
INTRODUCTION:

Weather classification is a task of classifying weather conditions by looking at the weather data. Accurate weather classification enables users to organize day to day activities for instance deciding what type of clothes to wear, planning outdoor travel or sports activity, solar technologies, etc. .

It plays a critical role in various other fields such as in agriculture, it helps farmers to decide which fertilizers or pesticides to use, whether to turn off the sprinklers etc. based on the outdoor weather conditions Accurate weather classification also helps in improving reliability of vehicle assistant driving and outdoor video surveillance systems .

Timely prediction of severe weather conditions helps saving people life and property. It helps to avoid road accidents, train derailment and ship collisions caused due to severe weather conditions like rain, fog, storm and snow .

It also assists in predicting natural calamities, thereby helping in saving people's life to a greater extent. Thus, timely and accurate classification of weather images is of utmost important to society and need.

It comprises multiple convolutional and pooling layers for better hierarchical representation and a Fully Connected Layer at the end for classification.

Numerous variants of CNN such as ResNet, Inception V3, Exception, VGG-16, and VGG-19, etc. have been developed by researchers in past for improving the accuracy of networks. Several features such as sky, shadow, reflection, haze, contrast, etc. are present in images which are used by the CNN for weather classification.

In this paper, a framework based on transfer learning techniques has been implemented for classifying the weather images into its appropriate category with the help of features extracted from pre-trained deep CNN models.

Automating the classification will not only save time and resources but also aids in increasing the reliability of the process. Section II highlights the work done by researchers in the field of weather image classification.

Limitation and challenges of existing technologies has been presented in Section III. Section IV defines the proposed framework. In Section V, the hardware and software required to implement the proposed framework have been discussed. The experimental results have been described in Section VI and the conclusion and future work has been presented in Section VII of the paper.

ABSTRUCT:

Classifying weather from outdoor images helps prevent road accidents, schedule outdoor activities, and improve the reliability of vehicle assistant driving and outdoor video surveillance systems. Weather classification has applications in various fields such as agriculture, aquaculture, transportation, tourism, etc.

Earlier, expensive sensors and huge manpower were used for weather classification making it very tedious and timeconsuming.

Automating the task of classifying weather conditions from images will save a huge time and resources. In this paper, a framework

based on the transfer learning technique has been proposed for classifying the weather images with the features learned from pre-trained deep CNN models in much lesser time. Further, the size of the training data affects the efficiency of the model.

The larger amount of high-quality data often leads to more accurate results. Hence, we have implemented the proposed framework using the spark platform making it scalable for big datasets. Extensive experiments have been performed on weather image dataset and the results proved that the proposed framework is reliable.

From the results, it can be concluded that weather classification with the InceptionV3 model and Logistic Regression classifier yields the best results with a maximum accuracy of 97.77%.

EXISTING SYSTEM ;

Many existing weather classification techniques make use of expensive sensors and huge manpower for classifying weather images. These methods rely on human observations, thus are more prone to errors and are very time-consuming.

Numerical Weather Prediction (NWP) models are also being widely used for forecasting the weather conditions, but the techniques are quite expensive and rely on the power of supercomputers for data processing. Recent work on weather classification includes classification of weather conditions from images.

CNN has an advantage of automatically detecting the important features from images without any human intervention. For getting the optimized results, CNN models require huge amount of processing power and large datasets for training.

Further, training weather images using CNN models also require tuning number of hyper-parameters such as number of convolutional and max-pooling layers, kernel size, regularization techniques, etc.

To overcome the above mentioned issues, transfer learning techniques are applied to transfer the knowledge learned from pre-trained deep CNN models to our weather image dataset. As transfer learning utilizes the features extracted from pre-trained CNN models, training time is considerably reduced. It too eliminates the need for tuning of number of hyper-parameters.

Transfer learning has proved its potential in number of applications such as natural language processing, sentiment classification, text classification, spam email detection, video classification, drug efficacy classification, etc. . Further, the size of weather images generated is also huge.

Existing weather classification techniques exploiting the transfer learning techniques do not make use of big data technologies as well as lacks scalability.

