© APR 2025 | IRE Journals | Volume 8 Issue 10 | ISSN: 2456-8880

Deep Learning-Based Fire Detection System

UTKARSHA PANDHARKAR[1], PROF. DR. DIPALI ADHYAPAK[2]
[1] *Student, Dept. ENTC, PES Modern college of Engineering, Pune.*
[2] *Assistant Prof., PES Modern college of Engineering, Pune.*

Abstract- In emergency situations like fire accidents fire causes environmental disaster, economic losses and deep damages to humans. To detect fire earlier and to avoid damages caused by fire we have built a project which is deep learning-based fire detection system. With the help of deep learning algorithms such as CNN we have developed this fire detection system. In this project we have introduced a system which is trained by using images as a dataset and with the help of CNN algorithm module is trained to detect fire from provided video. Input is provided in video format the input video is converted into images with the help of CNN layers. The trained module detects fire which is trained and working with the help of coding. The deep learning-based fire detection gives output in text format as 'fire detected' and alert is given to the nearest fire station or owner of the system is alerted by sending messages. This system results into early and accurate detection of fire through video or surveillance camera. This is efficient to early detection and fire is avoided by spreading. With this approach, fire detection systems are likely to become far more precise and effective.

Indexed Terms- Automated Fire Detection, Convolutional Neural Network (CNN), Early Fire Detection, Fire Prevention, Pattern Recognition, Neural Networks

I. INTRODUCTION

The number of fire incidents has been rising each year, largely due to human activities and increasingly dry climates. To avoid disaster of fire, many fire detection techniques have been widely studied to apply in practice. Most of traditional methods are based on sensors due to its low-cost and simple installation. These systems are not applicable for using outdoor where energy of flame affected by fire materials and the burning process affected by environment that has potential cause of false alarms.

To avoid this, Deep Learning based Fire Detection system is introduced. The need of this project is that it is able to detect fire from video or surveillance cameras which is quick and can be further modified into real time data. Visual-based approach of image or video processing was shown to be more reliable method to detect the fire since the closed-circuit television (CCTV) surveillance systems are now available at many public places, can help capture the fire scenes.

The current generation is more into artificial intelligence which involves deep learning and machine learning due to which it is able to do tasks like recognize patterns, make decision and judge like humans is possible by taking advantage of this we have built this system by using deep learning which attempts to simulate the behavior of the human brain. The motivation of this project is that it doesn't need any human interaction. The algorithm becomes more correct by experience or learning for algorithm it is a continuous process. Safety plays a vital role in the design of both residential and commercial buildings to help prevent loss of life and property damage.

Aim is to build a system which helps to detects the fire. In this project by using CNN algorithm, we have provided Input dataset from Kaggle datasets in video format. The input video undergoes preprocessing before it is fed to CNN algorithm which works on feature extraction of images. Further data segmentation and classification is done on that basis of fire is detected. This trained module detects fire which is trained with the help of coding. The deep learning-based fire detection gives output in text format as 'fire detected' and alert is given to the nearest fire station.

Objective: -

1. The input is provided in video format to the module.
2. To trained CNN will convert video into frames and feature detection will be done.
3. The fire will be detected and alert will be sent to the user.

4. The ultimate goal of system is to reduce the damage caused by fire by detecting them early.

II. LITERATURE REVIEW

In this section analysis of some existing fire detection system was carried out. This section provides a comparative analysis between the related works and our system.

In “Research on Image Fire Detection Based on Support Vector Machine” Ke Chen Yanying Cheng proposed that to detect and alarm early fire timely and effectively. Traditional temperature and smoke fire detectors are vulnerable to environmental factors such as the height of monitoring space, air velocity, and dust. They have introduced Vector machine which is proposed by studying the features of fire in digital image The flame color moment feature and texture feature are extracted and input into the support vector machine for classification and recognition. Data sets were formed by collecting Internet resources and fire videos taken by one and the trained support vector machine was tested. [1]

In their paper "Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications," Khan Muhammad and Jamil Ahmad highlight that while CNN-based fire detection systems offer strong performance, their high memory and computational demands pose challenges for real-world surveillance deployment. In this paper, they propose an origin, energy-friendly, and computationally efficient CNN architecture, inspired by the SqueezeNet architecture for fire detection. [2]

