

Fire Detection System using Python Open CV

Patil
Computer Engineering
Guru Gobind Singh
Polytechnic Nashik,India
pranavpatil2108@gmail.com

Bhavesh Nemade
Computer Engineering
Guru Gobind Singh Polytechnic Nashik,India
bnemade2845@gmail.com

Piyush More
Computer Engineering
Guru Gobind Singh Polytechnic Nashik,India
piyushmore292@gmail.com

Uday Sanap
Computer Engineering
Guru Gobind Singh Polytechnic Nashik,India
udaysanap20@gmail.com

Abstract— The technologies underlying fire systems play a crucial role in ensuring and delivering optimal performance in modern surveillance environments. In fact, fire can cause significant damage to lives and properties. Considering that the majority of cities have already installed camera-monitoring systems, this encouraged us to take advantage of the availability of these systems to develop cost-effective vision detection methods. However, this is a complex vision detection task from the perspective of deformations, unusual camera angles and viewpoints, and seasonal changes. To overcome these limitations, we propose a new method based on a deep learning approach, which uses a convolutional neural network that employs dilated convolutions. We evaluated our method by training and testing it on our custom-built dataset, which consists of images of fire that we collected from the internet and labelled manually. Our experimental results indicate that the classification performance and complexity of our method are superior. In addition, our method is designed to be well generalized for unseen data, which offers effective generalization and reduces the number of false alarms.

Keywords—*Fire Detection, OpenCV, Python, Tkinter Camera*

I. INTRODUCTION

Despite the rapid growth of technologies and smart systems, certain problems remain unsolved or are solved with methods that deliver poor performance. One of these problems is the unexpected outbreak of a fire, an abnormal situation that can rapidly cause significant damage to lives and properties. According to the Korea Statistical Information Service, the National Fire Agency recorded that, during the three years from 2016 to 2018, 129,929 fires occurred in South Korea, resulting in 1020 deaths, 5795 injuries, and damage to properties estimated at USD 2.4

billion . The latest technological advancements in sensors and sensing technologies have inspired businesses to determine whether these improvements can help to reduce the damage and harm caused by fire. This is the most frequent and widespread threat to public and social development as well as to individuals' lives. Although fire prevention is the top priority to ensure fires do not occur in the first place, it is nonetheless essential to spot fires and to extinguish them before they have serious consequences. In this regard, a large number of methods were introduced and tested for early fire detection to reduce the number of fire accidents and the extent of the damage. Accordingly, different types of detection technologies in automated fire alarm systems have been formulated and are widely implemented in practice.

II. EASE OF USE

A. Machine Learning

Machine learning is the study of the algorithms and the statistical models that computer system uses for the specific task without the use of instruction or relying on some patterns and 13 interferences instead. A machine learning algorithm is created by collecting the data and then represent in the mathematical model, which helps to make the predictions and make the decision referencing from the mathematical model. Machine Learning is used in a wide variety of field like in object detection, sorting of packages, filtering of documents, predicting the patterns which help perform the task very fast and more precisely than human. Machine learning is the science of making computers learn and act like humans by feeding the data and information without being explicitly programmed. Generally, machine learning has a different way of learning

B. Object Detection

Object detection is the computer vision that deals with the identifying of the object and locating the object with certain classes in it with the help of the pictures, videos, or the camera feed. Object detection technology is used almost everywhere these days to make works easy, safe, and fast. The use of this technology is endless, like video surveillance, face detection, self-driving vehicles, cell count, and the list goes on. Object detection before the advancement in deep learning was very lengthy and had several steps processes to execute. The first-time face detection work in 2001 with the help of a violone’s algorithm. They introduce to detect face by the help of webcam their algorithm only detects the face suppose if the face is in a different angle or downwards, then it would not detect. For this project, the primary process of detecting fire is inspired by the same algorithm proposed by the Viola-Jones face detecting algorithm

III. DESIGN AND IMPLEMENTATION

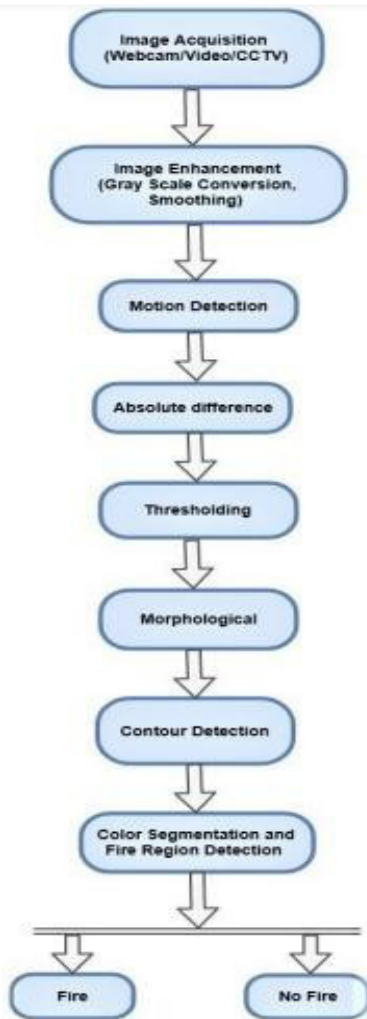


Fig Block Diagram

Step 1- Image Acquisition

At first the video camera is on and then the first frame

captured is considered as initial background image. Then continuously frames are captured from the camera. The frame is taken in RGB color space. The procedure is given below:

```

    Import cv2
    video = cv2.VideoCapture(0)
    check, frame = video.read()
    # frame is a NumPy array, it represents the first image that
    video captures
    # check is a bool data type, returns true if python is able to
    read the videocapture object
    
```

