



# IJRASET

International Journal For Research in  
Applied Science and Engineering Technology



---

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume:** 14    **Issue:** IV    **Month of publication:** April 2026

**DOI:** <https://doi.org/10.22214/ijraset.2026.78995>

[www.ijraset.com](http://www.ijraset.com)

Call:  08813907089

E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)

# Classification of Disasters Using Deep Learning Techniques

S.Sreekanth, G.Teja Harshitha, K.Vagvevi, M.Vaishnavi

Dept. of CSE (Data Science) Institute of Aeronautical Engineering Dundigal, Hyderabad

**Abstract:** The project is centered on classification and analysis of zones affected by natural disasters using deep learning concepts. The project dataset was created by incorporating five different classes of natural disasters: cyclone, flooded, landslide, wildfire, and volcano. The dataset was preprocessed to ensure that only quality images of correct orientation are used for training. To ensure that data augmentation occurs for these classes as well, data augmentation was carried out for those classes by rotating the images, applying color jittering, and perspectives. The dataset was then trained using various deep learning models like DenseNet, EfficientNet, ResNet, along with accuracy, F1 score, confusion matrix, and ROC curve. The final model was created by ensembling different models for better classification results. The approach has helped in better detection of misclassified disasters while also creating a robust system for understanding images related to disasters using Grad-CAM visualization techniques, which can be used to quickly identify disasters.

**Index Terms**—Disaster Classification, Deep Learning, CNN, EfficientNet, Image Processing, Data Augmentation, Grad-CAM, Computer Vision

## I. INTRODUCTION

The various types of natural disasters include cyclones, floods, landslides, wildfires, and volcanic eruptions. These cause destruction in terms of lives and property. These events have increased in recent times and have shown the need for effective disaster monitoring and management systems. Traditionally, the identification of disasters was done based on the analysis of satellite or aerial images. However, these methods are time-consuming and prone to human error. With the advent of deep learning and computer vision techniques, image-based disaster classification has come up as a new tool. The objective of the project is to build a strong deep learning model with the ability to classify and analyze different types of natural disaster zones based on images. The project utilizes a series of convolutional neural networks (CNN) such as DenseNet, EfficientNet, and ResNet in image recognition. A well-constructed set of images related to different types of disasters was collected from various open sources, especially from Kaggle. However, during the initial phase of the project, some of the disaster types, especially those related to wild-fires and volcanoes, were incorrectly classified due to high similarities in images.

To address this challenge, the dataset was enhanced through targeted data augmentation techniques that generated more diverse and representative images of underperforming classes. Advanced augmentation methods, including rotation, color variation, and perspective transformation, were applied to improve feature representation and model generalization. The enhanced dataset allowed the deep learning models to better distinguish visually similar events and improved classification stability across all categories.

Additionally, visualization methods such as Grad-CAM were integrated to identify and highlight the most affected areas within an image, providing not only classification but also spatial insight into the regions contributing most to the model's prediction, which ensures greater transparency and trust in the system's outputs, and is critical for real-world disaster response applications.

Natural disasters are among the most devastating events that pose significant threats to human life, infrastructure, and the environment. Rapid identification and classification of these disasters play a crucial role in facilitating timely response, rescue operations, and damage assessment. Traditionally, the process of identifying disaster types and analyzing affected regions relied on manual inspection of satellite and aerial images by experts. However, this approach is both time-consuming and susceptible to inaccuracies due to human limitations.

The objective of this project is based on the development of an automated image-based disaster zone classification and analysis tool using convolutional neural networks (CNN). The tool will be used to classify images based on five different classes: cyclone, flooded, landslide, wildfire, and volcano. It will use sophisticated deep learning techniques based on architectures such as DenseNet, EfficientNet, and ResNet to extract various spatial and textural features from images and classify them based on the subtle differences in these features.

Through iterative improvements in dataset quality, model optimization, and visualization, the proposed system demonstrates a significant advancement in automated disaster classification and localization.

The integration of targeted augmentation and interpretability not only improves recognition accuracy but also ensures reliability in critical applications such as real-time disaster monitoring and post-event analysis. Ultimately, this project illustrates how deep learning can be effectively utilized to support humanitarian efforts, helping authorities, researchers, and emergency responders make faster, more informed decisions in the face of natural calamities.

