A hybrid deep learning model by combining convolutional neural network and recurrent neural network to detect forest fire

Rajib Ghosh[1] (ID) · **Anupam Kumar**[1]

Abstract

Forest fire poses a serious threat to wildlife, environment, and all mankind. This threat has prompted the development of various intelligent and computer vision based systems to detect forest fire. This article proposes a novel hybrid deep learning model to detect forest fire. This model uses a combination of convolutional neural network (CNN) and recurrent neural network (RNN) for feature extraction and two fully connected layers for final classification. The final feature map obtained from the CNN has been flattened and then fed as an input to the RNN. CNN extracts various low level as well as high level features, whereas RNN extracts various dependent and sequential features. The use of both CNN and RNN for feature extraction is proposed in this article for the first time in the literature of forest fire detection. The performance of the proposed system has been evaluated on two publicly available fire datasets—Mivia lab dataset and Kaggle fire dataset. Experimental results demonstrate that the proposed model is able to achieve very high classification accuracy and outperforms the existing state-of-the-art results in this regard.

Keywords Forest fire · Deep learning · Convolutional neural network · Recurrent neural network

1 Introduction

Forest fire can potentially result in a large number of environmental disasters, causing vast economical and ecological losses apart from jeopardising human lives. These fires pose a serious threat to people, wildlife, and the environment. To preserve the natural resources and protect the properties and human lives, forest fire detection has become very crucial. It has lead to increasing number of research explorations in this area around the world. Early and accurate detection of forest fires is essential for mitigating the effect of the fire as once a forest fire spreads to a large area it becomes very difficult to control it and might result in

✉ Rajib Ghosh
  rajib.ghosh@nitp.ac.in

1 Department of CSE, National Institute of Technology Patna, Patna, India

a catastrophe. In its early stage any forest fire is relatively small and easy to control. Fire and smoke detector sensors can easily be installed in indoor environments, it is generally not the case for forest areas. Sensors also require the fire to burn for a while before they can be detected. On the contrary, vision based devices can be used to detect fire in real life and they can be deployed in any area by using different means. These systems are also cheap and easy to install.

Several investigation reports are available on the development of fire detection systems and these reports tried to improve the overall detection accuracy. Most of these studies have used colour and motion features [3, 9, 14]. As forest fires in different regions seem to have different colour and motion characteristics, these methods may fail in certain situations as they have high dependency on a few peculiar characteristics. Spatial and temporal features have also been reported in some of the works [12, 18, 28]. These methods could also fail if there is a lot of smoke accompanying the fire regions. Deep convolutional neural network (CNN) based methods have also been explored, but in a very little number of studies [21, 25, 30, 31] and with limited success. To the best of our knowledge, no study has been found in the literature in this domain employing a combination of CNN, recurrent neural network (RNN) and fully connected layers. Combining CNN and RNN results in a superior model which can use RNN to extract the dependent and sequential features of the input images. This article proposes a hybrid deep learning model for forest fire detection which uses a combination of CNN and RNN networks for feature extraction and two fully connected layers (FCs) for final classification. The novelty of the present investigation lies in proposing this deep learning model employing a combination of CNN, RNN, and fully connected layers.

The major contributions of the present work are:

- Proposing a combination of CNN and RNN networks for feature extraction and then using two fully connected layers for final classification for the first time in the literature.
- Detecting forest fires in images from diverse terrains with very high accuracy.

The remaining sections are organized as follows: Section 2 discusses about related works that has already been done in this field. Description of the dataset used in the present work is given in Section 3. The proposed method is discussed in Section 4. Results obtained from the presented method are discussed in Section 5. Finally, Section 6 concludes the paper with a direction for future research.

2 Literature survey

Compared to general object detection, studies on forest fire detection using computer vision based approaches are very few. Since fires are non-rigid objects with varying structures and sizes, the majority of the studies used spatial and temporal features or motion and colour features. Number of studies using deep learning techniques in this regard is negligible. Besides, most of the studies did not take into consideration the dynamic locations of camera.

2.1 Fire detection using image processing techniques

In the literature, conventional image processing techniques have generally been used to detect forest fires. There have already been some works done in this area using conventional image processing techniques.

Kim et al. [14] presented a colour model based algorithm to detect fire in video frames on wireless sensor network. The algorithm used background subtraction techniques for foreground detection and a Gaussian mixture model for the modelling of background. As fire has various characteristic features, colour information was applied to extract useful information from the gathered video sequences. The authors presented the colour detection method using RGB colour band on the growth area of distinct portions in the consecutive frames. The objects having similar colour to fire, which can be differentiated from background and objects which change their shape in consecutive frame bounding box, were analysed by looking at the temporal variation in each pixel. Celik et al. [3] proposed a real-time fire detection system employing both colour information and background scene. Colour information was determined using the statistical measurement of the images. Gomes et al. [9] presented a method for fire detection using vision-based approach. It used fixed surveillance and smart cameras for capturing images. The method used context-based learning and foreground extraction to improve the performance of the detection system. Toreyin et al. [28] proposed the wavelet domain analysis of videos to detect the forest fires. Liu et al. [18] presented another fire detection system that used the spectral, spatial, and temporal characteristics of fire regions in visual systems. Ho [12] proposed a machine video based flame and smoke detection system applicable in the surveillance system for early warnings. The method combined the spectral, spatial and temporal flame and smoke features to carry out the detection. Borges et al. [2] presented a system that found occurrences of fire in video sequences using colour information. The model used visual characteristic features of fire like area, boundary coarseness, colour, size, surface roughness, and skewness to make better detection of fire occurrences. The system is useful for both video classification as well as surveillance. Video classification feature of the model can be exploited to make real time fire detection at remote locations.

