various minima. It was suggested to integrate the snapshot ensembling approach with the FC-1024 ResNet50 training regime. Figure 8 contains the block diagram of the suggested network setup. Snapshot-based residual networks are what we refer to as SnapResNet. On the LSCIDMR dataset, we achieved cutting-edge classification performance using SnapResNet.

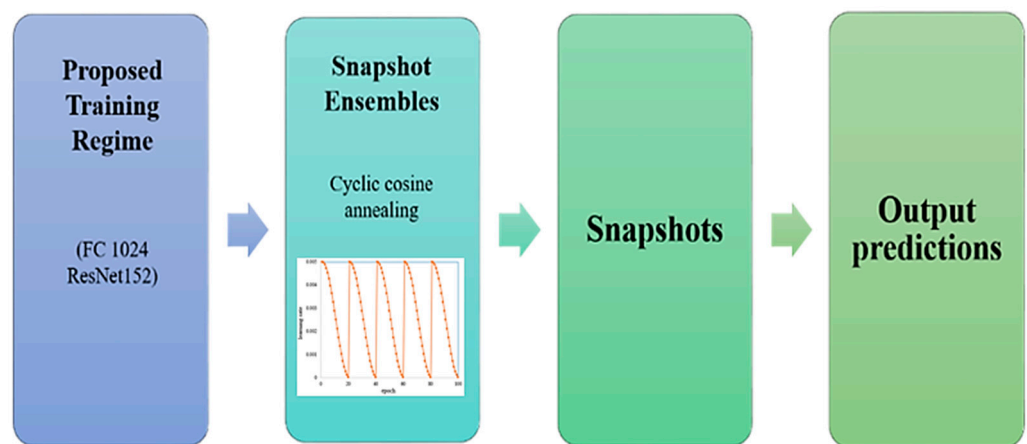


Figure 8. Proposed SnapResNet architecture. In the FC-1024 ResNet152 model, we apply snapshot ensembling with cyclic cosine annealing to then, in return, obtain snapshots where the model accuracy was higher. Now, the proposed model is converted into SnapResNet152 that is used to classify images.

4. Experiments and Results

For better comprehension of the findings, the proposed SnapResNet employs an ensembling approach similar to [31]. However, it differs in two ways. Firstly, the proposed architecture does not apply majority voting. Secondly, it does not combine the SnapEnsemble and Dilated Conv models in snapshot ensembling. The first tier deals with snapshots of the model, while the second layer emphasizes the impact of integrating several snapshots derived from the model.

4.1. Dataset and Preparation

The model was trained and tested using an LSCIDMR obtained by the Himawari-8 satellite. This is the first weather satellite capable of taking color images and monitoring authentic cloud images. The 11 classes are represented by the 104,390 high-resolution

(256 × 256) images in LSCIDMR. The temporal resolution is 10 min, and the spatial resolution is 2.0 km. The dataset contains descriptions of four systems as shown in Table 2. (1) Weather system: A weather system includes a tropical cyclone, extratropical cyclone, frontal surface, westerly jet, and snow. (2) Cloud system: The cloud system is made up of high ice clouds and low water clouds. (3) Terrestrial system: This group includes the ocean, the desert, and the vegetation. (4) Label-less. Both single and multiple-label annotations, known as LSCIDMR-S as shown in Figure 9 and LSCIDMR-M, respectively, are available for the dataset. The labels are carefully annotated, yielding a total of 40,625 single labels and 414,221 multiple labels.

Table 2. Parameters for choosing categories in LSCIDMR, developed by meteorology experts.

Type	Classification Criteria
Tropical Cyclone	The eye of a Tropical Cyclone should be in the slice.
Extratropical Cyclone	The eye of the Extratropical Cyclone should be in the slice
Snow	Snow is in the slice
Westerly Jet	Westerly jet is in the slice
Frontal Surface	Frontal Surface on the slice
High Ice Cloud	Area (High Ice Cloud) > 50% and Area (else) < 20%
Low Water Cloud	Area (Low Water Cloud) > 50% and Area (else) < 20%
Vegetation	Area (Vegetation) > 50% and Area (else) < 20%
Ocean	Area (Ocean) > 80% and Area (else) < 20%
Desert	Area (Desert) > 50% and Area (else) < 20%
Label-less	Do not belong to the above 10 classes

