# CHAPTER THREE

# Materials, Equipment and Methods

# 3.1 Introduction

The methodology approach adopted in this study for the detection and classification extreme cases of fire and flood for decision making in autonomous vehicles are is presented in this chapter. This includes the various order of techniques which begins with the identification of the variables related to extreme cases of fire and flood, along with the method of data collection applied in gathering appropriate data and attributes required for the development of the model.

The study proposed algorithms adopted in formulating the detection and classification model, coupled with procedures for the development of the model by applying collected historical datasets for training models and testing models to detect and classify the extreme cases of fire and flood. The Algorithm used for this system was the convolutional neural network (CNN). The language that was used in developing the system was Python programming language with Python-3 was used as the environment in developing the prediction models coupled with performance metrics applied for validation of the model for detection and also for classification.

# 3.2 Detection and Classification System

The classification of the system is in five major steps, which includes collection of the dataset, the preprocessing of the data, splitting the data, data augmentation and model formulationby passing it through the convolution network.

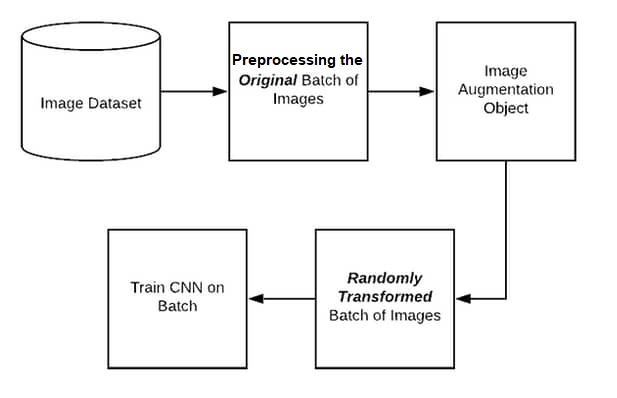


Figure 3.1: Processes in classification of the system

## 3.2.1 Dataset Collection

The collection or gathering of the necessary dataset is one of the foremost things to do in any machine or deep learning system as it is this collected data that will be processed and trained to enable the system gather as much information as it will need for effective classification and detection. Having reviewed some relevant literature in engineering as well as detection and classification models in extreme road incidents in autonomous Vehicles, the datasets that were used was collected through secondary source data repository which is from Kaggle data repository. Kaggle is an online repository that contains several datasets. Following the collection of data from the repository, the dataset encompasses images that show extreme cases of fire and extreme cases of flood. The data consist of image data that were recorded from a total of 18713 cases with 744 cases of extreme fire incidents, 441 cases of extreme flood incidents and 17, 517 normal incidents. Figure 3.2 shows sample images of the data used before preprocessing