Unmanned aerial vehicle (UAV) is capable of flying by aerodynamic lift and guided without an on-board crew. One of the important applications of UAVs is in the field of surveillance and inspection [24]. Cruz et al. [4] proposed a forest fire detection technique to apply in unmanned aerial systems through new colour index. The technique concentrated on detecting the regions of interest. The regions of interest were divided into two categories— smoke and flame. These regions of interest of various categories were then used for the calculation of colour indices. Then a thresholding technique was used to binarize the result, which finally separated the regions of interest from the rest of the image. In another recent investigation [19], another system to detect forest fire was presented using a rule based computer vision method together with temporal variation. It used background subtraction in order to find the moving objects in the foreground regions of interest. The system used temporal variations to make a distinction between fire and fire-coloured pixels. Khatami et al. [13] presented an image processing technique based fire alarm detection system using particle swarm optimization [10, 27] and K-medoids clustering. The system employed a conversion matrix to describe the colour-space using colour features. The authors developed a colour space using fire flame pixels, linear production of the conversion matrix, and colour features of the sample images. Yuan et al. [29] proposed a method to detect forest fire using both color and motion features with the help of UAV. Initially, the fire-colored pixels have been located as the candidate pixels using a decision rule. The motion features of the candidate pixels have then been calculated using an optical flow algorithm. Foggia et al. [7] presented a fire detection system by combining color, shape, and flame movements based features. This system has detected the fire occurrences by analyzing the videos obtained from surveillance cameras. In a recent study [17], an automated flame detection system in

videos was presented using the Dirichlet process Gaussian mixture color model. The system has proposed a flame detection framework based on the dynamics, color, and flickering features of the flames. In another recent research exploration [26], Sudhakar et al. proposed an UAV based forest fire detection method through color code identification and smoke motion recognition.

2.2 Fire detection using deep learning based techniques

Zhao et al. [31] presented a deep learning based wildfire detection system on images captured by UAV. The system proposed a detection algorithm for fast identification and segmentation of fire regions in the images. It used a 15 layer deep CNN architecture and produced a self-learning fire characteristic extractor classification system. The system significantly reduced the loss of features, which is generally caused when the dimensions of the images are changed. The fire localization algorithm was based on saliency detection and logistic regression classifier. In another study [30] on deep learning based fire detection system, Zhang et al. trained the classifier using a full image after extracting the features through deep CNN. Muhammad et al. [21] presented a deep CNN based fire detection system using video information. The attempt has been made to minimize the extraction time of traditional hand-crafted time-consuming features. A closed-circuit television (CCTV) network based system was designed for indoor as well as outdoor environments. In a recent study [25], Sousa et al. proposed a transfer learning based approach of wildfire detection. In another recent research exploration [22], Park et al. presented a densely connected CNN based framework for wildfire detection. Barmpoutis et al. [1] proposed a fire detection method using deep CNN and exploiting the textures of fire dynamic. Larsen et al. proposed a deep CNN based method for identifying fire smoke in satellite images very recently [16].

Limitations of some of the existing fire detection systems are mentioned in Table 1. Although several investigations have been reported in the literature in this research domain, no study has been found in this domain that has combined the CNN and RNN networks for feature extraction and then fully connected layers for final classification. Due to introducing this hybrid deep learning model in the present study, the proposed system can detect the

Table 1 Limitations of some of the existing fire detection systems

Study	Shortcomings
Kim et al. [14]	Low accuracy (not suitable for real world scenario)
Celik et al. [3]	Too much dependency on background and fire shape
Gomes et al. [9]	Low accuracy (not suitable for real world scenario)
Toreyin et al. [28]	Assumption of stationary camera, computationally inefficient for testing, not suitable for video sequences
Liu et al. [18]	Not suitable for detecting fire in diverse terrain (too much dependency on background)
Ho [12]	High false negative rate, low accuracy
Zhao et al. [31]	Inappropriately small dataset for deep learning
Cruz et al. [4]	Too small dataset
Mahmoud et al. [19]	Low accuracy (not suitable for real world scenario)
Khatami et al. [13]	Low accuracy (not suitable for real world scenario)

forest fires more accurately from the video frames in comparison to other existing studies (Table 5 may be referred for the same).

