# SYSTEM IMPLEMENTATION AND EVALUATION

# 4.1 INTRODUCTION

This chapter presents the implementation of a convolutional neural network for the detection and classification of extreme fire and flood.

# 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.

# 4.3 SYSTEM IMPLEMENTATION

The purpose of implementation phrase is to translate the software design into source code. Each component of the design is implemented as a program module. The end product of this phrase is a set of program module that will be individually tested. The steps involved involve downloading the dataset, after the dataset is downloaded and read by the system, important features to be used in the data were extracted for processing. The data was preprocessed, and fed to the CNN model. The model was used in building the detection and classification system.

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. The fire data was downloaded from (https://www.kaggle.com/datasets/phylake1337/fire-dataset), the flood was downloaded from (https://www.kaggle.com/datasets/saurabhshahane/roadway-flooding-image-dataset?select=Dataset) while the normal sets were downloaded from (https://www.kaggle.com/datasets/tallesrodrigues/autonomous-vehicle-roboflow).

As with some other machine learning algorithms, this system suffered some losses while training. The loss experienced while training this network was quite minimal and it proves to show that the classifier did a great job. The job of the loss is usually to measure how good the model performed.

Training was done over 100 epoch, loss value: 0.0923 and accuracy value: 0.9697. However, overall accuracy value is: 0.88.

Figure 4.1 shows the training loss and accuracy curve.

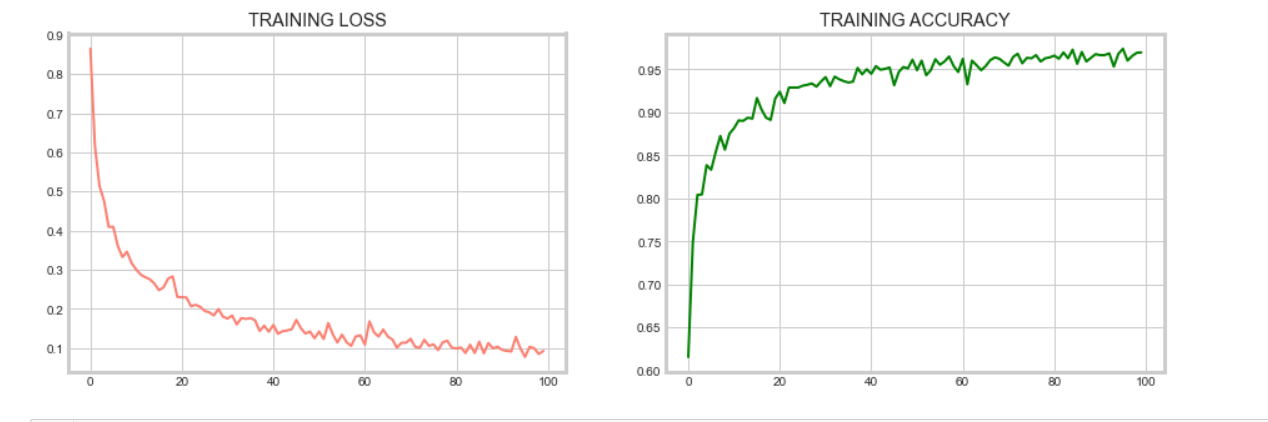


Figure 4.1: Plot of the Training loss and accuracy graph

## 4.1.1 Evaluation Metrics

The evaluation of the system was carried out using the following metrics, ranked retrieval metric i.e. Accuracy, Precision, Recall and F1-Score. The report from what the system gave in comparison to the actual answer was used to evaluate the system. These evaluation measures are defined as follows:

Where:

***A*** is true positive which is the number of DR that was correctly detected by the system

***B*** is true negative which is the number of DR that was correctly detected but not by the system

***C*** is false positive which is the number of DR that was not correctly detected by the system

***D*** is false negative which is the number of DR that was not correctly detected and not by the system

* + 1. **Evaluation of the System in terms of Accuracy, Precision, Recall and F1-Score**

From the images fed into the system and with the preconceived knowledge before feeding these images into the system (i.e., the images with each case have been known beforehand, this is what informed our opinion on how accurate it was in making predictions), these were used to measure the accuracy, precision, recall and F1-Score of the system**.** All these performance assessment criteria are derived from the confusion matrix. The confusion matrix is a square matrix table that is used to describe the performance of classification models on test dataset. Confusion matrix gives an idea of what classification models predict correctly and what they predict incorrectly. Figure 4.2 shows the performance matrix whileFigure 4.3 shows the results of the average Accuracy, Precision, Recall and F1-Score of the system with how it reacted when it was tested with images of different cases of fire, flood and normal incidents. This shows that the system performs efficiently. All these performance assessment criteria are derived from the confusion matrix. The confusion matrix is a square matrix table that is used to describe the performance of classification models on test dataset. Confusion matrix gives an idea of what classification models predict correctly and what they predict incorrectly. Figure 4.2 shows the image of the confusion matrix while figure 4.3 shows the results of the evaluation matrix of the detection system.

