Image Classification Based on Disaster type Using Deep Learning

Anisha Coopen and Sameerchand Pudaruth

ICT Department, FoICDT, University of Mauritius, Mauritius

Abstract

People nowadays use social media platforms to capture and share real-time incidents in the form of images, videos and text. However, sharing too much information at once makes it harder for first responders to determine where exactly individuals are in need and whether they require immediate assistance. In the past, machine learning techniques were used to automatically identify and infer disaster response from images, as manually identifying disaster types is currently challenging. Therefore, in this paper, deep learning models are used to investigate how well they can classify the images according to their disaster type by learning the features extracted from the input images on their own. In this study, 2 existing datasets namely the 'Comprehensive Disaster Dataset' (CDD) and 'Natural Disaster Dataset' (NDD) based on disaster types were customized into a dataset entitled as 'Customized Disaster Dataset'. The Customized Disaster Dataset comprises of a total of ten classes, three of which are non-damage images, Pre-trained models like the MobileNetV2, VGG16 and InceptionV3 were used to train the datasets to allow for further comparison with existing studies. Along with that, a customized neural network model was created and trained on the datasets. Different scenarios were devised to assess the top 3 performing models. The InceptionV3 being best model had a classification accuracy of 96.86%. In this study, we have demonstrated the effectiveness of CNN models as a tool for automatic disaster type classification.

Keywords: First aid responders, Convolutional Neural Networks (CNN), deep learning models, dataset, MobileNetV2, VGG16, InceptionV3

1 Introduction

In an era where natural calamities endanger human lives and destroy infrastructure, the ability to reliably categorize disaster-related images is critical for disaster response management and mitigation operations. Earthquakes, floods, landslides, and other natural calamities generate massive destruction. The Emergency Event Database (EM-DAT) reported 387 natural disasters in 2022, resulting in the deaths of 30,704 persons [1]. Natural catastrophes cause infrastructure damage, a high death toll, and local economic losses. Following a tragedy, millions of people use social media platforms to post videos, pictures, and tweets about the incident in hopes of receiving support from the rightful authorities to provide relief and medical treatment to victims. This is critical for first responders, disaster management, and non-governmental organizations (NGOs), as they are the ones that infer appropriate emergency responses based on the type of disaster [2]. However, manually filtering all disaster-related postings among other irrelevant posts such as random movies and advertisements is challenging. As a result, in order ensure an efficient emergency response, it is critical to first classify the images according to their disaster type. This has resulted in the surge to use deep learning algorithms to automate the disaster image classification process [3]. This paper attempts to address the disaster image classification problem by developing a robust deep learning model that can automatically recognize and categorize disaster-related images based on disaster categories such as 'Cyclone',' Earthquake', 'Flood', 'Drought', 'Landslide', 'Wild_Fire' and 'Urban_Fire' with the intention to improve the efficiency and efficacy of disaster response activities by automating the image classification process. By adopting deep learning algorithms, the time and necessary resources such as human intervention are reduced drastically allowing for a faster and more informed decision-making during crucial situations [4]. This paper has a dual focus on developing a custom image classification model and utilizing pretrained convolutional neural network (CNN) models separately for the classification of seven major disaster types: 'Earthquake', 'Cyclone', 'Wildfires, Urban_fires', 'Landslide', 'Drought' and 'Flood' along with three non-damage classes labelled as 'Non_Damage_Buildings_Streets', 'Non_Damage_Wildforest' and 'Sea'. These non-damage classes are added to test the model's ability to differentiate between catastrophe images like earthquakes, floods, and wildfires and non-disaster images like buildings, forests, and seas. These non-damage categories are chosen specifically since the class Earthquake and non-damage building both contain images of buildings but in different scenarios: Similarly, flood and sea contain comparable patterns, as do wildfires and wild forest. This research will encompass a wide range of imagery sources, including satellite imagery for the category 'flood' and ground-based images for the other categories, to ensure a comprehensive understanding of disaster classification across different data types. For the custom model, a neural network architecture is designed to accurately classify the different disaster types. In addition, the research leverages pre-trained CNN models for disaster type classification. These pre-trained models such as VGG16, MobileNetv2, and InceptionV3, are fine-tuned and adapted to the specific disaster-related dataset, enabling the extraction and utilization of generic visual features relevant to the different disaster types. This approach aims to benefit from the pre-trained models' knowledge of

2

generic visual patterns while incorporating the domain-specific information required for accurate classification [5].

This paper proceeds as follows: In the next section, we provide an overview of recent works that have been done on disaster image classification using computer vision and machine learning techniques. The methodology is described in section 3 while the implementation details, the results and their evaluation are provided in section 4. Section 5 concludes the paper with some ideas for future works.

