**IAES International Journal of Robotics and Automation (IJRA)**
Vol. 99, No. 1, May 2022, pp. 1~1x
ISSN: 2722-2586, DOI: 10.11591/ijra.v99i1.pp1-1x ❒ 

# FIRE DETECTION BY DEEP LEARNING

**Venkatarao Rebba[1], Manohar Akula[2], Kesava Prakash Thatavarthi[3]**
[1]School of Computing and Augmented Intelligence, Arizona State University, Tempe, United States of America
[2]School of Computing and Augmented Intelligence, Arizona State University, Tempe, United States of America
[3]School of Computing and Augmented Intelligence, Arizona State University, Tempe, United States of America

**Article Info**

***Article history:***

March 16, 2022
April 29 , 2022

***Keywords:***

Deep Learning
CNN classifier
Raspberry Pi 3
Localization
Fire Detection

**ABSTRACT**

As a result of recent advancements in vision-based systems, we can now create intelligent fire detection systems that are critical for enhancing both the safety and efficacy of overall fire detection systems. The goal of putting this work together is for it to be able to generate real-time information on the fire inside the Laser cutting machine. The goal of this endeavor is to address the shortcomings of traditional firefighting methods. To achieve the stated goal, the authors employed cutting-edge technologies such as deep learning. The fire detection system used Deep Learning to categorize items of interest from frames in real time. For detecting fire in a particular region, the suggested approach has a 96 percent accuracy.By extracting, processing, and analyzing key information from the provided frame, the supplied system is able to properly update operators with up-to-date scene information.



***Corresponding Author:***

Venkatarao Rebba
School of Computing and Augmented Intelligence, Arizona State University
Tempe, Arizona, USA
Email: vrebba@asu.edu

## 1. INTRODUCTION

The authors of this study have developed a Deep learning-based autonomous system capable of providing information that can increase operators' situational awareness of the area in which they are deployed. The Deep learning-based system can identify and categorize fire regions, as well as relay warning information to the operator, giving critical data for guiding essential planning choices. Traditional firefighting systems rely on temperature, UV, and fire sensors to identify the presence of fire. Smoke, on the other hand, is considerably less efficient at detecting fire since it only covers a small region or cannot view the entire area of interest.

Visual-based systems have two advantages: they can scan a greater area and give complete coverage, and visual cameras are very inexpensive. However, the same firefighting tactics continue to be deployed in structures, offices, and retail malls. However, using this detector, the reaction time increases as the space is expanded. Aside from that, there are image processing-based firefighting systems that depend on color, motion, and texture elements of the collected image for fire detection. In image processing-based approaches, the employment of histogram-based thresholding and optical flow-based motion vector is critical.

Fire is an abnormal event that can quickly cause property damage and significant injury. According to the National Fire Protection Association (NAPA), the United States fire department responded to an estimated 1,319,500 fires during 2017, which resulted in 3,400 civilian fire fatalities, 14,670 civilian fire injuries, and an estimated $23 billion in direct property loss. In order to prevent such disasters, fire detection without a false alarm during an early stage is crucial. Accordingly, various automatic fire detection technologies are being introduced, and are widely implemented in real life. In general, two broad categories of technologies can be seen: traditional fire alarm and fire detection using computer vision. Traditional fire alarm technology is based on heat or smoke sensors that require proximity for activation. These sensors need human involvement to confirm the situation in case of alarm. To overcome these limitations, research has been done on computer vision-based methods combined with various types of supplementary sensors. This approach gives larger surveillance coverage and offers an advantage of less human intervention with a faster response, and can be confirmed without requiring a visit to the location, and provides detailed information such as the fire location, size, and degree. Despite these advantages, some issues remain concerning the complexity of the system, and false detection for diverse reasons. Therefore, researchers have invested significant effort to address such issues in terms of computer vision technology. Early research on computer vision-based detection was focused on the color of a fire within the framework of a rule-based system that is often sensitive to environmental conditions. So, further studies improved the supplementary features to the color of a fire, including area, surface, boundary, and motion of the suspected region, with other types of decision-making algorithms, such as Bayes classifier etc.

In this paper, we propose a deep-learning-based fire detection model, which detects the origin of fire and also tracks its intensity using a bounding box. The proposed method, which uses a Convolutional Neural Network (CNN) is then deployed on a Raspberry-Pi 3 which is connected to a Raspberry Pi camera that is used for the surveillance of the trimming process that takes place at the Aerospec facility.

