International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16, No.1 (Sep-24)

http://dx.doi.org/10.12785/ijcds/1601109

Image Classification Based on Disaster type Using Deep Learning

Anisha Coopen [1] **and Sameerchand Pudaruth** [2]

[1,2] *ICT Department, FoICDT, University of Mauritius, Moka, Mauritius*

Received 14 Dec. 2023, Revised 14 Jun. 2024, Accepted 15 Jun. 2024, Published 22 Sep. 2024

Abstract: The continued rise of global temperatures is causing a major climate crisis and this is leading to devastating and deadly natural disasters. People 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 work, 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. Seven categories of disaster were considered in this study. These were: cyclones, earthquakes, floods, droughts, landslides, wildfires and urban fires. Two existing datasets namely the Comprehensive Disaster Dataset (CDD) and the Natural Disaster Dataset (NDD) were customised into a single dataset which we named as the Customised Disaster Dataset (CDD). The Customised Disaster Dataset comprises of a total of ten classes, three of which are images which are not related to disaster. These three classes are: regular images of buildings and streets, wild forest and sea. Three pre-trained deep learning models such MobileNetV2, VGG16 and InceptionV3 were used to train the datasets to allow for further comparison with existing studies. Along with that, a customised neural network model was created and trained on the datasets. Different scenarios were devised to assess the top performing models. The InceptionV3 had the best classification accuracy of 96.86% when the trainable layers were set to false. We also obtained an accuracy of 96.86% with the MobileNetV2 model but this time the trainable layers were set to true. In this study, we have demonstrated the effectiveness of CNN models as a tool for the automatic classification of disaster-related images. Most studies have used only two categories (disaster-related and non-disaster related images) or are restricted to only one type of disaster (water-related, land-related, etc.) while in our studies we have used seven categories of disasters. However, the accuracy of the models may be less if the images are taken at night or when the weather conditions are very bad.

Keywords: Natural disasters, Convolutional Neural Networks (CNN), deep learning models, 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 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 (food, safe drinking water, shelter), sanitation, and medical treatment to the 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]. In such times, it is very important to ensure that critical services remain operational so that a rapid assessment of the situation can be made. Such assessments may include the ability to carry search and rescue operations for the immediate protection of lives, assessment of critical infrastructures such as transportation networks and building, how prepared is the community to respond to the crisis, whether there are any environmental hazards that can occur, and what technologies could best be used in response. Disaster response is critical in alleviating the immediate and long-term impacts of such calamities. Continuous investment in preparedness, response strategies, and technological advancements is essential to protect communities and foster resilience.

During disaster response, technologies such as Artificial

E-mail address: anisha.coopen@umail.uom.ac.mu, s.pudaruth@uom.ac.mu

Intelligence, deep learning and machine learning can be used to assist first responders in quickly analysing vast quantities of visual data which can be obtained from drones and satellites. This can help to locate the most affected regions more precisely thereby prioritising their response actions so that the most impact areas receive the first attention [2]. Image classification can also help to deploy and allocate resources more efficiently. This can help the responders to identify both safe and hazardous routes. Rescue teams can use such images to identify survivors or potential survivors in order to save their lives. Preparation is an important element in responding to disasters. Images from past disasters can also be used to train first responders and to develop best practices to react to real scenarios in order to reduce the overall impact of natural disasters.

However, manually filtering all disaster-related postings among other irrelevant posts such as random movies and advertisements is very challenging for a large number of reasons. Firstly, this is a time consuming process as collecting and analysing a large number of images can take a significant amount of time [3]. This can lead to delays in response which can worsen the impact of the disasters. If the disaster has occurred on a very large scale, using manual methods would be highly impractical. Important elements from images may be missed or wrongly interpreted which can lead to an inadequate response. Remote or dangerous areas are difficult to reach and therefore manual methods would not be practical in such cases. Properly analysing disaster images requires some training and such skilled personnel may not be available at that time. Working during natural disasters is physically exhausting and stressful and this can lead to human error in the manual assessments.

