NATURAL DISASTERS INTENSITY ANALYSIS AND CLASSIFICATION USING ARTIFICAL INTELLIGENCE

Submitted by

I.JONISHA

A.ANISHA

M.J.SHILJIA

K.SINDHU

TEAM ID:PNT2022TMID34563

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

PONJESLY COLLEGE OF ENGINEERING,

NAGERCOIL

ANNA UNIVERSITY: CHENNAI-600 025

NOV 2022

ABSTRACT

Natural disasters not only disturb the human ecological system but also destroy the properties and critical infrastructures of human societies and even lead to permanent change in the ecosystem. Disaster can be caused by naturally occurring events such as earthquakes, cyclones, floods, and wildfires. Many deep learning techniques have been applied by various researchers to detect and classify natural disasters to overcome losses in ecosystems, but detection of natural disasters still faces issues due to the complex and imbalanced structures of images. To tackle this problem, we propose a multilayered deep convolutional neural network.

The proposed model works in two blocks: Block-I convolutional neural network (B-I CNN), for detection and occurrence of disasters, and Block-II convolutional neural network (B-II CNN), for classification of natural disaster intensity types with different filters and parameters. The model is tested on 4428 natural images and performance is calculated and expressed as different statistical values: sensitivity (SE), 97.54%; specificity (SP), 98.22%; accuracy rate (AR), 99.92%; precision (PRE), 97.79%; and F1-score (F1), 97.97%. The overall accuracy for the whole model is 99.92%, which is competitive and comparable with state-of-the-art algorithms.

CHAPTER 1

1.1 Introduction

Natural disasters are inevitable, and the occurrence of disasters drastically affects the economy, ecosystem and human life. Buildings collapse, ailments spread and sometimes natural disasters such as tsunamis, earthquakes, and forest fires can devastate nations. When earthquakes occur, millions of buildings collapse due to seismological effects.

Many machine learning approaches have been used for wildfire predictions since the 1990s. A recent study used a machine learning approach in Italy. This study used the random forest technique for susceptibility mapping of wildfire.

Floods are the most devastating natural disaster, damaging properties, human lives and infrastructures. To map flood susceptibility, an assembled machine learning technique based on random forest (RF), random subspace (RS) and support vector machine (SVM) was used.

As the population is growing rapidly, people need to acquire land to live on, and as a result the ecosystem is disturbed horrifically, which causes global warming and increases the number of natural disasters. Populations in underdeveloped countries cannot afford damages disasters cause to infrastructures.

The aftermath of disasters leaves the humans in miserable situations, and sometimes the devastating effects cannot be detected; additionally, rescue operations cannot take place in most of the places and victims are unable to be identified due to geographical factors of the different areas. Disasters such as forest fires spread rapidly in dense areas, so firefighting is difficult to carry out; in this case, development of the strategy to predict such circumstances is crucial so that such disasters can be prevented beforehand.

As the technologies are continuously improving, aviation systems have begun adopting smart technologies to develop unmanned aerial vehicles (UAVs) equipped with cameras, which can reach distant areas to identify aftereffects of natural disasters on human life, infrastructure, and transmission lines by capturing images and videos.

Data acquired from these UAVs helps to identify the facial expressions of victims, the intensity of their situation and their needs in a post disaster scenario. It helps to take actions and carry out necessary operations to tackle devastating scenarios. Raw images obtained from camera-equipped UAVs are processed and neural network-based feature extraction techniques are applied to analyze the intensity.

A deep learning method for the reconstruction of two-dimensional cardiac magnetic resonance images was proposed to enhance the image data acquisition process. Cascade deep convolutional neural networks use a 10-fold method to reconstruct the feature map for the MR images. In this way, feature extraction sequence becomes very fast and it takes less than 5 to 10 s to extract the feature matrix.

Neural networks provide multilevel network architectures, where Convolutional Neural Networks (CNNs) are the most frequently implemented architecture as the direct input of multidimensional vector images, speech recognition, and image processing can be carried out with low complexity. CNNs efficiently perform feature extraction by denoising the images and removing interference and achieve highly accurate results.

The proposed multilayered deep convolutional neural network method works in two blocks of convolutional neural networks. The first block, known as Block-I Convolutional Neural Network (B-I CNN), detects the occurrence of a natural disaster and the second one, known as Block-II Convolutional Neural Network (B-II CNN), defines the intensity of the natural disaster.

1.2 Objective

Natural disasters not only disturb the human ecological system but also destroy the properties and critical infrastructures of human societies and even lead to permanent

change in the ecosystem.

Disaster can be caused by naturally occurring events such as

earthquakes, cyclones, floods, and wildfires

1.3 Scope of the project

1.Natural disasters generally constitute an emergency since they require immediate intervention due to their high impact on human health and safety.

2.The affect the normal functioning of working infrastructure, interrupting normal day activities and representing a risk for residents and workers in affected areas.

3.Artificial intelligence (AI), in particular machine learning (ML), is playing an increasingly important role in disaster risk reduction (DRR) – from the forecasting of extreme events.

4.The development of hazard maps to the detection of events in real time, the provision of situational awareness and decision support, and beyond.

5.The application of science and technology can substantially reduce losses of lives and property.

CHAPTER 2

2.1 Related Work Studies

Analyzing the intensity of natural disasters have gained significant attention in the current decade processing video sources is a feasible task due to convolutional neural networks (CNNs) which require high performance computational resources including graphics hardware and thus a smart and cost-effective fire detection network is proposed based on architecture of convolutional neural networks.

In convolutional neural networks, a model to detect wildfire smoke named wildfire smoke dilated dense net is used . The candidate smoke region segmentation strategy using an advanced network architecture. performed an evaluation of building clusters affected by earthquakes by exploring the deep learning method, which uses long short-term memory.

Natural disasters are unpredictable events Enhanced multilayer perceptron algorithm by including convolutional neural network implemented on rasp-berry pi to find out the victims of natural disasters using streaming cameras and to aid the evacuation team to rescue the disaster victims is applying automatic natural disaster detection to a convolutional neural network using the features of disaster from resized satellite images of landslide and flood detections.

Aerial images are able to show more specific and wider surface area of the ground, which helps the amount of information about the occurrence of disaster.

Social media networks such as Twitter where people share their views and information have been used as data sources to carry out disaster analysis

2.2 IMPLEMENTATION

Implementation of a module made it possible to successfully achieve the detection of an earthquake and its announcement by the government beforehand using based tweets.

As the tweets provide a significant amount of information implemented a convolutional neural network to perform feature extraction on informative as well as noninformative tweets, categorizing dataset containing tweets by an artificial neural network.

Social media is considered as a main source of big data, with data shared in the form of images, videos and text; after the occurrence of a disaster, social platforms are over flowed with different sorts of information which helps response teams to rescue the victims.

The majority of the data contain ambiguous contents which makes it difficult for the rescue teams to make the right decisions. The reviewed previous research based on convolutional neural networks using social media as a dataset and efficiently analyzed the effectiveness of big data from social media during disaster management.

Using the two-layer architecture of a convolutional neural network (CNN), an efficient feature extraction method was applied to the extended of dataset to compare three object recognition techniques: linear support vector classification, linear discriminant analysis and soft max.

More than 90% performance rates, with low standard deviation. The use of manpower is difficult in case of natural disaster occurrence in hilly areas, and continuous electric power supply is highly affected in these areas due to maintenance issues of transmission lines.

In this case a to pilot aerial equipment is used to gather images, and hidden content from aerial images needs to be identified in case of natural disasters such as landslides and heavy snowfall.

They removed the noise from raw aerial images and extracted disaster characteristic using the interframe difference technique; they implemented a convolutional neural network to analyze the type of disaster.

In some regions, disasters such as earthquakes are occur due to geographical factors. To locate the victim in a short time and locating the victims was made possible by using a dedicated ground station server and proposed victim detection framework based on convolution neural networks.

A simulation of real calamities was developed to test he frame work. Floods are a calamitous and remarkable disaster. Floods impact greatly on human lives economically and financially affecting nations. With the help of a neural network is possible to predict floods and save the masses from the disaster. By implementing a convolutional neural network and Modified Particle Swarm Optimization (MPSO).

2.3 DEVELOPMENT

Developed a deep learning approach to foresee the flood circumstances and identify the individuals one.

Proposed unmanned aerial vehicle image-based forest fire detection images of forest fires, stabilized the histogram and applied filters to smoothen the images before testing via convolutional neural network. Smoke detection was carried out using the local binary pattern (LBP) and support vector machine (SVM). Comparison of processed and raw images was made to test the effectiveness of the proposed strategy.

Forest fires drastically affect human lives and economic situations, and locating the victims in a short time is complex task. Convolutional neural networks make it possible to help firefighters to locate the location of victims by detecting density of smoke from image acquired from the unmanned aerial vehicle. CNN-based simple feature extraction.

Alex Net single deconvolution proposed approach helps develop real time fire monitoring system successfully improved response time, reduced power consumption, and optimized performance by using pipelining among network layers of a CNN, executed on a field-programmable gate array. As the spatial resolution of satellite images was too low, these images could not be used for wildfire detection.

Modified deep convolutional networks for high spatial resolution images, VGG-13 and Google Net, utilizing UAVs, a disaster forecasting system, web-based visualization system, alert system, and disaster response scenario database and achieved highly accurate results for early wildfire detection. It is a hectic job for a disaster management organization to assess the damage caused by natural disasters. Using images obtained from social media during and after the occurrence off our major natural disasters proposed a method by adapting CNN features based on event-specific and cross-events.

The proposed a method to produce motion information images computing optical flow vectors and employed a CNN; the proposed method efficiently differentiated normal and abnormal behaviors of people during a natural disaster.

The UMN and PETS2009 datasets were used to performed experiments. proposed a wave-shaped neural network (W-Net) to label the density of smoke in images, which is difficult task, so virtual dataset was created. Convolutional encoder decoder architectures were assembled to maximize the input for information extraction from smoke density images and W-Net was proposed.

The accuracy of the proposed system is improved by feeding previous encoding outputs to the decoding layers and combining them. Several data mining application were implemented using contents of social media; user generated content helps in disastrous events to gain vast amount of information. The CNN model is used to extract flood images from raw images and color filters are used to refine the desired detection.

The proposed system's efficiency and accuracy were tested on several datasets and it out performed other methods to give the highest results. The proposed multilayered convolutional neural network in this research is used to detect and classify the natural disasters, as explained in the methodology section. Moreover, a comparison of the some of the state-of-the-art methods.

2.4 Table-I

Ref no	Methodology Name	Outcomes	Weakness
1	Signal processing, image processing and statistical technique	More accurate prediction of natural disasters	Limited statistical parameters for prediction
2	Particle swarm optimization	Predict magnitude of earthquake	Work only for prediction on seismic dataset
3	Neural network	Predict magnitude of earthquake	Limited parameters used for prediction
4	Text mining, regular log mining technique	Detect earthquake with speed and	Depends on public feedback to detect earthquake

Ref no	**Methodology Name**	**Outcomes**	**Weakness**
		accuracy on seismological data	
5	Decision tree	Utilize some parameters to access the model for flood damage area detection	Parametric limitation for the detection of flood damaging regions
6	Artificial neural network, genetic algorithm and wavelet transfer technique	Sum-up good results as compared to the already existing techniques in the southeast Asia	Work for monsoon floods in June and September for specific regions in India for time series data
7	Support vector machine, naïve Bayes	Classify the natural disasters on various parameters	Limited for only early stages of natural disasters
8	Machine learning technique	Predict the land sliding with the accuracy rate of 75 to 95	More guid line for model selection for prediction large scale landslide
9	Neural network and back propagation	Prediction on past dataset	Dynamic prediction is very much crucial for this system
10	Clustering for multivariable time series	Proposed a dynamic clustering approaches for time series	Dynamic time series data required

Ref no	Methodology Name	Outcomes	Weakness
		analysis and self-optimize organizing mapping technique	for clustering process
11	Data mining technique	A real time desktop-based GUI system is designed to predict local storm	Use parallel computing process that takes various amounts of time to predict storm
12	Text mining technique	Develop a public platform to inform early tsunami prediction and information	Public feedback is compulsory for prediction process
13	Random forest, long short-term model	Evaluate the flood severity in terms of sensitivity, specificity and accuracy as 71.4%, 85.9%, 81.13%, respectively	Particle swarm optimization and other deep learning techniques can be used as a future work
14	A learning-based wildfire model	Proposed method can predict the short term spread of wildfire	Real time rate of wildfire spread is required for initial stage
15	Machine learning technique	The gradient boosting tree and CLIPER model used	Model is still weak to produce velocity sensitivities

