International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 OPEN ACCESS Available Online at : www.ijsrcseit.com
doi : https://doi.org/10.32628/CSEIT2390453

Proposed Methodology for Disaster Classification Using Computer Vision and Federated Learning

Jash Shah[1], Divya Patel[1], Jinish Shah[1], Saurav Shah[1], Dr. Vinaya Sawant[2]
[1]Department of Information Technology, Dwarkadas J. Sanghvi College of Engineering, Maharashtra, India
[2]HOD, Department of Information Technology, Dwarkadas J. Sanghvi College of Engineering, Maharashtra, India

ARTICLEINFO

Article History:
Accepted: 10 Aug 2023
Published: 30 Aug 2023

Publication Issue
Volume 9, Issue 4
July-August-2023

Page Number
432-442

ABSTRACT

Classification of disasters is crucial for effective disaster management and response. This paper proposes a methodology that combines computer vision techniques and federated learning to improve the classification accuracy of disasters while addressing the issue of data transfer and the time squandered doing so. This methodology employs computer vision algorithms to analyze captured visual data from a variety of sources. It seeks to accurately classify disasters such as wildfires, floods, earthquakes, and cyclones by extracting pertinent features and patterns from these images. Using federated learning to resolve the issues of data privacy and transfer latency is the proposed solution. Federated learning makes it possible to train models on decentralized data sources without requiring data centralization. Each participating device or data source trains a local model using its own data, and only model updates are shared and aggregated to create a global model. Extensive experiments utilizing videos of actual disasters are conducted to evaluate the proposed methodology. The evaluation focuses on precision and effectiveness. This strategy is anticipated to result in improved disaster classification models, making them appropriate for deployment in disaster management systems.
Keywords: Disaster classification, Computer vision, Federated Learning, Deep Learning

I. INTRODUCTION

In image classification tasks, deep learning models have made substantial progress, offering up new opportunities for disaster management and response [1]. Making timely decisions, allocating resources, and implementing mitigation strategies all depend on effective catastrophe classification. Deploying picture classification models in actual crisis situations, however, has special problems with regard to data protection, confidentiality, and scalability. The rapid and precise identification and classification of media content is essential for disaster response as social network sites have becomekey sources of situational information during disasters. This research study suggests a unique method for catastrophe classification that blends computer vision methods with federated learning to overcome these difficulties. While maintaining security and privacy, federated learning enables models to be trained on dispersed data without data sharing. The suggested methodology uses federated learning to provide effective dis- aster classification while protecting the privacy of sensitive data. ResNet50 is one deep learning architecture that is used to further improve the system's accuracy and durability [2]. The proper categorization of disasters is essential for emergency response and resource management. The suggested methodcan analyze and categorize disaster-related photos in real time by fusing computer vision techniques with federated learning, helping decision-makers grasp the gravity of the situation. However, the volume and requirement for automatedapproaches make analyzing social media data challenging. By enabling distant nodes to build an extensive model without sharing local data, the newly developed federated learningapproach offers a viable alternative. In this method, every nodetrains its model locally before uploading the parameters for integration to a centralized server. This decentralized technique eliminates dependency on sending massive amounts of data toa central server and addresses concerns about data privacy [3] [4]. Disaster categorization is a multi-class classification problem, and hence the proposed approach has been evaluated on some benchmark classification metrics such as Precision, recall, and F1 score. The paper also discusses the value of using Federated Machine Learning for the said task. It also goes over some of the challenges faced as well as discussing future scope and ways to improve the system.

II. RELATED WORK

It is possible to think of disaster categorization as a particular use for event classification. A number of papers have addressed disaster detection using artificial intelligence techniques. The disaster categorization approaches can generally be divided into two categories: text-based methods and image-based methods, depending on the modalities of social media data. In other experiments, the goal was to extract textual elements that could be utilized to identify disparate data. Two innovative neural models for disaster information retrieval were put up by Basu et al. [18] and successfully incorporated word-level and character-level embeddings. In their method for categorizing disasters, Madichetty and Sridevi [19] used convolutional neural networks (CNN) to extract features from text input. A deep learning-based method for classifying social media text has been presented by Bhoi et al. [20] and could be used to categorize tweets about emergencies. Even though there aren't many ways for classifying disasters based on photos, numerous studies have shown that images published on social media during a crisis can aid humanitarian organizations [21]. Alam et al. [22], for example, proposed an image analysis system that could identify photographs in crisis situations and extract useful information using deep learning- based techniques. A number of CNN models for ground object detection from aerial pictures of the disaster's aftermath were introduced and assessed by the authors in [23].

