# IOT BASED INTELLEGENT FIRE ROOSTER


**By**

**Muhammad Shoaib Shabir**

**Hafiz Umer Mushtaq**

**Muhammad Mashhood Aslam**

**Nadeem Ishaq**

**Project Supervisor: Dr. Abdul Wakeel**


Submitted to
Faculty of Department of Electrical Engineering,
Military College of Signals, National University of Science and Technology, Islamabad, in partial fulfillment for the requirements of the degree of

**B.E. in Electrical (Telecommunication) Engineering**

**JULY 2020**

# Certificate

This is to certify that the Final Year Project titled

**IOT BASED INTELLEGENT FIRE ROOSTER**

submitted by

00000199905 Muhammad Shoaib Shabir

00000199909 Hafiz Umer Mushtaq

00000199908 Mashhood Aslam

00000201867 Nadeem Ishaq

has been accepted towards the requirements for the undergraduate degree in

**B.E. in Electrical (Telecommunication) Engineering**

___________________________

Dr. Abdul Wakeel

Assistant Professor

Military College of Signals (MCS),

National University of Sciences and Technology (NUST),

Islamabad, Pakistan

# ABSTRACT

Internet of things (IoT) is the network of entities that consists of electronics, programmable software, sensors, and communication facility that enables these entities to gather and transfer data. The objective of the proposed system is to alert the remote user while the fire accidents occur. This system can be installed at any remote premise which has threat of fire accidents. Using this system, detect the fire by camera. So, sensors are not required to detect fire. The Raspberry Pi controller processes the camera input and detects fire using heat signatures. By using image processing method, the report is automatically generated and sends to the person immediately after the fire is detected in any part of the frame using Wi-Fi/GSM. On detecting fire, the system will go into emergency mode. The major advantages in this method are: sending the information to the person at any time, any place and remote monitoring for immediate actions. Fire is dangerous that could bring the great loss for human life. To prevent these losses various alarm system has been developed.

*Dedicated to our beloved parents and our supervisor, who proved to be a source of motivation for us. It is their unconditional support and words of encouragement that led us to complete a work of such magnitude.*

.

# ACKNOWLEDGEMENTS

We would first like to thank Allah Almighty for bestowing upon us His countless blessings and without whom nothing is possible.

We are extremely thankful to our project supervisor Dr. Abdul Wakeel from Military College of Signals (MCS), who in addition to providing us with valuable technical help and guidance also provided us with moral support and encouragement throughout the development of the project.

Our special gratitude and acknowledgments are there for the faculty and staff of MCS who supported and guided us during our course work. Their advice, knowledge insight, training and guidance facilitated us in completing this difficult task.

We would also like to extent our gratitude to our families, friends, colleagues and a long list of well-wishers whose prayers and faith in us propelled us towards our goal.

Finally, we are indebted to Electrical (Telecomm) Department of the Military College of Signals, NUST.

# TABLE OF CONTENTS


1 INTRODUCTION .......... 1

1.1 Overview.......... 1

1.2 Motivation ..........1

1.3 Problem Statement……………………………………………..……………2

1.4 Approach.......... 2

1.5 Scope.......... 2

1.6 Aim & Objectives.......... 3

1.7 Contributions .......... 3

1.8 Organization .......... 3

2 LITERATURE REVIEW .......... 5

2.1 Introduction.......... 5

2.2 Danger of fire ..........………....5

2.3 Challenges Facing Fire Response in Pakistan..........……5

2.4 Solutions Proposed for The Challenges..........6

2.5 Fire Detection Devices…………………………………….………….....6

*2.5.1 Manually Activated Devices*……………………………….................6

*2.5.2 Modern Fire Alarms*……………………………………................... 6

2.6 IoT System..........6

2.7 Existing Systems and Designs.................................................................... 7
*2.7.1 Ionization Smoke Detectors*......................................................... 7
*2.7.2 Photoelectric*................................................................................... 7
*2.7.3 Heat*................................................................................................. 7
*2.7.4 Ionization/Photoelectric*...............................................................8
2.8 Machine Learning in Fire Alarms............................................................. 8
2.9 IoT base fire Detection................................................................................ 8
3 DESIGN AND DEVELOPMENT ............................................................ 10
3.1 Project Design ........................................................................................ 10
3.2 Computing and Processing Requirement.......................................................10
*3.2.1 IP Camera*........................................................................................11
*3.2.2 Raspberry Pi*......................................................................................11
*3.2.3 Mobile App*...................................................................................... 12
*3.2.4 SIM 909 module* ..............................................................................12
*3.2.5 Servo Motors*………………………………….……………..………..13
3.3 Techniques used for the detection of Fire..................................................... 14
3.3.1 *Convolutional Neural Network*.............................................................. 14
3.3.2 *Keras*................................................................................................ 16
3.3.3 *Deep Learning*.................................................................................... 16
3.4 Software Requirements.......................................................................17

*3.4.1 Python-IDE3*..........17

*3.4.2 Android Studio*..........18

4 PROJECT ANALYSISC AND EVALUATION ..........19

4.1 Keras Based Fire Rooster Model ..........19

*4.1.1 Fire detection with Keras and Deep Learning*..........19

*4.1.2 Using Convolutional Neural Network for Fire Detection*..........21

*4.1.3 Creation of the Training Script*..........22

*4.1.4 Training the fire detection model with Keras*..........22

4.2 Android Application.......... 24

*4.2.1 Overview*..........25

*4.2.2 Downloading the video*..........26

*4.2.3 A synch Promises*..........26

*4.2.4 Alert*..........26

*4.2.5 Buttons*..........27

5 FUTURE WORK AND CONCLUSION ..........29

5.1 Futurere Work .......... 29

5.2 Fire Extinguisher .......... 29

5.3 Gas Detectors.......... 30

5.4 Conclusion .......... 31

6 APPENDICES .......................................................................................... 32

6.1 Appendix A: *Synopsis* .................................................................... 32

6.2 Appendix C:*Android App Code*.................................................... 34

REFERENCES .......................................................................................... 38

# TABLE OF FIGURES:


Figure 3.1: Schematic Diagram of the Project .................................................................. 10

Figure 3.2: Diagram of IP Camera ………................................................................. 11

Figure 3.3: Diagram of Raspberry Pi…………………………………….................. 12

Figure 3.4: Diagramof SIM 909……….......................................................................... 13

Figure 3.5: Servo Motor…………………………..................................................... 14

Figure 3.6 A CNN Image Classification........................................................................... 15

Figure 3.7: Input image and classifying the objects based on values................................ 16

Figure 3.8: Deep Learning.................................................................................................. 17

Figure 4.1: Forest Fire......................................................................................................... 24

Figure 4.2: Fire Detection Net is based on deep learning ................................................. 25

Figure 4.3: Evaluating the optimal deep-learning-rate finder-plot.................................. 23

Figure 4.4: Accuracy / loss-curves for training................................................................. 24

Figure 4.5: React Native App Alert Message.................................................................... 26

Figure 4.6: Fire Rooster App............................................................................................. 27

# ABBREVIATIONS

| | |
|---|---|
| IoT | Internet of Things |
| M2M | Machine-to-Machine |
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| FNN | Fuzzy Neural Network |
| IDE | Integrated Development Environment) |
| SOC | System on Chip |
| IP | Internet Protocol |
| DIP | Digital Image Processing |
| App | Application |
| GPU | Graphics Processing Unit |
| (ROM) | Read-only mmemory |
| USB | Universal Serial Bus |
| HDMI | High-Definition Multimedia Interface |
| CNN | Convolutional neural network |

# Chapter: 1

# Introduction

## 1.1 Overview

IoT based Fire Rooster is a smart and intelligent device that can detect the fire and can prevent from the personal loses. It can detect between a simple fire and big fires. By this we will get rid of false alarms. This fire roster has an Android Application which will directly send a message to the owner of property and fire brigade department. The concept of Digital Image processing is a valuable asset in the detection of fire which will generate no false alarms and can detect the fireplace accurately. By the help of Android Application, it will send message to concerned department on time, give live location and hence, it will decrease the casualty rate.