2 Related Works

Amit et al. [6] created an automated approach for identifying catastrophes by analyzing satellite photos with convolutional neural networks (CNN). Three convolutional layers, two max-pooling layers, and two fully linked layers comprised their CNN design. Using 30,000 to 40,000 picture patches from Google Earth aerial photographs, the scientists generated a training dataset for landslides and floods in Japan and Thailand. Using a raster scan approach, the CNN was trained for fast extraction of disaster zones. To show the occurrence of a disaster, regions with high forecast values were highlighted by creating a 32x32 rectangular box and labeling it with 1. Both catastrophes had F1- Scores ranging from 80% to 90%. For feature extraction, the model utilised six RGB channels, prevailing over previous techniques that only used two grayscale channels. It should be pointed out, however, that their dataset only comprised images captured in bright weather conditions. Tackling the difficulty of diverse color changes associated with varying weather conditions remains a work in process [6]. Liu and Wu [7] used wacDAE-2 (Wavelet Auto-Encoder with 2 Hidden Layers) to build a deep learning-based approach to detect landslides in optical remote sensing images. They used a wavelet transformation to capture hidden characteristics. They also used a corrupting and denoising strategy to increase the resilience of the model in recognizing landslide characteristics. To learn high-level characteristics and representations for each picture, a deep autoencoder network with several hidden layers was employed. For class prediction, a softmax 20 classifier was applied. Google Earth remote sensing images were used in the evaluations. The suggested wavDAE-2 approach by Liu and Wu surpasses SVM and ANN classifiers in terms of efficiency and accuracy, reaching a classification accuracy of 97.40%. They intend to test the approach on real-world optical remote sensing datasets, compare it to existing methods, and investigate network optimization methodologies. They also intend to create a robust deep autoencoder network for highperformance computation on CUDA-enabled GPUs [7]. Dunnings and Breckon [8] implemented a real-time, automatic fire detection in videos using modified versions of AlexNet and InceptionV1 models, called AlexNet and InceptionV1-OnFire respectively. They used superpixel localization techniques. The implemented CNN architectures obtained a maximum accuracy of 93% for binary fire detection in images, and an accuracy of 89% within their superpixel localization framework. The models also performed significantly faster, processing frames at a rate of 17 frames per second. However, the study focused only on fire-related disasters [8]. Ofli et al. [9] proposed an early fusion multimodal deep learning architecture for joint representation learning of text and picture modalities. For text and pictures,

3

they employ two parallel architectures, including the VGG16 model for image classification and a customized CNN model for text. The multimodal architecture uses a shared dense layer to aggregate data from both modalities and softmax to predict output. The studies make use of the CrisisMMD dataset, which contains pictures from seven natural disasters in 2017. In unimodal experiments, image-only models outperform text-only models by 2.5% and 6.4% in informativeness and humanitarian categorization tasks respectively. The multimodal technique, on the other hand, performs marginally better, with a 1.1% improvement in informativeness classification and a 1.6% improvement in humanitarian categorization tasks [9]. Asif et al. [10] created a disaster taxonomy and emergency response pipeline for automated decision-making in emergency circumstances using deep learning algorithms. Card sorting was used to validate the taxonomy's correctness and completeness. The authors classified and identified objects found in disaster-related pictures using the VGG-16 and YOLO algorithms. The analytic hierarchy process (AHP) mapped catastrophe images to the taxonomy and chose relevant emergency response categories, while decision tables aligned intermediate results. With YOLOv4, the technique obtained 96% classification accuracy [10]. Zou et al. [11] investigated how to detect catastrophe images from social media using the VGG16 deep learning model and the FastText framework. Using the CrisisMMD dataset, they developed a data fusion model that used visual and linguistic features to categorize relevant photos. In Task 1, the multimodal approach outperformed unimodal methods, with an accuracy of 87.6% against 83.3% for image-only approaches and 85.2% for text-only approaches. In Task 2, the multimodal technique outperformed unimodal methods by 0.4%, with 92.6% accuracy vs 90.7% for text-only and 92.2% for image-only approaches. The study recognizes the problem of imbalanced data and intends to solve it in a future work [11]. Dinani and Caragea [12] investigated capsule networks against convolutional neural networks (CNNs) for classifying disaster photographs as useful or uninformative. They used images from various disasters, including the CrisisMMD dataset, to compare capsule network models to ResNet18 models in both in-domain and cross-domain situations. The results demonstrated that capsule 23 networks performed better when training datasets were small or imbalanced, outperforming ResNet18 models. The researchers intend to perform controlled experiments to further understand the effects of sample size and class imbalance, as well as adapt CapsNet models to additional multi-class classification challenges, such as classifying different types of disasters [12]. Hossain et al. [13] created a multimodal catastrophe detection system that uses textual and visual information to properly classify tweets. They extracted textual data using a bidirectional long-term memory (BiLSTM) network with an attention mechanism, while visual characteristics were extracted using a pretrained convolutional neural network (CNN) like ResNet50. For combined predictions, the system was fused using a feature fusion technique and a softmax classifier. To better capture word token dependencies, the researchers compared BiLSTM with attention mechanisms to CNN-based approaches. The multimodal system outperformed conventional unimodal and multimodal models, improving performance by around 1% and 7%, respectively [13]. Table 1 provides a summary of the research papers.