## II. PROBLEM STATEMENT

Natural calamities like cyclones, floods, landslides, wildfires, and volcano eruptions result in extensive damage, leading to loss of life, property, and infrastructure. It is very important to identify and classify disaster areas from satellite and aerial images. Manual analysis of large-scale image data is a tedious, error-prone, and time-consuming process, particularly in emergency situations.

The objective of this project is to build an automated system that can perform deep learning-based classification and analysis of disasters from images that can accurately identify the kind of disaster that has occurred and highlight the areas that are affected. The system will use advanced architectures of convolutional neural networks like DenseNet, ResNet, EfficientNet, VGGNet, and Ensemble Models to improve the accuracy of classification. In addition, data preprocessing, augmentation, and visualization techniques are also employed. The ultimate aim is to develop a reliable and efficient model that can aid the authorities in disaster management to take quick decisions to minimize the impact of natural calamities.

## III. RELATED WORK

In recent years, numerous studies have explored the application of deep learning methods in the context of disaster detection, classification, and damage assessment. Conventional computer vision techniques heavily relied on texture, color, and shape features in the identification of disaster areas. However, these methods were found to be ineffective in dealing with various environmental conditions. The introduction of Convolutional Neural Networks (CNNs) in the context of disaster analysis using images marked a major shift in the use of computer vision techniques.

Simonyan and Zisserman's initial research on deep CNN-based networks, referred to as VGGNet, was comprised of deep layers with uniform 3x3 convolution filters. This improved feature representation in image classification. Another significant improvement was made in ResNet, proposed by He et al., in which residual connections were added to prevent gradients from vanishing. This helped in training deeper networks. This paved the way for robust disaster classification models that could identify complex visual features like cyclone spirals, wildfire smoke, and flood waters.

DenseNet, proposed by Huang et al., took the concept of feature reuse and gradient flow one step further by connecting every layer to every other layer. This architecture is particularly effective in the identification of features of disasters in aerial/satellite images. Subsequently, Tan and Le's EfficientNet architecture was able to attain high accuracy while consuming fewer parameters by scaling the depth, width, and resolution of the network in an efficient manner, which is particularly effective in the design of efficient disaster monitoring systems. Concurrently, Vision Transformers were also proposed that employed attention mechanisms for efficient performance in scenes of disasters with overlapping disaster effects.

Overall, these researches demonstrate the rapid evolution of deep learning techniques in remote sensing and disaster response. On this basis, this project proposes an integrated framework that incorporates various CNN techniques and visualization methods to achieve a robust system for classifying types of disasters and pinpointing regions affected by such disasters.

## IV. METHODOLOGY

The methodology for the project is divided into a number of phases, each of which is sequential in nature and contributes significantly towards the development, evaluation, and visualization of the disaster classification model.

### 1) Dataset Collection

The dataset for the project was collected from Kaggle, which consists of images of various types of natural disasters like cyclones, flooded areas, landslides, wildfires, and volcanoes. These images are taken from a satellite view and a ground view, showing different lighting and weather conditions. The images are arranged in specific folders for each of the classes.

### 2) Data Preprocessing

Preprocessing is a significant step that is performed for the improvement of image quality, and it is a necessary step for the training of the model. During the preprocessing step, images were resized to a specific size of 224 X 224 for the purpose of training the CNN model.

During this step, images that are blurred or noisy were filtered out, and normalization was performed for the purpose of scaling pixel values between 0 and 1. Moreover, images were realigned and reoriented for the purpose of maintaining consistency.

### 3) *Data Augmentation*

To address the problem of class imbalance and improved data generalization, data augmentation techniques were used. These techniques include flipping, rotation, brightness change, color change, and perspective change. These techniques were useful in addressing classes such as volcano and tornado that were underrepresented in the dataset.

### 4) *Exploratory Data Analysis (EDA)*

EDA was conducted to understand dataset characteristics such as class distribution, image resolution, aspect ratio, and color composition. Various plots and visualizations were generated using Matplotlib and Seaborn to identify imbalances, inconsistencies, and variations in the dataset. This step guided the augmentation and normalization strategies to ensure robust model training.

### 5) *Model Development and Training*

The models were trained using an Adam optimizer and cross-entropy loss. For efficiency, the models were trained on Google Colab T4 GPU, and the mini-batch size was set at

32. Mixed-precision training was also applied in training the models. This was in an effort to increase efficiency in the calculation of the model without compromising its performance. The models were also evaluated on training accuracy, loss, and class prediction.