To ensure an efficient emergency response, it is critical to first classify the images according to their disaster type. This has resulted in the need to use deep learning algorithms to automate the disaster image classification process [3]. Deep learning is a branch of Artificial Intelligence which can be used to understand text, images and videos. In the context of disaster response, this technology can be used to analyse images received from different sources such as drones, satellites or social media. Since a lot of historical data already exists, these can be used to create robust deep learning models which can then be deployed in times of natural disasters. Deep learning models can be used to quickly analyse large quantities of images which can enable first responders to act quickly. Moreover, since this is an automated system and offers reasonably high accuracies especially for the classification of images, it reduces subjective assessments and biases when the same is done by humans.

This study 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 study 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 forests.

This research encompasses 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 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 40,000 picture patches from Google Earth aerial photographs, they 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 landslides. 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 classifier was applied. Google Earth remote sensing images were used in the evaluations. The approach suggested 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 high performance 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 the InceptionV1-OnFire. 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 process frames at a rate of 17 frames per second. However, this 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, they employ two parallel architectures, including the VGG16 model for image classification and a customised CNN model for text. The multimodal architecture uses a shared dense layer to aggregate data from both modalities and softmax to predict the 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. They also used the CrisisMMD dataset. 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 disaster images to the taxonomy and chose relevant emergency response categories. With YOLOv4, the technique obtained a classification accuracy of 96% [10].

Zou et al. [11] investigated how to identify disaster images from social media using the VGG16 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 an accuracy of 92.6% against an accuracy of 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 the CrisisMMD dataset to compare capsule network models to ResNet18 models both for in-domain and cross-domain situations. The results demonstrated that capsule networks performed better when the training datasets were small or imbalanced, outperforming the 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 disaster inference system from tweets that uses textual and visual information. 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 (ResNet50). 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 I provides a summary of the research papers.

This section has explored previous studies along with its techniques and approaches used for disaster image classification. This study aims to explore different deep learning models using existing disaster image datasets and integrate them into a single dataset for disaster image categorisation.

TABLE I. Summary of papers

References	Datasets	Classifier	Accuracy/F1-Score
Amit et al. [6]	Google Earth aerial images	CNN	80 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

3. Methodology

The main objective of this study is to identify the type of disaster from images. 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.

Figure 2 demonstrates the flowchart for the proposed system. The two datasets, namely the Comprehensive Disaster Dataset (CDD) and the Natural Disaster Dataset (NDD) are combined into a new dataset. Two variations of the dataset will be employed, one comprising 350 images and the other consisting of 700 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 utilised 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. The best performing model will be then integrated into a web application.

A. *Dataset and Preprocessing*

The Natural Disaster Dataset (NDD) and the Comprehensive Disaster Dataset (CDD) were integrated into another dataset named as the 'Customised Disaster Dataset' to focus on various disaster categories. To produce this customised 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. 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 added to the 'Earthquake' category. Figure 3 shows some one sample image for each of the seven disaster type categories from the Customised Disaster dataset while Figure 4 shows some sample images for the non-damage categories.

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

B. *Classification Phase Using Convolutional Neural Network*

Convolutional Neural Networks (CNNs) are a type of artificial neural network designed to recognise visual patterns from pixel images with minimal pre-processing. 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]. 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. Common convolutional neural network architectures include LeNet, AlexNet, ZFNet, GoogLeNet, VGGNet, and ResNet [15].

The convolutional layer uses filters to extract features from the input image. It involves identifying edges, colors, and textures [16]. 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 convert pooled feature maps into a flat vector hence preparing data for input to a Fully Connected Layer (FCL). The FCL is a neural network layer that performs classification