## 1.2 Motivation

The world is advancing gradually so need to develop an accurate safety system which take a very little time to detect fire and then also take necessary measures to handle that condition. So, it is required to design an efficient safety system that is accurate and reduces the personal causality rate. It is a up forward method and does not require any costly and heavy material. The proposed strategy was tested for a lot of images. It gives the accuracy of 90% true detection rate.

### 1.3 Problem Statement

It is a sad reality of the present age that destructions due to fires is gradually getting scare in the world .10 million people injured and 120000 people died across the world in 2017 .

Prevention should be the first priority in reducing the unendurable injuries and deaths which are increasing day by day. Fire disasters are mostly happened due to human errors or as a result of disappointment of the framework which causes efficient harm and bad luck of human lives. To prevent from this Fire Roster is such a safety system that is accurate, decrease the personal casualty rate and take necessary safety measures in case of emergency.

### 1.4 Approach

The design of the Fire Roster as put forth by us in this project aims to differentiate between fire and non-fire. Previously sensors were used for the detection of the fire but there were certain limitations. Sensors required frequent battery charge. Sensors only detect when they are close enough to the fire. This will lead to damage the sensors. Sensors also causes false alarms as they can’t distinguish between the friendly and unfriendly fire also between the smoke of the fire of the cigarette. Here, we have proposed a computer vision-based fire detection which is using camera to get pictures of fire and then by using image processing techniques to analyse either a fire exist or not. Then we have made an Android Application which will send message about fire and live location to both owner of the place and fire department.

### 1.5 Scope

This project has a very vast scope in Industrial, commercial and household usage since primary purpose of our project is to save human lives and overcome casualties. It can be deployed to monitor a place and to detect the fire. Additionally, the system can find its scope by protecting the environment from smoke and fire. Damages of property can be decreased. The simple IP camera used for monitoring of a building can also detect about the fire and application will send message to fire department on time. This message will also contain the location the place where the fire is detected.

## 1.6 Aim & Objectives

This project aim is to integrate computer vision and electronics to develop a computerized and precise surveillance and safety system. Use new technologies for safety of human beings. To propose an automate the extinguishing process to reduce the personal causality rate. To decrease the property loses and wealth loses due to fire.

The main objectives that this project focuses to achieve are following**:**

- To develop a system which has advance technologies.
- Increasing the efficiency of fire department.
- Decrease human casualty rate due to fire.
- To reduce the property damage caused by fire.
- Providing an easily available solution to monitor a place for fire.
- To secure the sensitive zones like vital industrial zones.

## 1.7 Contributions

This project is designed and developed mainly for the prevention of fire and such would contribute the most towards social, industrial and environmental

## 1.8 Organization

The first part of the thesis is the abstract which gives a basic insight of what our project is about. It is the crux of the thesis.

The division of the chapters and the material included in them are as under:

**Chapter 1** This is the introduction section where a brief overview of the project has been given. It deals with the problem statement, approach to be followed, scope and objectives of the project.

**Chapter 2** This section provides us with the literature review regarding the traditional, IoT and machine learning Fire Rooster systems. The concept of internet of things, its enabling technologies and details regarding fire detection have also been highlighted.

**Chapter 3** The design and the development part demonstrates the flow chart exhibiting the various project stages involved in Fire Rooster. Also included is a description of the hardware and software requirements of the project.

**Chapter 4** The analysis and evaluation part gives a detail of the results obtained through the testing of various images detection.

**Chapter 5** The future work gives further improvements that can be incorporated in the existing project and the additional developments that can be made to enhance the scope of the project.

The conclusion, appendices and bibliography are given at the end of the document.

# Chapter: 2

## Literature Review

### 2.1 Introduction

A comprehensive study was done before undertaking the design and development phase of the project. This section deals with a brief review of some systems that already exists regarding fire detection and the techniques for dealing with fire. prediction of water quality.

### 2.2 Danger of The Fire

Fire is among five causes for death in the world. Fire is main cause of incidental injury too. It is toxic for environment. Fire decreases the amount of oxygen and rises the amount of lethal gases like carbon dioxide and carbon monoxide in air. Suffocation kills three more persons than burns. If we want to prevent from the fire than we must prevent it from the beginning. That's the reason identification of a fire is the most significant.

### 2.3 Challenges Facing Fire Response In Pakistan

Growing population has significant impact on availability of emergency services due lack of capacity of the services. Fire fighters are unable to get through narrow, crowded streets. Challenges are summarized as:

- Lack of skills in firefighting
- No toll-free number available
- Hurdles due to traffic jam
- Unplanned development and lack of access
- Equipment is not enough for firefighting

## 2.4 Solutions Proposed for The Challenges:

- Fire safety campaigns through media
- Availability of Toll-free number.
- Enhancement of fire department
- Proper city planning for emergency like fire outbreak

## 2.5 Fire Detection Devices

There are two basic types of fire: manually activated and automatically actuated devices

### 2.5.1 Manually Activated Devices

Manually activated devices are placed near the exit location and often look like a red button on the wall. Anyone can push the button in a risk of fire. But the problem is if button is pushed after the flame than it's too late.

### 2.5.2 Modern Fire Alarms

Modern fire alarms are used to detect fire at the earliest possible stage. They are working for 24 hours a day without any intervention.

## 2.6 IoT System

The internet of Things, or IoT, is a system of connecting computing devices, machines and persons together, that can transform data over an internet connection without any interaction between humans-to-humans and humans-to=computers. In simple words we can say that connection of electronic device with the internet. IoT do its job of connecting in an intelligent way. This will must change the lifestyle of humans along with behavior and safety. Machine-to-Machine or M2M interaction will eliminate need for human intervention and to increase safety, efficiency and productivity and advancement.

The extent to which this concept can be incorporated into our everyday lives has no boundaries. Currently it is influencing the fields of traffic management, lighting, construction, industry automation, structure health monitoring, atmosphere, NFC, proximity detection, security, water management, autonomous agriculture, emergency services, human health monitoring, vehicle management etc. and the list is growing vastly.

## 2.7 Existing Systems and Designs

A brief overview of the existing systems and designs used for water quality monitoring and the prediction algorithms implemented have been discussed below

Traditionally Smoke & Fire Monitoring Systems

### 2.7.1 Ionization Smoke Detectors: .

In Ionization smoke detectors, current is pausing between two plates. when smoke pass in the device chamber the electric current passing in device is interrupted and in result it triggers the alarms. Major drawback is it is vvery sensitive. It is not as responsive to smouldering fires. Rradioactive material is used.

### 2.7.2 Photoelectric:

**[3] Manolakos E., Logaras E., Paschos F.** Purpose a wireless sensor networks or fire detection.

In photoelectric beam of light works instead of current. Light beam is scattered, and alarm is generated. For smaller fires it is a good detector. It is most reliable. Major drawback is too Sensitive to dust and insects. It needs a regular maintenance and it is Expensive to maintain. It also more electricity.

### 2.7.3 Heat:

when temperature increase due to flames than it is suitable to use heat detectors. They often generate false alarms. False alarms are not uninterruptedly employed such as warehouses or storage facilities. Major disadvantage is that they are sensitive only to heat. So, they are not considered as life saving devices.

### 2.7.4 Ionization/Photoelectric:

A favourite of many professionals, a mixture ofionization and photoelectric smoke detector is a considered by many to be the best type of detector. It is smoke detector in one unit to ensure that any presence of smoke is detected as soon as possible. It can generate false alarms.

## *2.8 Machine Learning in Fire Alarms*

Machine learning is an Artificial Intelligence (AI) technique and its purpose is that it makes the computer learn on the basis of the past data and gets equipped to make future decisions without programming it for each and every step along the way. By this machine learning technique current image of fire or non-fire is compared with existing data base and it get result for fire or non-fire.

## 2.9 IoT base fire Detection

**Soliman H., Sudan K**, proposed a detection of fire on early stages, by connection the the sensor systems and artificial intelligence for the detection of fire in forest **[**1**].**

**In 2017, Hsu. Yu-Liang [**2_**]** established a data synthesis expertise based on many sensors with AI and joining it with smart technology . Home appliances will be controlled. Location of residence can be found. Indoor environment is suitable.