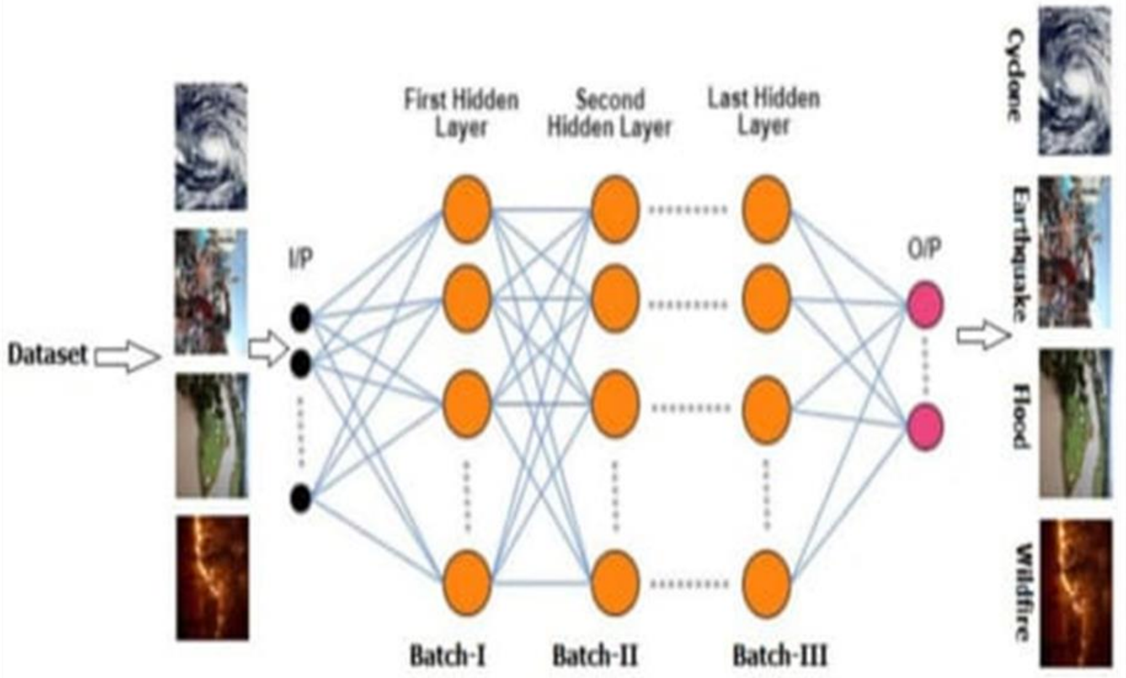
Ref no	**Methodology Name**	**Outcomes**	**Weakness**
		for cyclone prediction	
16	Machine learning technique with numerical weather prediction	The prediction method is used for China that shows significant improvement as compared to the traditional methods	Still lack symmetric parameters for numerical computations
17	Artificial neural network	A fully connected neural network for segmentation which is used for multivariable pattern recognition at different levels	It works on multivariable parameters rather than the pixel-by - pixel parameters

CHAPTER 3

3.1 Methodology

This section defines the overall method for natural disaster intensity analysis and classification based on multispectral images using a multi layered deep convolutional neural network. Moreover, this method consists of two blocks of a convolutional neural network. The first block detects a natural disaster occurring and the second one defines the intensity type of the natural disaster. Additionally, the first block consists of three mini convolutional blocks with four layers each, including an image input and fully connected layers. On the other hand, the second block also consists of three mini convolutional blocks with two layers each and includes an image input layer and fully connected layer. The overall flow of methodology is shown in Figure 3.1 and explained below.

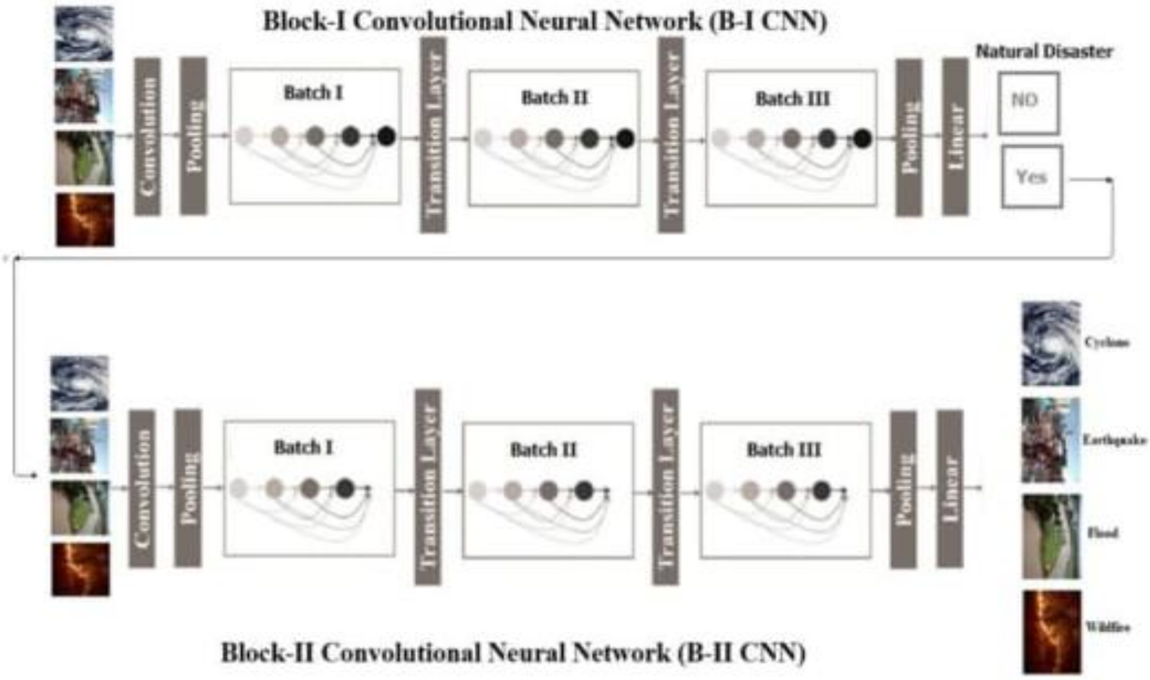
Figure 3.1



3.2. Block-I Convolutional Neural Network (B-I CNN)

According to block-I of the convolutional neural network, only a detection process occurred in this phase. However, this block also consists of three small batches having four layers each. Moreover, an image input layer and fully connected layers are present. Additionally, some parameters are also defined with learning rate 0.001 and epoch size 40. On the other hand, the convolutional layers use a filter size of 3×3, stride 1 and eight filters that increase in number from 16 to 32 for the second and third minibatches of convolutional neural networks.

Figure 3.2


Block-I Convolutional Neural Network (B-I CNN)
Convolution
Pooling
Batch I
Transition Layer
Batch II
Transition Layer
Batch III
Pooling
Linear
Natural Disaster
NO
Yes
Convolution
Pooling
Batch I
Transition Layer
Batch II
Transition Layer
Batch III
Pooling
Linear
Cyclone
Earthquake
Flood
Wildfire
Block-II Convolutional Neural Network (B-II CNN)

3.3. Block-I Convolutional Neural Network.

Block-I Convolutional Neural Network (B-I CNN) with Learning Rate = 0.001 and Epochs = 40

Layer Name and Batches		Parameters
	Image Input Layer	Height: 100, Width: 120, Channel: 3
Batch I:	Convolution Layer Batch Normalization Layer Relu Layer Max Pooling Layer	Filter size: 3 × 3, No. of filters = 8, stride = 1
Batch II:	Convolution Layer Batch Normalization Layer Relu Layer Max Pooling Layer	Filter size: 3 × 3, No. of filters = 16, stride = 1
Batch III:	Convolution Layer Batch Normalization Layer Relu Layer Max Pooling Layer	Filter size: 3 × 3, No. of filters = 32, stride = 1
	Fully Connected Layer	4 Classes

3.4. Block-II Convolutional Neural Network (B-II CNN)

The block-II convolutional neural network takes the output from the first block and finds the types of natural disaster with intensity. Moreover, this block also consists of three minibatches having three layers each with two extra layers such as image input and fully connected layers. Additionally, the same parameters as block-I have been defined for this block also.

3.5 Block-II convolutional neural network (B-II CNN)

Block-II Convolutional Neural Network (B-II CNN) with Learning Rate = 0.001 and Epochs = 30		
Layer Name and Batches		Parameters
	Image Input Layer	Height: 100, Width: 120, Channel: 3
Batch I:	Convolution Layer Batch Normalization Layer Max Pooling Layer	Filter size: 3 × 3, No. of filters = 8, stride = 1
Batch II:	Convolution Layer Batch Normalization Layer Max Pooling Layer	Filter size: 3 × 3, No. of filters = 16, stride = 1
Batch III:	Convolution Layer Batch Normalization	Filter size: 3 × 3, No. of filters = 32, stride = 1

Block-II Convolutional Neural Network (B-II CNN) with Learning Rate = 0.001 and Epochs = 30	
Layer Name and Batches	Parameters
Layer Max Pooling Layer	

3.6 Results and Discussion

The proposed multi layered deep convolutional neural network was simulated on the computer system with Core i7, Central Processing Unit (CPU) 2.8Ghz with 16 GB RAM in MATLAB 2018a and different types of results were calculated.

3.7 Dataset and Pre-processing

In our research, the dataset used was collected from PyImage Search readers, who used Google Images to collect the total number (4428) of images in different classes. The dataset was separated into four classes: cyclone, earthquake, flood and wildfire, with 928, 1350, 1073 and 1077. The dataset was preprocessed to remove the noise by using an adaptive histogram equalizer. The whole dataset was divided into three groups: training, testing and validation. In total, 60% of the dataset was used for training, 23% for testing and 17% for validation.

These percentages of the dataset were used to inform the machine on the percentage values of the dataset to be used for testing, training and validation purposes. The validation set was used In our research, the dataset used was collected from PyImage Search readers, who used Google Images to collect the total number (4428) of images in different classes.

The dataset was separated into four classes: cyclone, earthquake, flood and to count the number of epochs for the whole training process.

This raises the problem of the lack of data needed to train the algorithm properly. Conversely, small, imperceptible earthquakes occur daily, along the same fault lines from which high-intensity events originate and, moreover, they involve identical physics and mechanisms.

These “micro-earthquakes” therefore represent a useful source of untapped information in the quest to understand and predict earthquakes.

3.8 EVALUATION CRITERION

To evaluate the performance of the proposed multi layered deep convolutional neural network, uses a train–test validation schema. To train the whole model, the training dataset was used, while for the fine-tuning of model the validation set was used.

The performance of the whole framework was calculated on the basis of the test dataset. For the evaluation of the proposed model on the given dataset of classification for positive and negative values, four types of data were accrued.

True positive (TP), the number of correctly positive classified images; true negative (TN) the number of correctly negative classified images; false positive (FP), the number of incorrectly positive classified images; and false negative (FN), the number of images that are incorrectly classified as negative images. To calculate the performance of the model, the specificity (SP), sensitivity (SE), accuracy rate (RR), precision (PRE) statistical values were adopted as a criteria.

The score was used when a conflict occurred between accuracy and sensitivity to evaluate the performance The equations are given below.