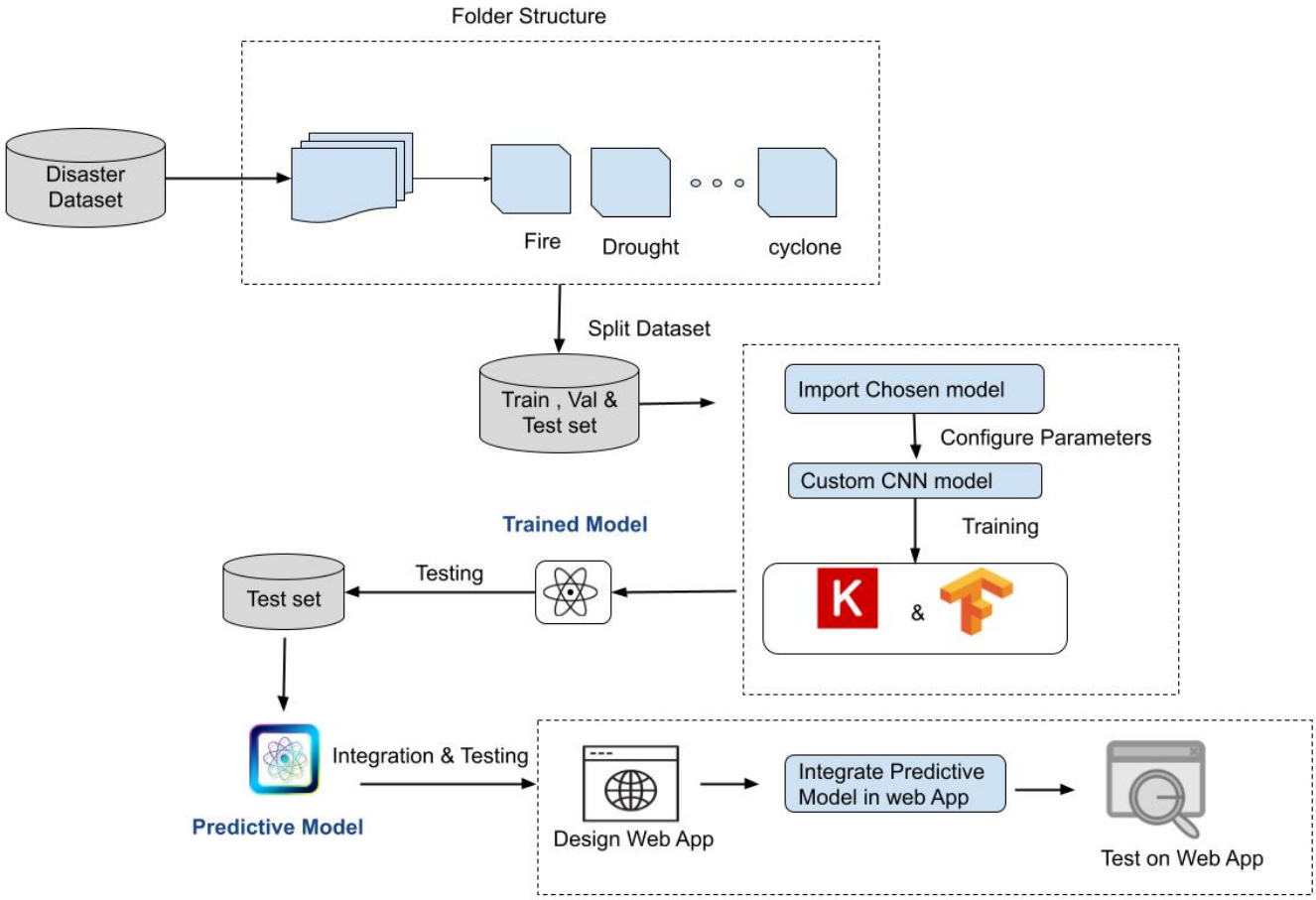

Figure 1. Proposed System Architecture

TABLE II. Split ratios

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

or identification based on extracted features. The 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 work. During the training phase, 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]. If the model achieves optimal performance sooner than expected, the Early Stopping function stops the training process. 'Patience' is one of the function's parameters. A number must be 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].

