Artificial Intelligence Review (2024) 57:207
https://doi.org/10.1007/s10462-024-10848-6

Fire identification based on novel dense generative adversarial networks

Tawfeeq Shawly[1] · Ahmed A. Alsheikhy[2]

Accepted: 25 June 2024 / Published online: 16 July 2024

Abstract
Increasing death rates, damage to properties, and loss of trees can be caused by fires. In Australia and the United States of America, many fire incidents are reported annually. Due to that, both governments struggle from the devastation beyond plants, buildings, and infrastructure. A lot of people have lost their properties and land. Various innovations in fire detection technologies have been implemented to minimize the impacts of fires on the economy and lives. Some of these solutions are costly, while others lack accuracy. In this article, a novel deep-learning model to detect fires is presented. This model is based on new Novel Dense Generative Adversarial Networks (NDGANs) and image preprocessing technologies for fire detection through a continuous monitoring system. This system produces alarms if a fire or smoke is detected. The proposed approach was trained and tested on five datasets. This system was evaluated using four performance quantities, which are accuracy, sensitivity, dice, and F-score, and attained 98.87%, 97.64%, 98.82%, and 98.69% for the considered quantities, respectively. In addition, the proposed method was compared with other developed approaches and outperformed these methods. The presented New Dense Generative Adversarial Networks technology is useful in fire detection as shown from the conducted simulation experiments on MATLAB.

Keywords Fire · Deep learning · NDGANs · Smoke · Flames · Artificial intelligence

1 Introduction

Forests are the homeland of numerous animals, birds, and plants. These forests are threatened by nature or humans (Ghali et al. 2018, 2022). Wildfires can start at any time by droughts, heat, lightning strikes, or unintentionally by mankind (Ghali et al. 2018,

✉ Ahmed A. Alsheikhy
aalsheikhy@nbu.edu.sa

Tawfeeq Shawly
tshawly@kau.edu.sa

1 Department of Electrical Engineering, Faculty of Engineering at Rabigh, King Abdulaziz University, 21589 Jeddah, Saudi Arabia

2 Department of Electrical Engineering, College of Engineering, Northern Border University, 91431 Arar, Saudi Arabia

2021, 2022; Shahid and Hua 2021). These wildfires can cause death to animals, birds, plants, and severe damage to properties and infrastructure. These fires destroy everything that comes in their way. Fires destroy around 350 million to 450 million hectares annually (Ghali et al. 2021). Consequently, many people lost their properties and lands. In 2021, numerous countries, such as European countries, two American continentals, Australia, Central Africa, and the Arabian Gulf suffered from fires, especially in the forests (Ghali et al. 2021, 2022). These fires increased nearly from 400 to 600% in the last decade in the United States only (Ghali et al. 2022). Statistics reported that about 8000 wildfires occur yearly in Canada and destroy 2.5 million hectares per year (Ghali et al. 2022). So, detecting fires early is extremely important to save lives, animals, and valuable resources (Singh 2021; Zhang et al. 2020; Niknejad and Bernardino 2111; Abdusalomov et al. 2021; Dampage et al. 2022). Various benefits can be obtained from forests, such as climate regulation and economic evolvement (Dampage et al. 2022). Fire remains a major concern even with technological advancement (Ghali et al. 2018).

Various solutions have been implemented to detect fire. These solutions can be categorized into two main methods, which are (1) traditional solutions and (2) visual-based solutions. The traditional solutions rely on sensors, smoke, temperature, and microelectronics (Zhang et al. 2020; Niknejad and Bernardino 2111; Regi et al. 2018; Zhao et al. 2022; Chopde et al. 2022). Numerous fire components, such as heat, smoke, and flames can be detected by sensors (Chopde et al. 2022; Hong and Choi 2012). These approaches work perfectly in a close range and a certain space (Zhang et al. 2020). These components require an intensive process to differentiate between threatening and non-threatening situations (Hong and Choi 2012). The number of damaged properties from fire in the United States dropped recently according to the National Fire Protection Association (NFPA) (Chopde et al. 2022; Hong and Choi 2012). These significant drops occurred due to the implementation of sensors and detectors. However, these methods suffer from different limitations, such as their delayed response time in real-time scenarios and limited coverage areas (Ghali et al. 2022).

Recently, the advancement of Computer Vision (CV) technologies has provided additional support to detect wildfires using visual-based methods, which collect data from cameras that capture images and videos according to the implemented approaches. These methods have been developed based on deep-learning technologies (DLTs), which have gained noticeable attention due to their accuracy, large coverage areas, and small probability errors (Ghali et al. 2022; Manoj et al. 2022). Several algorithms for fire detection using Artificial Intelligence (AI) have been implemented using innovative techniques. These methods include monitoring systems, heat and smoke sensors, and fire detection using videos (Manoj et al. 2022; Khan et al. 2022; Yandouzi et al. 2022; Dai et al. 2022; Gharge et al. 2014). Unfortunately, most of these methods provide a shallow analysis (Chopde et al. 2022; Hong and Choi 2012). Various approaches based on Generative Adversarial Networks (GANs) have been applied in the computer vision field in many areas, such as medical images to detect diseases or in the commercial field. Generative Adversarial Networks (GANs) are a type of deep-learning technology and contain two components, which are a Generator and Discriminator (Zhao 2022; Goodfellow et al. 2014; Wang 2017). These two components work simultaneously through training (Wang 2017; Zhai et al. 1611). The generator produces false data, and the discriminator distinguishes the right information from the false data (Dai et al. 1702; Kumar et al. 1901; Morizet 2020). This study deploys these GANs topologies to identify fires.

2 Research problem

Concerned researchers and teams have developed various methods to perform required tasks, such as detection and classification. Some countries like Australia, the United States of America, and Argentina suffer from fires annually. These fires destroy acres of land, and homes, and cause death to humans and animals. In the last years, the reported state-of-the-art methods in the literature used inputs from sensors or cameras to detect wildfires and localize these fires despite their shapes or sizes. Some of the recently developed approaches to detect fires, such as in (Ghali et al. 2021, 2022; Shahid and Hua 2021; Singh 2021; Zhang et al. 2020; Niknejad and Bernardino 2111; Abdusalomov et al. 2021) provided an average accuracy between 85.12% and 98.23%. However, accuracy was affected negatively if the resolution was low or the implemented methods were unable to differentiate between fires, flames, smoke, or similar colors, such as in lamps as reported in Abdusalomov et al. (2021). Thus, implementing a new algorithm with higher accuracy is required.

To address these problems, this paper presents a novel deep-learning architecture based on GANs for fire detection at anytime, anywhere, and under any conditions/circumstances. Moreover, alarms are generated when a fire, flame, or smoke is discovered. These alarms notify residents or concerned teams to take proper actions to save lives.

3 Research motivations and contributions

The summary of motivations that stimulated this study is:

- To align with the Saudi Vision 2030.
- To save lives, properties, and infrastructure.
- To detect fires or any component in a quick manner.
- To improve accuracy and other performance quantities.