4

Table 1 Summary of papers

References	Datasets	Classifier	Accuracy/F1-Score
Amit et al. [6]	Google Earth aerial images	CNN	$80 \sim 90$
Liu and Wu [7]	Google Earth remote sensing images	WacDAE-2	97.4
Dunnings and Breckon [8]	Fire related images	Alexnet & InceptionV1	93.0
Ofli et al. [9]	CrisisMMD	VGG16	83.3
Asif et al. [10]	CrisisMMD	VGG-16 & YOLOV4	96.0
Zou et al. [11]	CrisisMMD	VGG16 & FastText	92.2
Dinani and Caragea [12]	CrisisMMD	Capsule network & ResNet-18	92.2
Hossain et al. [13]	Twitter	ResNet50 & BiLSTM	81.88

This section explores previous studies along with its techniques and approaches used for disaster image classification. This paper aims to explore different deep learning models using two existing disaster image datasets on Kaggle, integrating them into a single dataset for disaster image categorization. Since the CrisisMMD dataset is a multimodal dataset that includes tweets and disaster related images from Twitter, it will not be employed in this research since its focus is unimodal.

3 Methodology

The main objective of this study is to perform automatic image disaster recognition. In this section, a solution has been proposed to overcome the main challenge of manual classification of disaster type. Figure 1 depicts an overall system architecture of the stages involved in creating the image disaster classification.

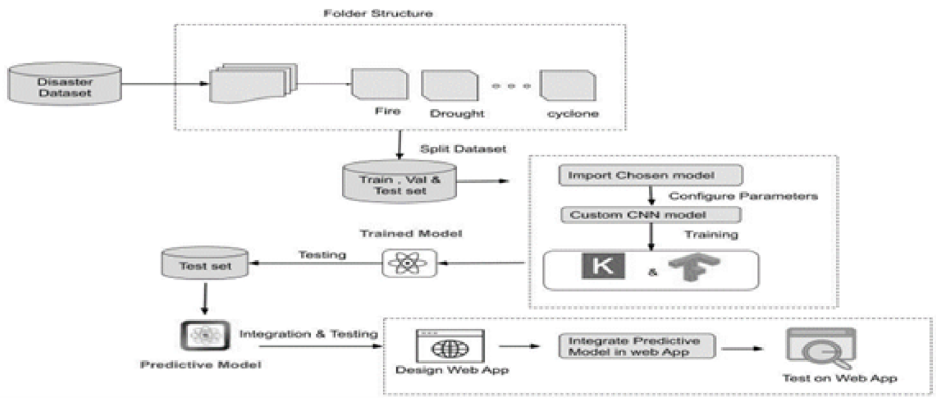

Fig. 1 Proposed System Architecture

5

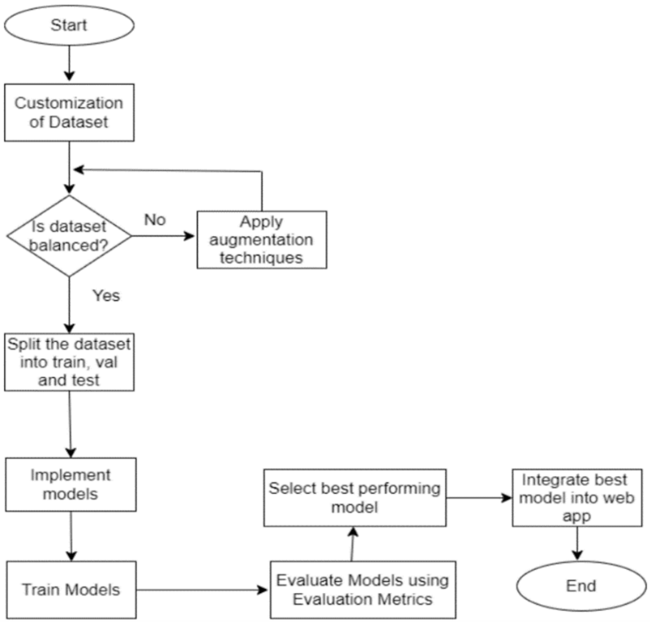

Fig. 2 Flowchart for the proposed system

Figure 2 demonstrates the flowchart for the proposed system. The two datasets, namely Comprehensive Disaster Dataset (CDD) and the Natural Disaster Dataset (NDD) are customized into a new dataset. Two variations of the dataset will be employed, one comprising 350 images and the other consisting of 750 images, to examine the impact of dataset size on model training. The dataset will be divided into two different ratios to observe how varying the number of images in the training and validation sets affects the outcomes. Three different pre-trained models, namely VGG16, MobileNetv2, and Inceptionv3, will be utilized for training the model through transfer learning. The model's performance will be evaluated using the test set. The evaluation metric employed will be the Classification Accuracy. Additionally, the best-performing models will be further evaluated under different scenarios. The top-performing model will be then integrated into a web application.