Thus, there is a need of weather classification framework which can scale on number of machines and can support big weather image datasets.

PROPOSED SYSTEM ;

To overcome the limitations discussed in previous section, a weather image classification framework has been proposed and implemented using spark platform. Describes the proposed framework which comprises of three steps: image loading, feature learning and multi-class classification.

Firstly, weather images are loaded into the spark via the readSchema() method of the ImageSchema library and a dataframe is created. Afterward, image data is given as input to the pre-trained deep CNN model to automatically learn hierarchical feature representation.

At this stage, either we can freeze all the layers of the pre-trained models or some of its layers are unfreeze and fine-tuned to learn the desired features . The final step is to use the learned features from the deep CNN models as input to the classifier model which classifies the outdoor weather image into its appropriate category.

In step 2, 5 deep CNN networks i.e. InceptionV3, Xception, ResNet50, VGG16, and VGG19 have been explored for feature learning and experiments have been conducted to determine the model which gave the highest accuracy.

VGG19 is 16 and 19 layered deep neural network whereas ResNet50 is 50 layered deep Convolutional neural networks. Inception V3 is also a very powerful neural network which is 42 layers deep and has an advantage of performing multiple different transformations.

Xception is a 71 layers deep neural network and incorporates depth-wise separable convolutions. All the layers from the pre-trained CNN models have been frozen in this case study.

In the last stage of proposed framework, one of the two classifiers i.e. Logistic Regression and Random Forests has been implemented for classification of weather images. Logistic Regression is a supervised ML technique which uses a sigmoid activation function for solving classification problems.

To reduce the over-fitting and generalization error, the Logistic Regression classifier is trained with elastic net regularization. Two hyper-parameters for Logistic Regression need to be optimized; alpha (α) which is used to assign the weight to both L1 and L2 penalty and lambda (λ), a regularization parameter that defines the balance between minimizing the training error and model complexity. If α is 0, the model follows L2 regularization while if α is 1, the model follows L1 regularization.

Random Forests is another popular supervised machine learning technique based on ensemble learning which combines the results of a large number of weak decision trees to form a robust classifier. As different combinations of the networks are possible and the same has been implemented using the Sparkdl library. Extensive experiments have been performed to determine the CNN model and classifier which provided the maximum accuracy for weather image classification.

EXPERIMENTAL SETUP:

To evaluate the efficiency of the proposed framework, Multi class weather image classification datasets freely available on Kaggle.com has been used . Dataset comprises 1125 images of four different categories i.e. cloudy, rainy, shiny, and sunrise. Sample images from each target class have been Categorization of target classes for the dataset has been presented .

Optimal tuning of spark parameters i.e. driver memory, driver cores, executor cores, executor memory, number of partitions etc. improves the performance of spark application to the greater extent [34,35,36,37].

As size of data is not very large, 4g memory is sufficient to store the data as well as to run the CNN models with Logistic Regression as classifier. While for Random Forest classifier, minimum 8g driver memory is required.

Number of cores allocated to driver/executor too impacts the application’s performance. If a cluster has n number of cores, number of partitions for input data should not be less than n as it might lead to under-utilization of resources.

Few cores will be left idle and resources would not be utilized efficiently. Thus, number of input partition should either be equal or greater than number of cores. In this research work, number of cores and partitions has been set to 2 and 4, respectively.

To evaluate the efficiency of proposed framework, following metrics have been used:

1) Accuracy: It can be classified as the ratio of truly predicted samples to the total number of predictions.

2) Precision: It can be defined as the ratio of truly predicted positive values to the total number of positive predictions.

3) Recall: It can be defined as the ratio of true positives and total positive samples.

4) F1 score: Calculated as the harmonic mean of precision and recall.

5) Training Time: Time required for training the model.

For Random Forest classifier, the number of trees in forests impact performance of classifier to a greater extent.

Thus, for different values of trees, its impact on the accuracy and run time of an algorithm has been studied and results are shown in Table III.It can be interpreted that till a certain value the accuracy value of the model increases with an increase in the number of trees in the forests. Afterwards, the accuracy value decreases.