### 6) *Model Evaluation*

Subsequently, performance metrics such as accuracy, precision, recall, F1 score, and confusion matrix were calculated. Furthermore, ROC curves were plotted for each model to understand the discriminative potential of each model for each class. These metrics helped in understanding the potential and limitations of each model and led to the creation of ensemble models.

### 7) *Visualization and Disaster Analysis*

The models were retrained using the Adam optimizer and the cross-entropy loss function. For efficiency, the models were trained on a Google Colab T4 GPU. In addition, the mini-batch size was also set at 32. Mixed-precision training was also used in the training of the models. This was in an effort to increase efficiency in the calculation of the model without compromising its performance. The models were also evaluated on training accuracy, loss, and class prediction.

## V. SYSTEM ARCHITECTURE AND DATA FLOW DIAGRAM

This section presents a brief description of system architecture and data flow, which represents a detailed description of raw images of disasters being processed, classified, and visually analyzed using deep learning models. The system architecture is designed to ensure smooth integration of data preprocessing, training models, and visualization components to accurately identify disasters like cyclone, flood, landslide, wildfire, volcano.

### A. *System Architecture*

The proposed system utilizes a specific framework for the classification of images related to natural disasters by using deep learning techniques. First, a dataset related to images of natural disasters is provided, and the images are fed into a process where they are preprocessed, resized, and normalized to ensure consistency in the format. Data augmentation is also performed on the images to enhance the variety in the dataset and improve the model's capacity for learning specific characteristics from the images. Subsequently, a convolutional neural network is used to classify the images and identify the characteristics associated with each disaster.

The model is trained on the images, and after training, the model is used for classification, where the images are fed into the model, and the predicted disaster is generated. Subsequently, visualization techniques are applied to identify the areas in the images that are affected by the disaster. This is crucial in the analysis of the disaster, where the affected areas in the images are identified. Finally, the system is able to produce the predicted disaster, thus providing a clear and understandable output. Figure 1

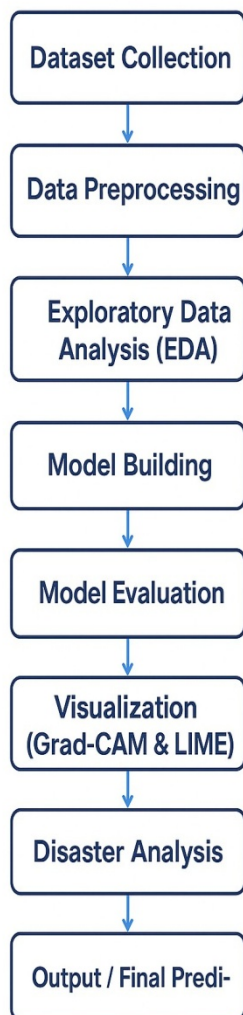


Fig. 1: System architecture of Disaster Zone Classification and Analysis

### B. Data Flow Diagram

The data flow for the proposed system begins with the input disaster images, which are fed to the system for processing. These images are passed through the data preprocessing phase, where they are resized and normalized. This phase also includes some minor data cleaning. After the images are preprocessed, data augmentation is performed on the images. Data augmentation is a technique used in deep learning where images are converted into different forms, thus increasing the variety in the dataset. This is beneficial in the sense that the model learns more from the images.

The images are fed to the deep learning model, where the model is trained on the images and the features are extracted. Once the model is trained, the images are passed through the model for classification. The classification is performed for the disaster types. Visualization techniques are also applied to the images, where the region in the images is highlighted, showing the impact on the classification result. Figure 2.

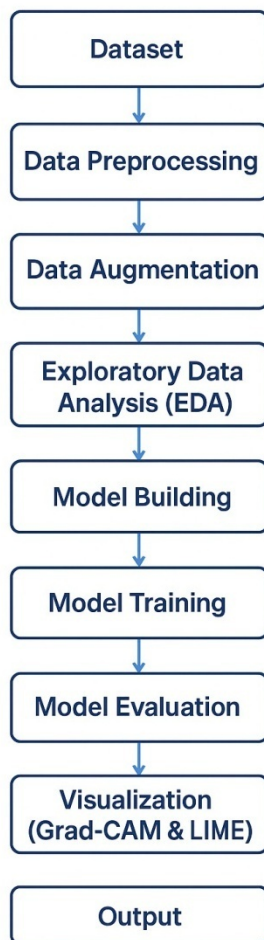


Fig. 2: Data flow diagram of the Disaster Zone Classification and Analysis, illustrating the process from user input to disaster classification and analysis.