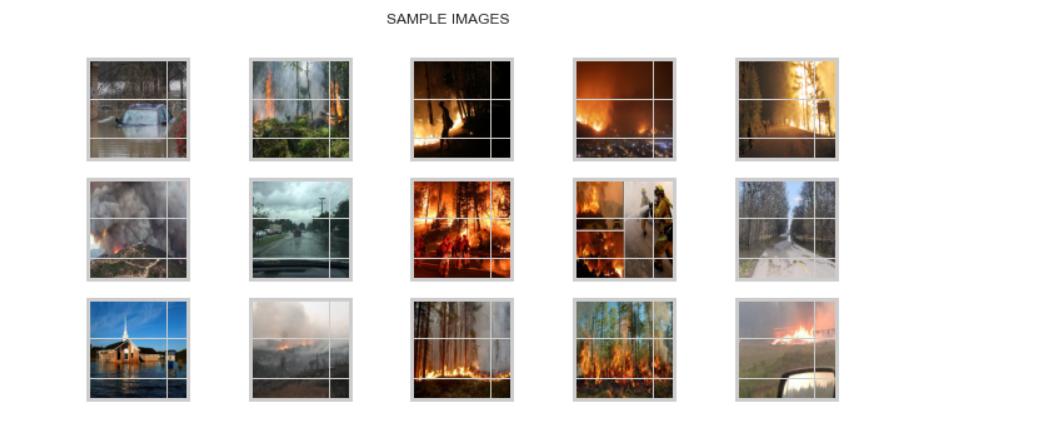


Figure 3.2: Sample Images used

## 3.2.2 Data Preprocessing

After identifying and gathering the datasets, then the next step taken by the researcher was to build an efficient and great system to preprocess the data. The data is first converted to a hierarchical data format so that it can be used for the purpose it was intended. This format ensures accurate preprocessing, augmentation and training of the model. The main aim of preprocessing the data is to remove noise and filter images from the collected data which will in turn help the organization of the images while training with the neural network [1].

Usually, when the images are collected from the databank, it may vary in sizes, color, some may not even be clear enough so it is expedient that the images are preprocessed for uniformity’s sake, it must have the same size, improved visibility, just as said earlier, noise must be removed for clarity. In preprocessing, the data must undergo few steps [2], Li, et al., [3] divided these steps into three steps which are, rescaling the images to the same, color divergence caused by different ophthalmoscopes is removed by ensuring the local average color value is set at 50% grayscale and lastly the periphery is moved by clipping 10% from the image borders; Akhila [1] divided the steps into two which is filtering and Conversion- they defined filtering as using a convolution filter to ensure the images are smoother and more reliable by removing the noise, shadows and refining the color variations while they termed conversion as the resizing of the images in the dataset to 256 by 256 pixels. Dutta, et al., [4] warned however that it is dangerous not to preprocess images as it will lead to lower accuracy and inefficient classification and detection.

For this research, the Images that were contained in the dataset were of various sizes, hence the need to resize the images in the bid to make them uniform and usable by our model. The resizing was done with a custom code. The custom code is written is such a way that it can chain together different kind of processes techniques.

## 3.2.3 Data Splitting and Label Binarization

After preprocessing the data and making sure that it is a good fit for modeling, there is the need to split that data into training and validation (Test). The data used in this research work was split into 80% training and 20% validation. Also, the labels which are categorical (i.e. are split into different categories) are then converted to numbers so it can have 3 unique values which are further split into binarized integer values that reflects the category which it is classified into.

## 3.2.4 Data Augmentation

Data augmentation helps to make the model more robust and versatile by increasing the number of the images through aforementioned processes, the model tends to learn through various sizes of the images of the dataset as against the original images captured as provided. To explain better in lay man’s language, when data is imported from the data source and has been preprocessed, the network will need to be fed as simply as possible so it doesn’t learn the wrong information.

a)b) 

Figure 3.3: Two cars; (a) represents Ford facing the left, (b) represents Chevrolet facing the right.[5]

If these images in figure 3.3 are fed into the network, the network may automatically train itself such that once it sees a car facing the left, its first intuition is that it is a ford car and when a car faces the right, it predicts it as a Chevrolet car. Augmentation is a means of avoiding positional variance just as seen in figure 3.3 where the system predicts due to the position the image is facing or the position the image takes (i.e. if the image faces left, the system calls it Ford and if the image faces right, the system calls it Chevrolet because of the varying position of the images). Variance is not just in the changing of position but also in its shape or color. Overfitting occurs because of model prediction due to the high variances in the image. Thus, what augmentation does is that, depending on the type of augmentation technique used, it tweaks the image in several other ways. Data augmentation encompasses a wide range of techniques used to generate new training samples from the original ones by applying random perturbations while making sure that the class labels are not changed. There are techniques that could flip the image to some specific degree; zoom the image, crop the image, lighten or deepen its color and so on just so the system can learn the image in different ways and not just see it in “black and white”. With the addition of the new augmented images, the dataset becomes more robust hence we have more dataset than we originally imported to train the model. These helps the system become more accurate, with data augmentation, the data is expected to be more robust as according to Hemanth, et al., [6], the more robust the data, the better the accuracy. Our dataset consisted of image data that were recorded from a total of 18713 cases with 744 cases of extreme fire incidents, 441 cases of extreme flood incidents and 17, 517 normal incidents. This meant the Positive class (extreme cases) accounted for only 0.064 percent of total cases, expressing that the data are very imbalanced.