Huang hongyu1, kuang ping1 proposes in improved YOLOv4 fire detection method based on CNN. They have improved the accuracy of the model through the self-built high-quality fire dataset, use the changed loss function to improve the detection ability of small-scale flames it has best performance for detect multi-scale fire in real-time. [3]

In their paper "Prototype of Fire Symptom Detection System," Oxsy Giandi and Riyanarto Sarno proposed a novel fire detection approach that analyzes gas leak concentrations to predict potential explosions and fires at an earlier stage. The fire predictor just shows the gas leak concentration and makes an alarm range. The simulation system output can send the data to MFC, but the MFC reader cannot parse it in real time. [4]

Jiang Feng et. Al. have proposed a system with lightweight direct regression detection algorithm. Using YOLOv3-tiny, we developed a compact local video detection system for ship fires. Through video testing and fire simulations, the RpiFire System demonstrated high accuracy and recall rates, proving its effectiveness for real-world ship fire detection. It is, based on the Raspberry Pi hardware conditions and the Keras deep learning framework. [5]

In “Early Fire detection system using deep learning and Open CV CNN by AlexNet architecture which allowed vision-based systems to detect Fire using Convolutional Neural Networks during surveillance. This model was proposed by Dhruvil Shah. CNN model has been Implemented for a cost-effective fire detection. [6]

Forest fire detection using machine learning Published by Georgie Vadakkadath Rajang, Sinumol Paul has proposed a novel system for detecting fire using CNN. This paper critically examines the potential of Artificial Intelligence (AI) in detecting threats and issuing alerts using video feeds from CCTV surveillance systems. In this project the test set of the dataset is given as input for Validating the algorithm and Experiments. [7]

III. METHODOLOGY

This section outlines the overall architecture, data processing pipeline, and model development approach used to implement the deep learning-based fire detection system.

System Overview

The proposed system is designed to process surveillance videos and detect the presence of fire using a trained Convolutional Neural Network (CNN). The core methodology involves converting video frames into images, preprocessing these images, and passing them through a trained CNN model for classification. Upon detecting fire, the system triggers an alert via an automated email notification to registered users or authorities.

Dataset Preparation

We utilized publicly available fire image datasets from Kaggle, consisting of fire and non-fire images and videos. The dataset was divided into two categories:

- Fire Images: Images that depict actual fire or flames.
- Non-Fire Images: Normal scenes without any flames.

All images were resized to 100x100 pixels to optimize training speed and reduce computational requirements. The dataset was then stored in .npy arrays for efficient loading. Labels were assigned as 0 for non-fire and 1 for fire.

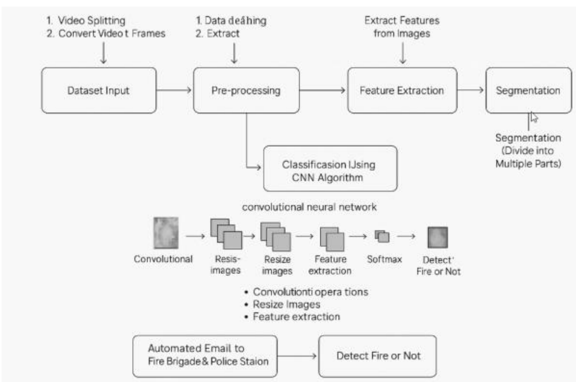

Figure 3.1. Block Diagram

Preprocessing

Data preprocessing includes:

- Resizing all images to 100x100 pixels
- Normalizing pixel values to the range [0, 1] by dividing by 255
- Converting colour spaces using OpenCV
- Splitting the dataset into training and testing sets
- Applying Gaussian blur to reduce noise
- Augmenting data to increase variability and robustness

Example preprocessing commands:

```
X_img = img.reshape(-1, img_cols, img_rows, 1)
X_img = X_img.astype('float32')
X_img /= 255
```

Feature Extraction

Feature extraction was automatically handled by the CNN model. Features such as shape, texture, and colour patterns associated with fire were learned by the network during training. OpenCV, TensorFlow, and Keras were used for efficient image processing and model development.

CNN Architecture and Training

The model was based on a modified version of the AlexNet architecture, using transfer learning. The architecture includes:

- Input layer: 224x224x3 image
- Convolution and pooling layers for feature extraction
- Fully connected layer with 2048 neurons
- Output layer with Softmax activation for binary classification (Fire, No Fire)

Transfer learning helped improve model accuracy with limited training data by leveraging pre-trained weights from AlexNet.