This research intends to develop a reliable and faster algorithm to detect fires using a deep-learning methodology that incorporates typical elements of GANs into one platform to perform the required operations. This platform is seen as a brain since it controls the system. The significant contributions include:

(a) Design and implement new Novel Dense GANs (NDGANs) model to detect fire and its elements.
(b) Integrate the model with various image preprocessing technologies.
(c) The presented algorithm is faster than the existing GANs model.
(d) The utilized parameters inside the presented method are less than what the developed ones include.
(e) The conducted analysis on five datasets shows that the performance quantities produced better results regarding all considered metrics.

This paper is presented as follows: the literature review is provided in SubSect. 1.3, and the complete description of the proposed system is in Sect. 2. Section 3 offers details of the system evaluation, and Sect. 4 contains a discussion. Section 5 concludes the article.

3.1 Related work

Several models have been implemented to detect fires based on Artificial Intelligence (AI), Machine Learning (ML), and deep-learning technologies.

Ghali et al. (2022) implemented a new deep-learning algorithm based on ensembling EfficientNet-B5 and DenseNet-201 approaches to detect wildfires. The authors utilized RBG aerial images in their model. TransUNet and TransFire transformers were applied for segmentation purposes. This method extracted two feature maps and propagated these maps into an average pooling layer after the concatenation process. In addition, the authors applied a dropout of 0.2 to prevent the overfitting issue. The authors used four methods to segment the images, which were a Convolutional Neural Network (CNN), EfficientSeg, and the two utilized transformers. The utilized CNN architecture consisted of ResNet-50, a pretrained Vision Transformer (ViT), Multi-Layer Perceptron (MLP), and Multihead Self-Attention (MSA) and worked as an encoder and a decoder. The decoder contained cascade up-sampler (CUP) blocks to decode the feature maps and perform a binary segmentation process. In addition, the ReLU activation function was used as well. The size of every convolutional layer was 3×3. Both CNN and the pretrained ViT were applied to get the feature maps. Lastly, the Sigmoid activation function was deployed to categorize images into their appropriate classes, which were fire or non-fire. The authors utilized a public dataset called FLAME, which stands for Fire Luminosity Airborne-based Machine Learning Evaluation for training and testing purposes. This dataset included aerial images and raw heat map footage that was captured by a drone using a visible spectrum and thermal cameras on board. In total, the authors used nearly 48,000 RBG aerial images, and these images were divided into two groups: fire and non-fire. Each input image was of size 512×512 pixels and the learning rates were 0.001 and 0.1 with a patch size of 16. The developed model achieved 85.12% accuracy and 84.77% F-score.

Ghali et al. (2021) developed a model to identify and segment forest fires using two Vision Transformers (ViTs), which were TransUNet and Medical Transformer (MedT) to predict the spread of the fires and provide support and aid to firefighters. Thus, the authors implemented two frameworks for this purpose. These frameworks extracted global and local features to achieve nearly 96.76% for the F-score. TransUNet included a hybrid CNN and a U-Net network. This transformer had twelve layers and various convolutional layers with a size of 3×3 for each. It worked as an encoder, and it was the first component to extract features. The positional information was encoded using a patch embedding operator. Both transformers applied a sequence-to-sequence strategy to overcome the issue of intrinsic locality in the convolutional layers. The authors applied the model to a dataset called ImageNet. The MedT element was designed to apply a Local–Global (LOGO) learning strategy and gated axial layer. The LOGO method had two branches: a global branch that used the original resolution of inputs and a local branch that deployed the patches of the inputs. This element contained six convolutional layers, a batch normalization layer, and ReLU as the activation function. The authors used nearly the same model in Ghali et al. (2022). The utilized dataset consisted of 1135 images and their corresponding masks, and this dataset was trained and tested on a machine that included NVIDIA GeForce RTX 2080Ti GPU and Pytorch. In addition, two augmentation techniques were applied. EfficientSeg and U^2-Net were used to perform a comparative evaluation against the two transformers. Readers are encouraged to read (Ghali et al. 2021) for additional information.

Shahid and Hua (2021) performed a profound analysis to prove that ViTs were valuable tools in fire detection. The utilized model contained an embedding encoding layer and a classifier layer. The embedding encoding layer had a multi-head self-attention, multi-perceptron, a normalization layer, and residual skip connections. In addition, two patch sizes were deployed, which were 16 and 32 on both transformers. Two datasets: BoWFire and Foggia were utilized. Unfortunately, the authors provided no information about the number of images that were used. The authors reached 94.04% accuracy and 98.08% for the F-score as maximum outputs on both datasets.

Zhang et al. (2020) developed a model based on a YOLO network with an attention mechanism to detect fire. The authors used YOLOv3, and the attention mechanism was added to the last three convolutional layers in a series mode. These convolutional layers had different scales. The authors updated the map features by summing all weights in all utilized channels. This procedure was able to capture the semantic dependencies between layers and enhanced the model generalization capability. The authors utilized RGB colors to extract features and detect fires. A flame dataset was constructed to obtain sample labels quickly using an RBG-IR camera. This camera collected multi-scale scenes with paired RBG-T samples in a total of 5 videos. The infrared images were applied to accurately annotate labels automatically and to segment the flame masks in the algorithm. The authors applied a mutual information technique to compute the transformation relation between an infrared image and its corresponding RGB one. To realize the annotation automatically. The authors cropped the infrared images and generated pixel-aligned pairs of RGB-T samples to process these samples further. Since the masks were the same in the infrared images and the RGB ones, it was easy to obtain a minimum bounding rectangular box. 92% accuracy was obtained by the model using two test sets with 600 images in total.

Niknejad and Bernardino (2021) developed a joint Convolutional Neural Network to detect and classify fire in images. A spatial attention mechanism was deployed to obtain dependencies between pixels, and the probability was computed using a new channel attention module. These two attention modules captured global scores and correlations in spatial locations of fires. The channel attention module was used to multiply the resultant attention weights to the output channel. These weights were the probabilities assigned by the applied network. The required characteristics were obtained from an encoder and a decoder using the sigmoid activation function. The encoder was created from Deeplab v3+. A dataset was constructed from RGB images and their associated segmentation masks from two sources. The authors obtained an average accuracy of 98.46% after comparing their method with the typical UNET network.

The authors in Abdusalomov et al. (2021) implemented a method to detect fires in real-time with high speed using a deep-learning topology. A special convolutional neural network was created based on the YOLOv3 algorithm. Real-time detector cameras were built on a Banana Pi M3 board. The implemented approach was tested in daylight and night and achieved reasonable outcomes regarding the shape and size of fires. A dataset of 9200 images was created using various deep-learning technologies from public resources including Google images. These images were captured in daylight and night. Numerous augmentation tools, such as rotating all images by 90, 180, and 270 degrees were used to increase accuracy. In addition, every image was rescaled to 608×608 pixels. The authors applied 50,000 iterations and reached 98.3% accuracy.

Table 1 displays a summary of some of the implemented works in the literature regarding the technologies that were used, outputs, advantages, and limitations.

From Table 1, we can conclude that the proposed model: new Novel Dense Generative Adversarial Networks (NDGANs) outperforms all mentioned approaches in this table regarding the evaluated performance indicators.