III.METHODOLOGY

A. Resnet50

ResNet-50 is a convolutional neural network (CNN) architecture that was introduced by researchers at Microsoft in 2015[1]. It is part of the ResNet (short for Residual Network) family of models, which are renowned for their exceptional performance in image classification tasks. Due to the issue of vanishing gradients, deep neural networks have difficulty training deeper architectures prior to ResNet-50[8]. The gradients propagated across the layers would become less strong as the network depth increased, making it more challenging for the network to learn efficiently. This issue was addressed by the ResNet architecture, which added" skip connections" or residual connections [9]. The idea of residual learning is the main innovation in ResNet-50. ResNet-50 includes residual blocks in addition to the more conventional stacked layers. These blocks have short-cut connections that let data from one layer of the network go quickly to a deeper layer. Thus, the network can efficiently learn residual mappings that capture the discrepancy between the current and desired outputs, improving the training process's effectiveness. The architecture of ResNet-50 consists of multiple layers, including convolutional, and fully connected layers. The network accepts a 224x224 pixel RGB image with three colour channels as input [7]. A convolutional layer with 7x7 filters and a stride of 2 is the first stage's first layer, and it is followed by a max pooling layer with a 3x3 filter and a stride of 2. This increases the number of channels while reducing the spatial dimensions. The residual blocks are the main constituent parts of ResNet-50. There are four different kinds of residual blocks in ResNet-50, each with a particular number of layers. Blocks known as 3x Residual Blocks have two convolutional layers with 64 filters each. Similarly, blocks known as 4x Residual Blocks have three convolutional layers with 128 filters each. Blocks with a 6x residual are made up of four convolutional layers and 256 filters. The final stage consists of three convolutional layers with 512 filters [7]. A layer of global average pooling is added after the residual blocks. It generates a fixed-size output after calculating the average value for each feature map. A fully connected layer with 1000 units, which corresponds to the number of classes in the ImageNet dataset, follows the global average pooling layer [6]. It acts as the categorization output layer. The network's output is transformed into a probability distribution over the classes in the final layer using a SoftMax activation function, allowing classification. Labelled data and optimization methods like stochastic gradient descent (SGD) or its variants are used to train the ResNet-50 network [17]. Convolutional layer and fully connected layer weights, among other model parameters, are iteratively changed during training to reduce the discrepancy between predicted and actual labels.

B. Federated Learning

Federated Machine Learning (FML) is a relatively recent approach that addresses privacy and data ownership concerns in traditional machine learning systems [3]. As researchers investigated ways to train models across different servers or devices without sharing raw data, the idea of distributed machine learning developed. Researchers at Google developed Federated Learning, which was released in 2016. In their seminal publication, they introduced a framework for directly training models on consumer devices while maintaining data privacy. The fundamental concept was to facilitate cooperation while protecting data privacy by having devices broadcast model updates to a central server rather than raw data. Federated Machine Learning (FML) does not have a specific fixed architecture like traditional machine learning models. Instead, it is a framework that involves a distributed and collaborative approach to training models on decentralized data sources. In FML, the training procedure is often coordinated by a central server. The complete training process is managed by the central server, which also aggregates model updates and distributes updated models to the participating devices or clients [13]. In FML, the training takes place right on the local hardware or clients that house the data, as opposed to transfer- ring the data to a centralized server. These gadgets could be cell phones, edge devices, Internet of Things (IoT) devices, or any other decentralized data sources. Using its own local data, each client trains the model and transmits model updates to the central server. To make the interchange of model updates easier, a communication protocol is built between the central server and the local devices. This protocol allows for effective communication in a federated setting while guaranteeing the privacy and security of the data during transmission. The local devices provide model updates and train their models using the data they have on hand. The changes or gradients in the model parameters based on the local data are often represented by these updates. The central server receives the model updates safely. The central server aggregates the model updates from many devices or clients and combines them into a single, global model.