3.1 Dataset and Preprocessing

Fig. 3 Samples Images from the Customized Disaster dataset

The Natural Disaster Dataset (NDD) and Comprehensive Disaster Dataset (CDD) were integrated into another dataset named 'Customized Disaster Dataset' to focus on various disaster categories. To produce the customized dataset, a subset of pictures from the NDD dataset were merged into some of the classes in the CDD dataset, excluding certain classes such as 'Human' and 'Human Damage'. Furthermore, the class 'Water_Disaster' was renamed 'Flood' to conform to the Natural Disaster dataset's naming convention. Furthermore, because the images for the classes 'Infrastructure Damage' and 'Earthquake' were similar, rather than considering them as different classes, several pictures from the 'Infrastructure Damage' folder were shuffled into the 'Earthquake' category following the dataset cleaning process. Figure 4 shows some sample images for the non- damage categories.

7

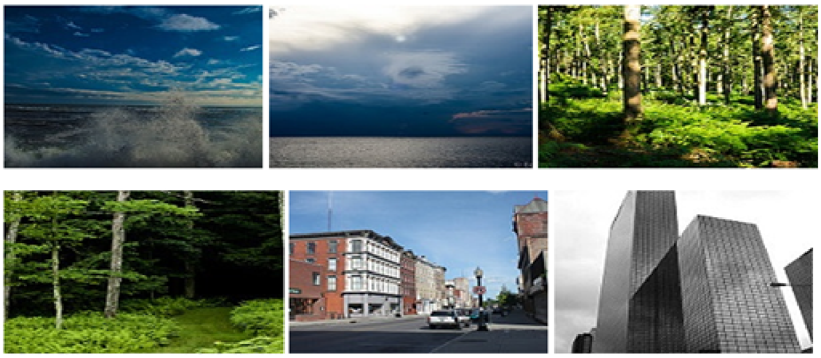

Fig. 4 Samples Images for the Non - Damage classes

These non-damage classes are incorporated in the customized disaster dataset so that the model can better distinguish between disasters and non-disaster categories that share similar patterns.

Both the datasets with 350 photos per class and the other one with 700 images per class, were split into various versions using the following split ratios shown in Table 2.

Table 2 Split ratios

Train(%)	Validation(%)	Test(%)
80	10	10
60	30	10

3.2 Classification Phase Using Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of artificial neural network designed to recognize visual patterns from pixel images with minimal pre-processing. These networks use features of the visual patterns. Convolutional neural networks consist of two simple elements: convolutional layers and pooling layers. Convolutional layers and pooling layers work together to allow CNNs to automatically learn hierarchical data representations, making them extremely effective for tasks such as image classification [14]. The Convolutional neural networks are popular due to their architecture, which eliminates the need for manual feature extraction. Instead, the system uses convolution of image and filters to generate invariant features, which are then passed on to the next layer. The features in the next layer are convoluted with different filters to generate more invariant and abstract features, resulting in an output that is invariant to occlusions. Common convolutional neural network architectures include LeNet, AlexNet, ZFNet, GoogLeNet, VGGNet, and ResNet [15]. A CNN architecture is depicted in Figure 5.

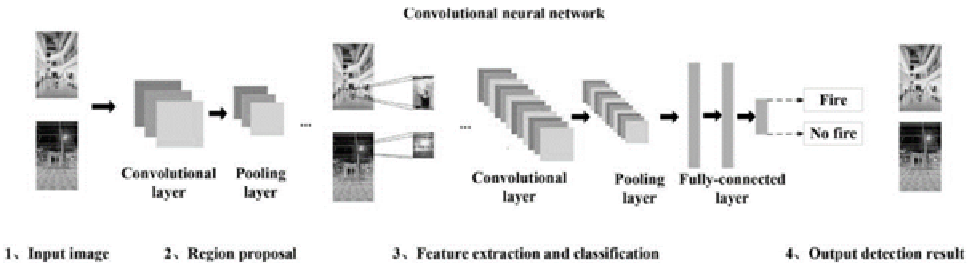

Fig. 5 Architecture of CNN (Li & Zhao) [16]

The convolutional layer uses filters to extract features from the input image. It involves identifying edges, colors, and textures. The Activation function (e.g ReLU) introduces non-linearity to the model and helps decide if neurons should be activated. The pooling layer reduces computational complexity by downsampling feature maps while retaining important information. The Flatten layers converts pooled feature maps into a flat vector hence preparing data for input to a Fully Connected Layer. The Fully connected layer is a neural network layer that performs classification or identification based on extracted features. Softmax function is used to convert raw scores to probability distributions while Cross-entropy is used as the loss function for training [17].