Table 1 The comparative study between some of the developed methods

References	Utilized technology	Attained results	Advantages	Limitations
Ghali et al. (2022), 2022	Deep learning and transformers	85.12% accuracy	Easiness was achieved by using U-Net technology	Unsatisfactory was obtained since the accuracy was less than 90% This model is composed of too many processes and requires much time to execute
Ghali et al. (2021), 2021	Deep vision transformers: TransUNet and MedT	96.76% F-score	TransUNet and MedT segmented fire pixels using a fusion color space approach	Unable to process video streams and no information about accuracy was provided
Shahid and Hua (2021), 2021	Vision transformers	94.04% accuracy and 98.08% F-score	The authors used two transformers: Base and Large to detect fires using spatio-temporal context	Video streams are not supported
Zhang et al. (2020), 2020	YOLOv3 network with an attention mechanism	92% accuracy	Updated the map features by summing all weights. This procedure was able to capture the semantic dependencies between layers	Only 600 images were used
Abdusalomov et al. (2021), 2021	YOLOv3	98.3% accuracy	False positive rates were minimized	Electrical lamps are considered real fires Unable to process images in extreme weather conditions

4 Materials and methods

This section provides a complete explanation of the proposed approach.

4.1 Typical generative adversarial networks (GANs)

GANs were developed in 2014 and are unsupervised deep-learning models that can distinguish between real and false data. A typical GANs architecture consists of (1) a generator to create false data and (2) a discriminator to learn the real data. The generator creates false or fake data and feeds these data into the discriminator, which intensively learns to distinguish between real and false data. Later, the outputs from the discriminator are propagated back into the discriminator itself and the generator for updating purposes. Figure 1 illustrates a typical GANs structure.

4.2 Datasets

In this research, five datasets from different sources are used for training, validation, and testing purposes. The first dataset from Thakur and Satish (2021) includes 999 images of fires and non-fires. The size of this dataset is 394 MB. 755 images contain fires while the rest contains no fires. The second dataset downloaded from https://www.kaggle.com/datasets/phylake1337/fire-dataset is 3.23 MB and contains 1171 images. All images include fires or flames. The third dataset from https://images.cv/download/flames/1219/CALL_FROM_SEARCH/%22flames%22 is 2.82 GB and includes 21,527 images. The fourth one from https://github.com/gaiasd/DFireDataset contains 12,062 images with a total size of 3.24 GB. The last dataset from https://ieee-dataport.org/open-access/large-scale-dataset-active-fire-detectionsegmentation-landsat-8 has 100 images and its size is 84.4 MB. In total, these datasets have 35,859 images and are categorized into three groups: training, validation, and testing. The first group represents 70%, the second group is 10% and the last group, which is testing, is assigned 20%. Table 2 provides a detail of the utilized images in this study. Figure 2 depicts the distribution of the utilized images in the training set.

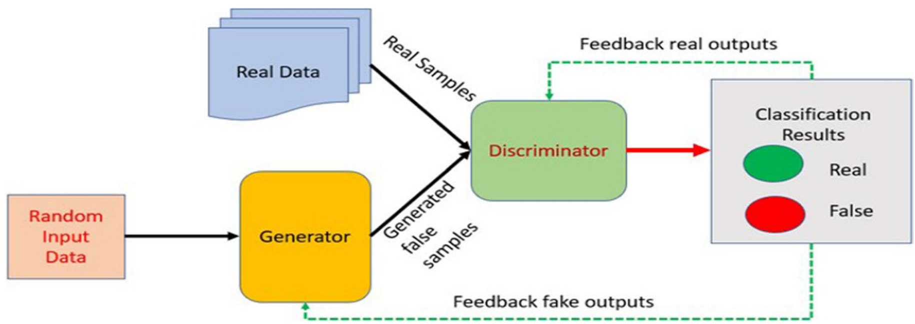

Fig. 1 The typical architecture of GANs

Table 2 A summary of the utilized images

Number of images					
Training		Validation		Testing	
Fire	Non-fire	Fire	Non-fire	Fire	Non-fire
13,784	11,317	1827	1759	5491	1681

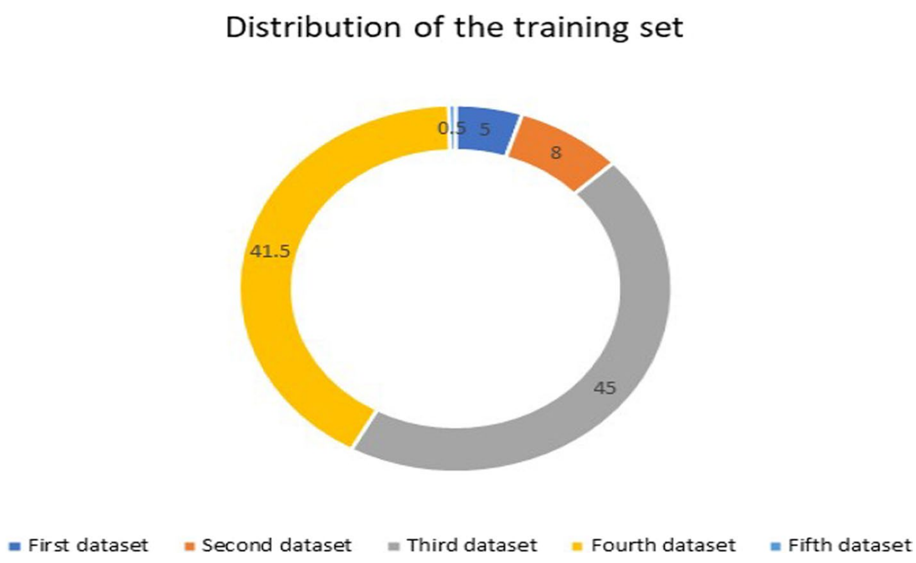

Fig. 2 The visual distribution of the training set

4.3 The proposed methodology

In this subsection, complete details of the proposed system to detect fires and their components efficiently and accurately despite their shapes and locations are provided. In this study, the NDGANs model is presented. The dense layers are used to establish a diverse multi-scale feature map. This map offers various benefits, such as enhancing characteristics propagation from one level into another, reuse of features, and minimizing the number of parameters. Initially, the noise is removed or reduced using a set of filters. This set includes Bilateral and Gabor filters. These two filters work based on pixels. A mathematical expression of the noise removal or reduction is shown in Eq. (1).

$$Y_1 = \frac{\sum(IsxIr)}{\Upsilon} \tag{1}$$

Y_1 represents the output, Is, and Ir denote the image space and range, respectively. Υ is a normalization ratio.

The resultant images are different in size, intensity, and contrast, which could make it difficult to learn. Thus, the contrast of every image is normalized through a dedicated normalization layer and all sizes are reduced to $228 \times 228 \times 3$ in another layer. Equation (2) shows a mathematical calculation of the normalization operation.

$$Y_2 = \frac{Y1}{sd} \tag{2}$$

Y_1 is the output of the noise removal or reduction and sd denotes the standard deviation per input. Then, these images are fed along with random false data into a brain, where a generator and discriminator are located. These two components are employed to act as the main backbone of the proposed model to perform all required operations to extract features from the inputs after removing the false data. In order to avoid the overfitting issue, a dropout layer is placed with a value = 0.1. Figure 3 illustrates a structure of the proposed NDGANs, and Fig. 4 depicts an internal architecture of the brain of the presented model.