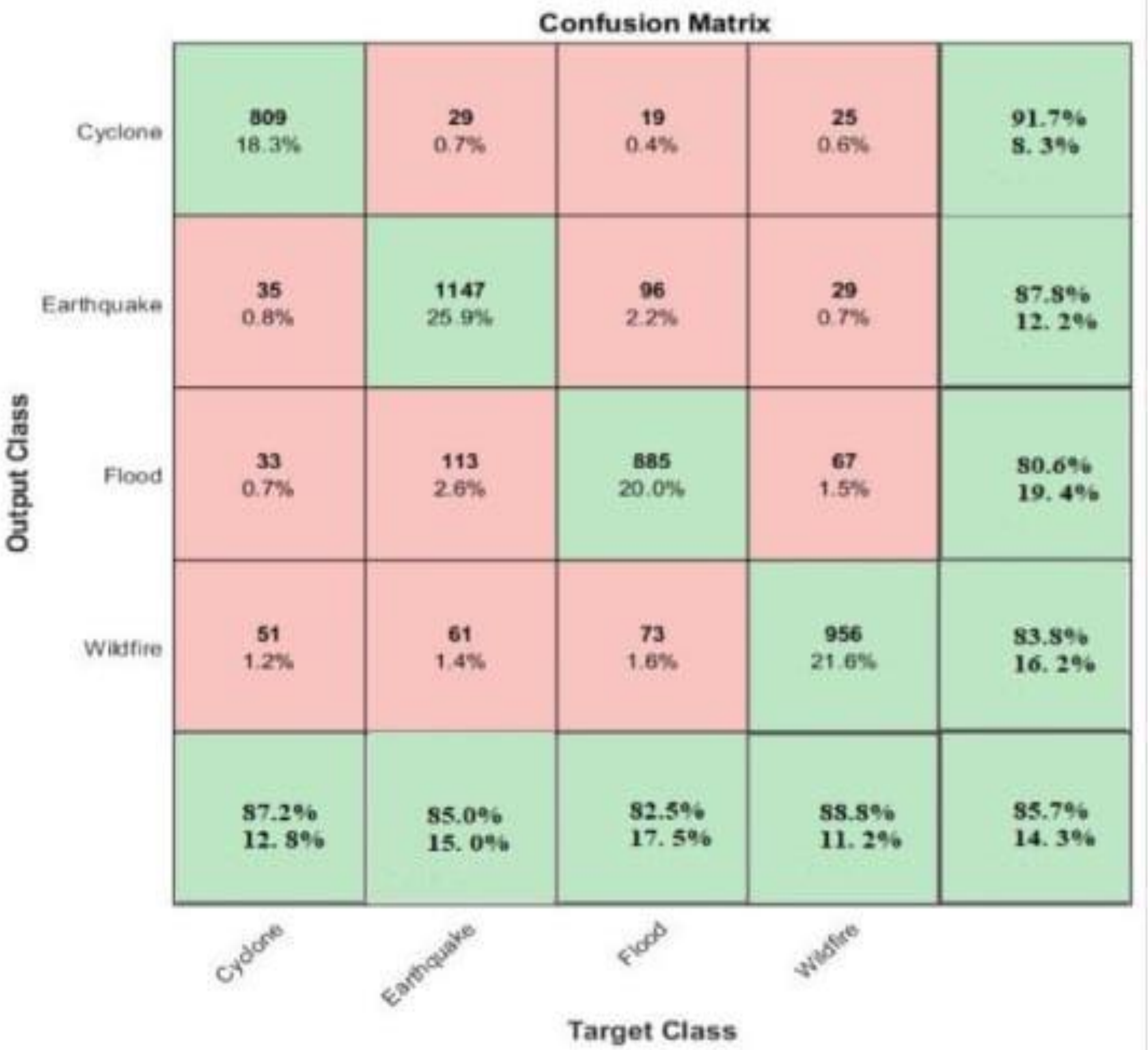
$$\textbf{Sensitivity (SE)} = \mathbf{TP/(TP+FN)}$$

Class matrix of disasters classification by using the proposed method on the testing dataset

Figure3.3

Confusion matrix of 4 -class of natural disaster classification by using the proposed method on the training dataset

Figure 3.4



3.9 EQUATION

The sensitivity (*SE*) in Equation (1) is the true positive measurement, the ratio of correctly identified values.

Specificity (SP) =TN/TP+FP (2)

Equation (2) shows the value of specificity (*SP*), the ratio of negatives which are correctly classified.

Accuracy Rate (AR) =TP+TN/TP+TN+FP+FN (3)

Equation (3) gives the value of accuracy rate (*AR*), which is equal to the actual measurement of specified values.

Precision (PRE) = TP/TP+FP (4)

The precision (*PRE*) in Equation (4) explains the proportion of closeness in measurement values.

F1−Score (F1) = 2(SE × PRE)/SE+ PRE (5)

The *F*1–Score (*F*1) in Equation (5) is the proportion of recall and precision which actually measure the model accuracy for the dataset.

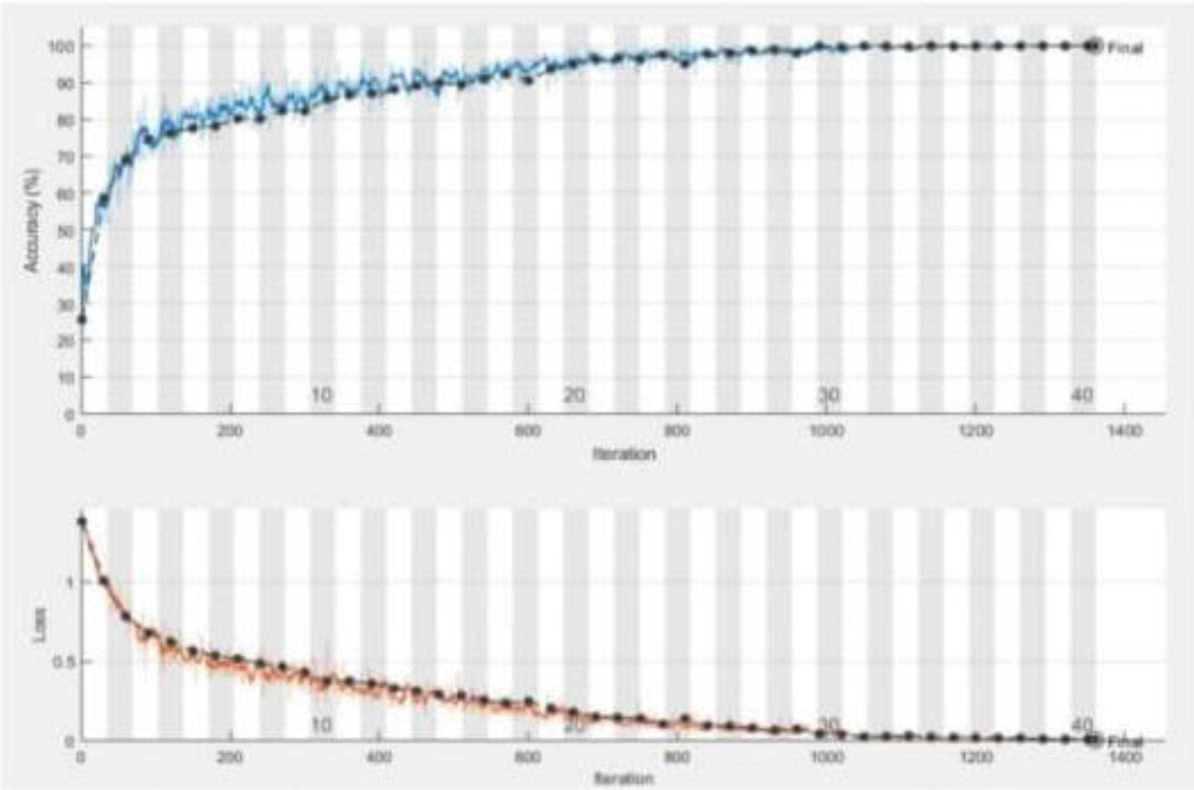
The graph is shows in training and validation accuracy rate, which is 99.92%, and also shows the validation and training loss.

The smooth line shows the training process and the dotted line shows the validation process for natural disasters dataset.

The calculated results in the shape of average statistical values: SE, 97.54%; SP, 98.22%; AR, 99.92%; PRE, 97.79%; and F1, 97.97% for the proposed model.

The obtained results are comparable with the state-of-the-art techniques and solved the complex queries related to analysis of the natural disasters.

Figure 3.5



CHAPTER 4

4.1IDEATION PHASE

LITERATURE SURVEY

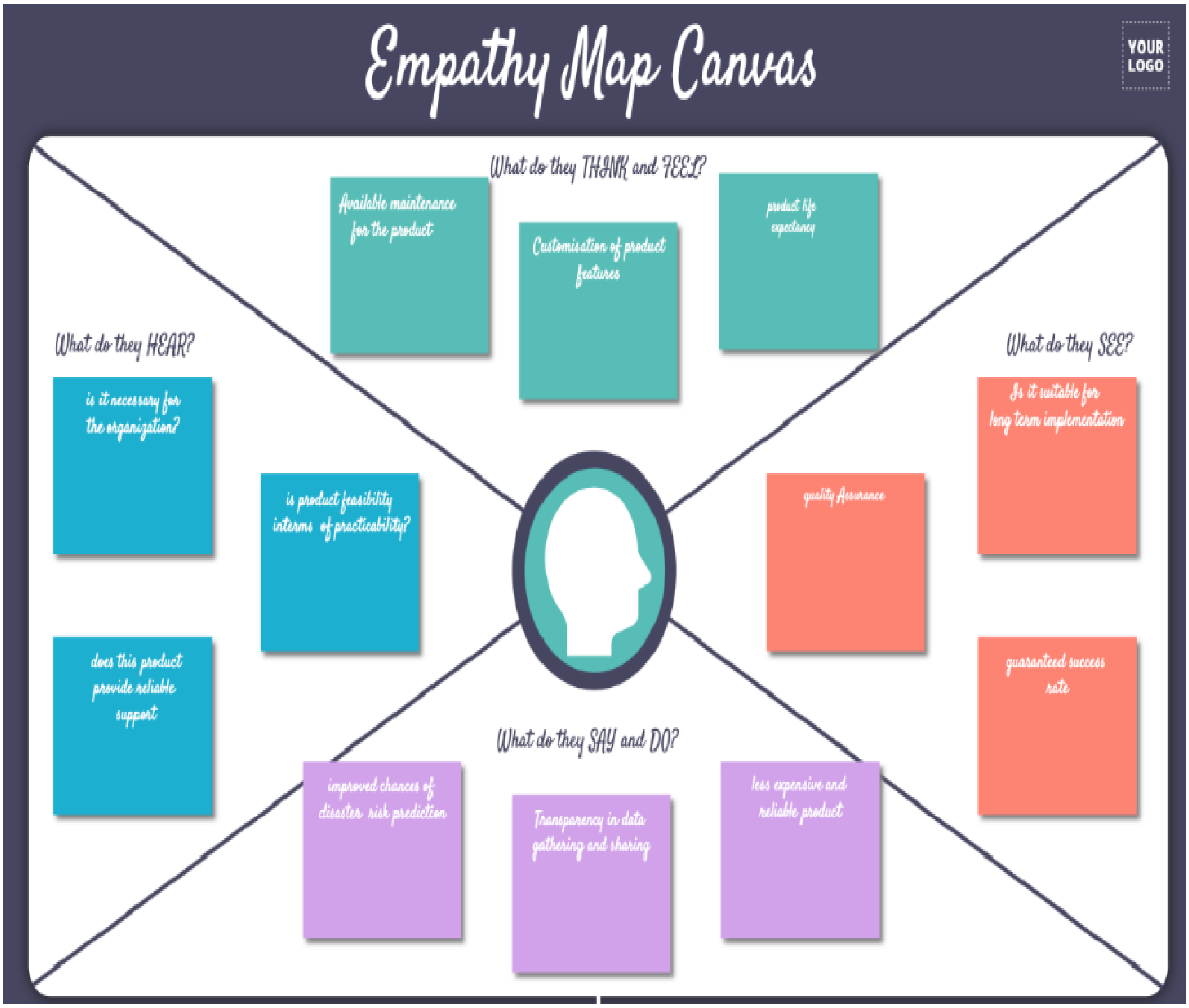
SNO	TITLE OF THE PAPER	DETIALS OF THE PAPER	OBJECTIVE	METHODOLOGY USED	TAKE AWAY
1	Land Surface Temperature retrieval using HJ- 1B/IRS data and analysis of its effect	2013 IEEE	To monitor pollution, ecosystem destruction and natural disaster on large-scale dynamically and around the clock	Generalized signal and channel alogrithm and parameter acquistion	In this paper, the generalized single-channel algorithm is utilized to achieve the LST from HJ-1B/IRS.
2	Study on Risk assessment model of urban Drought in Hilly Area of Central Sichuan Basin	2009 IEEE	It represents a model of risk assessment of urban drought which integrates hazard, exposure, vulnerability	Three methods are Natural disaster index method Weighted comprehensive evaluation method	In this paper it is used for mathematical model for the drought risk assessment and then use this model to calculate the

			and emergency response and recovery capability	Analytic Hierarchy Process	intensity of drought risk of Nanchong city in Hilly Area of Central Sichuan Basin from different perspective
3	Urban Damage Detection Using Decorrelation of SAR Interferometric Data	2002 IEEE	It indicates a fact that the building damage causes the interferometric decorrelation	It can be detected using interferometric decorrelation of ERS and JERS-1 SAR data.	In this paper, we progress in the study for quantitative discussion of the degree of decorrelation and the case of JERS-1 SAR interferometric data pairs to detect the damaged area by the earthquakes.
4	Quantifying change after natural disasters to estimate infrastructure damage with mobile phone data	2018 IEEE	It indicates that how mobility patterns are changing, in the post disaster time-frame, is crucial in order to settle	We describe the approach taken to work with aggregated CDR data	the relationship between the reach score changes and the damage index of the earthquake in urban areas, and it showed

			rescue center and send help to the most affected areas		that the correlation was negative on the day after the natural disaster
5	Spatio Temporal Analysis for Understanding the Traffic Demand After the 2016 Kumamoto Earthquake Using Mobile Usage Data	2018 IEEE	It mainly focuses on the effect of natural disasters on the population density transition	Analytical procedure and Spatial statistic methods are used.	We analysis that by using the SCICA and regression analysis captures the major travel demand patterns using the population density before the earthquake.
6	Degree of network damage: A measurement for intensity of network damage	2014 IEEE	To define degree of network damage (DND), a measurement used to classify the effect of a destructive event on network infrastructures,	A five-scale degree of network damage is developed to indicate the impact of disaster events on networks. We combine two network metrics to determine the degree of network damage from the	In this paper, we focus on a practical problem of providing an uniform criterion for accessing the impact of disasters on the network.