Call back functions are functions that are called repeatedly to evaluate the performance of the model during the training. ModelCheckpoint and EarlyStopping are inbuilt callback functions that are used in this paper.

ModelCheckpoint is one of the callback functions. During the training phase, the ModelCheckpoint is used to preserve the best model as well as the best model weights at each epoch interval [18]. The parameters that are associated with the function involves a file path to specify the model file path for saving, a 'monitor' to be used as a metric for early stopping detection and the 'save_best_only' parameter can be set to 'True' to save the best model and weights [19].

Early Stopping If the model achieves optimal performance sooner than expected, the Early Stopping function stops the training process. 'Patience' is one of the function's parameter. A number is assigned to it. This value denotes the number of epochs to wait for, if no improvement in performance is noted before halting the training [20].

3.3 Experimental Setup

This section aims to describe the different components of the system, the hardware and software requirements that are required to perform the disaster image recognition. Table 3 lists the libraries and tools utilized in the system's development.

All the codes were trained on Google Colab since it provides access to Tesla T4 GPU hardware for 12 hours a day. Furthermore, Google Drive may be readily

9

Table 3 Tools and Technologies used

Tools	Description
OpenCV	Open-Source library for image processing and machine learning.
Numpy	Matrix and multidimensional arrays.
TensorFlow	Allow the implementation of deep learning model.
Keras	API used for deep learning
Google Colab	Platform to train deep learning models. Provides usage of free GPU (eg. T4)

mounted on top of the Google Colab platform, from which the dataset can be accessible instantly.

3.4 Feature extraction using pre-trained models

In this section, various pre-trained models were utilized to extract features from the data. The pre-trained models employed include 'MobileNetV2', 'VGG16' and 'InceptionV3'. Freezing the base model layers prevents their weights from being updated during training, allowing them to be used as fixed feature extractors. Transfer learning is a technique that uses a pre-trained deep neural network models to perform task like image classification. These models are trained on large dataset like the ImageNet and COCO datasets. This technique yields better results than training with limited data, as the model leverages learned features to perform the task. It helps to prevent overfitting and reduces the computational burden during training since the gradients are not calculated or applied to these layers. If the value of the 'trainable.layer' is set to 'True', the weights will be updated during the training phase. Both scenarios are tested in this paper. Different image sizes were utilized depending on the specific pre-trained model used. Table 4 provides a summary of the pre-trained models used.

Table 4 Summary of pre-trained models Used

Feature Extraction	Image Size	Trainable Layer	Classifier
MobileNetv2	224*224	True False	Softmax
VGG16	299*299	True False	Softmax
InceptionV3	224*224	True False	Softmax

The base layer of a pre-trained model has been enhanced with custom layers such as Global Average Pooling, Dense layers, and Dropout layer to reduce overfitting. The last dense layer uses the Softmax function to generate probability values for the dataset's 10 classes.

A custom CNN model is implemented using the Sequential API in Tensorflow, consisting of Conv2D layers for feature extraction and Maxpooling2D layers for reducing spatial dimensions. Global AveragePooling2D layers average feature map values, which

10

are converted to 1D vectors by the flatten layer. The custom model has 4,472,970 of trainable parameters in total.

This study uses a custom callback function, ModelCheckpoint and EarlyStopping, to modify model parameters during training. A 'lr_schedule' learning rate schedule function is created, adjusting the learning rate value exponentially based on epoch count. If the epoch count is less than 10, the learning rate value remains intact; otherwise, if the epoch count is larger than 10, the learning rate value is lowered exponentially at a rate of 0.1. This fine-tunes the model's parameters and improves performance. The Adam optimizer is used, with an initial learning rate of 0.001 and a decay rate of 1e-6. The loss function is categorical cross entropy which is best suited for multi-class classification instances.

4 Results and Evaluation

The models that were tested with the different dataset versions and split ratios are evaluated in this section. For each dataset version, its classification accuracy is calculated and compared with all the models.

4.1 Pre-trained Model Results

The results for the pre-trained models trained on the different ratios for both the small and large dataset are presented in the tables below.

4.1.1 Large Dataset

Tables 5 and 6 show the findings for the large dataset for the ratios 8:1:1 and 6:3:1.