Step 2- Gray Scale Conversion

All the images captured are converted into grayscale. Because in RGB format, the R, G, B values of each pixel differ from each other, but in gray level the R, G, B values are same.

So, it becomes easier to compare two images in grayscale format rather than in RGB format.

The procedure is given below:

```

    # Convert RGB to grayscale.
    grayImage = cv2.cvtColor(frame,
    cv2.COLOR_BGR2GRAY)
    
```

Step 3- Preprocessing- Preprocessed to enhance the image according to the specific task.

Preprocessing for- • Noise Reduction • Brightness and contrast enhancement

1) Smoothing There are several situations, when image quality is reduced due to noise. So, before any further processing, in this step, noise cleaning of image is done. Smoothing is often used to reduce noise within an image or to produce a less pixelated image.

Convert the gray scale frame to Gaussian Blur. The procedure is given below:

```

    Gray = cv2.GaussianBlur(gray,(21,21),0)
    
```

Step 4- Motion Detection

System looks for if there is any motion in consequent frames. For this, background subtraction is used. Foreground objects result by calculating the difference between an image in the sequence and the background image (previously obtained).

Then the movement of these foreground objects between successive frames are determined. By subtracting consequent frames, we find the area where motion occurred. The frames captured from camera is counted.

First frame is considered as initial background frame.

As time passes, background image is updated according to foreground image.

There are several methods for separating foreground from the background.

Since video is used, so some built-in function of OpenCV is used directly to achieve faster performance.

The procedure is given below:

```

    Delta_frame = cv2.absdiff(first_frame,gray)
    #calculate the difference between the first frame and other
    frames
    
```

Step 5- Morphological

Morphology is the process of object study and use to reduce noise. Sometimes, there can be local noisy high intensity changing with fire colors when there is no fire. To improve performance, erosion and dilation are applied. So, system can reduce noisy effects of mobile fire colored objects. Dilation and Erosion

- Dilation increase white pixels in the image
- Erosion decrease black pixels in the image The procedure is given below:

```
cvDilate(fire_image, fire_image, 0, 1);  
cvErode(fire_image, fire_image, 0, 1);
```

Step 6- Contour detection

Define the contour area. Basically, add the borders. The procedure is given below:

```
(_,cnts,_)=cv2.findContours(thresh_delta.copy()  
,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
```

for contour in cnts:

```
    if cv2.contourArea(contour)<1000;  
        continue  
    print('1')
```

```
(x,y,w,h) = cv2.boundingRect(contour)  
Cv2.rectangle(frame, (x,y), (x+w, y+h), (0, 255, 0), 3)
```

Step 7- Color Segmentation and Fire Detection

Finding Fire Pixels

Converting grayscale image into RGB image.
Backtorgb=cv2.cvtColor(gray,cv2.COLOR_GRAY2RGB)

To find possible fire pixels, firstly we find fire colored pixels according to RGB values of video frames. We used following RGB values to detect fire:

```
R>220 G>200 124<B<185  
R>220 125<G<B<100  
R>220 175<G<B<125
```

Most of the fire colored pixels is in these three ranges. So, if one pixels RGB values are in these ranges, it is fire colored pixel. Another property to look is fire pixels have a relation among its color channels that is : $R > G > B$.

Some Common Mistakes

- Not considering the environment: Fire detection systems need to work in various lighting and temperature conditions, and these conditions can affect the accuracy of the system. Developers need to consider these environmental factors and design the system accordingly.
- Using low-quality images or videos: The quality of

the images or videos used for fire detection can significantly impact the accuracy of the system. Developers should use high-quality images or videos for training and testing the system.

- Not using enough data for training: Deep learning algorithms need a large amount of data to train on to achieve high accuracy. If there is not enough data, the model may not be able to accurately detect fires.
- Not using the right algorithm: There are various algorithms that can be used for fire detection, and the choice of algorithm can impact the accuracy of the system. Developers need to choose an algorithm that is well-suited for their specific use case.
- Overfitting the model: Overfitting occurs when the model is too closely fit to the training data and does not generalize well to new data. This can result in poor performance when detecting fires in real-world situations.
- Ignoring false positive detections: False positive detections can occur when the system incorrectly identifies a non-fire event as a fire. This can lead to false alarms and cause problems for the end-users of the system. Developers should design the system to minimize false positive detections.