In Fig. 3, an activation function is applied after the Max-pooling layers. This function is ReLU. SoftMax and Dropout layers are embedded in the last fully connected layer in Fig. 3. Sigmoid refers to the utilized activation function before achieving desired findings. In Fig. 4, after every Max-pooling layer in the generator part, the ReLU function is applied, while the Leaky ReLU activation function is placed after each Max-pooling layer in the discriminator section. In Figs. 3 and 4, a batch normalization operation is used in every convolutional layer.

In general, the generator and the discriminator have 3 blocks in each, where every block contains 2 cascade convolutional levels and one Max-pooling layer. Every true input, which refers to an image, is partitioned into different patches of type RGB. In this study, the augmentation process of colors is not performed, as it degrades accuracy according to Manoj et al. (2022). In addition, all images with small resolutions and brightness are enhanced. The extraction of the required characteristics is performed through twelve layers in the discriminator as depicted in Fig. 4. The balance of width, depth, and resolution in the proposed system is easily achieved without the need for complex interference. Since the flames or smoke can be small parts in some inputs, the dense layers are deployed to detect these small objects. In addition, the learning process is improved throughout the

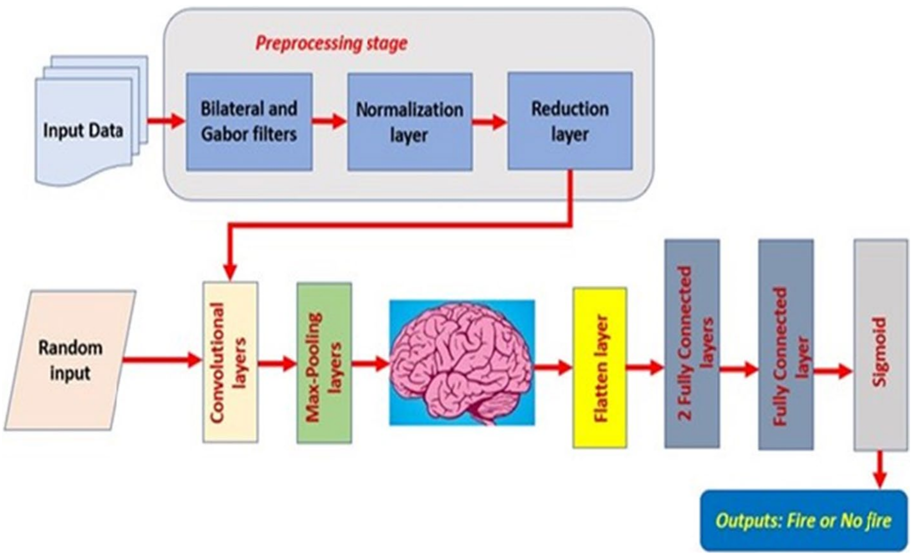

Fig. 3 The structure of the proposed NDGANs model

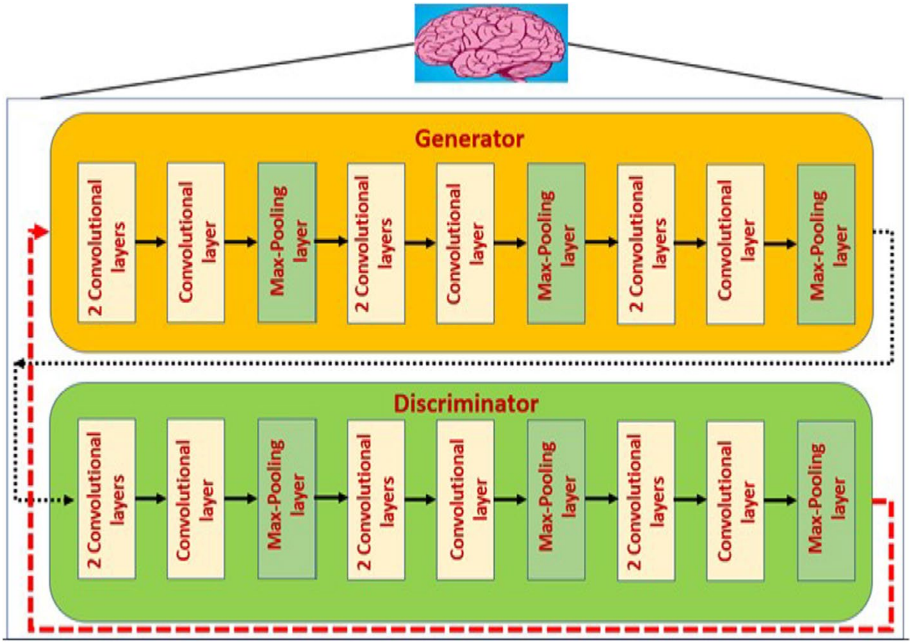

Fig. 4 The internal architecture of the presented brain

entire system, while the implemented layers reduce the network complexity and the overfitting problem. The convolutional layers are deployed to extract the needed features from all inputs. All acquired data are collected in the last convolutional layer.

In Fig. 4, the convolutional layers have different sizes, which form multi-scale convolutional levels. This form intends to simplify the design and reduce the complexity as well. Every convolutional layer is associated with its number of kernels. These kernels ensure that the proposed method gets diverse features. The Max-pooling layers limit the number of parameters that are required for inputs of large size by selecting a maximum value. The utilized Sigmoid activation function controls the gradient changes in inputs. This function is selected due to its accuracy. The 2 fully connected layers include 200 nodes per layer, whereas the last fully connected layer has 2 nodes. These two nodes refer to fire and non-fire labels, respectively. Table 3 lists the utilized hyperparameters of the developed NDGANs model to achieve the required outcomes.

In the proposed algorithm, the dependency between feature maps is utilized to update these maps according to the obtained weights. In addition, the cross-entropy loss (CL) value is computed to evaluate whether an image contains fire or not. According to Dampage et al. (2022), CL is calculated as shown in Eq. (3).

$$\mathrm{CL} = -\sum_{i=1}^{N} qx\log(P) \tag{3}$$

N refers to the number of classes being evaluated, in this study, N = 2. q represents a binary indicator, which is computed in the proposed system and P is the probability. This quantity provides a clear sight of how far the proposed model is from the needed results. Hence, the lower the value, the better results are achieved.