Table 5 Large_8_1_1 Results

Model	Trainable Layers	Classification Accuracy (%)
MobileNetV2	False	92.00
	True	96.86
VGG16	False	86.57
	True	92.86
InceptionV3	False	96.86
	True	94.86

For the MobileNetV2 when the trainable layers were set to 'False', an accuracy and a recall of 92.00%. When the trainable layers were changed to 'True', the accuracy increases by 4.8%. Additionally, for the VGG16 model, the accuracy has increased from 86.57% to 92.70% when the trainable layers were changed from 'False' to 'True'. The InceptionV3 model has achieved an accuracy and a recall of 96.86% with the trainable layers being set to 'True'. However, when setting the trainable layers to 'False', the model yielded a decrease in its accuracy by 2%.

For the MobileNetV2 when the trainable layers were changed from 'False' to 'True', the classification accuracy increases from 93.71% to 96.86%. However, for the VGG16

11

Table 6 Large_6_3_1 Results

Model	Trainable Layers	Classification Accuracy (%)
MobileNetV2	False	93.71
	True	96.86
VGG16	False	77.71
	True	65.71
InceptionV3	False	95.71
	True	95.71

model, the classification accuracy decreases from 77.71% to 65.71% when the trainable layers were changed from 'False' to 'True'. The InceptionV3 model has achieved an accuracy of 95.71% with the trainable layers being set to 'False'. The accuracy remained the same when the trainable layers were set to 'True'.

4.1.2 Small Dataset

Tables 7 and 8 show the findings for the small dataset for the ratios 8:1:1 and 6:3:1.

Table 7 Small_8_1_1 Results

Model	Trainable Layers	Classification Accuracy (%)
MobileNetV2	False	93.43
	True	95.71
VGG16	False	84.86
	True	84.29
InceptionV3	False	95.71
	True	95.43

For the MobileNetV2 when the trainable layers were set to 'True', an accuracy of 93.43. When the trainable layers were changed to 'False', the accuracy increases by 2.28%. However, for the VGG16 model, the accuracy has dropped from 84.86% to 84.29% when the trainable layers were changed from 'False' to 'True'. The InceptionV3 model has achieved an accuracy of 95.71% with the trainable layers being set to 'True'. However, when setting the trainable layers to 'False', the model yielded a decrease in its accuracy by 0.28%.

Table 8 Small_6_3_1 Results

Model	Trainable Layers	Classification Accuracy (%)
MobileNetV2	False	92.29
	True	96.00
VGG16	False	85.71
	True	74.86
InceptionV3	False	94.86
	True	93.43

12

The accuracy of the MobileNetV2 model is 96.00% when the model's trainable layers were set to 'True'. But when the trainable layers were set to "False," the accuracy fell by 4.29%. On the other hand, the VGG16 model's accuracy decreased from 85.71% to 74.86% when the trainable layers were enabled. Setting the trainable layers to "True" produced an accuracy of 93.43% for the InceptionV3 model.

The results reveal that setting false trainable layers in ImageNet weights improves accuracy in CNN pre-trained models trained on a large dataset of 1000 classes. The models successfully used transfer learning when combined with ImageNet weights, resulting in better classification accuracy rates on the disaster dataset. However, the performance was not significantly improved when the dataset size was doubled from 350 to 700 photos per category using pre-trained models like InceptionV3, MobileNetV2. The small dataset of 3500 photos already shows a balanced representation of classes, suggesting that increasing the dataset size might not improve class distribution or lead to higher performance.

4.2 Custom CNN Model Results

The findings for the custom model are tabulated in Table 9.

Table 9 Custom Model Results

Model	Classification Accuracy (%)
Custom_Small_8_1_1	71.11
Custom_Small_6_3_1	69.14
Custom_Large_8_1_1	88.57
Custom_Large_6_3_1	82.29

An accuracy of 71.11% was attained using the Custom_Small_8_1_1 dataset. Based on these findings, it can be concluded that the model performed relatively well in correctly identifying disaster images from the Custom_Small_8_1_1 dataset. The model's accuracy decreased to 69.14% when trained on the Custom_Small_6_3_1 dataset. A high accuracy of 88.57% were achieved by the Custom_Large_8_1_1 dataset.

The results obtained show the model's excellent capability to correctly categorize samples from the Custom_Large_8_1_1 dataset. Lastly, the Custom_Large_6_3_1 dataset has an accuracy of 82.29%.

The classification accuracy for the small dataset showed a slight improvement over the 8:1:1 split ratio, but not statistically significant. The difference between Custom_Small_8_1_1 and Custom_Large_8_1_1 showed a 17.46% increase in accuracy, indicating an overall improvement in the model's performance.

From the results obtained above, the top 3 best performing models were further evaluated under different conditions. The MobileNetV2, InceptionV3, and Custom models are assessed in this section under various situations such as variable light intensities, reduced picture quality, variation in image perspective angles, and occluded images for a test set of 160 photos with 40 images per scenario. Table 10 list the various scenarios used along with their descriptions.