			human, and traffic flows	perspective of an ISP	

4.1.1EMPATHY MAP



4.1.2Problem Statements

Customer Problem Statement Template:

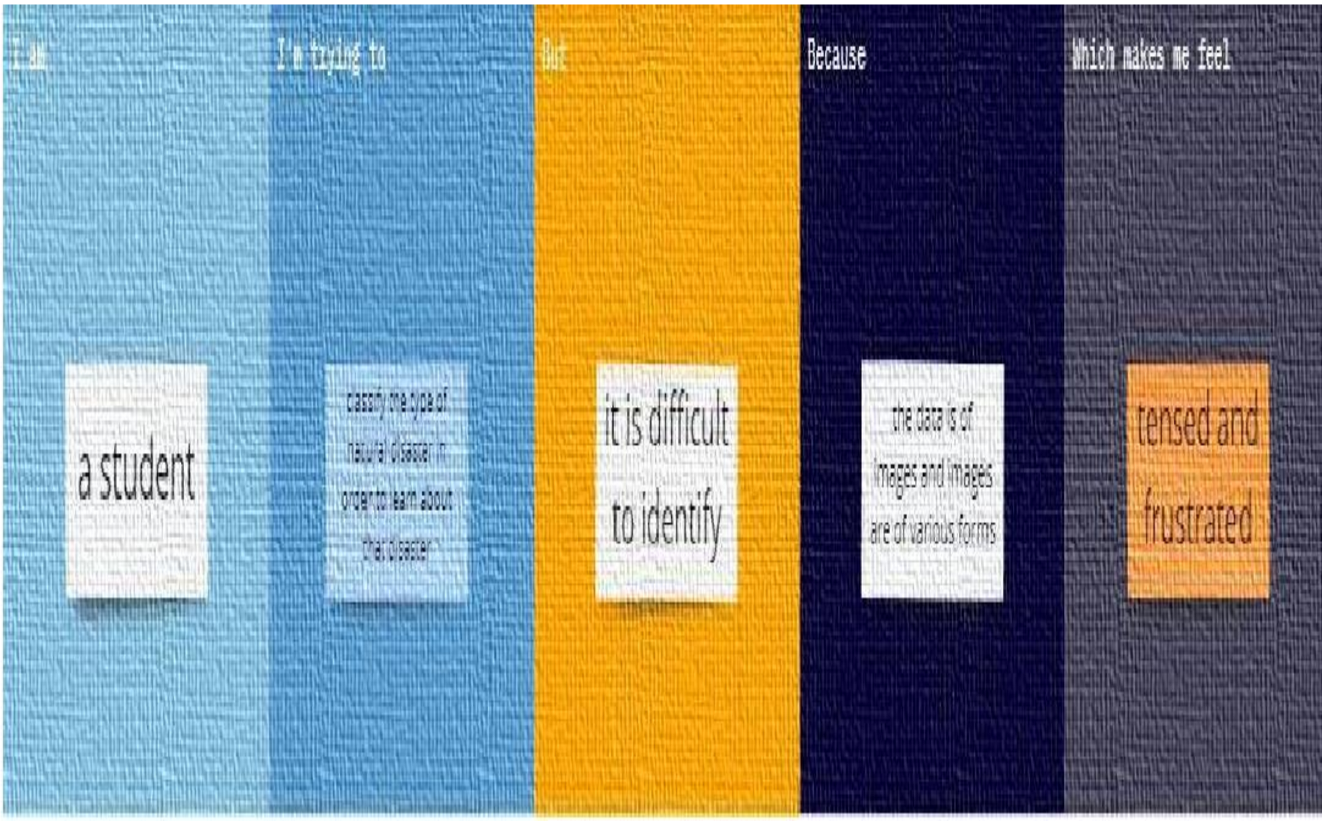
Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love. A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you’ll also be able to empathize with your customers, which helps you better understand how they perceive your product or service

I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers

4.1.3 Problem Statement for “Natural Disaster Intensity Analysis and Classification using Artificial Intelligence.”

PROBLEM STATEMENT 1:

PROBLEM STATEMENT 2:



4.2 PROJECT DESING PHASE I

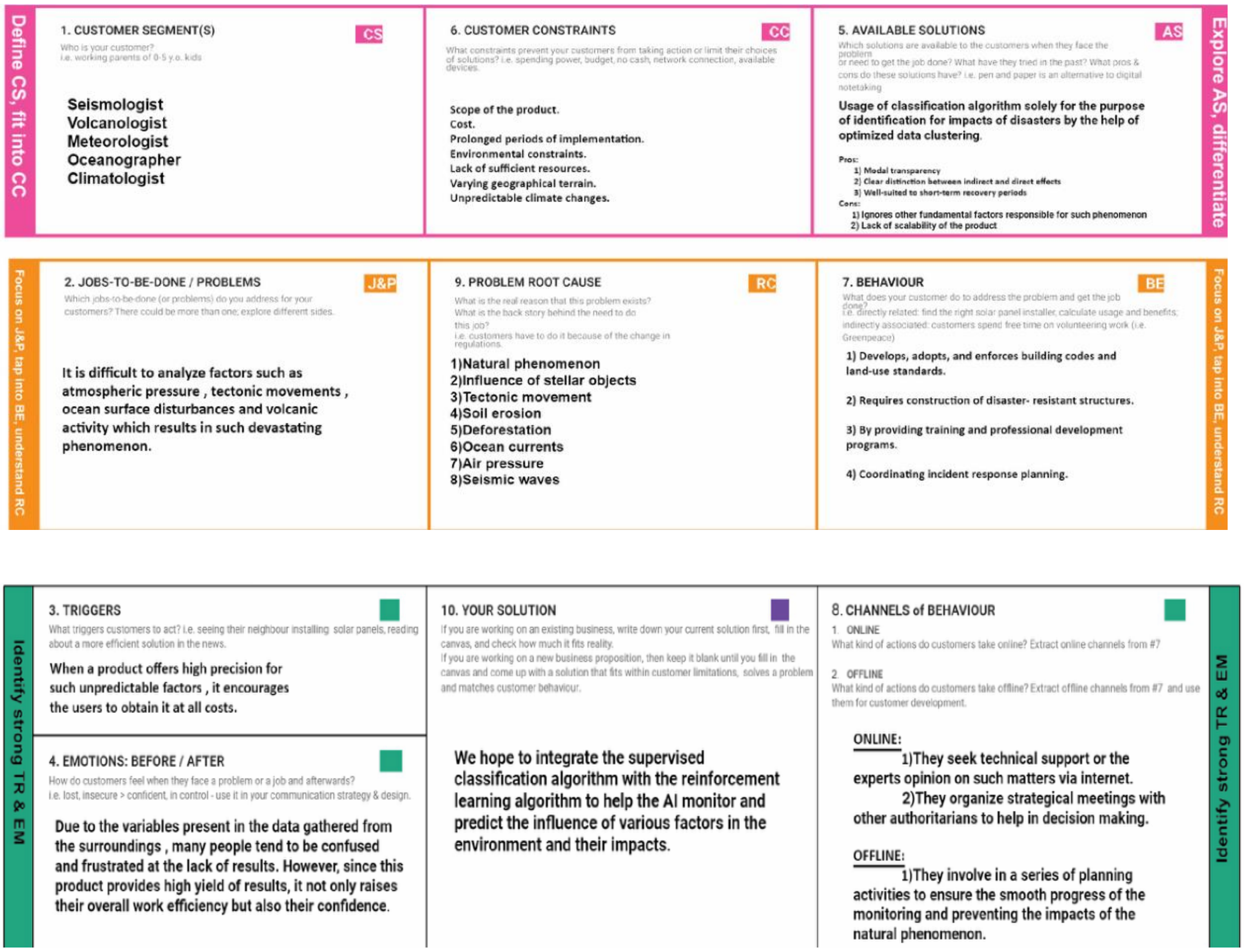
Proposed Solution Template

Project team shall fill the following information in proposed solution

SNO	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	To monitoring and predicting the disasters and its intensity of impacts on the region.
2.	Idea / Solution description	To use classification algorithm to identify the impacts of disaster.
3.	Novelty / Uniqueness	Usage of reinforcement learning algorithm to let the AI be self-sufficient and capable of gathering essential data on its own for prediction.
4.	Social Impact / Customer Satisfaction	This product will help in making crucial decision support at times of emergencies and also raise fundamental awareness of the impacts of disasters.

5.	Business Model (Revenue Model)	Revenue generated through Royalty payments, product license costs in department , research and educational platforms.
6.	Scalability of the Solution	Disintegration of geographical terrains into multiple provinces which can be interconnected as a grid to help alleviate its scale.

4.2.1PROBLEM SOLUTION FIT



4.2.2 Solution Architecture:

Solution architecture is a complex process – with many sub-processes that bridges the gap between business problems and technology solutions. Its goals are to:

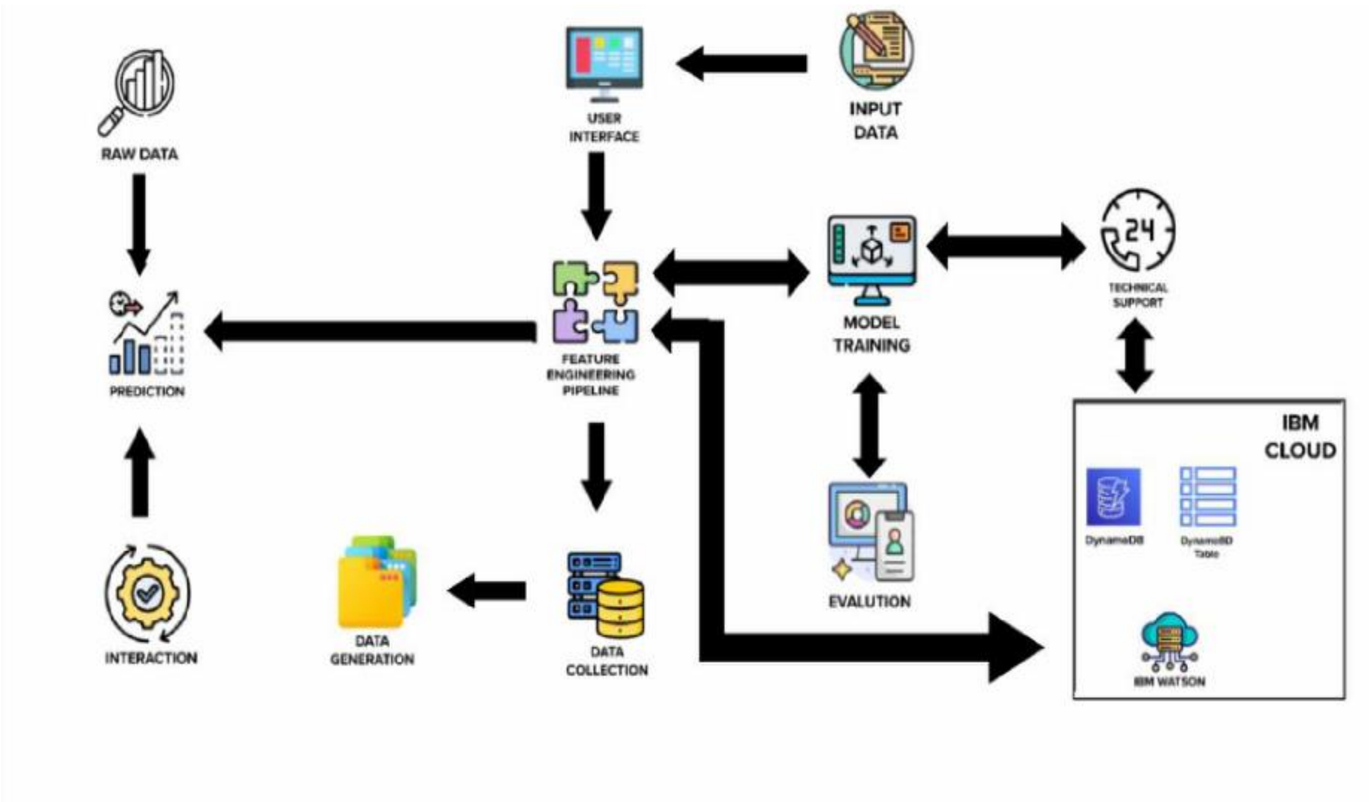
● Find the best tech solution to solve existing business problems.