Segmentation and Classification

To enhance detection, segmentation techniques such as thresholding were applied to isolate potential fire regions. OpenCV methods like cv2.threshold() and cv2.adaptiveThreshold() were used.

Images were classified based on the features extracted by the CNN. If fire was detected, the system sent an automated alert.

Implementation Details

- Hardware: Intel Core i5, 8GB RAM, 500GB HDD
- Software: Windows 10 (64-bit), Python 3.7/3.8, Spyder IDE
- Libraries: TensorFlow, OpenCV, Keras, NumPy

This methodology ensures a robust, scalable, and accurate fire detection solution capable of working with real-time surveillance video.

IV. ALGORITHM

Step 1: Start.

Step 2: Run GUI main for the purpose of registering new user in the database.

Step 3: Enter all details including username and Password to access the Admin Mode and register.

Step 4: Open login page enters username and password if password is not correct then Error message will be displayed and if password is correct then it'll proceed further to take choice from the user

Step 5: Select fire detection button make choice of available videos

Step 6: Processing of video will do using CNN.

Step 7: If fire detected mail will send.

Step 8: Normal events also detected

Step 9: fire detected mail sends to registered user
Step 10: End.

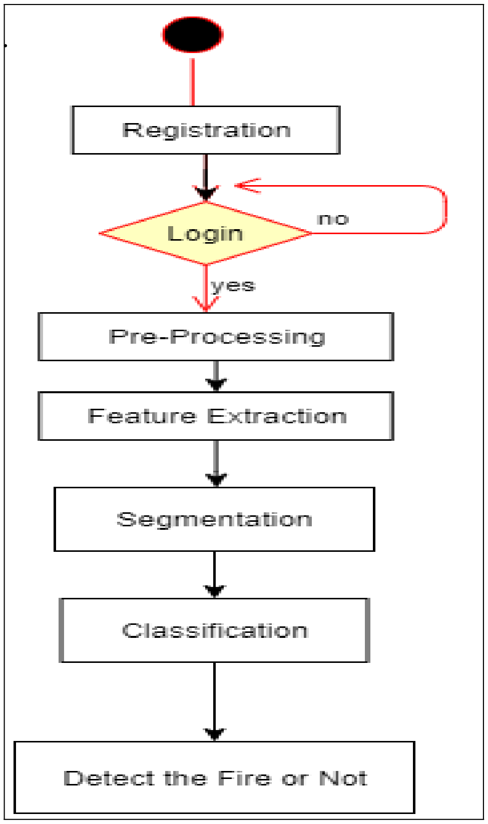

Figure 3.1. Work flow Diagram

CNN ALGORITHM

In this project Convolution Neural Network is used because of its image recognition capabilities. It is designed to automatically learn and extract features from input data, making it well-suited for tasks such as object detection, image classification and image segmentation.

CNNs are widely used for image classification because they typically deliver higher accuracy across various datasets compared to traditional feature engineering methods. This is due to the deep learning of features from raw data in the training process. CNNs consist of three main layers. The first layer is the convolutional layer, where filters (kernels) of various sizes process the input data to generate feature maps. The next is the pooling layer, which performs subsampling to reduce the dimensionality of these features. The proposed architecture consists of 120 layers, including two convolutional layers, four pooling layers, one average pooling layer, and a SoftMax classifier, as shown in the diagram.

In Convolution layer train the data using AlexNet. AlexNet classification precision with, small size architecture, and less memory requirements. The proposed architecture consists of 120 layers, including two convolutional layers, four pooling layers, one average pooling layer, and a SoftMax classifier, as illustrated in the diagram.