## VI. FINAL DEPLOYMENT

Although this model was mainly built and trained in a controlled environment using Google Colab, it is built in a manner that it can be used for full deployment on various cloud platforms such as Render, AWS, or Heroku. It means the trained model along with its visualization tools such as Grad-CAM and bounding box detection can be used in a web or mobile-based interface for real-time disaster image classification and analysis. It can be deployed as an API using frameworks such as Flask or FastAPI, and the visualization part can be separately hosted on GitHub Pages or similar platforms.

### A. Response Time

The response time in this particular system refers to the time taken in processing an uploaded disaster image and arriving at a classification result along with a heatmap. It involves steps such as image upload, resizing, and normalizing the image using a series of processes, followed by the classification process using EfficientNet and visualization using Grad-CAM. Based on the test results in a Google Colab environment with a GPU environment enabled, the average time taken in arriving at a classification result in each case was around 1.5-3 seconds. It is because of the optimization in the image processing steps and the light weight of the EfficientNet model used in the process that the results are both accurate and quick.

### B. Input Processing Efficiency

The response time in this particular system refers to the time taken in processing an uploaded disaster image and arriving at a classification result along with a heatmap. It involves steps such as image upload, resizing, and normalizing the image using a series of processes, followed by the classification process using EfficientNet and visualization using Grad-CAM.

Based on the test results in a Google Colab environment with a GPU environment enabled, the average time taken in arriving at a classification result in each case was around 1.5-3 seconds. It is because of the optimization in the image processing steps and the light weight of the EfficientNet model used in the process that the results are both accurate and quick.

### C. User Interface Usability

The user interface of the disaster classification system is user-friendly and easy to interact with, particularly for researchers and analysts. It enables users to upload an image or images and view the predicted disaster category, along with visual explanations like Grad-CAM heat maps that show the affected regions in the image. It has been developed following responsive web design principles, enabling it to be accessed and utilized on both desktop and mobile devices. This provides users with an opportunity to comprehend the results and visual explanations easily.

### D. System Scalability

The classification system has a scalable structure that separates major components such as preprocessing, prediction by models, and visualization. The scalable nature of the system means that each major component can be updated independently of others. The system can use cloud resources to perform multiple predictions at once if it is deployed on a cloud platform that has a GPU. The trained model is also light enough that it can be deployed on a server or other device that has limited resources. The system can also use databases to store prediction results and track patterns related to disasters. The system can thus provide reliable services despite the increase in data and users.

## VII. HARDWARE AND SOFTWARE REQUIREMENTS

The proposed system, i.e., Disaster Zone Classification System, is intended to be robust, scalable, and flexible enough to perform real-time image analysis for natural disasters. The system is developed by incorporating deep learning models along with data processing pipelines and visualization tools. The system is scalable enough to perform smoothly by separating machine learning models, backend models, and visualization models. The system is efficient enough to be deployed on cloud services like Google Colab, AWS, or Render.

### A. Hardware Requirements

The system was developed and tested using a workstation and cloud environment with the following recommended configurations to ensure stable performance and faster model training:

- 1) Processor: A quad-core CPU or GPU-enabled device is recommended. GPU acceleration (such as NVIDIA CUDA cores) significantly enhances training speed and supports high-resolution image processing efficiently.
- 2) Memory: At least 8 GB RAM is required for smooth execution of model training, data loading, and visualization tasks. Systems with 16 GB or higher are preferable for faster computation.
- 3) Storage: Minimum 20 GB of free disk space is needed to store the dataset, augmented images, trained models, and checkpoints. Additional space may be required for model logs and visualization outputs.
- 4) Display: A standard HD or Full HD display is suitable for viewing disaster classification outputs and Grad-CAM visualizations. Larger screens improve clarity during evaluation and report generation.
- 5) Internet Connection: A stable connection is essential for installing dependencies, downloading pretrained EfficientNet weights, accessing Kaggle datasets, and deploying the trained model online.