To account for the class imbalance ratio, the negative class which was 17,517 images were augmented and reduced to 800 images hence the final data for training the models were 744 cases of extreme fire incidents, 441 cases of extreme flood incidents and 800 cases of normal road incident cases. since the data was balanced, the metrics used for measuring the performance of the model was accuracy, precision, recall and f1 score. Figure 3.4 shows a graph of the data before augmentation while figure 3.5 shows the graph of the data after augmentation

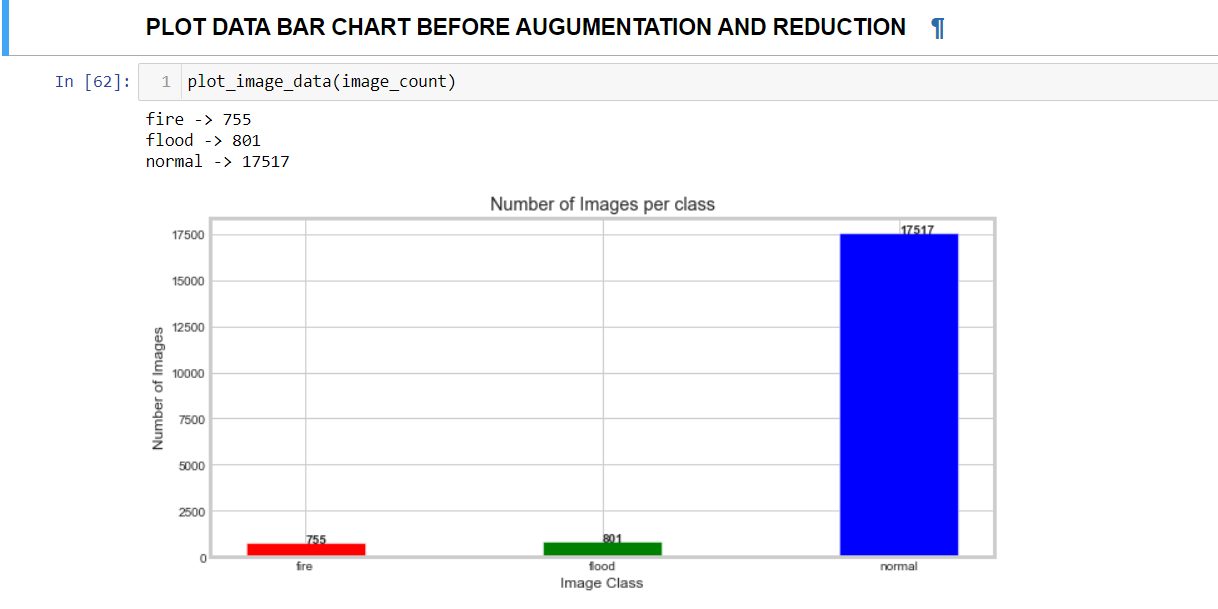


Figure 3.4: Graph of the data before augumentation

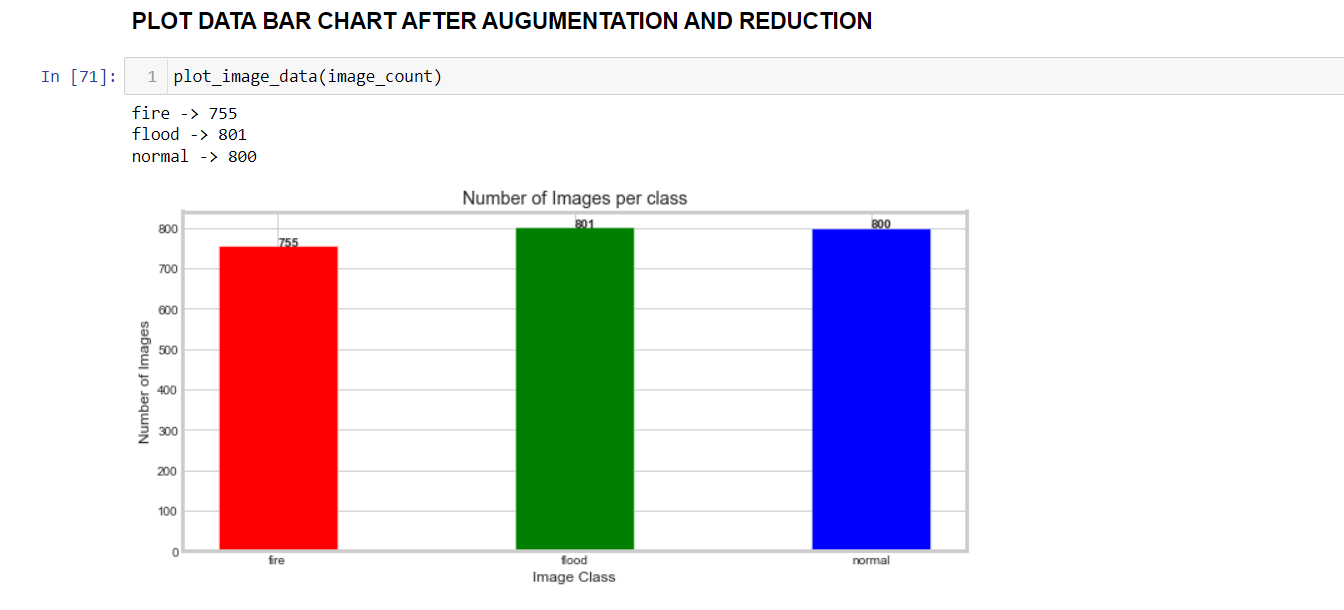


Figure 3:5 Graph of the data after augmentation

## 3.2.5 The Formulation of Model for detection and classification

After the collection of data from a secondary source and identification of non-redundant attributes as well as their validation, which was used to identify some relationship between the respective attributes contained in the dataset, a model for detection and classification was thereby formulated using Convolutional Neural Network (CNN). The CNN was used to select suitable attributes for detection and classification of extreme and normal road incidents for the autonomous vehicles. Then, attributes selected were used for the formulation of a detection and classification system. The dataset was divided into two: train and test dataset, in a proportion of 80% and 20% respectively.

### 3.2.5.1 Model Formulation and Feature Selection Using CNN

A convolutional neural network (CNN) is a machine learning technology motivated by the neuron system of humans, which permits learning from dataset representatives to explain a physical event or a decision process. Convolutional