Table 3 Configurations of the implemented algorithm

Name of the layer and parameters	Settings
Batch size	16
Learning rate	0.0001
Number of iterations	3100
Number of epochs	100
Optimizer	Adam
Convolutional layers	64 filters, kernel size 3 × 3, number of strides: 2, padding = same
Max-pooling layer	3 × 3, number of strides: 2
Dropout	0.1
Activation function	Sigmoid
Generator	
Convolutional layers	128 filters, kernel size 4 × 4, number of strides: 3, padding = same
Convolutional layer	128 filters, kernel size 3 × 3, number of strides: 2, padding = same
Max-pooling layer	3 × 3, number of strides: 2
Activation function	ReLU
Convolutional layers	256 filters, kernel size 8 × 8, number of strides: 4, padding = same
Convolutional layer	256 filters, kernel size 5 × 5, number of strides: 3, padding = same
Max-pooling layer	3 × 3, number of strides: 2
Activation function	ReLU
Convolutional layers	512 filters, kernel size 5 × 5, number of strides: 3, padding = same
Convolutional layer	128 filters, kernel size 4 × 4, number of strides: 3, padding = same
Max-pooling layer	3 × 3, number of strides: 2
Activation function	ReLU
Discriminator	
Convolutional layers	512 filters, kernel size 8 × 8, number of strides: 5, padding = same
Convolutional layer	512 filters, kernel size 3 × 3, number of strides: 3, padding = same
Max-pooling layer	3 × 3, number of strides: 2
Activation function	Leaky ReLU
Convolutional layers	256 filters, kernel size 5 × 5, number of strides: 4, padding = same
Convolutional layer	256 filters, kernel size 3 × 3, number of strides: 3, padding = same
Max-pooling layer	3 × 3, number of strides: 2
Activation function	Leaky ReLU
Convolutional layers	96 filters, kernel size 3 × 3, number of strides: 1, padding = same
Convolutional layer	96 filters, kernel size 3 × 3, number of strides: 1, padding = same
Max-pooling layer	3 × 3, number of strides: 2
Activation function	Leaky ReLU
2 fully connected layers	400 nodes
Fully connected layer	2 nodes

Numerous performance metrics are evaluated inside the presented NDGANs; these metrics are:

(1) True Positive (TP).
(2) False Positive (FP).

(3) True Negative (TN).
(4) False Negative (FN).
(5) Precision (PC): is computed as displayed in Eq. (4):

$$\mathrm{PC} = \frac{TP}{(TP + FP)} \tag{4}$$

(6) Sensitivity (SNV): is evaluated as shown in Eq. (5):

$$\mathrm{SNV} = \frac{TP}{(TP + FN)} \tag{5}$$

(7) Accuracy (ACY): is computed using Eq. (6):

$$\mathrm{ACY} = \frac{(TP + TN)}{(TP + TN + FN + FP)} \tag{6}$$

(8) F-score: is determined via Eq. (7):

$$\text{F-Score} = 2 \times \left[\frac{(PCxSNV)}{(PC + SNV)} \right] \tag{7}$$

(9) Dice (DC): is calculated as shown in Eq. (8):

$$\mathrm{DC} = \frac{2TP}{(2TP + FP + FN)} \tag{8}$$

(10) Jaccard Index (JAC): determines an overlap area between the detected fire area and the ground truth label. This quantity is computed as illustrated in Eq. (9) (Manoj et al. 2022):

$$\mathrm{JAC} = \frac{(TL \cap PL)}{(TL \cup PL)} \tag{9}$$

where TL refers to the true label and PL represents the predicted label.

5 Results

In this section, the experiments to evaluate the proposed model, analysis of its results, and any indicators that affect its procedures are provided. In addition, superiority is analyzed as well. Some performance quantities, such as accuracy, are evaluated through five various datasets. To ensure the correspondence between utilized images and their true labels, all datasets were distributed evenly into three groups as stated earlier. The proposed NDGANs method was tested on a MATLAB version. This MATLAB was installed on a hosting machine with Windows Pro. 11. The hosting machine had an Intel Core I7 8th Gen., 16 GB RAM, 64-bit Operating System, and 2 GHz. Tab. All input images were rescaled to 228 × 228. In addition, the ADAM optimizer was used, configured the weight decay to be 0.001, and the momentum parameter was 0.85. The strength of the presented algorithm was evaluated on five datasets that were downloaded from various sources from Thakur and Satish (2021) to https://ieee-dataport.org/open-access/large-scale-dataset-active-fire-detectionsegmentation-landsat-8. The training set has

Table 4 The obtained average values of the performance quantities

Performance quantities	Achieved results (%)
Precision	98.74
Sensitivity	98.64
Accuracy	98.87
F-score	98.69
Dice	98.82
Jaccard Index	98.35

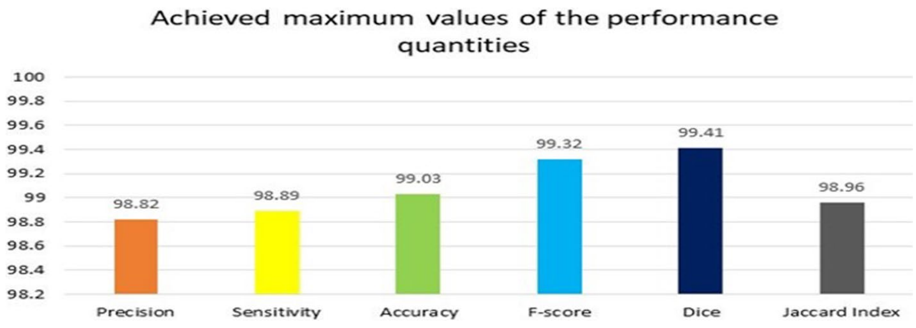

Fig. 5 Maximum achieved results using 100 epochs

25,101 images, as listed in Table 2, and the testing set includes 7172. The evaluated performance quantities of the proposed method are listed in Table 4. All values are average. The system reached 98.87% accuracy and 98.69% f-score. The model was able to detect fires and their components successfully due to the use of local and global feature extraction. This happened because the algorithm segmented pixels appropriately.

The proposed model found and detected all semantic dependencies using the weights of all pixels that are detected inside the Regions of Interest (RoIs). The accuracy level increased from 98.87% to nearly 99.03% when increasing the resolution of the inputs from 228×228 to 646×646. However, this improvement required penalties on the execution time. In addition, the patch size was minimized by half to enhance the execution time by nearly 35%. Figure 5 shows the maximum obtained values of all considered performance quantities by the model and Fig. 6 illustrates the obtained accuracy and loss function by the algorithm. Black dots and dashed lines represent the validation process. This process occurred every 30 iterations for 100 epochs.

The accuracy reached 98.8% and became stable after 60 epochs and its correspondence number of iterations (1783), while the loss function reached nearly 0.142 after 94 epochs. Determining the execution time of every image in seconds, the number of deployed parameters inside the model, and the number of Floating-Point Operations per Second (FLOPS) were considered and evaluated using $228 \times 228 \times 3$ as the input for each image. These evaluations denote the computation complexity that was generated by the algorithm. Both FLOPS and the number of parameters were in millions. Table 5 lists the results of the computation complexity. These results reveal that the model generated massive computations, however, the algorithm attained reasonable accuracy and other performance metrics.

