Enhanced Forest Fire Detection Using Deep Learning

KUNU SIREESHA1, AKSHAYAM PRASMITA2

#1Assistant Professor, Department of CSE-IoT, PBR Visvodaya Institute of Technology and Science, Kavali

#2 Assistant Professor, Department of CSE, PBR Visvodaya Institute of Technology and Science, Kavali

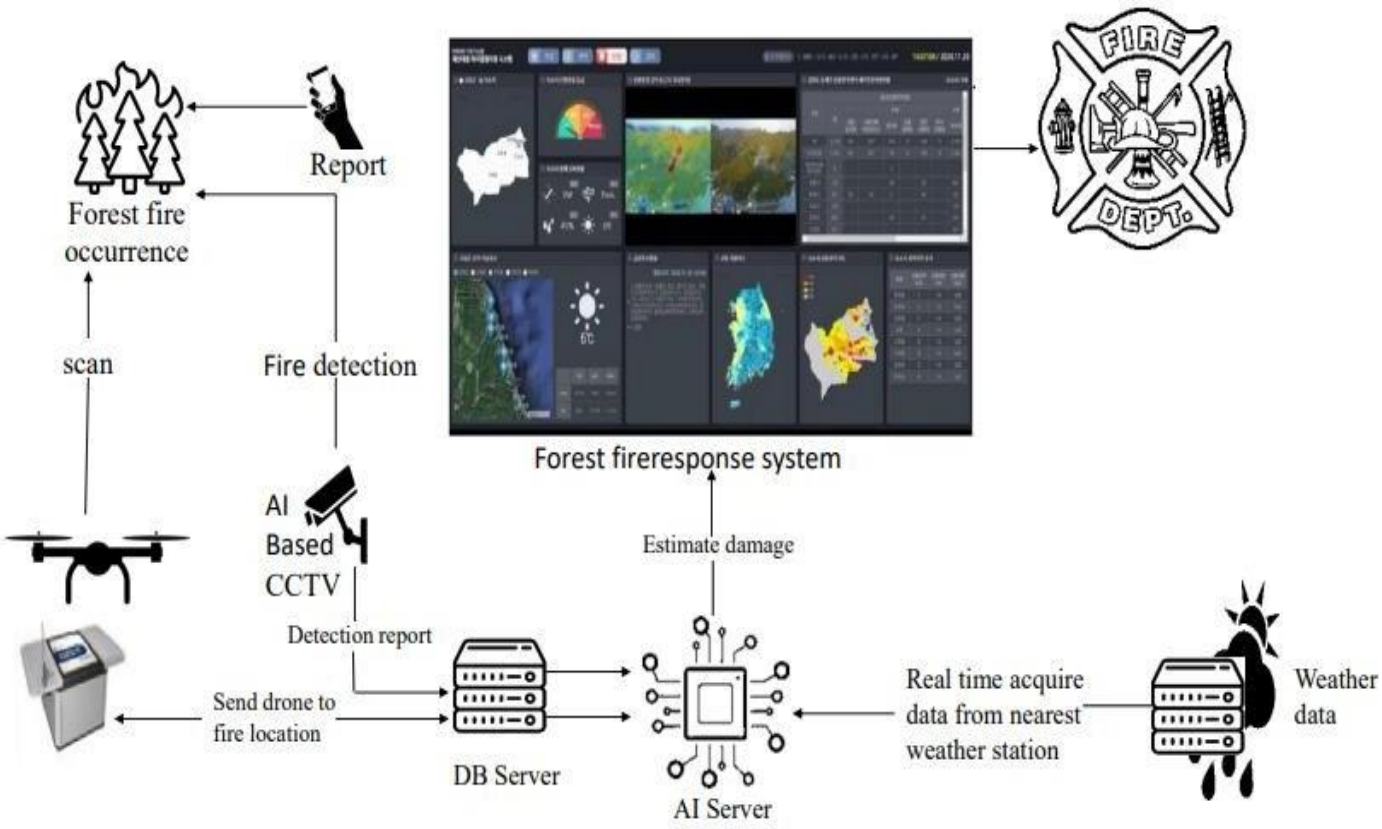

Figure 1: block diagram of deep learning- based forest fire response system

Figure 1 shows a block diagram of a deep learning-based forest fire response system, which includes a data acquisition module, a deep learning model for fire detection, a decision-making module, and a response module for initiating appropriate actions in response to detected fires.