The model updates can be combined using a variety of aggregation approaches, including averaging, weighted averaging, and other more intricate aggregation systems [13]. The modified global model is then distributed back to the participating devices or clients by the central server following aggregation. The new model considers the collective wisdom gained from the diverse data sources. To enhance the model's performance over time, updates, aggregation, and distribution are often carried out iteratively. The model may learn from several decentralized data sources and gain from the network's collective insight by going through several training cycles [3]. FedML, or Federated Machine Learning, can be an effective approach for disaster classification due to its unique characteristics and advantages. When it comes to disaster classification, where timely and accurate predictions are crucial for response and mitigation efforts, FedML offers several benefits. FedML permits collaboration and the pooling of data resources without the sharing of raw data. Instead, the training process occurs locally on local servers or devices, ensuring the confidentiality and security of the data. Using a wide variety of data sources, this decentralized method over-comes the difficulties associated with data silos and enables a more comprehensive comprehension of disasters. Disasters can occur in different locations and exhibit varying characteristics. By utilizing FedML, models can be trained locally on data specific to a particular disaster-prone area, considering local environmental factors, infrastructure, and demographics. This localized approach enhances the model's ability to capture the nuances and unique patterns associated with each disaster type, leading to improved classification performance.

IV.DATASET DESCRIPTION

The training dataset consists of 4400 images from four distinct disaster classes: cyclones (900 images), earthquakes (1400 images), floods (1000 images), and wildfires (1100 images). It is further divided into two datasets: 75 percent training data and 25 percent testing data. To prevent overfitting, the images in the dataset are taken from a variety of perspectives, magnification levels, and illumination conditions, which diversifies the data and challenges the model during training.

V. IMPLEMENTATION

The configuration for the implementation includes a simulation of four clients and one Central Server. In a real- world circumstance, the number of client nodes can be scaled according to the needs of the user. Due to the hardware limitations of simulating Federated learning and training on a single machine, they have been assigned to four. The training data is distributed unequally among clients, as it would be in a realistic scenario. Each client trains their respective model using only the allocated data. At the conclusion of training, test data is evaluated on the global model, i.e., the model on the central server, and the model's performance is evaluated based on a variety of benchmark performance metrics. Adopting a federated approach is primarily motivated by the desire to simulate a real-world scenario in which image data is shared across multiple sources and the transfer of data to a central server would be time-consuming, which should be avoided in this case, i.e. in the event of a natural disaster where time is of paramount importance. The training process begins with pre-processing of the image data. First, the labels to the images are extracted and appended to a label array. Then the image is read, converted to RGB channel ordering, and resized to 224×224 pixels, as the model expects. The model operates on an array of samples, where one image has 224×224 pixels and three channels. Given that the task is of multi-class classification, one-hot encoding is performed on the labels. One-hot encoding transforms categorical labels into binary vectors, where each class is represented by a binary vector with a single element set to 1 and the remaining elements set to 0 [10]. The training data is further split into train and validation sets, 90 percent training and 10 percent validation.

As the next step of pre-processing, the training data undergoes data augmentation. The purpose of data augmentation is to enhance the diversity and variability of the training dataset. Data augmentation introduces variations in the form of rotations, translations, flips, zooms, and more by applying random transformations to the existing training data. This technique helps prevent over-fitting. Data augmentation encourages the model to generalize to unseen examples and improves its robustness to various variations, such as orientation, scale, illumination conditions, and perspective changes [11] [12]. In the proposed approach the data is augmented by performing rotation, zooming, width and height shift, shearing, and flipping by a random value within a range. This completes the pre-processing stage. Next comes the model, which performs the task of classifying the data as one of the disaster classes based on deep CNN networks. The ResNet50 model is used as the base of the model. The pre-trained weights trained on the ImageNet dataset are loaded and the top fully-connected layers are removed to customize the model for the specific task. The ResNet50 model performs the task of feature extraction, and the extracted features are inputted into the custom classification layer [1]. This is the head of the model. This is the head of the model. First, the output of the base model is transformed into a 1D feature vector using the Flatten operation. A dense, fully connected layer with 512 units and a Rectified Linear Unit (ReLU) activation function is subsequently added. Rectified Linear Unit (ReLU) is an activation function that returns 0 if the input is negative and returns the value if positive [14]. A Dropout layer with a dropout rate of 0.5 is included to prevent over-fitting. It randomly sets a fraction of the input units to zero during each training step, effectively disabling them. This helps the model learn more robust and generalized representations by preventing reliance on specific input units and encouraging the network to utilize different combinations of units [15]. Class probabilities are