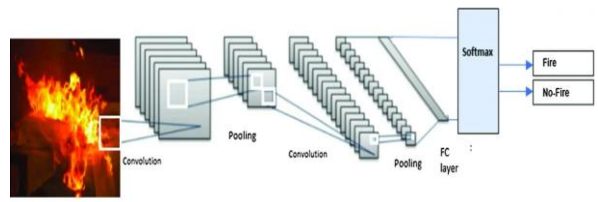

Figure 3.2. CNN(Network)

Layer number	Network layer	Description
1	Input layer	Size of input image 224*224*3
2	First convolution layer	360*6*1 convolutions
3	First pooling layer	Max pooling
4	Second convolution layer	128(3*3*16) convolutions
5	Second pooling layer	4*4 max pooling
6	Third convolutional layer	256(3*3*16)
7	Third pooling layer	4*4 max pooling
8	Fully connected layer	2048 hidden neurons
9	Classifier	SoftMax
10	Outputs	Two outputs,1: Fire, 2: No-Fire

V. TRAINING AND TESTING MODELS

Training and Testing are depicted in this section. We performed extensive experiments utilizing different images and surveillance videos.

1. Experiment 1: The Deep-CNN is tested with both datasets (Dataset1 and Dataset2). The performance evaluation of the classifier is tested through runs of convolution, pooling and classification layers.
2. Experiment 2: The Deep-CNN is also tested with both datasets (Dataset1 and Dataset2). The Deep learning network is joined with transfer learning. The proposed Deep-CNN model is trained on two datasets (Dataset1 and Dataset2), utilizing transfer learning from the pre-trained AlexNet architecture.

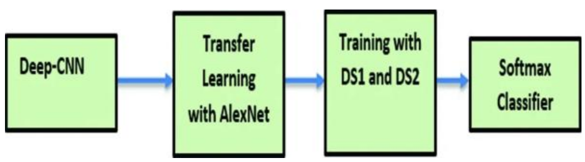

Figure 5.1. CNN(Network)

Performance of Dataset1:
The dataset Dataset1 contains 100 videos with different day and night size and distance variations. Dataset1 has 90 videos that include real fire flames, and 80 videos without flames. Dataset1 is a good example of a fire detection with different settings of colors and motion. Also, videos, with no fires, contain rigid objects that look like fires, clouds, and flames.

Models	Methodology	False Positive	False Negative	Accuracy
Exper1: Our proposed model Without transfer learning	Deep-CNN	1.50%	6.20%	90.30%
Flame detection	Covariance Matrix	11.65 %	0.50%	93.56%
Wildfire Detection system	Deep learning	17.50 %	7.80%	87.78%
Fire detection in video sequences	Generic color model	22.76 %	1.29%	83.60%

Performance on Dataset2

Dataset2 contains 200 images, including 100 images with fire and 100 normal images without flames. The size of Dataset2 seems small but it contains many important challenges Dataset2 includes non-flame objects that exhibit flame-like colors, as well as various scenes featuring sunlight in different conditions.

Models	Methodology	False Positive	False Negative	Accuracy
Exper2: Our proposed model with transfer learning	Deep CNN with transfer learning	0.23%	1.30%	96.70%

Results:

Model	Precision	Recall	F-Measure
Our proposed model without Transfer learning	0.95	0.945	0.94
Our proposed model with Transfer learning	0.98	0.975	0.97
Wildfire Detection	0.89	0.9	0.896
Early fire detection	0.825	0.9	0.89
Early fire detection with Transfer learning	0.92	0.93	0.935
Irregular fire flames	0.9	0.91	0.9

VII. RESULT

Final output will be the automated email generated by model, in the system where the model is imposed. Email contains an emergency message “Fire Detected”.

1. Website is created with the registration form by using Python. In this registration form, user needs to fill all the details including name, username, Password or email and address.

Fig. 7.1. Registration page

2. Account of user is created in the system. Next step is to login to the system using username or password which we have created while registration.

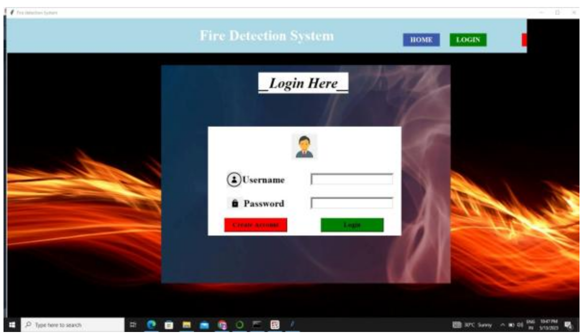

Fig. 7.2. Registration page

3. Video is selected for data set as an input by selecting fire detection button. For this, both dataset videos are used as video with fire and not fire.