C. 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 III lists the libraries and tools utilised 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 for free. Furthermore, Google Drive may be readily

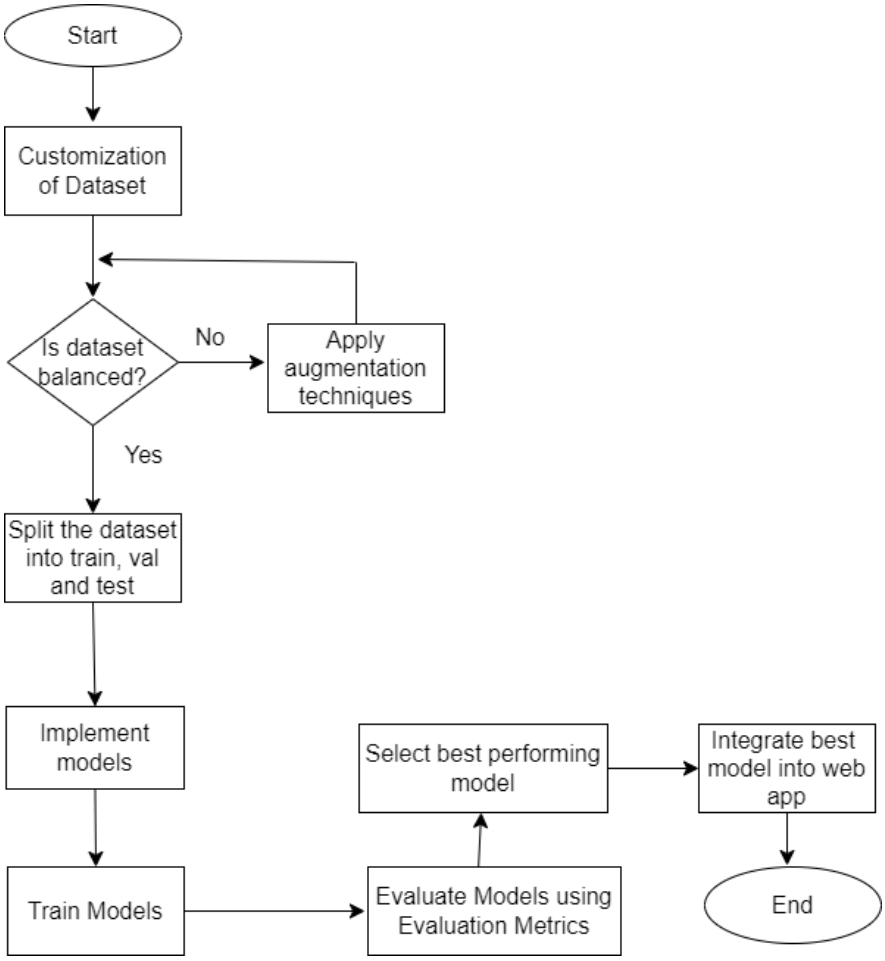

Figure 2. Flowchart for the proposed system

TABLE III. Tools and Technologies used

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

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

D. Feature extraction using pre-trained models

In this section, various pre-trained models were utilised 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 model to perform tasks like image classification. These models are trained on large datasets like the ImageNet and COCO datasets. This technique yields better results than training with limited data, as the model leverages learned features to perform a new 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

Figure 3. Sample Images from the Customised Disaster dataset

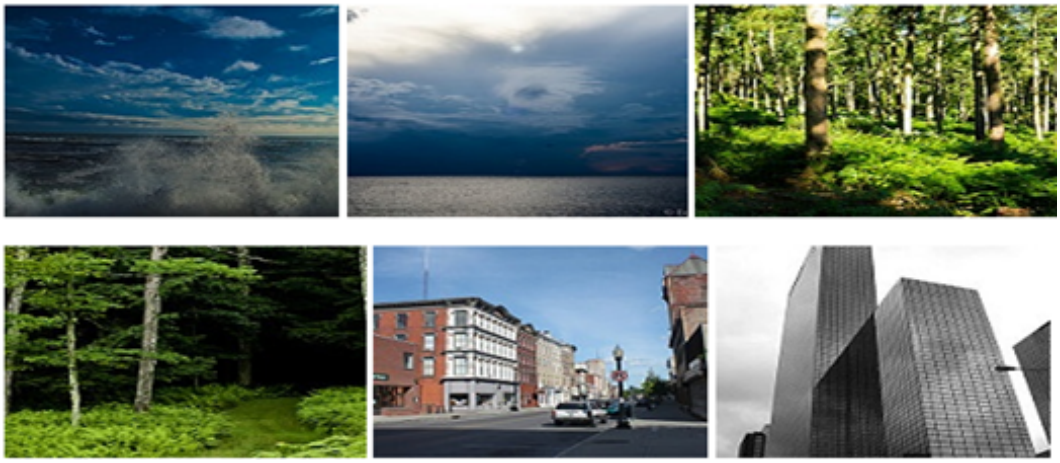

Figure 4. Sample Images for the Non-Damage classes

of the 'trainable.layer' is set to 'True', the weights will be updated during the training phase. Both scenarios are tested in this work. Different image sizes were utilised depending on the specific pre-trained model used. Table IV provides a summary of the models used.