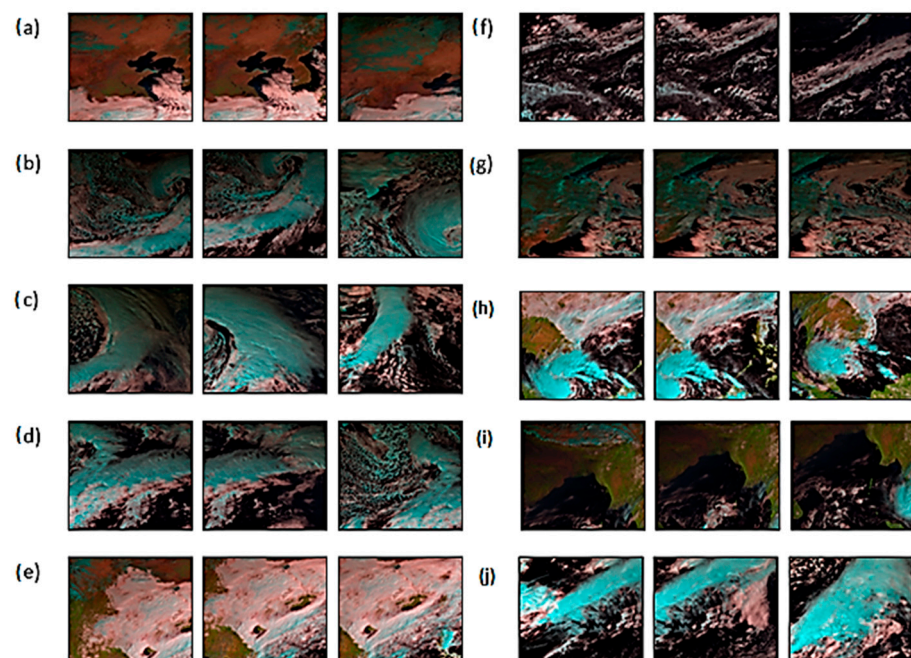


Figure 9. Images from LSCIDMR-S: Example of some images from ten classes. There are 40,625 images within ten classes. (a) Desert, (b) extratropical cyclone, (c) frontal surface, (d) high ice cloud, (e) low water cloud, (f) ocean, (g) snow, (h) tropical cyclone, (i) vegetation, (j) westerly jet.

Using the selected images, the author of [1] conducted a statistical analysis of seasons. The seasons in the northern hemisphere are divided into spring (March, April, and May), summer (June, July, and August), autumn (September, October, and November), and winter

(December, January, and February). The statistical analysis results are shown in Figure 10. Tropical cyclones and extratropical cyclones can be seen in all seasons, but tropical cyclones are most common in summer and autumn, while extratropical cyclones are seen in spring and winter. Snow falls mostly in the spring and winter, with some in the summer because high latitudes in the northern hemisphere remain snow-covered. Westerly jets and frontal surfaces are less common throughout the year.

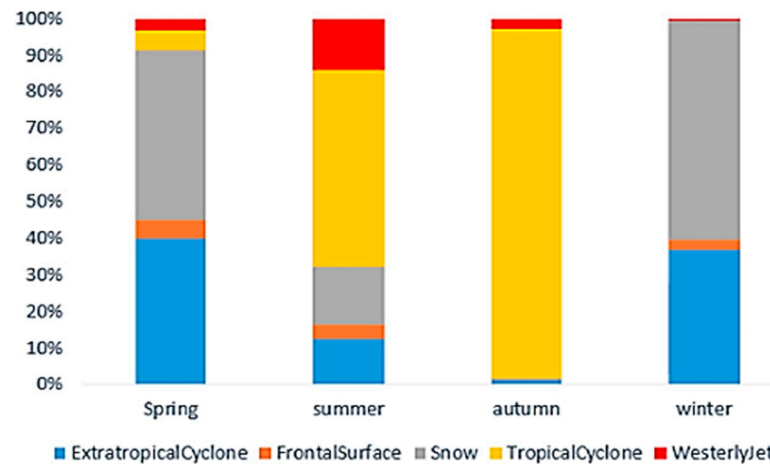


Figure 10. A statistical analysis of the proportion of images of each weather system in different seasons is shown. The seasons are divided according to the northern hemisphere.

4.2. Training Methodology

All of the experiments in this study were developed using the Keras framework on Windows 10 and trained on an NVIDIA GeForce RTX 3060 (12 GB) GPU. The experiment's software environment is Jupyter Notebook.

In the test, the input images to the neural network were 256×256 pixels in size from the LSCIDMR dataset. The maximum training epochs were set to 200 and the training batch size to 32. SGD optimizer has been applied for optimization, with a scheduling learning rate approach. In the experiment, ImageNet weights have been applied. Moreover, image flip (horizontal and vertical) and rotation augmentation techniques were applied earlier in each training epoch.