IV. OPEN CV

OpenCV is an open-source package and machine learning library, which is designed for Realtime computer vision applications. Open CV is the cross-platform library that supports many programming languages like Python, Java, C++, C, etc. This library is originally designed by the Intel corporation and free for use under the open-source BSD license. It is one of the most widely used libraries for implementing video detection, image detection, deep learning application, machine learning, human-computer interaction, 2D, and 3D feature toolkits. The library has more than 2500 algorithms, which include both computer vision and machine learning algorithms. Some of the syntaxes that are used in the program are: Import cv2 # used for importing the open cv library Import NumPy as np # importing the NumPy array Cv2.VideoCapture() # used for capturing the video from the camera Cv2.cvtColor() # used for converting an image from one color space to another gray = cv2.cvtColor(resized, cv2.COLOR_BGR2GRAY)# used for converting Bicolor to gray fire = Bicolor(gray, 1.1, 7) # using fire classifier in which objects are loaded with the classifier in Multiscale A multi-scales detection algorithm is another important steps in the fire detection algorithm. This feature is used in order to detect the fire of any size because the size of the fire is unknown. Since learning and testing are based on rectangular feature, those features should be calculated in any scale.

A. Authors and Affiliations

1. ZHANG, Y., & LIU, Z. (2020). AN IMPROVED FIRE DETECTION METHOD BASED ON FASTER R-CNN. JOURNAL OF COMPUTERS, 15(11), 2894-2903.

2. NAFIE, M., & ALI, A. (2019). FIRE DETECTION USING DEEP CONVOLUTIONAL NEURAL NETWORK. INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS, 179(9).

3. JAIN, A., & JAIN, A. (2018). A REVIEW ON FIRE DETECTION USING IMAGE PROCESSING. INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER ENGINEERING & TECHNOLOGY (IJARCET), 7(7),69-74.

4. WANG, Y., & ZHANG, L. (2017). FIRE DETECTION BASED ON COMPUTER VISION AND MACHINE LEARNING. PROCEEDINGS OF THE 2017 INTERNATIONAL CONFERENCE ON COMPUTER VISION AND MACHINE LEARNING.

5. ZHANG, L., & WANG, Y. (2017). FIRE DETECTION BASED ON MACHINE LEARNING AND COMPUTER VISION. PROCEEDINGS OF THE 2017 INTERNATIONAL CONFERENCE ON COMPUTER VISION AND MACHINE LEARNING.

ACKNOWLEDGMENT

Computer vision-based detection system using color and motion properties of fire. Fire region detection approach tries to find the moving areas of the image. As in the real world, fire regions grows and spreads as the time passes. Finding the movement in the image reduces the candidate regions to detect fire color pixels. Contour detection is performed to find the moving regions of the image which is already segmented using color property. Then the boundary of the moving region is found and the amount of fire pixels in this area is calculated. Since fire region varies in a great extent, based on this approach can find whether there is really fire in the moving fire colored region or not.

REFERENCE

S

- [1] S. Y. Foo, "A fuzzy logic approach to fire detection in aircraft dry bays and engine compartments", IEEE Trans. Industrial Electronics, vol. 47, no. 5, pp. 1161-1171, 2000
- [2] B.U. Töreyn, Y. Dedeoglu, U. Gudukbay and A. E. Cetin, "Computer vision based method for realtime fire and flame detection", Patt. Recog. L., vol. 27, pp. 49-58, 2006.
- [3] C-C Ho, "Machine vision-based real-time early flame and smoke detection", Meas. Sci. Technol., vol. 20, pp. 1-13, 2009.
- [4] B.C. Ko, K-H Cheong and J-Y Nam, "Fire detection based on vision sensor and support vector machines", Fire Safety J., vol. 44, pp. 322-329, 2009.
- [5] M. Doostfateme and S. C. Kremer, "New directions in fuzzy automata", Int. J. of Approximate Reasoning, vol. 38, pp. 175-214, 2005.
- [6] F. Lin and H. Ying, "Modeling and control of fuzzy discrete event systems", IEEE Trans. On SMC-B, vol. 32, pp. 408-415, 2002
- [7] C. M. Bishop, "Pattern recognition and machine learning, " Springer Publisher, 2006.
- [8] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends, " Comput. Netw., vol. 51, no. 12, pp. 3448-3470, 2007. (Pubitemid 46921030)
- [9] W. Phillips III, M. Shah and N. V. Lobo, "Flame recognition in video, " Pattern Recognition Letters, vol. 23(1- 3), pp. 319-327, 2002. (Pubitemid 33119526)
- [10] B. U. Toreyin, Y. Dedeoglu, and A. E. Petin, "Flame detection in video using hidden markov models, " ICIP '05, vol. 2, no. 2, pp. 1230- 1233, 2005. (Pubitemid 44539168)