Neural Networks are capable of establishing empirical relationships among independent and dependent variables, also mining information and complex knowledge from sample datasets, which can be regarded as one of their most impressive capabilities [7]. The capacity to handle noisy data gives Convolutional Neural Networks an advantage over regression-based algorithms. A layer of input nodes and a layer of output nodes will make up of CNN, which is linked by one or more layers of hidden nodes. Suppose that we have some N×N square neuron layer which is followed by our convolutional layer. If we use an m×m filter ω, our convolutional layer output will be of size (N−m+1) × (N−m+1). In order to compute the pre-nonlinearity input to some unit in our layer, we need to sum up the contributions (weighted by the filter components) from the previous layer cells:

3.1

This is just a convolution, which we can express via:

conv2(x, w, 'valid')

Then, the convolutional layer applies its nonlinearity:

3.2

Let's assume that we have some error function, E, and we know the error values at our convolutional layer. What, then, are the error values at the layer before it, and what is the gradient for each weight in the convolutional layer? Note that the error we know and that we need to compute for the previous layer is the partial of E with respect to each neuron output. Let's first figure out what the gradient component is for each weight by applying the chain rule. Note that in the chain rule, we must sum the contributions of all expressions in which the variable occurs.

3.3

In this case, we must sum over all expressions in which occurs. (This corresponds to weight-sharing in the neural network!) Note that we know that =, just by looking at the forward propagation equations.