3.1 IMPLEMENTATION

3.1.1 Image Acquisition

Image acquisition can be defined as the act of procuring an image from

sources.This can be done by hardware system such as cameras and datasets and also some encoders sensors also take place in this process.

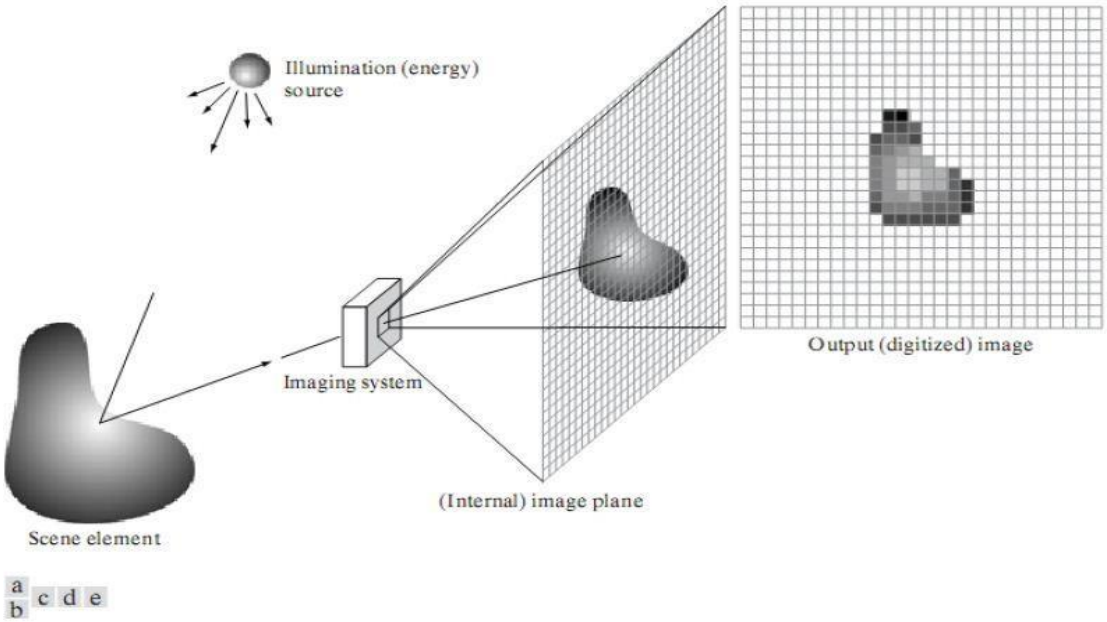

Figure 2: image aquistion

3.1.2 Pre Processing

In this step, the acquired images are pre-processed to enhance image quality and correct for any distortions or noise. This may involve techniques such as image filtering, color correction, and image registration.

Image processing

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually, Image Processing system includes treating images as two-dimensional signals while applying already set signal processing methods to them.

It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research

area within engineering and computer science disciplines too.

Image processing basically includes the following three steps.

- Importing the image with optical scanner or by digital photography.
- Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
- Output is the last stage in which result can be altered image or report that is based on image analysis.

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies.

To get over such flaws and to get originality of information, it has to undergo various phases of processing.

The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.

For this article we'll be using the following image:

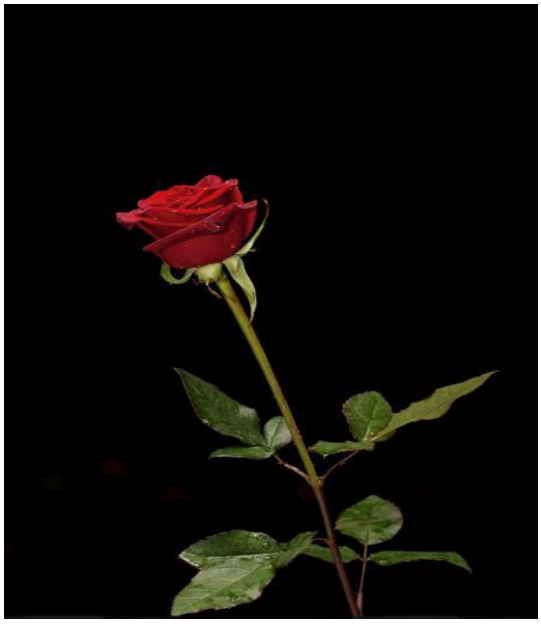

Figure 3: image used for image processing

Note: The image has been scaled for the sake of displaying it in this article, but the original size we are using is about 1180x786.