**Vikshaant et al**. **[**6,7,8 **]** designed a system which works for fire recognition in forest by merging sensor systems and compare it with fuzzy logic. Fire detection done by multisensors technology and early fire detection. Many sensors were used for information such as , humidity, light density and heat and then send it on the head of group by using event recognition devices.

**J Olivares-Mercado [**10**]** established a structure for detection of flames of fire on early stage by analysing visual smoke features such as colour, grey tenors. The system established by using videos standard with holding the smoke.

**JH Park. [**12**]** designed a system for fire detection on early bases for smart metropolises with use of AI . This AI background contains an algorithm based on fuzzy logic and algorithms based on machine learning.

# Chapter:3

## Design and Development

### 3.1 Project Design

This project is based on an ip based camera which is connected to Raspberry Pi. Raspberry Pi acts as a source of storage of the data coming from the cameras as well as do the processing on this data in real time to detect the fire and smoke from the streaming. Then this result is broadcasted through the internet over to the mobile application.

The Diagram  of the project design is depicted in the figure given below: -

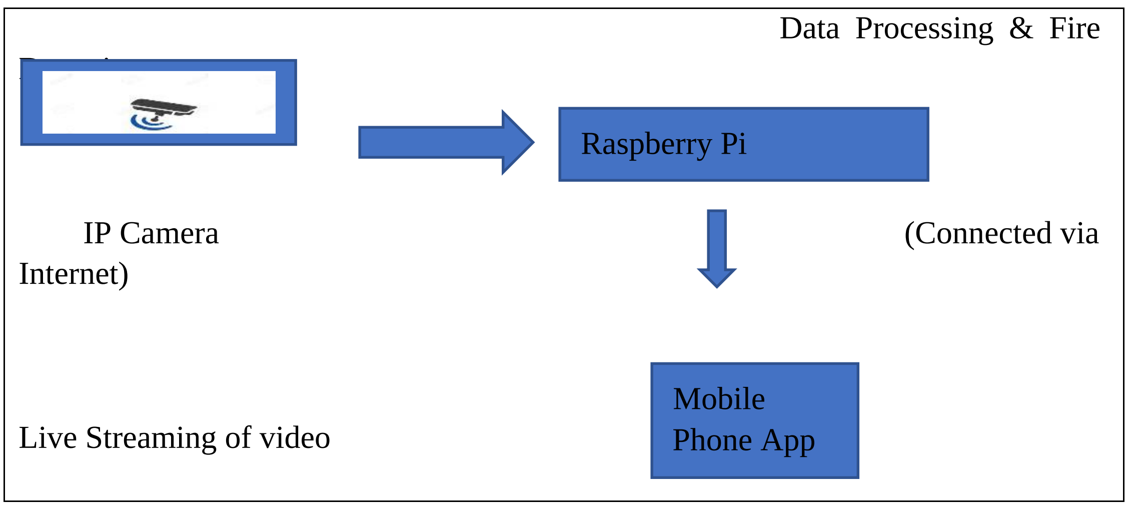


Figure 3.1: Schematic Diagram of the Project

Details and Specifications for the design has been explained as under.

### 3.2 Computing and Processing Requirement

Following components are required in this project.

#### 3.2.1 IP Camera

An Internet Protocol (IP) camera, is video camera that works digitally. It can receive a control data set and it sends data of images with the help of network. Now a days, these cameras are used commonly for purpose of surveillance and monitoring and for security purpose. But CCTV) cameras, and there is no need of resident footage instrument, only need a network.

IP Camera has been shown in the picture below.


Figure 3.2: Diagram of IP camera

### 3.2.2 Raspberry Pi

It is a computer and its size is no more than a credit card. It can be operated with the help of a keyboard and mouse just like traditional computers and provide people of all age groups to explore the world of computation. In this whole ssystem is on chip that contains a multicore processor,it has a read only memory and input ,output ports are inside it. It has a also a single board chip that contains memory, ethernet port, and micro HDMI and a host.

Raspberry Pi has been shown in the picture below.

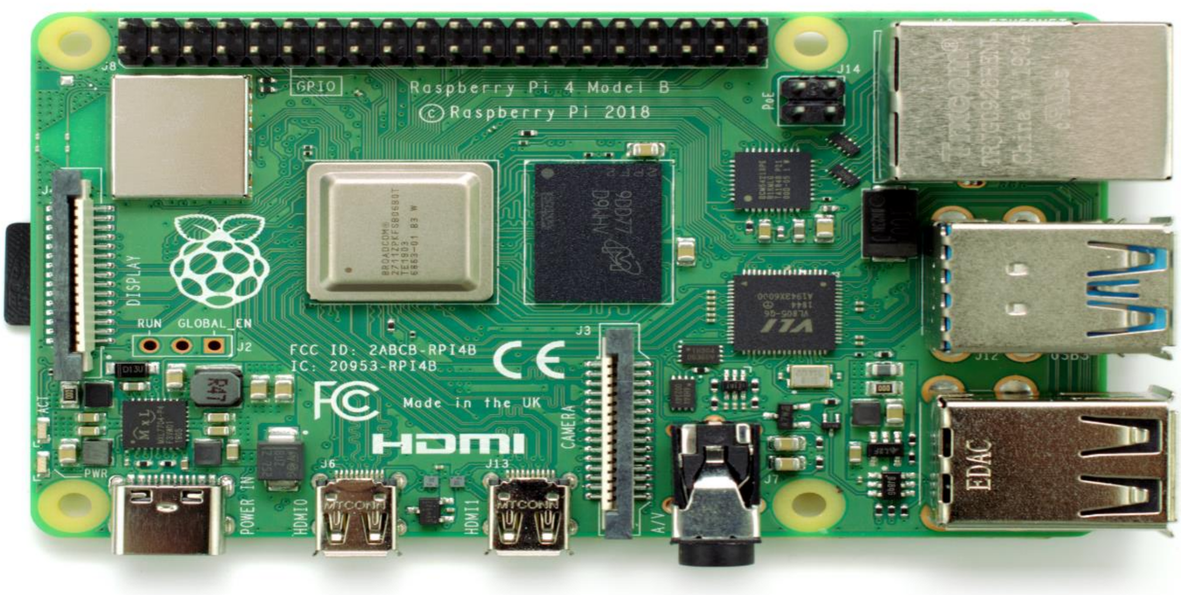


Figure 3.3: figure of Raspberry Pi

### 3.2.3 Mobile App

Smartphone app is built for the purpose of broad casting the live stream of the connected camera where the fire has started. This app is build using the React Native framework.

### 3.2.4 SIM 909 module

It is a Dual band GPRS and GPS solution in a SMT. The module that can be implanted in the application of costumers. It has a tiny arrangement, SIM909 can fit in nearly all the space necessities in user applications, especially for thin and solid demand of design. The module is accessible via AT commands. The baud rate from 9600-115200 can be find by AT command. Three wires are used in this except Power supply to crossing point with microprocessor or Host PC. The module has two set onboard power source interface for 5V and 3V power source, power used for operational on self start and control start.

Figure 3.4: Figure of SIM 909

### 3.2.5 Servo Motors

A servo machine could be a revolving actuator or a engine that allows for a exact control in terms of the exact position and controlling the spee. Essentially it has certain abilities that a routine motor does not have. In it use of a normal engine effectively and sets it with an instrument for location input.

Figure 3.5: Servo Motor

## 3.3 Techniques used for the detection of Fire

i. Convolutional Neural Network
ii. keras
iii. deep learning

### 3.3.1 CNN

CNN is convolutional neural networks and these are one of the key groups for acknowledgement of images, classifications of images. Detections of pictures, face detection etc., are nearly of the ranges where CNNs are mostly used.

First an image has come to the classifications of CNN, then a processing is done by Convolutional Neural Network on that specific image or object and organize or classify  it further down many classes like Monkey, Lion and Donkey). In Computers an array of pixels is appeared of input image, and then  it be a subject to on the  resolution of image. As main part is  the resolution of image, it will take as  d x h x w(ddimension,  height and wwidth respectively,). For example, image of 3 x 6 x 6 of RGB ,array of matrix and in this case three  refers to RGB values.