In order to compute the gradient, we need to know the values (which are often called "deltas"). The deltas are fairly straightforward to compute, once more using the chain rule:

3.4

As we can see, since we already know the error at the current layer , we can very easily compute the deltas at the current layer by just using the derivative of the activation function, σ′(x). Since we know the errors at the current layer, we now have everything we need to compute the gradient with respect to the weights used by this convolutional layer.

In addition to compute the weights for this convolutional layer, we need to propagate errors back to the previous layer. We can once more use the chain rule:

3.5

Looking back at the forward propagation equations, we can tell that from in equation 3.5 gives us the value for the error at the previous layer. As we can see, that looks slightly like a convolution! We have our filter ω being applied somehow to the layer; however, instead of having we have In addition, note that the expression above only makes sense for points that are at least m away from the top and left edges. In order to fix this, we must pad the top and left edges with zeros. If we do that, then this is simply a convolution using ω which has been flipped along both axes!

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data. Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. The Xception is a deep convolutional neural network architecture that involves Depthwise Separable Convolutions. This network was introduced Francois Chollet who works at Google, Inc. (Fun-Fact: He is the creator of keras). Xception is also known as “extreme” version of an Inception module. The Xception architecture used for this study is shown in figure 3.6.

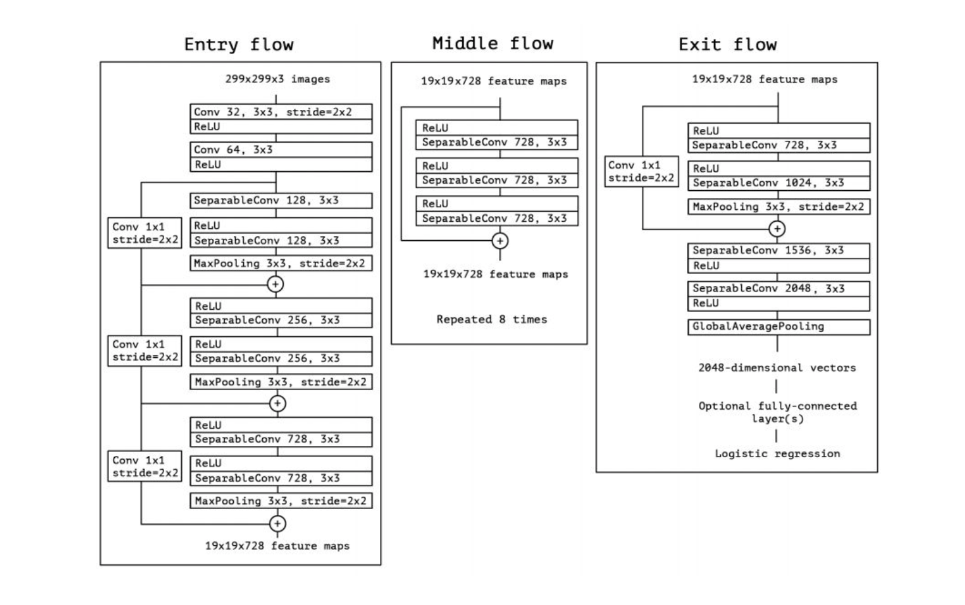


Figure 3.6. Xception architecture used for feature extraction

Likewise, A traditional neural network (ConvNet/CNN) is a deep learning algorithm that can take the image of inputs, identify the importance (weights and bias) of different aspects/objects in the image, and be able to distinguish between them and the other. The pre-processing required in a convnet is much lower than other classification algorithms. While in primitive methods the filters are manually designed, with adequate training, convection grids have the ability to learn these filters/properties. It designed for image classification.it also has an excellent capacity in sequent data analysis such as NLP.it contains two important operations, namely convolution, and pooling. the convolution operation used to extract features from images(dataset). the pooling operation used to reduce the dimensionality of features extracted from the convolution operation. maximum and average pooling are common Operations used in CNNs. it used the activation function is called relu to transfer the gradient in training using backpropagation [8].