4.3. Evaluation Metrics

In this experiment, the proposed architecture is compared and evaluated by Precision, Recall, Overall Accuracy, F1-Score and Confusion Matrix as shown in Equations (2)–(5), respectively.

$$\text{Precision} = \frac{\text{No. of correctly predicted positive instances}}{\text{No. of total positive predictions you made}} \quad (2)$$

$$\text{Recall} = \frac{\text{No. of correctly predicted positive instances}}{\text{No. of total positive instances in the dataset}} \quad (3)$$

$$\text{Overall Accuracy} = \frac{\text{No. of correct prediction}}{\text{No. of all predictions}} \quad (4)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

4.4. Experimental Results

System parameters and specifications utilized in training and testing are specified in Table 3. The first logical step was selecting the appropriate number of images for the

proposed model. The two snapshots were employed as a starting point for the network and the output has been examined. Following the cyclic learning rate schedule, the learning rate was raised purposely after every 100 epochs. The snapshots have been saved following every 100 training epochs as portrayed in Figure 11.

Table 3. Parameters used for model training and system specification.

S No	Parameter	Value
1.	Train/Test ratio	0.9/0.1
2.	Training/Test images	46,580/5182
3.	Epochs	100 epochs/snapshot (200 in total)
4.	Iterations per epoch	1456
5.	Batch size	32
6.	Learning rate	Cyclic cosine annealing initialized at 0.01
7.	Weight decay regularization	L2 regularization
8.	Weight decay factor	0.0005
9.	Dropout rate	0.2
10.	Optimizer	SGD(Stochastic Gradient Descent)
Development Tools/Platforms		
11.	Jupyter Notebook	
12.	NVIDIA GeForce RTX 3060 (12 GB)	

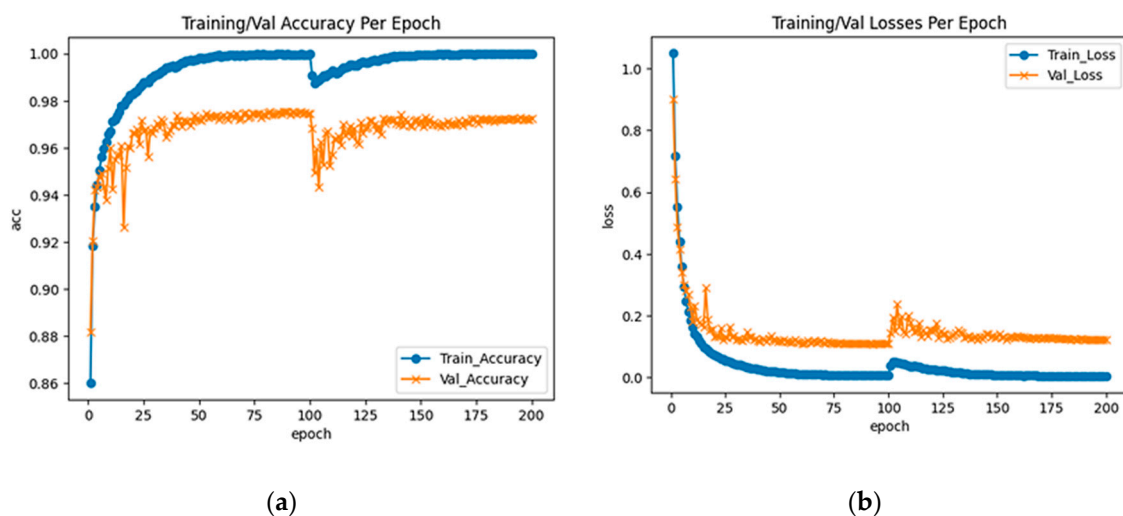


Figure 11. Snapshot ensembling: each snap is saved after 100 epochs. (a) Training and validation (val) accuracy in 2 snapshots (b) Training and validation (val) losses in 2 snapshots.

The question is whether the cyclic learning rate schedule has allowed SGD to converge to separate minima. Figure 12 shows how the accuracy of image classification dramatically increased after individually integrating the relevant images of the model. The increase in overall accuracy with each additional snapshot is adequate proof that SGD has reached different minima of the primary cost function in relation to the design of the network.