Fig. 7.3. Display page

4. Trained CNN model will detect the fire in this video. This is a screenshot in which fire is detected and message is displayed of Fire Detected.

Fig. 7.4. Fire detected output page

5. Result output when fire is not detected with message (normal event detected).

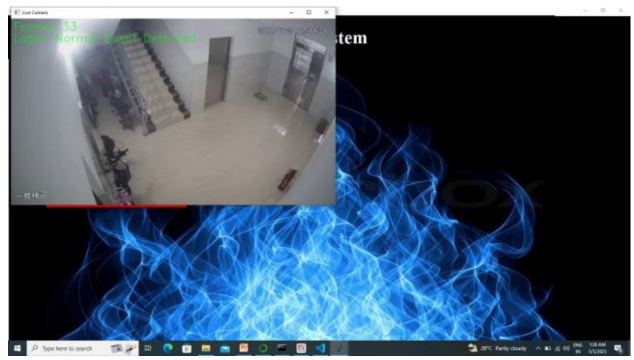

Fig. 7.5. Normal event detected output page

VIII. APPLICATIONS AND FUTURE MODIFICATION

Applications:

1. Fire Detection System.
2. Helps to Fire Fighting.

Future Modification:

Future research could focus on developing more challenging and specialized datasets for scene understanding in fire detection, along with conducting detailed experimental evaluations. Furthermore, reasoning theories and information hiding algorithms can be combined with fire detection systems to intelligently observe and authenticate the video stream and initiate appropriate action, in an autonomous way. In future live CCTV camera instead of video input.

CONCLUSION

CNN-based fire detection systems have proven to be effective and reliable in identifying and detecting fires in various environments. This study has demonstrated the effectiveness of CNN-based fire detection through comprehensive testing on a large-scale video dataset representing varied fire conditions. To improve detection performance, the system first extracts chromatic and motion characteristics of fire, then applies rule-based correction to accurately identify fire regions. In real-world fire surveillance, our framework demonstrates strong performance in real-world fire surveillance applications, achieving both high detection accuracy and low false alarm rates.

Despite their effectiveness, CNN-based fire detection systems may still have some limitations. They can encounter false positives or false negatives, where non-fire objects or actual fires may be misclassified. However, continuous improvements in CNN architectures, training techniques, and dataset quality are helping to minimize these issues.

Overall, CNN-based fire detection systems offer a promising approach to enhance fire safety and prevention measures. With further advancements in technology and ongoing research, these systems are expected to become even more accurate, reliable, and widely adopted in various settings, including residential, commercial, and industrial environments.

REFERENCES

[1] Ke Chen, Yanying Cheng, “Research on Image Fire Detection Based on Support Vector Machine”, (2020), IEEE

[2] Khan Muhammad Student Member, IEEE Jamil Ahmad, Student Member IEEE, Zhihan L Member IEEE Paolo Bellavista Senior Member IEEE Po Yang, Member, IEEE, and ung Wook Baik, Member, lEEE,’ efficient Deep NN-Based Fire Detection and Localization in Video Surveillance application’s' 2018 IEEE, http://www.ieee.org/publications- standard /publication /rights/index.html

[3] HUANG HONGYUl, KUANG PING I LIFANJ HI HUAXINJ "AN Ilvl- PROVED MULTI-SCALE FIRE DETECTIO METHOD BASED ON CONVO- LUTIO AL NEURAL NETWORK, 978-l-6654-0505-8/20/3J .00 © 2020 IEEE.

[4] Oxsy G.iandi, Riyanarto Sarno Prototype of Fire Symptom Detection System' 978-l-5386-0954-5/18/3 l.00©2018 IEEE.

[5] Jiang Feng, Yang Feng, "Design and experimental research of video detection sy tern for hip fire ©2019 IEEE.

[6] Sneba Wil on 2Shyni P Varghe e, 3Nikhi1 G A, 4 ManoJekshmi I, 5 Raji PG, A Comprehensive Study on Fire Detection', Proc. IEEE Conference on Emerging Devices and Smart Systems (ICEDSS 2018).

[7] C. Kao and. bang • an lnteUigent Real-Time Fire-Detection Method Based on Video Processing,' IEEE 37th nun. 2003 Int. Carnahan Conf. OnSe u1ity Tecbnol 2003 Proc. 2003.