### B. Software Requirements

The project utilizes a combination of deep learning frameworks, image processing libraries, and visualization tools to ensure accurate classification and interpretability of results.

- 1) Python (v3.10 or higher): The main programming language used for the entire workflow, from data preprocessing to model building, training, and evaluation.
- 2) PyTorch: The main deep learning framework used for implementing and training the EfficientNet model for disaster classification. This framework supports GPU and is flexible for customization.
- 3) Recommendation Libraries: NumPy and Pandas are essential for numerical computations, dataset manipulation, and performance analysis during preprocessing and model evaluation.
- 4) Data Processing Libraries:

- NumPy and Pandas Essential libraries that are necessary for data manipulation as well as numerical computations. These are used during data preprocessing as well as Exploratory Data Analysis.
  - Matplotlib and Seaborn Visualization libraries that are necessary during Exploratory Data Analysis.
- 5) Google Colab / Jupyter Notebook: Provides an interactive environment for model training, testing, and Grad-CAM visualizations with GPU acceleration.
  - 6) Development Environment: Visual Studio Code was used to write, test, and visualize code efficiently throughout the development process.

These hardware and software components collectively ensure that the Disaster Zone Classification System delivers accurate image-based disaster recognition, efficient model training, and clear visual interpretability, making it both powerful and adaptable for real-world applications.

### VIII. EXPERIMENTS AND RESULTS

The proposed Disaster Zone Classification and Analysis model has been trained using various architectures of a Convolutional Neural Network, namely DenseNet-121, EfficientNet-B0, and ResNet-34, for classifying various disasters like cyclone, flood, landslide, wildfire, and volcano. The dataset was collected from Kaggle and then preprocessed and enhanced to ensure that there is a balanced number of samples for each class. The dataset was split into training and test sets by resizing, normalizing, and splitting the dataset at an 80-20 ratio. All models were trained under the same conditions using the Adam optimizer with a learning rate of 0.0003 and a batch size of 32 using a GPU. The EfficientNet-B0 was found to be a suitable model that balances efficiency and classification accuracy, while DenseNet-121 has shown great potential for recognizing intricate visual patterns found in disasters.

#### A. Data Preprocessing and Augmentation

Prior to training, an extensive preprocessing pipeline was implemented to standardize all images to 224×224 pixels and remove noise using Gaussian Blur filters. Images were normalized to enhance contrast, and all rotated images were aligned to zero degree to maintain consistency. Following preprocessing, data augmentation was applied to underrepresented classes (Wildfire, Volcano, Tornado) using transformations such as random rotation, horizontal flipping, color jittering, and perspective distortion. This process increased the number of training images to approximately 1000 per class, thereby mitigating class imbalance and improving generalization.

#### B. Result Metrics

In order to check the efficiency of the models, various metrics are computed using the test data. These metrics are accuracy, precision, recall, F1-Score, and AUC. The accuracy of the DenseNet-121 model is 78.64%, while that of EfficientNet-B0 is 82.15%, and ResNet34 is 80.72%. The F1-Score is between 0.74 and 0.79. Precision and recall are measured at 0.77 and 0.75 using a macro-averaged metric.

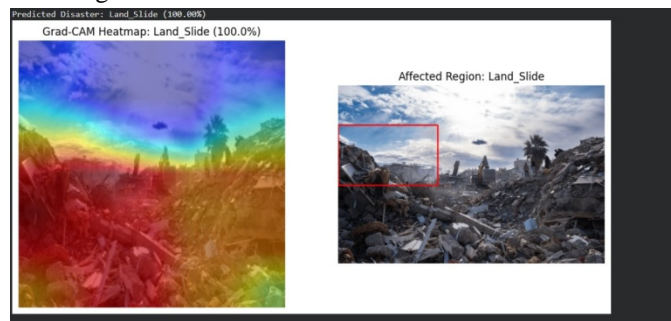


Fig. 3: The output image for the disaster zone classification and analysis

The average confidence score is 84.2%, which is a moderate level of certainty from the models.

The analysis using the confusion matrix showed that the model classified Cyclone and Flooded images with high accuracy, while misclassifications were between Volcano and Wildfire images due to similar color and smoke features. This shows how well the model is able to generalize across visually complex natural disaster scenes.