Random Forest classifier provided the best accuracy value of 97.73% with 150 trees in the forests. It can also be observed that the increase in the number of trees has no significant impact on the training time of an algorithm.

Train and test data have been divided in the ratio 80:20. Table IV describes the accuracy, F1 score, precision, recall, and training time for 10 different possible architectures specified in.

It can be interpreted that with the InceptionV3 pre-trained model and Logistic Regression classifier, the accuracy value is maximum 97.77%. Factorized convolutions, grid size reduction, asymmetric convolutions and use of auxiliary classifiers are some the significant features of InceptionV3 model resulting in its higher accuracy value.

The performance of proposed framework has been compared with the existing techniques utilizing the same weather image dataset for classification.

the comparison of accuracy value of the proposed framework with the four existing approaches. Our proposed framework yields the maximum accuracy value of 97.77% when Inception V3 model and Logistic Regression classifier (InceptionV3+LR) has been used.

Models proposed by Oluwafemi and Zenghui (2019), Togacar et al. (2020), and Sharma and Ismail (2022) had lower accuracy values of 86%, 95.85% and 94% (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 14, No. 1, 2023 342 | Page www.ijacsa.thesai.org respectively [14,38,39]. Model proposed by Al-Haija et al. had the accuracy value of 98.22%, thus performed slightly better than the proposed InceptionV3+LR model in terms of accuracy .

Further, Model proposed by Al-Haija et al. had precision and sensitivity values of 96.5% and 96.4%, respectively while the proposed InceptionV3+LR model has better precision and sensitivity values i.e. 97.75% and 97.74% respectively.

Hence, InceptionV3+LR model has better ability to capture false negatives and true positives than existing techniques. Time taken by the proposed framework for training is also very less i.e. 6 min 42 sec.

CONCLUSION;

Weather Prediction is a very vital activity performed before numerous day-to-day activities. Deep CNN is a deep learning technique widely used to classify image datasets but the techniques require very huge datasets and high computing devices for achieving optimal results.

Transfer learning is proven to be effective in scenarios with smaller datasets and the unavailability of high computing devices. It helped in minimizing the learning time by utilizing the features learned from the previously trained deep CNN models.

In this paper, a framework based on transfer learning has been proposed for classifying the weather conditions from outdoor images and the performance of various CNN models has been compared for accuracy, precision, recall, f1 score, and training time.

From the results, it can be observed that with the InceptionV3 pre-trained model and Logistic Regression classifier, the accuracy value is maximum i.e. 97.77%.

Further, it can be concluded that the time taken by the framework for training the dataset with the Logistic Regression classifier is quite low compared to the Random Forests.

The comparison has also been done with the existing techniques and the results proved the efficiency of the proposed framework.

Further, the proposed framework is scalable and has support for big datasets. In the future, some layers from pre-trained CNN models can be un-freeze and fine-tuned to learn better hierarchical representation.

PROCESS;

Automated weather classification using transfer learning involves leveraging pre-trained models on large-scale image datasets and adapting them for weather classification tasks. Transfer learning allows you to utilize the knowledge learned from a source domain (e.g., general image recognition) and apply it to a target domain (e.g., weather classification) with limited labeled data.

Here's a general outline of how you can approach automated weather classification using transfer learning:

1. Dataset collection: Gather a diverse dataset of weather images containing different weather conditions such as sunny, cloudy, rainy, foggy, etc. The dataset should be labeled with corresponding weather classes.

2. Pre-trained model selection: Choose a pre-trained convolutional neural network (CNN) model that has been trained on a large-scale dataset, such as ImageNet. Models like VGG16, ResNet, or Inception are commonly used for transfer learning.

3. Transfer learning: Initialize the selected pre-trained model with its learned weights from the source domain. Remove the last classification layer(s) of the model, which were specific to the source dataset. This layer will be replaced with new layers for weather classification.