## 2. METHODOLOGY

The concept of an artificial neural network (ANN) is a network of neurons in the human brain. The human brain conveys information to other neurons through neurons, and each neuron has many linked neurons. The ANN algorithm was developed as a result of this to perform simple pattern recognition. Each neuron retains information, and it is predicted that as neurons are linked, they will learn from one another until the output network is reached. The input layer, hidden layer, and output layer make up Artificial Neural Networks. Before accessing the output layer, neurons must pass through the name synapse/activation to determine the neuron's capabilities

A. Artificial Neural Networks Layer (ANN)

A mathematical model function for Artificial Neural Networks is defined as:

$$f : X \rightarrow Y \quad (1)$$

- The input layer is made up of neurons that receive data from variable X as input. If the network does not include hidden layers, all neurons in this layer can be linked to neurons in hidden levels or straight to the output layer.
- The hidden layer is made up of neurons that receive data from the input layer; the output layer is made up of neurons that receive data from hidden layers or directly from input layers, and whose output values indicate the X to Y computation results.

B. Backpropagation

A criterion used to lower the degree of error/error in a neural network is the backpropagation approach. After conducting a Feedforward Pass, which starts the process of constructing a neural network with three layers, namely the input layer, hidden layer, and output layer, this backpropagation or Backward Pass method is used. The weight and bias values are then determined at random, commonly using intervals of 0 to 1. Initialize the weight and bias functions to identify the input value with the minimum error level, then compute feedforward until the final value is found, then backward pass/backpropagation until the value matching is found. The process of backpropagation may be seen in

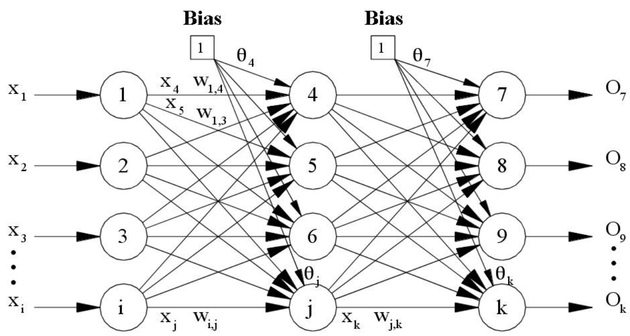


Fig.1 Examples of neural Networks Using Backpropagation

C. Fire Segmentation:

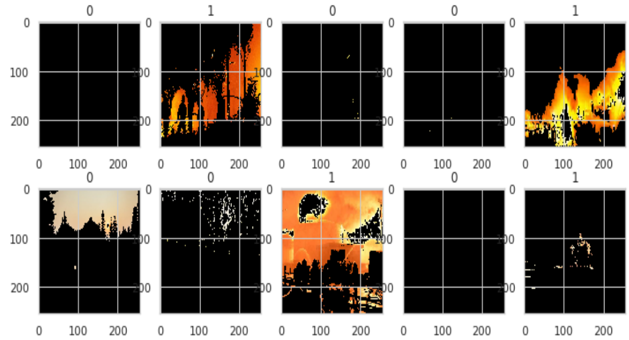

Fig.2 Fire Segmentation

During the segmentation process, the image is divided into different regions based on the characteristics of pixels to identify the fire boundaries and to locate its position. This helps in simplifying an image and helps in using it more efficiently. Figure 2 shows the output of Fire segmentation. It is clear that the pixels which did not contain the fire pixels were labeled as 0 and the one's with the fire segmented pixels are labeled as 1. This indicates the presence of fire in the given frame.

D. Convolution Neural Networks (CNN)

The main mechanism that underpins CNN is the convolutional layer. The Convolutional layer is made up of neurons arranged in a filter shape with length and height (pixels). The goal of image data convolution is to extract features from the input image.Convolution will result in linear modifications of input data based on spatial information.

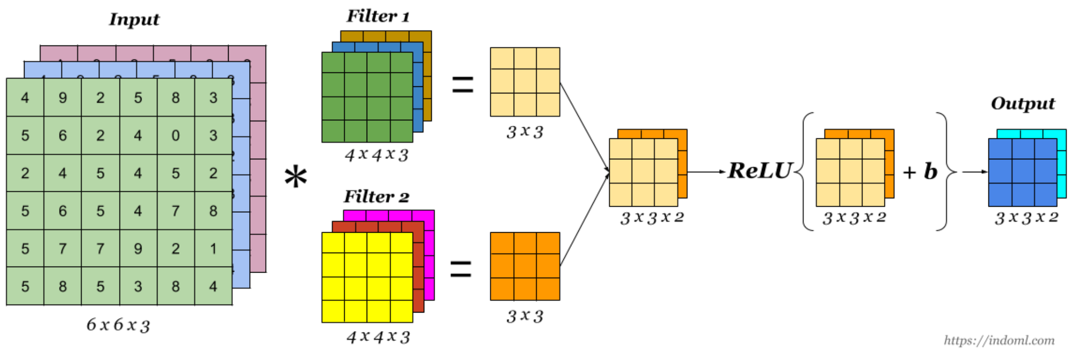


Fig.3 Convolution Layers

1. Pooling Layer

   The goal of this pooling layer is to divide the convolution layer's output into multiple tiny grids and then assemble the reduced picture matrix using the maximum value from each grid. The pooling layer's purpose is to lower the number of parameters and the convolution layer's complexity. The most generally used approach in the pooling layer is max-pooling, which takes the highest important value from the convolution layer.