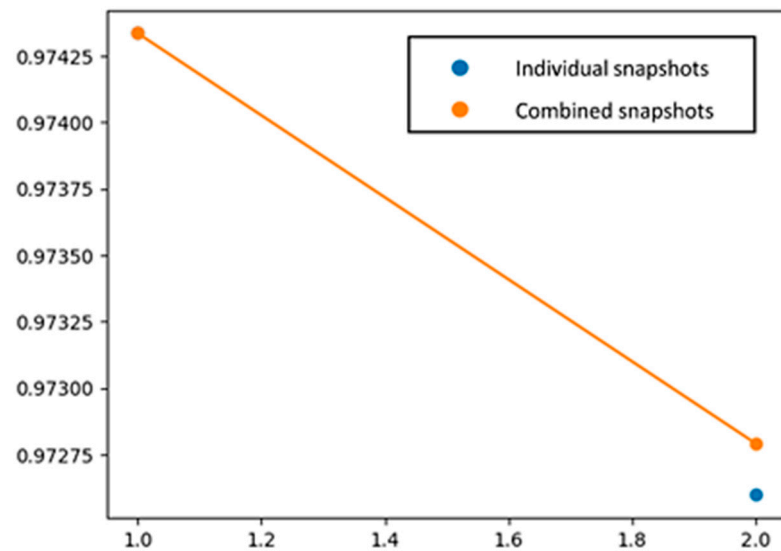


Figure 12. A blue dot shows a single Snapshot after 100 epochs and an orange dot shows combined snapshots after 200 epochs.

It is observed that, with FC-1024 ResNet152, the increase in overall accuracy that results from combining the first two snapshots is considerably bigger than the increase that results from using the single snapshot. The number of snapshots for the model was restricted to two, considering the computational expense associated with each extra snapshot. Subsequently to limiting the model’s snapshots to just two, it was observed that the suggested models had an overall accuracy of 97.25. The confusion matrix is displayed in Figure 13.

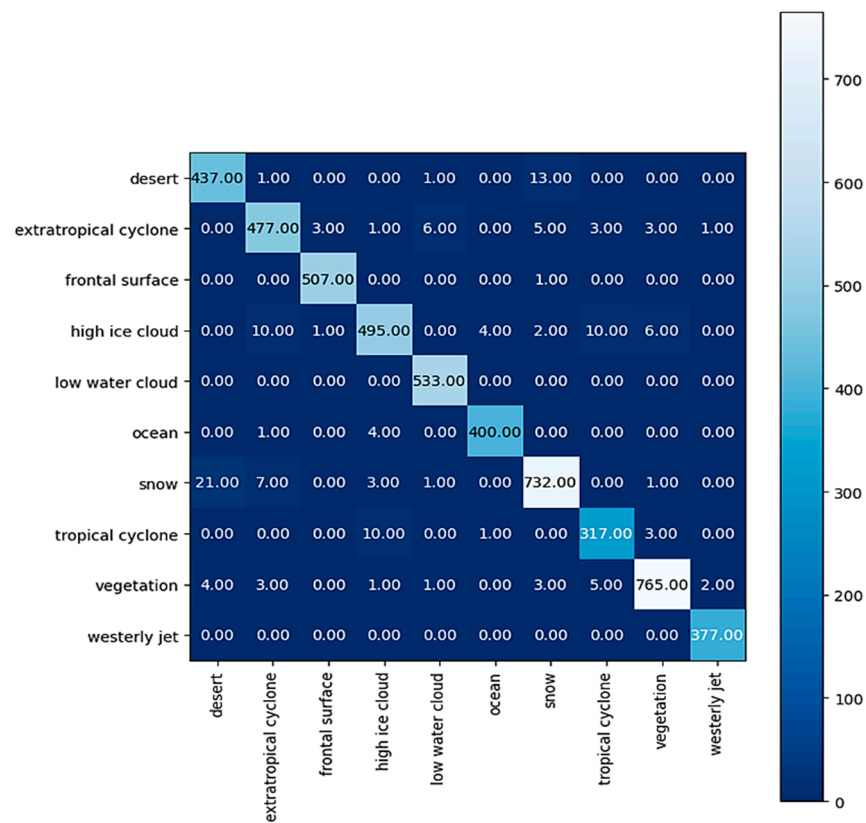


Figure 13. Confusion matrix for detailed analysis of the proposed scheme’s performance.

As baseline models for the suggested network design, we also took ResNet50, ResNet101, and ResNet152 into account. This was made feasible by the ResNet versions' shared fundamental structure; the only distinction was the number of convolution blocks, which determines the network's depth. Furthermore, 90% of the LSCIDMR was employed for training. After converting the baseline design from ResNet50 to ResNet101, an overall improvement in accuracy was observed. This was likely due to the deeper network's potential for learning more descriptive characteristics. Contrary to common perception, however, the behavior altered when converted from ResNet101 to ResNet152. For a given target dataset, the network's depth is not necessarily inversely related to the accuracy statistics that were acquired. There is a saturation threshold for learning the hidden complex features; hence, enhancing the depth past this level may lead to lower accuracy as the network itself becomes sophisticated enough to inevitably experience overfitting on the training dataset rather than learning complex features. This is the major reason that SnapResnet152 considerably improved performance.