● Describe the structure, characteristics, behavior, and other aspects of thesoftware to project stakeholders.

● Define features, development phases, and solution requirements.

● Provide specifications according to which the solution is defined, managed,and delivered.

Example - Solution Architecture Diagram



4.3 PROJECT DESING PHASE II

Customer Journey Map for “Natural Disaster Intensity Analysis and Classification Using Artificial Intelligence”:

Step-1: Goals and needs

Step 2: Journey Steps

Step 3: Journey Outcomes

4.3.1 Solution Requirements (Functional & Non-functional)

Following are the functional requirements of the proposed solution

FR NO	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration through the related website	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User profile	Personnel details
FR-4	Information about weather forecest	Through the related application
FR-5	Display the forecast of the place	Such as precipitation,humidity,wind

4.3.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution

FR NO	**Non-Functional Requirement**	**Description**
NFR-1	Usability	User friendly UI Friendly to the users
NFR-2	Security	There will be original and correct information
NFR-3	Reliability	The application must perform without failure

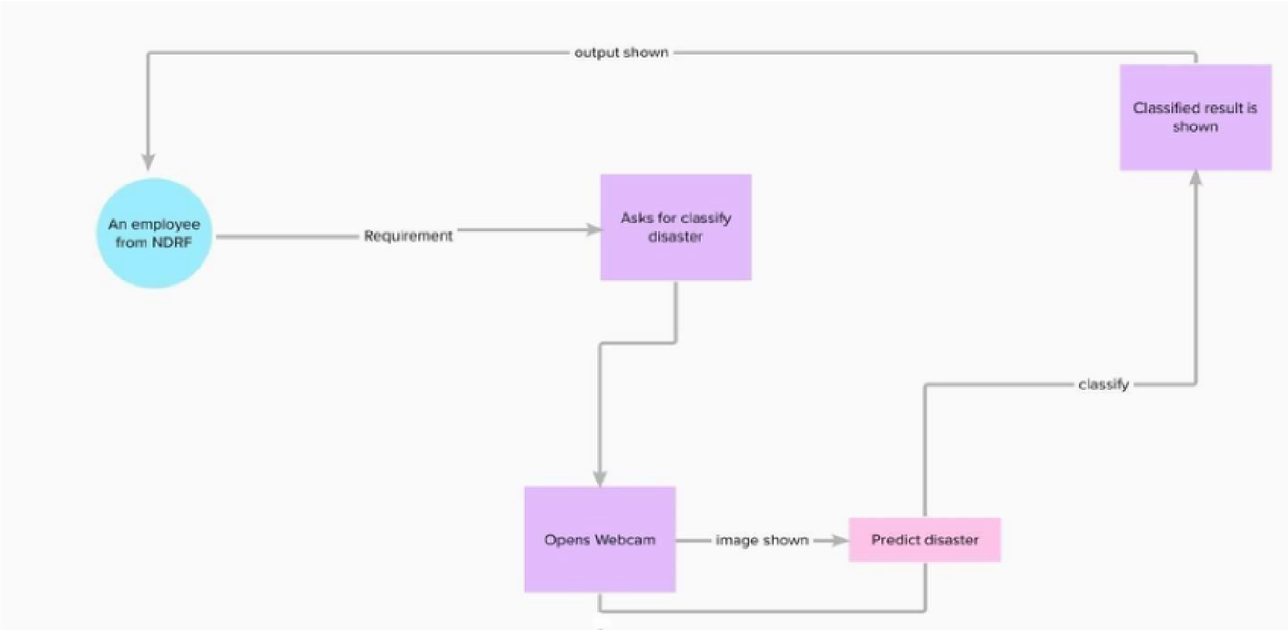
NFR-4	Performance	The landing page must support several users
NFR-5	Availability	Avaliable at all time
NFR-6	Scalability	It must be time saving and cost effective

4.3.3 Data Flow Diagram &User Stories

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

“Natural Disasters Intensity Analysis and Classification using Artificial Intelligence”:



4.3.4 User Stories

Here the list all the user stories for the project “Natural Disaster Intensity Analysis and Classification Using Artificial Intelligence”.

User Type	Functional Requirement (Epic)	User Story Number	User Story/ Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, registration should be done	Proper email id and password is accepted	High	Sprint-1Sprint-1
Customer	Area to be monitored	USN-2	As user,I can particularly select the area to be continuously checked and analyzed	The areas should be checked and selected without lapse.	Medium	Sprint-1
Customer	Safety	USN-3	As a user,I should monitor the device is in the secured place which should cover wide area	Safety measures should be done to prevent disaster	High	Sprint-2
Customer	Examination of Natural anamoly	USN-4	As a user,I should analyse the depth of the occurrence of the phenomena	I should monitor the factors which causes disaster	High	Sprint-1
Customer	Battery Backup	USN-5	As a user,I want to check the battery to prevent from power loss	Aware to always keep battery backup .Sometimes it may help in any crucial situations.	Low	Sprint-3

Customer	Algorithm to be used	USN-6	As a user,I should be very conscious in selecting required algorithm	Algorithm provides a correct understanding about the model designed.	Medium	Sprint-4
Customer	Internet Connectivity	USN-7	As a user,I should monitor the internet connection periodically	Strong internet connection is required in emergency situations.	High	Sprint-2
Customer	Social media	USN-8	As a user,I will be active in social media sites to know more updates about specific diasaster	Active in social media sites to know updates	Medium	Sprint-4
Customer	Prediction and analysis of data	USN-9	As a user,I can ale to predict and visualize data	Using algorithms and some visualization	High	Sprint-3
User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
Customer				techniques to predict disaster		
Customer(Web user)	Generating the possible outcome	USN-10	As a user, generating possible	Several disasters can be captured	High	Sprint-4

			output for the disaster occurrence	and output is shown		

4.3.5 Technical Architecture:

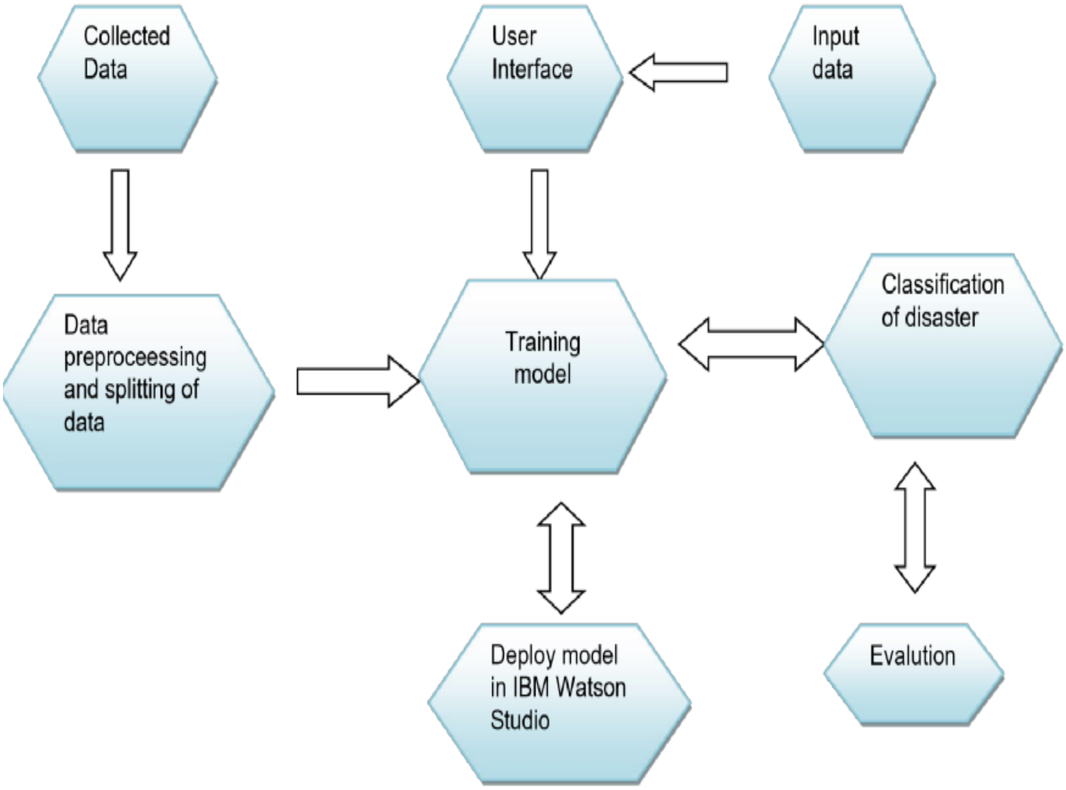


Table-1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	User interacts with application for the prediction of Any Natural disaster which will happen in future minutes.	HTML, CSS, JavaScript, Django, Python
2.	Disaster Prediction	This function is used to predict outcomes from the new trained data to perform new tasks and solve new problems.	Decision trees, Regression, Neural networks.
3.	Evaluation system	At monitors that how Algorithm performs on data as well as during training.	Chi-Square, Confusion Matrix, etc.
4.	Input data	To interact with our model and give it problems to solve. Usually this takes the form of an API, a user interface, or a command-line interface.	Application programming interface, etc

5.	Data collection unit	Data is only useful if it’s accessible, so it needs to be stored ideally in a consistent structure and conveniently in one place.	IBM Cloud, SQL Server.

Table-2: Application Characteristics

SNO	Characteristics	Description	Technology
1.	Open-Source Frameworks	An open source framework is a template for software development that is designed by a social network of software developers. These frameworks are free for public use and provide the foundation for building a software application.	Keras, Tensor flow.
2.	Authentication	This keeps our models secure	Encryption and Decryption (OTP).

		and makes sure only those who have permission can use them.	
3.	Application interface	User uses mobile application and web application to interact with model	Web Development (HTML, CSS)
4.	Availability (both Online and Offline work)	its include both online and offline work. As good internet connection is need for online work to explore the software perfectly. Offline work includes the saved data to explore for later time.	Caching, backend server.
5.	Regular Updates	The truly excellent software product needs a continuous process of improvements and updates. Maintain your	• Waterfall Approach • Incremental Approach • Spiral Approach

		server and make sure that your content is always up-to-date. Regularly update an app and enrich it with new features.	
6.	Personalization	Software has features like flexible fonts, backgrounds, settings, colour themes, etc. which make a software interface looks good and functional.	CSS

4.3 PROJECT PLANNING PHASE

Prepare milestone & activity list

Product Backlog, Sprint Schedule, and Estimation. Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priorit y	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	A.Anisha
Sprint-1		USN-2		1	High	Jonisha.I
Sprint-2		USN-3		2	Low	Shiljia.M. J
Sprint-1		USN-4		2	Medium	Sindhu. K
Sprint-1	login	USN-5		1	High	Jonisha.I
	Dashboard					

4.4.1Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 oct 2022	29 oct 2022	20	29 oct 2022
Sprint-2	20	6 Days	31 Nov 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