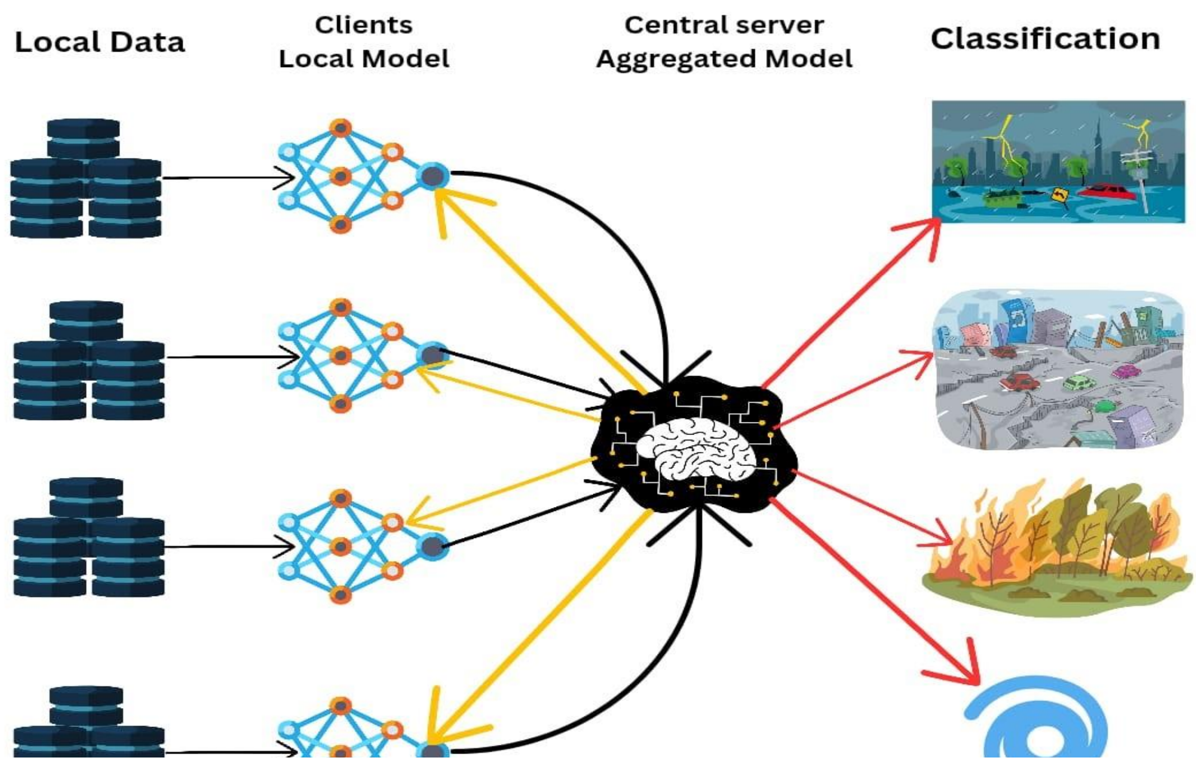

Figure 1. Proposed approach pictorial representation. The black arrows show updates from the clients' local model to central server and the golden arrows indicate sharing of global updates. The red arrows indicate the classification performed by the global weights