We measured the suggested model's accuracy outcomes of the SnapResNet architecture with the cutting-edge deep learning-based approaches for RSIC jobs. The overall accuracy of the proposed SnapResNet152 adopted from [31] and existing models on the LSCIDMR dataset is given in Table 4. The suggested architecture has outperformed all current optimal procedures, establishing a new benchmark with an OA of 97.25%. Precision, F1-score and recall values for different classes of dataset are mentioned in Table 5.

Table 4. Accuracy comparison between proposed and existing methods.

Methods	Accuracy
SnapResNet152	97.25
EfficientNet [5]	94.09
ResNet101 [5]	93.88
VGG19-Net [5]	93.19
AlexNet [5]	88.74

Table 5. Precision, recall and F1-score of 10 classes.

Classes	Precision	Recall	F1-Score
Desert	0.95	0.97	0.96
Extratropical Cyclone	0.96	0.96	0.96
Frontal Surface	0.99	1	1
High Ice Cloud	0.96	0.94	0.95
Low Water Cloud	0.98	1	0.99
Ocean	0.99	0.99	0.99
Snow	0.97	0.96	0.96
Tropical Cyclone	0.95	0.96	0.95
Vegetation	0.98	0.98	0.98
Westerly Jet	0.99	1	1

Figure 14 shows the cases where the proposed model SnapResNet152 fails to classify the images correctly with some examples of truly predicted images.

The following suggestions are put forth to outline the direction of future research considering this study:

- When working with deep networks, additional strategies, learning approaches, or combinations may be employed to enhance accuracy.
- Inter-class similarity in RSIC is still a problem that has to be addressed. To learn more about discriminative feature representations, necessary architectural and parametric adjustments may be investigated.

- This study was conducted in a single-GPU setting. Using multiple GPU configurations, the computational cost analysis may be carried out.
- In this experiment, a single labelled dataset is employed while disregarding the label-less class, and the suggested network may be expanded to field-level applications where a customized labelled dataset composed of images relevant to a certain application may be constructed.

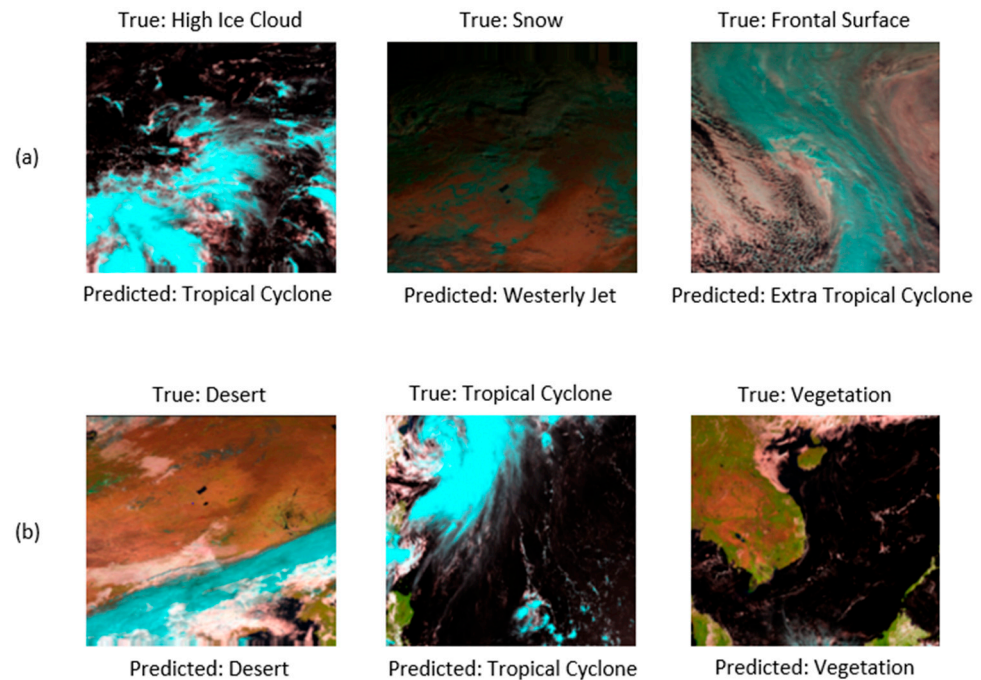


Figure 14. True and false predictions from SnapResNet152: (a) shows where true and predicted labels are different—false predictions; (b) shows where true and predicted labels are same—true predictions.