4. Model adaptation: Add new fully connected layers on top of the pre-trained model to learn weather-specific features. These layers will learn to map the extracted features from the pre-trained layers to the weather classes in your dataset. Ensure that the new layers have the appropriate number of nodes for your specific weather classification task.

5. Fine-tuning: Freeze the pre-trained layers and train only the newly added layers using the labeled weather dataset. Gradually unfreeze and fine-tune some of the pre-trained layers to adapt to the weather-specific features in your

dataset if necessary. This process helps the model learn the specific patterns and characteristics of different weather conditions.

6. Evaluation and optimization: Split your dataset into training and validation sets. Train the model using the training set and evaluate its performance on the validation set. Monitor metrics such as accuracy, precision, recall, and F1 score to assess the model's performance. Make adjustments to the model architecture, hyperparameters, or training process as needed to improve performance.

7. Testing and deployment: Once you're satisfied with the model's performance on the validation set, evaluate it on a separate test set to get an unbiased estimate of its performance. Finally, deploy the trained model to classify weather conditions in real-world scenarios.

Remember to preprocess your weather images, including resizing, normalization, and augmentation techniques, to ensure the model receives consistent and meaningful input.

Note that the success of automated weather classification using transfer learning heavily relies on the availability and quality of labeled weather datasets. It's important to have a diverse and representative dataset to ensure accurate classification across different weather conditions.

Automated weather classification using transfer learning tools can be a powerful approach to leverage pre-trained models for accurate weather prediction. Transfer learning allows you to take advantage of the knowledge learned from one domain (e.g., general image classification) and apply it to a different domain (e.g., weather classification).

Here's a step-by-step guide on how you can utilize transfer learning tools for automated weather classification:

1. Gather a weather dataset: Start by collecting a diverse dataset of weather images or weather-related data. This dataset should cover different weather conditions such as sunny, cloudy, rainy, foggy, snowy, etc. The more diverse and representative your dataset is, the better your model will perform.

2. Preprocess and augment the data: Preprocessing steps may include resizing the images, normalizing pixel values, and splitting the dataset into training and testing sets. Additionally, data augmentation techniques like rotation,

flipping, and zooming can be applied to increase the robustness of the model.

3. Choose a pre-trained model: Select a pre-trained convolutional neural network (CNN) model that has been trained on a large-scale dataset, such as ImageNet. Models like VGG, ResNet, Inception, or MobileNet are popular choices. These models have learned general image features that can be transferred to weather classification tasks.

4. Customize the model: Remove the last fully connected layer of the pre-trained model and replace it with a new layer suitable for weather classification. The new layer should have the desired number of output classes (e.g., sunny, cloudy, rainy). This new layer will be trained from scratch, while the rest of the pre-trained model's weights will remain frozen.

5. Train the model: Initialize the new layer's weights randomly and train the model using the training dataset. Use an appropriate optimization algorithm, such as stochastic gradient descent (SGD) or Adam, and define an appropriate loss function, such as categorical cross-entropy. Monitor the model's performance on the validation set and adjust hyperparameters if necessary.

6. Fine-tuning (optional): If your dataset is large enough, you can perform fine-tuning by unfreezing some of the pre-trained layers and allowing them to be updated during training. This step can help the model learn more specific features related to weather conditions.

7. Evaluate the model: Once training is complete, evaluate the model's performance on the testing dataset. Calculate metrics like accuracy, precision, recall, and F1 score to assess the model's effectiveness in weather classification.

8. Deploy and use the model: Once satisfied with the model's performance, you can deploy it for real-time weather classification. Provide new weather images to the deployed model, and it will predict the corresponding weather condition based on its training.

Remember to regularly update and retrain the model as new weather patterns or conditions emerge to ensure its accuracy and relevance.

By following these steps, you can leverage transfer learning tools to automate weather classification and enhance the accuracy and efficiency of weather prediction systems.

ADVANTAGE;

Automated weather classification using transfer learning offers several advantages over traditional methods. Here are some key benefits:

1. *Improved Performance*: Transfer learning leverages pre-trained models that have been trained on large-scale datasets, typically on a diverse range of tasks. By starting with a pre-trained model, the network has already learned general patterns and features that can be useful for weather classification. This allows the model to achieve higher accuracy and better generalization compared to training from scratch.