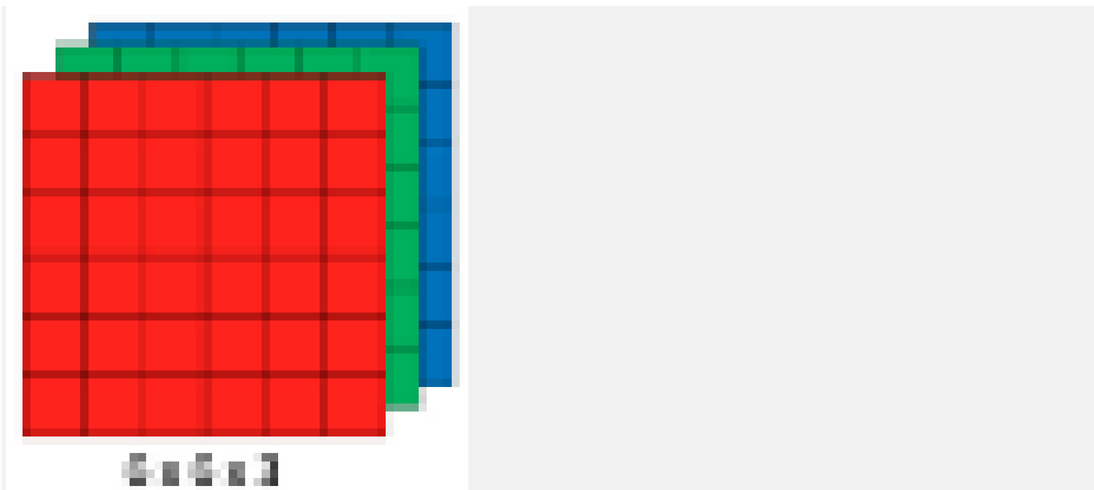

Figure 3.6: A CNN Image Classification

Theoretically, in deep learning model of Central Neural Networks for training it and test, respectively any input image will go through to convolution layers in a series manner along with filters, fully connected layers, Pooling, and spread over a function of SoftMax. This function categorizes an image to either 0 or 1. Figure shown in 3.7  is a comprehensive movement of CNN to development an image taken as input and uses these two values for differentiate.

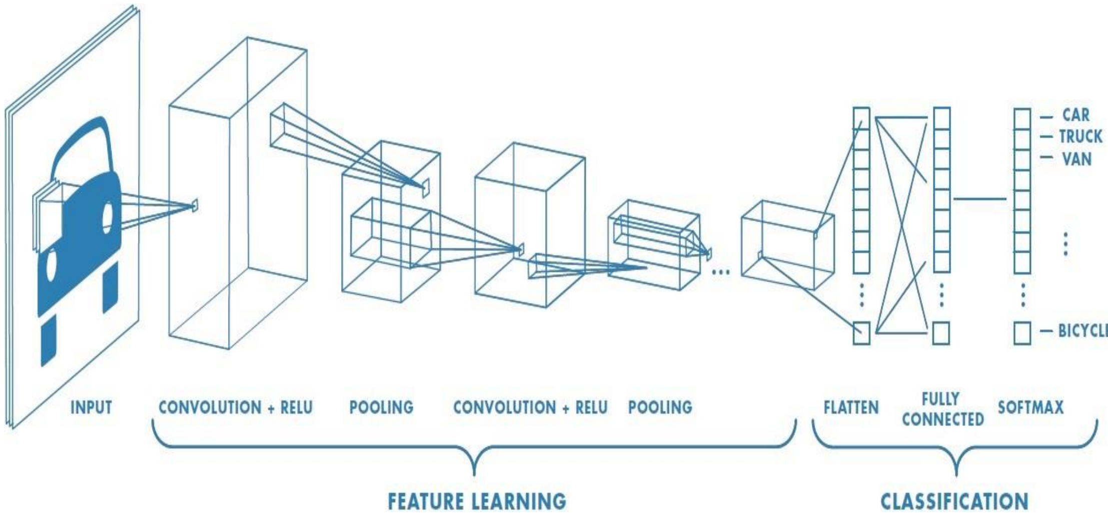


Figure 3.7: Image Taken as Inputand and Object classifying on values.

### 3.3.2 Keras

In Python, It is a library based on open-source neural-network. It works on top of Tensor Flow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Purpose for designing it to empower experimentation on fast bases with a deep neural network, it emphases to user-friendly, flexible, and extensible.

### 3.3.3 Deep Learning

In machine learning it is the part of a comprehensive family of methods by applying neural networks based on artificial intelligence with depiction of learning. Learning may be supervised, semi-supervised and unsupervised.

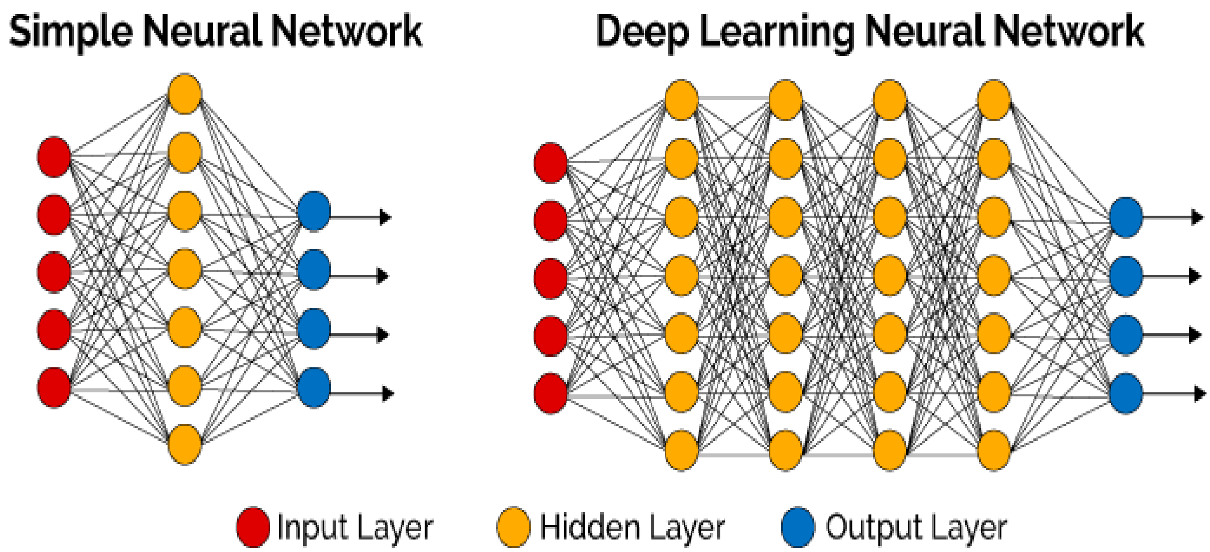


**Figure 3.8**: deep learning

### 3.4 Software Requirements

- Python-IDE
- React Native

#### 3.4.1 Python

The Raspberry Pi has been programmed with the help of Python language. It is able to do virtually anything that is possible via computers. Its an interpreted, oriented by object and a high level language of programming. The in built data structures together with the other features offered by it like the dynamic typing and binding makes it a go to programming language for Rapid Application Development. In addition to that it is extremely helpful in making a connection between the existing systems.

### 3.4.2 React Native

Using platform-native technology is the most intuitive method of building an application. Even if it is Kotlin or Java on platform of Andriod or Objective-C and Swift on iOS platform. With the presence of two prevailing operating systems i.e. Android and IOS, it is obvioustthat developers have been looking for methods for running an app with single code on both platforms without developing two different apps.
To answer this challenge, React Native framework was developed. It allows developers to work on both platforms by using JavaScript to build applications.
One app to build is allowed to developers by allowing to shear it between different platforms and it allows sharing up to 70% of the code effectively between different platforms. The separate teams and codes will interrelate in an unpredicted way but there is no risk. The whole shebang is managed in perfectly.