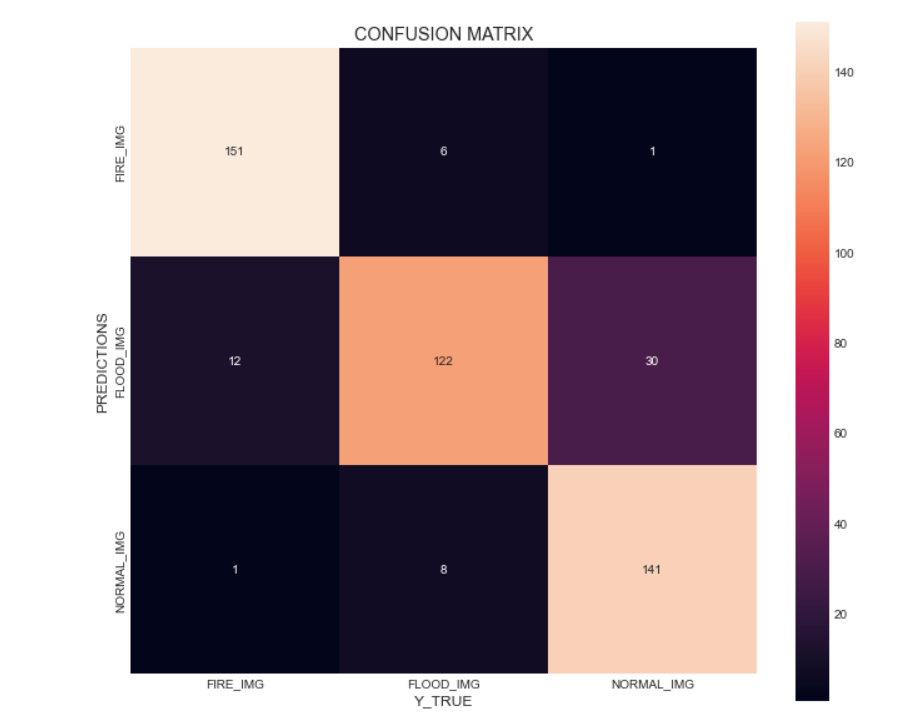


Figure 4.2:Confusion Matrix of the system

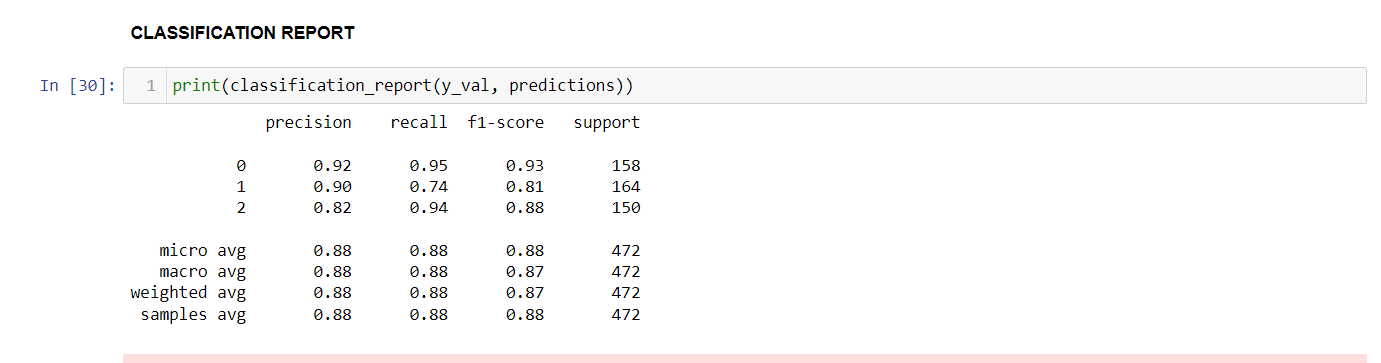


Figure 4.3: Image showing the result of the detection system with deep CNN

**DISCUSSION**

Thus the system could successfully detect fire incidents more as it has the higest precision, recall and f1 score. The detection accuracy is 88%.