2. *Reduced Training Time and Data Requirements*: Training deep learning models from scratch requires large amounts of labeled data and extensive computational resources. However, transfer learning allows you to use a pre-trained model as a starting point, significantly reducing the training time and the amount of labeled data needed. You can fine-tune the pre-trained model on your specific weather

classification task with a smaller labeled dataset, which is often more feasible to obtain.

3. *Domain Adaptation*: Weather patterns and characteristics can vary across different regions and time periods. Transfer learning enables the model to adapt to specific weather conditions and improve its performance on the target domain. By fine-tuning a pre-trained model on data from the target domain, the model can learn domain-specific features and achieve better classification results.

4. *Ability to Handle Limited Data*: Weather datasets can be limited, especially for rare events or specific weather phenomena. Transfer learning can mitigate the data scarcity problem by leveraging knowledge from larger datasets. The pre-trained model has already learned generic weather-related features, and fine-tuning on a smaller dataset can help the model specialize and adapt to the specific weather classes of interest.

5. *Transferable Knowledge*: Transfer learning allows the model to transfer knowledge learned from one weather classification task to another. For example, if a pre-trained model has been trained on a broad weather classification task, such as cloud classification or precipitation detection, it

can be fine-tuned on a narrower task, such as fog detection or thunderstorm classification. This transfer of knowledge can expedite the development of accurate models for specific weather classification tasks.

Overall, automated weather classification using transfer learning offers improved performance, reduced data requirements, domain adaptation capabilities, and the ability to handle limited data. These advantages make transfer learning an effective approach for developing accurate and robust weather classification models.

DISADVATAGE;

Transfer learning is a technique widely used in machine learning, including for tasks like automated weather classification. However, it is important to consider the potential disadvantages of using transfer learning in this context:

1. Limited flexibility: Transfer learning relies on pre-trained models that were trained on a specific dataset and task. While this can be advantageous when the pre-trained model's features are relevant to the target task, it can also limit the flexibility of the model. The pre-trained model might not capture all the nuances and intricacies of the weather classification task, leading to suboptimal performance.

2. Domain shift: Weather data can vary significantly across different regions, seasons, and atmospheric conditions. The pre-trained model may have been trained on a different dataset or weather conditions, which can introduce a domain shift. This domain shift can affect the model's ability to generalize well to new weather data, leading to reduced performance.

3. Limited customization: Transfer learning typically involves fine-tuning the pre-trained model on a target dataset. However, the extent to which the model can be customized is limited by the architecture and features of the pre-trained model. This limitation can hinder the ability to capture specific weather patterns or phenomena that are crucial for accurate classification.

4. Bias from pre-training data: The pre-trained model used for transfer learning might have been trained on a dataset that has inherent biases. These biases can carry over to the transferred model and influence its predictions. In the case of weather classification, biased data can lead to inaccurate predictions, especially in underrepresented or rare weather events.

5. Increased complexity: Transfer learning involves combining a pre-trained model with additional layers or modules to adapt it to the target task. This can increase the complexity of the overall model architecture, making it more challenging to train, deploy, and interpret. It may also require more computational resources and time for training and inference.

6. Dependency on pre-trained models: Transfer learning relies on the availability of suitable pre-trained models. If there is a lack of relevant pre-trained models for weather classification or if the existing ones are not sufficiently effective, it can limit the usefulness of transfer learning for this specific task.

While transfer learning can provide valuable starting points for automated weather classification, it is crucial to carefully evaluate these disadvantages and consider whether the benefits outweigh the limitations in a given context. Exploring alternative approaches, such as training models from scratch or using specialized weather models, may be necessary in some cases.