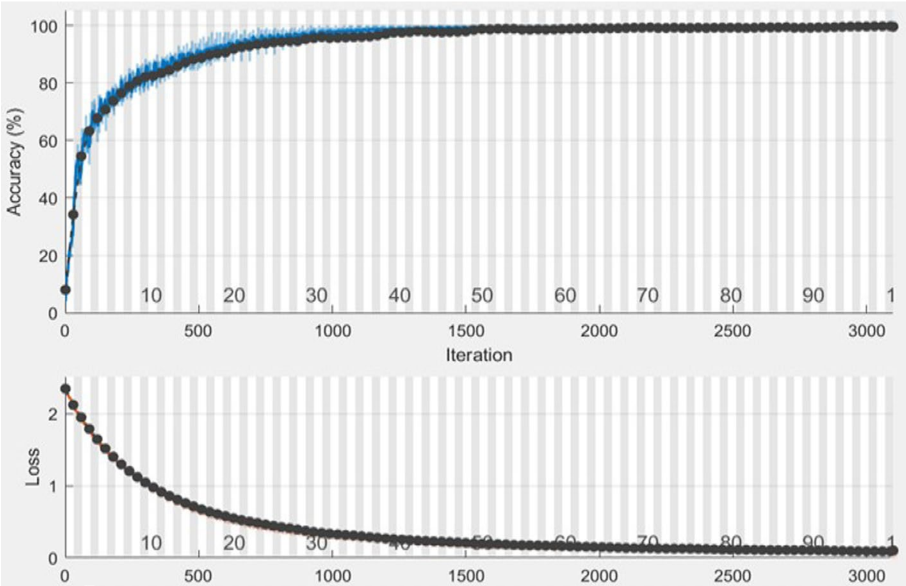

Fig. 6 The achieved accuracy and loss charts

Table 5 Computation complexity results

Pressing time	FLOPS (M)	Number of parameters (M)
6.29 s	33.46	75.81

Figure 7 illustrates some outputs of the proposed method. These outputs are the confusion matrix, error histogram, Mean Squared Error (MSE), and the receiver operating characteristic curve. In the confusion matrix, class 1 refers to the presence of fires or their elements and class 2 denotes that the images were fire free. The confusion matrix represents the obtained results of a sample of the testing set.

Figures 8 and 9 shows examples of the obtained findings. These results indicate that the model segmented fire pixels appropriately. In addition, it distinguished fires or their elements under different circumstances, such as foggy weather or rainy sky properly. Moreover, the algorithm was able to detect and segment the shape of fires. However, some shapes of small fires were missegmented. The presented system generates three results of any input. The first result represents an original image, the second output refers to the initial segmented stage, and the last result denotes the final detected fires.

Table 6 shows the categorization outputs by the system on the testing set. The categorization procedure produces two outcomes: fire or no fire.

The presented algorithm successfully classified 98.8% of the testing set. A comparative analysis between some state-of-the-art algorithms and the presented system is conducted in this research. The comparative analysis includes accuracy and F-score. Table 7 displays the comparative results regarding the considered quantities. These outcomes reveal that the proposed model outperforms other methods in all considered quantities.

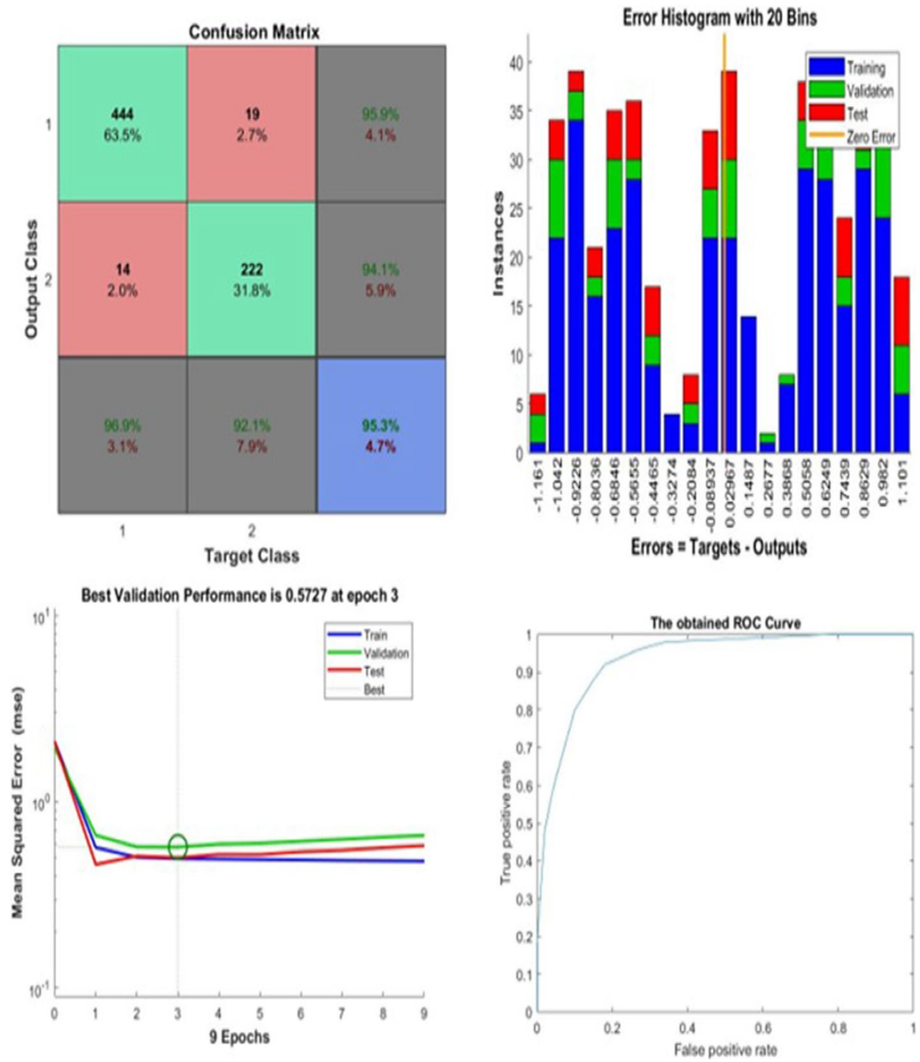

Fig. 7 Samples of the achieved outputs

6 Discussion

The proposed system's performance quantities, which are accuracy, precision, sensitivity, F-score, dice, and Jaccard index were analyzed along with the computation complexity and the processing time. When compared with some existing works in the literature, the proposed GANs approach achieved better results for accuracy and f-score than the developed method in Ghali et al. (2022). The average obtained outcomes were 98.87% and 98.69%, respectively, while in Ghali et al. (2022) the authors reached 85.12% accuracy. In addition, five datasets were employed in the presented approach, and the utilized activation function was Leaky ReLU. The obtained results show that the