You probably noticed that the image is currently colored, which means it is represented by three color channels i.e., Red, Green, and Blue. We will be converting the image to grayscale, as well as splitting the image into its individual channels using the code below.

3.1.3 Finding Image Details

After loading the image with the imread() function, we can then retrieve some simple propertiesabout it, like the number of pixels and dimensions:

```
import cv2
img = cv2.imread('rose.jpg') print("Image Properties")
print("- Number of Pixels: " + str(img.size))
print("- Shape/Dimensions: " + str(img.shape))
```

output:

Image Properties

- Number of Pixels: 2782440

- Shape/Dimensions: (1180, 786, 3)

Now we'll split the image in to its red, green, and blue components using OpenCV and display them:

```
from google.colab.patches import cv2_imshow
blue, green, red = cv2.split(img) # Split the image into its channels
img_gs = cv2.imread('rose.jpg', cv2.IMREAD_GRAYSCALE) # Convert image to grayscale
cv2_imshow(red) # Display the red channel in the image
cv2_imshow(blue) # Display the red channel in the image
cv2_imshow(green) # Display the red channel in the image
    cv2_imshow(img_gs) # Display the grayscale version of image
```

For brevity, we'll just show the grayscale image

Figure 4: grayscale image

3.1.4 Image Thresholding

The concept of thresholding is quite simple. As discussed above in the image representation, pixel values can be any value between 0 to 255. Let's say we wish to convert an image into a binary image i.e., assign a pixel either a value of 0 or 1. To do this, we can perform thresholding. For instance, if the Threshold (T) value is 125, then all pixels with values greater than 125 would be assigned a value of 1, and all pixels with values lesser than or equal to that would be assigned a value of 0. Let's do that through code to get a better understanding.

Figure :5. Image used for Thresholding

import cv2

```
# Read image
img = cv2.imread('image.png', 0)
# Perform binary thresholding on the image with T = 125
r,threshold=cv2.threshold(img,125,255,cv2.THRESH_BINARY)
cv2_imshow(threshold)
```

output:

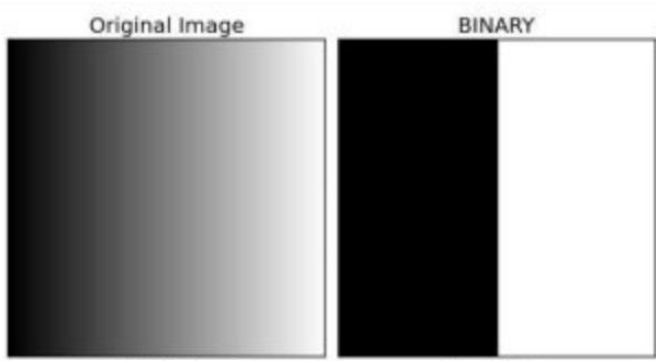

Figure 6 : resultant image of thresholding

As you can see, in the resultant image, two regions have been established, i.e., the black region (pixel value 0) and white region (pixel value 1). Turns out, the threshold we set was right in the middle of the image, which is why the black and white values are divided there.

3.1.5 Image Segmentation

Image segmentation is the process of dividing an image into multiple regions based on similarities in color, texture, or other features. This step is used to identify regions of interest in the image, such as objects or areas with specific characteristics.

3.1.6 Feature Extraction

In this step, key features of the segmented regions are extracted to represent the image in a more compact and meaningful way. These features may include shape, texture, color, or other properties that can be used to identify or classify the image.

3.1.7 Classification

The extracted features are used to classify the image into one or more predefined categories, based on the intended application of the image processing. This may involve using machine learning algorithms, such as neural networks or support vector machines, to train a model to recognize and classify images.

3.1.8 Post Pre-processing

Finally, the classified images may undergo post-processing, which may involve further enhancement, filtering, or compression to prepare them for storage or display.