3 Dataset details and investigation protocol

The present work has used Mivia lab dataset [6] and Kaggle fire dataset [23]. Both of these datasets are publicly available. The details of these two datasets are elaborated below.

3.1 Mivia lab dataset

Mivia lab dataset contains 150 video sequences of 10 minutes each. About half of these videos contain fire or smoke sequences of forest areas and the other half does not contain any fire or smoke sequence. Images have been extracted from these videos at an interval of 4 seconds. The image extraction process resulted in the accumulation of 22500 images. Among these images, 12000 images contain fire or smoke sequences and the remaining 10500 images do not contain any trace of fire or smoke. Few sample images from Mivia lab dataset are shown in Fig. 1.

3.2 Kaggle fire dataset

Kaggle fire dataset contains a total of 1000 images from forest areas. Out of these, 755 are fire images and the remaining 245 are non-fire images. Few sample images from Kaggle fire dataset are shown in Fig. 2.

Both the datasets have been augmented by flipping and zooming the images. All the images from these two datasets have been flipped both horizontally and vertically (horizontal and

Fig. 1 A few sample images from Mivia lab dataset

Fig. 2 A few sample images from Kaggle fire dataset

vertical flip operations). The horizontal flip has reversed the rows and columns of the image pixels horizontally. Similarly, the vertical flip has reversed the rows and columns of the image pixels vertically. For zooming, the random zoom augmentation technique has been used. This technique randomly zooms on a particular area of the image and then a resultant image is produced. After augmenting, three more extra images have been created from each original image present in both the datasets. Both of the augmented datasets have been divided into training and test sets. From Mivia lab dataset, 33600 fire images and 29400 non-fire images have been used for training purpose, whereas 2114 fire images and 686 non-fire images have been used for training purpose from Kaggle fire dataset. The rest of the images from both the augmented datasets have been used for testing purpose. The detailed statistics of two augmented datasets is presented in Table 2.

4 Proposed method

The proposed method consists of various steps. In the first step, all the images from the augmented datasets have been resized to have a fixed dimension, so that they could be fed

Table 2 Details of the two augmented datasets used for training and testing the model

Dataset	Total number of images	Fire images	Non-fire images	Training set size	Testing set size
Mivia lab	90000	48000	42000	63000	27000
Kaggle fire	4000	3020	980	2800	1200

to the proposed model. Resizing operation completes the preprocessing phase. Then the preprocessed images have been fed to the CNN-RNN feature extractor. The output of the feature extractor has been fed as an input to the fully connected layers which has made the final classification. The overall block diagram of the proposed method is shown in Fig. 3.

The proposed method is discussed below in details.

4.1 Preprocessing

The first step involves the preprocessing of the augmented datasets to make it suitable for the proposed model. The images from these two datasets did not have the same size initially. The images from both the datasets have been resized to have a fixed dimension of 128x128x3. The resizing of the images to a fixed dimension is required because the proposed model contains a few fully connected layers at the end and the fully connected layers require a fixed size feature map as input. Resizing the images to a smaller dimension makes the training faster. Resizing operation completes the preprocessing phase.
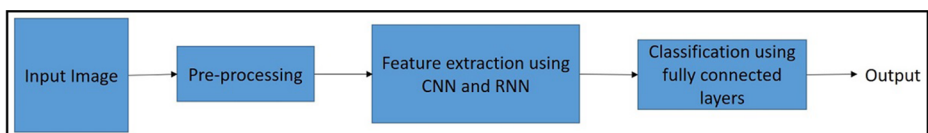
4.2 Feature extraction

After completion of the preprocessing phase, various features of the preprocessed images have been extracted by the CNN-RNN combined feature extractor. At first, the preprocessed input images have been fed to the CNN. The output of the CNN has been fed as an input to the RNN, which further has extracted various sequential and continuous features. The details of feature extraction using CNN and RNN are discussed below.

4.2.1 Feature extraction using CNN

CNN is responsible for extracting the characteristic features of images. The preprocessed input images have been directly fed to the CNN and as such it has the responsibility of extracting various low level as well as high level features of the input images. CNN consists of convolution and pooling layers. Convolution layers extract features using filters and the pooling layers reduce the dimension of those feature maps to make it computationally efficient for further layers. Convolutional layers take the inner product of the linear filter and underlying receptive field, followed by a nonlinear activation function at every local part of the input. This operation can be expressed as

$$y_i^l = f\left(\sum_i^{n-1} W_{pq} * x_i^{l-1} + b_i\right), \tag{1}$$

where y_i^l is the ith output of the lth convolution layer, $f(.)$ is an activation such as the rectified linear unit, W_{pq} is the trainable filters, x_i^{l-1} is the last feature maps or input data, b_i are the biases, and the symbol $*$ is a discrete convolution operator. The resulting outputs are called feature maps. The pooling layers use the maximum (or average) value of the receptive field

Fig. 3 The overall block diagram of the proposed method

at every local part of the feature maps. The use of CNN reduces the computational cost of extracting features. Also, the CNN can learn very sophisticated features from the input image, making classification simple for further layers.