The base layer of a pre-trained model has been enhanced with custom layers such as the 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 are converted to 1D vectors by the flatten layer. The custom model has 4,472,970 trainable parameters.

This study uses the callback functions, 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.

The computational complexity for a deep learning model depends on several factors. The computational complexity is directly related to the number of convolutional layers, pooling layers and fully connected layers that are present

TABLE IV. 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

in the forward and backward passes in the model. Moreover, the number of parameters used will also influence the time complexity. Thus, the size of the images used also has a direct impact on the computational complexity. Nevertheless, our proposed method is designed to balance accuracy and efficiency as the classification needs to be done in real-time and there is also the need to scale-up in order to classify thousands of images within minutes if such is available.

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.

A. *Results for Pre-Trained Models*

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.

1) *Large Dataset*

Tables V shows the results for the large dataset for the ratios 8:1:1 and 6:3:1.

For the MobileNetV2 when the trainable layers were set to 'False', an accuracy of 92.00% was obtained. When the trainable layers were changed to 'True', the accuracy increased by 4.86%. Additionally, for the VGG16 model, the accuracy has increased from 86.57% to 92.86% when the trainable layers were changed from 'False' to 'True'. The InceptionV3 model has achieved an accuracy of 96.86% with the trainable layers being set to 'True'. However, when setting the trainable layers to 'False', its accuracy decreased by 2%. For the MobileNetV2 when the trainable layers were changed from 'False' to 'True', the classification accuracy increased from 93.71% to 96.86%. However, for the VGG16 model, the classification accuracy decreased 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 tables set to 'False'. The accuracy remained the same when the trainable layers were set to 'True'. There is no general increase or decrease in the classification accuracy when the training data is reduced from 80% to 60%.

2) *Small Dataset*

Table VI shows the results for the small dataset for the ratios 8:1:1 and 6:3:1.

For the MobileNetV2 when the trainable layers were set to 'True', an accuracy of 93.43% was obtained. When the trainable layers were changed to 'False', the accuracy increased 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 set to 'False'. However, when setting the trainable layers to 'True', the accuracy decreased by 0.28%.The accuracy of the MobileNetV2 model was 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 3.71%. On the other hand, the VGG16 model's accuracy decreased from 85.71% to 74.86% when the trainable layers were changed from 'False' to 'True'. Setting the trainable layers to 'True' produced an accuracy of 93.43% for the InceptionV3 model while setting it to 'False' produced an accuracy of 94.86%. There is no significant increase or decrease in the classification accuracy when the training data is reduced from 80% to 60%.

The results reveal that setting the trainable layers to 'True' improves accuracy only for MobileNetV2 while this is not true for the other two models. This is perhaps because MobileNetV2 is more lightweight in terms of layers compared to the two other models. We also noticed that there was not much difference in accuracy when the dataset was increased from 3500 to 7000 images, suggesting that 350 images per category was adequate for the pre-trained models to learn the required features from the data.

B. *Custom CNN Model Results*

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

An accuracy of 71.11% was obtained using the Custom_Small_8_1_1 dataset. The model's accuracy decreased to 69.14% when the training data was reduced from 80% to 60%. An accuracy of 88.57% was obtained using the Custom_Large_8_1_1 dataset. Again, the accuracy decreased by a significant amount when the training data

TABLE V. Results for the large dataset

Model	Trainable	Accuracy (%) 8:1:1	Accuracy (%) 6:3:1
MobileNetV2	False	92.00	93.71
	True	96.86	96.86
VGG16	False	86.57	77.71
	True	92.86	65.71
InceptionV3	False	96.86	95.71
	True	94.86	95.71