If a reused of the code by developers as stated above, it is also possible to bring cross-channel experience by connecting the iOS and Android app with a web portal. It has low cost development and most of all it is much more accessible. VS code is a popular software for utilizing React Native Framework and this very software was used for creating React Native app using JavaScript for Fire Rooster Project.

# Chapter: 4

## Project Analysis and Evaluation

## 4.1 Keras Based Fire Rooster Model

### 4.1.1 Fire detection with Keras and Deep Learning

**Figure 4.1** Forest Fire

In the beginning we will discuss the 2 datasets that are going to be used for detection of fire and smokee.

By completing this, there will be a reviewing directory structure and then implementation of Fire Detection Net on it through the CNN architecture that is going to use for purpose of detection of fire and smoke in images and videos.
After achieving above purposes, fire detection model is trained for fire and non fire images and analyses the precision for outcomes in this process.

After that some of the restrictions and drawbacks of the applied method will be discussed, together with how the improved needed in this and extended.

The output/directory Includes:

- Model for our serialized fire detection. Karas and deep learning will be used to train our model.
- For optimal learning rate, before training, by generateing the plot and then the process of inspection to find the Learning Rate.
- After the completion of the training process a training history plot will be generated.
- For demonstration and verification purposes the subdirectory will be populated by predict_fire.py with marked up sample images.

The pyimagesearch module contains:

- config.py: The configuration which is customizable.
- FireDetectionNet : For the detection of fire and smoke, this class is specifically design by Keras CCN
- LearningRateFinder: For finding optimal learning rate for deep learning training, this Karas class is used.

The origin of the project contains 3 scripts:

- prune.sh: It is a modest script used for removing the useless images from the given fire dataset.
- train.py: This script is used to initiate the deep learning training. The 2 modes contained in this script are (1) the training mode and (2) The Learning Rate Finder mode.
- Predict_fire.py: For the quick generation of annotated *Fire/Non-fire* images for verification, from our sample image dataset this script is used.

### 4.1.2 Using Convolutional Neural Network for Fire Detection


**Figure 4.2** Fire Detection Net is based on deep learning fire/smoke classified network made by the use of Keras learning framework.

A Fire Detection Net will be implemented, which is a type of Convolutional Neural Network used in the detection of fire and smoke in the image.

Rather than standard convolution, the network used utilizes **depthwise separable convolution,** which is:

- **Is more efficient,** as the amount of power draw and CPU use will be limited by Edge/IoT devices.
- **Needs less memory,** as RAM limited for an IoT devices
- **It iinvolves a lesser amount of computation,** as amount of CPU power available is limited.
- **In some cases the performance is better than standard convolution,** which in return produces much better fire/smoke detector.

### 4.1.3 Creation of the Training Sscript

It will be responsible for the following:

1. *Non-fire* and *fire* combined dataset loading from the disk.
2. Fire Detection Net architecture instantiation.
3. Using Learning Rate Finder class for finding optimal learning rate.
4. For the full set of epochs, training script will take the best learning rate and training network.

- matplotlib : It is used for generating and saving the plots as image file in the backend with Python.
- TensorFlow. keras : For importing stochastic gradient descent optimizer, data augmentation, and one hot label encoder, it uses TensorFlow 2.0.
- sklearn : For classification reporting and splitting of dataset, two imports are used.
- Learning Rate Finder: Prior training, for finding optimal learning rate this class will be used. While operating the script in this particular mode, two forms of plots will be generated for us that the configuration file will have inserted in (1) optimal learning rate and (2) manually inspect.

- Fire Detection Net: The CCN built in previous section for fire and smoke detection.
- config: For the training script, it is the file configuration settings (settings for prediction script are also included in it).
- paths: For listing images to the directory tree from imutils package, it contains specific functions
- argparse: It is used in command line flags parsing.
- cv2: For preprocessing and loading images OpenCV is used.

### 4.1.4 Training the fire detection model with Keras

Using Karas and Deep Learning Shell for Fire and Smoke Detection

```
$-python-train.py -lr-find-1
[ I N F O ]-loading-data......
[ I N F O ]-finding-learning-rate......
Epoch–1 / 2 0
4 7 / 4 7  [===========] - 1 0 s  2 2 1 m s / step - loss : 1 . 2 9 4 9 - accuracy : 0 . 4 9 2 3
Epoch–2 / 2 0
4 7 / 4 7  [===========] - 1 1 s  2 2 8 m s / step - loss : 1 . 3 3 1 5 - accuracy : 0 . 4 8 9 7
Epoch-3 / 2 0
4 7 / 4 7  [===========] - 1 0 s  2 1 8 m s / step - loss : 1 . 3 4 0 9 - accuracy : 0 . 4 8 6 0
Epoch–4 / 2 0
4 7 / 4 7  [===========] – 1 0 s  2 1 5 m s / step – loss : 1 . 3 9 7 3  -  accuracy : 0 . 4 7 7 0
Epoch–5 / 2 0
4 7 / 4 7  [===========] – 1 0 s   2 1 9 m s / step – loss : 1 . 3 1 7 0  - accuracy :  0 . 4 9 5 7
. . .
Epoch–1 5 / 2 0
4 7 / 4 7  [===========] – 1 0 s   2 1 6 m s / step – loss : 0 . 5 0 9 7  - accuracy : 0 . 7 7 2 8
Epoch–1 6 / 2 0
4 7 / 4 7  [===========] -1 0  s  2 1 7 m s / step – loss : 0 . 5 5 0 7  - accuracy : 0 . 7 3 4 5
Epoch-1 7 / 2 0
4 7 / 4 7  [===========] – 1 0 s  2 2 0 m s / step – loss : 0 . 7 5 5 4  - accuracy :  0 . 7 0 8 9
Epoch–1 8 / 2 0
4 7 / 4 7  [===========] – 1 0 s  2 2 0 m s / step – loss : 1 . 1 8 3 3  - accuracy : 0 . 6 6 0 6
Epoch-1 9 / 2 0
3 7 / 4 7 [==========] – E T A : 2 s  - loss : 3 .1 4 4 6 – accuracy : 0 . 6 3 3 8
[ I N F O ]-learning-rate-finder-complete
[  I N F O ]-examine-plot-and-adjust-learning-rates-before-training
```

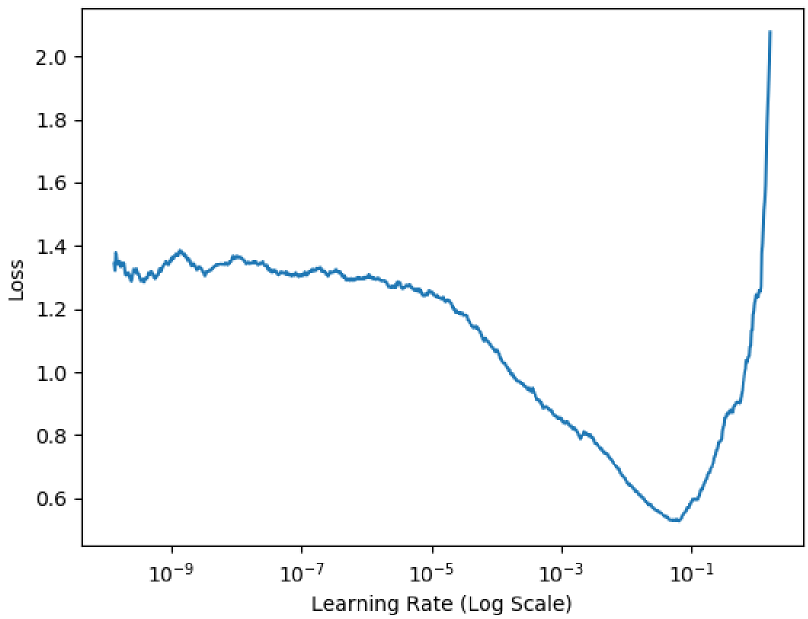


**Figure 4.3** Evaluating the optimal deep-learning-rate finder-plot. Optimal learning-rate for training a fire and smoke detector by the use of Python and Karas.