Table 10 Description for the various scenarios adopted

Scenario	Description
Light Intensity	The light intensity is reduced.
Image Clarity	The image clarity is reduced to 100% using Windows' Photos Editor
Viewpoints	Images are rotated and flipped to various angles and orientations ranging from 0 to 180 degrees
Occlusion	For each category, images with a noisy background are selected

4.2.1 MobileNetV2

Table 11 shows the results of testing the test set under various situations on the MobileNetV2 model. It can be observed that the MobileNetv2 model works quite effectively in varied conditions, particularly in low light, with a classification accuracy rate of 0.975 with just one misclassified image.

Table 11 Results for MobileNetV2

Scenarios	No Images in test set	No of Images correctly classified	Classification Accuracy Rate
Light Intensity	40	39	0.975
Reduced Image Clarity	40	37	0.925
Different Viewpoints	40	31	0.775
Occlusion	40	34	0.850

4.2.2 InceptionV3

Table 12 depicts the results achieved while testing the InceptionV3 model under various conditions. Overall, the InceptionV3 model performed highly in identifying the images under different conditions. In every scenario, the model obtained more than 85% accuracy in classification accuracy.

Table 12 Results for InceptionV3

Scenarios	No Images in test set	No of Images correctly classified	Classification Accuracy Rate
Light Intensity	40	39	0.975
Reduced Image Clarity	40	37	0.925
Different Viewpoints	40	35	0.875
Occlusion	40	35	0.875

14

4.2.3 Custom model

The results for the custom model are tabulated in Table 13.

Table 13 Results for Custom Model

Scenarios	**No Images in test set**	**No of Images correctly classified**	**Classification Accuracy Rate**
Light Intensity	40	28	0.70
Reduced Image Clarity	40	33	0.825
Different Viewpoints	40	26	0.65
Occlusion	40	32	0.80

Table 13 shows that the Custom Model is less performant than the other two models, MobileNetv2 and InceptionV3. There have been several misclassified images with a classification accuracy rate of 0.65 in the scenario for 'Different Viewpoints". However, the custom model performs relatively well when there is a reduction in image clarity. Therefore, from the results gathered from Table 11, 12 and 13, it can be concluded that the InceptionV3 model is the one with the highest classification accuracy rate followed by MobileNetv2 and the Custom Model. Since the top three models perform well in diverse light conditions, generating accurate predictions and can also generalize across multiple angles, hence these models are able to provide reliable and consistent performance when it comes to diverse types of disaster-related images, proving their capacity to generalize variety in images.

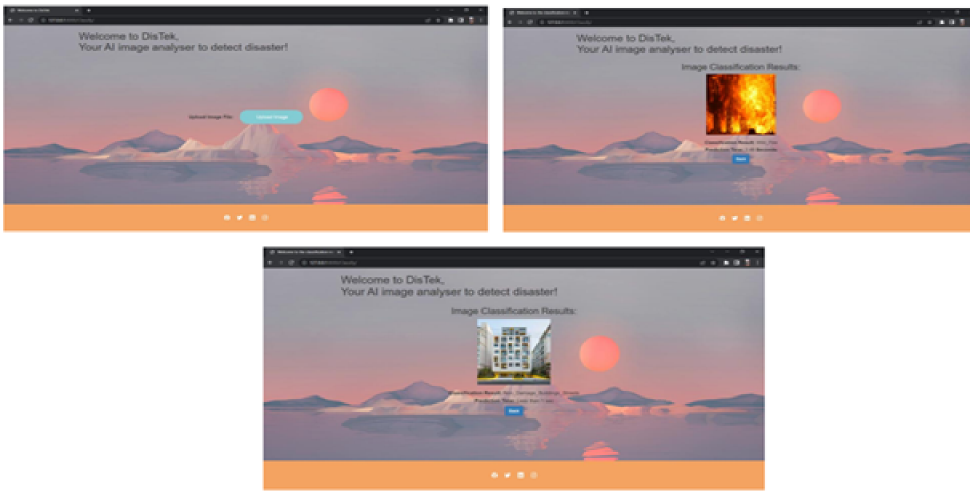

Fig. 6 Classification results from the web application

15

As seen in Figure 6, the user may select and upload an image by clicking the "Upload Image". The user can then click on the predict button to carry out the classification process. As seen by two of the snapshots in Figure 6, the model has effectively classified a picture of a building and a picture of a wildfire, demonstrating its ability to distinguish between disaster pictures and those that do not display any damage.