4.2.2 Feature extraction using RNN

The RNN portion of the model is responsible for extracting continuous and sequential features. The final feature map obtained from the CNN has been flattened and then fed as an input sequentially to the LSTM units of RNN after transposing the flattened feature map. RNNs are models that consist of standard recurrent cells, shown in Fig. 4. The typical feature of the RNN cell is a cyclic (or loop) connection, which enables the model to update the current state based on past states and current input data. Formally, the standard recurrent cell is defined as follows:

$$h_j = \phi(W_h h_{j-1} + W_z z_j + b) \tag{2}$$

$$o_j = h_j \tag{3}$$

where $z_j = (x, y, t)_j$ denotes the jth vector of the input signal $\mathbf{z} = (x, y, t)_{j=1,\dots,|z|}$ at timestep j, h_j is the hidden state of the cell, and o_j denotes the cell output, respectively; W_h and W_z are the weight matrices; b is the bias of the neurons; and ϕ is an activation function. Standard recurrent cells have achieved success in many sequence learning problems such as handwriting recognition [8], action recognition [5], or image captioning [20]. However, the standard recurrent cells are not capable of handling long-term dependencies. To solve this issue, the LSTM cells were developed [8, 11]. LSTM cells improve the capacity of the standard recurrent cell by introducing different gates, which are briefly described below.

The LSTM cell is defined as follows:

$$G_{ip} = \sigma(W_{ud}[h_{j-1}, z_j] + b_{ip}) \tag{4}$$

$$G_{fg} = \sigma(W_{fg}[h_{j-1}, z_j] + b_{fg}) \tag{5}$$

$$G_{op} = \sigma(W_{op}[h_{j-1}, z_j] + b_{op}) \tag{6}$$

$$c_j = G_{ud} \circ \tilde{c}_j + G_{fg} \circ c_{j-1} \tag{7}$$

$$\tilde{c}_j = \phi(W_c[h_{j-1}, z_j] + b_c) \tag{8}$$

$$h_j = G_{op} \circ \phi(c_j) \tag{9}$$

where c_j is an additional hidden state, W_* are weight matrices, b_* are biases, G_* denote cell gates (ip: input, fg: forget, op: output, ud: update), and ϕ and σ are activation functions (hyperbolic tangent and sigmoid, respectively). The operator $\circ$ denotes the Hadamard (element-wise) product. Fig. 5 shows the organization of one LSTM cell.

It may be noted that the LSTM has two kinds of hidden states: a "slow" state c_j that keeps long-term memory, and a "fast" state h_j that makes decisions over short periods of time. The forget gate decides which information will be kept in the cell state and which information will be thrown away from the cell state. Apart from hyperbolic tangent and sigmoid activation functions, there are various other non-linear activation functions have been promoted in the research literature. In the present work, as RNN receives the input from
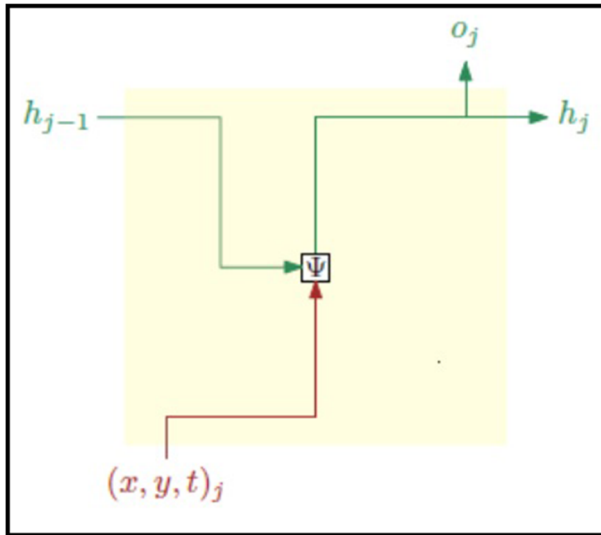
Fig. 4 A simple RNN cell

the CNN, so it is already receiving complex features extracted by the CNN. RNN extracts significant sequential information and then feeds this output to the fully connected layers.