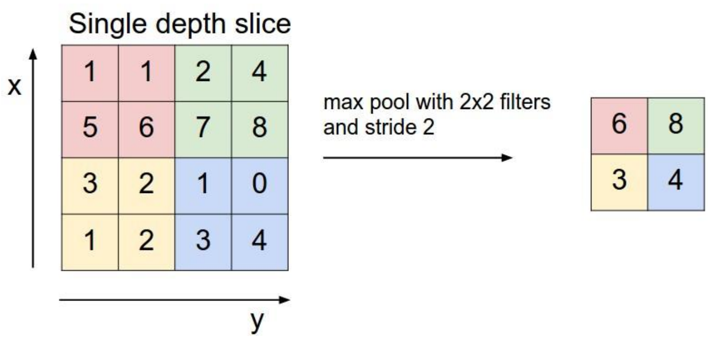


Fig.4 Pooling Layer

2. Fully connected layer
   The Fully Connected Layer, the last layer of CNN, may incorporate all of the individual features recognized in the input data by the preceding layer. The Fully-Connected layer can record the outcomes of the previous levels as vectors.

## 3. SYSTEM DESCRIPTION

Our approach involved several steps that are to be performed during the different stages of this work. Each and every step is discussed below in-detail:

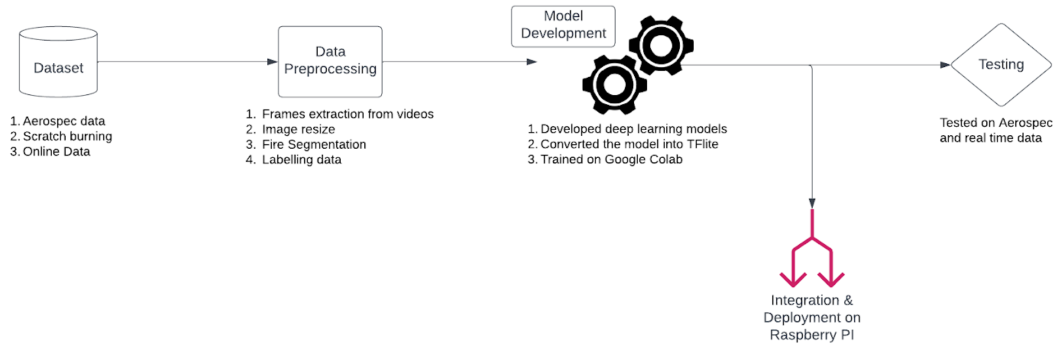


Fig. 5 Overview of the multiple stages involved in our approach

1. Dataset Creation: For our approach, in order to get accurate results, we needed a dataset that has the critical information of the environment and also the data of such incidents. We created a custom dataset that contains >20,000 images obtained from the videos provided by Aerospec, data created by us while testing the model and also included the online data.

2. Data preprocessing and Labeling the data: We extracted many frames from the dataset videos by running the python scripts. The frames obtained are then augmented for multiple scenarios to handle complexities. From these frames, we then categorized the data into 1) Fire and 2) Non- Fire scenarios. All the images that had fire in it were labeled as Fire and the frames that didn't have any anomaly were given the Non-Fire label. The labeling here is to distinguish between what is and is not fire. As shown in Fig.4, we currently use conditions 1 and 0 for fire and not fire. The label will display when the forecast is made.

   The following is the procedure for labeling a picture. The file name for each fire image will be 'fire-n,' and the non-fire image will be 'not fire-n,' followed by the def labeling code. We must label this data since it will allow the computer to recognize the information included in each image, which is required for machine learning.

3. Pre - Processing Data
   The dataset we utilized in Figure 4 is a fire image. Fire and Not Fire were the two labels created. We obtained fire photographs by shooting pictures with a webcam camera, which yielded a total of 20000 images, including non-fire shots. The shooting procedure is done from several angles, with around 3 degrees of angle in every photograph. Preprocessing, such as resizing and turning photos into arrays, is part of data collecting. The picture in the dataset is resized from 1920 x 1080 pixels to 320 x 240 pixels, and then it is turned to an array.