produced by appending a Dense layer with the SoftMax activation function. SoftMax activation is a mathematical function used to convert a vector of real values into a probability distribution, where the output values sum up to 1 and represent the likelihood of each class in a multiclass classification problem. This final layer contained the same number of units as classes in our classification task. To ensure that the pre-trained weights of the base model are not updated during the initial training, the trainable attribute of each layer in the base model is set to False. Combining the base model with the customized head model yields the entire model architecture. To compile the model categorical cross entropy is used. It is commonly used for multiclass classification as is for the task at hand. The categorical cross-entropy loss determines the mean logarithmic loss across all classes. The model is penalized more severely when it allocates low probabilities to the correct classes and high probabilities to the incorrect classes. The objective of minimizing the categorical cross-entropy loss is to motivate the model to generate more precise and confident predictions for each class. The lower the categorical cross-entropy loss, the better the model's predictions align with the true labels [16]. The employed optimizer is Stochastic Gradient Descent (SGD). It is more computationally efficient than traditional Gradient Descent because it adjusts the model's parameters iteratively by computing the gradients of the loss function on a small subset of training samples (mini-batch) at each step. SGD seeks to identify the optimal parameter values that minimize the loss function, enabling the model to converge over time to a superior solution [17]. Hyperparameter tuning has been performed to optimize the model further. Now, each of the four individual models is executed for eight rounds. Federated Averaging has been adopted for the aggregation of results back to the central server [13]. It enables all clients to perform multiple batch updates on local data and facilitates the exchange of the updated weights. In this method, the global weights are updated at the outset of each round based on the changes made to the local weights in the previous round. Each client then constructs a local model containing the previously defined hyper-parameters. The initial weight of the local model is set to the weight of the central server's global model. The client then employs the data it was allotted to train a local model. The local weights are then scaled using a scaling factor for each client, and once all local weights have been updated, they are aggregated and averaged to produce a new global weight for the subsequent global round. At the conclusion of eight cycles, the central server generates the final model weights for classification. The implementation code has been written in python. The model has been evaluated based on metrics discussed in the next section. The model has been tested on various video inputs using videos available on the internet

VI. RESULTS

A. Evaluation Parameters

Given that the problem statement is a multi-class classification problem, the following parameters have been selected as evaluation metrics in order to gain a more nuanced comprehension of the model's strengths and weaknesses in various classification task-related aspects.

Figure 2. Model output for Earthquake

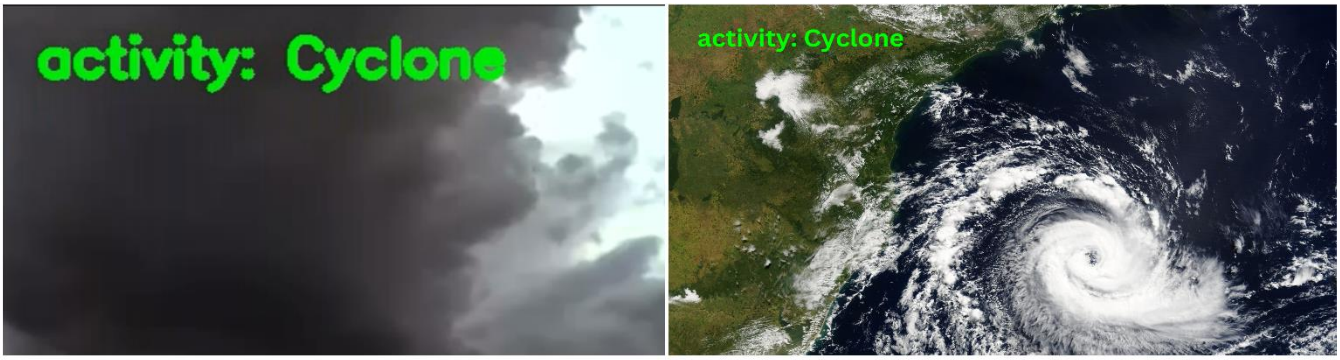

Figure 3. Model output for Cyclone

Figure 4. Model output for Flood

Figure 5. Model output for Wildfire

1) Accuracy: Accuracy is a commonly employed metric for assessing the performance of a classification model. It indicates the proportion of instances that were accurately predicted relative to the total number of instances in the dataset. In multiclass classification, accuracy is determined by averaging the accuracy of each class. It indicates the extent to which the model can correctly classify instances across all classes.

2) Precision: The capacity of a model to recognize only relevant items is referred to as precision. True positive (TP) denotes the number of the model's accurate positive predictions, whereas false positive (FP) denotes the number of the model's inaccurate positive predictions.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

3) Recall: Recall measures the proportion of actual positive cases that the model correctly identifies as positive, and is, therefore, an indication of the model's ability to detect positive cases. True positive (TP) represents the number of correct positive predictions made by the model, while false negative (FN) represents the number of incorrect negative predictions made by the model.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

4) F1 Score: The F1 score is an evaluation metric that gauges the correctness of a model. It blends a model's precision and recall scores. The accuracy statistic calculates how many times a model predicted correctly over the full dataset. The F1 score provides a balanced assessment of a model's ability to correctly identify both positive and negative cases.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

B. Observed parameter scores

With impressive metric scores, the model demonstrates tremendous potential. The observed values for individual disaster classes are presented in Table I, and the model's aggregate performance is summarised in Table 2.