4.3 Image classification

The fully connected layers of the model are responsible for making final classification. The output of the RNN has been fed to a dense layer of 1024 nodes, which itself is connected to another dense layer of 256 nodes. The second dense layer is connected to a single node in the output layer which makes the final classification of the images. The detailed architecture of the proposed model is shown in Fig. 6.
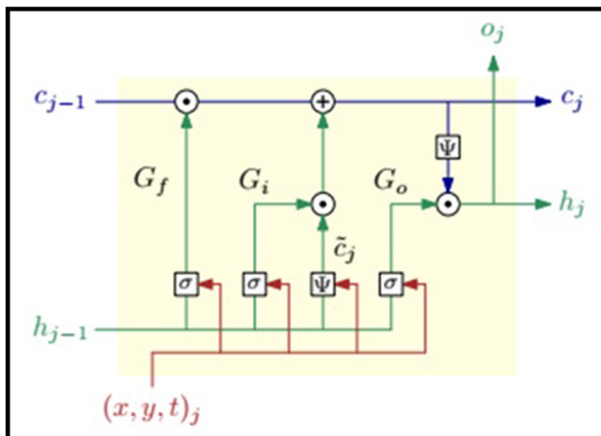
Fig. 5 A LSTM cell
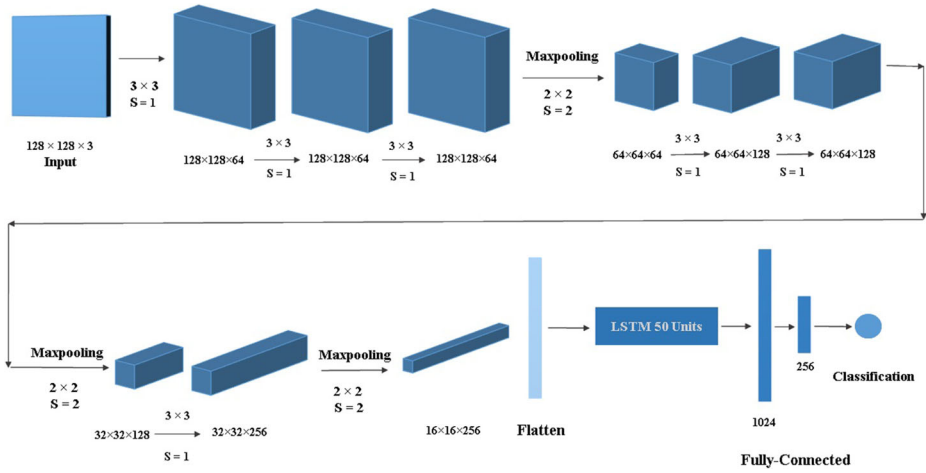
Fig. 6 Detailed architecture of the proposed CNN-RNN-FCs hybrid model

5 Experimental results and analysis

For the experimentation, both the datasets have been splitted into training set and test set. Two classes have been considered for classification: positive (fire) and negative (no fire). In both the training and test sets, all the images have been labelled as either positive class or negative class. The proposed model has been optimised using Adam optimization algorithm [15] and binary cross entropy loss function with a learning rate of 0.001.

5.1 Parameters of CNN

In the present system, CNN architecture consists of a total of six convolution layers (conv1, conv2, conv3, conv4, conv5, and conv6). The first three convolution layers (conv1, conv2, and conv3) have been applied consecutively on the processed input image. Maxpooling operation (pool1) has been applied after these three convolutions. Two more consecutive convolution layers (conv4 and conv5) have been used after this maxpooling. Another maxpooling operation (pool2) has been applied after conv5. Finally, another convolution layer (conv6) has been used, followed by the final maxpooling operation (pool3). Two-dimensional (2D) convolution filters (also called conv2D filters) of dimension 3x3 have been used for every convolution operations. These 3x3 dimensional 2D filters have been applied across all the channels of the feature map. Padding has also been applied during each convolution operation to maintain the height and width of the feature map. The padding of one pixel has been applied around the feature map of previous layer. For example, the input image is of dimension 128x128x3, the image is padded with one pixel along all the three channels. The number of 3x3 dimensional filters used at a particular layer depends on the dimension (particularly the depth) of the feature map required at the next layer. For the first three convolution layers (conv1, conv2, and conv3), 64 filters have been used. For the next two convolution layers (conv4 and conv5), 128 filters have been used, and for the final convolution layer (conv6), 256 filters have been used. Pooling operation has been executed on

Table 3 Accuracy of the proposed system on Mivia lab test set

Epoch	Batch Size	Block Size	Accuracy(%)
100	32	50	97.22
		75	97.41
		100	97.85
		50	96.92
	64	75	96.96
		100	96.60
150	32	50	99.16
		75	98.82
		100	98.75
	64	50	99.62
		75	99.53
		100	99.53
200	32	50	98.92
		75	99.13
		100	98.85
	64	50	99.42
		75	99.09
		100	98.78

2x2 subareas of feature map, with stride of 2. The processed images that have been fed to the CNN have a size of 128x128x3. The final feature map obtained after the last maxpooling (pool3) operation has a dimension of 16x16x256.