According to the **Figure 4.3** displayed above the starts to learn and gain traction at point near [1e ^ 5].

The lowest loss is present between [1e ^ 2] and [1e ^1]; but, at [1e ^ 1] a sharp increase in the loss loss is seen, this implies that the learning-rate is also large and due to which the network is beginning to be *overfitting.*

```
$-python-train.py
[ I N F O ]-loading-data…....
[ I N F O ]-compiling-model.......
[ I N F O ]-training-network.......
Epoch-1 / 5 0
4 6 / 4 6  [===============] – 1 1 s  2 3 3 m s / step – loss : 0 . 6 8 1 3  - accuracy :  0 . 6 9 7 4  -
 v a l _ loss : 0 . 6 5 8 3  -  val _ accuracy : 0 . 6 4 6 4
Epoch-2 / 5 0
4 6 / 4 6  [==============] – 1 1 s  2 3 2 m s / step – loss :  0 . 4 8 8 6  - accuracy : 0 . 76 3 1  - va
l _ loss : 0 . 7 7 7 4  -  val _ accuracy : 0 . 6 4 6 4
Epoch-3 / 5 0
4 6 / 4 6  [==============] – 1 0 s   2 2 4 m s / step – loss : 0 .4 4 1 4  - accuracy : 0 . 7 8 4 5  - va
l _ loss : 0 . 9 4 7 0  -  val _ accuracy : 0 . 6 4 6 4
Epoch-4 / 5 0
4 6 / 4 6  [==============] – 1 0 s   2 2 2 m s / step – loss :  0 .4 1 9 3  - accuracy : 0 . 7 9 1 7  - v
al _ loss : 1 . 0 7 9   - val _ accuracy : 0 . 6 4 6 4
```

```
Epoch-5 / 5 0
4 6 / 4 6  [==============] – 1 0 s   2 2 4 m s / step – loss : 0 . 4 0 1 5   - accuracy : 0 . 8 0 7 0  -
val _ loss: 1 . 2 0 3 4  - val _ accuracy :  0 . 6 4  6 4
...
Epoch-4 6 / 5 0
4 6 / 4 6  [============= ] – 1 0 s  2 2 2 m s / step – loss :  0 . 1 9 3 5  - accuracy : 0 . 9 2 7 5  - va
l _ loss : 0 . 2 9 8 5  - val _ accuracy : 0 . 8 7 8 1
Epoch-4 7 / 5 0
4 6 / 4 6 [==============] – 1 0 s   2 2 1 m s / step – loss : 0 . 1 8 1 2   - accuracy : 0 . 9 2 4 4  - v
al _ loss : 0 . 2 3 2 5 – val _ accuracy : 0. 9 0 3 1
Epoch-4 8 /5 0
4 6 / 4 6 [==============] – 1 0 s   2 26 m s / step – loss : 0 . 1 8 5 7  - accuracy :  0. 9 2 4 1 – val
 _ loss : 0 . 2 7 8 8   - val  _ accuracy : 0 . 8 9 1 1
Epoch-4 9 / 5 0
4 6 / 46   [==============] – 1 1 s   2 2 9 m s / step – loss : 0 . 2 0 6 5  - accuracy : 0 . 9 1 2 9  - v
al _ loss : 0 . 2 1 7 7  - val _ accuracy : 0 . 9 1 2 1
Epoch-5 0 / 5 0
4 6 / 4 6 [===============] – 6 3 s   1 m s / step – loss : 0 . 1 8 4 2  - accuracy : 0 . 9 3 1 6 – val _
 loss : 0 . 2 3 7 6     -  val _ accuracy :  0 . 9 1 1 1
[ I N F O ]-evaluating-network  . . . . .
            Precision-recall-f1-score-support

   N o n – F i r e     0 . 9 6     0 . 9 0      0 . 9 3      6 4 7
      F i r e          0 . 8 3     0 . 9 4      0 . 8 8      3 5 4

   a c c u r a c y             0 . 9 1     1 0 0 1
  m a c r o   a v g          0 . 9 0     0 . 9 2     0 . 9 1      1 0 0 1
w e i g h t e d   a v g      0 . 9 2     0 . 9 1     0 . 9 1      1 0 0 1

[ I N F O ]-serializing-network-to-'output/fire _ detection.model '. . . . .
```

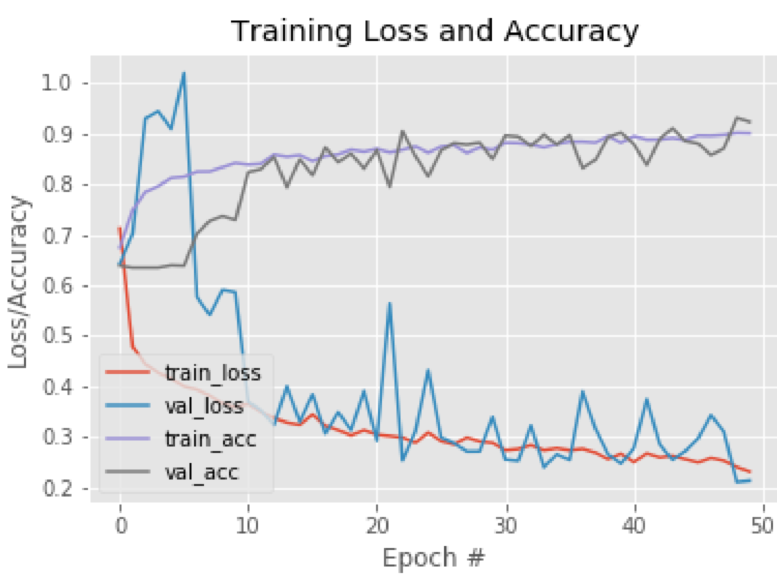


**Figure 4.4:** Accuracy / loss-curves for training

According to the graphs the results obtained were of **92% accuracy.**

## 4.2 Android Application

### 4.2.1 Overview

The whole code is made in reactnative the navigational ;ibrary used to navigate between the screens is reactNavigation. The particular version used for reactNavigation 4. The application has 3 screens. All the the 3 screens can be navigated through the tab view. 3 screens are as follows

- Home Screen
- Video feed Screen
- Data Screen

Home Screen
The Home Screen Represents the introductory data. It shows how to use the app and what features does is provide. The first screen contains all the static data stored in forms of texts and images.

Video Feed Scream
The library used for receiving the video from a particular URL on a remote server and displaying that video on the screen, expo-video.

### 4.2.2 Downloading the video

The other option for downloading the video was achieved by expo-Filesystem.
The method used for downloading the video was downloading Async. This method requires the following parameters:

- The URL to download the video from
- The local file path for storing the video

This method is a synchronous method, which implies that the screen will not further re-render until the data is downloaded.
This method gives the output in the form of a Promise.

### 4.2.3 Asynchronized Promises

The results can be achieved by using .then method and .catch method. These methods will be appended at the end of downloading Asynch method. In the case of receiving positive response, then method will be utilized and in the case of achieving the negative response .catch method will be utilized. The .then method requires a functional callback

to which it provides the data achieved. The functional callback can then utilize the data provided by .then method in any way possible.
Similarly the .catch method requires a functional call back as its parameter. This functional callback is provided with the error-data provided by .catch method.

### 4.2.4 Alert

In the body of functional callback the error data can be utilized and manipulated in any way possible. We can consol.log the data for debugging purposes or we can display the error data in the form of an alert notification to the user on the screen in real-time.

For displaying the alert data. The alert component from the react-native is used. The Alert.alert method is used for displaying the alert.