4. Splitting the Data:
   Split data is the process of separating data from a dataset, which is then separated into training and test data. The data we use to forecast our system is called test data. A system uses training data to train computers to distinguish fire and non-fire objects based on the criteria supplied. We split the test data by 20% and the training data by 80% of the entire dataset at this point.

5. Training the Data

   Sklearn's MLPClassifier library was utilized to assist with the training procedure in this work. The dataset is then required; in the dataset partitioned for training purposes, the machine requires learning since MLP, a supervised learning technique, is utilized in this manner. Because the training data used is not excessive, the training procedure takes only seconds.

6. Prediction:

   The forecast this time is based on label outputs 1 and 0. Prediction is achieved by creating a folder containing photographs downloaded from the internet, local and, Aerospec. Resulting to receive 98.75 percent accurate results.

7. Deployment on Raspberry Pi using TFlite:

   Our detection model was too large in size and this couldn't be integrated in the Raspberry Pi given its limited memory space. We converted our model into a TFlite model which allowed us to reduce the size and it also offered faster inference. During conversion we optimized the model size and decreased the latency to avoid loss in accuracy. This TFlite model was executed on the Raspberry Pi using the TensorFlow Lite interpreter.

## 4. RESULTS AND DISCUSSION

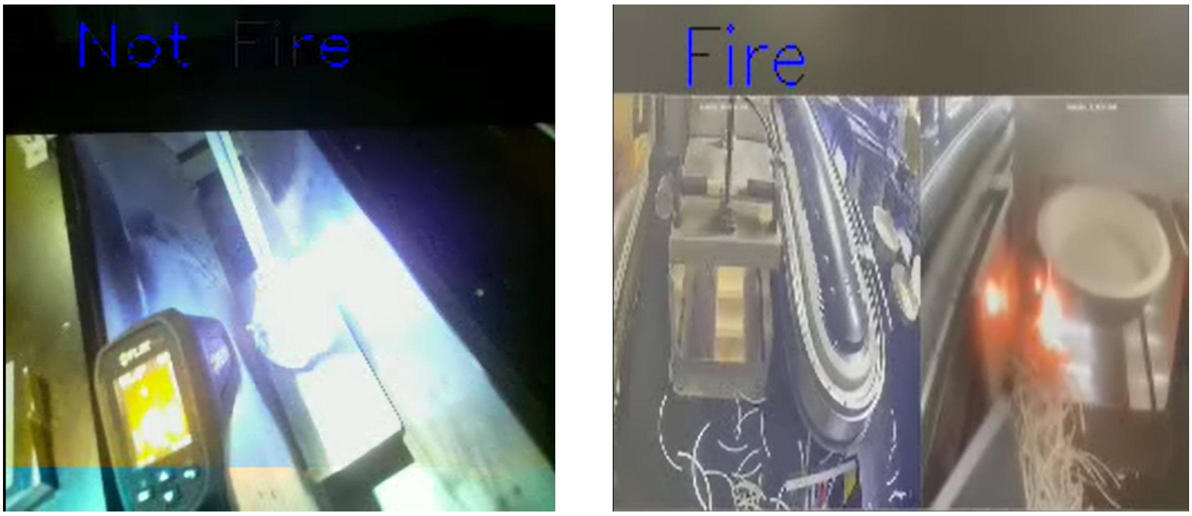


Fig.6: Classification outputs for the model deployed on Raspberry Pi

By observing the above images, It is evident that our model which is deployed on the Raspberry Pi is able to detect and classify the fire activity over a given sequence of frames.

Fig.7: Detection and Localization output on a PC

## 5. CONCLUSION

The proposed method has been experimentally proven to provide excellent fire detection accuracy, by reducing the false detections and misdetections, and to successfully interpret the temporal behavior of flame and smoke, which possibly reduces the false dispatch of firemen. In addition, we have constructed a large fire dataset which contains Aerospec data of the fire accidents, images and video clips from our experiments and from the popular public datasets. This dataset could be an asset for the future fire research

## ACKNOWLEDGEMENTS

We would like to thank Dr. Sangram Redkar and Aerospec Inc. for providing us this opportunity to work and provide solutions for a real-time problem.

## REFERENCES

[1] [1] K. Muhammad, J. Ahmad, I. Mehmood, et al., Convolutional neural networks based fire detection in surveillance videos, IEEE Access 6 (2018) 18174–18183.
[2] C. Tao, J. Zhang, P. Wang, Smoke detection based on deep convolutional neural networks, in: 2016 International Conference on Industrial Informatics -
Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), 2016, pp. 150–153.
[3] A. Filonenko, L. Kurnianggoro, K. Jo, Comparative study of modern convolutional neural networks for smoke detection on image data, in: 2017 10th International Conference on Human System Interactions (HSI), 2017, pp. 64–68.