To test the performance of the model on random images not part of the dataset. The results were coded from 0 to 2.

0 stands for fire incidents,

1 stands for flood, and

2 stands for normal incidents

The coding values of 0 to 2 are dependent on the probabilities of each class in the model, the class with the highest probability is the predicted class.

Also, the classification system was tested to know how much the trained model can easily classify each incidents and the results of the classification system are shown in figures 4.4 to 4.6

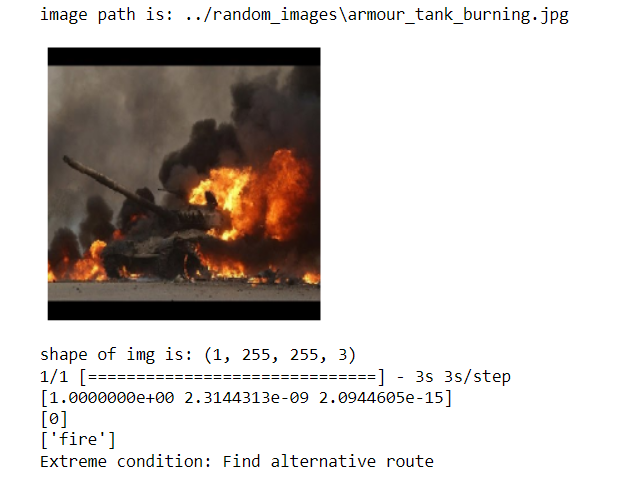


Figure 4.4: Testing the classification system with extreme fire image

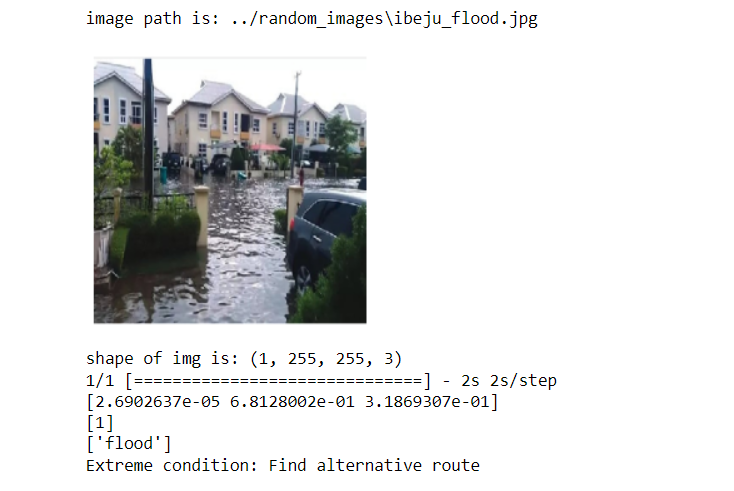


Figure 4.4: Testing the classification system with extreme flood image

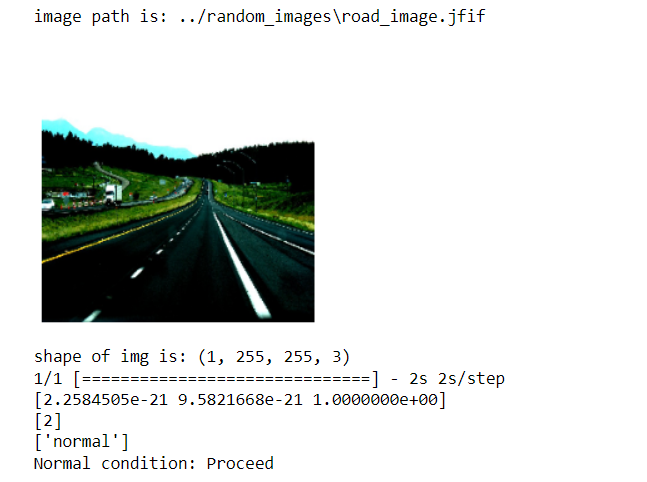


Figure 4.4: Testing the classification system with normal road images

Based on the coded output of the model ranging from 0 to 2, a decision making module was added to the CNN model and upon testing the system with different scenarios, the system was able to make recommendations for the AV either to continue on the route or find an alternative route. A case of extreme fire and flood incidents require the system to find an alternate route and normal road path kept informing the AV to proceed with its movement.