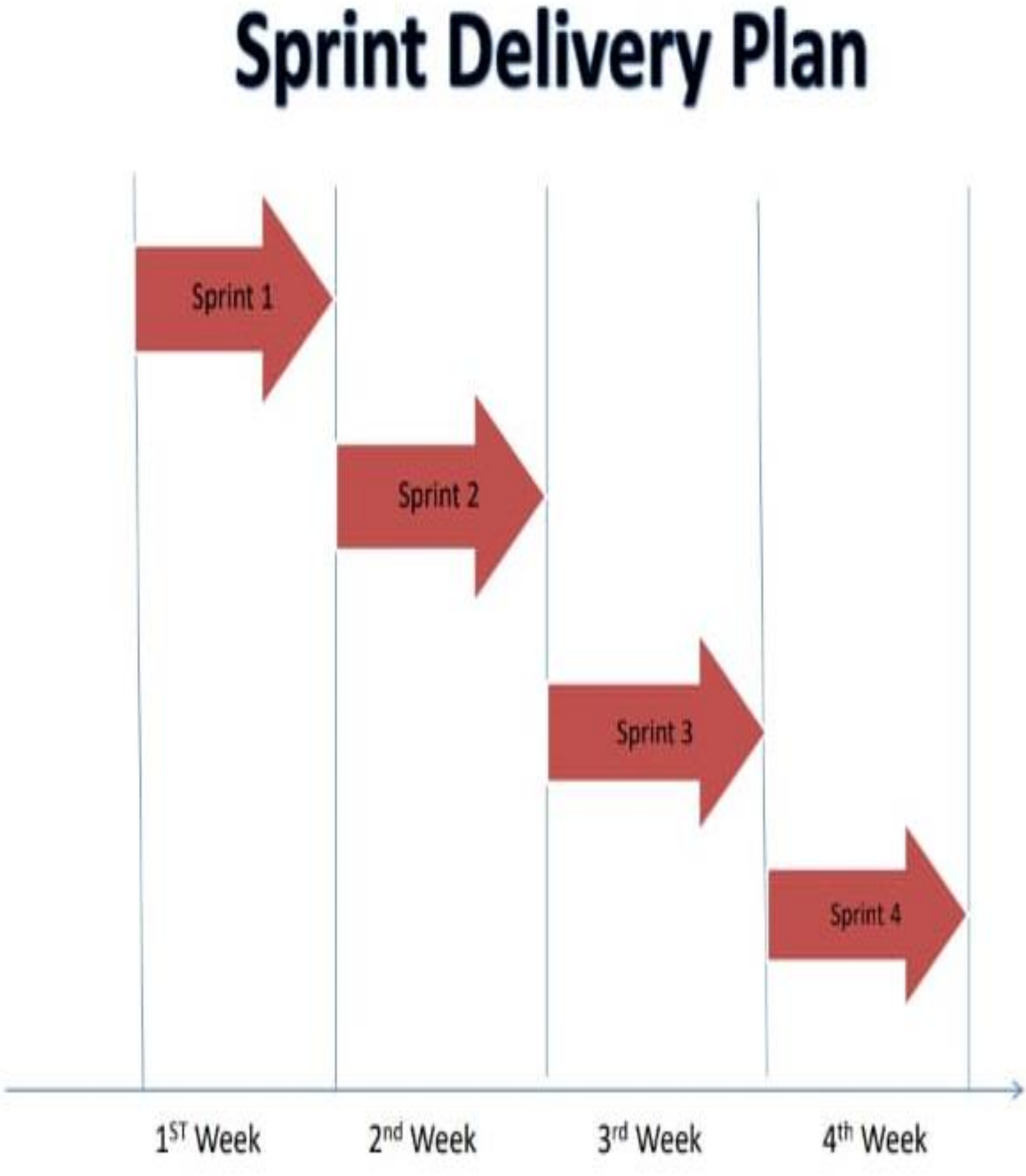
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

4.4.2 SPRINT DELIVERY PLAN



4.5 PROJECT DEVELOPMENT PHASE

4.5.1 Sprint -1 (DESIGN AN WEB APPLICATION)

As per Sprint delivery plan, Sprint-1 includes:

USER STORY NUMBER -1:

Design a web application which facilitates the image input.

CODE FOR WEB APPLICATION:

index.html:

```
{% extends "base.html" %} {% block content %}

<h2> Natural Disaster Intensity Analysis and Classification using AI</h2>

<div>

   <form id="upload-file" method="post" enctype="multipart/form-data">

        <label for="imageUpload" class="upload-label">

            Choose...

        </label>

        <input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">

   </form>


   <div class="image-section" style="display:none;">
```

```
<div class="img-preview">
            <div id="imagePreview">
            </div>
        </div>
        <div>
            <button type="button" class="btn btn- primary btn-lg "
id="btn-predict">Predict!</button>
        </div>
    </div>

    <div class="loader" style="display:none;"></div>

    <h3 id="result">
        <span> </span>
    </h3>

</div>

{% endblock %}
```

STATIC FILES :

main.css :

```
.img-preview {
    width: 256px;
    height: 256px;
```

```
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}

.img-preview >div {
    width: 100%;
    height: 100%;
    .img-preview >div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}

input [type="file"] {
    display: none;
}

.upload-label {
    display: inline-block;
    padding: 12px 30px;
```

```
    background: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s; background-size: 256px 2
cursor: pointer;
}

.upload-label:hover {
    background: #34495E;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

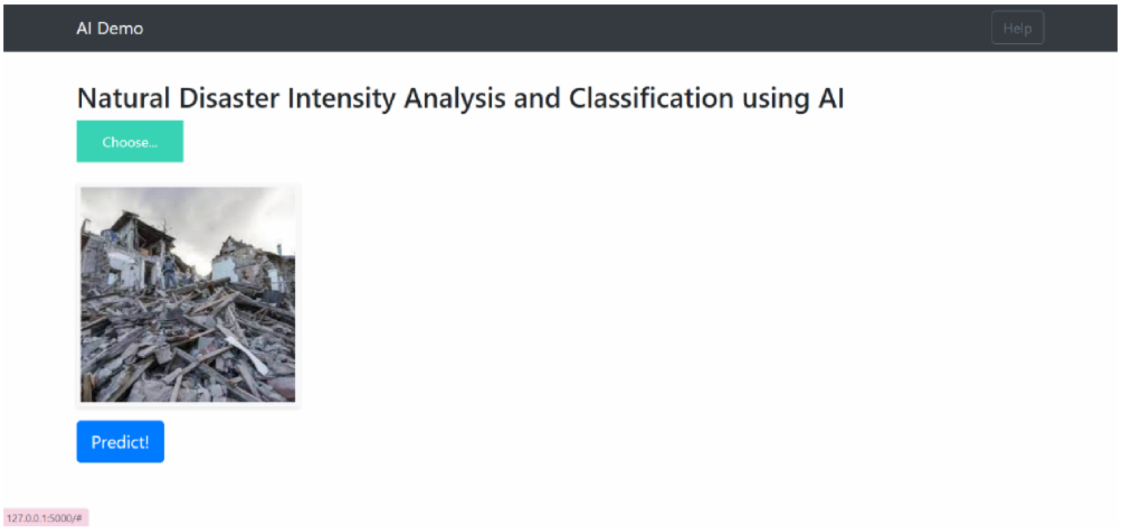
main.js :

```
$(document).ready( function () {
    // Init
    $( '.image-section').hide();
    $( '.loader').hide();
    $( '#result').hide();

    // Upload Preview
    function readURL(input) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            reader.onload = function (e) {
                $( '#imagePreview').css('background- image', 'url(' +
e.target.result + ')');
                $( '#imagePreview').hide();
                $( '#imagePreview').fadeIn(650);
            }
            reader.readAsDataURL( input.files[0]);

    }
    $( "#imageUpload").change(function () {
        $( '.image-section').show();
        $( '#btn-predict').show();
        $( '#result').text('');
```

```
        $( '#result').hide();
readURL( this);
    });

    // Predict
    $( '#btn-predict').click(function () {
        var form_data = new FormData($('#upload- file')[0]);

        // Show loading animation
        $( this).hide();
        $( '.loader').show();

        // Make prediction by calling api /predict
        $.ajax({
            type: 'POST',
            url: '/predict',
            data: form_data,
            contentType: false,
cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $( '.loader').hide();
                $( '#result').fadeIn(600);
```

```
                $( '#result').text(' Result:  ' + data);
                console.log( 'Success!');
            },
        });
    });


});
```

WEB APPLICATION DESIGN



Once we click the choose button, it will let the user to upload the image file present in his local directory.

4.5.2 Sprint -2 (DATA COLLECTION & IMAGE PRE-PROCESSING):

As per Sprint Delivery Plan, Sprint-2 includes:

USER STORY NUMBER -2:

The data required for building the model has to be collected from the Website.

USER STORY NUMBER -3:

Pre-process the collected data which is downloaded from the website it prevents the unnecessary variance or Bias problem.

Data

• The data consist of 4 classes Cyclone, Earthquake, Floods and Wildfire.

• The dataset is separated into training and validation set of 742 images in training set and 198 files in test set.

• All the class have almost equal number of training examples.

Image Data Generator

• Image Data Generator class can be imported from keras. preprocessing.image module.

• The attributes that has been applied to the image are:

rescale=1./255,

shear_range=0.2,

zoom_range=0.2,

horizontal_flip=True

• Once the image is pre-processed, convert the image into array and reshape it into the target size of 64,64.Create the batch size of 32.

• Apply the transformation on both train and test data. Given the pre-processed data to the model.

Image Pre-processing code:

train_datagen = ImageDataGenerator(

rescale=1./255,

shear_range=0.2,

zoom_range=0.2, horizontal_flip=True)

train_generator = train_datagen.flow_from_directory(

r"C:\Users\parameshreddy\Desktop\Nalayathiran_Project\dataset\train_set",

target_size=(64, 64),

batch_size=32,

class_mode='categorical')

test_datagen = Image Data Generator(

rescale=1./255,

shear_range=0.2,

zoom_range=0.2,

horizontal_flip=True)

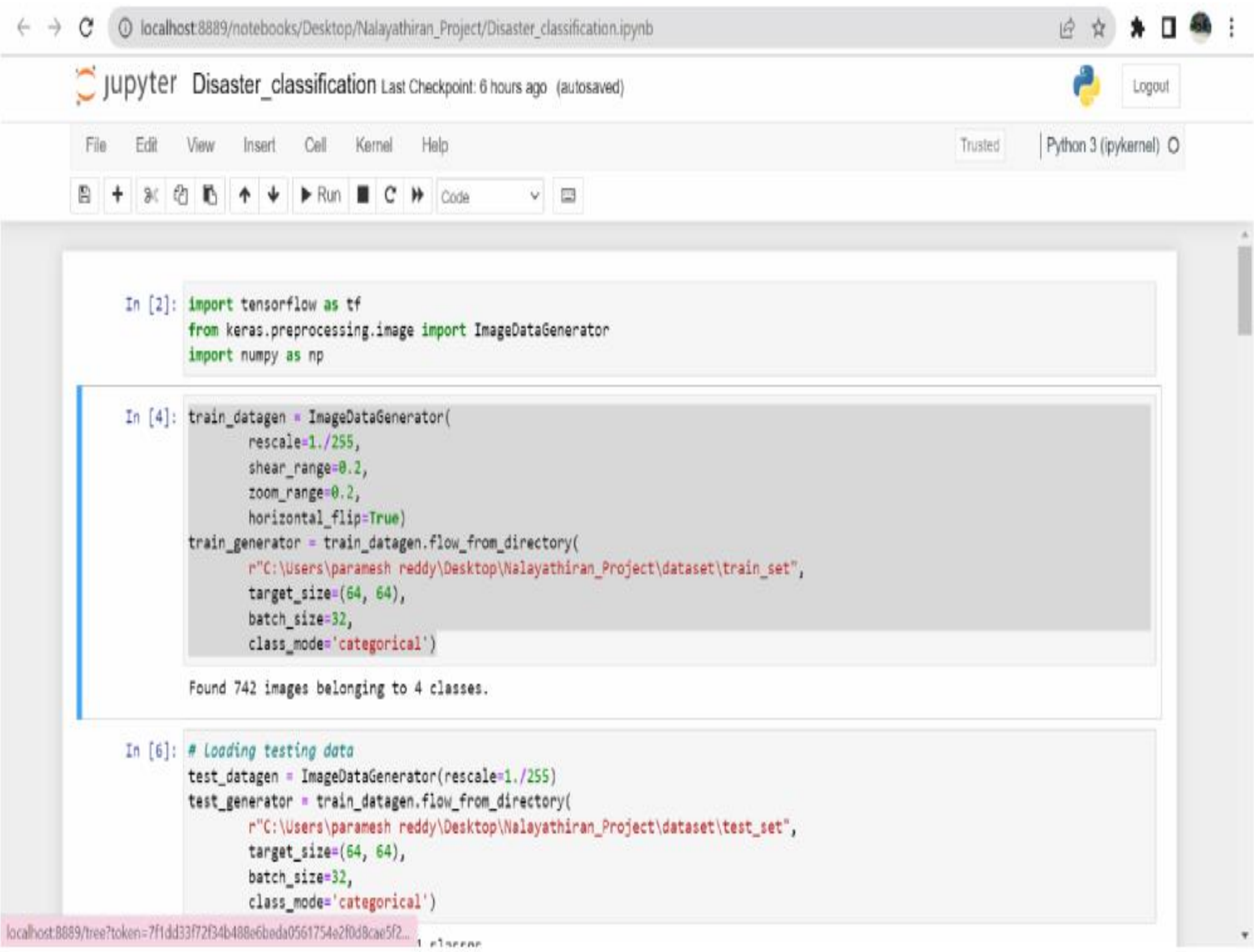
train_generator = train_datagen.flow_from_directory(

r"C:\Users\parameshreddy\Desktop\Nalayathiran_Project\dataset\test_set",

target_size=(64, 64),

batch_size=32,

class_mode='categorical')



4.5.3 SPRING -3 DETECTION AND ANALYSIS OF DATA:

After Testing and Training the model, data which given in dataset are analysed and visualised effectively to detect the Disaster Type. Using webcam, it can capture image or video stream of Disaster, to detect and analyse the type of Disaster

Inserting necessary libraries

```
In [1]: import numpy as np#used for numerical analysis
        import tensorflow #open source used for both ML and DL for computation
        from tensorflow.keras.models import Sequential #it is a plain stack of layers
        from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
        #Dense layer is the regular deeply connected neural network layer
        from tensorflow.keras.layers import Dense,Flatten
        #Faltten-used fot flattening the input or change the dimension
        from tensorflow.keras.layers import Conv2D,MaxPooling2D #Convolutional layer
        #MaxPooling2D-for downsampling the image
        from keras.preprocessing.image import ImageDataGenerator

        Using TensorFlow backend.