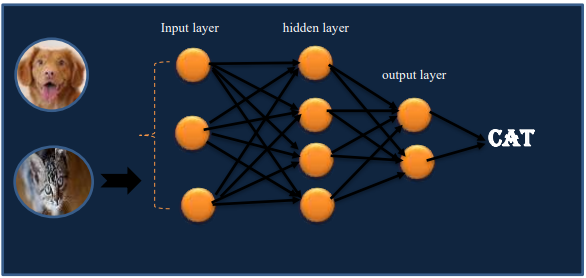


Figure 3.7 A classic CNN classifying between a dog and a cat [8]

CNNs are consisted of three types of layers. these are convolution layer, pooling layer, and fully connected layer (relu , and output).these are stacked in CNNs which shown in figure 3.8. A model for the CNN architecture used for the development of this system is also shown in figure 3.9.

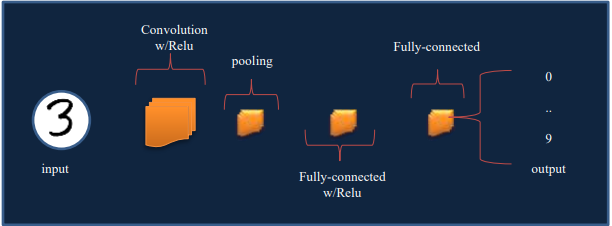


Figure 3.8 A simple generic CNN architecture [8]

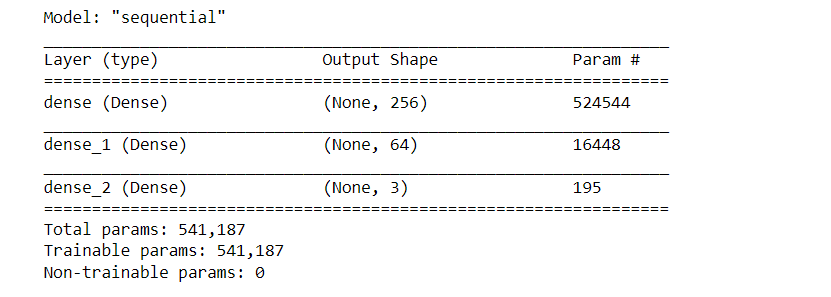


Figure 3.9: Model Architecture of the CNN model used

In general, the formula for the number of parameters in a convolutional layer performing two dimensional convolutions is (M ∗ X ∗ Y + 1) ∗ N, where M is the number of feature maps in the input (counting the redgreen-blue channels in colored images), X and Y the height and the width of the convolution (3 x 3, 5 x 5, etc.) and N the number of feature maps in the output, the one added to account for the bias. [9]

Machine learning's purpose is to develop a classifier (or a set of classifiers) that fits two basic requirements, regardless of the quality of the concrete data. For starters, for any given example, the tool should return as many of the tool's true classes as feasible; omitting even one of them would result in a false negative [10]. Similarly, the classifier should not assign a class to an example that does not belong to it because each "wrong" class would be a false positive. The advantages of using convolutional neural networks over traditional statistical methods include the following [8]

1. A convolutional neural network can identify and measure the difficulty of non-linear interactions that exist between input and output layers, also proficient in universal partially blocked patterns.
2. They have abilities to manage various kinds of data (continuous, discrete, etc.) together
3. CNNs are often used to build tedious models without explicit mathematical functions or make assumptions about the relationships between input and output parameters, allowing them to discern subtle interactions in the data [11,12].
4. Explicit Experimental designs are not always required, allowing incomplete data collected through various trial and error experimentations to be used [13].

In building the detection and classification system, the dataset was extracted and the images were scaled using grey and binarization. This process gave the model an avenue to give figures to each feature in the data. These features which had been binarized was used in training the CNN model thus helping the training the model on the images as accurately as possible.