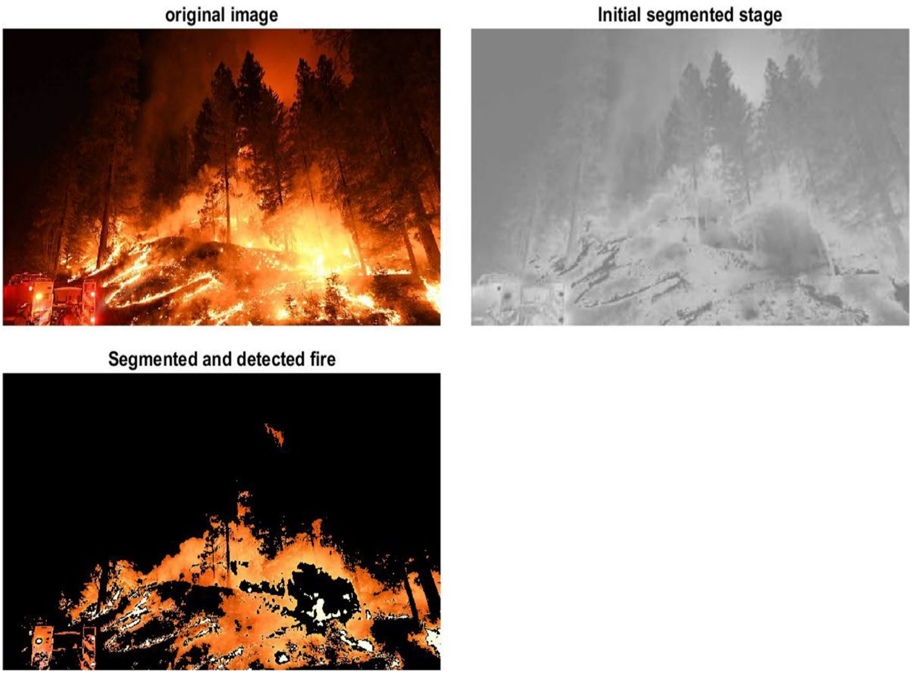

Fig. 8 The achieved results of a sample fire image

presented model offers better findings than the implemented algorithm in Ghali et al. (2022). In (Ghali et al. 2021), the authors attained 96.76% F-score, while the proposed model reached 98.87%, which is better. The works reported in Shahid and Hua (2021), Zhang et al. 2020), and (Abdusalomov et al. 2021) reached an accuracy between 92% and 98.3% and 98.08% F-score. In contrast, NDGANs obtained better results. Therefore, this model surpasses other state-of-the-art works and generates better outcomes. In addition, the presented model was compared with other developed methods as shown in Table 7. This comparison implies that the proposed model yields better results regarding accuracy and F-score except for the implemented model in Hong and Choi (2012), which achieved better outcomes (99.12%) regarding accuracy. In addition, the developed methods in Dampage et al. (2022), Dai et al. (2022), and Goodfellow et al. (2014) achieved the lowest accuracy, while the models in Regi et al. (2018), Chopde et al. (2022), and Gharge et al. (2014) reached moderate accuracy.

The proposed system proved its capability in detecting fires or their elements effectively and properly in various conditions. In addition, the model detects the perfect shape of areas that are covered by fires. Furthermore, the model obtained high results with 99.03% accuracy and 99.32% f-score, and this occurred because the method extracted a high level of information using pixels. The conducted analysis of the obtained findings revealed that the presented algorithm minimized pixel misprediction and provided reasonable outputs for detection purposes. However, the algorithm failed in detecting small fires and treated red and orange colors as true fires. Another drawback of the model is that it relies on a pre-trained network. This network requires huge computation times, and this time could be deadly and catastrophic. The model was able to segment fires properly even if detected objects had similar color intensities or movements, such as pixels in the background.

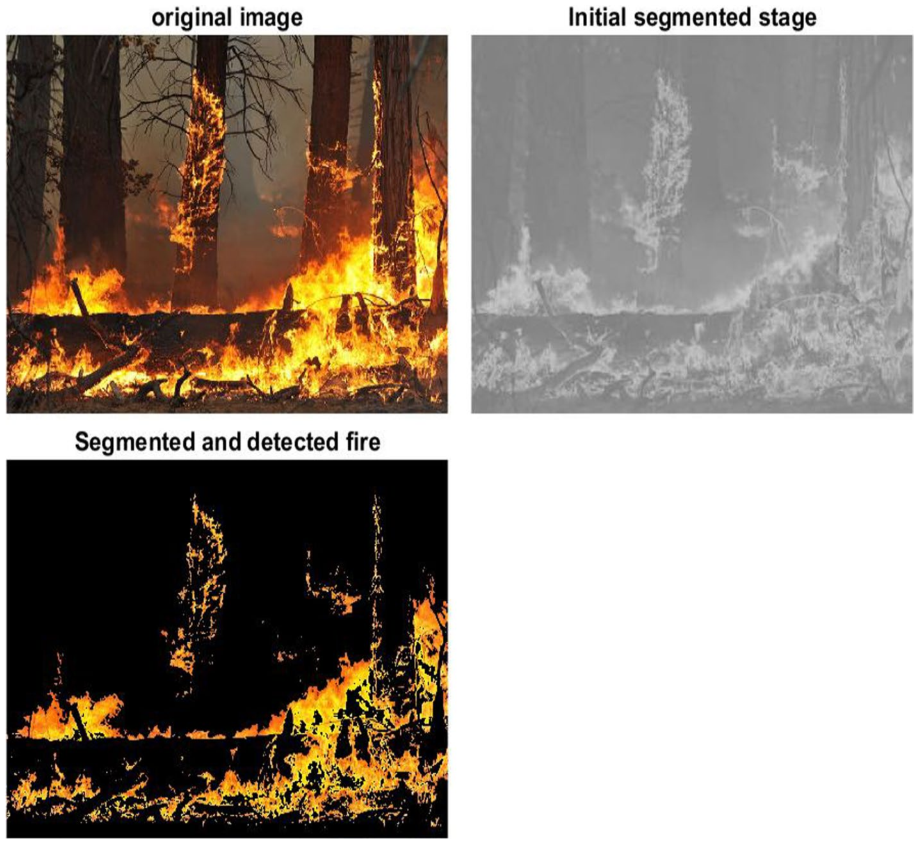

Fig. 9 The achieved results of another sample fire image

Table 6 The classification results by NDGANs on the applied datasets

Dataset number	Detected fires, smoke, flames	No fire
1	1145	865
2	2713	537
3	671	203
4	412	117
5	307	202

To conclude, the proposed algorithm proved its efficiency and acceptable performance through five datasets. In addition, it has the ability to locate fires precisely throughout its segmentation and determine the shapes as well.

Table 7 The assessment evaluation outputs

Works	Applied technology	Accuracy	F-score
Dampage et al. (2022), 2022	Deep ensemble learning method: EfficientNet-B5 and DenseNet-201, and two vision transformers	85.12%	84.77%
Dampage et al. (2022), 2021	Two vision transformers: TransUNet and MedT	N/A	96.76%
Regi et al. (2018), 2021	Vision transformer network	95.3%	N/A
Chopde et al. (2022), 2021	A YOLO network and attention mechanism	94.04%	98.08%
Hong and Choi (2012), 2021	A Convolutional Neural Network (CNN) and spatial self-attention mechanism	99.12%	N/A
Manoj et al. (2022), 2021	A YOLOv3 network	98.1%	99.5
Dai et al. (2022), 2022	A Fire-YOLO network	91.5%	71%
Gharge et al. (2014), 2022	Regional-based CNN	97.29%	N/A
Goodfellow et al. (2014), 2022	Multiple deep-learning tools	85.43%	N/A
The proposed method	New Dense Generative Adversarial Networks (NDGANs)	99.03%	99.32%

7 Conclusions

In this article, a new Novel Dense Generative Adversarial Network (NDGAN) system is proposed. This model works based on extracting the required features through deep-learning technology. It was tested using various fire datasets. The conducted simulation experiments on MATLAB revealed that the proposed algorithm was feasible and trustworthy in detecting fires and their elements. In addition, the model attained high performance regarding all considered quantities. The algorithm was explored using different datasets to spot fires using pixels. Numerous input sizes were applied to determine the best performance. In addition, an assessment analysis was performed to compare the presented approach and some developed models according to the technologies being applied, achieved accuracy, and F-score. The proposed approach minimized the false negative and false positive metrics. The presented approach surpasses all developed works in literature. The achieved accuracy by this method ranges from 98.8% to 99.03%.