Table 4 Accuracy of the proposed system on Kaggle fire test set

| Epoch	Batch Size	Block Size	Accuracy(%)
100	32	50	97.60
		75	97.30
		100	97.20
		50	96.90
	64	75	97.10
		100	96.80
150	32	50	98.30
		75	98.40
		100	98.15
	64	50	98.45
		75	98.35
		100	98.20
200	32	50	99.10
		75	98.80
		100	98.90
	64	50	98.80
		75	98.70
		100	98.60
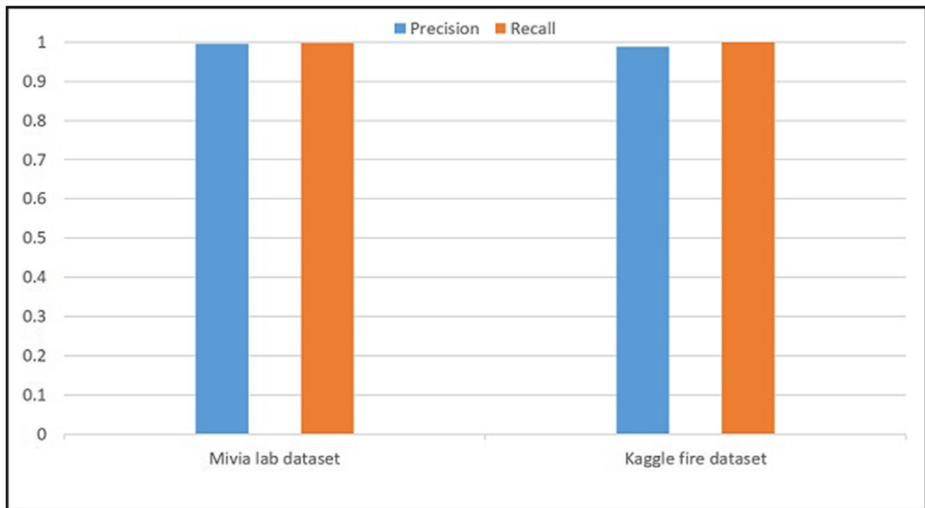
Fig. 7 Precision and recall of the proposed classification system on Mivia lab and Kaggle fire datasets

5.2 RNN parameters

In the proposed method, the RNN is comprised of long short-term memory (LSTM) units stacked together. The output feature map of the CNN has been flattened and then fed to the RNN. There are 50 recurrently connected memory blocks in the hidden layer of LSTM. The performance of the model has been evaluated with various combinations of memory blocks, but the instance using 50 blocks has produced the best results (Tables 3 and 4 may be referred). Gates have been activated using the Sigmoid function.

5.3 Quantitative performance on Mivia lab dataset

Various combinations of batch size, block size, and epoch have been used for training and testing the model on Mivia lab dataset. Out of all the combinations, the best results have
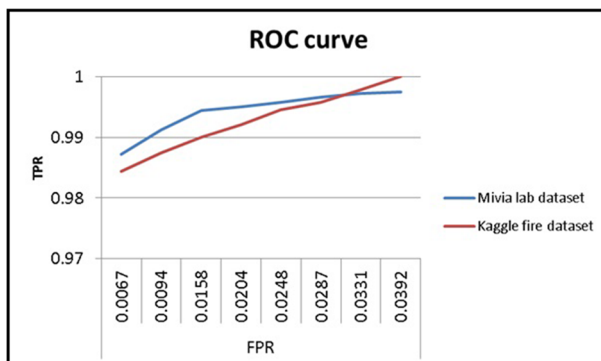
Fig. 8 The ROC curve analysis of the proposed classification system on Mivia lab and Kaggle fire datasets

been achieved at 150 epoch with batch size of 64 and block size of 50. The precision and recall values are 0.9954 and 0.9975 respectively. The detailed classification accuracies on different combinations are shown in Table 3. The precision-recall graph for the best combination of epoch, batch size and block size (150, 64, 50) on Mivia lab dataset is shown in Fig. 7. The Receiver Operating Characteristic (ROC) curve analysis of the proposed classification system on Mivia lab dataset is shown in Fig. 8.

Image sample	Original class	Obtained class
	Fire	✓ Fire
	Fire	✓ Fire
	Non-fire	✓ Non-fire
	Non-fire	✓ Non-fire

Fig. 9 Correct classification of few test samples from Mivia lab dataset

5.4 Quantitative performance on Kaggle fire dataset

In the experimentation on Kaggle fire dataset also, various combinations of batch size, block size and epoch have been used to train and test the model. Out of all these combinations, the best results have been achieved at 200 epoch with batch size of 32 and block size of 50. The precision and recall values are 0.9882 and 1.0000 respectively. The detailed classification accuracies on different combinations are shown in Table 4. The precision-recall graph for the best combination of epoch, batch size and block size (200, 32, 50) on Kaggle fire dataset is shown in Fig. 7. The ROC curve analysis of the proposed classification system on Kaggle fire dataset is shown in Fig. 8.
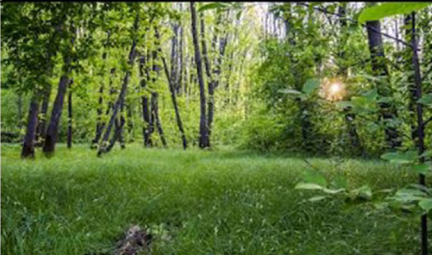
Image sample	Original class	Obtained class
	Fire	✓ Fire
	Fire	✓ Fire
	Non-fire	✓ Non-fire
	Non-fire	✓ Non-fire

Fig. 10 Correct classification of few test samples from Kaggle fire dataset

5.5 Qualitative performance on two datasets

Figure 9 shows correct classification of few test samples from **Mivia lab dataset**, whereas Fig. 10 shows correct classification of few test samples from Kaggle fire dataset. Figure 11 shows incorrect classification of few test samples from two datasets.