4.5. Discussion

It is worth noting that although the preceding study acquainted readers with the capability of the developed technique for classifying high-resolution satellite images under high inter-class similarity and class imbalance, the ensuing discussion will further provide readers with an insight into the algorithm with additional perspectives as follows:

- The methods compared in this research are contemporary image classification algorithms. It is observed that some methods only perform well on specific datasets; however, they cannot be generalized as they fail to perform on other datasets. For instance, in [1], the proposed large-scale cloud image database for meteorological research (LSCIDMR) deployed several deep learning methods (e.g., AlexNet, VGG-Net-19 ResNet-101 and EfficientNet) and the obtained results aided in establishing a baseline for future work. Moreover, the study presented in this research successfully classified images in the available dataset only.
- Authors in [20] suggested a novel lightweight data-driven weather forecasting model by researching temporal modelling approaches of LSTM and temporal convolutional networks. The deep model consists of two regressions: multi-input multi-output and multi-input single-output. When compared to the weather research forecasting (WRF) model, this model produces more accurate predictions up to 12 h. However, we also explore their classification ability on ten different classes of the LSCIDMR dataset. Also, in [21], authors presented a deep learning-based weather forecast system. Real-world datasets are employed in this approach for data volume and recency analysis. It is determined that having more data is beneficial to increasing model accuracy; however, data recency has no significant influence on model accuracy. Furthermore,

the classification accuracy of the proposed approach offers a thorough understanding of the satellite image classification in different classes. R. Meenal et al. [23] disclosed that by applying artificial intelligence-based technologies such as machine learning and deep learning, we might improve weather forecasting. Furthermore, comparing classification results with the proposed approach and numerous other techniques gives an immense amount of information on challenges in the satellite image classification area. Singh et al. [26] studied global data-driven precipitation models using a deep learning-based UNET with residual learning. This work demonstrates how a residual learning-based UNET may uncover physical links to target precipitation, and how such physical restrictions can be applied in dynamical operational models to enhance precipitation forecasting. Their findings open the door for the future development of online hybrid models. Finally, the extensive study and comparison presented in the preceding section suggest that this method may be modified further. Pengwei Du et al. [25] present an ensemble machine learning-based method (artificial neural network, support vector regression, Gaussian process) to forecast wind power production, which uses wind generation forecast made by numerical weather prediction (NWP) and meteorological observation data collected from weather stations. Our study, on the other hand, reports the application of this strategy to decrease overfitting in the result. We believe that the extensive analysis and comparisons offered in this publication will be useful to the research community in modifying any method for their specific requirements.

- Finally, the proposed study reports improved classification ability of the proposed method on high-resolution, imbalanced, and challenging inter-class similarity-based publicly available datasets. We expect that the analysis offered by the established approach and the extensive comparison described in this publication will provide the scientific community with further understanding. One of the objectives of the current study is to solve inter-class similarity and class imbalance in ten classes of the first publicly available satellite imagery-based dataset (LSCIDMR). Furthermore, to reduce the overfitting faced during the development of the proposed algorithm, we applied the ensembling method. The main reason to choose ResNet is its ability to utilize the residue of the residual block and learn more complicated features which in return improves classification ability.
- It has been observed that the proposed method still needs to perform well in different situations. Our experiment is conducted on 10 labelled classes; however, the dataset is composed of 11 classes. The 11th class has been excluded which is unlabeled as it was creating bias in classes, and the results were not satisfactory. In future research, the 11th class can be labelled as well to improve the overall process. For all the foregone discussion, the satellite images utilized in our experiments are of high resolution; however, their resolution can be improved further as Cong Bai et al. [1] utilized image compression before uploading the dataset. In addition, all the abovementioned work can be carried out on real-time satellite images.

5. Conclusions

In this study, a new SnapResNet is suggested to classify remote sensing satellite images. To fully utilize ResNets for RSIC, the suggested approach consists of a network (FC-1024) using ensembling and augmentation techniques. The overfitting problem is the most evident barrier to the deployment of deep nets when confronted with a lack of training data. Considering this, the suggested network was meticulously created through a series of tests to offer the greatest resistance to overfitting and consequently enhance the network's capacity for generalization. By modifying the architectural design as needed, the model also addressed the problem of inter-class similarity to learn more details and global attributes explicit to each class. The performance of the network was further improved by the introduction of a snapshot-based ensembling method, which utilized the SGD's ability to converge to unique minima on demand. Upon boosting and training on the

most difficult LSCIDMR satellite imaging dataset, the proposed SnapResNet yielded image classification results that were found competitive with state-of-the-art methods.