4.RESULTS AND DISCUSSION

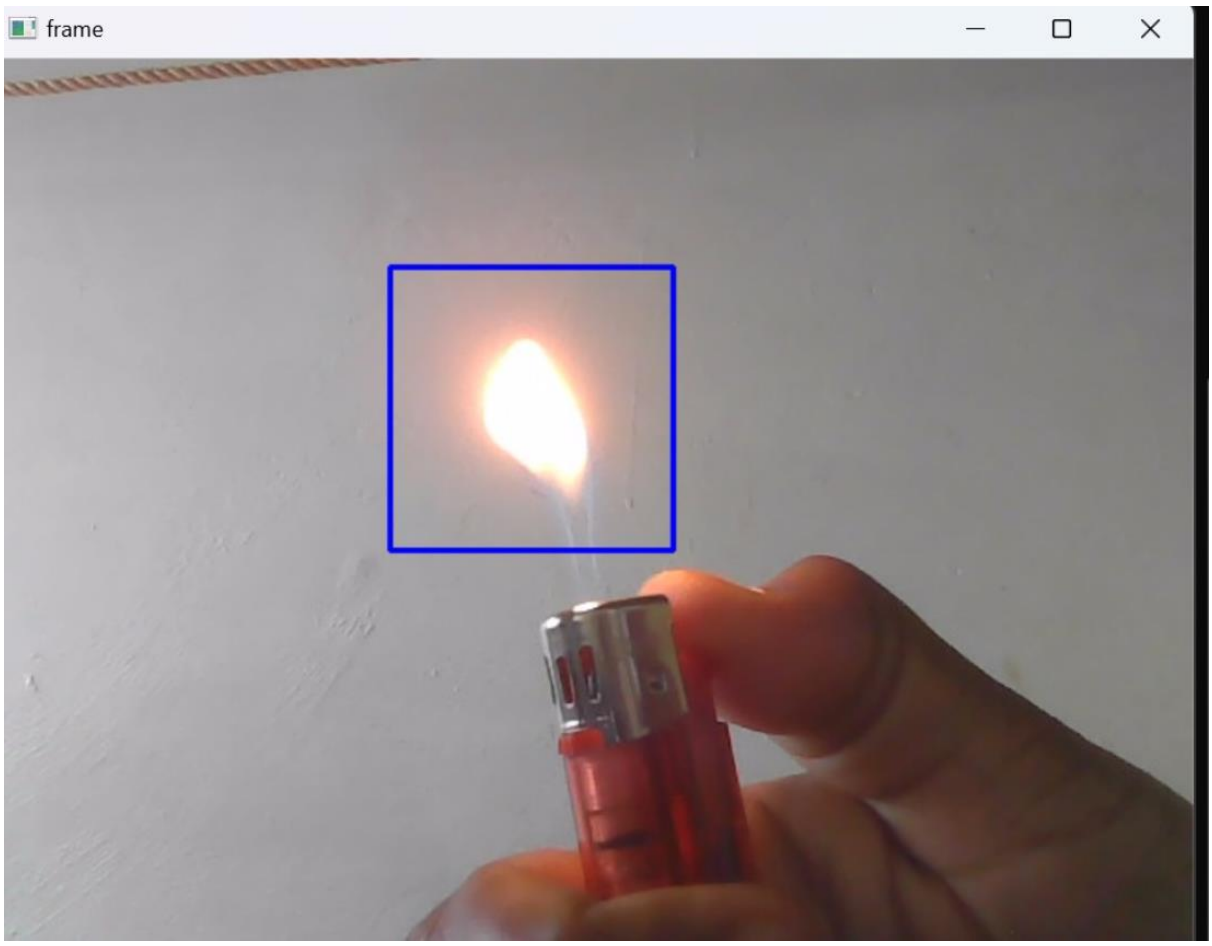

Fig 7:Fire Detected

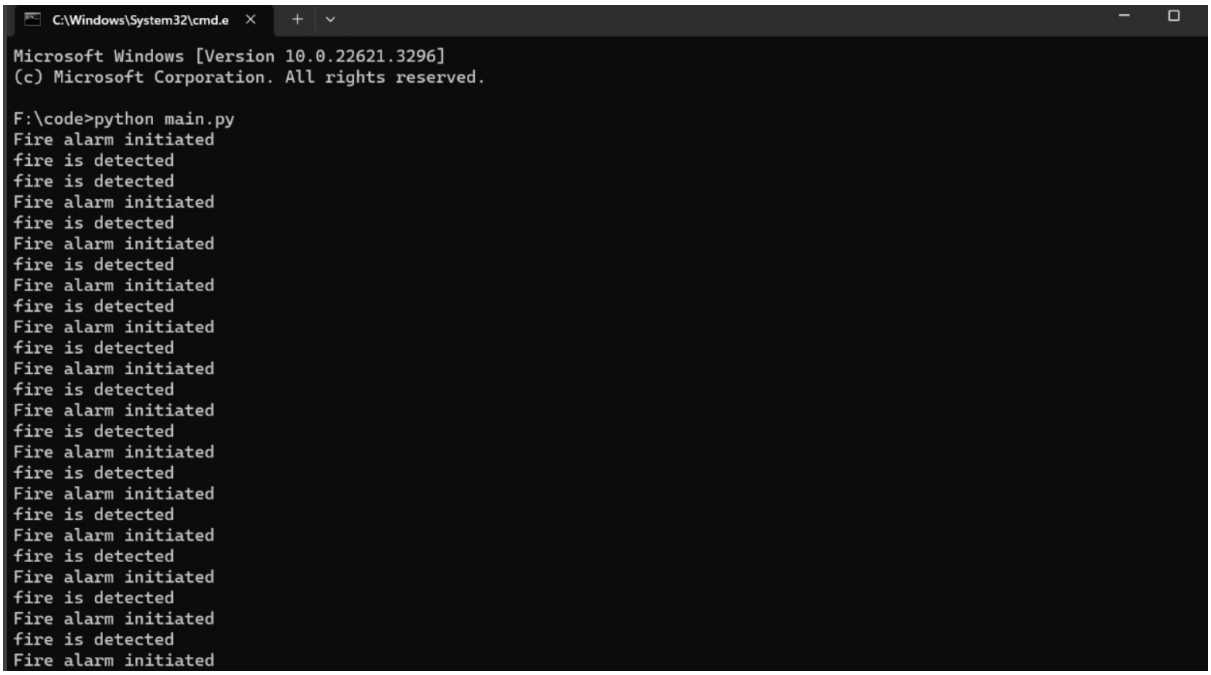

Fig8:Fire Alert Initialed

5.CONCLUSION

An important development in forest fire management is the deep learning-based forest fire response system. Through the use of real-time monitoring and deep learning algorithms, the system improves resource allocation, response time, and detection accuracy. By protecting people, property, and the environment from forest fires, this research may help make response operations more successful.

REFERENCES

[1] A.G. Howard, M. Zhu, B. Chen, D. Kanlenichenko, W. Wang, T. Weyand, M. Andreetto, and H, Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861 (2017).

[2] K.Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” CoRR, abs/1409.1556, 2014.

[3] M. Khan, J. Ahmad, and S.W. Baik. "Early fire detection using convolutional neural networks during surveillance for effective disaster management," Neurocomputing 288 (2018): pp. 30-42.

[4] P.V.K. Borges, E.

Izquierdo,"A probabilistic approach for visionbased fire detection in videos," IEEE Trans. Circuits Syst. Video Technol, 20.5 (2010), pp. 721–731.

[5] R.Bright and R.Custer, "Fire detection: The state of the art," NBS Technical Note, US Department of Commerce, 1974.