# 3.3 Ethics and Legal Issues Relating to the Dataset

Conducting research ethically is viewed as a cornerstone of good practice, with the most common principles underpinning ethical codes of practice being “mutual respect, non-coercion and non-manipulation, and support for democratic values and institutions”. Several questions to aid researchers study ethical issues and case studies to prove their application were involved. [14]. The ethical principles examined in this report are Respect for law, Privacy, Explainabilty and Discrimination from unintended data

## 3.3.1 Respect for law and public interest

In general, research on ethics conforms to related laws in appropriate jurisdictions. Research should always be in the interest of the public. Furthermore, research should be open, clear, reproducible and peer-reviewed.

## 3.3.2 Privacy

Data may contain personally identifiableinformation which may mean it needs to be protected andprocessed in accordance with relevant Data Privacy and Data Protection rules [12]. The General Data Protection Regulation (GDPR) allows specific measures to allow processing of personal data for scientific research in the public interest, subject to appropriate safeguards such as encryption, pseudonymisation, and data minimisation. It also mentions that scientific research should increase knowledge and that personal data should not be included in publications and thus personal data was not included in the course of this research. It specifically allows the processing of data collected for other purposes for scientific or historical research. It requires that the interests of data subjects be protected and that information about the data collected, how it is being processed and safeguarded, and who is responsible be made publicly available. It encourages the use of approved codes of conduct surrounding data processing and it may be helpful for research communities to develop such codes of conduct [15]. The data to be used in this research are non-copyright images that contains images of fire, flood and normal road incidents which do not violate any privacy regulations.

## 3.3.2 Explainability

Explainability is also one of these issues. It's critical for organizations to be able to explain whatever aspects of a transaction seem to be fraudulent when identifying and demonstrating a fraudulent transaction [16]. However, if AI is built incorrectly, it can endanger explainability. This is because more powerful "deep learning models," like as neural networks, can have extremely complicated mathematical representations that make them difficult to interpret by looking at their internal representations, effectively turning them into a "black box". An explainability gap of any kind allows an AI to begin detecting false positive examples, with human operators unable to investigate why the AI did so. This could end up in causing accidents for the autonomous vehicles due to the lack of adequate explainabilty.

## 3.3.3 Discrimination from unintended data

Another issue is that if AI is not properly developed and operated, it can make predictions based on unintended biases. If left unchecked, these negative biases might lead to outright discrimination based on protected characteristics like race, sex, or gender. This is because an AI's decision-making process is ultimately determined by the data it is "trained" with; if this training data is atypical or skewed in any way, an AI will inherit those biases and make decisions based on them.

# 3.4 Justification to the Study

Machine learning which is a subset of AI methods enables a system to acquire trend or information by itself, which can be used to forecast the likelihood of events. With the advent of autonomous vehicles, the use of learning models has helped to detect irregularities in a dataset and make recommendations and prediction for the vehicles in case of a traffic based on what it learns from the dataset. Thus, AI models have impacted the safety of autonomous cars as they provide the means of decision support system. This study will be beneficial to those in the engineering sectors who are concerned with the development of safe autonomous vehicles to prevent accidents and the vehicles taking a wrong decision when faced with any obstacle; once it is incorporated into their system

# 3.5 The Processes and Environment for Modeling

With the identification of the required algorithms for the formulation of detection and classification models on autonomous vehicles, development of the detection model was carried out with image datasets, that had been acquired earlier from the Kaggle data source which consisted of fire, flood and normal images. Python 3 software package was used as the development environment for detection and classification model building.

Data collected for the sake of this study was split into two sections, where the first part is referred to as training data and the second part is referred to as testing data. The training data was used to develop the model, while the test data was used to evaluate and validate the model. From the numerous literatures studied, after the procedures for training and testing have been passed, there are few methods for evaluating and validating the performance of the model but for this study, the metrics that will be employed are accuracy, precision, f1 score and recall.

To predict a classifier's performance on new data, the prediction model's accuracy had to be evaluated on a dataset that had no bearing on the classifier's development. The test dataset, like training data, was an independent dataset that was used to represent a sample of underlying issues. It was crucial that the test dataset not be used in any way to construct the classifier since machine learning classifiers have two stages: one to show up with a basic structure for the predictive model and the second to optimize key parameters in that structure.