Author Contributions: Conceptualization, K.K., R.Y. and H.Z.U.R.; methodology, H.Z.U.R., K.K. and Z.M.; software, R.Y. and K.K.; validation, R.Y. and A.F.; formal analysis, Z.M.; investigation and resources, A.F. and Z.M.; data curation, H.Z.U.R. and K.K.; writing—original draft preparation, R.Y.; writing—review and editing, Z.H.K., S.M.Q. and A.J.S.; visualization, K.K., R.Y. and H.Z.U.R.; supervision, H.Z.U.R. and K.K.; project administration, H.Z.U.R. and K.K.; funding acquisition, Z.H.K., S.M.Q. and A.J.S. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the support received from King Fahd University of Petroleum & Minerals (KFUPM). Work of A.J.S. was supported in part by SDAIA-KFUPM Joint Research Center for Artificial Intelligence under Grant# JRC-AI-RFP-17.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: (<https://github.com/Zjut-MultimediaPlus/LSCIDMR>, accessed on 20 November 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bai, C.; Zhang, M.; Zhang, J.; Zheng, J.; Chen, S. LSCIDMR: Large-Scale Satellite Cloud Image Database for Meteorological Research. *IEEE Trans. Cybern.* **2021**, *52*, 12538–12550. [[CrossRef](#)]
2. Anaman, K.A.; Quaye, R.; Owusu-Brown, B. Benefits of Aviation Weather Services: A Review of International Literature. *Res. World Econ.* **2017**, *8*, 45–58. [[CrossRef](#)]
3. Kim, J.-H.; Lee, D.-B.; Kim, S.-H.; Strahan, M.; Pettegrew, B.; Gill, P.; Williams, P.D.; Schumann, U.; Tenenbaum, J.; Lee, Y.-G.; et al. Research Collaborations for Better Predictions of Aviation Weather Hazards. *Bull. Am. Meteorol. Soc.* **2017**, *98*, ES103–ES107. [[CrossRef](#)]
4. Gultepe, I.; Heymsfield, A.J.; Field, P.R.; Axisa, D. Ice-Phase Precipitation. *Meteorol. Monogr.* **2017**, *58*, 6.1–6.36. [[CrossRef](#)]
5. Bauer, P.; Thorpe, A.; Brunet, G. The quiet revolution of numerical weather prediction. *Nature* **2015**, *525*, 47–55. [[CrossRef](#)]
6. Reichstein, M.; Camps-Valls, G.; Stevens, B.; Jung, M.; Denzler, J.; Carvalhais, N.; Prabhat, F. Deep learning and process understanding for data-driven Earth system science. *Nature* **2019**, *566*, 195–204. [[CrossRef](#)]
7. Eraslan, G.; Avsec, Ž.; Gagneur, J.; Theis, F.J. Deep learning: New computational modelling techniques for genomics. *Nat. Rev. Genet.* **2019**, *20*, 389–403. [[CrossRef](#)]
8. Bai, C.; Huang, L.; Pan, X.; Zheng, J.; Chen, S. Optimization of deep convolutional neural network for large scale image retrieval. *Neurocomputing* **2018**, *303*, 60–67. [[CrossRef](#)]
9. Wu, W.; Yin, Y.; Wang, X.; Xu, D. Face Detection With Different Scales Based on Faster R-CNN. *IEEE Trans. Cybern.* **2019**, *49*, 4017–4028. [[CrossRef](#)] [[PubMed](#)]
10. Lin, D.; Zhang, R.; Ji, Y.; Li, P.; Huang, H. SCN: Switchable Context Network for Semantic Segmentation of RGB-D Images. *IEEE Trans. Cybern.* **2020**, *50*, 1120–1131. [[CrossRef](#)] [[PubMed](#)]
11. Han, J.; Chen, H.; Liu, N.; Yan, C.; Li, X. CNNs-Based RGB-D Saliency Detection via Cross-View Transfer and Multiview Fusion. *IEEE Trans. Cybern.* **2018**, *48*, 3171–3183. [[CrossRef](#)]
12. Cheng, G.; Li, Z.; Han, J.; Yao, X.; Guo, L. Exploring Hierarchical Convolutional Features for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6712–6722. [[CrossRef](#)]
13. Cheng, G.; Yang, C.; Yao, X.; Guo, L.; Han, J. When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 2811–2821. [[CrossRef](#)]
14. Huang, X.; Li, S.; Li, J.; Jia, X.; Li, J.; Zhu, X.X.; Benediktsson, J.A. A Multispectral and Multiangle 3-D Convolutional Neural Network for the Classification of ZY-3 Satellite Images Over Urban Areas. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 10266–10285. [[CrossRef](#)]
15. Cheng, G.; Li, Z.; Yao, X.; Guo, L.; Wei, Z. Remote Sensing Image Scene Classification Using Bag of Convolutional Features. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1735–1739. [[CrossRef](#)]
16. Li, H. Longevity of the CMS ECAL and Scintillator-Based Options for Electromagnetic Calorimetry at HL-LHC. *IEEE Trans. Nucl. Sci.* **2016**, *63*, 580–585. [[CrossRef](#)]
17. Sanabia, E.R.; Barrett, B.S.; Celone, N.P.; Cornelius, Z.D. Satellite and Aircraft Observations of the Eyewall Replacement Cycle in Typhoon Sinlaku (2008). *Mon. Weather Rev.* **2015**, *143*, 3406–3420. [[CrossRef](#)]
18. Marchuk, G. *Numerical Methods in Weather Prediction*; Elsevier Science: Amsterdam, The Netherlands, 2012; Available online: <https://books.google.com.pk/books?id=Z3jv9QSQJnQC> (accessed on 1 March 2023).
19. Ren, X.; Li, X.; Ren, K.; Song, J.; Xu, Z.; Deng, K.; Wang, X. Deep Learning-Based Weather Prediction: A Survey. *Big Data Res.* **2021**, *23*, 100178. [[CrossRef](#)]