5 Conclusion

In this paper, CNN pre-trained models such as MobileNetv2, VGG16, and InceptionV3 were trained along with a custom neural network model to identify disasters based on their characteristics. The models were trained on a customized dataset called "Customized Disaster Dataset" with ten classes with three of them being non-damage classes. This paper generated two versions of the Customized Disaster Dataset: a small dataset with 350 pictures per class and a large dataset with 700 images per class using two different split ratios. The experiments were carried out to determine the most effective model in terms of performance and robustness. The InceptionV3 model performed well on the large dataset with a split ratio of 8:1:1, achieving the highest accuracy of 96.86%. It accurately classified most images in various scenarios. This research demonstrates the potential of deep learning models for automating disaster classification processes. The InceptionV3 model was then integrated into a web application to automate the disaster classification process, making it easier for first aid responders. In the future, the system can be further enhanced to support both the upload of videos for classification and real-time disaster classification. The difficulty faced during this research work was that for models having longer training time has posed a problem when utilizing the Google Colab since access to the GPU was revoked numerous times and the model had to be trained again and again.

Funding

No funding was received to assist with the preparation of this manuscript.

Conflict of interest

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial in-terest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

[1] World, 2022 Disasters in Numbers: 2022 Disasters in Numbers - World. https://reliefweb.int/report/world/2022-disasters-numbers. Accessed on March 17, 2023 (2023)

[2] Lassa, J.A.: Roles of non-government organizations in disaster risk reduction. In: Oxford Research Encyclopedia of Natural Hazard Science, (2018)

16

[3] Asif, A., Khatoon, S., Hasan, M.M., Alshamari, M.A., Abdou, S., Elsayed, K.M., Rashwan, M.: Automatic analysis of social media images to identify disaster type and infer appropriate emergency response. Journal of Big Data **8**(1), 83 (2021)

[4] Shrestha, Y.R., Krishna, V., Krogh, G.: Augmenting organizational decision-making with deep learning algorithms: Principles, promises, and challenges. Journal of Business Research **123**, 588–603 (2021)

[5] Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., *et al.*: Pre-trained models: Past, present and future. AI Open **2**, 225–250 (2021)

[6] Amit, S.N.K.B., Shiraishi, S., Inoshita, T., Aoki, Y.: Analysis of satellite images for disaster detection. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 5189–5192 (2016). IEEE

[7] Liu, Y., Wu, L.: Geological disaster recognition on optical remote sensing images using deep learning. Procedia Computer Science **91**, 566–575 (2016)

[8] Dunnings, A.J., Breckon, T.P.: Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 1558–1562 (2018). IEEE

[9] Ofli, F., Alam, F., Imran, M.: Analysis of social media data using multimodal deep learning for disaster response. arXiv preprint arXiv:2004.11838 (2020)

[10] Asif, A., Khatoon, S., Hasan, M.M., Alshamari, M.A., Abdou, S., Elsayed, K.M., Rashwan, M.: Automatic analysis of social media images to identify disaster type and infer appropriate emergency response. Journal of Big Data **8**(1), 83 (2021)

[11] Zou, Z., Gan, H., Huang, Q., Cai, T., Cao, K.: Disaster image classification by fusing multimodal social media data. ISPRS International Journal of Geo-Information **10**(10), 636 (2021)

[12] Dinani, S.T., Caragea, D.: Disaster image classification using capsule networks. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2021). IEEE

[13] Hossain, E., Hoque, M.M., Hoque, E., Islam, M.S.: A deep attentive multimodal learning approach for disaster identification from social media posts. IEEE Access **10**, 46538–46551 (2022)

[14] Yamashita, R., Nishio, M., Do, R.K.G., Togashi, K.: Convolutional neural networks: an overview and application in radiology. Insights into imaging **9**, 611–629 (2018)

17

[15] Sanghvi, K., Aralkar, A., Sanghvi, S., Saha, I.: A survey on image classification techniques. Available at SSRN 3754116 (2020)

[16] Li, P., Zhao, W.: Image fire detection algorithms based on convolutional neural networks. Case Studies in Thermal Engineering **19**, 100625 (2020)

[17] Abraham, T., Todd, A., Orringer, D.A., Levenson, R.: Applications of artificial intelligence for image enhancement in pathology. In: Artificial Intelligence and Deep Learning in Pathology, pp. 119–148. Elsevier, ??? (2021). https://doi.org/10.1016/b978-0-323-67538-3.00007-5 . https://doi.org/10.1016/b978-0-323-67538-3.00007-5

[18] Kohir, V.: Essential TensorFlow and Keras Callbacks for your Neural Networks. https://kvirajdatt.medium.com/essential-tensorflow-and-keras-callbacks-for-your-neural-networks-54539244db39. Accessed on March 17, 2023 (2022)

[19] Brownlee, J.: Use early stopping to halt the training of neural networks at the right time. Machine Learning Mastery **9** (2018)

[20] Chen, B.: Early stopping in practice: an example with keras and tensorflow 2.0. Towards Data Science **28** (2020)

18