BLOCK DIAGRAM;

Certainly! Here's a simplified block diagram explaining the concept of automated weather classification using transfer learning:

```
              +---------------------+
Input Image -->|    Preprocessing    |
              +---------------------+
                         |
                         V
              +---------------------+
          +--> |    Feature Extraction  |
          |    +---------------------+
          |              |
          |              V
          |    +---------------------+
          +--> |    Transfer Learning  |
          |    +--------------------+
          |              |
          |              V
          |    +--------------------+
          +--> |   Classification   |
              +--------------------+
                         |
                         V
                    Output Class Label
```

Let's go through each block in detail:

1. *Preprocessing*: The input image, typically obtained from a weather satellite or weather station, goes through a preprocessing step. This step may include operations like resizing, normalization, noise removal, and image

enhancement to prepare the image for further processing.

2. *Feature Extraction*: In this block, a feature extraction algorithm is applied to the preprocessed image. This step aims to identify and extract meaningful features from the image that are relevant for weather classification. Various techniques can be used, such as convolutional neural networks (CNNs) or handcrafted feature extraction methods.

3. *Transfer Learning*: Transfer learning is a technique where a pre-trained model, which has been trained on a large dataset for a different task, is utilized as a starting point for a new task. In this block, a pre-trained model, such as a CNN trained on a large-scale image dataset (e.g., ImageNet), is used as a feature extractor. The weights of the pre-trained model are frozen, and only the final layers of the model are retrained on

the specific weather classification dataset. This allows the model to leverage the knowledge learned from the large dataset and adapt it to the weather classification task.

4. *Classification*: The output of the transfer learning block is fed into a classification algorithm. This algorithm, often implemented using fully connected layers or other classifiers, maps the extracted features to specific weather classes or labels. The model is trained on a labeled dataset containing images and their corresponding weather classes. During inference, the trained model predicts the weather class based on the input image.

5. *Output*: The final output of the system is the predicted weather class or label corresponding to the input image. It could be weather conditions like sunny, cloudy, rainy, or other specific weather phenomena.

This block diagram provides a high-level overview of the automated weather classification using transfer learning approach. It demonstrates how the combination of preprocessing, feature extraction, transfer learning, and classification enables the system to classify weather conditions accurately based on input images.

RESULT;

Automated weather classification using transfer learning can be a useful approach to leverage pre-trained models and adapt them to classify weather conditions. Transfer learning involves using a pre-trained neural network model, typically trained on a large dataset, and fine-tuning it on a smaller target dataset related to weather conditions.

To provide a more specific result, we would need additional information about the data you have and the type of weather conditions you want to classify. However, I can give you a general outline of the steps involved in using transfer learning for weather classification:

1. Dataset collection: Gather a dataset of weather-related images or other relevant data. This dataset should be labeled with the corresponding weather conditions you want to classify (e.g., sunny, cloudy, rainy, etc.).

2. Pre-processing: Clean and preprocess the collected data. This step may involve resizing the images, normalizing pixel values, and splitting the data into training and testing sets.

3. Transfer learning: Select a pre-trained model that has been trained on a large-scale dataset, such as ImageNet. Popular pre-trained models include VGG, ResNet, and Inception. Remove the

original classification layer(s) of the pre-trained model.

4. Fine-tuning: Add a new set of fully connected layers on top of the pre-trained model to match the number of weather classes you want to classify. The weights of the pre-trained layers can be frozen to preserve their learned features, while the newly added layers can be trained using the target dataset.

5. Training: Train the modified model on your weather dataset. Adjust the hyperparameters, such as learning rate, batch size, and number of epochs, to optimize the training process. Monitor the training progress by evaluating the model's performance on the validation set.

6. Evaluation: Once training is complete, evaluate the model's performance on the testing set.

Calculate metrics such as accuracy, precision, recall, and F1 score to assess the classification performance.

7. Deployment: After satisfactory evaluation results, the model can be deployed to classify weather conditions in real-time. It can be integrated into an application, website, or any other system where weather classification is required.

Remember that the success of transfer learning depends on the availability and quality of the dataset, as well as the suitability of the pre-trained model for the specific task. It's essential to ensure the dataset is diverse and representative of the weather conditions you want to classify.

GITHUB LINK

https://github.com/shalini12345m/shalini