In [2]: tensorflow.__version__

Out[2]: '2.5.0'

In [3]: tensorflow.keras.__version__

Out[3]: '2.5.0'
```

Image Data Augumentation

```
In [4]: #setting parameter for Image Data agumentation to the training data
        train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
        #Image Data agumentation to the testing data
```

4.5.4 MODEL BUILDING:

Building a Model with web application named “FLASK”, model building process consist several steps like,

- Import the model building Libraries
- Initializing the model
- Adding CNN Layers
- Adding Hidden Layer
- Adding Output Layer
- Configure the Learning Process
- Training and testing the model

all the above processes are done and saved in a model.

Inserting necessary libraries

In [1]:
```python
import numpy as np#used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten
#Faltten-used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D,MaxPooling2D #Convolutional layer
#MaxPooling2D-for downsampling the image
from keras.preprocessing.image import ImageDataGenerator
```

```
Using TensorFlow backend.
```

In [2]:
```python
tensorflow.__version__
```

Out[2]: '2.5.0'

In [3]:
```python
tensorflow.keras.__version__
```

Out[3]: '2.5.0'

Image Data Augumentation

In [4]:
```python
#setting parameter for Image Data agumentation to the training data
train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

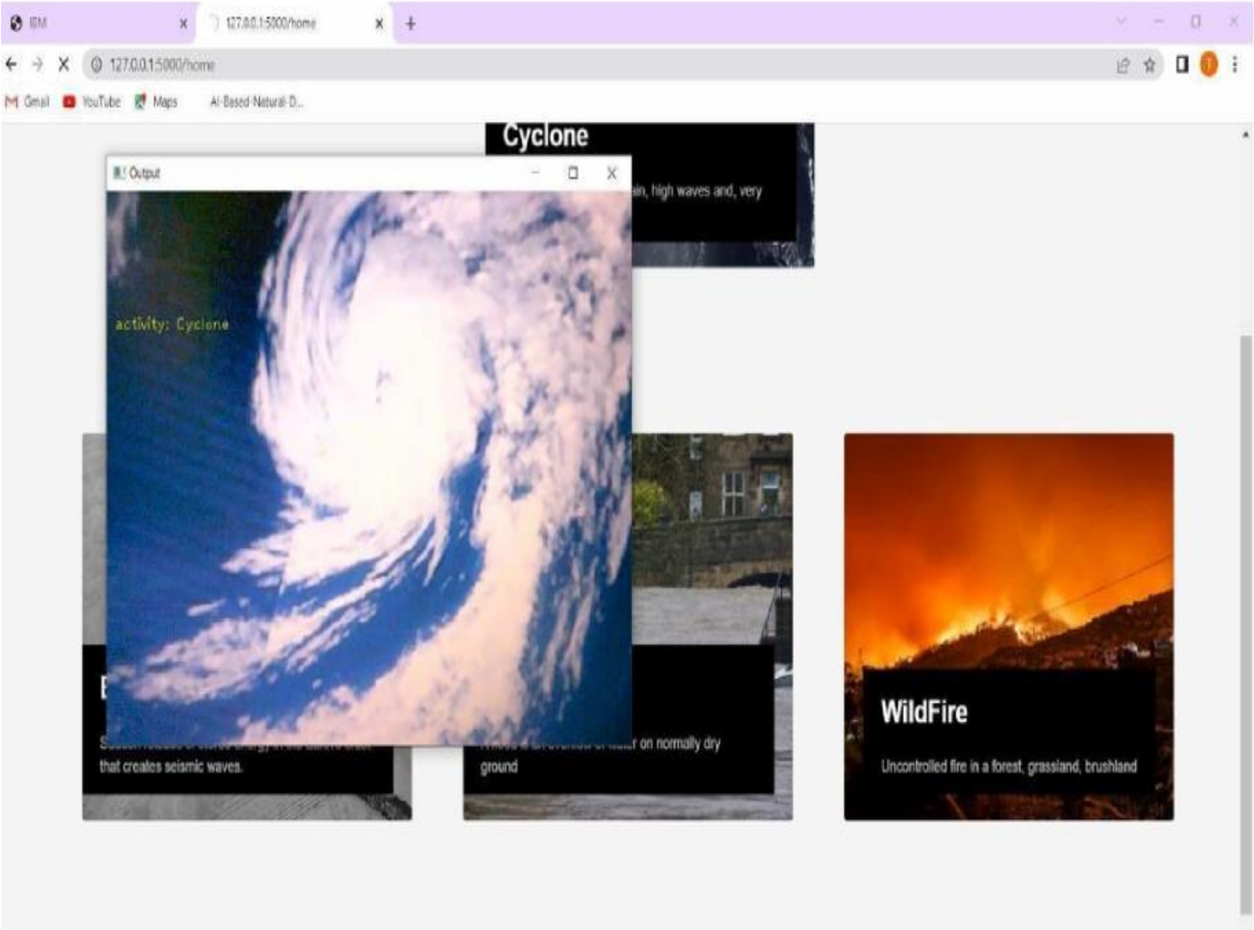
```
# import the necessary packages
from flask import Flask,render_template,request
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
#import operator
import cv2 # opencv library
from tensorflow.keras.models import load_model#to load our trained model
import numpy as np
#import os
from werkzeug.utils import secure_filename
#from playsound import playsound
#from gtts import gTTS
'''
def playaudio(text):
    speech=gTTS(text)
    print(type(speech))
    speech.save("output1.mp3")
    playsound("output1.mp3")
    return
'''
app = Flask(__name__,template_folder="templates") # initializing a flask app
# Loading the model
model=load_model(r'C:\Users\user\Desktop\IBM\Flask\templates\disaster.h5')
print("Loaded model from disk")


app=Flask(__name__,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')
@app.route('/home', methods=['GET'])
def home():
    return render_template('home.html')
@app.route('/intro', methods=['GET'])
def about():
    return render_template('intro.html')
@app.route('/upload', methods=['GET', 'POST'])
```

4.5 SPRING -4 INTEGRATE THE WEB APP WITH AI MODEL:

After creating the Model, the Model should be integrated with the web app using the Flask application. The coding part is named as app.py and it will be running in the localhost through the generated l ink. By navigating the local host the webpage will be visible

```python
            output = frame.copy()
            #print("apple")
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            frame = cv2.resize(frame, (64, 64))
            #frame = frame.astype("float32")
            x=np.expand_dims(frame, axis=0)
            result = np.argmax(model.predict(x), axis=-1)
            index=['Cyclone','Earthquake','Flood','Wildfire']
            result=str(index[result[0]])
            #print(result)
            #result=result.tolist()

            cv2.putText(output, "activity: {}".format(result), (10, 120), cv2.FONT_HERSHEY_PLAIN,
                        1, (0,255,255), 1)
            #playaudio("Emergency it is a disaster")
            cv2.imshow("Output", output)
            key = cv2.waitKey(1) & 0xFF

            # if the `q` key was pressed, break from the loop
            if key == ord("q"):
                break

        # release the file pointers
        print("[INFO] cleaning up...")
        vs.release()
        cv2.destroyAllWindows()
        return render_template("upload.html")


if __name__ == '__main__':
    app.run(debug=False,threaded=True)
```

OUTPUT



MODEL DEPLOYMENT:

The trained model which is running in the localhost without any error is deployed in the IBM Cloud for making available for the users to predict the Disaster's type and its intensity. It is integrated with the Flask application.

CONCLUSION

Many researchers have attempted to use different deep learning methods for detection of natural disasters. However, the detection of natural disasters by using deep learning techniques still faces various issues due to noise and serious class imbalance problems. To address these problems, we proposed a multi layered deep convolutional neural network for detection and intensity classification of natural disasters. The proposed method works in two blocks—one for detection of natural disaster occurrence and the second block is used to remove imbalanced class issues. The results were calculated as average statistical values: sensitivity, 97.54%; specificity, 98.22%; accuracy rate, 99.92%; precision, 97.79%; and F1-score, 97.97% for the proposed model. The proposed model achieved the highest accuracy as compared to other state-of-the-art methods due to its multi layered structure. The proposed model performs significantly better for natural disaster detection and classification, but in the future the model can be used for various natural disaster detection processes.

REFERENCES

1. Mignan A., Broccardo M. Neural network applications in earthquake prediction (1994–2019): Meta-analytic and statistical insights on their limitations. *Seism. Res. Lett.* 2020;**91**:2330–2342. doi: 10.1785/0220200021. [CrossRef] [Google Scholar]

2. Tonini M., D'Andrea M., Biondi G., Degli Esposti S., Trucchia A., Fiorucci P. A Machine Learning-Based Approach for Wildfire Susceptibility Mapping. The Case Study of the Liguria Region in Italy. *Geosciences.* 2020;**10**:105. doi: 10.3390/geosciences10030105. [CrossRef] [Google Scholar]

3. Islam A.R.M.T., Talukdar S., Mahato S., Kundu S., Eibek K.U., Pham Q.B., Kuriqi A., Linh N.T.T. Flood susceptibility modelling using advanced ensemble machine learning models. *Geosci. Front.* 2021;**12**:101075. doi: 10.1016/j.gsf.2020.09.006. [CrossRef] [Google Scholar]

4. Schlemper J., Caballero J., Hajnal V., Price A.N., Rueckert D. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. *IEEE*

Trans. Med. Imaging. 2017;**37**:491–503. doi: 10.1109/TMI.2017.2760978. [PubMed] [CrossRef] [Google Scholar]

5. Tang C., Zhu Q., Wu W., Huang W., Hong C., Niu X. PLANET: Improved convolutional neural networks with image enhancement for image classification. *Math. Probl. Eng.* 2020;**2020** doi: 10.1155/2020/1245924. [CrossRef] [Google Scholar]

6. Ashiquzzaman A., Oh S.M., Lee D., Lee J., Kim J. *Smart Trends in Computing and Communications, Proceedings of the SmartCom 2020, Paris, France, 29–31 December 2020.* Springer; Berlin/Heidelberg, Germany: 2021. Context-aware deep convolutional neural network application for fire and smoke detection in virtual environment for surveillance video analysis; pp. 459–467. [Google Scholar]

7. Li T., Zhao E., Zhang J., Hu C. Detection of Wildfire Smoke Images Based on a Densely Dilated Convolutional Network. *Electronics.* 2019;**8**:1131. doi: 10.3390/electronics8101131. [CrossRef] [Google Scholar]

8. Mangalathu S., Burton H.V. Deep learning-based classification of earthquake-impacted buildings using textual damage descriptions. *Int. J. Disaster Risk Reduct.* 2019;**36**:101111. doi: 10.1016/j.ijdrr.2019.101111. [CrossRef] [Google Scholar]

9. Hartawan D.R., Purboyo T.W., Setianingsih C. Disaster Victims Detection System Using Convolutional Neural Network (CNN) Method; Proceedings of the 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT); Bali, Indonesia. 1–3 July 2019; pp. 105–111. [Google Scholar]