Fig. 11 Incorrect classification of few test samples from two datasets

5.6 Comparison with state-of-the-art results

Table 5 lists the performance of some existing significant fire detection systems available in the literature and the proposed one. The performance of the existing systems have been evaluated on the same datasets (Mivia lab dataset and Kaggle fire dataset) as used in the present work. The rightmost column of Table 5 presents the processing speed of the existing systems and the proposed one in frames per second (fps).

Table 5 Performance comparison of the proposed system with some existing significant fire detection systems

Study	Method	Dataset	Accuracy(%)	Processing speed (in fps)
Kim et al. [14]	Background subtraction	Mivia lab	96.69	16
		Kaggle fire	96.50	16
Celik et al. [3]	Background subtraction	Mivia lab	98.89	16
		Kaggle fire	98.56	16
Gomes et al. [9]	Background subtraction	Mivia lab	93.10	16
		Kaggle fire	93.20	16
Toreyin et al. [28]	Spatial and temporal analysis	Mivia lab	99.20	18
		Kaggle fire	98.80	18
Liu et al. [18]	Spectral, spatial and temporal analysis	Mivia lab	93.42	18
		Kaggle fire	93.80	18
Ho [12]	Spectral, spatial and temporal analysis	Mivia lab	82.38	18
		Kaggle fire	82.15	18
Zhao et al. [31]	Deep learning	Mivia lab	98.00	16
		Kaggle fire	97.72	16
Muhammad et al. [21]	Deep learning	Mivia lab	96.38	16
		Kaggle fire	96.04	16
Sousa et al. [25]	Deep learning	Mivia lab	94.56	18
		Kaggle fire	94.24	18
Park et al. [22]	Deep learning	Mivia lab	98.26	18
		Kaggle fire	97.88	18
Barmpoutis et al. [1]	Deep learning	Mivia lab	95.67	18
		Kaggle fire	95.44	18
Borges et al. [2]	Probabilistic colour model	Mivia lab	99.32	18
		Kaggle fire	98.90	18
Cruz et al. [4]	New colour index	Mivia lab	97.67	18
		Kaggle fire	97.10	18
Mahmoud et al. [19]	Background subtraction	Mivia lab	92.80	16
		Kaggle fire	93.15	16
Khatami et al. [13]	New colour space	Mivia lab	93.40	18
		Kaggle fire	93.15	18
Proposed Method	CNN-RNN combination	Mivia lab	99.62	18
		Kaggle fire	99.10	18

5.7 Strengths of the proposed system

The major strengths of the proposed model are listed below:

- Combining CNN and RNN creates a superior model and as such the proposed model is capable of detecting forest fires in images from different geographical terrains. Forest fires in different regions have different colour and motion characteristics. The existing methods, relying on colour and motion features, may fail in certain situations as they have high dependency on a few peculiar colour and motion characteristics. Similarly, the existing methods, employing spatial and temporal features, could also fail if there is a lot of smoke accompanying the fire regions. These drawbacks of the existing systems are overcome by the proposed system due to combining CNN and RNN networks together for feature extraction.
- The performance of the model has been tested on two large public datasets.
- The high accuracy of the model makes it suitable for deployment in real forest fire detection applications.

6 Conclusion and future work

This article proposes a combination of CNN and RNN based deep learning method for forest fire detection. The evaluation of the performance of the present system has been done on two different public datasets. The proposed forest fire detection system outperforms the existing studies in this regard. The present work also overcomes various drawbacks of the existing systems. It is evident from the high classification accuracy of the present system that the present system can be employed to detect forest fires in the real world scenarios. The present work shall provide fresh insight to the researchers in carrying out the new researches on fire detection using computer vision based techniques.

In future, the attempt will be made to carry out the research work in this problem area by employing other sophisticated deep learning techniques. The plan is also there to develop a fire detection system in non-forest areas, especially fires that occur in residential areas and industrial areas. Other possible future directions of this research work include the exploration of the possibility of employing the proposed model for low resolution satellite images covering large geographical areas.