Table I : Precision, Recall and F1 Score of individual disaster classes

Disaster Class	Precision	Recall	F1-Score
Cyclone	0.95	0.96	0.95
Earthquake	0.94	0.92	0.93
Flood	0.90	0.93	0.91
Wildfire	0.95	0.93	0.94

The effectiveness of the system in classifying individual disasters is detailed in Table I. The Cyclone class has the highest performance because it was trained with higher- quality satellite images. Then come wildfires, earthquakes, and floods respectively. They all have comparable scores and perform exceptionally well on the classification task.

Table II : Evaluation score of overall model

Evaluation Metric	Score
Accuracy	0.94
Precision	0.92
Recall	0.93
F1-Score	0.925

Table II details the overall performance of the model, which received high evaluation scores. The model's accuracy is 0.94, its precision is 0.93, its recall rate is 0.92, and its F1 score is 0.925. This demonstrates its robustness and operational excellence.

VII. CHALLENGES FACED

A. Dataset

Finding an appropriate dataset posed a significant challenge, as we sought to procure a diverse set of images that not only exhibited high resolution, but also featured low-quality depictions, variable environments, inadequate lighting and other forms of interference. To implement Federated Learning, a larger dataset was required so as to ensure there is sufficient data on each local client for training.

B. Hardware requirements

The hardware requirements are one of the most challenging aspects of implementing the disaster classification method with Federated Learning (FedML) and computer vision. The computational requirements of training deep learning models and the resource-intensive character of Federated Learning impose significant hardware constraints. High-performance GPUs or specialized hardware accelerators are often necessary to efficiently process the large volume of image data and perform complex operations involved in feature extraction and model training. In addition, the decentralized nature of Federated Learning necessitates that each client has sufficient hardware resources to execute local model training and contribute to the collaborative learning process. To simulate this environment a device with sufficient power was required.

VIII. CONCLUSION

In conclusion, this research paper proposed a methodology for disaster classification using computer vision and Federated Learning (FedML). ResNet50 was utilized as a feature extractor with a custom model head for classification, and the FedAvg algorithm was implemented for the Federated Learning framework. The use of computer vision techniques allowed for the extraction of meaningful features from disaster images, enabling accurate classification of various disaster scenarios. ResNet50, a widely adopted pre-trained convolutional neural network, served as a powerful feature extractor, capturing important visual patterns and representations from the input images. The paper suggests the use of Federated Learning to address the challenges of data privacy and distributed data storage in disaster classification. FedML facilitated the collaborative training of the classification model across multiple clients without requiring centralized data aggregation. These client model updates were aggregated using the FedAvg algorithm. The proposed method yielded promising results in disaster classification, demonstrating the efficacy of integrating computer vision techniques with Federated Learning. The model obtained a high degree of precision in classifying images of disasters, thereby facilitating rapid response and decision-making in emergency situations.

IX. FUTURE SCOPE

A. Threat level categorization

Incorporate categorization of various threat or severity levels related to catastrophes. This could entail classifying dangers or their seriousness into categories like low, moderate, or high, or into levels particular to the kind of disaster, such as earthquake intensity levels or flood intensity levels.

B. Finding malicious clients

Due to secured averaging, all FL clients are anonymous, which allows any malicious clients to upload ambiguous updates to the server and launch a targeted model poisoning attack on the server to harm the performance of the global model. It is essential to look for and identify these malicious model upgrades and the attackers behind them. Byzantine- robust FL approaches are among the few defence measures against suspect clients, although they cannot experimentally ensure whether or not the predicting labels used for testing are altered.

C. Developing a plan to mitigate the negative impact of the unreliable nodes

Unreliable nodes in Federated Learning (FL) refer to de- vices or servers that exhibit inconsistent behaviour or provide inaccurate updates throughout the training process. Unreliable nodes can have a negative effect on the final model's quality and dependability. Mitigating this improves the accuracy and dependability of the model by minimising the impact of inconsistent or imprecise updates. It assures more reliable node selection, robust aggregation algorithms, and outlier detection mechanisms, resulting in a more accurate and reliable aggregated model.