This method requires following parameters:

- The title of the alert in form of a string
- The body of alert in the form of string
- And three optional buttons

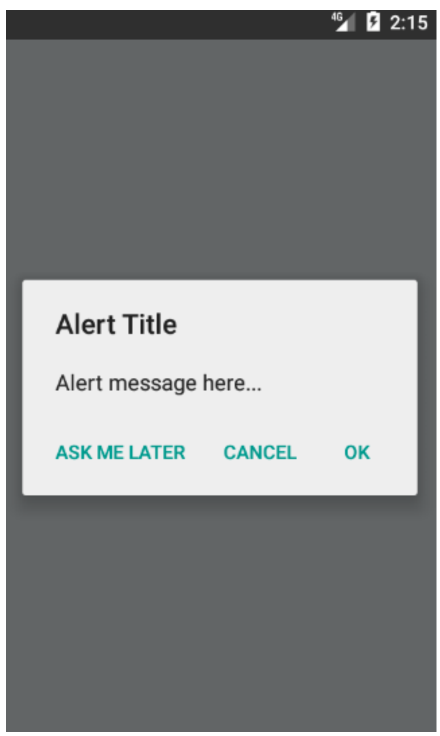


**Figure 4.5:** React Native App Alert Message

### 4.2.5 Buttons

The button component is taken from the react-native library. The button component requires the following props:

- title
- style
- onPress

There are also many other props that can be used for the buttons, but they are not mandatory.

The title prop receives a string value to display in the center of the button. The style prop requires a styling input in the form of a JavaScript object. We can change the colour, height, opacity, border radius and many other styling inputs for the button.

The onPress gets a functional call back or a pointer to a method or a function. We used the functional callback to provide the logic for the button. The onPress prop provides no parameters for the functional callback by default. In the body of the functional callback we can produce any kind of logic. We used the body of the functional callback to dismiss the alert when the alert has been shown.

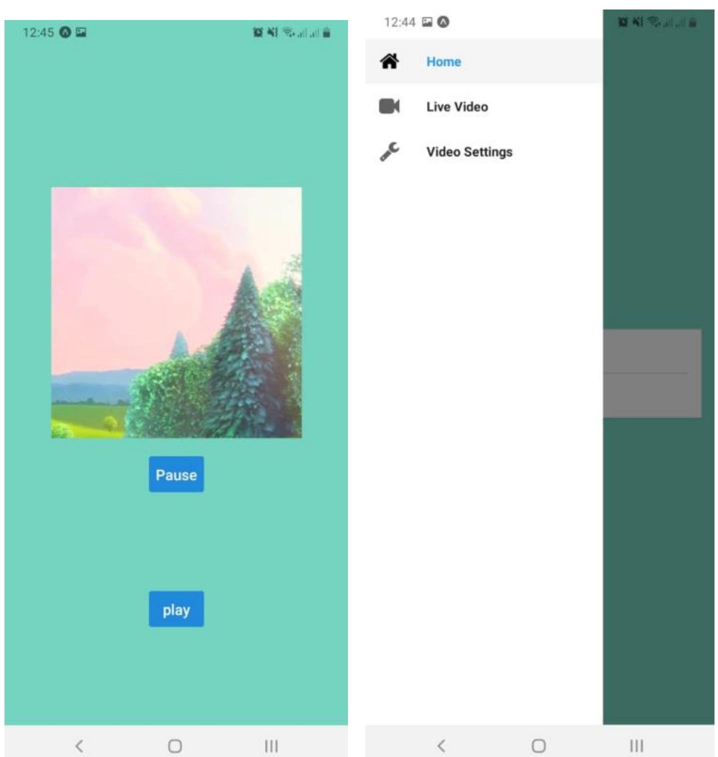


**Figure 4.6:** Fire Rooster App

# Chapter: 5

## Future Work in Project and Conclusion

### 5.1 Future Work

Detection of fire and then send a message via android application are the basic objectives of this project but we can venture deeper into the project and utilize it in achieving some additional goals and objectives as well. They have been mentioned below:

### 5.2 Fire extinguisher

Along with detection of fire, fire extinguisher system could be implemented in the future. Such a system would incorporate the concept of initial response that would facilitate the prevention of fire on early stages. This could result in to reduce the massive destruction due to fire. In result casualty rate, property loses and other loses due to fire can be reduced.

### 5.3 Gas Detectors

Gas detectors can be used for detection of gases like carbon mono oxide (CO), hydrogen Sulphide (*H2S*), and other toxic gases. These gases produce suffocation and cause for major casualty rate. These gases can be detected by using different sensors.

In industries like gas, medical, oil, nuclear, mining, food and construction, these sensors can be used combine with this project. Air quality in offices can be checked by these sensors.

## 5.4 Conclusion

In view of present condition of Fires, monitoring a place and detection of fire has become a much-needed thing. The introduction of the concept of communication possible amongst the machines i.e. Machine to Machine (M2M) communication has led them make the future decisions based upon the past given set of data. This intelligence has made it possible for the deployment of the Fire Rooster systems in a building for safety.

A fire detection and monitoring setup was established along with the android application. An Internet of Things platform was erected to connect physical parameters of the environment to the internet. The Fire Roster design as given by us has the foundation of M2M technology. A commercially deployable application was created for consumer use which provided real-time data that would not only provide video feed but would also send the message of fire and of location to both consumer and firefighting department.This constructed system aims to help in the prevention of fire and to decrease casualty rate, property loses and other issues related to firefighting department.

With further additions to our design, it can be utilized not only for the purposes of fire detection but could also find its usefulness for extinguisher system. False alarms generation by sensors can be decreased by this project. It can differentiate between smoke of fire and big fires. Another addition that can be made in the design is to accumulate the gas detectors used for detection of toxic gases to overcome the issues related to suffocation which is primary cause of death in these scenarios.

# APPENDIX

## 6.1 Appendix A: *Synopsis*

**FIRE ROOSTER**

| |
|---|
| **Extended Title: IOT Based Intelligent Fire Monitoring and Extinguishing System** |
| **Brief Description/Thesis with Salient Specifications:** Utilizing Image Processing using Open CV-Python, the entire project depends on the algorithm to recognize the fire. Initially the picture is captured by the camera which will be further processed in the OpenCV, and contend it measures the intensity of flame. Camera is covering the full room by using the servo motor which will rotate the camera to get the whole room view. When the fire is detected, using GSM module it will send an emergency message to the fire department along with the location. A specialized Android app will be designed and integrated with camera system so that an alert notification is sent to the user's phone along with a live video feed of the fire. The immediate response to the fire will be handled by firefighting department by sending them exact location of fireplace through the Android application. |
| **Scope of Work:** By the completion of this project, we want to make am efficient algorithm for fire detection by utilizing image processing techniques and implemented the project in a real-time scenario. The purpose of this project is to decrease human casualty rate and property damage caused by fire. This project contains three main portions. The first portion is to develop an algorithm for the detection of fire pixels in the image while, the second one is building an android application and integrating it with our camera system to warn the user and sending live video feed to him\her and warning the concerned authorities about the fire. And third is fire extinguishing system for the fire. |
| **Academic Objectives:**<br>• Digital Image Processing<br>• Programming Techniques (Python, JAVA)<br>• Android App Design (Android Studio)<br>• To go through the process of professional project development |
| **Application/ End Goal Objectives:**<br>• The proposed system will assist in protecting the ecological environment of water resources.<br>• The system minimizes the time and costs in detecting of fire on early stage.<br>• The system reduces the casualty rates due to fire.<br>• To overcome issues of false alarms.<br>• The proposed system is efficient and accurate<br>• The system reduces property loss due to fire in homes and industries**.** |

| |
|---|
| |
| **Previous Work Done on the Subject:** Different variants of the DIP based fire recognition systems have been designed in the past years. They Smart Apparatus for fire evacuation by S Majumder MIT, Fuzzy Logic Based Fire Monitoring by B Sarwar, IoT-based intelligent modeling of smart home environment for fire prevention and safety by A Khan etc. These are currently under research and development and some are deployed in industry. |
| **Material Resources Required:** IP Camera, GSM module, Android Mobile Application, Machine Learning, Raspberry Pi, Programming Languages i-e Python and java |
| **No. of Students Required: 0**4<br>**Group Members:**<br>• ASC Shoaib Shabir<br>• PC Mashhood Aslam<br>• ASC Nadeem Ishaq<br>• ASC Hafiz Umer Mushtaq |
| **Special Skills Required:**<br>• Programming<br>• Machine Learning<br>• Image Processing<br>• Circuit Designing |