References

1. Barmpoutis P, Stathaki T, Dimitropoulos K, Grammalidis N (2020) Early fire detection based on aerial 360-Degree sensors, deep convolution neural networks and exploitation of fire dynamic textures. Remote Sens 12:3177
2. Borges PVK, Izquierdo E (2010) A probabilistic approach for vision-based fire detection in videos. IEEE Trans Circ Syst Video Technol 20(5):721–731
3. Celik T, Demirel H, Ozkaramanli H, Uyguroglu M (2007) Fire detection using statistical color model in video sequences. J Vis Commun Image Represent 18(2):176–185
4. Cruz H, Eckert M, Meneses J, Martínez JF (2016) Efficient forest fire detection index for application in unmanned aerial systems (UASs). Sensors 16(6):893–909
5. Du W, Wang Y, Qiao Y (2018) Recurrent spatial-temporal attention network for action recognition in videos. IEEE Trans Image Process 27:1347–1360
6. Foggia P, Saggese A, Vento M Mivia lab dataset, 2015, Retrieved September 2020 from http://mivia. unisa.it/datasets

7. Foggia P, Saggese A, Vento M (2015) Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion. IEEE Trans Circ Syst Video Technol 25(9):1545–1556

8. Ghosh R, Vamshi C, Kumar P (2019) RNN Based online handwritten word recognition in Devanagari and Bengali scripts using horizontal zoning. Pattern Recogn 92:203–218

9. Gomes P, Santana P, Barata J (2014) A vision-based approach to fire detection. Int J Adv Robot Syst 11(9):1–12

10. Gong YJ, Li JJ, Zhou Y, Li Y, Chung HSH, Shi YH, Zhang J (2015) Genetic learning particle swarm optimization. IEEE Trans Cybern 46(10):2277–2290

11. Graves A, Liwicki M, Fernandez S, Bertolami R, Bunke H, Schmidhuber J (2009) A novel connectionist system for unconstrained handwriting recognition. IEEE Trans Pattern Anal Mach Intell 31(5):855–868

12. Ho CC (2009) Machine vision-based real-time early flame and smoke detection. Meas Sci Technol 20(4):1–13

13. Khatami A, Mirghasemi S, Khosravi A, Lim CP, Nahavandi S (2017) A new PSO-based approach to fire flame detection using K-Medoids clustering. Expert Syst Appl 68:69–80

14. Kim YH, Kim A, Jeong HY (2014) RGB Color model based the fire detection algorithm in video sequences on wireless sensor network. Int J Distrib Sens Netw 10(4):1–10

15. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations, San Diego, pp 1–15

16. Larsen A, Hanigan I, Reich BJ, Qin Y, Cope M, Morgan G, Rappold AG (2021) A deep learning approach to identify smoke plumes in satellite imagery in near-real time for health risk communication. J Expos Sci Environ Epidemiol 31:170–176

17. Li Z, Mihaylova LS, Isupova O, Rossi L (2018) Autonomous flame detection in videos with a Dirichlet process Gaussian mixture color model. IEEE Trans Ind Inf 14(3):1146–1154

18. Liu CB, Ahuja N (2004) Vision based fire detection. In: Proceedings of the 17th International Conference on Pattern Recognition. Cambridge, pp 134–137

19. Mahmoud MA, Ren H (2018) Forest fire detection using a rule-based image processing algorithm and temporal variation. Math Probl Eng 2018:1–8

20. Mao J, Xu W et al (2015) Deep captioning with multimodal recurrent neural networks (m-RNN). In: Proceedings of ICLR

21. Muhammad K, Ahmad J, Lv Z, Bellavista P, Yang P, Baik SW (2019) Efficient deep CNN Based fire detection and localization in video surveillance applications. IEEE Trans Syst Man Cybern Syst 49(7):1419–1434

22. Park M, Tran DQ, Jung D, Park S (2020) Wildfire-detection Method Using DenseNet and cycleGAN Data Augmentation-Based Remote Camera Imagery. Remote Sens 12:3715

23. Saied A (2018) FIRE Dataset. Retrieved October 2020 from https://www.kaggle.com/phylake1337/fire-dataset

24. Saripalli S, Montgomery JF, Sukhatme GS (2003) Visually guided landing of an unmanned aerial vehicle. IEEE Trans Robot Autom 19(3):371–380

25. Sousa MJ, Moutinho A, Almeida M (2975) Wildfire detection using transfer learning on augmented datasets. Expert Syst Appl 2020(11):142

26. Sudhakar S, Vijayakumar V, Kumar CS, Priya V, Ravi L, Subramaniyaswamy V (2020) Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires. Comput Commun 149:1–16

27. Sun Z, Liu Y, Tao L (2018) Attack localization task allocation in wireless sensor networks based on multi-objective binary particle swarm optimization. J Netw Comput Appl 112:29–40

28. Töreyin BU, Dedeoğlu Y, Güdükbay U, Cetin AE (2006) Computer vision based method for real-time fire and flame detection. Pattern Recogn Lett 27(1):49–58

29. Yuan C, Liu Z, Zhang Y (2016) Vision-based forest fire detection in aerial images for firefighting using UAVs. In: Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, pp 1200–1205

30. Zhang Q, Xu J, Xu L, Guo H (2016) Deep convolutional neural networks for forest fire detection. In: Proceedings of the International Forum on Management, Education and Information Technology Application, Guangzhou, pp 568–575

31. Zhao Y, Ma J, Li X, Zhang J (2018) Saliency detection and deep learning-based wildfire identification in UAV imagery. Sensors 18(3):712–731