X. REFERENCES

[1]. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[2]. Agrawal, T. and Meleet, M., 2021, September. Classification of natural disaster using satellite and drone images with CNN using transfer learning. In 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) (pp. 1-5). IEEE.

[3]. He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H. and Zhu, X., 2020. Fedml: A research library and benchmark for federated machine learning. arXiv preprint arXiv:2007.13518.

[4]. He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H. and Zhu, X., 2020. Fedml: A research library and benchmark for federated machine learning. arXiv preprint arXiv:2007.13518.

[5]. Mascarenhas, S. and Agarwal, M., 2021, November. A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENT- CON) (Vol. 1, pp. 96-99). IEEE.

[6]. Mascarenhas, S. and Agarwal, M., 2021, November. A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENT- CON) (Vol. 1, pp. 96-99). IEEE.

[7]. Li, Y., Wang, H., Sun, S. and Buckles, B., 2020, June. Integrating multiple deep learning models to classify disaster scene videos. In 2020 IEEE High Performance Extreme Computing Conference.

[8]. Borawar, L. and Kaur, R., 2023, March. ResNet: Solving Vanishing Gradient in Deep Networks. In Proceedings of International Conference on Recent Trends in Computing: ICRTC 2022 (pp. 235-247). Singapore: Springer Nature Singapore.

[9]. Wu, D., Wang, Y., Xia, S.T., Bailey, J. and Ma, X., 2020. Skip connec- tions matter: On the transferability of adversarial examples generated with resnets. arXiv preprint arXiv:2002.05990.

[10]. Brownlee, J., 2020. Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python. Machine Learning Mastery.

[11]. Rebuffi, S.A., Gowal, S., Calian, D.A., Stimberg, F., Wiles, O. and Mann, T.A., 2021. Data augmentation can improve robustness. Advances in Neural Information Processing Systems, 34, pp.29935-29948.

[12]. Shorten, C. and Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. Journal of big data, 6(1), pp.1-48.

[13]. Nilsson, A., Smith, S., Ulm, G., Gustavsson, E. and Jirstrand, M., 2018, December. A performance evaluation of federated learning algorithms. In Proceedings of the second workshop on distributed infrastructures for deep learning (pp.1-8).

[14]. Agarap, A.F., 2018. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

[15]. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdi- nov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), pp.1929- 1958.

[16]. Zhang, Z. and Sabuncu, M., 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems, 31.

[17]. Keskar, N.S. and Socher, R., 2017. Improving generalization perfor- mance by switching from adam to sgd. arXiv preprint arXiv:1712.07628.

[18]. Basu, M., Shandilya, A., Khosla, P., Ghosh, K. and Ghosh, S., 2019. Extracting resource needs and availabilities from microblogs for aiding post-disaster relief operations. IEEE Transactions on Computational Social Systems, 6(3), pp.604-618.

[19]. Madichetty, S. and Sridevi, M., 2019, January. Detecting informative tweets during disaster using deep neural networks. In 2019 11th International Conference on Communication Systems and Networks (COMSNETS) (pp. 709-713). IEEE.

[20]. Bhoi, A., Pujari, S.P. and Balabantaray, R.C., 2020. A deep learning- based social media text analysis framework for disaster resource management. Social Network Analysis and Mining, 10, pp.1-14.

[21]. Chaudhuri, N. and Bose, I., 2019. Application of image analytics for disaster response in smart cities.

[22]. Alam, F., Ofli, F. and Imran, M., 2018. Processing social media images by combining human and machine computing during crises. International Journal of Human–Computer Interaction, 34(4), pp.311-327.

[23]. Pi, Y., Nath, N.D. and Behzadan, A.H., 2020. Convolutional neural networks for object detection in aerial imagery for disaster response and recovery. Advanced Engineering Informatics, 43, p.101009.

Cite this article as :

Jash Shah, Divya Patel, Jinish Shah, Saurav Shah, Dr. Vinaya Sawant, "Proposed Methodology for Disaster Classification Using Computer Vision and Federated Learning", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 4, pp.432-442, July-August-2023. Available at doi : https://doi.org/10.32628/CSEIT2390453
Journal URL : https://ijsrcseit.com/CSEIT2390453