[6] S. Frizzi, R. Kaabi, M. Bouchouicha, J. M. Ginoux, E. Moreau, and F. Fnaiech, "Convolutional neural network for video fire and smoke detection,"In IECON 2016 42nd Annual Conference of the IEEE Industrial Electronics Society, Oct 2016, pp. 887-882.

[7] S. Hochreiter,J. Schmidhuber, "LSTM can solve hard long time lag problems," In Advances in Neural Information Processing Systems; NIPS: San Diego, CA, USA, 1997; pp. 473–479.

[8] S. Kethavath, and M. Dua. "Early Discovery of Disaster Events from Sensor Data Using Fog Computing," International Conference on Intelligent Computing, Information and Control Systems. Springer, Cham, 2019.

[9] S. Kethavath, and M. Dua. "Fog Computing and Deep CNN Based Efficient Approach to Early Forest Fire Detection with Unmanned Aerial Vehicles," International Conference on Inventive Computation Technologies. Springer, Cham, 2019.

[10] V.Vipin, " Image processing-based forest fire detection,"International Journal of Emerging Technology and Advanced Engineering, 2.2 (2012), pp. 87-95.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haoner,"Gradient-based learning applied to document recognition," Proceedings of the IEEE, Nov 1998, 86(11), pp. 2278-2324.