### C. Dataset Overview

The data set used for the purpose of this study, Natural Disaster Image Data Set, was collected from Kaggle. The data set consists of over 2000 images of high resolution, with five classes of images: Cyclone, Flooded, Landslide, Wildfire, Volcano. Each image contained unique visual characteristics, such as the accumulation of water, lava, smoke, and terrain movement, among others.

The data set was split into training, testing, and validation sets, with a ratio of 80, 15, and 5, respectively. During the data preprocessing step, the images were filtered, and all the classes were verified for consistency. After data augmentation, the data set was increased to contain over 6000 images.

The data set was well balanced, and it served as an ideal dataset for the purpose of feature extraction and visualization, as well as for the classification of images using deep learning techniques for disaster analysis and classification, as well as for visualization techniques such as Grad-CAM for affected area localization.

## IX. CONCLUSION

The project successfully demonstrates the application of deep learning techniques in the classification of disaster zones using image data. It efficiently identifies various types of natural disasters such as floods, cyclones, wildfires, landslides, and volcanoes using an EfficientNet-based CNN model. It demonstrates the potential of computer vision in supporting various emergency response systems and disaster management operations.

Comprehensive data preprocessing and augmentation were performed, focusing on enriching the diversity of the data set and thereby enhancing the robustness of the model, especially for classes that are less represented. Another step in the methodology was exploratory data analysis, which aided in comprehending the data set and thereby forming a better strategy for data augmentation and balancing. Despite the challenges of data imbalance and similarity in images for some disaster types, the model was able to train stably and attain a reasonable accuracy within a constrained dataset environment.

## X. ACKNOWLEDGMENT

We extend our heartfelt gratitude to the Institute of Aeronautical Engineering (IAE) for their unwavering support and for providing the necessary computational and technical infrastructure that made the design, training, and evaluation of the Disaster Zone Classification and Analysis project possible. The resources, guidance, and collaborative learning environment offered by the institute have been instrumental in the successful completion of this work.

## REFERENCES

- [1] H. Song, D. H. Lee, H. G. Baek, B. Bae & S. H. Park, "Automatic Classification of Disaster Images Based on Deep Learning," *Journal of Korean Institute of Communications and Information Sciences*, vol. 48, no. 12, pp. 1633-1636, Dec. 2023.
- [2] L. Wen, Z. Xiao, X. Xu & B. Liu, "Disaster Recognition and Classification Based on Improved ResNet-50 Neural Network," *Applied Sciences*, vol. 15, no. 9, article 5143, May 2025.
- [3] F. Alam, T. Alam, M. A. Hasan et al., "MEDIC: a multi-task learning dataset for disaster image classification," *Neural Computing & Applications*, vol. 35, pp. 2609-2632, Jan. 2023.
- [4] S. Gardoll & O. Boucher, "Classification of tropical cyclone containing images using a convolutional neural network: performance and sensitivity to the learning dataset," *Geoscientific Model Development (GMD)*, vol. 15, no. 18, pp. 7051-7064, 2022.
- [5] Z. Zou, H. Gan, Q. Huang, T. Cai & K. Cao, "Disaster Image Classification by Fusing Multimodal Social Media Data," *ISPRS International Journal of Geo-Information*, vol. 10, no. 10, article 636, 2021.
- [6] S. Sonang, Y. Yuhandri & M. Tajuddin, "Hybrid CNN Approach for Post-Disaster Building Damage Classification Using Satellite Imagery," *Journal of Applied Data Sciences*, 2024.
- [7] A. Calantropio, F. Chiabrandio, M. Codastefano & E. Bourke, "Deep learning for automatic building damage assessment: application in post-disaster scenarios using UAV data," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. V-1-2021, pp. 113-120, 2021.
- [8] Harish K\*, Harish Abinav & D. Kumar, "Deep Learning-Driven Disaster and Aerial Image Segmentation using U-Net and Multi-U-Net Architectures," *Atlantis Press*, 2024.
- [9] "Disaster management using deep learning on social media," Bhavana & P. Ramasubramanian, *International Journal of Applied Science and Engineering*, vol. 18, no. 2, June 2021.
- [10] "Leveraging CNNs and Ensemble Learning for Automated Disaster Image Classification," A. Rathod, V. Pariawala, M. Surana & K. Saxena, 2023.



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)