TABLE VI. Results for the small dataset

Model	Trainable Layers	Accuracy (%) 8:1:1	Accuracy (%) 6:3:1
MobileNetV2	False	93.43	92.29
	True	95.71	96.00
VGG16	False	84.86	85.71
	True	84.29	74.86
InceptionV3	False	95.71	94.86
	True	95.43	93.43

TABLE VII. Results for the custom model

Model Name	Classification Accuracy (%) 8:1:1	Classification Accuracy (%) 6:3:1
Custom_Small_8_1_1	71.11	69.14
Custom_Large_8_1_1	88.57	82.29

was reduced. The difference in accuracy between the Custom_Small_8_1_1 and Custom_Large_8_1_1 models is 17.46%, indicating a significant overall improvement in the model's performance when the dataset has been doubled from 3500 to 7000 images. However, it is still quite far from the best performance of the pre-trained MobileNetV2 model which was 96.86%. The three best 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, rotation, and occluded images for a test set of 160 photos with 40 images per scenario. Table VIII lists the various scenarios used along with their descriptions.

Table IX shows the results from the test set under various situations on the MobileNetV2 model. It can be observed that the MobileNetV2 and the InceptionV3 pre-trained models work quite effectively in varied conditions, particularly in low light, with a classification accuracy of 97.5%, i.e., with just one misclassified image out of 40. Overall, the results indicate that the InceptionV3 model is most robust to disturbance in the images, especially with respect to rotation where the accuracy is still at 87.5%. The custom model delivers the worst performance in all the scenarios showing that it would be better to rely on pre-trained models for the identification of disaster type from images.

In terms of computational complexity, MobileNetV2 is designed for efficiency, making it particularly suitable for applications with limited computational resources such as mobile devices. The model contains approximately 3.4 million parameters, which is relatively low compared to more complex models such as VGG16 or ResNet50. This efficiency enables MobileNetV2 to achieve high-speed inference with reduced computational demands, making it ideal for real-time image classification tasks.

TABLE VIII. Description for the various scenarios adopted

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

TABLE IX. Results for the three best models

Scenarios	Number of images in the test set	Accuracy MobileNetV2	Accuracy InceptionV3	Accuracy Custom Model
Light Intensity	40	97.5	97.5	70.0
Reduced Image Clarity	40	92.5	92.5	82.5
Different rotation angles	40	77.5	87.5	65.0
Occlusion	40	85.0	87.5	80.0

Figure 5. A sample classification result from the web application

A web application was also implemented as part of this study in order to deploy the model. The InceptionV3 model was selected for deployment due to its robustness to noise. A user may select and upload an image by clicking the 'Upload Image' button. The user can then click on the 'Predict' button to perform the classification. Figure 5 shows that the model has correctly identified an uploaded picture as belonging to the wildfire category.

5. Conclusion

In this study, CNN models such as MobileNetV2, VGG16, and InceptionV3 were trained along with a custom neural network model to identify disasters. The models were trained on a customised dataset called the 'Customised Disaster Dataset' with ten classes with three of them consisting of non-disaster images. This study generated two versions of the Customised Disaster Dataset: a small dataset with 350 pictures per class and a large dataset with 700 images per class. The experiments were carried out to determine the most effective model in terms of performance and robustness. The MobileNetV2 and the InceptionV3 models performed well on the large dataset with a split ratio of 8:1:1 achieving the highest accuracy of 96.86%. In terms of robustness, we found that the InceptionV3 model delivered the best scores followed closely by MobileNetV2. The InceptionV3 model was then integrated into a web application to automate the disaster classification process. This research demonstrates the potential of deep learning models for automating the disaster images classification process. In the future, we intend to work on a model that can identify the disaster type from short videos.

References

[1] ReliefWeb, "2022 disasters in numbers," 2023, accessed: 2024-05-25. [Online]. Available: https://reliefweb.int/report/world/2022-disasters-numbers

[2] J. A. Lassa, "Roles of non-government organizations in disaster risk reduction," in *Oxford research encyclopedia of natural hazard science*, 2018.