## Approval Status:

**Supervisor Name:** Dr. Abdul Wakeel **Signature**____________________

**Assigned to:**

ASC Shoaib Shabir
PC Mashhood Aslam
ASC Hafiz Umer Mushtaq

ASC Nadeem Ishaq

**HOD Signature** ________________________

## R&D SC Record Status:

**File #** ____________________ **Coordinator Signature** __________________

## 6.2 Appendix B: CODE

### 6.2.1 FRONT-END

```
import React, { useState } from 'react';
//import Video from 'react-native-video';
import { StyleSheet, Text, View, Image, WebView } from 'react-native';
import { Button, Icon } from 'react-native-elements';
import { Video } from 'expo-av';
import { createDrawerNavigator, DrawerItems } from 'react-navigation-drawer';
import Firstscreen from './screens/Firstscreen';
import Secondscreen from './screens/Secondscreen';
import Home from './screens/Home';
import { createAppContainer } from 'react-navigation';



const Drawer = createDrawerNavigator({
  Home:{
    screen:Home,
    navigationOptions:({navigation})=>({
      title:'Home',
      drawerIcon:<Icon type='font-awesome' name='home' size={25}
onPress={()=>{navigation.toggleDrawer();}}/>,

    })
  },
  Firstscreen:{
    screen:Firstscreen,
    navigationOptions:({navigation})=>({
      title:'Live Video',
      drawerIcon:<Icon type='font-awesome' name='video-camera' size={25}
onPress={()=>{navigation.toggleDrawer();}}/>
    })
  },
```

```
  Secondscreen:{
    screen:Secondscreen,
    navigationOptions:({navigation})=>({
      title:'Video Settings',
            drawerIcon:<Icon        type='font-awesome'        name='wrench'        size={25}
onPress={()=>{navigation.toggleDrawer();}}/>
    })
  },

})

const Navigatoor = createAppContainer(Drawer);

export default function App() {

  return(
    <Navigatoor/>
  )
}



const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
    flexDirection: 'column'
  },
  backgroundVideo: {
    position: 'absolute',
    top: 0,
    left: 0,
    bottom: 0,
    right: 0,
  },
});
```

### 6.2.2 FIRST-SCREEN-CODE

```
import React, { useState } from 'react';
//import Video from 'react-native-video';
import { StyleSheet, Text, View, Image, WebView } from 'react-native';
import { Button } from 'react-native-elements';
import { Video } from 'expo-av';
import { createDrawerNavigator } from 'react-navigation-drawer';


const Firstscreen = () => {
  const [state, setstate] = useState('');
  const [videostate, setvideostate] = useState(true);

  return (
    <View style={styles.container}>
      <View style={{height:200}}></View>
      <View style={{ flex: 1, }}>
        <Video
          source={{ uri: '0' }}
          rate={1.0}
          volume={1.0}
          isMuted={false}
          resizeMode="cover"
          shouldPlay={videostate}
          isLooping
          style={{ width: 300, height: 300 }}
        />
        <Text>{state}</Text>
      </View>
      <View style={{ flex: 1, flexDirection: 'column' }}>
        <View style={{ flex: 1 }}>
          <Button
            title='Pause'
            onPress={() => { setvideostate(false) }}
          />
        </View>
        <View style={{ flex: 1 }}>
          <Button
            title='play'
            onPress={() => { setvideostate(true) }}
          />
```

```
        </View>
      </View>
    </View>

  )
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#74d4c0',
    alignItems: 'center',
    justifyContent: 'center',
    flexDirection: 'column'
  },

})

export default Firstscreen;
```

### 6.2.3 HOME-SCREEN-CODE

```
import React, { useState } from 'react';
//import Video from 'react-native-video';
import { StyleSheet, Text, View, Image, WebView, ImageBackground } from 'react-native';
import { Button, Card } from 'react-native-elements';
import { Video } from 'expo-av';
import { createDrawerNavigator } from 'react-navigation-drawer';


const Home = ()=>{
  return(
    <View style={styles.container}>
      <Card title='Title' featuredSubtitle='Sub Title'>
        <Text>The body </Text>
      </Card>
```

```
        </View>
    )
}

const styles = StyleSheet.create({
    container:{
        flex:1,
        alignContent:'center',
        justifyContent:'center',
        backgroundColor:'#74d4c0'
    }
})

export default Home;
```

### 6.2.4 SECOND-SCREEN-CODE

```
import React, { useState } from 'react';
//import Video from 'react-native-video';
import { StyleSheet, Text, View, Image, WebView } from 'react-native';
import { Button,Card } from 'react-native-elements';
import { Video } from 'expo-av';
import { createDrawerNavigator } from 'react-navigation-drawer';

 const Secondscreen = ()=>{

    return(
        <View  style={styles.constainer}>
            <Card title='Stats'>
                <Text>The body for Stats</Text>
            </Card>
        </View>
    )
}
const styles = StyleSheet.create({
    constainer:{
        flex:1,
```

```
        alignContent:'center',
        justifyContent:'center',
        backgroundColor:'#74d4c0'
    }
})

export default Secondscreen;
```

# References

**[1]** Soliman H., Sudn K., Misra A. A Smart Forest Fire Early Detection Sensory System, Another Aproach of Utilizing Wireless Sensor and Neural Networks; Proceedings of the IEEE SENSORS 2010 Conference

**[2]** SoliChou P.H., Hsu Y.L., Lee W.L., Kuo Y.C., Chang C.C., Cheng Y.S., Chang H.C., Lin S.L., Yang S.C., Lee H.H. Development of a smart home system based on multi-sensor data fusion technology; Proceedings of the international conference on applied system innovation (ICASI); Sapporo, Japan.

**[3]** Manolakos E., Logaras E., Paschos F. Wireless Sensor Network Application or Fire Hazard Detection and Monitoring. Lecture Notesof the Institute for Computer Sciences. Soc. Inform. Telecommun.

**[4]** Muralidharan A., Joseph F. Fire Detection System using Fuzzy logic. Int. J. Eng. Sci. Res. Technol.

**[5]** Maksimovic M., Vujovic V., Perišic B., Miloševic V. Developing a fuzzy logic based system for monitoring and early detection of residential fire based on thermistor sensors. Comput. Sci. Inf. Syst

**[6]** Tan W., Wang Q., Huang H., Guo Y., Zhan G. Mine Fire Detection System Based on Wireless Sensor Network; Procedings of the Conference on Information Acquisition (ICIA'07); Seogwipo-si, Korea.

**[7]** Aslan Y.E., Korpeoglu I., Ulusoy Ö. A framework for use of wireless sensor networks in forest fire detection and monitoring. Comput. Environ. Urban Syst.

**[8]** Son B., Her Y.S., Kim J.G. A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea Mountains. Int. J. Comput. Sci. Netw. Secur.

**[9]** Sowah R., Ampadu K.O., Ofoli A., Koumadi K., Mills G.A., Nortey J. Design and Implementation of a Fire Detection and Control System for Automobiles using Fuzzy logic; Proceedings of the IEEE Industry Applications Society Annual Meeting; Portland, OR, USA

**[10]** Olivares-Mercado J., Toscano-Medina K., Sánchez-Perez G., Hernandez-Suarez A., Perez-Meana H., Sandoval Orozco A.L., García Villalba L.J. Early Fire Detection on Video Using LBP and Spread Ascending of Smoke.

**[11]** P. D. A. Kirti Waykole, "Fire Detection in Video," *Advanced Research in Electronics and Communication Engineering*, vol. 3, no. 3, p. 5, 2014.

**[12]** Park J.H., Lee S., Yun S., Kim H., Kim W.T. Dependable fire detection system with multifunctional artificial intelligence framework. Sensors.

**[13]** Chiang S.Y., Kan Y.C., Chen Y.S., Tu Y.C., Lin H.C. Fuzzy computing model of activity recognition on WSN movement data for ubiquitous healthcare measurement. Sensors

**[14]** Sarwar B., Bajwa I., Ramzan S., Ramzan B., Kausar M. Design and Application of Fuzzy logic Based Fire Monitoring and Warning Systems for Smart Buildings. Symmetry