10. Amit S.N.K.B., Aoki Y. Disaster detection from aerial imagery with convolutional neural network; Proceedings of the 2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC); Surabaya, Indonesia. 26–27 September 2017; pp. 239–245. [Google Scholar]

11. Yang S., Hu J., Zhang H., Liu G. Simultaneous Earthquake Detection on Multiple Stations via a Convolutional Neural Network. *Seism. Res. Lett.* 2021;**92**:246–260. doi: 10.1785/0220200137. [CrossRef] [Google Scholar]

12. Madichetty S., Sridevi M. Detecting informative tweets during disaster using deep neural networks; Proceedings of the 2019 11th International Conference on Communication Systems & Networks (COMSNETS); Bangalore, India. 7–11 January 2019; pp. 709–713. [Google Scholar]

13. Nunavath V., Goodwin M. The role of artificial intelligence in social media big data analytics for disaster management-initial results of a systematic literature review; Proceedings of the 2018 5th International Conference on Information and Communication Technologies for Disaster Management (ICT-DM); Sendai, Japan. 4– December 2018; pp. 1–4. [Google Scholar]

14. Boonsuk R., Sudprasert C., Supratid S. An Investigation on Facial Emotional Expression Recognition Based on Linear-Decision-Boundaries Classifiers Using Convolutional Neural Network for Feature Extraction; Proceedings of the 2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE); Pattaya, Thailand. 10–11 October 2019; pp. 1–5. [Google Scholar]

15. Zhou F., Huang J., Sun B., Wen G., Tian Y. Intelligent Identification Method for Natural Disasters along Transmission Lines Based on Inter-Frame Difference and Regional Convolution Neural Network; Proceedings of the 2019 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom); Xiamen, China. 16–18 December 2019; pp. 218–222. [Google Scholar]

16. Sulistijono I.A., Imansyah T., Muhajir M., Sutoyo E., Anwar M.K., Satriyanto E., Basuki A., Risnumawan A. Implementation of Victims Detection Framework on Post Disaster Scenario; Proceedings of the 2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA); Bali, Indonesia. 29–30 October 2018; pp. 253–259. [Google Scholar]

17. Padmawar P.M., Shinde A.S., Sayyed T.Z., Shinde S.K., Moholkar K. Disaster Prediction System using Convolution Neural Network; Proceedings of the 2019 International Conference on Communication and Electronics Systems (ICCES); Coimbatore, India. 17–19 July 2019; pp. 808–812. [Google Scholar]

18. Chen Y., Zhang Y., Xin J., Wang G., Mu L., Yi Y., Liu H., Liu D. UAV Image-based Forest Fire Detection Approach Using Convolutional Neural Network; Proceedings of the 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA); Xi'an, China. 18–21 June 2019; pp. 2118–2123. [Google Scholar]

19. Gonzalez A., Zuniga M.D., Nikulin C., Carvajal G., Cardenas D.G., Pedraza M.A., Fernández C., Munoz R., Castro N., Rosales B., et al. Accurate fire detection through fully convolutional network; Proceedings of the 7th Latin American Conference on Networked and Electronic Media (LACNEM 2017); Valparaiso, Chile. 6–7 November 2017. [Google Scholar]

20. Samudre P., Shende P., Jaiswal V. Optimizing Performance of Convolutional Neural Network Using Computing Technique; Proceedings of the 2019 IEEE 5th International Conference for Convergence in Technology (I2CT); Pune, India. 29–31 March 2019; pp. 1–4. [Google Scholar]

21. Lee W., Kim S., Lee Y.-T., Lee H.-W., Choi M. Deep neural networks for wild fire detection with unmanned aerial vehicle; Proceedings of the 2017 IEEE International Conference on Consumer Electronics (ICCE); Las Vegas, NV, USA. 8–11 January 2017; pp. 252–253. [Google Scholar]

22. Nguyen D.T., Ofli F., Imran M., Mitra P. Damage assessment from social media imagery data during disasters; Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining; Sydney, NSW, Australia. 31 July–3 August 2017; pp. 569–576. [Google Scholar]

23. Direkoglu C. Abnormal Crowd Behavior Detection Using Motion Information Images and Convolutional Neural Networks. *IEEE Access.* 2020;**8**:80408–80416. doi: 10.1109/ACCESS.2020.2990355. [CrossRef] [Google Scholar]

24. Yuan F., Zhang L., Xia X., Huang Q., Li X. A wave-shaped deep neural network for smoke density estimation. *IEEE Trans. Image Process.* 2019;**29**:2301–2313. doi: 10.1109/TIP.2019.2946126. [PubMed] [CrossRef] [Google Scholar]

25. Layek A.K., Poddar S., Mandal S. Detection of Flood Images Posted on Online Social Media for Disaster Response; Proceedings of the 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP); Gangtok, Sikkim, India. 28–28 February 2019; pp. 1–6. [Google Scholar]

26. Amezquita-Sanchez J., Valtierra-Rodriguez M., Adeli H. Current efforts for prediction and assessment of natural disasters: Earthquakes, tsunamis, volcanic eruptions, hurricanes, tornados, and floods. *Sci. Iran.* 2017;**24**:2645–2664. doi: 10.24200/sci.2017.4589. [CrossRef] [Google Scholar]

27. Zhang X.Y., Li X., Lin X. The data mining technology of particle swarm optimization algorithm in earthquake prediction. *Adv. Mater. Res.* 2014;**989–994**:1570–1573. doi: 10.4028/www.scientific.net/AMR.989-994.1570. [CrossRef] [Google Scholar]

28. Adeli H., Panakkat A. A probabilistic neural network for earthquake magnitude prediction. *Neural Netw.* 2009;**22**:1018–1024. doi: 10.1016/j.neunet.2009.05.003. [PubMed] [CrossRef] [Google Scholar]

29. Kradolfer U. SalanderMaps: A rapid overview about felt earthquakes through data mining of web-accesses; Proceedings of the EGU General Assembly Conference; Vienna, Austria. 7–12 April 2013; pp. EGU2013–6400. [Google Scholar]

30. Merz B., Kreibich H., Lall U. Multi-variate flood damage assessment: A tree-based data-mining approach. *Nat. Hazards Earth Syst. Sci.* 2013;**13**:53–64. doi: 10.5194/nhess-13-53-2013. [CrossRef] [Google Scholar]

31. Sahay R.R., Srivastava A. Predicting monsoon floods in rivers embedding wavelet transform, genetic algorithm and neural network. *Water Resour. Manag.* 2014;**28**:301–317. doi: 10.1007/s11269-013-0446-5. [CrossRef] [Google Scholar]

32. Venkatesan M., Thangavelu A., Prabhavathy P. An improved Bayesian classification data mining method for early warning landslide susceptibility model using GIS; Proceedings of the Seventh International Conference on Bio-

Inspired Computing: Theories and Applications (BIC-TA 2012); Springer, Gwalior, India. 12–16 December 2012; pp. 277–288. [Google Scholar]

33. Korup O., Stolle A. Landslide prediction from machine learning. *Geol. Today.* 2014;**30**:26–33. doi: 10.1111/gto.12034. [CrossRef] [Google Scholar]

34. Di Salvo R., Montalto P., Nunnari G., Neri M., Puglisi G. Multivariate time series clustering on geophysical data recorded at Mt. Etna from 1996 to 2003. *J. Volcanol. Geotherm. Res.* 2013;**251**:65–74. doi: 10.1016/j.jvolgeores.2012.02.007. [CrossRef] [Google Scholar]

35. Das H.S., Jung H. An efficient tool to assess risk of storm surges using data mining. *Coast. Hazards.* 2013:80–91. doi: 10.1061/9780784412664.008. [CrossRef] [Google Scholar]

36. Chatfield A.T., Brajawidagda U. Twitter early tsunami warning system: A case study in Indonesia's natural disaster management; Proceedings of the 2013 46th Hawaii International Conference on System Sciences; Maui, HI, USA. 7–10 January 2013; pp. 2050–2060. [Google Scholar]

37. Khalaf M., Alaskar H., Hussain A.J., Baker T., Maamar Z., Buyya R., Liatsis P., Khan W., Tawfik H., Al-Jumeily D. IoT-enabled flood severity prediction via ensemble machine learning models. *IEEE Access.* 2020;**8**:70375–70386. doi: 10.1109/ACCESS.2020.2986090. [CrossRef] [Google Scholar]

38. Zhai C., Zhang S., Cao Z., Wang X. Learning-based prediction of wildfire spread with real-time rate of spread measurement. *Combust. Flame.* 2020;**215**:333–341. doi: 10.1016/j.combustflame.2020.02.007. [CrossRef] [Google Scholar]

39. Tan J., Chen S., Wang J. Western North Pacific tropical cyclone track forecasts by a machine learning model. *Stoch. Environ. Res. Risk Assess.* 2020:1–14. doi: 10.1007/s00477-020-01930-w. [CrossRef] [Google Scholar]

40. Liu Y.Y., Li L., Liu Y.S., Chan P.W., Zhang W.H., Zhang L. Estimation of precipitation induced by tropical cyclones based on machine-learning-enhanced analogue identification of numerical prediction. *Meteorol. Appl.* 2021;**28**:e1978. doi: 10.1002/met.1978. [CrossRef] [Google Scholar]

41. Meadows M., Wilson M. A Comparison of Machine Learning Approaches to Improve Free Topography Data for Flood Modelling. *Remote Sens.* 2021;**13**:275. doi: 10.3390/rs13020275. [CrossRef] [Google Scholar]

42. Nisa A.K., Irawan M.I., Pratomo D.G. Identification of Potential Landslide Disaster in East Java Using Neural Network Model (Case Study: District of Ponogoro) *J. Phys. Conf. Ser.* 2019;**1366**:012095. doi: 10.1088/1742-6596/1366/1/012095. [CrossRef] [Google Scholar]

43. Chen X., Xu Y., Wong D.W.K., Wong T.Y., Liu J. Glaucoma detection based on deep convolutional neural network; Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); Milan, Italy. 25–29 August 2015; pp. 715–718. [Google Scholar]

44. Asaoka R., Murata H., Iwase A., Araie M. Detecting preperimetric glaucoma with standard automated perimetry using a deep learning classifier. *Ophthalmology.* 2016;**123**:1974–1980. doi: 10.1016/j.ophtha.2016.05.029. [PubMed] [CrossRef] [Google Scholar]

45. Salam A.A., Khalil T., Akram M.U., Jameel A., Basit I. Automated detection of glaucoma using structural and non structural features. *Springerplus.* 2016;**5**:1519. doi: 10.1186/s40064-016-3175-4. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

46. Claro M., Santos L., Silva W., Araújo F., Moura N., Macedo A. Automatic glaucoma detection based on optic disc segmentation and texture feature extraction. *CLEI Electron. J.* 2016;**19**:5. doi: 10.19153/cleiej.19.2.4. [CrossRef] [Google Scholar]

47. Abbas Q. Glaucoma-deep: Detection of glaucoma eye disease on retinal fundus images using deep learning. *Int. J. Adv. Comput. Sci. Appl.* 2017;**8**:41–45. doi: 10.14569/IJACSA.2017.080606. [CrossRef] [Google Scholar]

48. Li Y., Xie X., Shen L., Liu S. Reverse active learning based atrous DenseNet for pathological image classification. *BMC Bioinform.* 2019;**20**:445. doi: 10.1186/s12859-019-2979-y. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

49. Thakoor K.A., Li X., Tsamis E., Sajda P., Hood D.C. Enhancing the Accuracy of Glaucoma Detection from OCT Probability Maps using Convolutional Neural Networks; Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); Berlin, Germany. 23–27 July 2019; pp. 2036–2040. [PubMed] [Google Scholar]

50. Li L., Xu M., Wang X., Jiang L., Liu H. Attention Based Glaucoma Detection: A Large-scale Database and CNN Model; Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Long Beach, CA, USA. 16–20 June 2019; pp. 10571–10580. [Google Scholar]

51. Aamir M., Irfan M., Ali T., Ali G., Shaf A., Al-Beshri A., Alasbali T., Mahnashi M.H. An Adoptive Threshold-Based Multi-Level Deep Convolutional Neural Network for Glaucoma Eye Disease Detection and Classification. *Diagnostics.* 2020;**10**:602. doi: 10.3390/diagnostics10080602. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

52. Aamir M., Ali T., Shaf A., Irfan M., Saleem M.Q. ML-DCNNet: Multi-level Deep Convolutional Neural Network for Facial Expression Recognition and Intensity Estimation. *Arab. J. Sci. Eng.* 2020;**45**:10605–10620. doi: 10.1007/s13369-020-04811-0. [CrossRef] [Google Scholar]