# 3.6 Performance Metric for Model Validation

During the evaluation phase, a variety of metrics were used to characterize the model's performance. To determine these metrics, many parameters were identified from the results of the classifier's predictions during model testing. The metric includes positive rate and negative rate as well as other parameters. Where positive represents normal incidents and negatives represent fire or flood incidents. For the sake of performance comparison, parameters used include accuracy, precision, F1 and recall.

# 4.2 SYSTEM REQUIREMENTS

The minimum hardware and software requirements to set the prediction models:

The designed system was implemented on a system with the following hardware specifications:

Personal Computer System with the following specifications:

* 1. Intel® Core™ i5 4300U CPU @1.90GHz
  2. 3GB of RAM

The following are the software tools required for the implementation of the proposed system:

* 1. Python 3.7
  2. Jupyter Notebook

All the program codes are implemented in Python programming language via the Jupyter Notebook on desktop which is a suitable work environment where python codes can be managed and implemented in .ipynb format running on Python version 3.7.

# REFERENCES

1. Akhila , T., Ambarish , A., & Unnikrishnan , K. S. (2019). Diabetic Retinopathy Detection Using Deep Neural Network. International Journal of Computer Science and Mobile Computing, 8(5), 126-131. Retrieved from www.ijcsmc.com
2. Lam, C., Yi, D., Guo, M., & Lindsay, T. (2018). Automated Detection of Diabetic Retinopathy using Deep Learning. 147-155.
3. Li, Y.-H., Yeh, N.-N., Chen , S.-J., & Chung, Y.-C. (2019). Computer-Assisted Diagnosis for Diabetic Retinopathy Based on Fundus Images Using Deep Convolutional Neural Network. doi:https://doi.org/10.1155/2019/6142839
4. Dutta, S., Manideep, B. C., Basha, S. M., Caytiles, R. D., & Iyengar, N. C. (2018). Classification of Diabetic Retinopathy Images by Using Deep Learning Models. International Journal of Grid and Distributed Computing, 11(1), 89-106. doi:http://dx.doi.org/10.14257/ijgdc.2018.11.1.09
5. Gandhi, A. (2018, July). Data Augmentation | How to use Deep Learning when you have Limited Data — Part 2. Retrieved July 9, 2020, from https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/
6. Hemanth, J., Deperlioglu, O., & Kose, U. (2019). An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network. Neural Computing and Applications.
7. Rehan, S., & Manuel, J. R. (2019). Empirical models to predict disinfection by-products in drinking water: An updated review. Encyclopedia of Environmental Health (Second Edition), 324-338
8. Sakib, Shadman & Ahmed, & Jawad, Ahmed & Kabir, Jawad & Ahmed, Hridon. (2018). An Overview of Convolutional Neural Network: Its Architecture and Applications. 10.20944/preprints201811.0546.v1.
9. Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285, 2016
10. Miroslav, K. (2017). An Introduction to machine learning (2nd Ed). Cham, Switzerland: Springer.
11. Landin, M., & Rowe, R. C. (2013). Artifi cial neural networks technology to model, understand, and optimize drug formulations. Science direct.
12. Colbourn, E. A., & Rowe, R. C. (2009). Novel approaches to neural and evolutionary computing in pharmaceutical formulation: Challenges and new possibilities. *Future Med Chem*, 713-726.
13. Colbourn, E. (2003). Neural computing: enable intelligent formulations. Pharmaceutical Technology Supplement, 16-20.
14. Erb, R. J. (1993). Introduction to back propagation Neural Network Compution. Pharm Res.
15. Daniel R. Thomas, Sergio Pastrana, Alice Hutchings, Richard Clayton, and Alastair R. Beresford (2017). Ethical issues in research using datasets of illicit origin. In *Proceedings of IMC ’39, London,* *UK, November 3–5, 4239.* DOI: 32.3367/5353587.53535
16. Barik, R. C., & Naik, B. (2015). A Novel Feature Extraction and Classification Technique for Machine Learning Using Time Series and Statistical Approach. *Computational Intelligence in Data Mining*, 217- 228.