20. Hewage, P.; Trovati, M.; Pereira, E.; Behera, A. Deep learning-based effective fine-grained weather forecasting model. *Pattern Anal. Appl.* **2021**, *24*, 343–366. [[CrossRef](#)]
21. Booz, J.; Yu, W.; Xu, G.; Griffith, D.; Golmie, N. A Deep Learning-Based Weather Forecast System for Data Volume and Recency Analysis. In Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019; pp. 697–701. [[CrossRef](#)]
22. Mohan, P.; Patil, K.K. Deep Learning Based Weighted SOM to Forecast Weather and Crop Prediction for Agriculture Application. *Int. J. Intell. Eng. Syst.* **2018**, *11*, 167–176. [[CrossRef](#)]
23. Meenal, R.; Binu, D.; Ramya, K.C.; Michael, P.A.; Vinoth Kumar, K.; Rajasekaran, E.; Sangeetha, B. Weather Forecasting for Renewable Energy System: A Review. *Arch. Comput. Methods Eng.* **2022**, *29*, 2875–2891. [[CrossRef](#)]
24. Utku, A.; Can, Ü. Deep Learning Based Effective Weather Prediction Model for Tunceli City. In Proceedings of the 2021 6th International Conference on Computer Science and Engineering (UBMK), Ankara, Turkey, 15–17 September 2021; pp. 56–60. [[CrossRef](#)]
25. Du, P. Ensemble Machine Learning-Based Wind Forecasting to Combine NWP Output with Data from Weather Station. *IEEE Trans. Sustain. Energy* **2019**, *10*, 2133–2141. [[CrossRef](#)]
26. Singh, M.; Kumar, B.; Rao, S.; Gill, S.S.; Nanjundiah, R.S.; Niyogi, D. Deep learning for improved global precipitation in numerical weather prediction systems. *arXiv* **2021**, arXiv:2106.12045.
27. Abdalla, A.M.; Ghaith, I.H.; Tamimi, A.A. Deep Learning Weather Forecasting Techniques: Literature Survey. In Proceedings of the 2021 International Conference on Information Technology (ICIT), Amman, Jordan, 14–15 July 2021; pp. 622–626. [[CrossRef](#)]
28. Sil, R.; Roy, A.; Bhushan, B.; Mazumdar, A.K. Artificial Intelligence and Machine Learning based Legal Application: The State-of-the-Art and Future Research Trends. In Proceedings of the 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 18–19 October 2019; pp. 57–62. [[CrossRef](#)]
29. Kang, L.-W.; Chou, K.-L.; Fu, R.-H. Deep Learning-Based Weather Image Recognition. In Proceedings of the 2018 International Symposium on Computer, Consumer and Control (IS3C), Greater Noida, India, 6–8 December 2018; pp. 384–387. [[CrossRef](#)]
30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
31. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
32. Awais, M.; Iqbal, M.T.B.; Bae, S.-H. Revisiting Internal Covariate Shift for Batch Normalization. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 5082–5092. [[CrossRef](#)]
33. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
34. Santurkar, S.; Tsipras, D.; Ilyas, A.; Madry, A. How Does Batch Normalization Help Optimization? *arXiv* **2019**, arXiv:1805.11604.
35. Dauphin, Y.N.; Cubuk, E.D. Deconstructing the Regularization of batchnorm. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
37. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
38. Li, X.; Chen, S.; Hu, X.; Yang, J. Understanding the Disharmony Between Dropout and Batch Normalization by Variance Shift. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 2677–2685. [[CrossRef](#)]
39. Krogh, A.; Hertz, J.A. A Simple Weight Decay Can Improve Generalization. *Adv. Neural Inf. Process. Syst.* **1992**, *4*, 950–957.
40. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807. [[CrossRef](#)]
41. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [[CrossRef](#)]
42. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random Erasing Data Augmentation. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 13001–13008. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.