The model, unfortunately, misdetected some small fires and faces issues when dealing with red and orange colors. These two colors lead to improper detection results. In addition, high computations occur, and this problem can be resolved or overridden using a dedicated Graphical Processing Unit (GPU).

Future work is planned to improve the detection accuracy of small fires and remove the issue of red and orange colors. Moreover, reducing the computation complexity is another objective of the planned future work.

Acknowledgements The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FFR-2023-0094".

Author contributions Conceptualization, A.A.A. and T.S..; data curation, T.S.; formal analysis, A.A.A. and T.S.; funding acquisition, T.S.; investigation, T.S.; methodology, A.A.A.; project administration, T.S.; resources, A.A.A. and T.S.; software, A.A.A.; supervision, T.S.; validation, A.A.A.; visualization, T.S.; writing—original draft, A.A.A.; writing—review and editing, T.S. All authors have read and agreed to the published version of the manuscript.

Funding This work was funded by the Deanship of Scientific Research at Northern Border University, Arar, KSA under grant No. "NBU-FFR-2023-0094".

Data availability The utilized datasets were downloaded from public websites and are available at the following links: https://www.kaggle.com/datasets/phylake1337/fire-dataset, https://images.cv/download/flames/1219/CALL_FROM_SEARCH/%22flames%22, https://github.com/gaiasd/DFireDataset, https://ieee-dataport.org/open-access/large-scale-dataset-active-fire-detectionsegmentation-landsat-8, https://www.kaggle.com/datasets/dataclusterlabs/fire-and-smoke-dataset

Declarations

Conflict of interest The authors declare no conflicts of interest.

Informed consent Not applicable.

Institutional review board statement Not applicable.

References

Abdusalomov A, Baratov N, Kutlimuratov A, Whangbo TK (2021) An improvement of the fire detection and classification method using YOLOv3 for surveillance systems. Sensors 21:1–18

Chopde A, Magon A, Bhatkar S (2022) Forest fire detection and prediction from image processing using RCNN," *Proceedings of the 7th World Congress on Civil, Structural, and Environmental Engineering*, pp. 1–8

Dai P, Zhang Q, Lin G, Shafique MM, Huo Y, Tu R, Zhang Y (2022) Multi-scale video flame detection for early fire warning based on deep learning. Front Energy Res. https://doi.org/10.3389/fenrg.2022.848754

Dai Z, Almahairi A, Bachman P, Hovy E, Courville A (2017) Calibrating energy-based generative adversarial networks, arXiv:1702.01691v2, pp. 1–18

Dampage U, Banadaranayake L, Wanasinghe R, Kottahachchi K, Jayasanka B (2022) Forest fire detection system using wireless sensor networks and machine learning. Sci Rep 12(46):1–11

Ghali R, Jmal M, Mseddi WS, Attia R (2018) Recent advances in fire detection and monitoring systems: a review," Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT'18), vol. 1, pp. 332–340

Ghali R, Akhloufi MA, Jmal M, Mseddi WS, Attia R (2021) Wildfire segmentation using deep vision transformers. Remote Sensing 13(3527):1–24

Ghali R, Akhloufi MA, Mseddi WS (2022) Deep learning and transformer approaches for UAV-based wildfire detection and segmentation. Sensors 22(1977):1–18

Gharge S, Birla S, Pandey S, Dargad R, Pandita R (2014) Smoke and fire detection. Int J Sci Res Publ 4(7):1–15

Goodfellow IJ, Abadie JP, Mirza M, Xu B, Farley DW, Ozair S, Courville A, Benigo Y (2014) Generative Adversarial Nets, arXiv:1406.2661v1, pp. 1–9

Hong SH, Choi MS (2012) An intelligent fire detection algorithm for fire detector. Int J Saf 11(1):6–10

https://images.cv/download/flames/1219/CALL_FROM_SEARCH/%22flames%22.

https://www.kaggle.com/datasets/phylake1337/fire-dataset.

https://github.com/gaiasd/DFireDataset.

https://ieee-dataport.org/open-access/large-scale-dataset-active-fire-detectionsegmentation-landsat-8.
https://www.kaggle.com/datasets/dataclusterlabs/fire-and-smoke-dataset.
Khan F, Xu Z, Sun J, Khan FM, Ahmed A, Zhao Y (2022) Recent advances in sensors for fire detection. Sensors 22(3310):1–24
Kumar R, Goyal A, Courville A, Benigo Y (2019) Maximum entropy generators for energy-based models, arXiv:1901.08508v1, pp. 1–10
Manoj K, Suniti P, Karthik KMD, Kumar KN, Vamsi KR, Varma KVR (2022) Smoke and fire detection using deep learning: a review. Int J Adv Res Sci Commun Technol 2(1):312–322
Morizet N (2020) Introduction to generative adversarial networks, Technical Report, Advestis, hal-0289937, pp. 1–9
Niknejad M, Bernardino A (2021) Attention on classification for fire segmentation, arXiv:2111.03129v1, pp. 1–6
Regi M, Varghese RG, Sidharth V, Thomas J (2018) Deep learning based fire detection system. Int J Knowl Based Comput Syst 6(1):18–22
Shahid M, Hua KL (2021) Fire detection using transformer network, ICMR '21: Proceedings of the 2021 International Conference on Multimedia Retrieval, pp. 627–630
Singh YK (2021) Detection of fire regions using RGB color variance. Int J Res Eng Sci 9(6):10–14
Thakur A, Satish M (2021) Generative adversarial networks. Int J Res Appl Sci Eng Technol 9(8):2307–2325
Wang S (2017) Generative adversarial networks (GAN) a gentle introduction, Tutorial on GAN in LIN395C: Research in Computational Linguistics, University of Texas at Austin, pp. 1–11
Yandouzi M, Grari M, Idrissi I, Boukabous M, Moussaoui O, Azizi M, Ghoumid K, Elmiad AK (2022) Forest fires detection using deep transfer learning. Int J Adv Comput Sci Appl 13(8):268–275
Zhai S, Cheng Y, Feris R, Zhang Z (2016) Generative adversarial networks as variational training of energy based models, arXiv:1611.01799v1, pp. 1–13
Zhang X, Qian K, Jing K, Yang J, Yu H (2020) Fire detection based on convolutional neural networks with channel attention, 2020 Chinese Automation Congress (CAC), Shanghai, China, pp. 3080–3085
Zhao L, Zhi L, Zhao C, Zheng W (2022) Fire-YOLO: a small target object detection method for fire inspection. Sustainability 14(4930):1–14
Zhao F (2022) Application research of image processing technology for fire detection and fire alarm based on blockchain, Mobile Information Systems, 2022, Article ID 9304991, pp. 1–11